

32

# SH-2E SH7058 F-ZTAT™

Hardware Manual

Renesas SuperH™ RISC engine

Hardware Manual

Rev. 3.00  
Revision date: Sep. 17, 2004

RenesasTechnology  
[www.renesas.com](http://www.renesas.com)

## Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.  
Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

## Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.  
The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.  
Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.  
Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

## 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 they are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

### 2. Treatment of Unused Input Pins

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

### 3. Processing before Initialization

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

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

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

# Preface

The SH7058 is a single-chip RISC (reduced instruction set computer) microcomputer that has the 32-bit internal architecture CPU, SH-2E, as its core, and also includes peripheral functions necessary for system configuration.

The SH7058 is equipped with on-chip peripheral functions necessary for system configuration, including a floating-point unit (FPU), large-capacity ROM and RAM, a direct memory access controller (DMAC), timers, a serial communication interface (SCI), controller area network (HCAN), A/D converter, and I/O ports, therefore, it can be used as a microprocessor built in a high-level control system.

The SH7058 is an F-ZTAT™\* (Flexible Zero Turn-Around Time) version with flash memory as its on-chip ROM, and it can rapidly and flexibly deal with each situation on an application system with fluid specifications from an early stage of mass production to full-scale production.

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

**Target users:** This manual was written for users who will be using the SH7058 F-ZTAT in the design of application systems. Users of this manual are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

**Objective:** This manual was written to explain the hardware functions and electrical characteristics of the SH7058 F-ZTAT to the above users.  
Refer to the SH-2E Programming 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-2E Programming Manual.

**Rule:** Bit order: The MSB (most significant bit) is on the left and the LSB (least significant bit) is on the right.

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

SH7058 F-ZTAT manuals:

<b>Manual Title</b>	<b>Document No.</b>
SH7058 F-ZTAT Hardware Manual	This manual
SH-2E Programming Manual	

Users manuals for development tools:

<b>Manual Title</b>	<b>Document No.</b>
SH Series C/C++ Compiler, Assembler, Optimizing Linkage Editor User's Manual	ADE-702-246
SH Series Simulator/Debugger (for Windows) User's Manual	ADE-702-186
SH Series Simulator/Debugger (for UNIX) User's Manual	ADE-702-203
High-performance Embedded Workshop User's Manual	ADE-702-201

Application note:

<b>Manual Title</b>	<b>Document No.</b>
C/C++ Compiler	

# Main Revisions for this Edition

Item	Page	Revisions (See Manual for Details)
1.2 Block Diagram	7	Corrected errors
Figure 1.1 Block Diagram		
1.3.1 Pin Arrangement	9	Newly added
Figure 1.3 Pin Assignments		
1.3.2 Pin Functions	10-18	BP-272 added
Table 1.2 Pin Functions		
1.3.3 Pin Assignments	19-27	BP-272 added
Table 1.3 Pin Assignments		
7.1.1 Features	101	<ul style="list-style-type: none"> <li>Notification of interrupt occurrence can be reported externally (IRQOUT pin)</li> </ul> <p>For example, it is possible to request the bus if an external bus master is informed that an on-chip peripheral module interrupt request has occurred when the chip has released the bus.</p>
7.4.1 Interrupt Sequence	120	Note amended
Figure 7.2 Interrupt Sequence Flowchart		<ol style="list-style-type: none"> <li>As <math>\overline{\text{IRQOUT}}</math> is synchronized with a peripheral clock <math>\text{P}\phi</math>, it may be output later than a CPU interrupt request.</li> </ol>

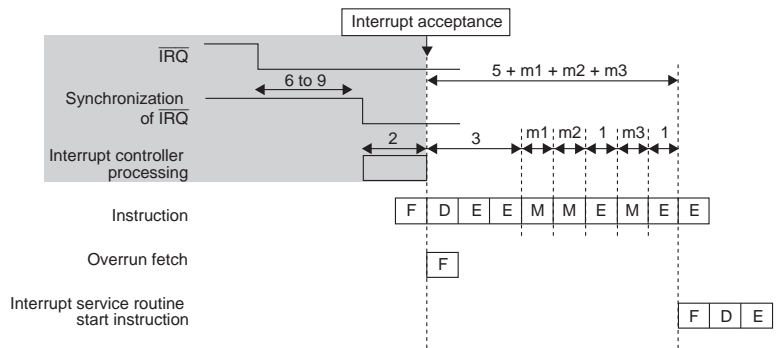
7.5 Interrupt Response Time 122 Table amended

Table 7.5 Interrupt Response Time

Item	Peripheral Module	Number of States		Notes
		NMI	IRQ	
Synchronizing input signal (synchronized with peripheral clock P <sub>φ</sub> ) with internal clock φ and DMAC activation judgment	0 or 6 [0 or 3]	1 to 4 [1 or 2]	6 to 9 [3 to 5]	For the number of states required for each interrupt, see the note (*) below. The values enclosed in [ ] are values for when the multiplication ratio is 4.
Compare identified interrupt priority with SR mask level	2	2	2	
Wait for completion of sequence currently being executed by CPU	X (≥ 0)			The longest sequence is for interrupt or address-error exception processing (X = 4 + m1 + m2 + m3 + m4). If an interrupt-masking instruction follows, however, the time may be even longer.
Time from start of interrupt exception processing until fetch of first instruction of exception service routine starts	5 + m1 + m2 + m3			Performs the PC and SR saves and vector address fetch.
Interrupt response time	Total:	(7 or 13) + m1 + m2 + m3 + X	(8 to 11) + m1 + m2 + m3 + X	(13 to 16) + m1 + m2 + m3 + X
	Minimum:	10	11	16
	Maximum:	17 + 2 (m1 + m2 + m3) + m4	15 + 2 (m1 + m2 + m3) + m4	20 + 2 (m1 + m2 + m3) + m4
<p>Note: * Number of states needed for synchronization and DMAC activation judgment                      The relations between numbers of states needed for synchronizing an input signal (synchronized with the peripheral clock P<sub>φ</sub>) with the internal clock φ and DMAC activation judgment and vector numbers are shown below.                      0 state: 9, 10, 12, 13, 14, 72, 74, 76, 78, 189, 193, and 224                      6 states: Peripheral module interrupts other than the above. However, vector number 222 (HCAN0/RM0) is different from the others.                      For an interrupt with vector number 222 (HCAN0/RM0), the needed states differ from other interrupts since the interrupt by HCAN0 mailbox 0 can activate the DMAC.                      HCAN0 mailbox 0: 7 states                      Other than above: 6 states                      The same number of states is needed to cancel interrupt sources.                      If the necessary number of states is not secured after flag clear of the interrupt source, the interrupt may occur again.</p>				

7.5 Interrupt Response Time 123 Figure amended

Figure 7.4 Example of Pipeline Operation when an IRQ Interrupt is Accepted



9.1.5 Address Map 146 Newly added

Table 9.3 Address Map

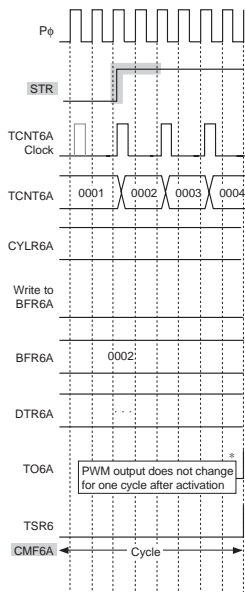
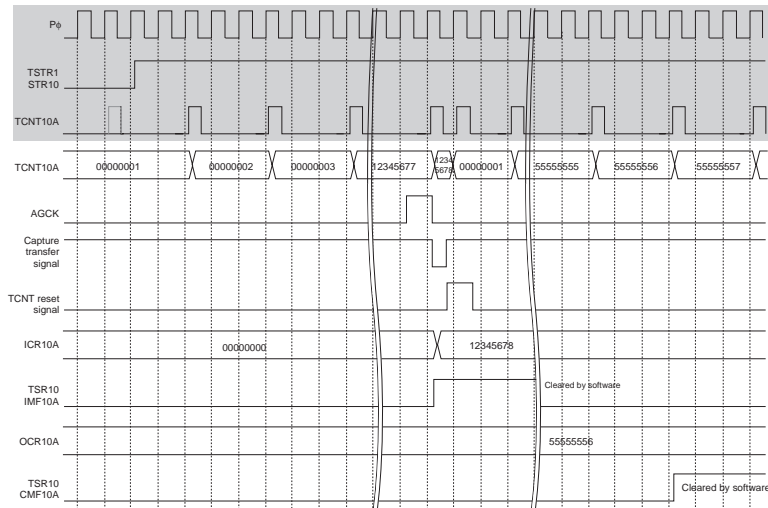
- Number of Access Cycles for On-Chip Peripheral Module Registers

10.3.2 DMA Transfer Requests 179 Description added

In on-chip peripheral module request mode, when the DMAC accepts the transfer request, the next transfer request is ignored until a single transfer ends in cycle steal mode or all transfers end in burst mode. Only when the address reload function is used, the next transfer request is accepted after the fourth transfer.

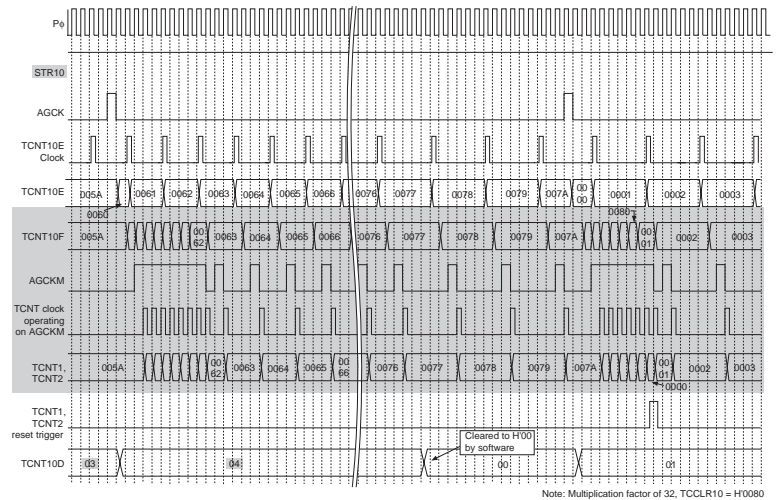
Item	Page	Revisions (See Manual for Details)																				
10.3.2 DMA Transfer Requests	180,	Table amended																				
Table 10.2 Selecting On-Chip Peripheral Module Request Modes with the RS Bits	182	<table border="1"> <thead> <tr> <th>RS4</th> <th>RS3</th> <th>RS2</th> <th>RS1</th> <th>RS0</th> <th>DMAC Transfer Request Source</th> <th>DMAC Transfer Request Signal</th> <th>Transfer Source</th> <th>Transfer Destination</th> <th>Bus Mode</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>HCAN0</td> <td>RM0 (HCAN0 receive interrupt)</td> <td>MB0-MB15</td> <td>Don't care*</td> <td>Burst/cycle-steal</td> </tr> </tbody> </table>	RS4	RS3	RS2	RS1	RS0	DMAC Transfer Request Source	DMAC Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode	0	0	0	1	1	HCAN0	RM0 (HCAN0 receive interrupt)	MB0-MB15	Don't care*	Burst/cycle-steal
RS4	RS3	RS2	RS1	RS0	DMAC Transfer Request Source	DMAC Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode													
0	0	0	1	1	HCAN0	RM0 (HCAN0 receive interrupt)	MB0-MB15	Don't care*	Burst/cycle-steal													
		MB0-MB15: HCAN0 message data																				
10.3.11 DMAC Access from CPU	193	<p>Description amended</p> <p>The space addressed by the DMAC is 4-cycle space. Therefore, when the CPU becomes the bus master and accesses the DMAC, a minimum of four internal clock cycles (<math>\phi</math>) are required for one bus cycle. Also, since the DMAC is located in word space, while a word-size access to the DMAC is completed in one bus cycle, a longword-size access is automatically divided into two word accesses, requiring two bus cycles (eight basic clock cycles). These two bus cycles are executed consecutively; a different bus cycle is never inserted between the two word accesses. This applies to both write accesses and read accesses.</p>																				
11.2.21 Offset Base Registers (OSBR)	341	<p>Bit table amended</p> <p>Dedicated input capture registers with the same input trigger signal as that for channel 0 ICR0A</p> <p>Description amended</p> <p>OSBR1 and OSBR2 are 16-bit read-only registers used exclusively for input capture. Same as the channel 0 input capture register (ICR0A), OSBR1 and OSBR2 use the TIOA input as their trigger signal, and store the TCNT1A or TCNT2A value on detection of an edge.</p>																				
Offset Base Registers 1 and 2 (OSBR1, OSBR2)																						
11.3.1 Overview	366	<p>Description amended</p> <p>Channel 10: (3) Multiplied clock correction block</p> <p>...When the 16-bit correction counter 10F (TCNT10F) value exceeds that in 16-bit correction counter 10E (TCNT10E), no count-up operation is performed.</p>																				
11.3.2 Free-Running Counter Operation and Cyclic Counter Operation	367	<p>Figure amended</p>																				
Figure 11.13 Free-Running Counter Operation and Overflow Timing																						



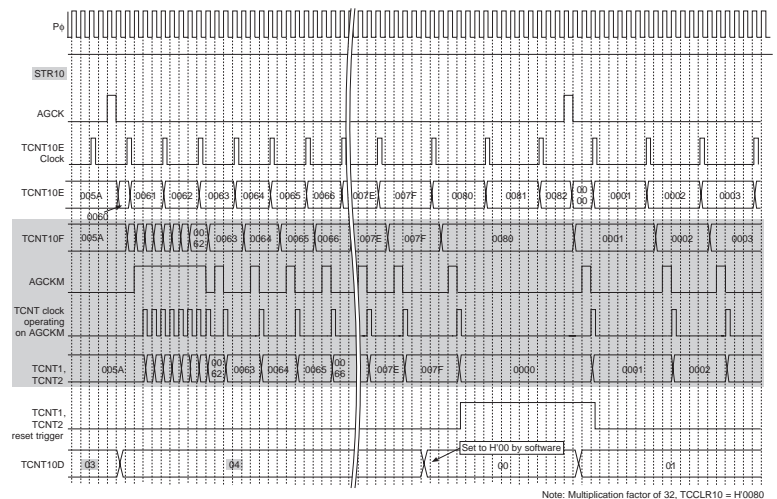
Item	Page	Revisions (See Manual for Details)
11.3.8 Twin-Capture Function	373	Description amended When TCNT0, TCNT1A, and TCNT2A in channel 0, channel 1, and channel 2 are started by a setting in the timer start register (TSTR), and an edge of <b>TIOA</b> input (a trigger signal) is detected, the TCNT1A value is transferred to OSBR1, and the TCNT2A value to OSBR2.
11.3.9 PWM Timer Function Figure 11.21 PWM Timer Operation	375	Figure amended 
11.3.9 PWM Timer Function Figure 11.22 Complementary PWM Mode Operation	376	Figure replaced
11.3.12 Channel 10 Functions Figure 11.28 TCNT10A Capture Operation and Compare-Match Operation	381	Figure amended 
Multiplied Clock Generation Function: Figure 11.30 TCNT10C Operation	383	Description amended TST10→ <b>STR10</b>

Item	Page	Revisions (See Manual for Details)
Multiplied Clock Correction Function: Figure 11.32 TCNT10D Operation Figure 11.33 TCNT10E Operation Figure 11.34 TCNT10F Operation (At Startup)	384, 385	Description amended TST10→STR10 When the TCNT10F value exceeds the TCNT10E value, no count-up operation is performed.

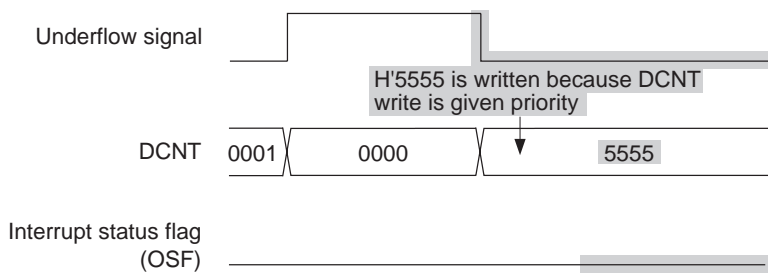
Multiplied Clock Correction Function:  Figure 11.35 TCNT10F Operation (End of Cycle, Acceleration, Deceleration)	386	Description amended TST10→STR10 Figure amended
--	-----	--



Multiplied Clock Correction Function: Figure 11.36 TCNT10F Operation (End of Cycle, Steady-State)	387	
--	-----	--



11.7 Usage Notes Contention between DCNT Write and Counter Clearing by Underflow:	420	Note amended Note: In the SH7055, a write to DCNT from the CPU is not attempted, but retention of H'0000 takes precedence. Note that its operation is different
--	-----	--

Item	Page	Revisions (See Manual for Details)												
11.7 Usage Notes Figure 11.72 Contention between DCNT Write and Underflow	420	Figure amended 												
12.1.4 Register Configuration Table 12.2 Advanced Pulse Controller Register	429	Note amended Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.												
14.1.3 Register Configuration Table 14.1 Register Configuration	453	Note amended Notes: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles for byte access and word access, and eight or nine internal clock ( $\phi$ ) cycles for longword access.												
15.1.4 Register Configuration Table 15.2 Registers	467	Note amended Notes: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles for byte access and word access, and eight or nine internal clock ( $\phi$ ) cycles for longword access.												
Section 16 Controller Area Network-II (HCAN-II)	519-616	Description amended Register name <table border="1" data-bbox="771 1207 1526 1396"> <thead> <tr> <th>Before</th> <th>After</th> </tr> </thead> <tbody> <tr> <td>HCAN-II_bit configuration register</td> <td>HCAN-II_bit timing configuration register</td> </tr> <tr> <td>Transmit wait register</td> <td>Transmit Pending Request Register</td> </tr> <tr> <td>Transmit wait cancel register</td> <td>Transmit Cancel Register</td> </tr> <tr> <td>Receive complete register</td> <td>Data Frame Receive Pending Register</td> </tr> <tr> <td>Remote request register</td> <td>Remote Frame Receive Pending Register</td> </tr> </tbody> </table>	Before	After	HCAN-II_bit configuration register	HCAN-II_bit timing configuration register	Transmit wait register	Transmit Pending Request Register	Transmit wait cancel register	Transmit Cancel Register	Receive complete register	Data Frame Receive Pending Register	Remote request register	Remote Frame Receive Pending Register
Before	After													
HCAN-II_bit configuration register	HCAN-II_bit timing configuration register													
Transmit wait register	Transmit Pending Request Register													
Transmit wait cancel register	Transmit Cancel Register													
Receive complete register	Data Frame Receive Pending Register													
Remote request register	Remote Frame Receive Pending Register													
16.1.1 Features	519, 520	Description amended <ul style="list-style-type: none"> <li>Supports CAN specification 2.0A/2.0B and ISO-11898-1</li> </ul> Description deleted <ul style="list-style-type: none"> <li>Flexible interrupt structure</li> <li>Read section 16.8, Usage Notes carefully.</li> </ul> The following features have been added in the HCAN-II. <ul style="list-style-type: none"> <li>IRR0 function to notify a software reset and halt</li> </ul>												

Item	Page	Revisions (See Manual for Details)
16.1.1 Features	520	<p>Description deleted</p> <ul style="list-style-type: none"> <li>• Halt mode status bit and error passive status bit added to GSR.</li> </ul> <p>■</p> <ul style="list-style-type: none"> <li>• Supports various test modes</li> </ul>
16.2.2 Each Block Function	522	Description deleted
(1) Microprocessor Interface (MPI)		The MPI allows communication between the host CPU and the HCAN's registers/mailboxes to control the memory interface, and the data controller, etc.
(2) Mailboxes		<p>Description deleted</p> <ul style="list-style-type: none"> <li>• CAN message data (for CAN data frames)</li> </ul> <p>■</p> <ul style="list-style-type: none"> <li>• Local acceptance filter mask (LAFM) during reception</li> </ul> <p>■</p> <ul style="list-style-type: none"> <li>• 3-bit mailbox configuration, automatic transmit bit for remote request, new message control bit</li> </ul>
(4) Timer		<p>Important: added</p> <p><b>Important:</b> The timer function is not supported by the SH7058.</p>
16.3.1 Mailbox Configuration	526	<p>Note amended</p> <p>Note: The message control (STDID/EXTID/RTR/ZDE), timestamp, and LAFM/transmission trigger time fields can only be accessed in word size (16 bits), whereas the message control (NMC/ATX/MBC/DLC) and the message data area can be accessed in word (16-bit) or byte (8-bit) size. Also, when the setting of the MBC bits makes the mailbox inactive, all settings other than the MBC bits must be initialized to 0 because an unused mailbox affects the RAM configuration. When the LAFM is not used to receive messages, it must be cleared to 0.</p>
16.3.1 Mailbox Configuration	528	Note amended
Figure 16.3 Mailbox-N Configuration		Notes: 1. All bits shadowed in gray are reserved and the write value should be 0. The value read as the initial value is not guaranteed.

## 16.3.2 Message Control Field

530- Table amended

532

Register Name	Address	Bit	Bit Name	Description
MBx[4], MBx[5]*	H'104 + N x 32	15	CCM	CAN-ID Compare Match When this bit is set, message reception in the corresponding mailbox can generate two triggers. If TCR9 is set to 1, TCR14 is cleared to freeze ICR0. If TCR10 is set to 1, TCNTR (timer counter register) is automatically cleared and the LOSR (local offset register) value is set. <b>Important:</b> This function is not supported by the SH7058. Thus the write value should be 0.
		14	TTE	Time Trigger Enable When this bit is set, a mailbox in which TXPR has been already set transmits a message at a time set in the Tx trigger time field. <b>Important:</b> If this bit is set, a failure occurs during message transmission. Therefore setting is prohibited. The write value should be 0. The value read as the initial value is not guaranteed.
MBx[4], MBx[5]*	H'104 + N x 32	13	NMC	New Message Control When this bit is cleared, a mailbox in which PXPR/PFPR has been already set does not store the new message but retains the previous one and sets the UMSR corresponding bit. When this bit is set, a mailbox in which PXPR/PFPR has been already set stores the new message and sets the UMSR corresponding bit. If a message is received in a mailbox in overwrite mode (NMC = 1), the host CPU must perform an additional check at the end of the data reading from the mailbox in order to guarantee that the mailbox data have not been corrupted during such operation by another receive message. The additional check, to be performed at the end of the mailbox access, consists in verifying that the associated bit of UMSR has not been set and so no overwrite has occurred; in case such bit is set data have been corrupted and so the message must be discarded.
MBx[4], MBx[5]*	H'104 + N x 32	11	DART	Disable Automatic Retransmission 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. When this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is cleared, the HCAN tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR. <b>Important:</b> This function is not supported by the SH7058. Thus the write value should be 0. The value read as the initial value is not guaranteed.

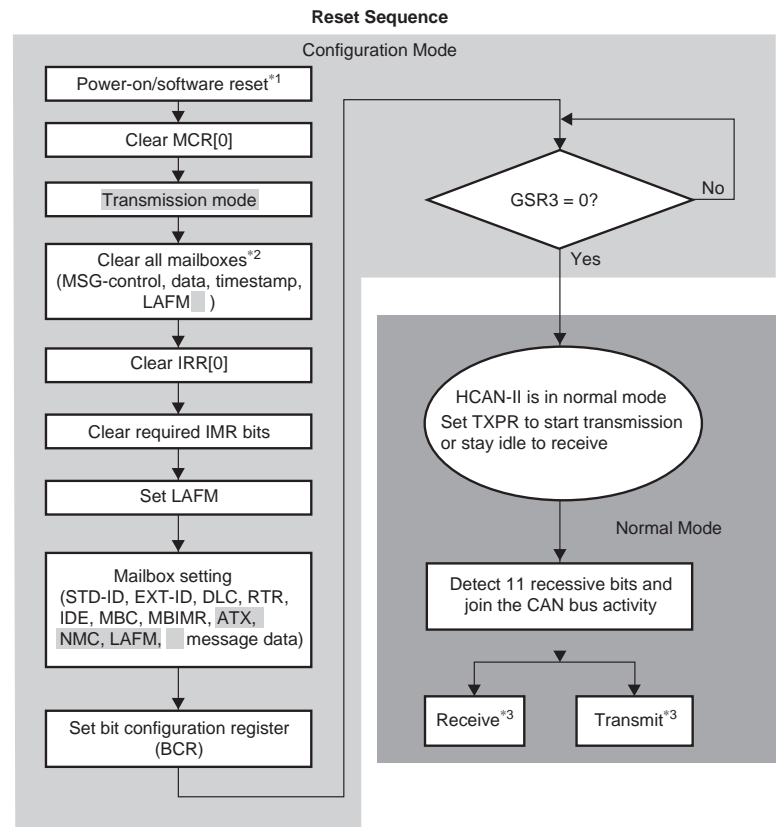
Item	Page	Revisions (See Manual for Details)				
16.3.2 Message Control Field	533- 534, 537	MBx[4], MBx[5]*	H'104 + N x 32	6	TCT	<p><b>Timer Counter Transfer</b> When this bit is set, a mailbox is set for transmission, and the DLC is set to 4, the TCNTR value, at the SOF, is embedded in the second and third bytes of the message data, instead of MSG_DATA_2 and MSG_DATA_3, and the CYCLE_COUNT in the first byte instead of MSG_DATA_0[3:0] when this mailbox starts transmission.</p> <p>This function will be useful when the HCAN performs a time master role to transmit the time reference message. For example, considering that two HCAN controllers are connected in the same network and that the receiver stores the message in mailbox N, the data format is shown as figure 16.4 depending on the endian setting for the CAN bus (MCR4).</p> <p><b>Important:</b> This function is not supported by the SH7058. Thus the write value should be 0.</p> <p>The value read as the initial value is not guaranteed.</p>
		MBx[4], MBx[5]*	H'104 + N x 32	5	CBE	<p><b>CAN Bus Error</b> An external fault-tolerant CAN transceiver can be used together with the HCAN module. If the error output pin of the transceiver (normally active low) is connected to the CAN_NERR pin of this LSI, the value of the CAN_NERR pin is stored into this bit at the end of each transmission/reception (if the message is stored). The inverted value of the CAN_NERR pin is set to this bit. If the error output pin is active high, the setting value is not inverted. When this bit is set, it indicates a potential physical error with the CAN bus. As the CAN_NERR value is updated after the transmission or reception in the corresponding mailbox, non-interrupt is dedicated to this function but instead the normal transmit end interrupt (IRR6) and normal receive end interrupt (IRR2) should be considered.</p> <p><b>Important:</b> This function is not supported by the SH7058. Thus the write value should be 0. The value read as the initial value is not guaranteed.</p>
					4	CLE
		MBx[6]*	H'106 + N x 32	15	TimeStamp to 0 [15:0]	<p><b>Message Reception:</b> During message reception, when the SOF or EOF is detected, ICR1 (input capture register 1) always captures the TCNTR (timer counter register) value or the value of Cycle_Counter + TCNTR[15:4], depending on the value of bit 3 in TMR (Timer mode register), at either SOF or EOF depending on the value in TCR13 (timer control register), and the ICR1 value is stored into the timestamp field of the corresponding mailbox.</p> <p><b>Important:</b> Capturing at the SOF is not supported by the SH7058. Thus TCR13 should be set to EOF detection mode.</p> <p><b>Message Transmission:</b> During message transmission, the TCNTR (timer counter register) value or the value of Cycle_Counter + TCNTR[15:4], depending on the value of bit 3 in TMR (timer mode register) is captured when either the TXPR bit or TXACK bit is set depending on the value in TCR12, and the captured value is stored into the timestamp field of the corresponding mailbox.</p> <p><b>Important:</b> Capturing when the TXPR bit is set is not supported by the SH7058. Activation of the TCNR (timer) causes a problem in the SH7058 (timer usage is prohibited). Therefore, the timestamp function is not supported.</p> <p>The write value should be 0. The value read as the initial value is not guaranteed.</p>

Item	Page	Revisions (See Manual for Details)															
16.3.4 Local Acceptance Filter Mask (LAFM)/Tx-Trigger Time (TTT) LAFM Field:	539, 540	Table amended <table border="1"> <tr> <td>MBx[15], MBx[16]</td> <td>H'110 + N x 32</td> <td>15</td> <td>—</td> <td>Reserved The write value should be 0. The value read as the initial value is not guaranteed.</td> </tr> <tr> <td></td> <td></td> <td>14 to 4</td> <td>STDID_LAFM [10:0]</td> <td>Filter Mask Bits[10:0] for CAN Base ID[10:0] 0: Corresponding bit to CAN base ID set in mailbox is valid 1: Corresponding bit to CAN base ID set in mailbox is invalid</td> </tr> <tr> <td></td> <td></td> <td>3, 2</td> <td>—</td> <td>Reserved The write value should be 0. The value read as the initial value is not guaranteed.</td> </tr> </table>	MBx[15], MBx[16]	H'110 + N x 32	15	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.			14 to 4	STDID_LAFM [10:0]	Filter Mask Bits[10:0] for CAN Base ID[10:0] 0: Corresponding bit to CAN base ID set in mailbox is valid 1: Corresponding bit to CAN base ID set in mailbox is invalid			3, 2	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.
MBx[15], MBx[16]	H'110 + N x 32	15	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.													
		14 to 4	STDID_LAFM [10:0]	Filter Mask Bits[10:0] for CAN Base ID[10:0] 0: Corresponding bit to CAN base ID set in mailbox is valid 1: Corresponding bit to CAN base ID set in mailbox is invalid													
		3, 2	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.													
Tx-Trigger Time Field:		<table border="1"> <tr> <td>MBx[17], MBx[18]*</td> <td>H'112 + N x 32</td> <td>15 to 12</td> <td>—</td> <td>Reserved The write value should be 0. The value read as the initial value is not guaranteed.</td> </tr> <tr> <td></td> <td></td> <td>7 to 4</td> <td>—</td> <td>Reserved The write value should be 0. The value read as the initial value is not guaranteed.</td> </tr> </table>	MBx[17], MBx[18]*	H'112 + N x 32	15 to 12	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.			7 to 4	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.					
MBx[17], MBx[18]*	H'112 + N x 32	15 to 12	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.													
		7 to 4	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.													
16.4.2 Master Control Register_n (MCR_n) (n = 0, 1)	544, 545	Bit 7: Description amended Auto-wake Mode Bit 5: Important: added <b>Important:</b> Usage of sleep mode is limited. Be sure to carefully read section 16.8, Usage Notes.															
16.4.3 General Status Register_n (GSR_n) (n = 0, 1)	550	Bit 2: Description amended Message Transmission In Progress Flag															
16.4.4 HCAN-II_Bit timing Configuration Register n (HCAN-II_BCR0_n, HCAN-II_BCR1_n)(n = 0, 1) Table 16.5 TSEG1 and TSEG2 Settings	554	Important: added <b>Important:</b> When BRP[7:0] = H'00, TSEG2[2:0] ≠ B'001															
16.4.5 Interrupt Register_n (IRR_n) (n = 0, 1)	555-557, 559, 560	Bit table amended <table border="1"> <tr> <td>Bit:</td> <td>9</td> <td>8</td> </tr> <tr> <td></td> <td>IRR9</td> <td>IRR8</td> </tr> <tr> <td>Initial Value:</td> <td>0</td> <td>0</td> </tr> <tr> <td>R/W:</td> <td>R</td> <td>R</td> </tr> </table> Bit 12: Description amended Wake-up on Bus Activity Interrupt Flag Bit 9: R/W amended R Bit 4: Description amended Receive Overload Warning Interrupt Flag Bit 3: Description amended Transmit Overload Warning Interrupt Flag Bit 1: Description amended Data Frame Received Interrupt Flag	Bit:	9	8		IRR9	IRR8	Initial Value:	0	0	R/W:	R	R			
Bit:	9	8															
	IRR9	IRR8															
Initial Value:	0	0															
R/W:	R	R															

Item	Page	Revisions (See Manual for Details)
16.6 Timer Registers	581	Address amended
Table 16.7 HCAN Timer Registers		Channel 0 : H'D08C
16.7.2 HCAN Settings	598	Figure and notes amended

• Reset Sequence

Figure 16.7 Reset Sequence



Notes:

1. A software reset can be performed at any time by setting MCR [0] = 1

4. Deleted

16.7.3 Message Transmission Sequence	599	Figure amended
(1) Event Triggered Transmission		HCAN is in normal mode (MBC[x]=0x000 or 0x001 )
• Message Transmission Request		
Figure 16.8 Transmission Request		

16.7.4 Message Transmission Cancellation Sequence	602	Figure amended
Figure 16.10 Transmission Cancellation Sequence		Set ABACK[N] * <sup>2</sup>
		Set TXACK[N]

16.7.8 Interrupt Sources	607	Note added
Table 16.10 Interrupt Sources		

Interrupt Vector			Interrupt Flag (IRR Bit)	DMAC Activation	
HCAN0	HCAN1	Description		HCAN0	HCAN1
RM0	RM1	Data frame reception	IRR1	Possible*	
		Remote frame reception	IRR2		

Note: \* Mailbox 0 only



Item	Page	Revisions (See Manual for Details)
16.7.10 HCAN-II Port Settings	609	Note added  Note: * When the HCAN-II is used as a 64-buffer with one channel, care is required. Be sure to carefully read section 16.8, Usage Notes.
16.7.11 CAN Bus Interface Figure 16.16 High-Speed Interface Using HA13721	610	Newly added
16.8.4 TXPR Setting during Transmission	612	Description amended  When the HCAN-II is used with the baud rate set to 1 Mbps and the TXPR setting is made during transmission, there are the following limitations on the number of transmit mailboxes (MB) and the number of accesses to mailboxes until transmission is completed. Note that there is no limitation when 500 kbps of baud rate is used.
16.8.6 Mailbox Access in HCAN Sleep Mode Figure 16.17 HCAN Sleep Mode Flowchart	614, 615	Newly added
16.8.7 Notes on Port Settings for 64-Buffer HCAN-II with One Channel	616	Newly added
17.1.1 Features	617	<ul style="list-style-type: none"> <li>High-speed conversion</li> </ul> Conversion time: minimum 13.3 $\mu$ s per channel (when peripheral clock ( $P\phi$ ) = 20 MHz)
17.1.3 Pin Configuration	620	Description amended  The $\overline{ADTRG0}$ and $\overline{ADTRG1}$ pins are used to provide A/D conversion start timing from off-chip.  When the low level of a pulse is applied to one of these pins, A/D0, A/D1, or A/D2 starts conversion.
17.1.4 Register Configuration Table 17.2 A/D Converter Registers	624	Note replaced  Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

Item	Page	Revisions (See Manual for Details)																																										
17.2.3 A/D Control Registers 0 to 2 (ADCR0 to ADCR2) • Bit 7—Trigger Enable (TRGE):	631	Description amended  When ATU triggering is selected, clear bit 7 of registers $\overline{\text{ADTRGR0}}$ to $\overline{\text{ADTRGR2}}$ to 0. When external triggering is selected, upon input of the low level of a pulse to the $\overline{\text{ADTRG0}}$ or $\overline{\text{ADTRG1}}$ pin after TRGE has been set to 1, the A/D converter detects the falling edge of a pulse and sets the ADST bit to 1 in ADCR. The same operation is subsequently performed when 1 is written in the ADST bit by software. External triggering of A/D conversion is only enabled when the ADST bit is cleared to 0.  When external triggering is used, the low level input to the $\overline{\text{ADTRG0}}$ or $\overline{\text{ADTRG1}}$ pin must be at least 1.5 P $\phi$ clock cycles in width.																																										
17.4.3 Analog Input Sampling and A/D Conversion Time  Table 17.4 A/D Conversion Time (Single Mode)	644	Table amended  <table border="1"> <thead> <tr> <th rowspan="2">Item</th> <th rowspan="2">Symbol</th> <th colspan="3">CKS = 0: Peripheral Clock (P<math>\phi</math>) = 10 to 20 MHz</th> <th colspan="3">CKS = 1: Peripheral Clock (P<math>\phi</math>) = 10 MHz</th> <th rowspan="2">Unit</th> </tr> <tr> <th>Min</th> <th>Typ</th> <th>Max</th> <th>Min</th> <th>Typ</th> <th>Max</th> </tr> </thead> <tbody> <tr> <td>A/D conversion start delay time</td> <td><math>t_D</math></td> <td>10</td> <td>–</td> <td>17</td> <td>6</td> <td>–</td> <td>9</td> <td>States (peripheral clock (P<math>\phi</math>))</td> </tr> <tr> <td>Input sampling time</td> <td><math>t_{\text{SPL}}</math></td> <td>–</td> <td>64</td> <td>–</td> <td>–</td> <td>32</td> <td>–</td> <td></td> </tr> <tr> <td>A/D conversion time</td> <td><math>t_{\text{CONV}}</math></td> <td>259</td> <td>–</td> <td>266</td> <td>131</td> <td>–</td> <td>134</td> <td></td> </tr> </tbody> </table>	Item	Symbol	CKS = 0: Peripheral Clock (P $\phi$ ) = 10 to 20 MHz			CKS = 1: Peripheral Clock (P $\phi$ ) = 10 MHz			Unit	Min	Typ	Max	Min	Typ	Max	A/D conversion start delay time	$t_D$	10	–	17	6	–	9	States (peripheral clock (P $\phi$ ))	Input sampling time	$t_{\text{SPL}}$	–	64	–	–	32	–		A/D conversion time	$t_{\text{CONV}}$	259	–	266	131	–	134	
Item	Symbol	CKS = 0: Peripheral Clock (P $\phi$ ) = 10 to 20 MHz			CKS = 1: Peripheral Clock (P $\phi$ ) = 10 MHz			Unit																																				
		Min	Typ	Max	Min	Typ	Max																																					
A/D conversion start delay time	$t_D$	10	–	17	6	–	9	States (peripheral clock (P $\phi$ ))																																				
Input sampling time	$t_{\text{SPL}}$	–	64	–	–	32	–																																					
A/D conversion time	$t_{\text{CONV}}$	259	–	266	131	–	134																																					
17.4.4 External Triggering of A/D Conversion	646	Description amended  The A/D converter can be activated by input of an external A/D conversion start trigger.  To activate the A/D converter with an external trigger, first set the pin functions with the PFC (pin function controller), then set the TRGE bit to 1 in the A/D control register (ADCR), and set the EXTRG bit to 1 in the A/D trigger register (ADTRGR). When a low level is input to the $\overline{\text{ADTRG}}$ pin after these settings have been made, the A/D converter detects the falling edge of a pulse and sets the ADST bit to 1. Figure 17.7 shows the timing for external trigger input.  The ADST bit is set to 1 two states after the A/D converter samples the falling edge on the $\overline{\text{ADTRG}}$ pin. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written into the ADST bit by software.																																										
18.5.7 Operation Waveform Examples (C) Hardware Operation	690	Description amended  2. An interrupt is generated if the A/D cycle enable bit (ADCYLR) in the A/D trigger interrupt enable register (ADTIER) is set.																																										

18.5.7 Operation Waveform Examples 691

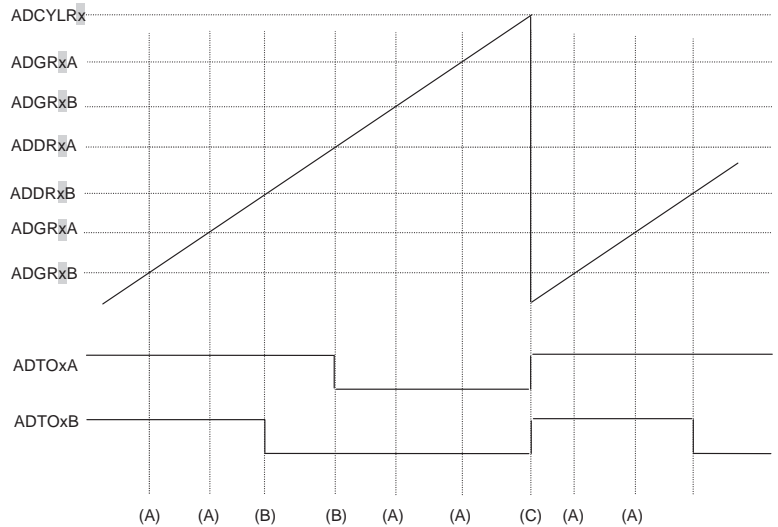
Title amended

(C)

Figure amended

Software Operation

Figure 18.5 Example of Output Waveform from MTAD PWM



DTSELxA, DTSELxB=0 (On-duty output is selected for PWM.)

Note: x = 0 or 1

20.5.1 Initialization

737

Description amended

4. When the AUDSRST bit is set to 1 in the SYSCR1 register (see section 25.2.2)

5. When the MSTOP3 bit is set to 1 in the SYSCR2 register (see section 25.2.3)

21.3.8 Port D IO Register (PDIOR)

759

Bit table amended

Bit 8 : PD8IOR

22.2.1 Register Configuration

804

Note amended

Table 22.1 Register Configuration

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

22.3.1 Register Configuration

806

Note amended

Table 22.3 Register Configuration

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

22.4.1 Register Configuration

808

Note amended

Table 22.5 Register Configuration

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

22.5.1 Register Configuration

810

Note amended

Table 22.7 Register Configuration

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

Item	Page	Revisions (See Manual for Details)
22.6.1 Register Configuration Table 22.9 Register Configuration	813	Note amended Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.
22.7.1 Register Configuration Table 22.11 Register Configuration	816	Note amended Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.
22.8.1 Register Configuration Table 22.13 Register Configuration	819	Note amended Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.
22.9.1 Register Configuration Table 22.15 Register Configuration	822	Note amended Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.
22.10.1 Register Configuration Table 22.17 Register Configuration	824	Note amended Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.
22.11.1 Register Configuration Table 22.19 Register Configuration	826	Note amended Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.
22.12.1 Register Configuration Table 22.21 Register Configuration	828	Note amended Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.
23.4.1 Registers Table 23.4 (1) Register Configuration	844	Note added 4. The registers except RAMER can be accessed only in bytes, and the access requires four cycles. Since RAMER is in the BSC, when it is accessed in bytes or words, the access requires four cycles, and when it is accessed in longwords, the access requires eight cycles.
23.4.3 Programming/Erasing Interface Parameters (2.2) Flash user branch address setting parameter (FUBRA: general register R5 of CPU)	855	Description amended Store general registers R8 to R15 and the control register GBR. General registers R0 to R7 are available without storing them.

Item	Page	Revisions (See Manual for Details)																								
23.5.2 User Program Mode (2.3) VBR is cleared to 0 and 1 is written to the SCO bit of FCCS, and then download is executed.	872	Description amended If an access by the DMAC or AUD occurs during download, operation cannot be guaranteed. Therefore, access by the DMAC or AUD must not be executed.																								
24.1 Overview	941	Description amended On-chip RAM data can always be accessed in one cycle for a read and two states for a write, making the RAM ideal for use as a program area, stack area, or data area, which require high-speed access.																								
25.1.3 Related Registers Table 25.3 Related Registers	945	Note amended 1. Register access with an internal clock multiplication ratio of 4 requires four internal clock ( $\phi$ ) cycles for SBYCR, and four or five internal clock ( $\phi$ ) cycles for SYSCR1 and SYSCR2.																								
27.2 DC Characteristics Table 27.2 Correspondence between Power Supply Names and Pins	959-968	Description amended Pin No. (FP-256H)																								
27.5 Flash Memory Characteristics Table 27.20 Flash Memory Characteristics	1001	Table amended <table border="1"> <thead> <tr> <th>Item</th> <th>Symbol</th> <th>Min</th> <th>Typ</th> <th>Max</th> <th>Unit</th> </tr> </thead> <tbody> <tr> <td>Programming time*1*2*4</td> <td><math>t_p</math></td> <td>-</td> <td>3</td> <td>200</td> <td>ms/128 bytes</td> </tr> <tr> <td>Erase time*1*3*5</td> <td><math>t_E</math></td> <td>-</td> <td>2</td> <td>20</td> <td>s/block</td> </tr> <tr> <td>Reprogramming count</td> <td><math>N_{WEC}</math></td> <td>100</td> <td>-</td> <td>-</td> <td>Times</td> </tr> </tbody> </table> <p>4. The total reprogramming time (programming time + erasing time) is as follows. 40 s (typ.), reference value: 60 s, 80 s (max.) However, 90% of the values are within the reference value.</p> <p>5. <math>t_p</math>, <math>t_E</math> distributes focusing on near the typ. value.</p>	Item	Symbol	Min	Typ	Max	Unit	Programming time*1*2*4	$t_p$	-	3	200	ms/128 bytes	Erase time*1*3*5	$t_E$	-	2	20	s/block	Reprogramming count	$N_{WEC}$	100	-	-	Times
Item	Symbol	Min	Typ	Max	Unit																					
Programming time*1*2*4	$t_p$	-	3	200	ms/128 bytes																					
Erase time*1*3*5	$t_E$	-	2	20	s/block																					
Reprogramming count	$N_{WEC}$	100	-	-	Times																					
A.1 Address Table A.1 Address	1065	Table amended <table border="1"> <thead> <tr> <th rowspan="2">Register Name</th> <th rowspan="2">Abbreviation</th> <th colspan="5">Bit Names</th> </tr> <tr> <th>Bit 7</th> <th>Bit 3</th> <th>Bit 2</th> <th>Bit 1</th> <th>Bit 0</th> </tr> </thead> <tbody> <tr> <td>H'FFFFFF70A</td> <td>SYCSR2*1</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> </tbody> </table>	Register Name	Abbreviation	Bit Names					Bit 7	Bit 3	Bit 2	Bit 1	Bit 0	H'FFFFFF70A	SYCSR2*1	-	-	-	-	-					
Register Name	Abbreviation	Bit Names																								
		Bit 7	Bit 3	Bit 2	Bit 1	Bit 0																				
H'FFFFFF70A	SYCSR2*1	-	-	-	-	-																				
Appendix B Pin States Table B.1 Pin States	1080	Table amended <table border="1"> <thead> <tr> <th>Type</th> <th>Pin Name</th> <th>AUD Module Standby</th> </tr> </thead> <tbody> <tr> <td>UBC</td> <td><math>\overline{UBCTRG}</math></td> <td><math>\bigcirc</math></td> </tr> </tbody> </table>	Type	Pin Name	AUD Module Standby	UBC	$\overline{UBCTRG}$	$\bigcirc$																		
Type	Pin Name	AUD Module Standby																								
UBC	$\overline{UBCTRG}$	$\bigcirc$																								
Appendix C Product Lineup Table C.1 SH7058 F-ZTAT Product Lineup	1083	Table amended <table border="1"> <thead> <tr> <th>Product Type</th> <th>Model Name</th> <th>Mark Model Name</th> <th>Package</th> <th>Operating Temperature (Except for W/E of Flash Memory)</th> </tr> </thead> <tbody> <tr> <td rowspan="4">SH7058</td> <td rowspan="2">F-ZTAT</td> <td>HD64F7058BF80L</td> <td>64F7058F80</td> <td>256-pin (FP-256H) -40°C to 105°C</td> </tr> <tr> <td>HD64F7058BF80K</td> <td>64F7058F80</td> <td>256-pin (FP-256H) -40°C to 125°C</td> </tr> <tr> <td rowspan="2"></td> <td>HD64F7058BP80L</td> <td>64F7058BP80</td> <td>272-pin (BP-272) -40°C to 105°C</td> </tr> <tr> <td>HD64F7058BP80K</td> <td>64F7058BP80</td> <td>272-pin (BP-272) -40°C to 125°C</td> </tr> </tbody> </table>	Product Type	Model Name	Mark Model Name	Package	Operating Temperature (Except for W/E of Flash Memory)	SH7058	F-ZTAT	HD64F7058BF80L	64F7058F80	256-pin (FP-256H) -40°C to 105°C	HD64F7058BF80K	64F7058F80	256-pin (FP-256H) -40°C to 125°C		HD64F7058BP80L	64F7058BP80	272-pin (BP-272) -40°C to 105°C	HD64F7058BP80K	64F7058BP80	272-pin (BP-272) -40°C to 125°C				
Product Type	Model Name	Mark Model Name	Package	Operating Temperature (Except for W/E of Flash Memory)																						
SH7058	F-ZTAT	HD64F7058BF80L	64F7058F80	256-pin (FP-256H) -40°C to 105°C																						
		HD64F7058BF80K	64F7058F80	256-pin (FP-256H) -40°C to 125°C																						
		HD64F7058BP80L	64F7058BP80	272-pin (BP-272) -40°C to 105°C																						
		HD64F7058BP80K	64F7058BP80	272-pin (BP-272) -40°C to 125°C																						

---

Item	Page	Revisions (See Manual for Details)
Appendix D Package Dimensions Figure D.2 SH7058 Package Dimensions (BP-272)	1086	Newly added

---



# Contents

Section 1	Overview	1
1.1	Features	1
1.2	Block Diagram	7
1.3	Pin Description	8
1.3.1	Pin Arrangement	8
1.3.2	Pin Functions	10
1.3.3	Pin Assignments	19
Section 2	CPU	29
2.1	Register Configuration	29
2.1.1	General Registers (Rn)	29
2.1.2	Control Registers	30
2.1.3	System Registers	31
2.1.4	Floating-Point Registers	32
2.1.5	Floating-Point System Registers	33
2.1.6	Initial Values of Registers	33
2.2	Data Formats	34
2.2.1	Data Format in Registers	34
2.2.2	Data Formats in Memory	34
2.2.3	Immediate Data Format	34
2.3	Instruction Features	35
2.3.1	RISC-Type Instruction Set	35
2.3.2	Addressing Modes	38
2.3.3	Instruction Format	41
2.4	Instruction Set by Classification	43
2.4.1	Instruction Set by Classification	43
2.5	Processing States	58
2.5.1	State Transitions	58
Section 3	Floating-Point Unit (FPU)	61
3.1	Overview	61
3.2	Floating-Point Registers and Floating-Point System Registers	62
3.2.1	Floating-Point Register File	62
3.2.2	Floating-Point Communication Register (FPUL)	62
3.2.3	Floating-Point Status/Control Register (FPSCR)	62
3.3	Floating-Point Format	65
3.3.1	Floating-Point Format	65
3.3.2	Non-Numbers (NaN)	66
3.3.3	Denormalized Number Values	66



3.3.4	Other Special Values.....	67
3.4	Floating-Point Exception Model.....	68
3.4.1	Enable State Exceptions.....	68
3.4.2	Disable State Exceptions.....	68
3.4.3	FPU Exception Event and Code.....	68
3.4.4	Floating-Point Data Arrangement in Memory .....	68
3.4.5	Arithmetic Operations Involving Special Operands .....	68
3.5	Synchronization with CPU.....	69
3.6	Usage Notes .....	70
 <b>Section 4 Operating Modes .....</b>		<b>71</b>
4.1	Operating Mode Selection .....	71
 <b>Section 5 Clock Pulse Generator (CPG).....</b>		<b>73</b>
5.1	Overview.....	73
5.1.1	Block Diagram.....	73
5.1.2	Pin Configuration.....	74
5.1.3	Related Register .....	74
5.2	Frequency Ranges and Clock Selection .....	74
5.2.1	Frequency Ranges.....	74
5.2.2	Clock Selection .....	75
5.2.3	Notes on Register Access.....	76
5.3	Clock Source.....	77
5.3.1	Connecting a Crystal Oscillator .....	77
5.3.2	External Clock Input Method.....	78
5.4	Oscillation Stop Detection Function .....	79
5.4.1	Overview.....	79
5.4.2	Settings of Oscillation Stop Detection Function .....	79
5.4.3	Related Register .....	81
5.4.4	Precautions for Performing Oscillation Stop Detection Function.....	81
5.5	Usage Notes .....	82
 <b>Section 6 Exception Processing.....</b>		<b>85</b>
6.1	Overview.....	85
6.1.1	Types of Exception Processing and Priority .....	85
6.1.2	Exception Processing Operations.....	86
6.1.3	Exception Processing Vector Table .....	87
6.2	Resets.....	90
6.2.1	Types of Reset .....	90
6.2.2	Power-On Reset .....	90
6.2.3	Manual Reset .....	91
6.3	Address Errors .....	92
6.3.1	Address Error Sources .....	92

6.3.2	Address Error Exception Processing.....	93
6.4	Interrupts.....	93
6.4.1	Interrupt Sources.....	93
6.4.2	Interrupt Priority Level.....	94
6.4.3	Interrupt Exception Processing.....	94
6.5	Exceptions Triggered by Instructions.....	95
6.5.1	Types of Exceptions Triggered by Instructions.....	95
6.5.2	Trap Instructions.....	95
6.5.3	Illegal Slot Instructions.....	96
6.5.4	General Illegal Instructions.....	96
6.5.5	Floating-Point Instructions.....	96
6.6	When Exception Sources Are Not Accepted.....	97
6.7	Stack Status after Exception Processing Ends.....	98
6.8	Usage Notes.....	99
6.8.1	Value of Stack Pointer (SP).....	99
6.8.2	Value of Vector Base Register (VBR).....	99
6.8.3	Address Errors Caused by Stacking of Address Error Exception Processing.....	99
<b>Section 7 Interrupt Controller (INTC).....</b>		<b>101</b>
7.1	Overview.....	101
7.1.1	Features.....	101
7.1.2	Block Diagram.....	102
7.1.3	Pin Configuration.....	103
7.1.4	Register Configuration.....	103
7.2	Interrupt Sources.....	104
7.2.1	NMI Interrupts.....	104
7.2.2	User Break Interrupt.....	104
7.2.3	H-UDI Interrupt.....	104
7.2.4	IRQ Interrupts.....	104
7.2.5	On-Chip Peripheral Module Interrupts.....	105
7.2.6	Interrupt Exception Vectors and Priority Rankings.....	106
7.3	Description of Registers.....	115
7.3.1	Interrupt Priority Registers A–L (IPRA–IPRL).....	115
7.3.2	Interrupt Control Register (ICR).....	117
7.3.3	IRQ Status Register (ISR).....	118
7.4	Interrupt Operation.....	119
7.4.1	Interrupt Sequence.....	119
7.4.2	Stack after Interrupt Exception Processing.....	121
7.5	Interrupt Response Time.....	122
7.6	Data Transfer with Interrupt Request Signals.....	124
7.6.1	Handling CPU Interrupt Sources, but Not DMAC Activating Sources.....	124
7.6.2	Handling DMAC Activating Sources but Not CPU Interrupt Sources.....	124

Section 8	User Break Controller (UBC)	125
8.1	Overview	125
8.1.1	Features	125
8.1.2	Block Diagram	126
8.1.3	Register Configuration	127
8.2	Register Descriptions	127
8.2.1	User Break Address Register (UBAR)	127
8.2.2	User Break Address Mask Register (UBAMR)	128
8.2.3	User Break Bus Cycle Register (UBBR)	130
8.2.4	User Break Control Register (UBCR)	132
8.3	Operation	133
8.3.1	Flow of the User Break Operation	133
8.3.2	Break on On-Chip Memory Instruction Fetch Cycle	135
8.3.3	Program Counter (PC) Values Saved	135
8.4	Examples of Use	135
8.4.1	Break on CPU Instruction Fetch Cycle	135
8.4.2	Break on CPU Data Access Cycle	136
8.4.3	Break on DMA Cycle	137
8.5	Usage Notes	138
8.5.1	Simultaneous Fetching of Two Instructions	138
8.5.2	Instruction Fetches at Branches	138
8.5.3	Contention between User Break and Exception Processing	139
8.5.4	Break at Non-Delay Branch Instruction Jump Destination	139
8.5.5	User Break Trigger Output	139
8.5.6	Module Standby	140
8.5.7	Internal Clock ( $\phi$ ) Multiplication Ratio and $\overline{\text{UBCTR}}\overline{\text{G}}$ Pulse Width	140
Section 9	Bus State Controller (BSC)	141
9.1	Overview	141
9.1.1	Features	141
9.1.2	Block Diagram	142
9.1.3	Pin Configuration	143
9.1.4	Register Configuration	143
9.1.5	Address Map	144
9.2	Description of Registers	146
9.2.1	Bus Control Register 1 (BCR1)	146
9.2.2	Bus Control Register 2 (BCR2)	148
9.2.3	Wait Control Register (WCR)	151
9.2.4	RAM Emulation Register (RAMER)	152
9.3	Accessing External Space	154
9.3.1	Basic Timing	154
9.3.2	Wait State Control	155
9.3.3	$\overline{\text{CS}}$ Assert Period Extension	157

9.4	Waits between Access Cycles .....	158
9.4.1	Prevention of Data Bus Conflicts.....	158
9.4.2	Simplification of Bus Cycle Start Detection .....	159
9.5	Bus Arbitration.....	160
9.6	Memory Connection Examples.....	161
<b>Section 10 Direct Memory Access Controller (DMAC) .....</b>		<b>163</b>
10.1	Overview.....	163
10.1.1	Features.....	163
10.1.2	Block Diagram.....	165
10.1.3	Register Configuration.....	166
10.2	Register Descriptions .....	167
10.2.1	DMA Source Address Registers 0–3 (SAR0–SAR3) .....	167
10.2.2	DMA Destination Address Registers 0–3 (DAR0–DAR3).....	168
10.2.3	DMA Transfer Count Registers 0–3 (DMATCR0–DMATCR3).....	169
10.2.4	DMA Channel Control Registers 0–3 (CHCR0–CHCR3).....	170
10.2.5	DMAC Operation Register (DMAOR).....	175
10.3	Operation .....	177
10.3.1	DMA Transfer Flow .....	177
10.3.2	DMA Transfer Requests .....	179
10.3.3	Channel Priority .....	182
10.3.4	DMA Transfer Types.....	182
10.3.5	Dual Address Mode .....	183
10.3.6	Bus Modes .....	189
10.3.7	Relationship between Request Modes and Bus Modes by DMA Transfer Category .....	190
10.3.8	Bus Mode and Channel Priorities .....	191
10.3.9	Source Address Reload Function.....	191
10.3.10	DMA Transfer Ending Conditions.....	192
10.3.11	DMAC Access from CPU.....	193
10.4	Examples of Use .....	194
10.4.1	Example of DMA Transfer between On-Chip SCI and External Memory .....	194
10.4.2	Example of DMA Transfer between A/D Converter and On-Chip Memory (Address Reload On).....	194
10.4.3	Example of DMA Transfer between External Memory and SCI1 Transmitting Side (Indirect Address on).....	196
10.5	Usage Notes .....	198
<b>Section 11 Advanced Timer Unit-II (ATU-II).....</b>		<b>199</b>
11.1	Overview.....	199
11.1.1	Features.....	199
11.1.2	Pin Configuration.....	204
11.1.3	Register Configuration.....	208

11.1.4	Block Diagrams .....	218
11.1.5	Inter-Channel and Inter-Module Signal Communication Diagram.....	228
11.1.6	Prescaler Diagram.....	229
11.2	Register Descriptions .....	230
11.2.1	Timer Start Registers (TSTR).....	230
11.2.2	Prescaler Registers (PSCR).....	234
11.2.3	Timer Control Registers (TCR) .....	235
11.2.4	Timer I/O Control Registers (TIOR).....	244
11.2.5	Timer Status Registers (TSR) .....	256
11.2.6	Timer Interrupt Enable Registers (TIER) .....	285
11.2.7	Interval Interrupt Request Registers (ITVRR).....	306
11.2.8	Trigger Mode Register (TRGMDR) .....	311
11.2.9	Timer Mode Register (TMDR).....	312
11.2.10	PWM Mode Register (PMDR) .....	313
11.2.11	Down-Count Start Register (DSTR) .....	315
11.2.12	Timer Connection Register (TCNR).....	322
11.2.13	One-Shot Pulse Terminate Register (OTR) .....	327
11.2.14	Reload Enable Register (RLDENR) .....	331
11.2.15	Free-Running Counters (TCNT).....	332
11.2.16	Down-Counters (DCNT) .....	334
11.2.17	Event Counters (ECNT).....	336
11.2.18	Output Compare Registers (OCR) .....	336
11.2.19	Input Capture Registers (ICR) .....	337
11.2.20	General Registers (GR).....	338
11.2.21	Offset Base Registers (OSBR).....	341
11.2.22	Cycle Registers (CYLR).....	341
11.2.23	Buffer Registers (BFR).....	342
11.2.24	Duty Registers (DTR).....	343
11.2.25	Reload Register (RLDR).....	344
11.2.26	Channel 10 Registers .....	344
11.3	Operation .....	360
11.3.1	Overview.....	360
11.3.2	Free-Running Counter Operation and Cyclic Counter Operation.....	367
11.3.3	Compare-Match Function .....	368
11.3.4	Input Capture Function .....	369
11.3.5	One-Shot Pulse Function .....	370
11.3.6	Offset One-Shot Pulse Function and Output Cutoff Function .....	371
11.3.7	Interval Timer Operation .....	372
11.3.8	Twin-Capture Function.....	373
11.3.9	PWM Timer Function.....	374
11.3.10	Channel 3 to 5 PWM Function .....	376
11.3.11	Event Count Function and Event Cycle Measurement .....	378
11.3.12	Channel 10 Functions .....	379

11.4	Interrupts .....	388
11.4.1	Status Flag Setting Timing.....	388
11.4.2	Status Flag Clearing .....	393
11.5	CPU Interface.....	395
11.5.1	Registers Requiring 32-Bit Access .....	395
11.5.2	Registers Permitting 8-Bit, 16-Bit, or 32-Bit Access.....	397
11.5.3	Registers Requiring 16-Bit Access .....	398
11.5.4	8-Bit or 16-Bit Accessible Registers.....	399
11.5.5	Registers Requiring 8-Bit Access .....	400
11.6	Sample Setup Procedures.....	400
11.7	Usage Notes .....	412
11.8	ATU-II Registers and Pins .....	425
<b>Section 12 Advanced Pulse Controller (APC).....</b>		<b>427</b>
12.1	Overview.....	427
12.1.1	Features.....	427
12.1.2	Block Diagram.....	428
12.1.3	Pin Configuration.....	429
12.1.4	Register Configuration.....	429
12.2	Register Descriptions .....	430
12.2.1	Pulse Output Port Control Register (POPCR).....	430
12.3	Operation .....	431
12.3.1	Overview.....	431
12.3.2	Advanced Pulse Controller Output Operation .....	432
12.4	Usage Notes .....	435
<b>Section 13 Watchdog Timer (WDT).....</b>		<b>437</b>
13.1	Overview.....	437
13.1.1	Features.....	437
13.1.2	Block Diagram.....	438
13.1.3	Pin Configuration.....	438
13.1.4	Register Configuration.....	439
13.2	Register Descriptions .....	439
13.2.1	Timer Counter (TCNT).....	439
13.2.2	Timer Control/Status Register (TCSR).....	440
13.2.3	Reset Control/Status Register (RSTCSR).....	442
13.2.4	Register Access.....	443
13.3	Operation .....	444
13.3.1	Watchdog Timer Mode.....	444
13.3.2	Interval Timer Mode.....	446
13.3.3	Timing of Setting the Overflow Flag (OVF) .....	446
13.3.4	Timing of Setting the Watchdog Timer Overflow Flag (WOVF).....	447
13.4	Usage Notes .....	447

13.4.1	TCNT Write and Increment Contention .....	447
13.4.2	Changing CKS2 to CKS0 Bit Values.....	448
13.4.3	Changing between Watchdog Timer/Interval Timer Modes.....	448
13.4.4	System Reset by $\overline{\text{WDTOVF}}$ Signal.....	448
13.4.5	Internal Reset in Watchdog Timer Mode.....	449
13.4.6	Manual Reset in Watchdog Timer .....	449
13.4.7	Multiplication Factor for Internal Clock Signal ( $\phi$ ) and Overflow Time.....	449
<b>Section 14 Compare Match Timer (CMT) .....</b>		<b>451</b>
14.1	Overview.....	451
14.1.1	Features.....	451
14.1.2	Block Diagram.....	452
14.1.3	Register Configuration.....	453
14.2	Register Descriptions .....	454
14.2.1	Compare Match Timer Start Register (CMSTR) .....	454
14.2.2	Compare Match Timer Control/Status Register (CMCSR) .....	455
14.2.3	Compare Match Timer Counter (CMCNT) .....	456
14.2.4	Compare Match Timer Constant Register (CMCOR).....	457
14.3	Operation .....	457
14.3.1	Cyclic Count Operation .....	457
14.3.2	CMCNT Count Timing.....	458
14.4	Interrupts.....	458
14.4.1	Interrupt Sources and DTC Activation .....	458
14.4.2	Compare Match Flag Set Timing.....	458
14.4.3	Compare Match Flag Clear Timing .....	459
14.5	Usage Notes .....	460
14.5.1	Contention between CMCNT Write and Compare Match.....	460
14.5.2	Contention between CMCNT Word Write and Incrementation .....	461
14.5.3	Contention between CMCNT Byte Write and Incrementation .....	462
<b>Section 15 Serial Communication Interface (SCI) .....</b>		<b>463</b>
15.1	Overview.....	463
15.1.1	Features.....	463
15.1.2	Block Diagram.....	464
15.1.3	Pin Configuration.....	465
15.1.4	Register Configuration.....	466
15.2	Register Descriptions .....	468
15.2.1	Receive Shift Register (RSR) .....	468
15.2.2	Receive Data Register (RDR).....	468
15.2.3	Transmit Shift Register (TSR) .....	468
15.2.4	Transmit Data Register (TDR).....	469
15.2.5	Serial Mode Register (SMR).....	469
15.2.6	Serial Control Register (SCR).....	472

15.2.7	Serial Status Register (SSR) .....	476
15.2.8	Bit Rate Register (BRR) .....	480
15.2.9	Serial Direction Control Register (SDCR).....	486
15.2.10	Inversion of SCK Pin Signal.....	487
15.3	Operation .....	488
15.3.1	Overview.....	488
15.3.2	Operation in Asynchronous Mode .....	490
15.3.3	Multiprocessor Communication.....	500
15.3.4	Synchronous Operation.....	507
15.4	SCI Interrupt Sources and the DMAC .....	515
15.5	Usage Notes .....	515
15.5.1	TDR Write and TDRE Flag .....	515
15.5.2	Simultaneous Multiple Receive Errors .....	516
15.5.3	Break Detection and Processing (Asynchronous Mode Only).....	516
15.5.4	Sending a Break Signal (Asynchronous Mode Only) .....	516
15.5.5	Receive Error Flags and Transmitter Operation (Synchronous Mode Only).....	517
15.5.6	Receive Data Sampling Timing and Receive Margin in Asynchronous Mode....	517
15.5.7	Constraints on DMAC Use .....	518
15.5.8	Cautions on Synchronous External Clock Mode .....	518
15.5.9	Caution on Synchronous Internal Clock Mode.....	518
 Section 16 Controller Area Network-II (HCAN-II) .....		 519
16.1	Overview.....	519
16.1.1	Features.....	519
16.2	Architecture.....	521
16.2.1	Block Diagram.....	521
16.2.2	Each Block Function.....	522
16.2.3	Pin Configuration.....	523
16.2.4	Memory Map .....	523
16.3	Mailboxes.....	526
16.3.1	Mailbox Configuration.....	526
16.3.2	Message Control Field .....	529
16.3.3	Message Data Fields .....	538
16.3.4	Local Acceptance Filter Mask (LAFM)/Tx-Trigger Time (TTT).....	538
16.4	HCAN Control Registers .....	541
16.4.1	Register Descriptions .....	542
16.4.2	Master Control Register_n (MCR_n) (n = 0, 1).....	542
16.4.3	General Status Register_n (GSR_n) (n = 0, 1).....	549
16.4.4	HCAN-II_Bit timing Configuration Register n (HCAN-II_BCR0_n, HCAN-II_BCR1_n) (n = 0, 1) .....	550
16.4.5	Interrupt Register_n (IRR_n) (n = 0, 1) .....	555
16.4.6	Interrupt Mask Register_n (IMR_n) (n = 0, 1) .....	561



16.4.7	Transmit Error Counter <sub>n</sub> (TEC <sub>n</sub> ) (n = 0, 1)/ Receive Error Counter <sub>n</sub> (REC <sub>n</sub> ) (n = 0, 1) .....	562
16.5	HCAN Mailbox Registers .....	563
16.5.1	Transmit Pending Request Register n (TXPR0n, TXPR1n) (n = 0, 1) .....	566
16.5.2	Transmit Cancel Register n (TXCR1n, TXCR0n) (n = 0, 1) .....	569
16.5.3	Transmit Acknowledge Register n (TXACK1n, TXACK0n) (n = 0, 1) .....	571
16.5.4	Abort Acknowledge Register n (ABACK1n, ABACK0n) (n = 0, 1) .....	573
16.5.5	Data Frame Receive Pending Register n (RXPR1n, RXPR0n) (n = 0, 1) .....	574
16.5.6	Remote Frame Receive Pending Register n (RFPR1n, RFPR0n) (n = 0, 1) .....	576
16.5.7	Mailbox Interrupt Mask Register n (MBIMR1n, MBIMR0n) (n = 0, 1) .....	577
16.5.8	Unread Message Status Register n (UMSR1n, UMSR0n) (n = 0, 1) .....	579
16.6	Timer Registers .....	580
16.6.1	Timer Counter Register n (TCNTRn) (n = 0, 1) .....	582
16.6.2	Timer Control Register <sub>n</sub> (TCR <sub>n</sub> ) (n = 0, 1) .....	583
16.6.3	Timer Status Register <sub>n</sub> (TSR <sub>n</sub> ) (n = 0, 1) .....	586
16.6.4	Timer Mode Register <sub>n</sub> (TMR <sub>n</sub> ) (n = 0, 1) .....	589
16.6.5	Timer Drift Correction Register n (TDCRn) (n = 0, 1) .....	590
16.6.6	Local Offset Register n (LOSRn) (n = 0, 1) .....	590
16.6.7	Cycle Counter Register n (CCRn) (n = 0, 1) .....	591
16.6.8	Cycle Counter Double-Buffer Register n (CCR_buf n) (n = 0, 1) .....	591
16.6.9	Cycle Maximum Register n (CMAXn) (n = 0, 1) .....	593
16.6.10	Input Capture Registers n (ICR0_cc n, ICR0_buf, ICR0_tm n, ICR1 n) (n = 0, 1) .....	593
16.6.11	Timer Compare Match Registers n (TCMR0n, TCMR1n, TCMR2n) (n = 0, 1) .....	595
16.7	Operation .....	597
16.7.1	Test Mode Settings .....	597
16.7.2	HCAN Settings .....	598
16.7.3	Message Transmission Sequence .....	599
16.7.4	Message Transmission Cancellation Sequence .....	601
16.7.5	Message Receive Sequence .....	603
16.7.6	Reconfiguration of Mailboxes .....	604
16.7.7	List of Registers .....	606
16.7.8	Interrupt Sources .....	607
16.7.9	DMAC Interface .....	608
16.7.10	HCAN-II Port Settings .....	609
16.7.11	CAN Bus Interface .....	610
16.8	Usage Notes .....	611
16.8.1	TXPR Setting during Reception .....	611
16.8.2	Transmit Cancellation Setting immediately after Transmission Setting in Bus Idle .....	611
16.8.3	Failure on Transmit Cancellation at Mailbox 31 .....	612
16.8.4	TXPR Setting during Transmission .....	612

16.8.5	Time Triggered Transmission Setting/Timer Operation Disabled.....	614
16.8.6	Mailbox Access in HCAN Sleep Mode .....	614
16.8.7	Notes on Port Settings for 64-Buffer HCAN-II with One Channel .....	616
<b>Section 17</b>	<b>A/D Converter.....</b>	<b>617</b>
17.1	Overview.....	617
17.1.1	Features.....	617
17.1.2	Block Diagram.....	618
17.1.3	Pin Configuration.....	620
17.1.4	Register Configuration.....	623
17.2	Register Descriptions .....	625
17.2.1	A/D Data Registers 0 to 31 (ADDR0 to ADDR31).....	625
17.2.2	A/D Control/Status Registers 0 and 1 (ADCSR0, ADCSR1) .....	626
17.2.3	A/D Control Registers 0 to 2 (ADCR0 to ADCR2).....	631
17.2.4	A/D Control/Status Register 2 (ADCSR2).....	633
17.2.5	A/D Trigger Registers 0 to 2 (ADTRGR0 to ADTRGR2) .....	636
17.3	CPU Interface.....	637
17.4	Operation .....	638
17.4.1	Single Mode.....	638
17.4.2	Scan Mode .....	640
17.4.3	Analog Input Sampling and A/D Conversion Time.....	644
17.4.4	External Triggering of A/D Conversion .....	646
17.4.5	A/D Converter Activation by ATU-II.....	647
17.4.6	ADEND Output Pin .....	647
17.5	Interrupt Sources and DMA Transfer Requests .....	648
17.6	Usage Notes .....	648
17.6.1	A/D conversion accuracy definitions.....	649
<b>Section 18</b>	<b>Multi-Trigger A/D Converter (MTAD).....</b>	<b>651</b>
18.1	Overview.....	651
18.1.1	Feature .....	651
18.1.2	Block Diagram.....	651
18.1.3	Input/Output Pins.....	653
18.1.4	Register Configuration.....	653
18.2	Register Descriptions .....	655
18.2.1	A/D Trigger Control Registers 0 and 1 (ADTCR0 and ADTCR1).....	655
18.2.2	A/D Trigger Status Registers 0 and 1 (ADTSR0 and ADTSR1).....	657
18.2.3	A/D Trigger Interrupt Enable Registers 0 and 1 (ADTIER0 and ADTIER1).....	660
18.2.4	A/D Free-Running Counters (ADCNT0 and ADCNT1) .....	663
18.2.5	A/D General Registers A and B (ADGR0A, ADGR0B, ADGR1A, and ADGR1B) .....	664
18.2.6	A/D Cycle Registers 0 and 1 (ADCYLR0 and ADCYLR1).....	664

18.2.7	A/D Duty Registers A and B (ADDR0A, ADDR0B, ADDR1A, and ADDR1B) .....	665
18.3	Interrupt Interface .....	666
18.3.1	On-Chip Peripheral Module Interrupts .....	666
18.3.2	Interrupt Exception Vectors and Priority Rankings .....	666
18.3.3	Interrupt Priority Registers A–L (IPRA–IPRL) .....	675
18.4	PFC and I/O Port Interfaces .....	677
18.4.1	PFC Interface .....	677
18.4.2	Port A Control Registers H and L (PACRH, PACRL) .....	677
18.4.3	I/O Port A.....	681
18.5	Operation .....	682
18.5.1	Overview.....	682
18.5.2	PWM Operation.....	683
18.5.3	Compare Match Operation.....	683
18.5.4	Multi-Trigger A/D Conversion Operation .....	683
18.5.5	Interrupts.....	688
18.5.6	Usage Notes .....	689
18.5.7	Operation Waveform Examples.....	689
18.6	Appendices.....	692
18.6.1	On-Chip Peripheral Module Registers .....	692
18.6.2	Pin States.....	693
18.6.3	AC Characteristics .....	693
<b>Section 19 High-performance User Debug Interface (H-UDI) .....</b>		<b>695</b>
19.1	Overview.....	695
19.1.1	Features.....	695
19.1.2	H-UDI Block Diagram.....	696
19.1.3	Pin Configuration.....	697
19.1.4	Register Configuration.....	697
19.2	External Signals .....	698
19.2.1	Test Clock (TCK) .....	698
19.2.2	Test Mode Select (TMS).....	698
19.2.3	Test Data Input (TDI) .....	698
19.2.4	Test Data Output (TDO) .....	698
19.2.5	Test Reset ( $\overline{\text{TRST}}$ ) .....	699
19.3	Register Descriptions .....	699
19.3.1	Instruction Register (SDIR) .....	699
19.3.2	Status Register (SDSR).....	701
19.3.3	Data Register (SDDR) .....	702
19.3.4	Bypass Register (SDBPR) .....	702
19.3.5	Boundary scan register (SDBSR).....	702
19.3.6	ID code register (SDIDR) .....	719
19.4	Operation .....	720

19.4.1	TAP Controller .....	720
19.4.2	H-UDI Interrupt and Serial Transfer.....	721
19.4.3	H-UDI Reset .....	724
19.5	Boundary Scan.....	724
19.5.1	Supported Instructions .....	724
19.5.2	Notes on Use.....	725
19.6	Usage Notes .....	726
<b>Section 20 Advanced User Debugger (AUD).....</b>		<b>729</b>
20.1	Overview.....	729
20.1.1	Features.....	729
20.1.2	Block Diagram.....	730
20.2	Pin Configuration.....	730
20.2.1	Pin Descriptions.....	731
20.3	Branch Trace Mode.....	733
20.3.1	Overview.....	733
20.3.2	Operation .....	733
20.4	RAM Monitor Mode .....	735
20.4.1	Overview.....	735
20.4.2	Communication Protocol .....	735
20.4.3	Operation .....	736
20.5	Usage Notes .....	737
20.5.1	Initialization.....	737
20.5.2	Operation in Software Standby Mode.....	737
<b>Section 21 Pin Function Controller (PFC).....</b>		<b>739</b>
21.1	Overview.....	739
21.2	Register Configuration.....	744
21.3	Register Descriptions .....	745
21.3.1	Port A IO Register (PAIOR).....	745
21.3.2	Port A Control Registers H and L (PACRH, PACRL) .....	746
21.3.3	Port B IO Register (PBIOR) .....	750
21.3.4	Port B Control Registers H and L (PBCRH, PBCRL).....	751
21.3.5	Port B Invert Register (PBIR).....	756
21.3.6	Port C IO Register (PCIOR) .....	757
21.3.7	Port C Control Register (PCCR).....	758
21.3.8	Port D IO Register (PDIOR).....	759
21.3.9	Port D Control Registers H and L (PDCRH, PDCRL) .....	760
21.3.10	Port E IO Register (PEIOR).....	764
21.3.11	Port E Control Register (PECR) .....	765
21.3.12	Port F IO Register (PFIOR) .....	770
21.3.13	Port F Control Registers H and L (PFCRH, PFCRL) .....	771
21.3.14	Port G IO Register (PGIOR).....	776

21.3.15	Port G Control Register (PGCR).....	777
21.3.16	Port H IO Register (PHIOR).....	778
21.3.17	Port H Control Register (PHCR).....	779
21.3.18	Port J IO Register (PJIOR).....	785
21.3.19	Port J Control Registers H and L (PJCRH, PJCRL) .....	786
21.3.20	Port K IO Register (PKIOR).....	790
21.3.21	Port K Control Registers H and L (PKCRH, PKCRL) .....	790
21.3.22	Port K Invert Register (PKIR) .....	795
21.3.23	Port L IO Register (PLIOR).....	796
21.3.24	Port L Control Registers H and L (PLCRH, PLCRL).....	797
21.3.25	Port L Invert Register (PLIR) .....	801
Section 22	I/O Ports (I/O).....	803
22.1	Overview.....	803
22.2	Port A.....	803
22.2.1	Register Configuration.....	804
22.2.2	Port A Data Register (PADR).....	804
22.2.3	Port A Port Register (PAPR) .....	805
22.3	Port B.....	806
22.3.1	Register Configuration.....	806
22.3.2	Port B Data Register (PBDR) .....	807
22.3.3	Port B Port Register (PBPR).....	808
22.4	Port C.....	808
22.4.1	Register Configuration.....	808
22.4.2	Port C Data Register (PCDR) .....	809
22.5	Port D.....	810
22.5.1	Register Configuration.....	810
22.5.2	Port D Data Register (PDDR).....	811
22.5.3	Port D Port Register (PDPR) .....	812
22.6	Port E.....	813
22.6.1	Register Configuration.....	813
22.6.2	Port E Data Register (PEDR).....	814
22.7	Port F.....	816
22.7.1	Register Configuration.....	816
22.7.2	Port F Data Register (PFDR) .....	817
22.8	Port G.....	818
22.8.1	Register Configuration.....	819
22.8.2	Port G Data Register (PGDR).....	819
22.9	Port H.....	821
22.9.1	Register Configuration.....	822
22.9.2	Port H Data Register (PHDR).....	822
22.10	Port J.....	823
22.10.1	Register Configuration.....	824

22.10.2	Port J Data Register (PJDR).....	824
22.10.3	Port J Port Register (PJPR).....	825
22.11	Port K.....	826
22.11.1	Register Configuration.....	826
22.11.2	Port K Data Register (PKDR).....	827
22.12	Port L.....	828
22.12.1	Register Configuration.....	828
22.12.2	Port L Data Register (PLDR).....	829
22.12.3	Port L Port Register (PLPR).....	830
22.13	POD (Port Output Disable) Control.....	831
<b>Section 23 ROM.....</b>		<b>833</b>
23.1	Features.....	833
23.2	Overview.....	835
23.2.1	Block Diagram.....	835
23.2.2	Operating Mode.....	836
23.2.3	Mode Comparison.....	838
23.2.4	Flash Memory Configuration.....	839
23.2.5	Block Division.....	840
23.2.6	Programming/Erasing Interface.....	841
23.3	Pin Configuration.....	843
23.4	Register Configuration.....	843
23.4.1	Registers.....	843
23.4.2	Programming/Erasing Interface Registers.....	846
23.4.3	Programming/Erasing Interface Parameters.....	851
23.4.4	RAM Emulation Register (RAMER).....	862
23.5	On-Board Programming Mode.....	864
23.5.1	Boot Mode.....	864
23.5.2	User Program Mode.....	868
23.5.3	User Boot Mode.....	878
23.6	Protection.....	881
23.6.1	Hardware Protection.....	881
23.6.2	Software Protection.....	882
23.6.3	Error Protection.....	883
23.7	Flash Memory Emulation in RAM.....	885
23.8	Usage Notes.....	888
23.8.1	Switching between User MAT and User Boot MAT.....	888
23.8.2	Interrupts during Programming/Erasing.....	889
23.8.3	Other Notes.....	893
23.9	Programmer Mode.....	894
23.9.1	Pin Arrangement of Socket Adapter.....	895
23.9.2	Programmer Mode Operation.....	897
23.9.3	Memory-Read Mode.....	898

23.9.4	Auto-Program Mode .....	899
23.9.5	Auto-Erase Mode.....	899
23.9.6	Status-Read Mode.....	900
23.9.7	Status Polling .....	900
23.9.8	Time Taken in Transition to Programmer Mode .....	901
23.9.9	Notes on Programming in Programmer Mode .....	901
23.10	Further Information.....	901
23.10.1	Serial Communication Interface Specification for Boot Mode.....	901
23.10.2	AC Characteristics and Timing in Programmer Mode.....	927
23.10.3	Storable Area for Procedure Program and Programming Data .....	933
<b>Section 24</b>	<b>RAM .....</b>	<b>941</b>
24.1	Overview.....	941
24.2	Operation .....	942
<b>Section 25</b>	<b>Power-Down State .....</b>	<b>943</b>
25.1	Overview.....	943
25.1.1	Power-Down States.....	943
25.1.2	Pin Configuration.....	945
25.1.3	Related Registers .....	945
25.2	Register Descriptions .....	945
25.2.1	Standby Control Register (SBYCR) .....	945
25.2.2	System Control Register 1 (SYSCR1) .....	946
25.2.3	System Control Register 2 (SYSCR2) .....	947
25.2.4	Notes on Register Access.....	949
25.3	Hardware Standby Mode .....	949
25.3.1	Transition to Hardware Standby Mode .....	949
25.3.2	Canceling Hardware Standby Mode .....	949
25.3.3	Hardware Standby Mode Timing.....	950
25.4	Software Standby Mode.....	950
25.4.1	Transition to Software Standby Mode .....	950
25.4.2	Canceling Software Standby Mode.....	950
25.4.3	Software Standby Mode Application Example.....	952
25.5	Sleep Mode .....	953
25.5.1	Transition to Sleep Mode.....	953
25.5.2	Canceling Sleep Mode .....	953
<b>Section 26</b>	<b>Reliability .....</b>	<b>955</b>
26.1	Reliability.....	955
<b>Section 27</b>	<b>Electrical Characteristics .....</b>	<b>957</b>
27.1	Absolute Maximum Ratings .....	957
27.2	DC Characteristics .....	959

27.3	AC Characteristics .....	975
27.3.1	Timing for swicthing the power supply on/off .....	975
27.3.2	Clock timing .....	976
27.3.3	Control Signal Timing .....	978
27.3.4	Bus Timing .....	982
27.3.5	Advanced Timer Unit Timing and Advance Pulse Controller Timing .....	986
27.3.6	I/O Port Timing.....	987
27.3.7	Watchdog Timer Timing.....	988
27.3.8	Serial Communication Interface Timing.....	989
27.3.9	HCAN Timing .....	991
27.3.10	A/D Converter Timing.....	992
27.3.11	H-UDI Timing .....	994
27.3.12	AUD Timing .....	996
27.3.13	UBC Trigger Timing.....	998
27.3.14	Measuring Conditions for AC Characteristics .....	999
27.4	A/D Converter Characteristics .....	1000
27.5	Flash Memory Characteristics.....	1001
27.6	Usage Note.....	1002
27.6.1	Notes on Connecting External Capacitor for Current Stabilization .....	1002
27.6.2	Notes on Mode Pin Input .....	1002
Appendix A On-chip peripheral module Registers.....		1005
A.1	Address .....	1005
A.2	Register States in Reset and Power-Down States .....	1073
Appendix B Pin States .....		1079
Appendix C Product Lineup .....		1083
Appendix D Package Dimensions .....		1085





# Section 1 Overview

## 1.1 Features

The SH7058 is a single-chip RISC microcontroller that integrates a RISC CPU core using an original Renesas architecture with peripheral functions required for system configuration.

The CPU has a RISC-type instruction set. Basic instructions can be executed in one state (one system clock cycle), which greatly improves instruction execution speed. In addition, the 32-bit internal architecture enhances data processing power. With this CPU, it has become possible to assemble low-cost, high-performance/high-functionality systems even for applications such as real-time control, which could not previously be handled by microcontrollers because of their high-speed processing requirements.

In addition, the SH7058 includes on-chip peripheral functions necessary for system configuration, such as a floating-point unit (FPU), ROM, RAM, a direct memory access controller (DMAC), timers, a serial communication interface (SCI), controller area network-II (HCAN-II), A/D converter, interrupt controller (INTC), and I/O ports.

ROM and SRAM can be directly connected by means of an external memory access support function, greatly reducing system cost.

On-chip ROM is available as flash memory in the F-ZTAT™\* (Flexible Zero Turn Around Time) version. The flash memory can be programmed with a programmer that supports SH7058 programming, and can also be programmed and erased by software. Since the programming/erasing control program is included as firmware, programming and erasing can be performed by calling this program with a user program. This enables the chip to be programmed by the user while mounted on a board.

The features of the SH7058 are summarized in table 1.1.

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

**Table 1.1 SH7058 Features**

<b>Item</b>	<b>Features</b>
CPU	<ul style="list-style-type: none"><li>• Maximum operating frequency: 80 MHz</li><li>• Original Renesas SH-2E CPU</li><li>• 32-bit internal architecture</li><li>• General register machine<ul style="list-style-type: none"><li>— Sixteen 32-bit general registers</li><li>— Three 32-bit control registers</li><li>— Four 32-bit system registers</li></ul></li><li>• Instruction execution time: Basic instructions execute in one state (12.5 ns/instruction at 80 MHz operation)</li><li>• Address space: Architecture supports 4 Gbytes</li><li>• Five-stage pipeline</li></ul>
Operating states	<ul style="list-style-type: none"><li>• Operating modes<ul style="list-style-type: none"><li>— Single-chip mode</li><li>— 8/16-bit bus expanded mode<ul style="list-style-type: none"><li>• Mode with on-chip ROM</li><li>• Mode with no on-chip ROM</li></ul></li></ul></li><li>• Processing states<ul style="list-style-type: none"><li>— Reset state</li><li>— Program execution state</li><li>— Exception handling state</li><li>— Bus-released state</li><li>— Power-down state</li></ul></li><li>• Power-down state<ul style="list-style-type: none"><li>— Sleep mode</li><li>— Software standby mode</li><li>— Hardware standby mode</li><li>— Module standby</li></ul></li></ul>
Multiplier	<ul style="list-style-type: none"><li>• <math>32 \times 32 \rightarrow 64</math> multiply operations executed in two to four cycles</li><li>• <math>32 \times 32 + 64 \rightarrow 64</math> multiply-and-accumulate operations executed in two to four cycles</li></ul>

**Table 1.1 SH7058 Features (cont)**

<b>Item</b>	<b>Features</b>
Floating-point unit (FPU)	<ul style="list-style-type: none"><li>• SuperH architecture coprocessor</li><li>• Supports single-precision floating-point operations</li><li>• Supports a subset of the data types specified by the IEEE standard</li><li>• Supports invalid operation and division-by-zero exception detection (subset of IEEE standard)</li><li>• Supports Round to Zero as the rounding mode (subset of IEEE standard)</li><li>• Sixteen 32-bit floating-point data registers</li><li>• Supports the FMAC instruction (multiply-and-accumulate instruction)</li><li>• Supports the FDIV instruction (divide instruction)</li><li>• Supports the FLDI0/FLDI1 instructions (constant 0/1 load instructions)</li><li>• Instruction delay time: Two cycles for each of FMAC, FADD, FSUB, and FMUL instructions</li><li>• Execution pitch: One cycle for each of FMAC, FADD, FSUB, and FMUL instructions</li></ul>
Clock pulse generator (CPG/PLL)	<ul style="list-style-type: none"><li>• On-chip clock pulse generator (maximum operating frequency: 80 MHz)</li><li>• Independent generation of CPU system clock and peripheral clock for peripheral modules</li><li>• On-chip clock-multiplication PLL circuit (<math>\times 4</math>, <math>\times 8</math>)</li><li>• Internal clock frequency range: 5 to 10 MHz</li></ul>
Interrupt controller (INTC)	<ul style="list-style-type: none"><li>• Nine external interrupt pins (NMI, <math>\overline{\text{IRQ0}}</math> to <math>\overline{\text{IRQ7}}</math>)</li><li>• 117 internal interrupt sources (ATU-II <math>\times 75</math>, SCI <math>\times 20</math>, DMAC <math>\times 4</math>, A/D <math>\times 5</math>, WDT <math>\times 1</math>, UBC <math>\times 1</math>, CMT <math>\times 2</math>, HCAN-II <math>\times 8</math>, H-UDI <math>\times 1</math>)</li><li>• 16 programmable priority levels</li></ul>
User break controller (UBC)	<ul style="list-style-type: none"><li>• Requests an interrupt when the CPU or DMAC generates a bus cycle with specified conditions (interrupt can also be masked)</li><li>• Trigger pulse output (UBCTRG) on break condition — Selection of trigger pulse width (<math>\phi \times 1</math>, <math>\times 4</math>, <math>\times 8</math>, <math>\times 16</math>)</li><li>• Simplifies configuration of an on-chip debugger</li></ul>

**Table 1.1 SH7058 Features (cont)**

<b>Item</b>	<b>Features</b>
Bus state controller (BSC)	<ul style="list-style-type: none"><li>• Supports external memory access (SRAM and ROM directly connectable)<ul style="list-style-type: none"><li>— 8/16-bit bus space</li></ul></li><li>• 3.3 V bus interface</li><li>• 16 MB address space divided into four areas, with the following parameters settable for each area:<ul style="list-style-type: none"><li>— Bus size (8 or 16 bits)</li><li>— Number of wait cycles</li><li>— Chip select signals (<math>\overline{CS0}</math> to <math>\overline{CS3}</math>) output for each area</li></ul></li><li>• Wait cycles can be inserted using an external <math>\overline{WAIT}</math> signal</li><li>• External access in minimum of two cycles</li><li>• Provision for idle cycle insertion to prevent bus collisions</li></ul>
Direct memory access controller (DMAC) (4 channels)	<ul style="list-style-type: none"><li>• DMA transfer possible for the following devices:<ul style="list-style-type: none"><li>— External memory, on-chip memory, on-chip peripheral modules (excluding DMAC, UBC, BSC)</li></ul></li><li>• DMA transfer requests by on-chip modules<ul style="list-style-type: none"><li>— SCI, A/D converter, ATU-II, HCAN-II</li></ul></li><li>• Cycle steal or burst mode transfer</li><li>• Dual address mode<ul style="list-style-type: none"><li>— Direct transfer mode</li><li>— Indirect transfer mode (channel 3 only)</li></ul></li><li>• Address reload function (channel 2 only)</li><li>• Transfer data width: Byte/word/longword</li></ul>
Advanced timer unit-II (ATU-II)	<ul style="list-style-type: none"><li>• Maximum 65 inputs or outputs can be processed<ul style="list-style-type: none"><li>— Four 32-bit input capture inputs</li><li>— Thirty 16-bit input capture inputs/output compare outputs</li><li>— Sixteen 16-bit one-shot pulse outputs</li><li>— Eight 16-bit PWM outputs</li><li>— Six 8-bit event counters</li><li>— One gap detection function</li></ul></li><li>• I/O pin output inversion function</li></ul>
Advanced pulse controller (APC)	<ul style="list-style-type: none"><li>• Maximum eight pulse outputs on reception of ATU-II (channel 11) compare-match signal</li></ul>

**Table 1.1 SH7058 Features (cont)**

<b>Item</b>	<b>Features</b>
Watchdog timer (WDT) (1 channel)	<ul style="list-style-type: none"> <li>• Can be switched between watchdog timer and interval timer function</li> <li>• Internal reset, external signal, or interrupt generated by counter overflow</li> <li>• Two kinds of internal reset <ul style="list-style-type: none"> <li>— Power-on reset</li> <li>— Manual reset</li> </ul> </li> </ul>
Compare-match timer (CMT) (2 channels)	<ul style="list-style-type: none"> <li>• Selection of 4 counter input clocks</li> <li>• A compare-match interrupt can be requested independently for each channel</li> </ul>
Serial communication interface (SCI) (5 channels)	<ul style="list-style-type: none"> <li>• Selection of asynchronous or synchronous mode</li> <li>• Simultaneous transmission/reception (full-duplex) capability</li> <li>• Serial data communication possible between multiple processors (asynchronous mode)</li> <li>• Clock inversion function</li> <li>• LSB-/MSB-first selection function for transmission</li> </ul>
Controller area network-II (HCAN-II) (2 channels)	<ul style="list-style-type: none"> <li>• CAN version: Bosch 2.0B active compatible</li> <li>• Buffer size (per channel): Transmit/receive × 31, receive-only × 1</li> <li>• Receive message filtering capability</li> </ul>
A/D converter	<ul style="list-style-type: none"> <li>• Thirty-two channels</li> <li>• Three sample-and-hold circuits <ul style="list-style-type: none"> <li>— Independent operation of 12 channels × 2 and 8 channels × 1</li> </ul> </li> <li>• Selection of two conversion modes <ul style="list-style-type: none"> <li>— Single conversion mode</li> <li>— Scan mode <ul style="list-style-type: none"> <li>• Continuous scan mode</li> <li>• Single-cycle scan mode</li> </ul> </li> </ul> </li> <li>• Can be activated by external trigger or ATU-II compare-match</li> <li>• 10-bit resolution</li> <li>• Accuracy: ±2 LSB</li> </ul>
Multi-trigger A/D (MTAD)	<ul style="list-style-type: none"> <li>• While performing conversion on the specified channels in scan mode, A/D conversion on the channels for which conversion has been requested can be performed prior to the other channels when a compare match occurs with respect to the timer in the A/D converter</li> </ul>

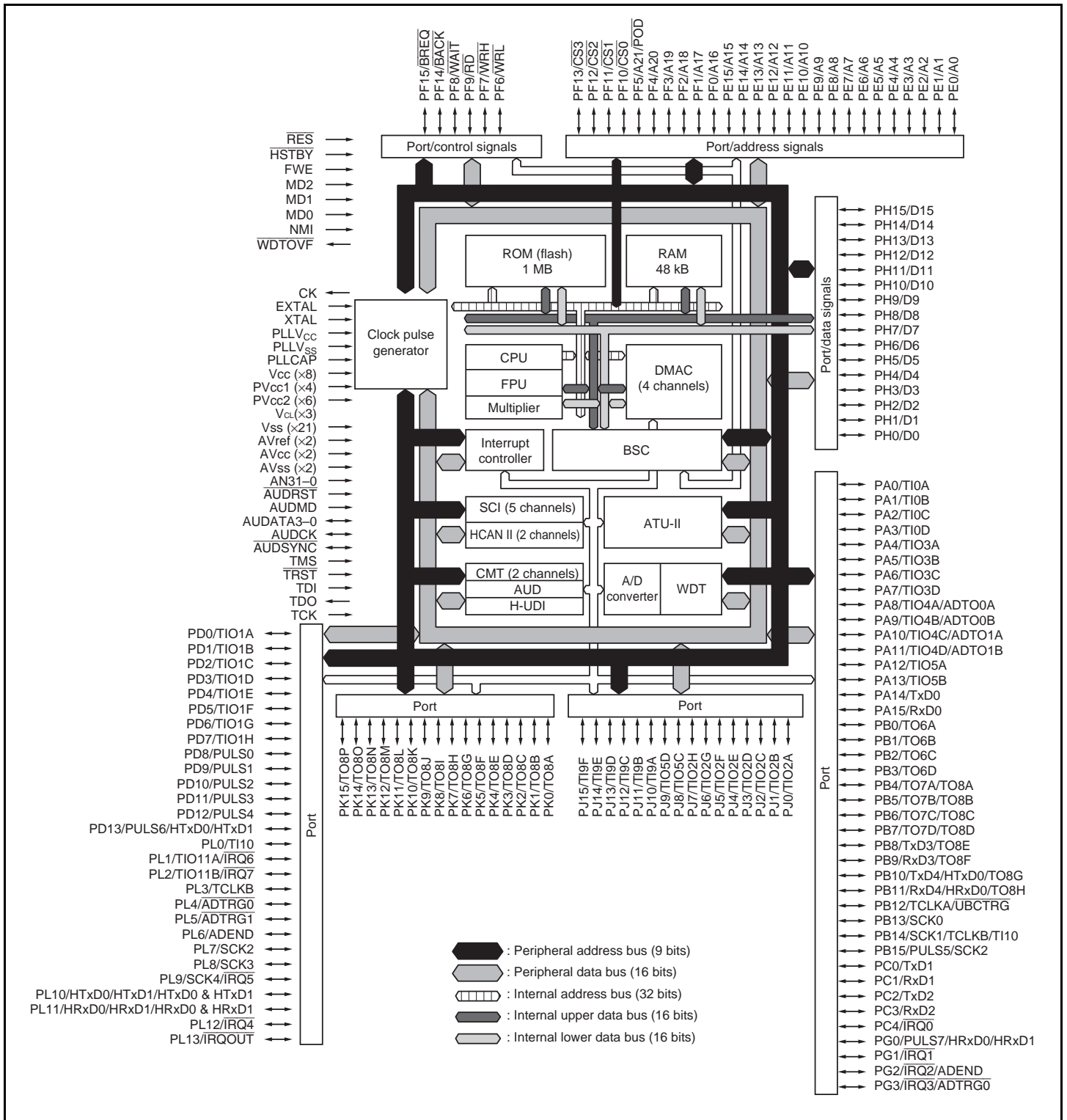
**Table 1.1 SH7058 Features (cont)**

Item	Features
High-performance user debug interface (H-UDI)	<ul style="list-style-type: none"> <li>• Compliant with IEEE1149.1               <ul style="list-style-type: none"> <li>— Five test signals (TCK, TDI, TD0, TMS, and <math>\overline{\text{TRST}}</math>)</li> <li>— TAP controller</li> <li>— Instruction register</li> <li>— Data register</li> <li>— Bypass register</li> </ul> </li> <li>• Test mode compliant with IEEE1149.1               <ul style="list-style-type: none"> <li>— Standard instructions: BYPASS, SAMPLE/PRELOAD, EXTEST</li> <li>— Optional instructions: CLAMP, HIGHZ, IDCODE</li> </ul> </li> <li>• H-UDI interrupt               <ul style="list-style-type: none"> <li>— H-UDI interrupt request to INTC</li> </ul> </li> </ul>
Advanced user debugger (AUD)	<ul style="list-style-type: none"> <li>• Eight dedicated pins</li> <li>• RAM monitor mode               <ul style="list-style-type: none"> <li>— Data input/output frequency: 10 MHz or less</li> <li>— Possible to read/write to a module connected to the internal/external bus</li> </ul> </li> <li>• Branch address output mode</li> </ul>
I/O ports (including timer I/O pins, address and data buses)	<ul style="list-style-type: none"> <li>• Dual-function input/output pins: 149</li> <li>• Schmitt input pins: NMI, <math>\overline{\text{IRQn}}</math>, <math>\overline{\text{RES}}</math>, <math>\overline{\text{HSTBY}}</math>, FWE, TCLK, IC, IC/OC, SCK, <math>\overline{\text{ADTRG}}</math></li> <li>• Input port protection</li> </ul>
ROM	<ul style="list-style-type: none"> <li>• 1-MB flash memory</li> <li>• 1-MB divided into 16 blocks               <ul style="list-style-type: none"> <li>— Small blocks: 4 kB × 8</li> <li>— Medium block: 96 kB × 1</li> <li>— Large blocks: 128 kB × 7</li> </ul> </li> <li>• RAM emulation function (using 4 kB small block)</li> <li>• Programming/erasing control program included as firmware</li> <li>• Flash memory programming methods               <ul style="list-style-type: none"> <li>— Boot mode</li> <li>— User program mode</li> <li>— User boot mode</li> <li>— Programmer mode</li> </ul> </li> </ul>

**Table 1.1 SH7058 Features (cont)**

Item	Features
RAM	• 48 kB SRAM

## 1.2 Block Diagram

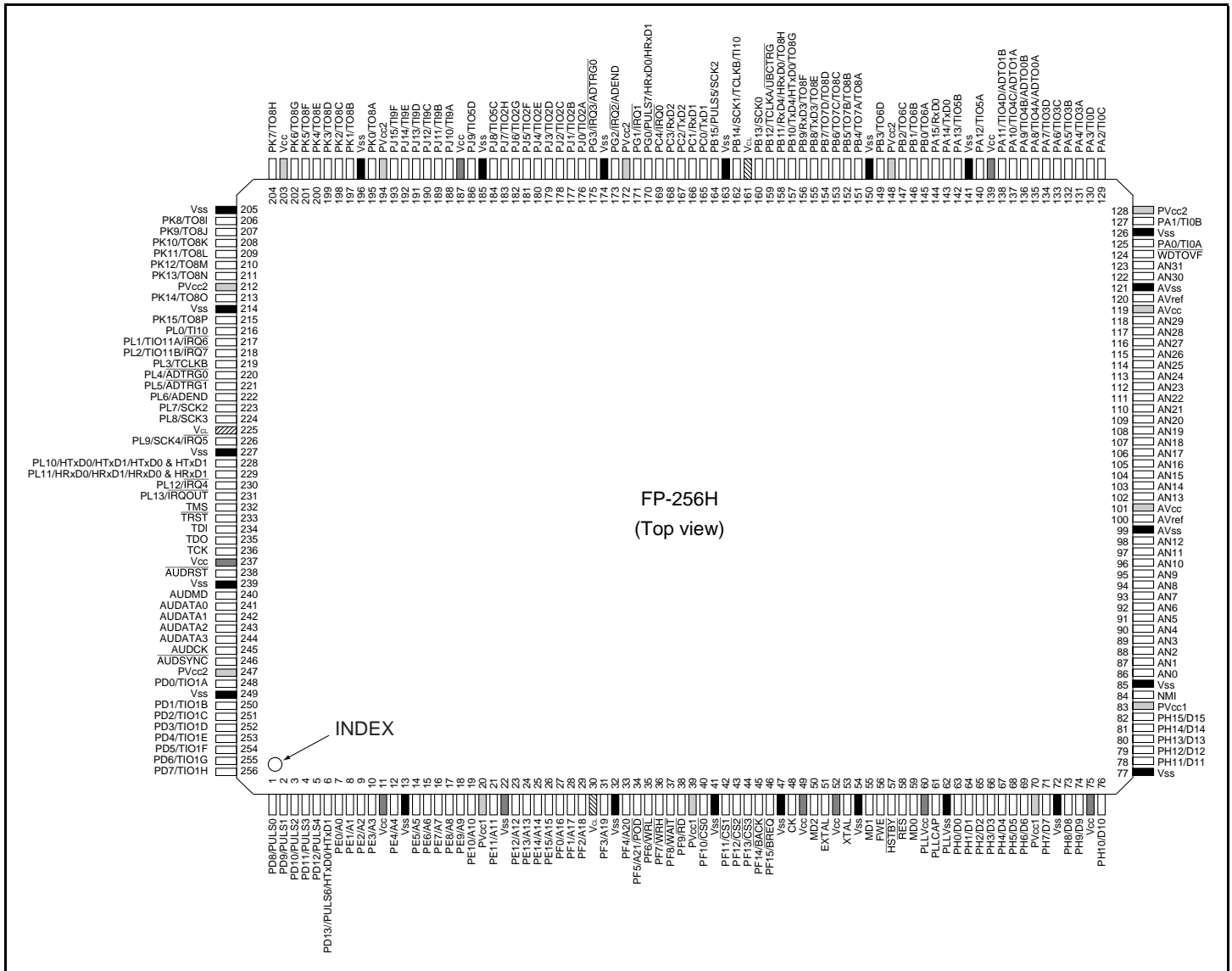


**Figure 1.1 Block Diagram**



# 1.3 Pin Description

## 1.3.1 Pin Arrangement



**Figure 1.2 Pin Assignments (FP-256H)**

20		PH5/D5	PH13/D13	PH15/D15	NMI	AN3	AN5	AN8	AVss	AVcc	AN15	AN16	AN19	AN21	AN24	AVcc	AVss	PA2/TIOc	PA4/TIO3A	
19	PH2/D2	PH4/D4	PH7/D7	PH12/D12	PH14/D14	Vss	AN2	AN7	AN9	AVref	AN13	AN18	AN22	AN23	AN28	AVref	PVcc2	PA3/TIOd	PA6/TIO3C	PA7/TIO3D
18	PLLvss	PH1/D1	PH3/D3	PH8/D8	PH11/D11	PVcc1	AN1	AN4	AN10	AN11	AN14	AN20	AN26	AN27	AN31	Vss	PA0/TIOA	PA5/TIO3B	PA8/TIO4A/ ADTO0A	PA10/TIO4C/ ADTO1A
17	PLLvcc	PLLCAP	PH6/D6	PH9/D9	PH10/D10	Vcc	Vss	AN0	AN6	AN12	AN17	AN25	AN29	AN30	WDTOVF	PA1/TIOB	PA9/TIO4B/ ADTO0B	PA11/TIO4D/ ADTO1B	PA12/ TIO5A	PA14/ TxDO
16	HSTBY	RES	MD0	PVcc1													Vcc	Vss	PA13/ TIO5B	PA15/ RxDO
15	XTAL	Vss	MD1	PH0/D0													PB0/TO6A	PVcc2	PB1/TO6B	PB2/TO6C
14	EXTAL	Vcc	MD2	FWE													PB4/TO7A/ TO8A	PB5/TO7B/ TO8B	PB6/TO7C/ TO8C	PB3/TO6D
13	CK	Vss	PF15/ BREQ	Vcc													PB9/RxD3/ TO8F	PB10/TxD4/ HTxD0/TO8G	PB8/TxD3/ TO8E	PB7/TO7D/ TO8D
12	PF11/CS1	PF13/CS3	PF12/CS2	PF14/ BACK	Vss				Vss				Vss				Vss			
11	PF10/CS0	PF9/RD	PVcc1	Vss	Vss				Vss				Vss				Vss			
10	PF8/WAIT	PF6/WRL	PF5/A21/ POD	PF7/WRH	Vss				Vss				Vss				Vss			
9	Vss	VCL	PF4/A20	PF3/A19	Vss				Vss				Vss				Vss			
8	PF2/A18	PF1/A17	PF0/A16	PE15/A15													PJ3/TIO2D	PJ2/TIO2C	PG3/IRQ3/ ADTRG0	PG1/IRQ1
7	PE14/A14	PE13/A13	PE12/A12	Vss													PJ8/TIO5C	PJ7/TIO2H	PJ4/TIO2E	PJ0/TIO2A
6	PE11/A11	PE10/A10	PVcc1	PE9/A9													Vcc	Vss	PJ6/TIO2G	PJ1/TIO2B
5	PE8/A8	PE7/A7	PE6/A6	Vcc													Vss	PVcc2	PJ11/TI9B	PJ5/TIO2F
4	PE5/A5	PE4/A4	PE2/A2	PE1/A1	PD8/ PULS0	PD6/ TIO1G	PD1/ TIO1B	PVcc2	Vss	TDI	PL8/SCK3	PL4/ ADTRG0	PL0/TI10	Vss	PK8/TO8I	PK6/ TO8G	Vcc	PJ15/TI9F	PJ13/TI9D	PJ9/TIO5D
3	PE3/A3	PE0/A0	PD11/ PULS3	PD9/ PULS1	PD7/ TIO1H	PD5/ TIO1F	Vss	AUDMD	Vcc	TMS	PL9/SCK4/ IRG5	PL7/SCK2	PL3/ TCLKB	PVcc2	PK12/ TO8M	PK10/ TO8K	PK9/TO8J	PK1/TO8B	PJ14/TI9E	PJ10/TI9A
2	PD13/PULS6/ HTxD0/HTxD1	PD12/ PULS4	PD10/ PULS2	PD4/ TIO1E	PD2/ TIO1C	AUDSYNC	AUDATA2	AUDRST	TCK	PL12/ IRQ4	PL11/HRxD0/ HRxD1/ HRxD0&HRxD1	VCL	PL6/ ADEND	PL1/TIO11A/ IRQ6	PK14/ TO8O	PK11/ TO8L	PK7/TO8H	PK3/TO8D	PK2/TO8C	PJ12/TI9C
1		PD3/ TIO1D	PD0/ TIO1A	AUDCK	AUDATA3	AUDATA1	AUDATA0	TDO	TRST	PL13/ IRQOUT	PL10/HTxD0/ HTxD1/ HTxD0&HTxD1	Vss	PL5/ ADTRG1	PL2/TIO11B/ IRQ7	PK15/ TO8P	PK13/ TO8N	PK5/TO8F	PK4/TO8E	PK0/TO8A	
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y

Figure 1.3 Pin Assignments

## 1.3.2 Pin Functions

Table 1.2 summarizes the pin functions.

**Table 1.2 Pin Functions**

Type	Symbol	Pin No.		I/O	Name	Function
		FP-256H	BP-272			
Power supply	$V_{CC}$	11, 49, 52, 75, 139, 187, 203, 237	D5, D13, B14, F17, U16, U6, U4, J3	Input	Power supply	Power supply for chip-internal and system ports ( $\overline{RES}$ , MD2–MD0, FWE, $\overline{HSTBY}$ , NMI, CK, EXTAL, XTAL, H-UDI port). Connect all $V_{CC}$ pins to the system power supply. The chip will not operate if there are any open pins.
	$PV_{CC1}$	20, 39, 70, 83	C6, V11, D16, F18	Input	Port power supply 1	Power supply for bus ports (ports E, F, and H). Connect all $PV_{CC1}$ pins to the system bus power supply. The chip will not operate if there are any open pins.
	$PV_{CC2}$	128, 148, 172, 194, 212, 247	U19, V15, V9, V5, P3, H4	Input	Port power supply 2	Power supply for peripheral module ports (ports A, B, C, D, G, J, K, and L, the AUD port, and $\overline{WDTOVF}$ ). Connect all $PV_{CC2}$ pins to the system peripheral module power supply. The chip will not operate if there are any open pins.
	$V_{CL}$	30, 161, 225	B9, Y11, M2	Input	Internal step-down power supply	Pins for connection to a capacitor used for stabilizing the voltage of the internal step-down power supply.  Connect $V_{SS}$ to this pin through a (0.33,0.47)- $\mu$ F capacitor. The capacitor should be located near the pin. Do not connect an external power supply to the pin.

**Table 1.2 Pin Functions (cont)**

Type	Symbol	Pin No.			Name	Function
		FP-256H	BP-272	I/O		
Power supply	$V_{SS}$	13, 22, 32, 41, 47, 54, 72, 77, 85, 126, 141, 150, 163, 174, 185, 196, 205, 214, 227, 239, 249	A9, B13, B15, D7, D11, F19, G3, G17, J4, J9-12, K9-12, L9-12, M1, M9-12, P4, T18, U5, U9, V6, V16, W11	Input	Ground	For connection to ground.  Connect all $V_{SS}$ pins to the system ground. The chip will not operate if there are any open pins.
Flash memory	FWE	56	D14	Input	Flash write enable	Connected to ground in normal operation.  Apply $V_{CC}$ during on-board programming.
Clock	$PLL V_{CC}$	60	A17	Input	PLL power supply	On-chip PLL oscillator power supply.  For power supply connection, see section 5, Clock Pulse Generator (CPG).
	$PLL V_{SS}$	62	A18	Input	PLL ground	On-chip PLL oscillator ground.  For power supply connection, see section 5, Clock Pulse Generator (CPG).
	PLLCAP	61	B17	Input	PLL capacitance	On-chip PLL oscillator external capacitance connection pin.  For external capacitance connection, see section 5, Clock Pulse Generator (CPG).
	EXTAL	51	A14	Input	External clock	For connection to a crystal resonator. An external clock source can also be connected to the EXTAL pin.
	XTAL	53	A15	Input	Crystal	For connection to a crystal resonator.
	CK	48	A13	Output	Peripheral clock	Supplies the peripheral clock to peripheral devices.

**Table 1.2 Pin Functions (cont)**

Type	Symbol	Pin No.		I/O	Name	Function
		FP-256H	BP-272			
System control	$\overline{\text{RES}}$	58	B16	Input	Power-on reset	Executes a power-on reset when driven low.
	$\overline{\text{WDTOVF}}$	124	R17	Output	Watchdog timer overflow	WDT overflow output signal.
	$\overline{\text{BREQ}}$	46	C13	Input	Bus request	Driven low when an external device requests the bus.
	$\overline{\text{BACK}}$	45	D12	Output	Bus request acknowledge	Indicates that the bus has been granted to an external device. The device that output the $\overline{\text{BREQ}}$ signal recognizes that the bus has been acquired when it receives the $\overline{\text{BACK}}$ signal.
Operating mode control	MD0 to MD2	59, 55, 50	C16, C15, C14	Input	Mode setting	These pins determine the operating mode. Do not change the input values during operation.
	$\overline{\text{HSTBY}}$	57	A16	Input	Hardware standby	When driven low, this pin forces a transition to hardware standby mode.
Interrupts	NMI	84	E20	Input	Nonmaskable interrupt	Nonmaskable interrupt request pin.  Acceptance on the rising edge or falling edge can be selected.
	$\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$	169, 171, 173, 175, 230, 226, 217, 218	V10, Y8, W9, W8, K2, L3, P2, P1	Input	Interrupt requests 0 to 7	Maskable interrupt request pins.  Level input or edge input can be selected.
	$\overline{\text{IRQOUT}}$	231	K1	Output	Interrupt request output	Indicates that an interrupt has been generated.  Enables interrupt generation to be recognized in the bus-released state.

**Table 1.2 Pin Functions (cont)**

Type	Symbol	Pin No.			Name	Function
		FP-256H	BP-272	I/O		
Address bus	A0–A21	7–10, 12, 14–19, 21, 23–29, 31, 33, 34	B3, D4, C4, A3, B4, A4, C5, B5, A5, D6, B6, A6, C7, B7, A7, D8, C8, B8, A8, D9, C9, C10	Output	Address bus	Address output pins.
Data bus	D0–D15	63–69, 71, 73, 74, 76, 78–82	D15, B18, A19, C18, B19, B20, C17, C19, D18, D17, E17, E18, D19, C20, E19, D20	Input/ output	Data bus	16-bit bidirectional data bus pins.
Bus control	$\overline{\text{CS0}}\text{--}\overline{\text{CS3}}$	40, 42–44	A11, A12, C12, B12	Output	Chip select 0 to 3	Chip select signals for external memory or devices.
	$\overline{\text{RD}}$	38	B11	Output	Read	Indicates reading from an external device.
	$\overline{\text{WRH}}$	36	D10	Output	Upper write	Indicates writing of the upper 8 bits of external data.
	$\overline{\text{WRL}}$	35	B10	Output	Lower write	Indicates writing of the lower 8 bits of external data.
	$\overline{\text{WAIT}}$	37	A10	Input	Wait	Input for wait cycle insertion in bus cycles during external space access.
Advanced timer unit-II (ATU-II)	TCLKA TCLKB	159, 162, 219	W12, Y10, N3	Input	ATU-II timer clock input	ATU-II counter external clock Input pins.
	TIOA–TIOD	125, 127, 129, 130	U18, T17, V20, V19	Input	ATU-II input capture (channel 0)	Channel 0 input capture input pins.
	TIO1A– TIO1H	248, 250–256	C1, G4, E2, B1, D2, F3, F4, E3	Input/ output	ATU-II input capture/output compare (channel 1)	Channel 1 input capture input/output compare output pins.
	TIO2A– TIO2H	176–183	Y7, Y6, V8, U8, W7, Y5, W6, V7	Input/ output	ATU-II input capture/output compare (channel 2)	Channel 2 input capture input/output compare output pins.

**Table 1.2 Pin Functions (cont)**

Type	Symbol	Pin No.		I/O	Name	Function
		FP-256H	BP-272			
Advanced timer unit-II (ATU-II)	TIO3A– TIO3D	131–134	W20, V18, W19, Y19	Input/ output	ATU-II input capture/output compare/PWM output (channel 3)	Channel 3 input capture input/output compare/PWM output pins.
	TIO4A– TIO4D	135–138	W18, U17, Y18, V17	Input/ output	ATU-II input capture/output compare/PWM output (channel 4)	Channel 4 input capture input/output compare/PWM output pins.
	TIO5A– TIO5D	140, 142, 184, 186	W17, W16, U7, Y4	Input/ output	ATU-II input capture/output compare/PWM output (channel 5)	Channel 5 input capture input/output compare/PWM output pins.
	TO6A–TO6D	145–147, 149	U15, W15, Y15, Y14	Output	ATU-II PWM output (channel 6)	Channel 6 PWM output pins.
	TO7A–TO7D	151–154	U14, V14, W14, Y13	Output	ATU-II PWM output (channel 7)	Channel 7 PWM output pins.
	TO8A– TO8P	151–158, 195, 197–202, 204, 206–211, 213, 215	U14, V14, W14, Y13, W13, U13, V13, V12, W1, V3, W2, V2, V1, U1, T4, U2, R4, U3, T3, T2, R3, T1, R2, R1	Output	ATU-II one-shot pulse (channel 8)	Channel 8 down-counter one-shot pulse output pins.
	TI9A– TI9F	188–193	Y3, W5, Y2, W4, W3, V4	Input	ATU-II event input (channel 9)	Channel 9 event counter input pins.
	TI10	162, 216	Y10, N4	Input	ATU-II multiplied clock generation (channel 10)	Channel 10 external clock input pin.

**Table 1.2 Pin Functions (cont)**

Type	Symbol	Pin No.			I/O	Name	Function
		FP-256H	BP-272				
Advanced timer unit-II (ATU-II)	TIO11A, TIO11B	217, 218	P2, P1	Input/output	ATU-II input capture/output compare	Channel 11 input capture input/output compare output pins.	
Advanced pulse controller (APC)	PULS0–PULS7	1–6, 164, 170	E4, D3, C2, C3, B2, A2, U12, Y9	Output	APC pulse outputs 0 to 7	APC pulse output pins.	
Serial communication interface (SCI)	TxD0–TxD4	143, 165, 167, 155, 157	Y17, V11, W10, W13, V13	Output	Transmit data (channels 0 to 4)	SCI0 to SCI4 transmit data output pins.	
	RxD0–RxD4	144, 166, 168, 156, 158	Y16, U11, U10, U13, V12	Input	Receive data (channels 0 to 4)	SCI0 to SCI4 receive data input pins.	
	SCK0–SCK4	160, 162, 223, 224, 226, 164	Y12, Y10, M3, L4, L3, U12	Input/output	Serial clock (channels 0 to 4)	SCI0 to SCI4 clock input/output pins.	
Controller area network-II (HCAN-II)	HTxD0, HTxD1	157, 228, 6	V13, L1, A2	Output	Transmit data	CAN bus transmit data output pins.	
	HRxD0, HRxD1	158, 229, 170	V12, L2, Y9	input	Receive data	CAN bus receive data input pins.	
A/D converter	$AV_{CC}$	101, 119	K20, T20	Input	Analog power supply	A/D converter power supply.	
	$AV_{SS}$	99, 121	J20, U20	Input	Analog ground	A/D converter power supply.	
	$AV_{ref}$	100, 120	K19, T19	Input	Analog reference power supply	Analog reference power supply input pins.	
	AN0–AN31	86–98, 102–118, 122, 123	H17, G18, G19, F20, H18, G20, J17, H19, H20, J19, J18, K18, K17, L19, L18, L20, M20, L17, M19, N20, M18, P20, N19, P19, R20, M17, N18, P18, R19, N17, P17, R18	Input	Analog input	Analog signal input pins.	



**Table 1.2 Pin Functions (cont)**

Type	Symbol	Pin No.		I/O	Name	Function
		FP-256H	BP-272			
A/D converter	$\overline{\text{ADTRG0}}$ , $\overline{\text{ADTRG1}}$	175, 220, 221	W8, M4, N1	Input	A/D conversion trigger input	External trigger input pins for starting A/D conversion.
	ADEND	173, 222	W9, N2	Output	ADEND output	A/D2 channel 31 conversion timing monitor output pins.
	ADTO0A	135	W18	Output	PWM output	PWM output pin for multi-trigger A/D conversion.
	ADTO0B	136	U17	Output	PWM output	PWM output pin for multi-trigger A/D conversion.
	ADTO1A	137	Y18	Output	PWM output	PWM output pin for multi-trigger A/D conversion.
	ADTO1B	138	V17	Output	PWM output	PWM output pin for multi-trigger A/D conversion.
User break controller (UBC)	$\overline{\text{UBCTR}}\overline{\text{G}}$	159	W12	Output	User break trigger output	UBC condition match trigger output pin.
High-performance user debug interface (H-UDI)	TCK	236	J2	Input	Test clock	Test clock input pin.
	TMS	232	K3	Input	Test mode select	Test mode select signal input pin.
	TDI	234	K4	Input	Test data input	Instruction/data serial input pin.
	TDO	235	H1	Output	Test data output	Instruction/data serial output pin.
	$\overline{\text{TRST}}$	233	J1	Input	Test reset	Initialization signal input pin.
Advanced user debugger (AUD)	AUDATA0– AUDATA3	241–244	G1, F1, G2, E1	Input/ output	AUD data	Branch trace mode: Branch destination address output pins.  RAM monitor mode: Monitor address input / data input/output pins.
	$\overline{\text{AUDRST}}$	238	H2	Input	AUD reset	Reset signal input pin.
	AUDMD	240	H3	Input	AUD mode	Mode select signal input pin.  Branch trace mode: Low RAM monitor mode: High

**Table 1.2 Pin Functions (cont)**

Type	Symbol	Pin No.		I/O	Name	Function
		FP-256H	BP-272			
Advanced user debugger (AUD)	AUDCK	245	D1	Input/output	AUD clock	Branch trace mode: Serial clock output pin. RAM monitor mode: Serial clock input pin.
	$\overline{\text{AUDSYNC}}$	246	F2	Input/output	AUD synchronization signal	Branch trace mode: Data start position identification signal output pin. RAM monitor mode: Data start position identification signal input pin.
I/O ports	$\overline{\text{POD}}$	34	C10	Input	Port output disable	Input pin for port pin drive control when general port is set for output.
	PA0–PA15	125, 127, 129–138, 140, 142–144	U18, T17, V20, V19, W20, V18, W19, Y19, W18, U17, Y18, V17, W17, W16, Y17, Y16	Input/output	Port A	General input/output port pins. Input or output can be specified bit by bit.
	PB0–PB15	145–147, 149, 151–160, 162, 164	U15, W15, Y15, Y14, U14, V14, W14, Y13, W13, U13, V13, V12, W12, Y12, Y10, U12	Input/output	Port B	General input/output port pins. Input or output can be specified bit by bit.
	PC0–PC4	165–169	V11, U11, W10, U10, V10	Input/output	Port C	General input/output port pins. Input or output can be specified bit by bit.
	PD0–PD13	248, 250–256, 1–6	C1, G4, E2, B1, D2, F3, F4, E3, E4, D3, C2, C3, B2, A2	Input/output	Port D	General input/output port pins. Input or output can be specified bit by bit.

**Table 1.2 Pin Functions (cont)**

Type	Symbol	Pin No.			Name	Function
		FP-256H	BP-272	I/O		
I/O ports	PE0–PE15	7–10, 12, 14–19, 21, 23–26	B3, D4, C4, A3, B4, A4, C5, B5, A5, D6, B6, A6, C7, B7, A7, D8	Input/ output	Port E	General input/output port pins.  Input or output can be specified bit by bit.
	PF0–PF15	27–29, 31, 33–38, 40, 42–46	C8, B8, A8, D9, C9, C10, B10, D10, A10, B11, A11, A12, C12, B12, D12, C13	Input/ output	Port F	General input/output port pins.  Input or output can be specified bit by bit.
	PG0–PG3	170, 171, 173, 175	Y9, Y8, W9, W8	Input/ output	Port G	General input/output port pins.  Input or output can be specified bit by bit.
	PH0–PH15	63–69, 71, 73, 74, 76, 78–82	D15, B18, A19, C18, B19, B20, C17, C19, D18, D17, E17, E18, D19, C20, E19, D20	Input/ output	Port H	General input/output port pins.  Input or output can be specified bit by bit.
	PJ0–PJ15	176–184, 186, 188–193	Y7, Y6, V8, U8, W7, Y5, W6, V7, U7, Y4, Y3, W5, Y2, W4, W3, V4	Input/ output	Port J	General input/output port pins.  Input or output can be specified bit by bit.
	PK0–PK15	195, 197–202, 204, 206–211, 213, 215	W1, V3, W2, V2, V1, U1, T4, U2, R4, U3, T3, T2, R3, T1, R2, R1	Input/ output	Port K	General input/output port pins.  Input or output can be specified bit by bit.
PL0–PL13	216–224, 226, 228–231	N4, P2, P1, N3, M4, N1, N2, M3, L4, L3, L1, L2, K2, K1	Input/ output	Port L	General input/output port pins.  Input or output can be specified bit by bit.	

### 1.3.3 Pin Assignments

**Table 1.3 Pin Assignments**

Pin No.			
FP-256H	BP-272	MCU Mode	Programmer Mode
1	E4	PD8/PULS0	NC
2	D3	PD9/PULS1	NC
3	C2	PD10/PULS2	NC
4	C3	PD11/PULS3	NC
5	B2	PD12/PULS4	NC
6	A2	PD13/PULS6/HTxD0/HTxD1	NC
7	B3	PE0/A0	A0
8	D4	PE1/A1	A1
9	C4	PE2/A2	A2
10	A3	PE3/A3	A3
11	D5	V <sub>cc</sub>	V <sub>cc</sub>
12	B4	PE4/A4	A4
13	*	V <sub>ss</sub>	V <sub>ss</sub>
14	A4	PE5/A5	A5
15	C5	PE6/A6	A6
16	B5	PE7/A7	A7
17	A5	PE8/A8	A8
18	D6	PE9/A9	A9
19	B6	PE10/A10	A10
20	C6	PV <sub>cc</sub> 1	V <sub>cc</sub>
21	A6	PE11/A11	A11
22	*	V <sub>ss</sub>	V <sub>ss</sub>
23	C7	PE12/A12	A12
24	B7	PE13/A13	A13
25	A7	PE14/A14	A14
26	D8	PE15/A15	A15
27	C8	PF0/A16	A16
28	B8	PF1/A17	A17
29	A8	PF2/A18	A18
30	B9	V <sub>CL</sub>	V <sub>CL</sub>

**Table 1.3 Pin Assignments (cont)**

Pin No.			
FP-256H	BP-272	MCU Mode	Programmer Mode
31	D9	PF3/A19	A19
32	*	Vss	Vss
33	C9	PF4/A20	NC
34	C10	PF5/A21/ $\overline{\text{POD}}$	NC
35	B10	PF6/ $\overline{\text{WRL}}$	NC
36	D10	PF7/ $\overline{\text{WRH}}$	NC
37	A10	PF8/ $\overline{\text{WAIT}}$	Vcc
38	B11	PF9/ $\overline{\text{RD}}$	NC
39	C11	PVcc1	Vcc
40	A11	PF10/ $\overline{\text{CS0}}$	NC
41	*	Vss	Vss
42	A12	PF11/ $\overline{\text{CS1}}$	Vcc
43	C12	PF12/ $\overline{\text{CS2}}$	Vcc
44	B12	PF13/ $\overline{\text{CS3}}$	Vss
45	D12	PF14/ $\overline{\text{BACK}}$	NC
46	C13	PF15/ $\overline{\text{BREQ}}$	Vcc
47	*	Vss	Vss
48	A13	CK	NC
49	D13	Vcc	Vcc
50	C14	MD2	Vss
51	A14	EXTAL	EXTAL
52	B14	Vcc	Vcc
53	A15	XTAL	XTAL
54	*	Vss	Vss
55	C15	MD1	Vcc
56	D14	FWE	FWE
57	A16	$\overline{\text{HSTBY}}$	Vcc
58	B16	$\overline{\text{RES}}$	$\overline{\text{RES}}$
59	C16	MD0	Vcc
60	A17	PLLVcc	PLLVcc
61	B17	PLLCAP	PLLCAP

**Table 1.3 Pin Assignments (cont)**

Pin No.			
FP-256H	BP-272	MCU Mode	Programmer Mode
62	A18	PLL <sub>Vss</sub>	PLL <sub>Vss</sub>
63	D15	PH0/D0	D0
64	B18	PH1/D1	D1
65	A19	PH2/D2	D2
66	C18	PH3/D3	D3
67	B19	PH4/D4	D4
68	B20	PH5/D5	D5
69	C17	PH6/D6	D6
70	D16	PV <sub>cc1</sub>	V <sub>cc</sub>
71	C19	PH7/D7	D7
72	*	V <sub>ss</sub>	V <sub>ss</sub>
73	D18	PH8/D8	NC
74	D17	PH9/D9	NC
75	F17	V <sub>cc</sub>	V <sub>cc</sub>
76	E17	PH10/D10	NC
77	*	V <sub>ss</sub>	V <sub>ss</sub>
78	E18	PH11/D11	NC
79	D19	PH12/D12	NC
80	C20	PH13/D13	NC
81	E19	PH14/D14	NC
82	D20	PH15/D15	NC
83	F18	PV <sub>cc1</sub>	V <sub>cc</sub>
84	E20	NMI	V <sub>ss</sub>
85	*	V <sub>ss</sub>	V <sub>ss</sub>
86	H17	AN0	NC
87	G18	AN1	NC
88	G19	AN2	NC
89	F20	AN3	NC
90	H18	AN4	NC
91	G20	AN5	NC
92	J17	AN6	NC

**Table 1.3 Pin Assignments (cont)**

Pin No.		MCU Mode	Programmer Mode
FP-256H	BP-272		
93	H19	AN7	NC
94	H20	AN8	NC
95	J19	AN9	NC
96	J18	AN10	NC
97	K18	AN11	NC
98	K17	AN12	NC
99	J20	AVss	Vss
100	K19	AVref	Vcc
101	K20	AVcc	Vcc
102	L19	AN13	NC
103	L18	AN14	NC
104	L20	AN15	NC
105	M20	AN16	NC
106	L17	AN17	NC
107	M19	AN18	NC
108	N20	AN19	NC
109	M18	AN20	NC
110	P20	AN21	NC
111	N19	AN22	NC
112	P19	AN23	NC
113	R20	AN24	NC
114	M17	AN25	NC
115	N18	AN26	NC
116	P18	AN27	NC
117	R19	AN28	NC
118	N17	AN29	NC
119	T20	AVcc	Vcc
120	T19	AVref	Vcc
121	U20	AVss	Vss
122	P17	AN30	NC
123	R18	AN31	NC

**Table 1.3 Pin Assignments (cont)**

Pin No.		MCU Mode	Programmer Mode
FP-256H	BP-272		
124	R17	WDTOVF	NC
125	U18	PA0/TIOA	NC
126	*	Vss	Vss
127	T17	PA1/TIOB	NC
128	U19	PVcc2	Vcc
129	V20	PA2/TIOC	NC
130	V19	PA3/TIOD	NC
131	W20	PA4/TIO3A	NC
132	V18	PA5/TIO3B	NC
133	W19	PA6/TIO3C	NC
134	Y19	PA7/TIO3D	NC
135	W18	PA8/TIO4A/ADTO0A	NC
136	U17	PA9/TIO4B/ADTO0B	NC
137	Y18	PA10/TIO4C/ADTO1A	NC
138	V17	PA11/TIO4D/ADTO1B	NC
139	U16	Vcc	Vcc
140	W17	PA12/TIO5A	NC
141	*	Vss	Vss
142	W16	PA13/TIO5B	NC
143	Y17	PA14/TxD0	NC
144	Y16	PA15/RxD0	NC
145	U15	PB0/TO6A	NC
146	W15	PB1/TO6B	NC
147	Y15	PB2/TO6C	NC
148	V15	PVcc2	Vcc
149	Y14	PB3/TO6D	NC
150	*	Vss	Vss
151	U14	PB4/TO7A/TO8A	NC
152	V14	PB5/TO7B/TO8B	NC
153	W14	PB6/TO7C/TO8C	NC
154	Y13	PB7/TO7D/TO8D	NC



**Table 1.3 Pin Assignments (cont)**

Pin No.			
FP-256H	BP-272	MCU Mode	Programmer Mode
155	W13	PB8/TxD3/TO8E	NC
156	U13	PB9/RxD3/TO8F	NC
157	V13	PB10/TxD4/HTxD0/TO8G	NC
158	V12	PB11/RxD4/HRxD0/TO8H	NC
159	W12	PB12/TCLKA/ $\overline{UBCTR\overline{G}}$	NC
160	Y12	PB13/SCK0	NC
161	Y11	V <sub>CL</sub>	V <sub>CL</sub>
162	Y10	PB14/SCK1/TCLKB/TI10	NC
163	*	V <sub>ss</sub>	V <sub>ss</sub>
164	U12	PB15/PULS5/SCK2	NC
165	V11	PC0/TxD1	NC
166	U11	PC1/RxD1	NC
167	W10	PC2/TxD2	NC
168	U10	PC3/RxD2	NC
169	V10	PC4/ $\overline{IRQ0}$	NC
170	Y9	PG0/PULS7/HRxD0/HRxD1	NC
171	Y8	PG1/ $\overline{IRQ1}$	NC
172	V9	PV <sub>cc2</sub>	V <sub>cc</sub>
173	W9	PG2/ $\overline{IRQ2}$ /ADEND	NC
174	*	V <sub>ss</sub>	V <sub>ss</sub>
175	W8	PG3/ $\overline{IRQ3}$ /ADTRG $\overline{0}$	NC
176	Y7	PJ0/TIO2A	NC
177	Y6	PJ1/TIO2B	NC
178	V8	PJ2/TIO2C	NC
179	U8	PJ3/TIO2D	NC
180	W7	PJ4/TIO2E	NC
181	Y5	PJ5/TIO2F	NC
182	W6	PJ6/TIO2G	NC
183	V7	PJ7/TIO2H	NC
184	U7	PJ8/TIO5C	NC
185	*	V <sub>ss</sub>	V <sub>ss</sub>

**Table 1.3 Pin Assignments (cont)**

Pin No.		MCU Mode	Programmer Mode
FP-256H	BP-272		
186	Y4	PJ9/TIO5D	NC
187	U6	Vcc	Vcc
188	Y3	PJ10/TI9A	NC
189	W5	PJ11/TI9B	NC
190	Y2	PJ12/TI9C	NC
191	W4	PJ13/TI9D	NC
192	W3	PJ14/TI9E	NC
193	V4	PJ15/TI9F	NC
194	V5	PVcc2	Vcc
195	W1	PK0/TO8A	NC
196	*	Vss	Vss
197	V3	PK1/TO8B	NC
198	W2	PK2/TO8C	NC
199	V2	PK3/TO8D	NC
200	V1	PK4/TO8E	NC
201	U1	PK5/TO8F	NC
202	T4	PK6/TO8G	NC
203	U4	Vcc	Vcc
204	U2	PK7/TO8H	NC
205	*	Vss	Vss
206	R4	PK8/TO8I	NC
207	U3	PK9/TO8J	NC
208	T3	PK10/TO8K	NC
209	T2	PK11/TO8L	NC
210	R3	PK12/TO8M	NC
211	T1	PK13/TO8N	NC
212	P3	PVcc2	Vcc
213	R2	PK14/TO8O	NC
214	*	Vss	Vss
215	R1	PK15/TO8P	NC
216	N4	PL0/TI10	NC

**Table 1.3 Pin Assignments (cont)**

Pin No.		MCU Mode	Programmer Mode
FP-256H	BP-272		
217	P2	PL1/TIO11A/ $\overline{\text{IRQ6}}$	NC
218	P1	PL2/TIO11B/ $\overline{\text{IRQ7}}$	$\overline{\text{CE}}$
219	N3	PL3/TCLKB	NC
220	M4	PL4/ $\overline{\text{ADTRG0}}$	NC
221	N1	PL5/ $\overline{\text{ADTRG1}}$	NC
222	N2	PL6/ADEND	NC
223	M3	PL7/SCK2	NC
224	L4	PL8/SCK3	NC
225	M2	$V_{\text{CL}}$	$V_{\text{CL}}$
226	L3	PL9/SCK4/ $\overline{\text{IRQ5}}$	$\overline{\text{WE}}$
227	*	Vss	Vss
228	L1	PL10/HTxD0/HTxD1/HTxD0 & HTxD1	NC
229	L2	PL11/HRxD0/HRxD1/HRxD0 & HRxD1	NC
230	K2	PL12/ $\overline{\text{IRQ4}}$	$\overline{\text{OE}}$
231	K1	PL13/ $\overline{\text{IRQOUT}}$	NC
232	K3	TMS	NC
233	J1	$\overline{\text{TRST}}$	NC
234	K4	TDI	NC
235	H1	TDO	NC
236	J2	TCK	NC
237	J3	Vcc	Vcc
238	H2	$\overline{\text{AUDRST}}$	NC
239	*	Vss	Vss
240	H3	AUDMD	NC
241	G1	AUDATA0	NC
242	F1	AUDATA1	NC
243	G2	AUDATA2	NC
244	E1	AUDATA3	NC
245	D1	AUDCK	NC
246	F2	$\overline{\text{AUDSYNC}}$	NC
247	H4	PVcc2	Vcc

**Table 1.3 Pin Assignments (cont)**

Pin No.		MCU Mode	Programmer Mode
FP-256H	BP-272		
248	C1	PD0/TIO1A	NC
249	*	Vss	Vss
250	G4	PD1/TIO1B	NC
251	E2	PD2/TIO1C	NC
252	B1	PD3/TIO1D	NC
253	D2	PD4/TIO1E	NC
254	F3	PD5/TIO1F	NC
255	F4	PD6/TIO1G	NC
256	E3	PD7/TIO1H	NC
—	A1	NC	NC
—	A20	NC	NC
—	Y1	NC	NC
—	Y20	NC	NC

\* Vss is connected in the board.



# Section 2 CPU

## 2.1 Register Configuration

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

In addition, the FPU has eighteen internal registers: sixteen 32-bit floating-point registers and two 32-bit floating-point system registers.

### 2.1.1 General Registers (Rn)

The sixteen 32-bit general registers (Rn) are numbered R0–R15. General registers are used for data processing and address calculation. R0 is also used as an index register. Several instructions have R0 fixed as their only usable register. R15 is used as the hardware stack pointer (SP). Saving and recovering the status register (SR) and program counter (PC) in exception processing is accomplished by referencing the stack using R15. Figure 2.1 shows the general registers.

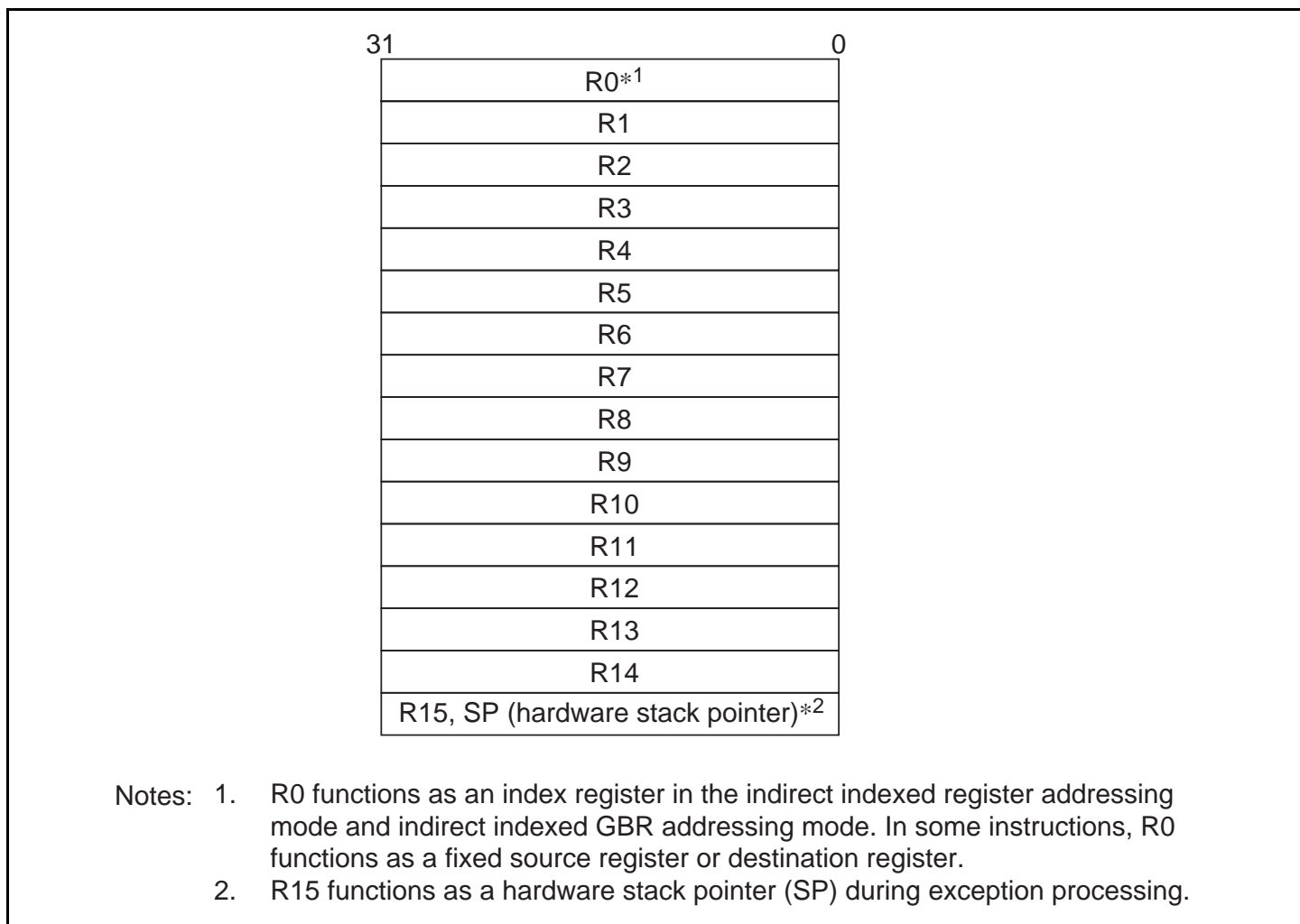
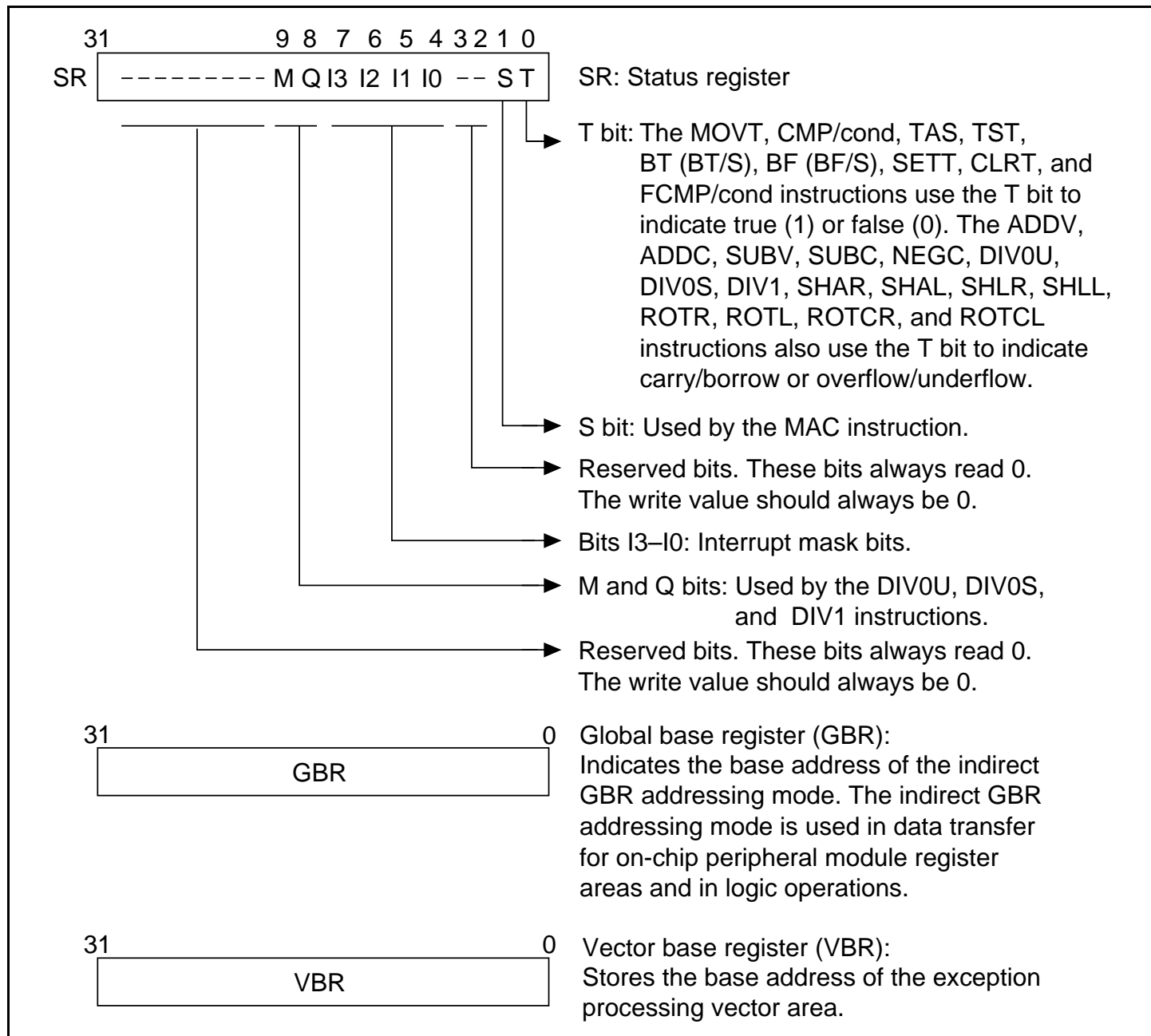


Figure 2.1 General Registers

## 2.1.2 Control Registers

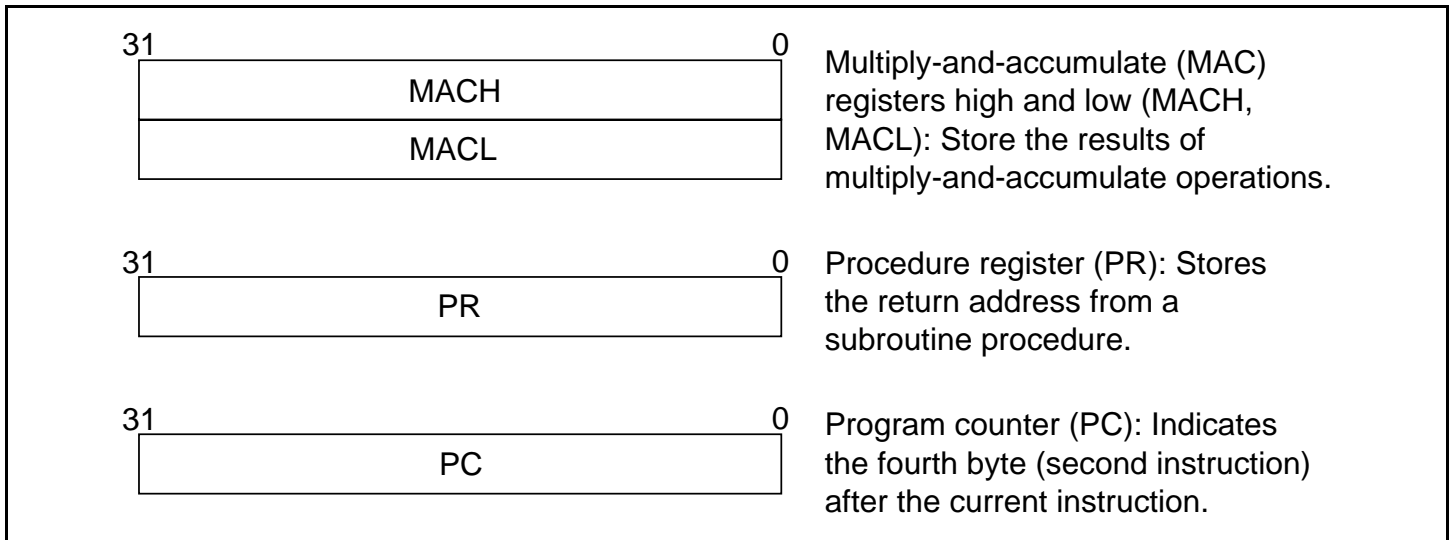
The 32-bit control registers consist of the 32-bit status register (SR), global base register (GBR), and vector base register (VBR). The status register indicates processing states. The global base register functions as a base address for the indirect GBR addressing mode to transfer data to the registers of on-chip peripheral modules. The vector base register functions as the base address of the exception processing vector area (including interrupts). Figure 2.2 shows the control registers.



**Figure 2.2 Control Register Configuration**

### 2.1.3 System Registers

System registers consist of four 32-bit registers: high and low multiply and accumulate registers (MACH and MACL), the procedure register (PR), and the program counter (PC). The multiply-and-accumulate registers store the results of multiply-and-accumulate operations. The procedure register stores the return address from a subroutine procedure. The program counter stores program addresses to control the flow of the processing. Figure 2.3 shows the system registers.

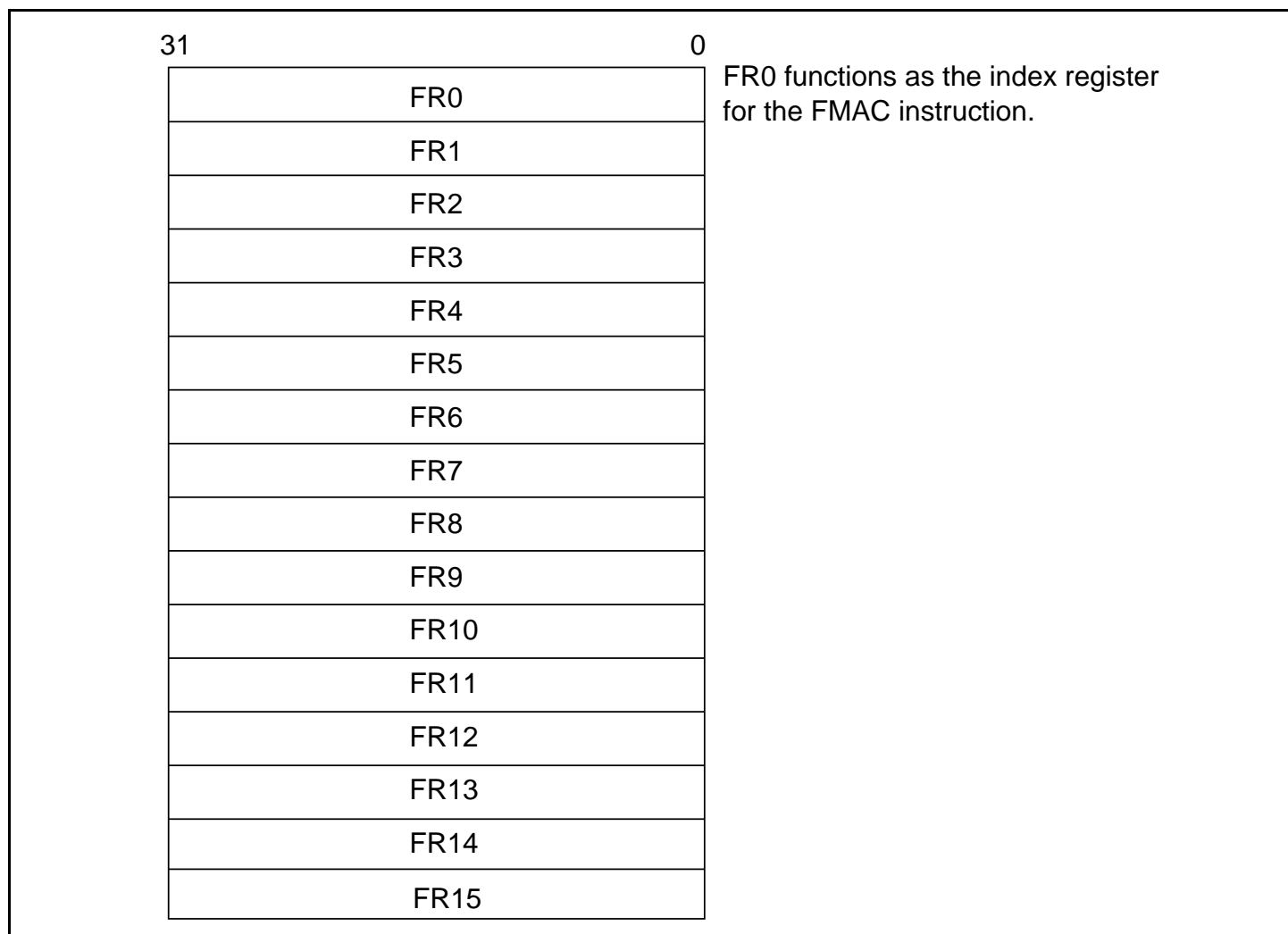


**Figure 2.3 System Register Configuration**



## 2.1.4 Floating-Point Registers

There are sixteen 32-bit floating-point registers, designated FR0 to FR15, which are used by floating-point instructions. FR0 functions as the index register for the FMAC instruction. These registers are incorporated into the floating-point unit (FPU). For details, see section 3, Floating-Point Unit (FPU).

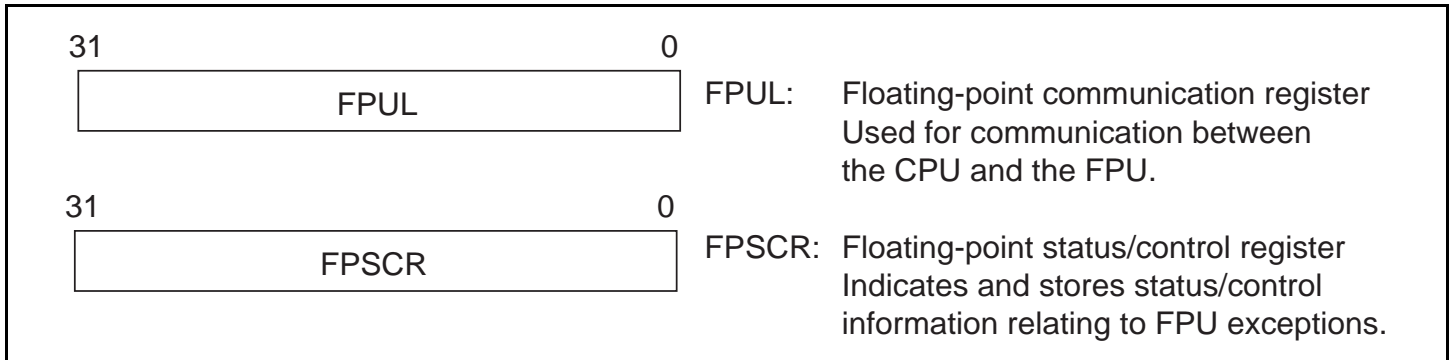


**Figure 2.4 Floating-Point Registers**

## 2.1.5 Floating-Point System Registers

There are two 32-bit floating-point system registers: the floating-point communication register (FPUL) and the floating-point status/control register (FPSCR). FPUL is used for communication between the CPU and the floating-point unit (FPU). FPSCR indicates and stores status/control information relating to FPU exceptions.

These registers are incorporated into the floating-point unit (FPU). For details, see section 3, Floating-Point Unit (FPU).



**Figure 2.5 Floating-Point System Registers**

## 2.1.6 Initial Values of Registers

Table 2.1 lists the values of the registers after reset.

**Table 2.1 Initial Values of Registers**

Classification	Register	Initial Value
General registers	R0–R14	Undefined
	R15 (SP)	Value of the stack pointer in the vector address table
Control registers	SR	Bits I3–I0 are 1111 (H'F), reserved bits are 0, and other bits are undefined
	GBR	Undefined
	VBR	H'00000000
System registers	MACH, MACL, PR	Undefined
	PC	Value of the program counter in the vector address table
Floating-point registers	FR0–FR15	Undefined
Floating-point system registers	FPUL	Undefined
	FPSCR	H'00040001

## 2.2 Data Formats

### 2.2.1 Data Format in Registers

Register operands are always longwords (32 bits). When the memory operand is only a byte (8 bits) or a word (16 bits), it is sign-extended into a longword when loaded into a register (figure 2.6).

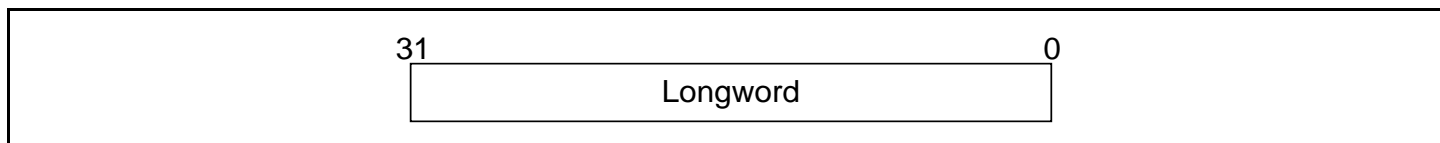


Figure 2.6 Data Format in Registers

### 2.2.2 Data Formats in Memory

Memory data formats are classified into bytes, words, and longwords. Byte data can be accessed from any address, but an address error will occur if an attempt is made to access word data starting from an address other than  $2n$  or longword data starting from an address other than  $4n$ . In such cases, the data accessed cannot be guaranteed. The hardware stack area, referred to by the hardware stack pointer (SP, R15), uses only longword data starting from address  $4n$  because this area holds the program counter and status register (figure 2.7).

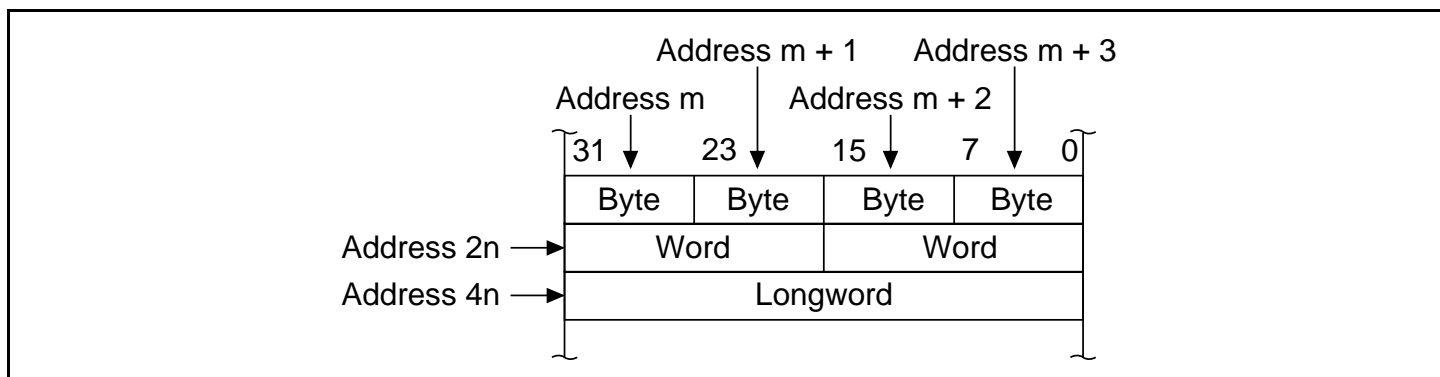


Figure 2.7 Data Formats in Memory

### 2.2.3 Immediate Data Format

Byte (8 bit) immediate data resides 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.

Word or longword immediate data is not located in the instruction code, but instead is stored in a memory table. An immediate data transfer instruction (MOV) accesses the memory table using the PC relative addressing mode with displacement.

## 2.3 Instruction Features

### 2.3.1 RISC-Type Instruction Set

All instructions are RISC type. This section details their functions.

**16-Bit Fixed Length:** All instructions are 16 bits long, increasing program code efficiency.

**One Instruction per Cycle:** The microprocessor can execute basic instructions in one cycle using the pipeline system. Instructions are executed in 25 ns at 40 MHz.

**Data Length:** Longword is the standard data length for all operations. Memory can be accessed in bytes, words, or longwords. Byte or word data accessed from memory is sign-extended and handled as longword data. Immediate data is sign-extended for arithmetic operations or zero-extended for logic operations. It also is handled as longword data (table 2.2).

**Table 2.2 Sign Extension of Word Data**

SH7058 CPU	Description	Example of Conventional CPU
MOV.W    @(disp,PC),R1	Data is sign-extended to 32 bits, and R1 becomes H'00001234. It is next operated upon by an ADD instruction.	ADD.W    #H'1234,R0
ADD       R1,R0		
. . . . .		
.DATA.W   H'1234		

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

**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.

**Delayed Branch Instructions:** Unconditional branch instructions are delayed branch instructions. With a delayed branch instruction, the branch is taken after execution of the instruction following the delayed branch instruction. There are two types of conditional branch instructions: delayed branch instructions and ordinary branch instructions.

**Table 2.3 Delayed Branch Instructions**

SH7058 CPU		Description	Example of Conventional CPU	
BRA	TRGET	Executes the ADD before branching to TRGET.	ADD.W	R1,R0
ADD	R1,R0		BRA	TRGET

**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 + 64bit  $\rightarrow$  64-bit multiply-and-accumulate operations are executed in two to four cycles.

**T Bit:** The T bit in the status register changes according to the result of the comparison, and in turn is the condition (true/false) that determines if the program will branch. The number of instructions that change the T bit is kept to a minimum to improve the processing speed (table 2.4).

**Table 2.4 T Bit**

SH7058 CPU		Description	Example of Conventional CPU	
CMP/GE	R1,R0	T bit is set when $R0 \geq R1$ . The program branches to TRGET0 when $R0 \geq R1$ and to TRGET1 when $R0 < R1$ .	CMP.W	R1,R0
BT	TRGET0		BGE	TRGET0
BF	TRGET1		BLT	TRGET1
ADD	#1,R0	T bit is not changed by ADD. T bit is set when $R0 = 0$ . The program branches if $R0 = 0$ .	SUB.W	#1,R0
CMP/EQ	#0,R0		BEQ	TRGET
BT	TRGET			

**Immediate Data:** Byte (8-bit) immediate data resides in the instruction code. Word or longword immediate data is not input via instruction codes but is stored in a memory table. An immediate data transfer instruction (MOV) accesses the memory table using the PC relative addressing mode with displacement (table 2.5).

**Table 2.5 Immediate Data Accessing**

Classification	SH7058 CPU		Example of Conventional CPU
8-bit immediate	MOV	#H'12,R0	MOV.B #H'12,R0
16-bit immediate	MOV.W	@(disp,PC),R0	MOV.W #H'1234,R0
		.....	
	.DATA.W	H'1234	
32-bit immediate	MOV.L	@(disp,PC),R0	MOV.L #H'12345678,R0
		.....	
	.DATA.L	H'12345678	

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

**Absolute Address:** When data is accessed by absolute address, the value already in the absolute address is placed in the memory table. Loading the immediate data when the instruction is executed transfers that value to the register and the data is accessed in the indirect register addressing mode (table 2.6).

**Table 2.6 Absolute Address Accessing**

Classification	SH7058 CPU		Example of Conventional CPU
Absolute address	MOV.L	@(disp,PC),R1	MOV.B @H'12345678,R0
	MOV.B	@R1,R0	
		.....	
	.DATA.L	H'12345678	

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

**16-Bit/32-Bit Displacement:** When data is accessed by 16-bit or 32-bit displacement, the pre-existing displacement value is placed in the memory table. Loading the immediate data when the instruction is executed transfers that value to the register and the data is accessed in the indirect indexed register addressing mode (table 2.7).

**Table 2.7 Displacement Accessing**

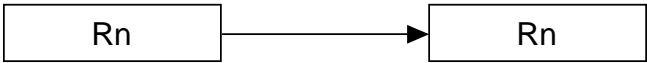
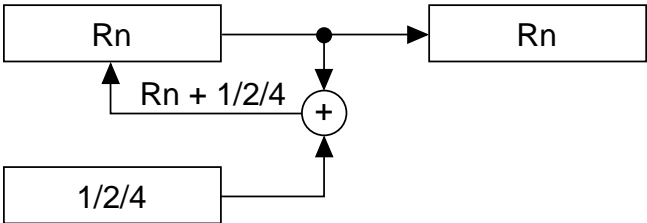
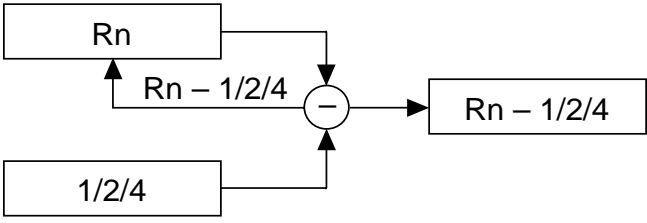
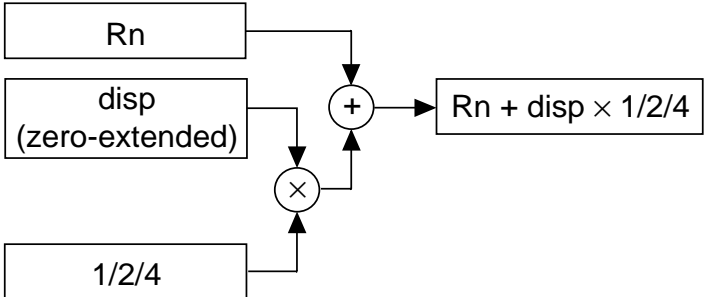
Classification	SH7058 CPU		Example of Conventional CPU
16-bit displacement	MOV.W	@(disp,PC),R0	MOV.W @(H'1234,R1),R2
	MOV.W	@(R0,R1),R2	
		.....	
	.DATA.W	H'1234	

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

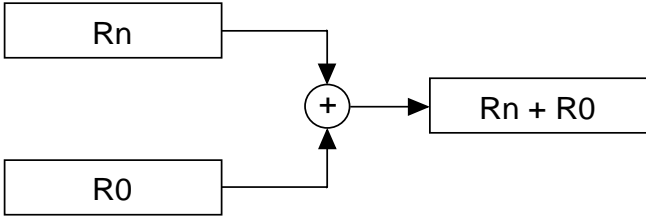
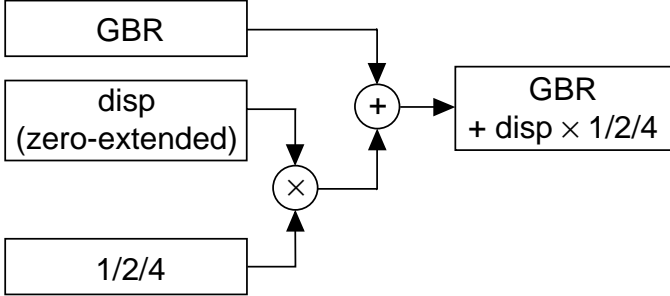
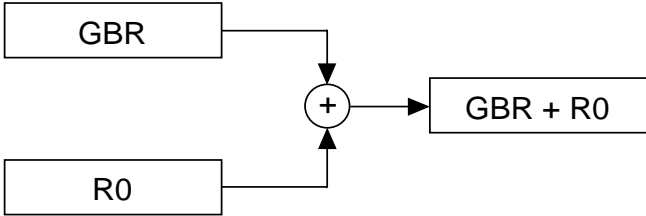
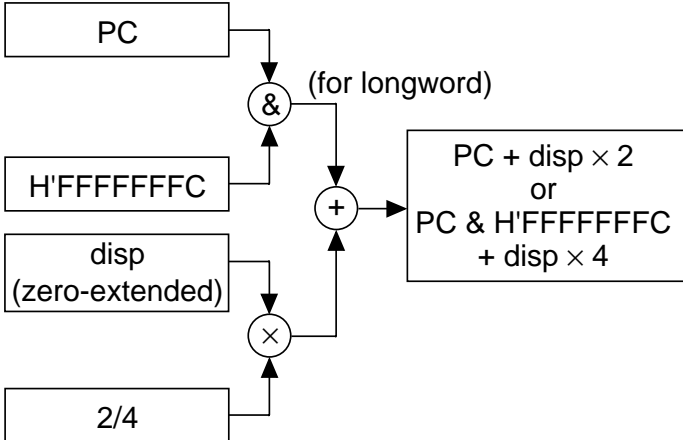
## 2.3.2 Addressing Modes

Table 2.8 describes addressing modes and effective address calculation.

**Table 2.8 Addressing Modes and Effective Addresses**

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Direct register addressing	Rn	The effective address is register Rn. (The operand is the contents of register Rn.)	—
Indirect register addressing	@Rn	The effective address is the contents of register Rn. Rn 	
Post-increment indirect register addressing	@Rn+	The effective address is the contents of register Rn. Rn A constant is added to the content of Rn after the instruction is executed. 1 is added for a byte operation, 2 for a word operation, and 4 for a longword operation. 	<p>(After the instruction executes)</p> <p>Byte: <math>Rn + 1 \rightarrow Rn</math></p> <p>Word: <math>Rn + 2 \rightarrow Rn</math></p> <p>Longword: <math>Rn + 4 \rightarrow Rn</math></p>
Pre-decrement indirect register addressing	@-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. 	<p>Byte: <math>Rn - 1 \rightarrow Rn</math></p> <p>Word: <math>Rn - 2 \rightarrow Rn</math></p> <p>Longword: <math>Rn - 4 \rightarrow Rn</math> (Instruction executed with Rn after calculation)</p>
Indirect register addressing with displacement	@(disp:4, Rn)	The effective address is Rn plus a 4-bit displacement (disp). The value of disp is zero-extended, and remains the same for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	<p>Byte: <math>Rn + disp</math></p> <p>Word: <math>Rn + disp \times 2</math></p> <p>Longword: <math>Rn + disp \times 4</math></p>

**Table 2.8 Addressing Modes and Effective Addresses (cont)**

Addressing Mode	Instruction Format	Effective Address Calculation	Equation
Indirect indexed register addressing	@(R0, Rn)	The effective address is the Rn value plus R0. 	$Rn + R0$
Indirect GBR addressing with displacement	@(disp:8, GBR)	The effective address is the GBR value plus an 8-bit displacement (disp). The value of disp is zero-extended, and remains the same for a byte operation, is doubled for a word operation, and is quadrupled for a longword operation. 	Byte: $GBR + disp$ Word: $GBR + disp \times 2$ Longword: $GBR + disp \times 4$
Indirect indexed GBR addressing	@(R0, GBR)	The effective address is the GBR value plus R0. 	$GBR + R0$
Indirect PC addressing with displacement	@(disp:8, PC)	The effective address is the PC value plus an 8-bit displacement (disp). The value of disp is zero-extended, and is doubled for a word operation, and quadrupled for a longword operation. For a longword operation, the lowest two bits of the PC value are masked. 	Word: $PC + disp \times 2$ Longword: $PC \& H'FFFFFFFC + disp \times 4$



**Table 2.8 Addressing Modes and Effective Addresses (cont)**

Addressing Mode	Instruction Format	Effective Addresses Calculation	Equation
PC relative addressing	$disp:8$	The effective address is the PC value sign-extended with an 8-bit displacement ( $disp$ ), doubled, and added to the PC value.	$PC + disp \times 2$
	$disp:12$	The effective address is the PC value sign-extended with a 12-bit displacement ( $disp$ ), doubled, and added to the PC value.	$PC + disp \times 2$
	$R_n$	The effective address is the register PC value plus $R_n$ .	$PC + R_n$
Immediate addressing	$\#imm:8$	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 quadrupled.	—

### 2.3.3 Instruction Format

Table 2.9 lists the instruction formats for the source operand and the destination operand. 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
- iii: Immediate data
- dddd: Displacement

**Table 2.9 Instruction Formats**

Instruction Formats	Source Operand	Destination Operand	Example
0 format <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> <span style="float: left;">15</span> <span style="float: right;">0</span> <div style="display: flex; justify-content: space-around; width: 100%;"> <span>xxxx</span> <span>xxxx</span> <span>xxxx</span> <span>xxxx</span> </div> </div>	—	—	NOF
n format <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> <span style="float: left;">15</span> <span style="float: right;">0</span> <div style="display: flex; justify-content: space-around; width: 100%;"> <span>xxxx</span> <span>nnnn</span> <span>xxxx</span> <span>xxxx</span> </div> </div>	—	nnnn: Direct register	MOVT Rn
	Control register or system register	nnnn: Direct register	STS MACH, Rn
	Control register or system register	nnnn: Indirect pre-decrement register	STC.L SR, @-Rn
m format <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> <span style="float: left;">15</span> <span style="float: right;">0</span> <div style="display: flex; justify-content: space-around; width: 100%;"> <span>xxxx</span> <span>mmmm</span> <span>xxxx</span> <span>xxxx</span> </div> </div>	mmmm: Direct register	Control register or system register	LDC Rm, SR
	mmmm: Indirect post-increment register	Control register or system register	LDC.L @Rm+, SR
	mmmm: Direct register	—	JMP @Rm
	mmmm: PC relative using Rm	—	BRAF Rm

**Table 2.9 Instruction Formats (cont)**

Instruction Formats	Source Operand	Destination Operand	Example				
nm format 15 <span style="float:right">0</span> <table border="1" style="width:100%; text-align:center;"><tr><td>xxxx</td><td>nnnn</td><td>mmmm</td><td>xxxx</td></tr></table>	xxxx	nnnn	mmmm	xxxx	mmmm: Direct register	nnnn: Direct register	ADD Rm, Rn
xxxx	nnnn	mmmm	xxxx				
	mmmm: Direct register	nnnn: Indirect register	MOV.L Rm, @Rn				
	mmmm: Indirect post-increment register (multiply-and-accumulate) nnnn*: Indirect post-increment register (multiply-and-accumulate)	MACH, MACL	MAC.W @Rm+, @Rn+				
	mmmm: Indirect post-increment register	nnnn: Direct register	MOV.L @Rm+, Rn				
	mmmm: Direct register	nnnn: Indirect pre-decrement register	MOV.L Rm, @-Rn				
	mmmm: Direct register	nnnn: Indirect indexed register	MOV.L Rm, @(R0, Rn)				
md format 15 <span style="float:right">0</span> <table border="1" style="width:100%; text-align:center;"><tr><td>xxxx</td><td>xxxx</td><td>mmmm</td><td>dddd</td></tr></table>	xxxx	xxxx	mmmm	dddd	mmmmdddd: Indirect register with displacement	R0 (Direct register)	MOV.B @(disp, Rn), R0
xxxx	xxxx	mmmm	dddd				
nd4 format 15 <span style="float:right">0</span> <table border="1" style="width:100%; text-align:center;"><tr><td>xxxx</td><td>xxxx</td><td>nnnn</td><td>dddd</td></tr></table>	xxxx	xxxx	nnnn	dddd	R0 (Direct register)	nnnndddd: Indirect register with displacement	MOV.B R0, @(disp, Rn)
xxxx	xxxx	nnnn	dddd				
nmd format 15 <span style="float:right">0</span> <table border="1" style="width:100%; text-align:center;"><tr><td>xxxx</td><td>nnnn</td><td>mmmm</td><td>dddd</td></tr></table>	xxxx	nnnn	mmmm	dddd	mmmm: Direct register	nnnndddd: Indirect register with displacement	MOV.L Rm, @(disp, Rn)
xxxx	nnnn	mmmm	dddd				
	mmmmdddd: Indirect register with displacement	nnnn: Direct register	MOV.L @(disp, Rm), Rn				

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

**Table 2.9 Instruction Formats (cont)**

Instruction Formats	Source Operand	Destination Operand	Example
d format 15 <span style="float: right;">0</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;">             xxxx    xxxx    dddd    dddd           </div>	dddddddd: Indirect GBR with displacement R0 (Direct register) dddddddd: PC relative with displacement —	R0 (Direct register) dddddddd: Indirect GBR with displacement R0 (Direct register) dddddddd: PC relative	MOV.L @(disp,GBR),R0 MOV.L R0,@(disp,GBR) MOVA @(disp,PC),R0 BF      label
d12 format 15 <span style="float: right;">0</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;">             xxxx    dddd    dddd    dddd           </div>	—	dddddddddddd: PC relative	BRA      label (label = disp + PC)
nd8 format 15 <span style="float: right;">0</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;">             xxxx    nnnn    dddd    dddd           </div>	dddddddd: PC relative with displacement	nnnn: Direct register	MOV.L @(disp,PC),Rn
i format 15 <span style="float: right;">0</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;">             xxxx    xxxx    iiii    iiii           </div>	iiiiii: Immediate iiiiii: Immediate iiiiii: Immediate	Indirect indexed GBR R0 (Direct register) —	AND.B #imm,@(R0,GBR) AND    #imm,R0 TRAPA #imm
ni format 15 <span style="float: right;">0</span> <div style="border: 1px solid black; padding: 2px; display: inline-block;">             xxxx    nnnn    iiii    iiii           </div>	iiiiii: Immediate	nnnn: Direct register	ADD    #imm,Rn

## 2.4 Instruction Set by Classification

### 2.4.1 Instruction Set by Classification

Table 2.10 lists the instructions according to their classification.

**Table 2.10 Classification of Instructions**

<b>Classification</b>	<b>Types</b>	<b>Operation Code</b>	<b>Function</b>	<b>No. of Instructions</b>
Data transfer	5	MOV	Data transfer, immediate data transfer, peripheral module data transfer, structure data transfer	39
		MOVA	Effective address transfer	
		MOVT	T bit transfer	
		SWAP	Swap of upper and lower bytes	
		XTRCT	Extraction of the middle of registers connected	
Arithmetic operations	21	ADD	Binary addition	33
		ADDC	Binary addition with carry	
		ADDV	Binary addition with overflow check	
		CMP/cond	Comparison	
		DIV1	Division	
		DIV0S	Initialization of signed division	
		DIV0U	Initialization of unsigned division	
		DMULS	Signed double-length multiplication	
		DMULU	Unsigned double-length multiplication	
		DT	Decrement and test	
		EXTS	Sign extension	
		EXTU	Zero extension	
		MAC	Multiply-and-accumulate, double-length multiply-and-accumulate operation	
		MUL	Double-length multiply operation	
		MULS	Signed multiplication	
		MULU	Unsigned multiplication	
		NEG	Negation	
		NEGC	Negation with borrow	
		SUB	Binary subtraction	
		SUBC	Binary subtraction with borrow	
SUBV	Binary subtraction with underflow			

**Table 2.10 Classification of Instructions (cont)**

Classification	Types	Operation		No. of Instructions
		Code	Function	
Logic operations	6	AND	Logical AND	14
		NOT	Bit inversion	
		OR	Logical OR	
		TAS	Memory test and bit set	
		TST	Logical AND and T bit set	
		XOR	Exclusive OR	
Shift	10	ROTL	One-bit left rotation	14
		ROTR	One-bit right rotation	
		ROTCL	One-bit left rotation with T bit	
		ROTCR	One-bit right rotation with T bit	
		SHAL	One-bit arithmetic left shift	
		SHAR	One-bit arithmetic right shift	
		SHLL	One-bit logical left shift	
		SHLLn	n-bit logical left shift	
		SHLR	One-bit logical right shift	
		SHLRn	n-bit logical right shift	
Branch	9	BF	Conditional branch, conditional branch with delay (Branch when T = 0)	11
		BT	Conditional branch, conditional branch with delay (Branch when T = 1)	
		BRA	Unconditional branch	
		BRAF	Unconditional branch	
		BSR	Branch to subroutine procedure	
		BSRF	Branch to subroutine procedure	
		JMP	Unconditional branch	
		JSR	Branch to subroutine procedure	
		RTS	Return from subroutine procedure	

**Table 2.10 Classification of Instructions (cont)**

<b>Classification</b>	<b>Types</b>	<b>Operation Code</b>	<b>Function</b>	<b>No. of Instructions</b>
System control	11	CLRT	T bit clear	31
		CLRMAC	MAC register clear	
		LDC	Load to control register	
		LDS	Load to system register	
		NOP	No operation	
		RTE	Return from exception processing	
		SETT	T bit set	
		SLEEP	Transition to power-down mode	
		STC	Store control register data	
		STS	Store system register data	
		TRAPA	Trap exception handling	
Floating-point instructions	15	FABS	Floating-point absolute value	22
		FADD	Floating-point addition	
		FCMP	Floating-point comparison	
		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	Integer-to-floating-point conversion	
		FMAC	Floating-point multiply-and-accumulate operation	
		FMOV	Floating-point data transfer	
		FMUL	Floating-point multiplication	
		FNEG	Floating-point sign inversion	
		FSTS	Floating-point store from system register FPUL	
		FSUB	Floating-point subtraction	
		FTRC	Floating-point conversion with rounding to integer	
FPU-related CPU instructions	2	LDS	Load into floating-point system register	8
		STS	Store from floating-point system register	
Total:		79		172

Table 2.11 shows the format used in tables 2.12 to 2.19, which list instruction codes, operation, and execution states in order by classification.

**Table 2.11 Instruction Code Format**

Item	Format	Explanation
Instruction	OP . Sz SRC , DEST	OP: Operation code Sz: Size (B: byte, W: word, or L: longword) SRC: Source DEST: Destination Rm: Source register Rn: Destination register imm: Immediate data disp: Displacement* <sup>1</sup>
Instruction code	MSB ↔ LSB	mmmm: Source register nnnn: Destination register 0000: R0 0001: R1 . . . 1111: R15 iiii: Immediate data dddd: Displacement
Operation	→, ←	Direction of transfer
	(xx)	Memory operand
	M/Q/T	Flag bits in the SR
	&	Logical AND of each bit
		Logical OR of each bit
	^	Exclusive OR of each bit
	~	Logical NOT of each bit
	<<n	n-bit left shift
	>>n	n-bit right shift
Execution cycles	—	Value when no wait states are inserted* <sup>2</sup>
T bit	—	Value of T bit after instruction is executed. An em-dash (—) in the column means no change.

Notes: 1. Depending on the operand size, displacement is scaled ×1, ×2, or ×4. For details, see the *SH-2E Programming Manual*.

2. Instruction execution cycles: The execution cycles shown in the table are minimums. The actual number of cycles may be increased when (1) contention occurs between instruction fetches and data access, or (2) when the destination register of the load instruction (memory → register) and the register used by the next instruction are the same.



**Table 2.12 Data Transfer Instructions**

Instruction	Instruction Code	Operation	Execution Cycles	T Bit
MOV #imm,Rn	1110nnnniiiiiii	#imm → Sign extension → Rn	1	—
MOV.W @(disp,PC),Rn	1001nnnnnddddddd	(disp × 2 + PC) → Sign extension → Rn	1	—
MOV.L @(disp,PC),Rn	1101nnnnnddddddd	(disp × 4 + PC) → Rn	1	—
MOV Rm,Rn	0110nnnnmmmm0011	Rm → Rn	1	—
MOV.B Rm,@Rn	0010nnnnmmmm0000	Rm → (Rn)	1	—
MOV.W Rm,@Rn	0010nnnnmmmm0001	Rm → (Rn)	1	—
MOV.L Rm,@Rn	0010nnnnmmmm0010	Rm → (Rn)	1	—
MOV.B @Rm,Rn	0110nnnnmmmm0000	(Rm) → Sign extension → Rn	1	—
MOV.W @Rm,Rn	0110nnnnmmmm0001	(Rm) → Sign extension → Rn	1	—
MOV.L @Rm,Rn	0110nnnnmmmm0010	(Rm) → Rn	1	—
MOV.B Rm,@-Rn	0010nnnnmmmm0100	Rn-1 → Rn, Rm → (Rn)	1	—
MOV.W Rm,@-Rn	0010nnnnmmmm0101	Rn-2 → Rn, Rm → (Rn)	1	—
MOV.L Rm,@-Rn	0010nnnnmmmm0110	Rn-4 → Rn, Rm → (Rn)	1	—
MOV.B @Rm+,Rn	0110nnnnmmmm0100	(Rm) → Sign extension → Rn, Rm + 1 → Rm	1	—
MOV.W @Rm+,Rn	0110nnnnmmmm0101	(Rm) → Sign extension → Rn, Rm + 2 → Rm	1	—
MOV.L @Rm+,Rn	0110nnnnmmmm0110	(Rm) → Rn, Rm + 4 → Rm	1	—
MOV.B R0,@(disp,Rn)	1000000nnnnndddd	R0 → (disp + Rn)	1	—
MOV.W R0,@(disp,Rn)	1000001nnnnndddd	R0 → (disp × 2 + Rn)	1	—
MOV.L Rm,@(disp,Rn)	0001nnnnmmmmddddd	Rm → (disp × 4 + Rn)	1	—
MOV.B @(disp,Rm),R0	10000100mmmmddddd	(disp + Rm) → Sign extension → R0	1	—
MOV.W @(disp,Rm),R0	10000101mmmmddddd	(disp × 2 + Rm) → Sign extension → R0	1	—
MOV.L @(disp,Rm),Rn	0101nnnnmmmmddddd	(disp × 4 + Rm) → Rn	1	—
MOV.B Rm,@(R0,Rn)	0000nnnnmmmm0100	Rm → (R0 + Rn)	1	—

**Table 2.12 Data Transfer Instructions (cont)**

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

**Table 2.13 Arithmetic Operation Instructions**

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
ADD	Rm, Rn	0011nnnnmmmm1100	$Rn + Rm \rightarrow Rn$	1	—
ADD	#imm, Rn	0111nnnniiiiiii	$Rn + imm \rightarrow Rn$	1	—
ADDC	Rm, Rn	0011nnnnmmmm1110	$Rn + Rm + T \rightarrow Rn$ , Carry $\rightarrow T$	1	Carry
ADDV	Rm, Rn	0011nnnnmmmm1111	$Rn + Rm \rightarrow Rn$ , Overflow $\rightarrow T$	1	Overflow
CMP/EQ	#imm, R0	10001000iiiiiii	If $R0 = imm$ , $1 \rightarrow T$	1	Comparison result
CMP/EQ	Rm, Rn	0011nnnnmmmm0000	If $Rn = Rm$ , $1 \rightarrow T$	1	Comparison result
CMP/HS	Rm, Rn	0011nnnnmmmm0010	If $Rn = Rm$ with unsigned data, $1 \rightarrow T$	1	Comparison result
CMP/GE	Rm, Rn	0011nnnnmmmm0011	If $Rn = Rm$ with signed data, $1 \rightarrow T$	1	Comparison result
CMP/HI	Rm, Rn	0011nnnnmmmm0110	If $Rn > Rm$ with unsigned data, $1 \rightarrow T$	1	Comparison result
CMP/GT	Rm, Rn	0011nnnnmmmm0111	If $Rn > Rm$ with signed data, $1 \rightarrow T$	1	Comparison result
CMP/PL	Rn	0100nnnn00010101	If $Rn > 0$ , $1 \rightarrow T$	1	Comparison result
CMP/PZ	Rn	0100nnnn00010001	If $Rn = 0$ , $1 \rightarrow T$	1	Comparison result
CMP/STR	Rm, Rn	0010nnnnmmmm1100	If Rn and Rm have an equivalent byte, $1 \rightarrow T$	1	Comparison result
DIV1	Rm, Rn	0011nnnnmmmm0100	Single-step division ( $Rn \div Rm$ )	1	Calculation result
DIV0S	Rm, Rn	0010nnnnmmmm0111	MSB of Rn $\rightarrow Q$ , MSB of Rm $\rightarrow M$ , $M \wedge Q \rightarrow T$	1	Calculation result
DIV0U		0000000000011001	$0 \rightarrow M/Q/T$	1	0

**Table 2.13 Arithmetic Operation Instructions (cont)**

Instruction	Instruction Code	Operation	Execution Cycles	T Bit
DMULS.L Rm, Rn	0011nnnnmmmm1101	Signed operation of $Rn \times Rm \rightarrow MACH, MACL$ $32 \times 32 \rightarrow 64$ bits	2 to 4*	—
DMULU.L Rm, Rn	0011nnnnmmmm0101	Unsigned operation of $Rn \times Rm \rightarrow MACH, MACL$ $32 \times 32 \rightarrow 64$ bits	2 to 4*	—
DT Rn	0100nnnn00010000	$Rn - 1 \rightarrow Rn$ , when $Rn$ is 0, $1 \rightarrow T$ . When $Rn$ is nonzero, $0 \rightarrow T$	1	Comparison result
EXTS.B Rm, Rn	0110nnnnmmmm1110	Byte in Rm is sign-extended $\rightarrow Rn$	1	—
EXTS.W Rm, Rn	0110nnnnmmmm1111	Word in Rm is sign-extended $\rightarrow Rn$	1	—
EXTU.B Rm, Rn	0110nnnnmmmm1100	Byte in Rm is zero-extended $\rightarrow Rn$	1	—
EXTU.W Rm, Rn	0110nnnnmmmm1101	Word in Rm is zero-extended $\rightarrow Rn$	1	—
MAC.L @Rm+, @Rn+	0000nnnnmmmm1111	Signed operation of $(Rn) \times (Rm) + MAC \rightarrow MAC$ $32 \times 32 + 64 \rightarrow 64$ bits	3/(2 to 4)*	—
MAC.W @Rm+, @Rn+	0100nnnnmmmm1111	Signed operation of $(Rn) \times (Rm) + MAC \rightarrow MAC$ $16 \times 16 + 64 \rightarrow 64$ bits	3/(2)*	—
MUL.L Rm, Rn	0000nnnnmmmm0111	$Rn \times Rm \rightarrow MACL$ , $32 \times 32 \rightarrow 32$ bits	2 to 4*	—
MULS.W Rm, Rn	0010nnnnmmmm1111	Signed operation of $Rn \times Rm \rightarrow MACL$ $16 \times 16 \rightarrow 32$ bits	1 to 3*	—
MULU.W Rm, Rn	0010nnnnmmmm1110	Unsigned operation of $Rn \times Rm \rightarrow MACL$ $16 \times 16 \rightarrow 32$ bits	1 to 3*	—
NEG Rm, Rn	0110nnnnmmmm1011	$0 - Rm \rightarrow Rn$	1	—
NEGC Rm, Rn	0110nnnnmmmm1010	$0 - Rm - T \rightarrow Rn$ , Borrow $\rightarrow T$	1	Borrow

**Table 2.13 Arithmetic Operation Instructions (cont)**

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
SUB	Rm, Rn	0011nnnnmmmm1000	$Rn - Rm \rightarrow Rn$	1	—
SUBC	Rm, Rn	0011nnnnmmmm1010	$Rn - Rm - T \rightarrow Rn$ , Borrow $\rightarrow T$	1	Borrow
SUBV	Rm, Rn	0011nnnnmmmm1011	$Rn - Rm \rightarrow Rn$ , Underflow $\rightarrow T$	1	Overflow

Note: \*The normal minimum number of execution cycles. (The number in parentheses is the number of cycles when there is contention with following instructions.)

**Table 2.14 Logic Operation Instructions**

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

**Table 2.15 Shift Instructions**

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
ROTL	Rn	0100nnnn00000100	$T \leftarrow Rn \leftarrow \text{MSB}$	1	MSB
ROTR	Rn	0100nnnn00000101	$\text{LSB} \rightarrow Rn \rightarrow T$	1	LSB
ROTCL	Rn	0100nnnn00100100	$T \leftarrow Rn \leftarrow T$	1	MSB
ROTCR	Rn	0100nnnn00100101	$T \rightarrow Rn \rightarrow T$	1	LSB
SHAL	Rn	0100nnnn00100000	$T \leftarrow Rn \leftarrow 0$	1	MSB
SHAR	Rn	0100nnnn00100001	$\text{MSB} \rightarrow Rn \rightarrow T$	1	LSB
SHLL	Rn	0100nnnn00000000	$T \leftarrow Rn \leftarrow 0$	1	MSB
SHLR	Rn	0100nnnn00000001	$0 \rightarrow Rn \rightarrow T$	1	LSB
SHLL2	Rn	0100nnnn00001000	$Rn \ll 2 \rightarrow Rn$	1	—
SHLR2	Rn	0100nnnn00001001	$Rn \gg 2 \rightarrow Rn$	1	—
SHLL8	Rn	0100nnnn00011000	$Rn \ll 8 \rightarrow Rn$	1	—
SHLR8	Rn	0100nnnn00011001	$Rn \gg 8 \rightarrow Rn$	1	—
SHLL16	Rn	0100nnnn00101000	$Rn \ll 16 \rightarrow Rn$	1	—
SHLR16	Rn	0100nnnn00101001	$Rn \gg 16 \rightarrow Rn$	1	—

**Table 2.16 Branch Instructions**

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
BF	label	10001011ddddddddd	If T = 0, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ ; if T = 1, nop	3/1*	—
BF/S	label	10001111ddddddddd	Delayed branch, if T = 0, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ ; if T = 1, nop	3/1*	—
BT	label	10001001ddddddddd	If T = 1, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ ; if T = 0, nop	3/1*	—
BT/S	label	10001101ddddddddd	Delayed branch, if T = 1, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$ ; if T = 0, nop	2/1*	—
BRA	label	1010ddddddddd	Delayed branch, $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$	2	—
BRAF	Rm	0000mmmm00100011	Delayed branch, $\text{Rm} + \text{PC} \rightarrow \text{PC}$	2	—
BSR	label	1011ddddddddd	Delayed branch, $\text{PC} \rightarrow \text{PR}$ , $\text{disp} \times 2 + \text{PC} \rightarrow \text{PC}$	2	—
BSRF	Rm	0000mmmm00000011	Delayed branch, $\text{PC} \rightarrow \text{PR}$ , $\text{Rm} + \text{PC} \rightarrow \text{PC}$	2	—
JMP	@Rm	0100mmmm00101011	Delayed branch, $\text{Rm} \rightarrow \text{PC}$	2	—
JSR	@Rm	0100mmmm00001011	Delayed branch, $\text{PC} \rightarrow \text{PR}$ , $\text{Rm} \rightarrow \text{PC}$	2	—
RTS		0000000000001011	Delayed branch, $\text{PR} \rightarrow \text{PC}$	2	—

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

**Table 2.17 System Control Instructions**

Instruction	Instruction Code	Operation	Execution Cycles	T Bit
CLRT	0000000000001000	0 → T	1	0
CLRMAC	000000000101000	0 → MACH, MACL	1	—
LDC Rm, SR	0100mmmm00001110	Rm → SR	1	LSB
LDC Rm, GBR	0100mmmm00011110	Rm → GBR	1	—
LDC Rm, VBR	0100mmmm00101110	Rm → VBR	1	—
LDC.L @Rm+, SR	0100mmmm00000111	(Rm) → SR, Rm + 4 → Rm	3	LSB
LDC.L @Rm+, GBR	0100mmmm00010111	(Rm) → GBR, Rm + 4 → Rm	3	—
LDC.L @Rm+, VBR	0100mmmm00100111	(Rm) → VBR, Rm + 4 → Rm	3	—
LDS Rm, MACH	0100mmmm00001010	Rm → MACH	1	—
LDS Rm, MACL	0100mmmm00011010	Rm → MACL	1	—
LDS Rm, PR	0100mmmm00101010	Rm → PR	1	—
LDS.L @Rm+, MACH	0100mmmm00000110	(Rm) → MACH, Rm + 4 → Rm	1	—
LDS.L @Rm+, MACL	0100mmmm00010110	(Rm) → MACL, Rm + 4 → Rm	1	—
LDS.L @Rm+, PR	0100mmmm00100110	(Rm) → PR, Rm + 4 → Rm	1	—
NOP	000000000001001	No operation	1	—
RTE	000000000101011	Delayed branch, stack area → PC/SR	4	—
SETT	000000000011000	1 → T	1	1
SLEEP	000000000011011	Sleep	3*	—
STC SR, Rn	0000nnnn00000010	SR → Rn	1	—
STC GBR, Rn	0000nnnn00010010	GBR → Rn	1	—
STC VBR, Rn	0000nnnn00100010	VBR → Rn	1	—
STC.L SR, @-Rn	0100nnnn00000011	Rn - 4 → Rn, SR → (Rn)	2	—
STC.L GBR, @-Rn	0100nnnn00010011	Rn - 4 → Rn, GBR → (Rn)	2	—
STC.L VBR, @-Rn	0100nnnn00100011	Rn - 4 → Rn, BR → (Rn)	2	—
STS MACH, Rn	0000nnnn00001010	MACH → Rn	1	—
STS MACL, Rn	0000nnnn00011010	MACL → Rn	1	—
STS PR, Rn	0000nnnn00101010	PR → Rn	1	—



**Table 2.17 System Control Instructions (cont)**

<b>Instruction</b>	<b>Instruction Code</b>	<b>Operation</b>	<b>Execution Cycles</b>	<b>T Bit</b>
STS.L MACH,@-Rn	0100nnnn00000010	Rn - 4 → Rn, MACH → (Rn)	1	—
STS.L MACL,@-Rn	0100nnnn00010010	Rn - 4 → Rn, MACL → (Rn)	1	—
STS.L PR,@-Rn	0100nnnn00100010	Rn - 4 → Rn, PR → (Rn)	1	—
TRAPA #imm	11000011iiiiiii	PC/SR → stack area, (imm × 4 + VBR) → PC	8	—

Note: \* The number of execution cycles before the chip enters sleep mode: The execution cycles shown in the table are minimums. The actual number of cycles may be increased when (1) contention occurs between instruction fetches and data access, or (2) when the destination register of the load instruction (memory → register) and the register used by the next instruction are the same.

**Table 2.18 Floating-Point Instructions**

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
FABS	FRn	1111nnnn01011101	FRn  → FRn	1	—
FADD	FRm, FRn	1111nnnnmmmm0000	FRn + FRm → FRn	1	—
FCMP/EQ	FRm, FRn	1111nnnnmmmm0100	(FRn = FRm)? 1:0 → T	1	Comparison result
FCMP/GT	FRm, FRn	1111nnnnmmmm0101	(FRn > FRm)? 1:0 → T	1	Comparison result
FDIV	FRm, FRn	1111nnnnmmmm0011	FRn/FRm → FRn	13	—
FLDI0	FRn	1111nnnn10001101	0x00000000 → FRn	1	—
FLDI1	FRn	1111nnnn10011101	0x3F800000 → FRn	1	—
FLDS	FRm, FPUL	1111mmmm00011101	FRm → FPUL	1	—
FLOAT	FPUL, FRn	1111nnnn00101101	(float) FPUL → FRn	1	—
FMAC	FR0, FRm, FRn	1111nnnnmmmm1110	FR0 × FRm + FRn → FRn	1	—
FMOV	FRm, FRn	1111nnnnmmmm1100	FRm → FRn	1	—
FMOV.S	@(R0, Rm), FRn	1111nnnnmmmm0110	(R0 + Rm) → FRn	1	—
FMOV.S	@Rm+, FRn	1111nnnnmmmm1001	(Rm) → FRn, Rm+ = 4	1	—
FMOV.S	@Rm, FRn	1111nnnnmmmm1000	(Rm) → FRn	1	—
FMOV.S	FRm, @(R0, Rn)	1111nnnnmmmm0111	FRm → (R0 + Rn)	1	—
FMOV.S	FRm, @-Rn	1111nnnnmmmm1011	Rn- = 4, FRm → (Rn)	1	—
FMOV.S	FRm, @Rn	1111nnnnmmmm1010	FRm → (Rn)	1	—
FMUL	FRm, FRn	1111nnnnmmmm0010	FRn × FRm → FRn	1	—
FNEG	FRn	1111nnnn01001101	-FRn → FRn	1	—
FSTS	FPUL, FRn	1111nnnn00001101	FPUL → FRn	1	—
FSUB	FRm, FRn	1111nnnnmmmm0001	FRn - FRm → FRn	1	—
FTRC	FRm, FPUL	1111mmmm00111101	(long) FRm → FPUL	1	—

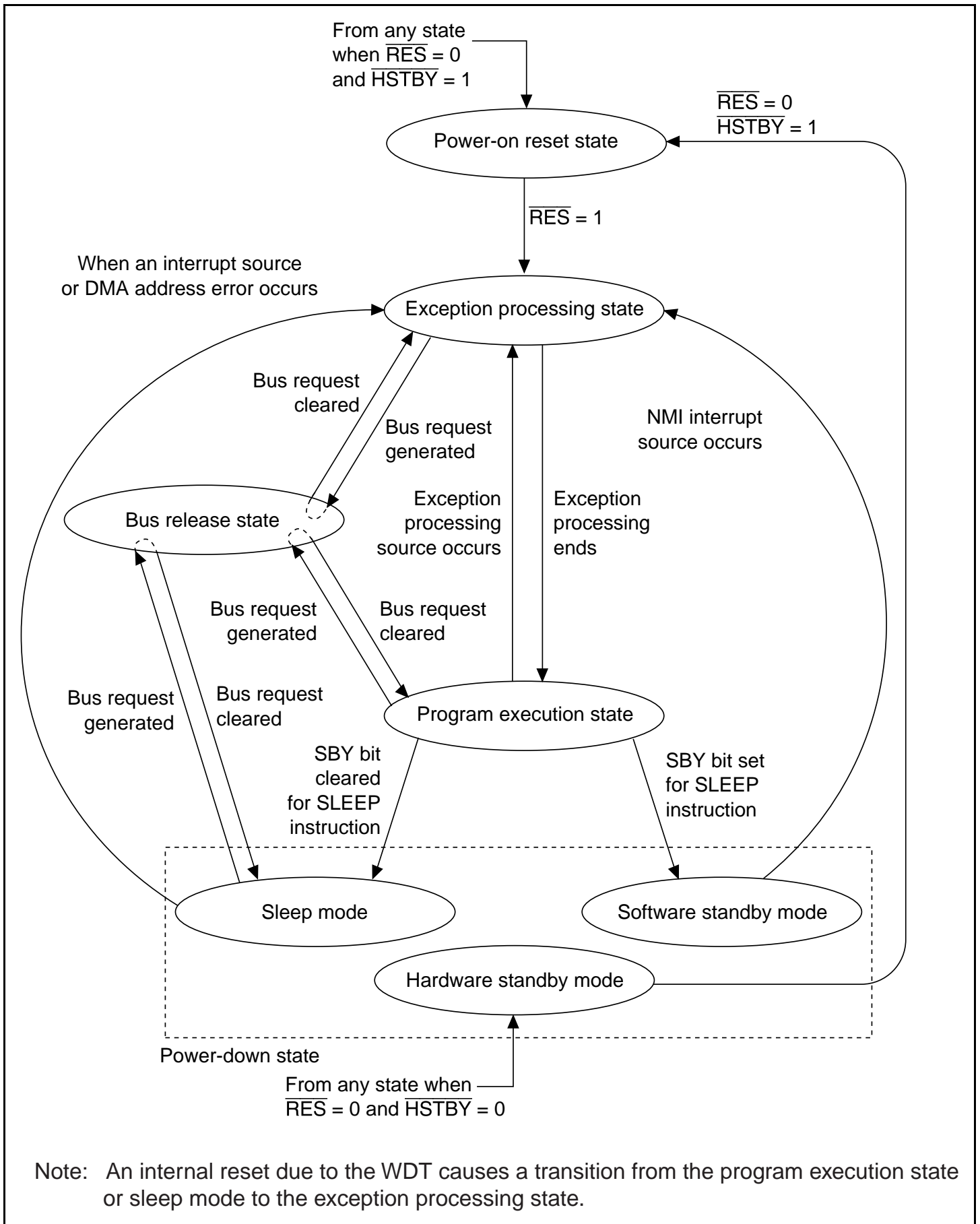
**Table 2.19 FPU-Related CPU Instructions**

Instruction		Instruction Code	Operation	Execution Cycles	T Bit
LDS	Rm, FPSCR	0100mmmm01101010	Rm → FPSCR	1	—
LDS	Rm, FPUL	0100mmmm01011010	Rm → FPUL	1	—
LDS.L	@Rm+, FPSCR	0100mmmm01100110	@Rm → FPSCR, Rm+ = 4	1	—
LDS.L	@Rm+, FPUL	0100mmmm01010110	@Rm → FPUL, Rm+ = 4	1	—
STS	FPSCR, Rn	0000nnnn01101010	FPSCR → Rn	1	—
STS	FPUL, Rn	0000nnnn01011010	FPUL → Rn	1	—
STS.L	FPSCR, @-Rn	0100nnnn01100010	Rn- = 4, FPSCR → @Rn	1	—
STS.L	FPUL, @-Rn	0100nnnn01010010	Rn- = 4, FPUL → @Rn	1	—

## 2.5 Processing States

### 2.5.1 State Transitions

The CPU has five processing states: power-on reset, exception processing, bus release, program execution and power-down. Figure 2.8 shows the transitions between the states.



**Figure 2.8 Transitions between Processing States**

**Power-On Reset State:** The CPU resets in the reset state. When the  $\overline{\text{HSTBY}}$  pin is driven high and the  $\overline{\text{RES}}$  pin level goes low, the power-on reset state is entered.

**Exception Processing State:** The exception processing state is a transient state that occurs when exception processing 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 processing 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 processing vector table; the CPU then branches to that address and the program starts executing, thereby entering the program execution state.

**Program Execution State:** In the program execution state, the CPU sequentially executes the program.

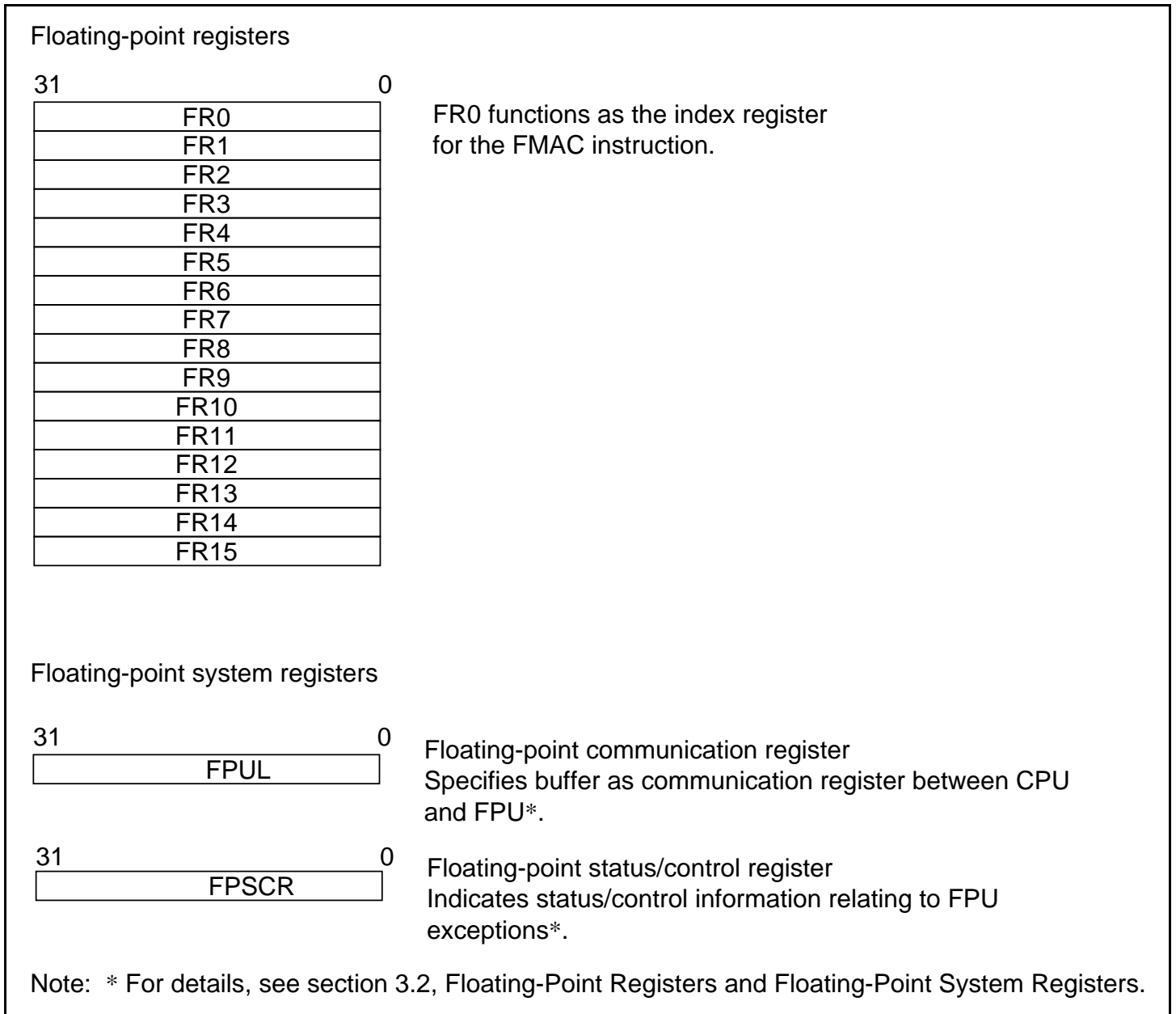
**Power-Down State:** In the power-down state, the CPU operation halts and power consumption declines. The SLEEP instruction places the CPU in the sleep mode or the software standby mode. If the  $\overline{\text{HSTBY}}$  pin is driven low when the  $\overline{\text{RES}}$  pin is low, the CPU will enter the hardware standby mode.

**Bus Release State:** In the bus release state, the CPU releases access rights to the bus to the device that has requested them.

# Section 3 Floating-Point Unit (FPU)

## 3.1 Overview

The SH7058 has an on-chip floating-point unit (FPU), The FPU's register configuration is shown in figure 3.1.



**Figure 3.1 Overview of Register Configuration  
(Floating-Point Registers and Floating-Point System Registers)**

## 3.2 Floating-Point Registers and Floating-Point System Registers

### 3.2.1 Floating-Point Register File

The SH7058 has sixteen 32-bit single-precision floating-point registers. Register specifications are always made as 4 bits. In assembly language, the floating-point registers are specified as FR0, FR1, FR2, and so on. FR0 functions as the index register for the FMAC instruction.

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

Information for transfer between the FPU and the CPU is transferred via the FPUL communication register, which resembles MACL and MACH in the integer unit. The SH7058 is provided with this communication register since the integer and floating-point formats are different. The 32-bit FPUL is a system register, and is accessed by the CPU by means of LDS and STS instructions.

### 3.2.3 Floating-Point Status/Control Register (FPSCR)

The SH7058 has a floating-point status/control register (FPSCR) that functions as a system register accessed by means of LDS and STS instructions (figure 3.2). FPSCR can be written to by a user program. This register is part of the process context, and must be saved when the context is switched. It may also be necessary to save this register when a procedure call is made.

FPSCR is a 32-bit register that controls the storage of detailed information relating to the rounding mode, asymptotic underflow (denormalized numbers), and FPU exceptions. The module stop bit that disables the FPU itself is provided in the module standby control register (MSTCR). For details, see section 25, Power-Down State. After a reset start, the FPU is enabled.

Table 3.1 shows the flags corresponding the five kinds of FPU exception. A sixth flag is also provided as an FPU error flag that indicates an floating-point unit error state not covered by the other five flags.

**Table 3.1 Floating-Point Exception Flags**

Flag	Meaning	Support in SH7058
E	FPU error	—
V	Invalid operation	Yes
Z	Division by zero	Yes
O	Overflow (value not expressed)	—
U	Underflow (value not expressed)	—
I	Inexact (result not expressed)	—

The bits in the cause field indicate the exception cause for the instruction executing at the time. The cause bits are modified by a floating-point instruction. These bits are set to 1 or cleared to 0 according to whether or not an exception state occurred during execution of a single instruction.

The bits in the enable field specify the kinds of exception to be enabled, allowing the flow to be changed to exception processing. If the cause bit corresponding to an enable bit is set by the currently executing instruction, an exception occurs.

The bits in the flag field are used to keep a tally of all exceptions that occur during a series of instructions. Once one of these bits is set by an instruction, it is not reset by a subsequent instruction. The bits in this field can only be reset by the explicit execution of a store operation on FPSCR.



Reserved	Cause field				Enable field			Flag field			RM					
	DN	CE	CV	CZ	CO	CU	CI	EV	EZ	EO		EU	EI	FV	FZ	FO

**DN:** Denormalized bit

In the SH7058 this bit is always set to 1, and the source or destination operand of a denormalized number is 0. This bit cannot be modified even by an LDS instruction.

**CV:** Invalid operation cause bit

When 1: Indicates that an invalid operation exception occurred during execution of the current instruction.

When 0: Indicates that an invalid operation exception has not occurred.

**CZ:** Division-by-zero cause bit

When 1: Indicates that a division-by-zero exception occurred during execution of the current instruction.

When 0: Indicates that a division-by-zero exception has not occurred.

**EV:** Invalid operation exception enable

When 1: Enables invalid operation exception generation.

When 0: An invalid operation exception is not generated, and a qNaN is returned as the result.

**EZ:** Division-by-zero exception enable

When 1: Enables exception generation due to division-by-zero during execution of the current instruction.

When 0: A division-by-zero exception is not generated, and infinity with the sign (+ or -) of the current expression is returned as the result.

**FV:** Invalid operation exception flag bit

When 1: Indicates that an invalid operation exception occurred during instruction execution.

When 0: Indicates that an invalid operation exception has not occurred.

**FZ:** Division-by-zero exception flag bit

When 1: Indicates that a division-by-zero exception occurred during instruction execution.

When 0: Indicates that a division-by-zero exception has not occurred.

**RM:** Rounding bit.

In the SH7058, the value of these bits is always 01, meaning that rounding to zero (RZ mode) is being used. These bits cannot be modified even by an LDS instruction.

In the SH7058, the cause field OUI bits (CE, CO, CU, and CI), enable field OUI bits (EO, EU, and EI), and flag field OUI bits (FO, FU, and FI), and the reserved area, are preset to 0, and cannot be modified even by using an LDS instruction.

**Figure 3.2 Floating-Point Status/Control Register**

## 3.3 Floating-Point Format

### 3.3.1 Floating-Point Format

The SH7058 supports single-precision floating-point operations, and fully complies with the IEEE754 floating-point standard.

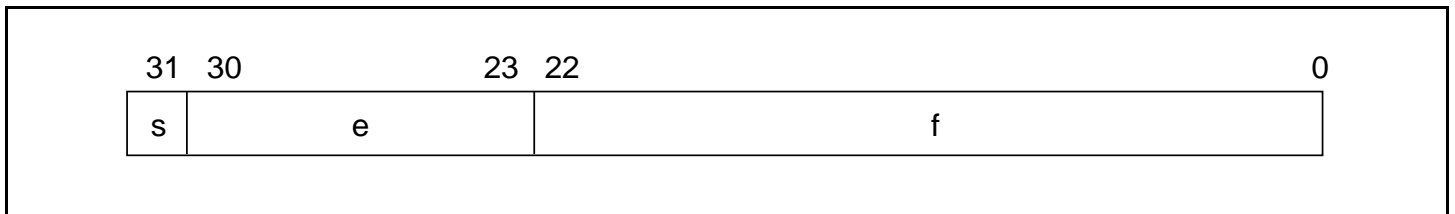
A floating-point number consists of the following three fields:

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

The exponent is expressed in biased form, as follows:

$$e = E + \text{bias}$$

The range of unbiased exponent  $E$  is  $E_{\min} - 1$  to  $E_{\max} + 1$ . The two values  $E_{\min} - 1$  and  $E_{\max} + 1$  are distinguished as follows.  $E_{\min} - 1$  indicates zero (both positive and negative sign) and a denormalized number, and  $E_{\max} + 1$  indicates positive or negative infinity or a non-number (NaN). In a single-precision operation, the bias value is 127,  $E_{\min}$  is  $-126$ , and  $E_{\max}$  is 127.



**Figure 3.3 Floating-Point Number Format**

Floating-point number value  $v$  is determined as follows:

If  $E = E_{\max} + 1$  and  $f \neq 0$ ,  $v$  is a non-number (NaN) irrespective of sign  $s$

If  $E = E_{\max} + 1$  and  $f = 0$ ,  $v = (-1)^s$  (infinity) [positive or negative infinity]

If  $E_{\min} \leq E \leq E_{\max}$ ,  $v = (-1)^s 2^E (1.f)$  [normalized number]

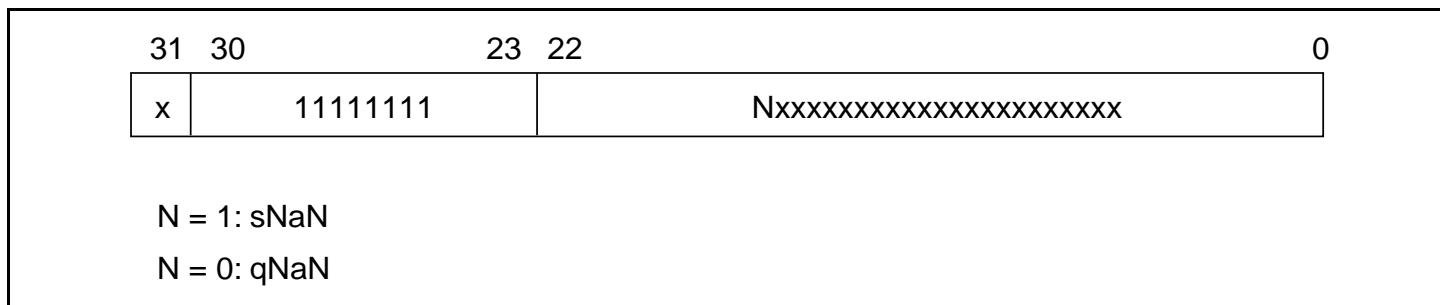
If  $E = E_{\min} - 1$  and  $f \neq 0$ ,  $v = (-1)^s 2^{E_{\min}} (0.f)$  [denormalized number]

If  $E = E_{\min} - 1$  and  $f = 0$ ,  $v = (-1)^s 0$  [positive or negative zero]

### 3.3.2 Non-Numbers (NaN)

With non-number (NaN) representation in a single-precision operation value, at least one of bits 22 to 0 is set. If bit 22 is set, this indicates a signaling NaN (sNaN). If bit 22 is reset, the value is a quiet NaN (qNaN).

The bit pattern of a non-number (NaN) is shown in the figure below. Bit N in the figure is set for a signaling NaN and reset for a quiet NaN. x indicates a don't care bit (with the proviso that at least one of bits 22 to 0 is set). In a non-number (NaN), the sign bit is a don't care bit.



**Figure 3.4 NaN Bit Pattern**

If a non-number (sNaN) is input in an operation that generates a floating-point value:

- When the EV bit in the FPSCR register is reset, the operation result (output) is a quiet NaN (qNaN).
- When the EV bit in the FPSCR register is set, an invalid operation exception will be generated. In this case, the contents of the operation destination register do not change.

If a quiet NaN is input in an operation that generates a floating-point value, and a signaling NaN has not been input in that operation, the output will always be a quiet NaN irrespective of the setting of the EV bit in the FPSCR register. An exception will not be generated in this case.

Refer to the SH-2E Programming Manual for details of floating-point operations when a non-number (NaN) is input.

### 3.3.3 Denormalized Number Values

For a denormalized number floating-point value, the biased exponent is expressed as 0, the fraction as a non-zero value, and the hidden bit as 0. In the SH7058's floating-point unit, a denormalized number (operand source or operation result) is always flushed to 0 in a floating-point operation that generates a value (an operation other than copy).

### 3.3.4 Other Special Values

Floating-point value representations include the seven different kinds of special values shown in table 3.2.

**Table 3.2 Representation of Special Values in Single-Precision Floating-Point Operations Specified by IEEE754 Standard**

<b>Value</b>	<b>Representation</b>
+0.0	0x00000000
-0.0	0x80000000
Denormalized number	As described in section 3.3.3, Denormalized Number Values
+INF	0x7F800000
-INF	0xFF800000
qNaN (quiet NaN)	As described in section 3.3.2, Non-Numbers (NaN)
sNaN (signaling NaN)	As described in section 3.3.2, Non-Numbers (NaN)

## 3.4 Floating-Point Exception Model

### 3.4.1 Enable State Exceptions

Invalid operation and division-by-zero exceptions are both placed in the enable state by setting the enable bit. All exceptions generated by the FPU are mapped as the same exception event. The meaning of a particular exception is determined by software by reading system register FPSCR and analyzing the information held there.

### 3.4.2 Disable State Exceptions

If the EV enable bit is not set, a qNaN will be generated as the result of an invalid operation (except for FCMP and FTRC). If the EZ enable bit is not set, division-by-zero will return infinity with the sign (+ or -) of the current expression. Overflow will generate a finite number which is the largest value that can be expressed by an absolute value in the format, with the correct sign. Underflow will generate zero with the correct sign. If the operation result is inexact, the destination register will store that inexact result.

### 3.4.3 FPU Exception Event and Code

All FPU exceptions have a vector table address offset in address H'00000034 as the same general exception event; that is, an FPU exception.

### 3.4.4 Floating-Point Data Arrangement in Memory

Single-precision floating-point data is located in memory at a 4-byte boundary; that is, it is arranged in the same form as an SH7058 long integer.

### 3.4.5 Arithmetic Operations Involving Special Operands

All arithmetic operations involving special operands (qNaN, sNaN, +INF, -INF, +0, -0) comply with the specifications of the IEEE754 standard. Refer to the SH-2E Programming Manual for details.

## 3.5 Synchronization with CPU

**Synchronization with CPU:** Floating-point instructions and CPU instructions are executed in turn, according to their order in the program, but in some cases operations may not be completed in the program order due to a difference in execution cycles. When a floating-point instruction accesses only FPU resources, there is no need for synchronization with the CPU, and a CPU instruction following an FPU instruction can finish its operation before completion of the FPU operation. Consequently, in an optimized program, it is possible to effectively conceal the execution cycle of a floating-point instruction that requires a long execution cycle, such as a divide instruction. On the other hand, a floating-point instruction that accesses CPU resources, such as a compare instruction, must be synchronized to ensure that the program order is observed.

**Floating-Point Instructions That Require Synchronization:** Load, store, and compare instructions, and instructions that access the FPUL or FPSCR register, must be synchronized because they access CPU resources. Load and store instructions access a general register. Post-increment load and pre-decrement store instructions change the contents of a general register. A compare instruction modifies the T bit. An FPUL or FPSCR access instruction references or changes the contents of the FPUL or FPSCR register. These references and changes must all be synchronized with the CPU.

### 3.6 Usage Notes

Of the arithmetic operations that come up with special operand in this FPU, the following two patterns generate values whose sign is different from that defined in IEEE754 Standard.

(1) FADD FRm, FRn

FRm = -INF (0xFF800000)

FRn = MAX (0x7F7FFFFFFF)

In this case, although the expectation value in IEEE754 is -INF (0xFF800000), the result is +INF (0xFF800000).

(2) FSUB FRm, FRn

FRm = +INF (0x7F800000)

FRn = MAX (0x7F7FFFFFFF)

In this case, although the expectation value in IEEE754 is -INF (0xFF800000), the result is +INF (0x7F800000).

# Section 4 Operating Modes

## 4.1 Operating Mode Selection

The SH7058 has five operating modes that are selected by pins MD2 to MD0 and FWE. The mode setting pins should not be changed during operation of the SH7058, and only the setting combinations shown in table 4.1 should be used.

The  $PV_{cc1}$  power supply voltage must be within the range shown in table 4.1.

**Table 4.1 Operating Mode Selection**

Operating Mode No.	Pin Settings				Mode Name	On-Chip ROM	Area 0 Bus	$PV_{cc1}$ Voltage
	FWE	MD2	MD1	MD0			Width	
Mode 0	0	1	0	0	MCU expanded mode	Disabled	8 bits	$3.3\text{ V} \pm 0.3\text{ V}$
Mode 1	0	1	0	1		Enabled	16 bits	
Mode 2	0	1	1	0		Enabled	Set by BCR1	
Mode 3	0	1	1	1	MCU single-chip mode	Enabled	—	$5.0\text{ V} \pm 0.5\text{ V}$
Mode 4	1	1	0	0	Boot mode	Enabled	Set by BCR1	$3.3\text{ V} \pm 0.3\text{ V}$
Mode 5	1	1	0	1			—	$5.0\text{ V} \pm 0.5\text{ V}$
Mode 6	1	1	1	0	User program mode	Enabled	Set by BCR1	$3.3\text{ V} \pm 0.3\text{ V}$
Mode 7	1	1	1	1			—	$5.0\text{ V} \pm 0.5\text{ V}$
Mode 8	1	0	0	0	User boot mode	Enabled	Set by BCR1	$3.3\text{ V} \pm 0.3\text{ V}$
Mode 9	1	0	0	1			—	$5.0\text{ V} \pm 0.5\text{ V}$
—	0/1	0	1	1	Programmer mode	—	—	$3.3\text{ V} \pm 0.3\text{ V}$

There are two MCU operating modes: MCU single-chip mode and MCU expanded mode.

Modes in which the flash memory can be programmed are boot mode, user boot mode and user program mode (the two on-board programming modes) and programmer mode in which programming is performed with an EPROM programmer (a type which supports programming of this device).

For details, see section 23, ROM.





# Section 5 Clock Pulse Generator (CPG)

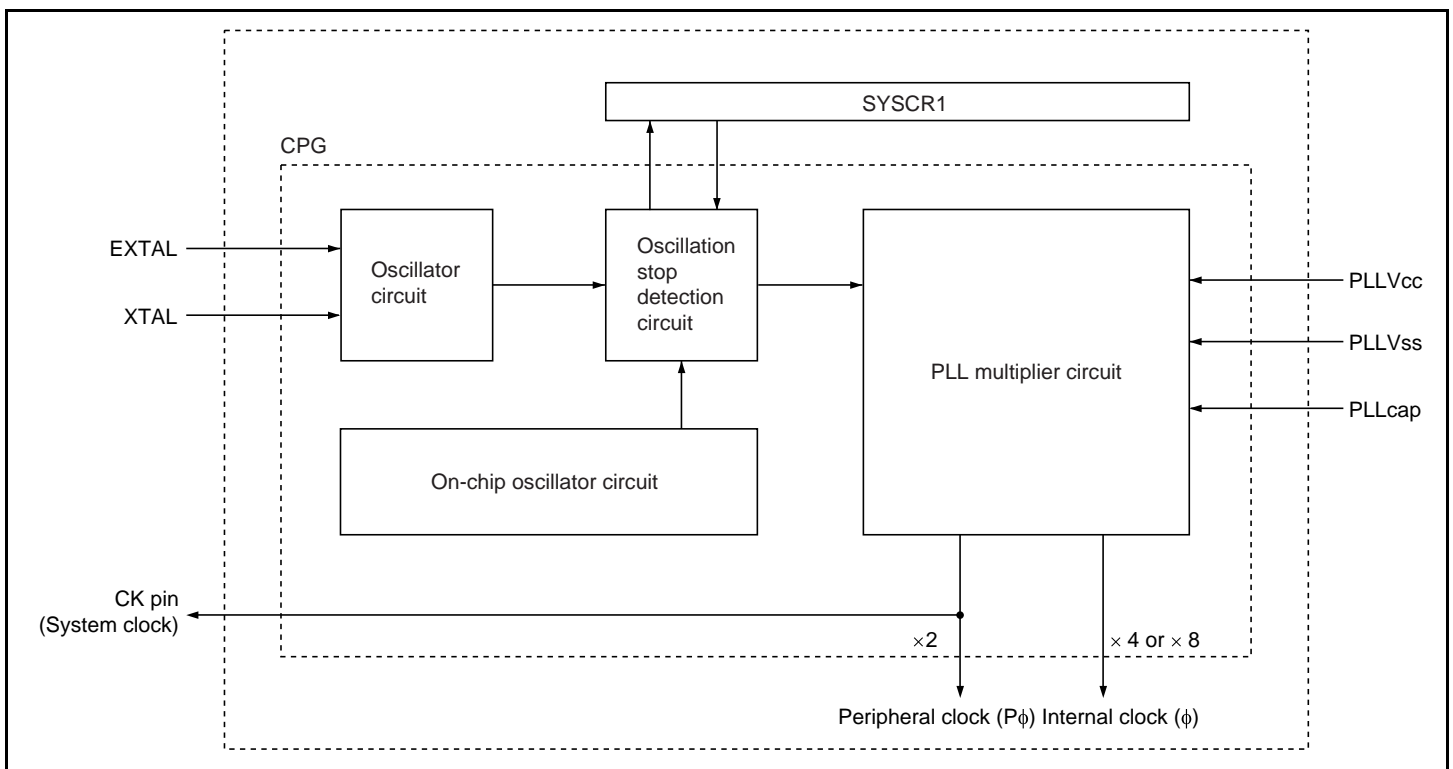
## 5.1 Overview

The clock pulse generator (CPG) supplies clock pulses inside the SH7058 chip and to external devices. The SH7058 CPG consists of an oscillator circuit and a PLL multiplier circuit. There are two methods of generating a clock with the CPG: by connecting a crystal resonator, or by inputting an external clock. The oscillator circuit oscillates at the same frequency as the input clock. Two types of clock signals, internal clock ( $\phi$ ) and peripheral clock ( $P\phi$ ) signals, are supplied and used by the SH7058. The internal clock signal ( $\phi$ ), with frequency either four or eight times the frequency of the clock signal input from the EXTAL pin, is mainly supplied to the bus master modules. The peripheral clock signal ( $P\phi$ ), with frequency two times the frequency of the clock signal input from the EXTAL pin, is mainly supplied to the on-chip peripheral modules. The CK pin outputs the peripheral clock signal ( $P\phi$ ).

The CPG is halted in software standby mode and hardware standby mode.

### 5.1.1 Block Diagram

A block diagram of the clock pulse generator is shown in figure 5.1.



**Figure 5.1 Block Diagram of Clock Pulse Generator**

## 5.1.2 Pin Configuration

The pins relating to the clock pulse generator are shown in table 5.1.

**Table 5.1 CPG Pins**

Pin Name	Abbreviation	I/O	Description
External clock	EXTAL	Input	Crystal oscillator or external clock input
Crystal	XTAL	Input	Crystal oscillator connection
System clock	CK	Output	System clock output
PLL power supply	PLL <sub>V<sub>cc</sub></sub>	Input	PLL multiplier circuit power supply
PLL ground	PLL <sub>V<sub>ss</sub></sub>	Input	PLL multiplier circuit ground
PLL capacitance	PLLCAP	Input	PLL multiplier circuit oscillation external capacitance pin

## 5.1.3 Related Register

The register relating to the clock pulse generator is shown in table 5.2.

**Table 5.2 CPG Register**

Name	Abbreviation	R/W	Initial Value	Address		Access Size
				Write	Read	
System control register 2	SYSCR2	R/W	H'01	H'FFFFFF70A* <sup>1</sup>	H'FFFFFF70B* <sup>2</sup>	8, 16

Notes: 1. Data should be written in words; data cannot be written in bytes or longwords.  
2. Data should be read in bytes; correct data cannot be read in words or longwords.

## 5.2 Frequency Ranges and Clock Selection

### 5.2.1 Frequency Ranges

The input frequency and operating frequency ranges are shown in table 5.3.

**Table 5.3 Input Frequency and Operating Frequency**

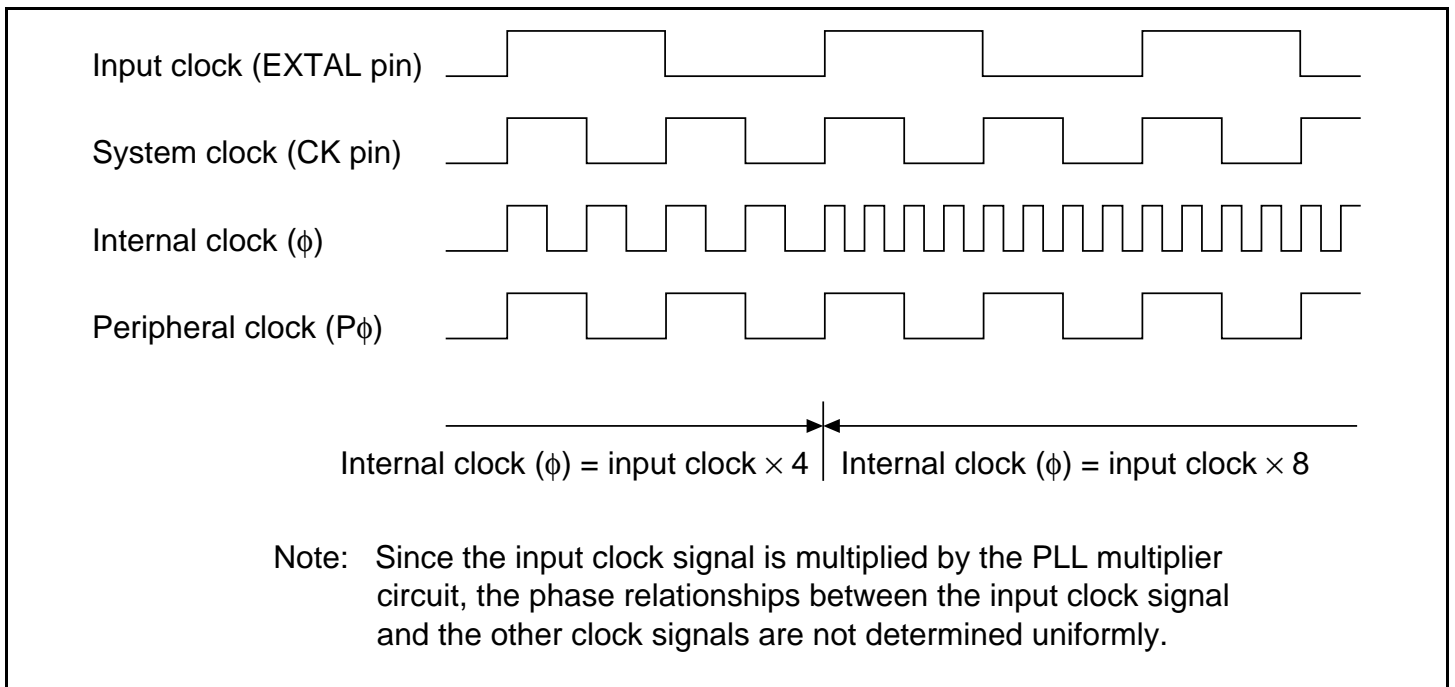
Input Frequency Range (MHz)	PLL Multiplication Factor	Internal Clock ( $\phi$ ) Frequency Range (MHz)	Peripheral Clock ( $P\phi$ ) Frequency Range (MHz)	System Clock Frequency Range (MHz)
5 to 10	$\times 4$	20 to 40	10 to 20	10 to 20
	$\times 8$	40 to 80		

Note: Crystal oscillator and external clock input

Two types of clock signals, internal clock ( $\phi$ ) and peripheral clock ( $P\phi$ ) signals, are supplied and used by the SH7058.

The internal clock signal ( $\phi$ ), with frequency either four or eight times the frequency of the clock signal input from the EXTAL pin, is mainly supplied to the bus master modules such as CPU, FPU, and DMAC.

The peripheral clock signal ( $P\phi$ ), with frequency two times the frequency of the clock signal input from the EXTAL pin, is mainly supplied to the on-chip peripheral modules. The CK pin outputs the peripheral clock signal ( $P\phi$ ) signal as the system clock signal.



**Figure 5.2 Frequencies and Phases of Clock Signals**

### 5.2.2 Clock Selection

The frequency of the internal clock signal ( $\phi$ ) can be either four or eight times the frequency of the input clock signal (EXTAL pin), and the frequency can be selected via the CKSEL bit in system control register 2 (SYSCR2).

## System Control Register 2 (SYSCR2)

Bit:	7	6	5	4	3	2	1	0
	CKSEL	–	–	–	MSTOP3	MSTOP2	MSTOP1	MSTOP0
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

System control register 2 (SYSCR2) is an 8-bit readable/writable register that selects the internal clock signal ( $\phi$ ) and controls the standby state of the AUD, H-UDI, FPU, and UBC.

SYSCR2 is initialized to H'01 by a power-on reset.

Bit 7—Internal Clock ( $\phi$ ) Select (CKSEL): Selects the frequency of the internal clock signal ( $\phi$ ).

When writing to this bit, follow the procedure below.

1. Halt the DMAC and AUD (do not allow a bus cycle to be generated for the DMAC or AUD immediately after writing to this register). However, the AUD does not need to be halted during AUD branch trace.
2. Disable interrupts.
3. Place four NOP instructions after writing to this bit.

### Bit 7: CKSEL Description

0	Frequency of internal clock signal ( $\phi$ ) is four times the input clock frequency (Initial value)
1	Frequency of internal clock signal ( $\phi$ ) is eight times the input clock frequency

For bits 6 to 0, see section 25, Power-Down State.

### 5.2.3 Notes on Register Access

The method of writing to system control register 2 (SYSCR2) is different from that of ordinary registers to prevent inadvertent rewriting.

Be certain to use a word transfer instruction when writing data to SYSCR2. Data cannot be written by a byte transfer instruction. As shown in figure 5.3, set the upper byte to H'3C and transfer data using the lower byte as write data.

Data can be read by the same method as for ordinary registers.

SYSCR2 is allocated to address H'FFFFFF70A. Always use a byte transfer instruction to read data.

## When writing to SYSCR2

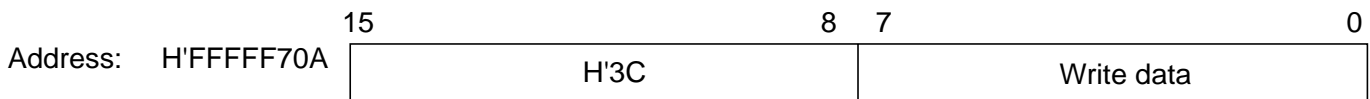


Figure 5.3 Writing to SYSCR2

## 5.3 Clock Source

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

### 5.3.1 Connecting a Crystal Oscillator

**Circuit Configuration:** Figure 5.4 shows an example of connecting a crystal oscillator. Use the damping resistance (Rd) shown in table 5.4. An AT-cut parallel-resonance type crystal oscillator should be used. Load capacitors (CL1, CL2) must be connected as shown in the figure.

The clock pulses generated by the crystal oscillator and internal oscillator are sent to the PLL multiplier circuit, where a multiplied frequency is selected and supplied inside the SH7058 chip and to external devices.

The crystal oscillator manufacturer should be consulted concerning the compatibility between the crystal oscillator and the chip.

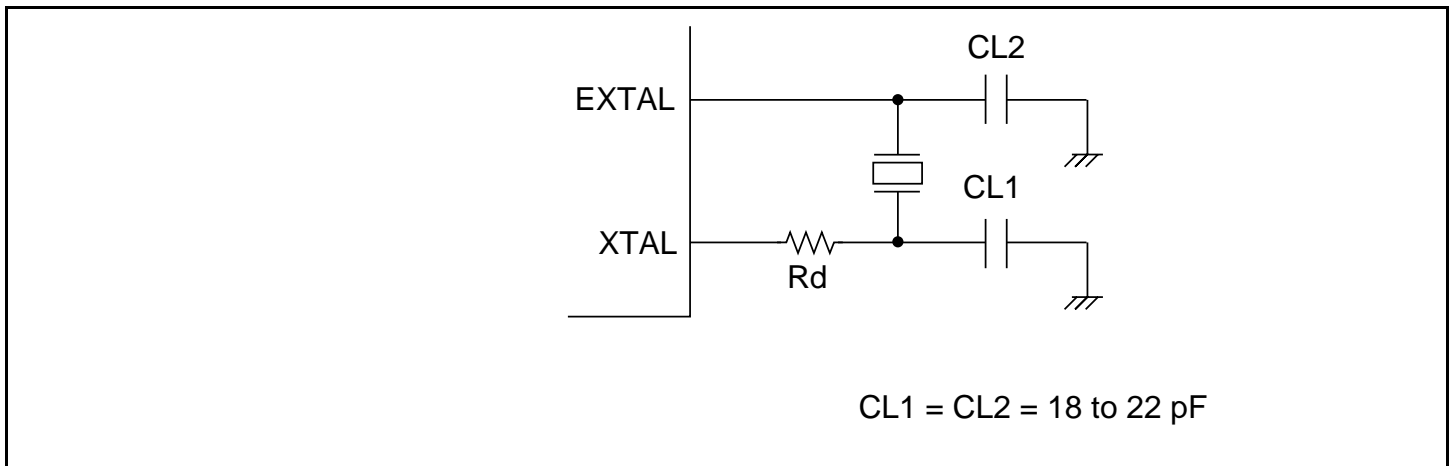
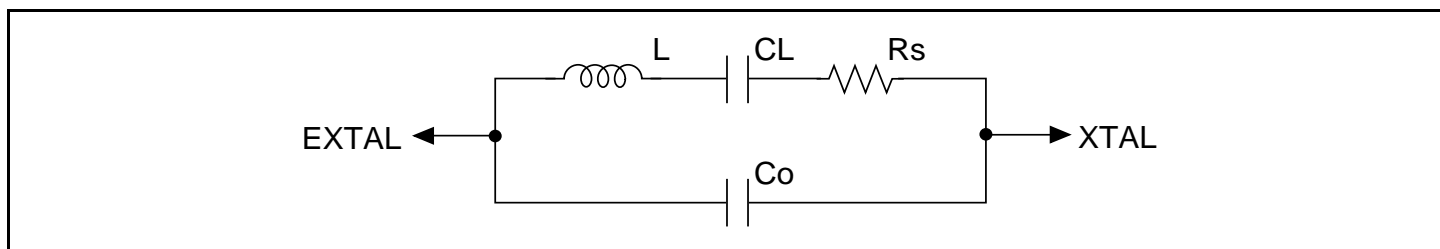


Figure 5.4 Connection of Crystal Oscillator (Example)

Table 5.4 Damping Resistance Values (Recommended Values)

Parameter	Frequency (MHz)	
	5	10
Rd ( $\Omega$ )	500	0

**Crystal Oscillator:** Figure 5.5 shows an equivalent circuit of the crystal oscillator. Use a crystal oscillator with the characteristics listed in table 5.5.



**Figure 5.5 Crystal Oscillator Equivalent Circuit**

**Table 5.5 Crystal Oscillator Parameters (Recommended Values)**

Parameter	Frequency (MHz)	
	5	10
Rs max ( $\Omega$ )	100	50
Co max (pF)	7	7

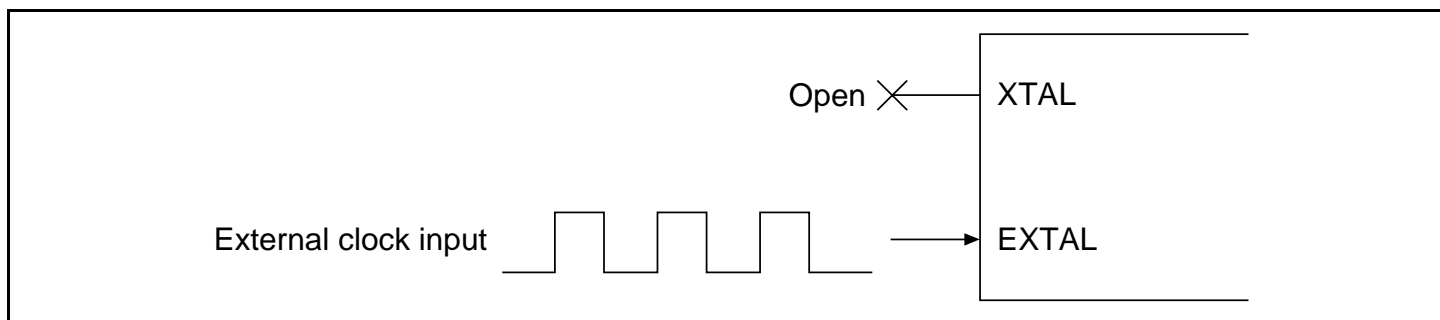
The crystal oscillator manufacturer should be consulted concerning the compatibility between the crystal oscillator and the chip.

### 5.3.2 External Clock Input Method

An example of external clock input connection is shown in figure 5.6.

When the XTAL pin is placed in the open state, the parasitic capacitance should be 10 pF or less.

Even when an external clock is input, provide for a wait of at least the oscillation settling time when powering on or exiting standby mode in order to secure the PLL settling time.



**Figure 5.6 External Clock Input Method (Example)**

## 5.4 Oscillation Stop Detection Function

### 5.4.1 Overview

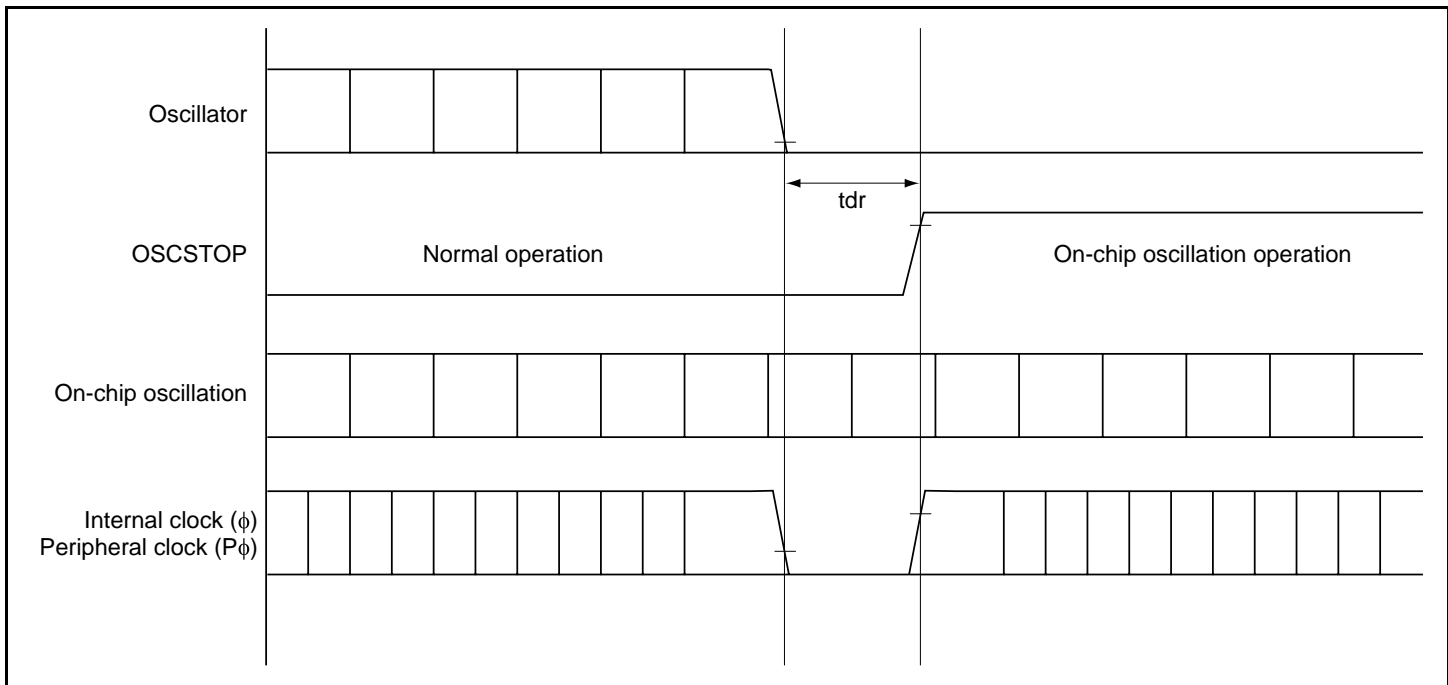
The oscillation stop detection circuit detects errors in the crystal oscillator and sets flags in internal peripheral registers. To make this function effective, the INOSCE bit in SYSCR1 needs to be set to 1 (initial value: 0). If the crystal oscillator performs abnormal operation, such as oscillation stop, for some reason or other, the OSCSTOP bit in SYSCR1 is set to 1 (initial value: 0). In addition, the LSI shuts off clocks from the crystal oscillator and continues to operate by using clocks from the on-chip oscillator circuit. If abnormal operation of the crystal oscillator is detected once, the status is retained until the next reset is released or software standby is released. When this function of oscillation stop detection is made ineffective (the INOSCE bit in SYSCR1 is set to 0), abnormal operations of the crystal oscillator are not detected. Switchover to on-chip oscillation does not take place, either.

### 5.4.2 Settings of Oscillation Stop Detection Function

To make the oscillation stop detection function effective, the INOSCE bit in SYSCR1 needs to be set to 1 (initial value: 0). The INOSCE bit in SYSCR1 is always cleared to 0 after reset or software standby is released. To make this function effective, make sure to set the INOSCE bit to 1 after reset or software standby is released.

While the oscillation stop detection function is made effective, if the clocks from the crystal oscillator are not supplied for a certain period (see table 5.6), the status is determined as abnormal operation of the crystal oscillator and the LSI starts to use the clocks from the on-chip oscillator circuit (see table 5.7). In this case, the LSI sets the OSCSTOP bit in SYSCR1 to 1 (initial value: 0). Once switchover to on-chip oscillation takes place, operation by on-chip oscillation continues even if the crystal oscillator performs normally afterwards. To use the clocks from the crystal oscillator, perform reset start again.





**Figure 5.7 Oscillation Stop Detection Timing**

**Table 5.6 Abnormal Operation Detection with Oscillation Stop Detection Circuit**

	min	typ	max	Precaution
Detection time (tdr)	—	—	1.0 (ms)	As the capability, it ranges from about 30 to 20 $\mu$ s.

**Table 5.7 Frequencies of On-Chip Oscillator Circuit (Internal Clock  $\phi$  Frequency)**

	min	typ	max	Precaution
Internal clock $\phi$ frequency (for multiplication by 8)	10M(Hz)	30M(Hz)	40M(Hz)	On-chip oscillation frequency differs depending on temperature and operation voltage.
Internal clock $\phi$ frequency (for multiplication by 4)	5M(Hz)	15M(Hz)	20M(Hz)	On-chip oscillation frequency differs depending on temperature and operation voltage.

### 5.4.3 Related Register

Register bits relating to the oscillation stop detection function are mapped to bits 7 and 6 in the SYSCR1 register.

Bit:	7	6	5	4	3	2	1	0
	OSCSTOP	INOSCE					AUDSRST	RAME
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R	R	R	R/W	R/W

Bit 7: Crystal Oscillator Abnormal Detection (OSCSTOP)

**Table 5.8 Description on OSCSTOP Bit**

Bit 7

OSCSTOP	Description
0	Normal operation of crystal oscillator
1	Detection of abnormal operation of crystal oscillator and supply of clocks from the on-chip oscillator circuit

Bit 6: Oscillation Stop Detection Function Enable (INOSCE)

**Table 5.9 Description on INOSCE Bit**

Bit 6

INOSCE	Description
0	Enables detection function of abnormal operation of the crystal oscillator.
1	Disables detection function of abnormal operation of the crystal oscillator.

Bits 5 to 0: See section 25, Power -Down State.

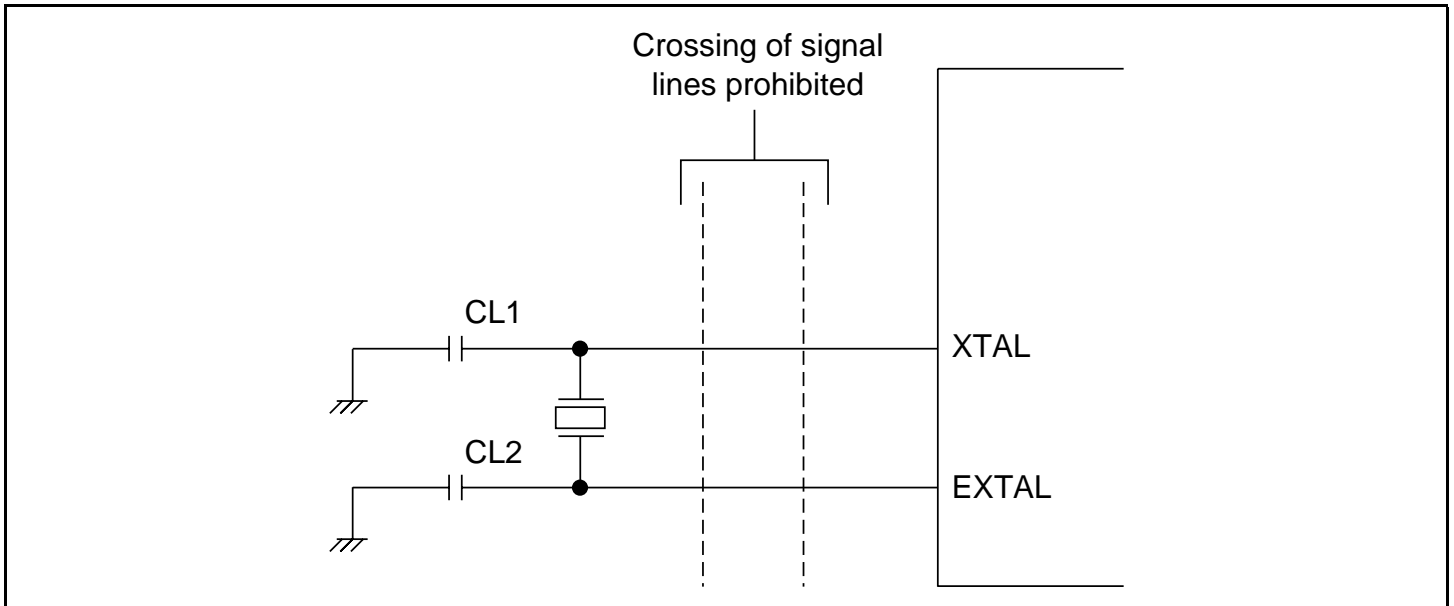
### 5.4.4 Precautions for Performing Oscillation Stop Detection Function

In the status where the on-chip oscillation is used due to abnormal operation of the crystal oscillator, do not disable the oscillation stop detection function (by clearing the INOSCE bit to 0). If the oscillation stop detection function is disabled, this LSI's operation is not guaranteed.

## 5.5 Usage Notes

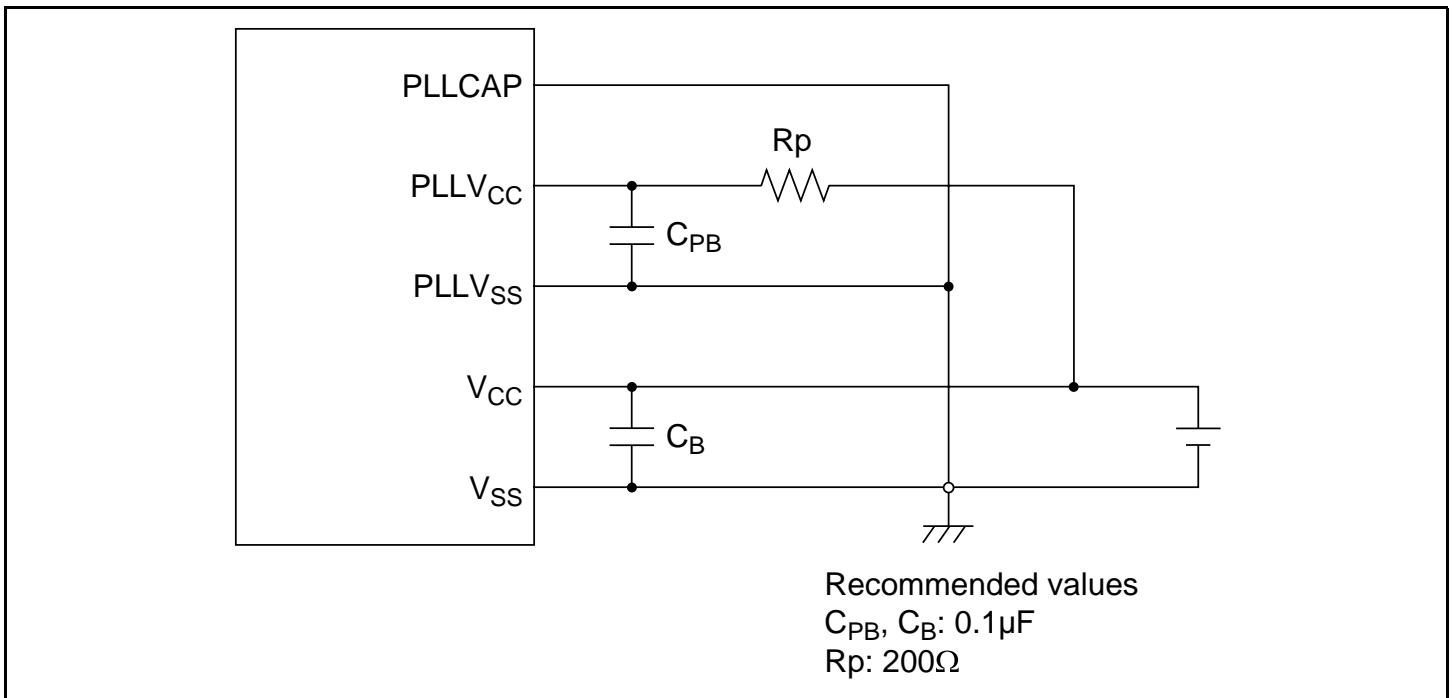
**Notes on Board Design:** Place the crystal oscillator and its load capacitors as close as possible to the XTAL and EXTAL pins.

To prevent induction from interfering with correct oscillation, do not allow any signal lines to cross the XTAL or EXTAL lines (figure 5.8).

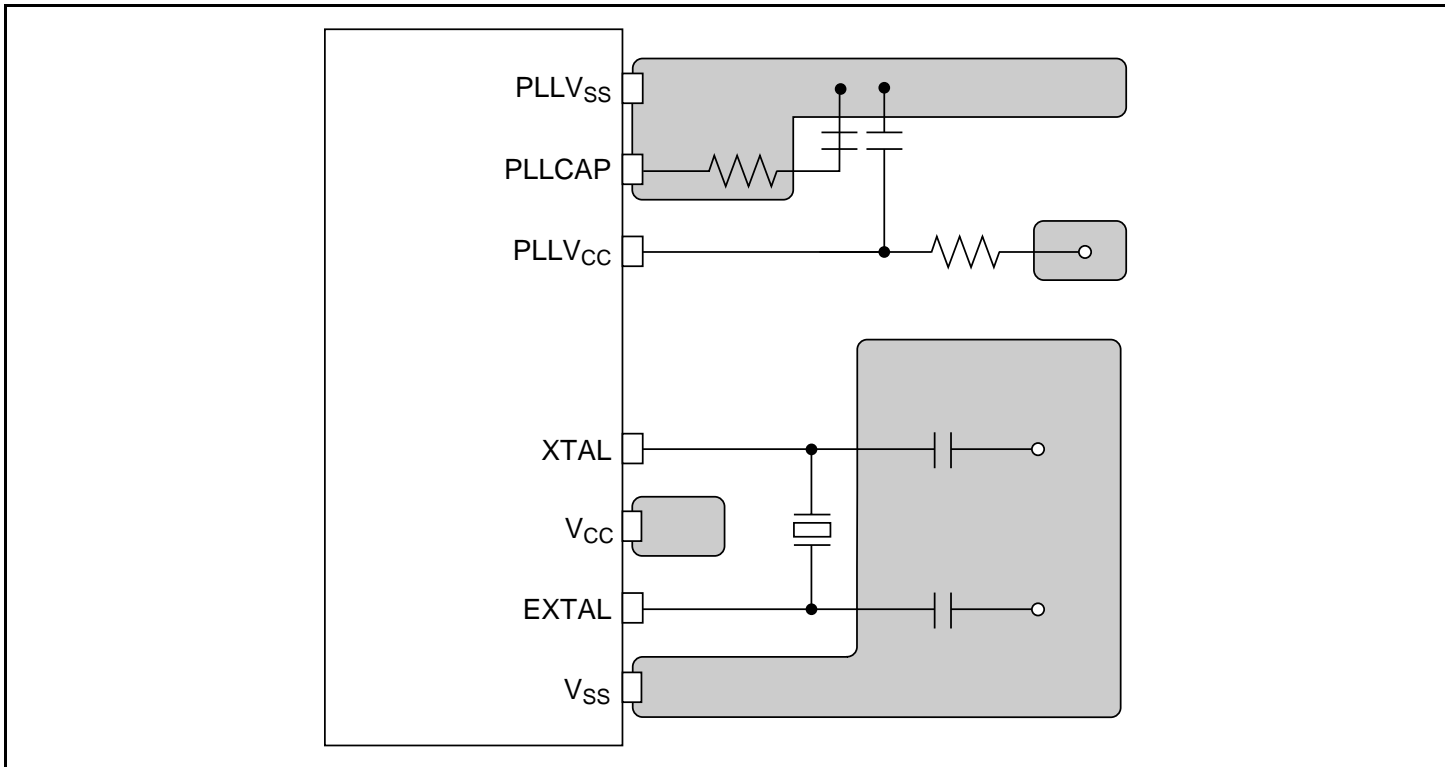


**Figure 5.8 Precautions for Oscillator Circuit System Board Design**

**PLL Oscillation Power Supply:** Separate PLLV<sub>CC</sub> and PLLV<sub>SS</sub> from the other V<sub>CC</sub> and V<sub>SS</sub> lines at the board power supply source, and be sure to insert bypass capacitors C<sub>PB</sub> and C<sub>B</sub> close to the pins.



**Figure 5.9 Points for Caution in PLL Power Supply Connection**



**Figure 5.10 Actual Example of Board Design**



# Section 6 Exception Processing

## 6.1 Overview

### 6.1.1 Types of Exception Processing and Priority

Exception processing is started by four sources: resets, address errors, interrupts and instructions and have the priority shown in table 6.1. When several exception processing sources occur at once, they are processed according to the priority shown.

**Table 6.1 Types of Exception Processing and Priority Order**

Exception Source		Priority	
Reset	Power-on reset		
	Manual reset		
Address error	CPU address error		
	DMAC address error		
Instructions	FPU exception		
Interrupt	NMI		
	User break		
	H-UDI		
	IRQ		
	On-chip peripheral modules:		<ul style="list-style-type: none"> <li>• Direct memory access controller (DMAC)</li> <li>• Advanced timer unit-II (ATU-II)</li> <li>• Compare match timer 0 (CMT0)</li> <li>• Multi trigger A/D0 (MTAD0)</li> <li>• A/D converter channel 0 (A/D0)</li> <li>• Compare match timer 1 (CMT1)</li> <li>• Multi trigger A/D1 (MTAD1)</li> <li>• A/D converter channel 1 (A/D1)</li> <li>• A/D converter channel 2 (A/D2)</li> <li>• Serial communication interface (SCI)</li> <li>• Controller area network 0 (HCAN0)</li> <li>• Watchdog timer (WDT)</li> <li>• Controller area network 1 (HCAN 1)</li> </ul>

**Table 6.1 Types of Exception Processing and Priority Order (cont)**

Exception Source	Priority
Instructions Trap instruction (TRAPA instruction)	High
General illegal instructions (undefined code)	↑ ↓
Illegal slot instructions (undefined code placed directly after a delay branch instruction* <sup>1</sup> or instructions that rewrite the PC* <sup>2</sup> )	Low

- Notes: 1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF.  
 2. Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF.

### 6.1.2 Exception Processing Operations

The exception processing sources are detected and begin processing according to the timing shown in table 6.2.

**Table 6.2 Timing of Exception Source Detection and Start of Exception Processing**

Exception	Source	Timing of Source Detection and Start of Processing
Reset	Power-on reset	Starts when the $\overline{RES}$ pin changes from low to high or when the WDT overflows.
	Manual reset	Starts when the WDT overflows.
Address error		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Interrupts		Detected when instruction is decoded and starts when the previous executing instruction finishes executing.
Instructions	Trap instruction	Starts from the execution of a TRAPA instruction.
	General illegal instructions	Starts from the decoding of undefined code anytime except after a delayed branch instruction (delay slot).
	Illegal slot instructions	Starts from the decoding of undefined code placed in a delayed branch instruction (delay slot) or of instructions that rewrite the PC.
	Floating point instructions	Starts when a floating-point instruction causes an invalid operation exception (IEEE754 specification) or division-by-zero exception.

When exception processing starts, the CPU operates as follows:

1. Exception processing triggered by reset:

The initial values of the program counter (PC) and stack pointer (SP) are fetched from the exception processing vector table (PC and SP are respectively the H'00000000 and H'00000004 addresses for power-on resets and the H'00000008 and H'0000000C addresses for manual resets). See section 6.1.3, Exception Processing Vector Table, for more information. H'00000000 is then written to the vector base register (VBR) and HF (1111) is written to the interrupt mask bits (I3–I0) of the status register (SR). The program begins running from the PC address fetched from the exception processing vector table.

2. Exception processing triggered by address errors, interrupts and instructions:

SR and PC are saved to the stack indicated by R15. For interrupt exception processing, the interrupt priority level is written to the SR's interrupt mask bits (I3–I0). For address error and instruction exception processing, the I3–I0 bits are not affected. The start address is then fetched from the exception processing vector table and the program begins running from that address.

### 6.1.3 Exception Processing Vector Table

Before exception processing begins running, the exception processing vector table must be set in memory. The exception processing vector table stores the start addresses of exception service routines. (The reset exception processing 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 processing, the start addresses of the exception service routines are fetched from the exception processing vector table, which is indicated by this vector table address.

Table 6.3 shows the vector numbers and vector table address offsets. Table 6.4 shows how vector table addresses are calculated.

**Table 6.3 Exception Processing Vector Table**

Exception Sources		Vector Numbers	Vector Table Address†Offset
Power-on reset	PC	0	H'00000000–H'00000003
	SP	1	H'00000004–H'00000007
Manual reset	PC	2	H'00000008–H'0000000B
	SP	3	H'0000000C–H'0000000F
General illegal instruction		4	H'00000010–H'00000013
(Reserved by system)		5	H'00000014–H'00000017



**Table 6.3 Exception Processing Vector Table (cont)**

Exception Sources		Vector Numbers	Vector Table Address†Offset
Slot illegal instruction		6	H'00000018–H'0000001B
(Reserved by system)		7	H'0000001C–H'0000001F
		8	H'00000020–H'00000023
CPU address error		9	H'00000024–H'00000027
DMAC address error		10	H'00000028–H'0000002B
Interrupts	NMI	11	H'0000002C–H'0000002F
	User break	12	H'00000030–H'00000033
FPU exception		13	H'00000034–H'00000037
H-UDI		14	H'00000038–H'0000003B
(Reserved by system)		16	H'0000003C–H'00000043
		:	:
		31	H'0000007C–H'0000007F
Trap instruction (user vector)		32	H'00000080–H'00000083
		:	:
		63	H'000000FC–H'000000FF
Interrupts	IRQ0	64	H'00000100–H'00000103
	IRQ1	65	H'00000104–H'00000107
	IRQ2	66	H'00000108–H'0000010B
	IRQ3	67	H'0000010C–H'0000010F
	IRQ4	68	H'00000110–H'00000113
	IRQ5	69	H'00000114–H'00000117
	IRQ6	70	H'00000118–H'0000011B
	IRQ7	71	H'0000011C–H'0000011F
On-chip peripheral module*		72	H'00000120–H'00000124
		:	:
		255	H'000003FC–H'000003FF

Note: \* The vector numbers and vector table address offsets for each on-chip peripheral module interrupt are given in table 7.3, Interrupt Exception Processing Vectors and Priorities, in section 7, Interrupt Controller (INTC).

**Table 6.4 Calculating Exception Processing Vector Table Addresses**

<b>Exception Source</b>	<b>Vector Table Address Calculation</b>
Resets	Vector table address = (vector table address offset) = (vector number) × 4
Address errors, interrupts, instructions	Vector table address = VBR + (vector table address offset) = VBR + (vector number) × 4

- Notes:
1. VBR: Vector base register
  2. Vector table address offset: See table 6.3.
  3. Vector number: See table 6.3.

## 6.2 Resets

### 6.2.1 Types of Reset

A reset is the highest-priority exception processing source. There are two kinds of reset, power-on and manual. As shown in table 6.5, the CPU state is initialized in both a power-on reset and a manual reset. On-chip peripheral module registers are also initialized by a power-on reset, but not by a manual reset.

**Table 6.5 Exception Source Detection and Exception Processing Start Timing**

Type	Conditions for Transition to Reset State		Internal States		
	RES	WDT Overflow	CPU/MULT/FPU/INTC	On-Chip Peripheral Modules	PFC, IO Port
Power-on reset	Low	—	Initialized	Initialized	Initialized
	High	Power-on reset	Initialized	Initialized	Not initialized
Manual reset	High	Manual reset	Initialized	Not initialized	Not initialized

### 6.2.2 Power-On Reset

**Power-On Reset by Means of  $\overline{\text{RES}}$  Pin:** When the  $\overline{\text{RES}}$  pin is driven low, the chip enters the power-on reset state. To reliably reset the chip, the  $\overline{\text{RES}}$  pin should be kept at the low level for at least the duration of the oscillation settling time at power-on or when in standby mode (when the clock is halted), or at least  $20 t_{\text{cyc}}$  when the clock is running. In the power-on reset state, the CPU's internal state and all the on-chip peripheral module registers are initialized.

In the power-on reset state, power-on reset exception processing starts when the  $\overline{\text{RES}}$  pin is first driven low for a set period of time 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 processing vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception processing vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3-I0) of the status register (SR) are set to H'F (1111).
4. The values fetched from the exception processing vector table are set in the PC and SP, and the program begins executing.

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

**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 the WDT's TCNT overflows, the chip enters the power-on reset state.

The pin function controller (PFC) registers and I/O port registers are not initialized by the reset signal generated by the WDT (these registers are only initialized by a power-on reset from off-chip).

If reset caused by the input signal at the  $\overline{\text{RES}}$  pin and a reset caused by WDT overflow occur simultaneously, the  $\overline{\text{RES}}$  pin reset has priority, and the WOVF bit in RSTCSR is cleared to 0. When WDT-initiated power-on reset processing is started, the CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception processing vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception processing vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3-I0) of the status register (SR) are set to H'F (1111).
4. The values fetched from the exception processing vector table are set in the PC and SP, and the program begins executing.

### 6.2.3 Manual Reset

When a setting is made for a manual reset to be generated in the WDT's watchdog timer mode, and the WDT's TCNT overflows, the chip enters the power-on reset state.

When WDT-initiated manual reset processing is started, the CPU operates as follows:

1. The initial value (execution start address) of the program counter (PC) is fetched from the exception processing vector table.
2. The initial value of the stack pointer (SP) is fetched from the exception processing vector table.
3. The vector base register (VBR) is cleared to H'00000000 and the interrupt mask bits (I3-I0) of the status register (SR) are set to H'F (1111).
4. The values fetched from the exception processing vector table are set in the PC and SP, and the program begins executing.

When a manual reset is generated, the bus cycle is retained, but if a manual reset occurs while the bus is released or during DMAC burst transfer, manual reset exception processing will be deferred until the CPU acquires the bus. However, if the interval from generation of the manual reset until the end of the bus cycle is equal to or longer than the internal manual reset interval of 512 cycles, the internal manual reset source is ignored instead of being deferred, and manual reset exception processing is not executed.

## 6.3 Address Errors

### 6.3.1 Address Error Sources

Address errors occur when instructions are fetched or data read or written, as shown in table 6.6.

**Table 6.6 Bus Cycles and Address Errors**

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

Note: \* See section 9, Bus State Controller (BSC), for details of the on-chip peripheral module space.

## 6.3.2 Address Error Exception Processing

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

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

## 6.4 Interrupts

### 6.4.1 Interrupt Sources

Table 6.7 shows the sources that start up interrupt exception processing. These are divided into NMI, user breaks, H-UDI, IRQ, and on-chip peripheral modules.

**Table 6.7 Interrupt Sources**

Type	Request Source	Number of Sources
NMI	NMI pin (external input)	1
User break	User break controller	1
H-UDI	High-performance user debug interface	1
IRQ	$\overline{\text{IRQ0}}\text{--}\overline{\text{IRQ7}}$ (external input)	8
On-chip peripheral module	Direct memory access controller (DMAC)	4
	Advanced timer unit (ATU-II)	75
	Compare match timer (CMT)	2
	A/D converter	3
	Serial communication interface (SCI)	20
	Watchdog timer (WDT)	1
	Controller area network (HCAN)	8

Each interrupt source is allocated a different vector number and vector table offset. See table 7.3, Interrupt Exception Processing Vectors and Priorities, in section 7, Interrupt Controller (INTC), for more information on vector numbers and vector table address offsets.

## 6.4.2 Interrupt Priority Level

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

The priority order of interrupts is expressed as priority levels 0–16, with priority 0 the lowest and priority 16 the highest. The NMI interrupt has priority 16 and cannot be masked, so it is always accepted. The user break interrupt and H-UDI interrupt priority level is 15. IRQ interrupts and on-chip peripheral module interrupt priority levels can be set freely using the INTC's interrupt priority registers A through L (IPRA to IPRL) as shown in table 6.8. The priority levels that can be set are 0–15. Level 16 cannot be set. See section 7.3.1, Interrupt Priority Registers A–L (IPRA-IPRL), for details of the interrupt priority registers.

**Table 6.8 Interrupt Priority Order**

Type	Priority Level	Comment
NMI	16	Fixed priority level. Cannot be masked.
User break	15	Fixed priority level.
H-UDI	15	Fixed priority level.
IRQ	0–15	Set with interrupt priority level setting registers A through L (IPRA to IPRL).
On-chip peripheral module	0–15	Set with interrupt priority level setting registers A through L (IPRA to IPRL).

## 6.4.3 Interrupt Exception Processing

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 bits (I3–I0) of the status register (SR).

When an interrupt is accepted, exception processing begins. In interrupt exception processing, the CPU saves SR and the program counter (PC) to the stack. The priority level value of the accepted interrupt is written to SR bits I3–I0. For NMI, however, the priority level is 16, but the value set in I3–I0 is H'F (level 15). Next, the start address of the exception service routine is fetched from the exception processing vector table for the accepted interrupt, that address is jumped to and execution begins. See section 7.4, Interrupt Operation, for further details.

## 6.5 Exceptions Triggered by Instructions

### 6.5.1 Types of Exceptions Triggered by Instructions

Exception processing can be triggered by trap instructions, general illegal instructions, and illegal slot instructions, and floating-point instructions, as shown in table 6.9.

**Table 6.9** Types of Exceptions Triggered by Instructions

Type	Source Instruction	Comment
Trap instructions	TRAPA	
Illegal slot instructions	Undefined code placed immediately after a delayed branch instruction (delay slot) and instructions that rewrite the PC	Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF Instructions that rewrite the PC: JMP, JSR, BRA, BSR, RTS, RTE, BT, BF, TRAPA, BF/S, BT/S, BSRF, BRAF
General illegal instructions	Undefined code anywhere besides in a delay slot	
Floating-point instructions	Instruction causing an invalid operation exception defined in the IEEE754 standard or a division-by-zero exception	FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FNEG, FABS, FTRC

### 6.5.2 Trap Instructions

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

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the start address of the instruction to be executed after the TRAPA instruction.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the vector number specified in the TRAPA instruction. That address is jumped to and the program starts executing. The jump that occurs is not a delayed branch.



### 6.5.3 Illegal Slot Instructions

An instruction placed immediately after a delayed branch instruction is said to be placed in a delay slot. When the instruction placed in the delay slot is undefined code, illegal slot exception processing starts up when that undefined code is decoded. Illegal slot exception processing also starts up when an instruction that rewrites the program counter (PC) is placed in a delay slot. The processing starts when the instruction is decoded. The CPU handles an illegal slot instruction as follows:

1. The status register (SR) is saved to the stack.
2. The program counter (PC) is saved to the stack. The PC value saved is the jump address of the delayed branch instruction immediately before the undefined code or the instruction that rewrites the PC.
3. The exception service routine start address is fetched from the exception processing vector table that corresponds to the exception that occurred. That address is jumped to and the program starts executing. The jump that occurs is not a delayed branch.

### 6.5.4 General Illegal Instructions

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

When the FPU has been stopped by means of the module stop bit, floating-point instructions and FPU-related CPU instructions are treated as illegal instructions.

### 6.5.5 Floating-Point Instructions

When the V or Z bit is set in the enable field of the FPSCR register, an FPU exception occurs. This indicates that a floating-point instruction has caused an invalid operation exception defined in the IEEE754 standard or a division-by-zero exception. Floating-point instructions which can cause an exception are as follows:

FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FNEG,  
FABS, FTRC

An FPU exception occurs only if the corresponding enable bit is set. When the FPU detects an exception source, FPU operation is suspended and the occurrence of the exception is reported to the CPU. When exception processing is started, the CPU saves the SR and PC contents to the stack (the PC value saved is the start address of the instruction following the last instruction executed), and branches to the address stored in VBR + H'00000034.

The exception flag bits in the FPSCR are always updated, regardless of whether or not an FPU exception is accepted, and remain set until the user clears them explicitly with an instruction. FPSCR cause bits change each time an FPU instruction is executed.

Exception events other than those defined in the IEEE754 standard (i.e., underflow, overflow, and inexact exceptions) are detected by the FPU but do not result in the generation of any kind of exception. Neither is an FPU exception generated by a floating-point instruction relating to data transfer, such as FLOAT.

## 6.6 When Exception Sources Are Not Accepted

When an address error or interrupt is generated after a delayed branch instruction or interrupt-disabled instruction, it is sometimes not accepted immediately but stored instead, as shown in table 6.10. When this happens, it will be accepted when an instruction that can accept the exception is decoded.

**Table 6.10 Generation of Exception Sources Immediately after a Delayed Branch Instruction or Interrupt-Disabled Instruction**

Point of Occurrence	Exception Source		
	Bus Error	Interrupt	FPU Exception
Immediately after a delayed branch instruction*1	Not accepted	Not accepted	Not accepted
Immediately after an interrupt-disabled instruction*2	Not accepted*4	Not accepted	Accepted
Immediately after an FPU instruction*3	Not accepted	Not accepted	Accepted

- Notes:
1. Delayed branch instructions: JMP, JSR, BRA, BSR, RTS, RTE, BF/S, BT/S, BSRF, BRAF
  2. Interrupt-disabled instructions: LDC, LDC.L, STC, STC.L, LDS, LDS.L, STS, STS.L
  3. FPU instructions: Table 2.18, Floating-Point Instructions, and table 2.19, FPU-Related CPU Instructions, in section 2.4.1, Instruction Set by Classification.
  4. In the SH-2 a bus error is accepted.

## 6.7 Stack Status after Exception Processing Ends

The status of the stack after exception processing ends is as shown in table 6.11.

**Table 6.11 Stack Status After Exception Processing Ends**

Exception Type	Stack Status
Address error	<p>SP → Address of instruction after executed instruction 32 bits SR 32 bits</p>
Trap instruction	<p>SP → Address of instruction after TRAPA instruction 32 bits SR 32 bits</p>
General illegal instruction	<p>SP → Address of general illegal instruction 32 bits SR 32 bits</p>
Interrupt	<p>SP → Address of instruction after executed instruction 32 bits SR 32 bits</p>
Illegal slot instruction	<p>SP → Jump destination address of delay branch instruction 32 bits SR 32 bits</p>
FPU exception	<p>SP → Address of instruction after FPU exception instruction 32 bits SR 32 bits</p>

## 6.8 Usage Notes

### 6.8.1 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 processing.

### 6.8.2 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 processing.

### 6.8.3 Address Errors Caused by Stacking of Address Error Exception Processing

When the stack pointer is not a multiple of four, an address error will occur during stacking of the exception processing (interrupts, etc.) and address error exception processing will start up as soon as the first exception processing is ended. Address errors will then also occur in the stacking for this address error exception processing. To ensure that address error exception processing 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 processing 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 7 Interrupt Controller (INTC)

## 7.1 Overview

The interrupt controller (INTC) ascertains the priority of interrupt sources and controls interrupt requests to the CPU. The INTC has registers for setting the priority of each interrupt which can be used by the user to order the priorities in which the interrupt requests are processed.

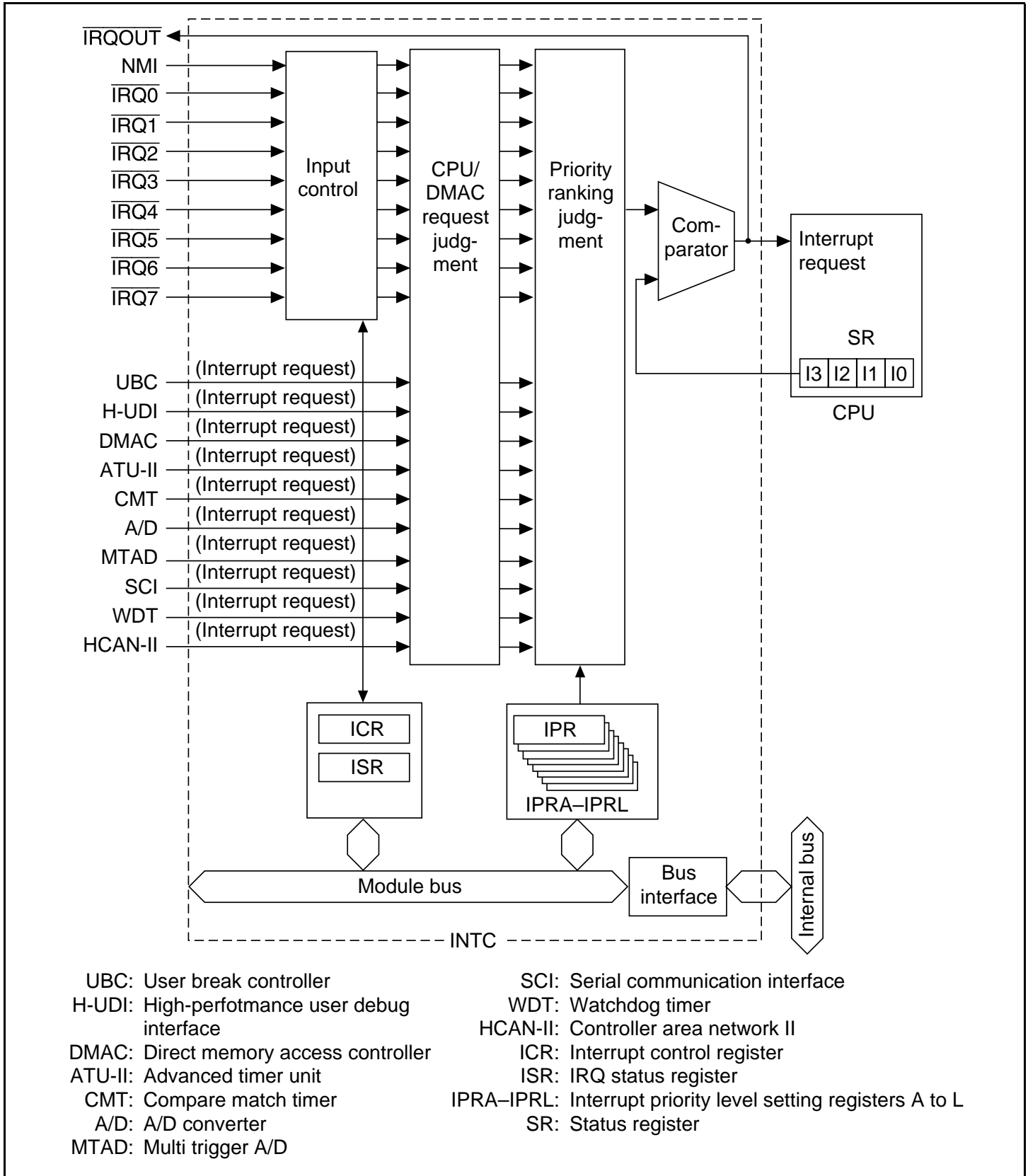
### 7.1.1 Features

The INTC has the following features:

- 16 levels of interrupt priority  
By setting the twelve interrupt-priority level registers, the priorities of IRQ interrupts and on-chip peripheral module interrupts can be set in 16 levels for different request sources.
- NMI noise canceler function  
NMI input level bits indicate the NMI pin status. By reading these bits with the interrupt exception service routine, the pin status can be confirmed, enabling it to be used as a noise canceler.
- Notification of interrupt occurrence can be reported externally ( $\overline{\text{IRQOUT}}$  pin)  
For example, it is possible to request the bus if an external bus master is informed that an on-chip peripheral module interrupt request has occurred when the chip has released the bus.

## 7.1.2 Block Diagram

Figure 7.1 is a block diagram of the INTC.



**Figure 7.1 INTC Block Diagram**

### 7.1.3 Pin Configuration

Table 7.1 shows the INTC pin configuration.

**Table 7.1 Pin Configuration**

Name	Abbreviation	I/O	Function
Non-maskable interrupt input pin	NMI	I	Input of non-maskable interrupt request signal
Interrupt request input pins	$\overline{\text{IRQ0}}\text{--}\overline{\text{IRQ7}}$	I	Input of maskable interrupt request signals
Interrupt request output pin	$\overline{\text{IRQOUT}}$	O	Output of notification signal when an interrupt has occurred

### 7.1.4 Register Configuration

The INTC has the 14 registers shown in table 7.2. These registers set the priority of the interrupts and control external interrupt input signal detection.

**Table 7.2 Register Configuration**

Name	Abbr.	R/W	Initial Value	Address	Access Sizes
Interrupt priority register A	IPRA	R/W	H'0000	H'FFFF ED00	8, 16, 32
Interrupt priority register B	IPRB	R/W	H'0000	H'FFFF ED02	8, 16, 32
Interrupt priority register C	IPRC	R/W	H'0000	H'FFFF ED04	8, 16, 32
Interrupt priority register D	IPRD	R/W	H'0000	H'FFFF ED06	8, 16, 32
Interrupt priority register E	IPRE	R/W	H'0000	H'FFFF ED08	8, 16, 32
Interrupt priority register F	IPRF	R/W	H'0000	H'FFFF ED0A	8, 16, 32
Interrupt priority register G	IPRG	R/W	H'0000	H'FFFF ED0C	8, 16, 32
Interrupt priority register H	IPRH	R/W	H'0000	H'FFFF ED0E	8, 16, 32
Interrupt priority register I	IPRI	R/W	H'0000	H'FFFF ED10	8, 16, 32
Interrupt priority register J	IPRJ	R/W	H'0000	H'FFFF ED12	8, 16, 32
Interrupt priority register K	IPRK	R/W	H'0000	H'FFFF ED14	8, 16, 32
Interrupt priority register L	IPRL	R/W	H'0000	H'FFFF ED16	8, 16, 32
Interrupt control register	ICR	R/W	* <sup>1</sup>	H'FFFF ED18	8, 16, 32
IRQ status register	ISR	R/(W)* <sup>2</sup>	H'0000	H'FFFF ED1A	8, 16, 32

Notes: In register access, four cycles are required for byte access and word access, and eight cycles for longword access.

1. The value when the NMI pin is high is H'8000; when the NMI pin is low, it is H'0000.
2. Only 0 can be written, in order to clear flags.



## 7.2 Interrupt Sources

There are five types of interrupt sources: NMI, user breaks, H-UDI, IRQ, and on-chip peripheral modules. Each interrupt has a priority expressed as a priority level (0 to 16, with 0 the lowest and 16 the highest). Giving an interrupt a priority level of 0 masks it.

### 7.2.1 NMI Interrupts

The NMI interrupt has priority 16 and is always accepted. Input at the NMI pin is detected by edge. Use the NMI edge select bit (NMIE) in the interrupt control register (ICR) to select either the rising or falling edge. NMI interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to level 15.

### 7.2.2 User Break Interrupt

A user break interrupt has a priority of level 15, and occurs when the break condition set in the user break controller (UBC) is satisfied. User break interrupt requests are detected by edge and are held until accepted. User break interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to level 15. For more information about the user break interrupt, see section 8, User Break Controller (UBC).

### 7.2.3 H-UDI Interrupt

A serial debug interface (H-UDI) interrupt has a priority level of 15, and occurs when an H-UDI interrupt instruction is serially input. H-UDI interrupt requests are detected by edge and are held until accepted. H-UDI exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to level 15. For more information about the H-UDI interrupt, see section 19, High-performance User Debug Interface (H-UDI).

### 7.2.4 IRQ Interrupts

IRQ interrupts are requested by input from pins  $\overline{\text{IRQ0}}\text{--}\overline{\text{IRQ7}}$ . Set the IRQ sense select bits (IRQ0S–IRQ7S) of the interrupt control register (ICR) to select low level detection or falling edge detection for each pin. The priority level can be set from 0 to 15 for each pin using interrupt priority registers A and B (IPRA–IPRB).

When IRQ interrupts are set to low level detection, an interrupt request signal is sent to the INTC during the period the IRQ pin is low. Interrupt request signals are not sent to the INTC when the IRQ pin becomes high. Interrupt request levels can be confirmed by reading the IRQ flags (IRQ0F–IRQ7F) of the IRQ status register (ISR).

When IRQ interrupts are set to falling edge detection, interrupt request signals are sent to the INTC upon detecting a change on the IRQ pin from high to low level. IRQ interrupt request detection results are maintained until the interrupt request is accepted. Confirmation that IRQ interrupt requests have been detected is possible by reading the IRQ flags (IRQ0F–IRQ7F) of the IRQ status register (ISR), and by writing a 0 after reading a 1, IRQ interrupt request detection results can be withdrawn.

In IRQ interrupt exception processing, the interrupt mask bits (I3–I0) of the status register (SR) are set to the priority level value of the accepted IRQ interrupt.

## 7.2.5 On-Chip Peripheral Module Interrupts

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

- Direct memory access controller (DMAC)
- Advanced timer unit (ATU-II)
- Compare match timer (CMT)
- A/D converter (A/D)
- Multi trigger A/D (MTAD)
- Serial communication interface (SCI)
- Watchdog timer (WDT)
- Controller area network (HCAN)

A different interrupt vector is assigned to each interrupt source, so the exception service routine does not have to decide which interrupt has occurred. Priority levels between 0 and 15 can be assigned to individual on-chip peripheral modules in interrupt priority registers C–L (IPRC–IPRL).

On-chip peripheral module interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to the priority level value of the on-chip peripheral module interrupt that was accepted.

## 7.2.6 Interrupt Exception Vectors and Priority Rankings

Table 7.3 lists interrupt sources and their vector numbers, vector table address offsets and interrupt priorities.

Each interrupt source is allocated a different vector number and vector table address offset. Vector table addresses are calculated from vector numbers and address offsets. In interrupt exception processing, the exception service routine start address is fetched from the vector table indicated by the vector table address. See table 6.4, Calculating Exception Processing Vector Table Addresses, in section 6, Exception Processing.


IRQ interrupts and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers A–L (IPRA–IPRL). The ranking of interrupt sources for IPRC–IPRL, however, must be the order listed under Priority within IPR Setting Range in table 7.3 and cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts and on-chip peripheral module interrupts. If the same priority level is assigned to two or more interrupt sources and interrupts from those sources occur simultaneously, their priority order is the default priority order indicated at the right in table 7.3.

**Table 7.3 Interrupt Exception Processing Vectors and Priorities**

Interrupt Source	Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR		Default Priority
	Vector No.	Vector Table Address Offset			Setting Range		
NMI	11	H'0000002C to H'0000002F	16	—	—		High
UBC	12	H'00000030 to H'0000003B	15	—	—		
H-UDI	14	H'00000038 to H'0000003B	15	—	—		
IRQ0	64	H'00000100 to H'0000013B	0 to 15 (0)	IPRA (15–12)	—		
IRQ1	65	H'00000104 to H'00000107	0 to 15 (0)	IPRA (11–8)	—		
IRQ2	66	H'00000108 to H'0000010B	0 to 15 (0)	IPRA (7–4)	—		
IRQ3	67	H'0000010C to H'0000010F	0 to 15 (0)	IPRA (3–0)	—		
IRQ4	68	H'00000110 to H'00000113	0 to 15 (0)	I PRB (15–12)	—		
IRQ5	69	H'00000114 to H'00000117	0 to 15 (0)	I PRB (11–8)	—		
IRQ6	70	H'00000118 to H'0000011B	0 to 15 (0)	I PRB (7–4)	—		
IRQ7	71	H'0000011C to H'0000011F	0 to 15 (0)	I PRB (3–0)	—		
DMAC0	DEI0	72	H'00000120 to H'00000123	0 to 15 (0)	I PRC (15–12)	↑ 1	
DMAC1	DEI1	74	H'00000128 to H'0000012B	0 to 15 (0)		↓ 2	
DMAC2	DEI2	76	H'00000130 to H'00000133	0 to 15 (0)	I PRC (11–8)	↑ 1	
DMAC3	DEI3	78	H'00000138 to H'0000013B	0 to 15 (0)		↓ 2	Low



**Table 7.3 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range		Default Priority				
			Vector No.	Vector Table Address Offset									
ATU0	ATU01	ITV1/ ITV2A/ ITV2B	80	H'00000140 to H'00000143	0 to 15 (0)	IPRC (7-4)			High 				
			ATU02	ICI0A			84	H'00000150 to H'00000153		0 to 15 (0)	IPRC (3-0)	↑ 1	
		ICI0B	86	H'00000158 to H'0000015B			↓ 2						
	ATU03	ICI0C	88	H'00000160 to H'00000163	0 to 15 (0)	IPRD (15-12)	↑ 1						
							ICI0D	90		H'00000168 to H'0000016B		↓ 2	
	ATU04	OVI0	92	H'00000170 to H'00000173	0 to 15 (0)	IPRD (11-8)							
	ATU1	ATU11	IMI1A/ CMI1	96	H'00000180 to H'00000183	0 to 15 (0)	IPRD (7-4)	↑ 1					
				IMI1B	97			H'00000184 to H'00000187			2		
				IMI1C	98			H'00000188 to H'0000018B			3		
				IMI1D	99			H'0000018C to H'0000018F			↓ 4		
		ATU12	IMI1E	100	H'00000190 to H'00000193	0 to 15 (0)	IPRD (3-0)	↑ 1					
								IMI1F		101	H'00000194 to H'00000197		2
								IMI1G		102	H'00000198 to H'0000019B		3
IMI1H								103	H'0000019C to H'0000019F		↓ 4		
ATU13		OVI1A/ OVI1B	104	H'000001A0 to H'000001A3	0 to 15 (0)	IPRE (15-12)			Low				

**Table 7.3 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR		Default Priority
		Vector No.	Vector Table Address Offset			Setting Range		
ATU2	ATU21	IMI2A/ CMI2A	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPRE (11–8)	↑ 1	High
		IMI2B/ CMI2B	109	H'000001B4 to H'000001B7			2	
		IMI2C/ CMI2C	110	H'000001B8 to H'000001BB			3	
		IMI2D/ CMI2D	111	H'000001BC to H'000001BF			↓ 4	
	ATU22	IMI2E/ CMI2E	112	H'000001C0 to H'000001C3	0 to 15 (0)	IPRE (7–4)	↑ 1	
		IMI2F/ CMI2F	113	H'000001C4 to H'000001C7			2	
		IMI2G/ CMI2G	114	H'000001C8 to H'000001CB			3	
		IMI2H/ CMI2H	115	H'000001CC to H'000001CF			↓ 4	
	ATU23	OVI2A/ OVI2B	116	H'000001D0 to H'000001D3	0 to 15 (0)	IPRE (3–0)		
ATU3	ATU31	IMI3A	120	H'000001E0 to H'000001E3	0 to 15 (0)	IPRF (15–12)	↑ 1	
		IMI3B	121	H'000001E4 to H'000001E7			2	
		IMI3C	122	H'000001E8 to H'000001EB			3	
		IMI3D	123	H'000001EC to H'000001EF			↓ 4	
	ATU32	OVI3	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPRF (11–8)		Low

**Table 7.3 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range		Default Priority
			Vector No.	Vector Address Offset					
ATU4	ATU41	IMI4A	128	H'00000200 to H'00000203	0 to 15 (0)	IPRF (7–4)	↑	1	High
		IMI4B	129	H'00000204 to H'00000207			2		
		IMI4C	130	H'00000208 to H'0000020B			3		
		IMI4D	131	H'0000020C to H'0000020F			↓	4	
	ATU42	OVI4	132	H'00000210 to H'00000213	0 to 15 (0)	IPRF (3–0)			
ATU5	ATU51	IMI5A	136	H'00000220 to H'00000223	0 to 15 (0)	IPRG (15–12)	↑	1	
		IMI5B	137	H'00000224 to H'00000227			2		
		IMI5C	138	H'00000228 to H'0000022B			3		
		IMI5D	139	H'0000022C to H'0000022F			↓	4	
	ATU52	OVI5	140	H'00000230 to H'00000233	0 to 15 (0)	IPRG (11–8)			
ATU6		CMI6A	144	H'00000240 to H'00000243	0 to 15 (0)	IPRG (7–4)	↑	1	
		CMI6B	145	H'00000244 to H'00000247			2		
		CMI6C	146	H'00000248 to H'0000024B			3		
		CMI6D	147	H'0000024C to H'0000024F			↓	4	

**Table 7.3 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source	Interrupt Vector			Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR		Default Priority
	Vector No.	Vector Address Offset	Vector Table Address Offset			Setting Range		
ATU7	CMI7A	148	H'00000250 to H'00000253	0 to 15 (0)	IPRG (3-0)	↑	1	High
	CMI7B	149	H'00000254 to H'00000257			2		
	CMI7C	150	H'00000258 to H'0000025B			3		
	CMI7D	151	H'0000025C to H'0000025F			↓	4	
ATU8 ATU81	OSI8A	152	H'00000260 to H'00000263	0 to 15 (0)	IPRH (15-12)	↑	1	↑
	OSI8B	153	H'00000264 to H'00000267			2		
	OSI8C	154	H'00000268 to H'0000026B			3		
	OSI8D	155	H'0000026C to H'0000026F			↓	4	
ATU82	OSI8E	156	H'00000270 to H'00000273	0 to 15 (0)	IPRH (11-8)	↑	1	↓
	OSI8F	157	H'00000274 to H'00000277			2		
	OSI8G	158	H'00000278 to H'0000027B			3		
	OSI8H	159	H'0000027C to H'0000027F			↓	4	
ATU83	OSI8I	160	H'00000280 to H'00000283	0 to 15 (0)	IPRH (7-4)	↑	1	↓
	OSI8J	161	H'00000284 to H'00000287			2		
	OSI8K	162	H'00000288 to H'0000028B			3		
	OSI8L	163	H'0000028C to H'0000028F			↓	4	



**Table 7.3 Interrupt Exception Processing Vectors and Priorities (cont)**

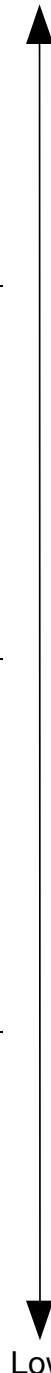
Interrupt Source			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range		Default Priority	
			Vector No.	Vector Address Offset			↑	↓		
ATU8	ATU84	OSI8M	164	H'00000290 to H'00000293	0 to 15 (0)	IPRH (3–0)	↑	1	High	
		OSI8N	165	H'00000294 to H'00000297			2			
		OSI8O	166	H'00000298 to H'0000029B			3			
		OSI8P	167	H'0000029C to H'0000029F			↓	4		
ATU9	ATU91	CMI9A	168	H'000002A0 to H'000002A3	0 to 15 (0)	IPRI (15–12)	↑	1	↑	
		CMI9B	169	H'000002A4 to H'000002A7			2			
		CMI9C	170	H'000002A8 to H'000002AB			3			
		CMI9D	171	H'000002AC to H'000002AF			↓	4		
	ATU92	CMI9E	172	H'000002B0 to H'000002B3	0 to 15 (0)	IPRI (11–8)	↑	1		
		CMI9F	174	H'000002B8 to H'000002BB			↓	2		
	ATU10	ATU101	CMI10A	176	H'000002C0 to H'000002C3	0 to 15 (0)	IPRI (7–4)	↑		1
			CMI10B	178	H'000002C8 to H'000002CB			↓		2
ATU102		ICI10A/ CMI10G	180	H'000002D0 to H'000002D3	IPRI (3–0)					
ATU11		IMI11A	184	H'000002E0 to H'000002E3	0 to 15 (0)	IPRJ (15–12)	↑	1	↓	
		IMI11B	186	H'000002E8 to H'000002EB			2			
		OVI11	187	H'000002EC to H'000002EF			↓	3		Low

**Table 7.3 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source	Interrupt Vector			Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range		Default Priority	
	Vector No.	Vector Address Offset	Vector Table Address Offset			Setting	Range		
CMT0	CMTI0	188	H'000002F0 to H'000002F3	0 to 15 (0)	I PRJ (11–8)	↑	1	High ↑ ↓ Low	
MTAD0	ADT0	189	H'000002F4 to H'000002F7				2		
A/D0	ADI0	190	H'000002F8 to H'000002FB			↓	3		
CMT1	CMTI1	192	H'00000300 to H'00000303	0 to 15 (0)	IPRJ (7–4)	↑	1		
MTAD1	ADT1	193	H'00000304 to H'00000307				2		
A/D1	ADI1	194	H'00000308 to H'0000030B			↓	3		
A/D2	ADI2	196	H'00000310 to H'00000313	0 to 15 (0)	IPRJ (3–0)				
SCI0	ERI0	200	H'00000320 to H'00000323	0 to 15 (0)	IPRK (15–12)	↑	1		
	RXI0	201	H'00000324 to H'00000327						2
	TXI0	202	H'00000328 to H'0000032B						3
	TEI0	203	H'0000032C to H'0000032F				↓	4	
SCI1	ERI1	204	H'00000330 to H'00000333	0 to 15 (0)	IPRK (11–8)	↑	1		
	RXI1	205	H'00000334 to H'00000337					2	
	TXI1	206	H'00000338 to H'0000033B					3	
	TEI1	207	H'0000033C to H'0000033F				↓	4	

**Table 7.3 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source	Interrupt Vector			Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR		Default Priority
	Vector No.	Vector Address Offset	Vector Table Address Offset			Setting Range	Range	
SCI2	ERI2	208	H'00000340 to H'00000343	0 to 15 (0)	IPRK (7-4)	↑	1	High
	RXI2	209	H'00000344 to H'00000347			2		
	TXI2	210	H'00000348 to H'0000034B			3		
	TEI2	211	H'0000034C to H'0000034F			↓	4	
SCI3	ERI3	212	H'00000350 to H'00000353	0 to 15 (0)	IPRK (3-0)	↑	1	
	RXI3	213	H'00000354 to H'00000357			2		
	TXI3	214	H'00000358 to H'0000035B			3		
	TEI3	215	H'0000035C to H'0000035F			↓	4	
SCI4	ERI4	216	H'00000360 to H'00000363	0 to 15 (0)	IPRL (15-12)	↑	1	
	RXI4	217	H'00000364 to H'00000367			2		
	TXI4	218	H'00000368 to H'0000036B			3		
	TEI4	219	H'0000036C to H'0000036F			↓	4	
HCAN0	ERS0	220	H'00000370 to H'00000373	0 to 15 (0)	IPRL (11-8)	↑	1	
	OVR0	221	H'00000374 to H'00000377			2		
	RM0	222	H'00000378 to H'0000037B			3		
	SLE0	223	H'0000037C to H'0000037F			↓	4	



**Table 7.3 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR		Default Priority
		Vector No.	Vector Table Address Offset			Setting Range		
WDT	ITI	224	H'00000380 to H'00000383	0 to 15 (0)	IPRL (7-4)			High
HCAN1	ERS1	228	H'00000390 to H'00000393	0 to 15 (0)	IPRL (3-0)	↑	1	↑ ↓
	OVR1	229	H'00000394 to H'00000397			2		
	RM1	230	H'00000398 to H'0000039B			3		
	SLE1	231	H'0000039C to H'0000039F			↓ 4		
								Low

## 7.3 Description of Registers

### 7.3.1 Interrupt Priority Registers A-L (IPRA-IPRL)

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

Interrupt priority registers A-L (IPRA-IPRL) are 16-bit readable/writable registers that set priority levels from 0 to 15 for IRQ interrupts and on-chip peripheral module interrupts. Correspondence between interrupt request sources and each of the IPRA-IPRL bits is shown in table 7.4.

**Table 7.4 Interrupt Request Sources and IPRA–IPRL**

Register	Bits			
	15–12	11–8	7–4	3–0
Interrupt priority register A	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register B	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register C	DMAC0, 1	DMAC2, 3	ATU01	ATU02
Interrupt priority register D	ATU03	ATU04	ATU11	ATU12
Interrupt priority register E	ATU13	ATU21	ATU22	ATU23
Interrupt priority register F	ATU31	ATU32	ATU41	ATU42
Interrupt priority register G	ATU51	ATU52	ATU6	ATU7
Interrupt priority register H	ATU81	ATU82	ATU83	ATU84
Interrupt priority register I	ATU91	ATU92	ATU101	ATU102
Interrupt priority register J	ATU11	CMT0, A/D0, MTAD0	CMT1, A/D1, MTAD1	A/D2
Interrupt priority register K	SCI0	SCI1	SCI2	SCI3
Interrupt priority register L	SCI4	HCAN0	WDT	HCAN1

As indicated in table 7.4, four  $\overline{\text{IRQ}}$  pins or groups of 4 on-chip peripheral modules are allocated to each register. Each of the corresponding interrupt priority ranks are established by setting a value from H'0 (0000) to H'F (1111) in each of the four-bit groups 15–12, 11–8, 7–4 and 3–0. Interrupt priority rank becomes level 0 (lowest) by setting H'0, and level 15 (highest) by setting H'F. If multiple on-chip peripheral modules are assigned to the same bit (DMAC0 and DMAC1, DMAC2 and DMAC3, CMT0, A/D0, and MTAD0, and CMT1, A/D1, and MTAD1), those multiple modules are set to the same priority rank.

IPRA–IPRL are initialized to H'0000 by a reset and in hardware standby mode. They are not initialized in software standby mode.

### 7.3.2 Interrupt Control Register (ICR)

Bit:	15	14	13	12	11	10	9	8
	NMIL	—	—	—	—	—	—	NMIE
Initial value:	*	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	IRQ0S	IRQ1S	IRQ2S	IRQ3S	IRQ4S	IRQ5S	IRQ6S	IRQ7S
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: \* When NMI input is high: 1; when NMI input is low: 0

ICR is a 16-bit register that sets the input signal detection mode of the external interrupt input pin NMI and  $\overline{\text{IRQ0}} - \overline{\text{IRQ7}}$  and indicates the input signal level at the NMI pin. A reset and hardware standby mode initialize ICR but the software standby mode does not.

- Bit 15—NMI Input Level (NMIL): Sets the level of the signal input at the NMI pin. This bit can be read to determine the NMI pin level. This bit cannot be modified.

Bit 15: NMIL	Description
0	NMI input level is low
1	NMI input level is high

- Bits 14 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—NMI Edge Select (NMIE)

Bit 8: NMIE	Description
0	Interrupt request is detected on falling edge of NMI input (Initial value)
1	Interrupt request is detected on rising edge of NMI input

- Bits 7 to 0—IRQ0–IRQ7 Sense Select (IRQ0S–IRQ7S): These bits set the IRQ0–IRQ7 interrupt request detection mode.

Bits 7-0: IRQ0S–IRQ7S	Description
0	Interrupt request is detected on low level of IRQ input (Initial value)
1	Interrupt request is detected on falling edge of IRQ input

### 7.3.3 IRQ Status Register (ISR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	IRQ0F	IRQ1F	IRQ2F	IRQ3F	IRQ4F	IRQ5F	IRQ6F	IRQ7F
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

ISR is a 16-bit register that indicates the interrupt request status of the external interrupt input pins  $\overline{\text{IRQ0}}\text{--}\overline{\text{IRQ7}}$ . When IRQ interrupts are set to edge detection, held interrupt requests can be withdrawn by writing 0 to  $\text{IRQnF}$  after reading  $\text{IRQnF} = 1$ .

A reset and hardware standby mode initialize ISR but software standby mode does not.

- Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 7 to 0—IRQ0–IRQ7 Flags (IRQ0F–IRQ7F): These bits display the IRQ0–IRQ7 interrupt request status.

#### Bits 7-0:

IRQ0F–IRQ7F	Detection Setting	Description
0	Level detection	No IRQn interrupt request exists [Clearing condition] When $\overline{\text{IRQn}}$ input is high
	Edge detection	No IRQn interrupt request was detected (Initial value) [Clearing conditions] <ul style="list-style-type: none"> <li>• When 0 is written after reading <math>\text{IRQnF} = 1</math></li> <li>• When IRQn interrupt exception processing has been executed</li> </ul>
1	Level detection	An IRQn interrupt request exists Setting condition: When $\overline{\text{IRQn}}$ input is low
	Edge detection	An IRQn interrupt request was detected Setting condition: When a falling edge occurs at an $\overline{\text{IRQn}}$ input

n = 7 to 0

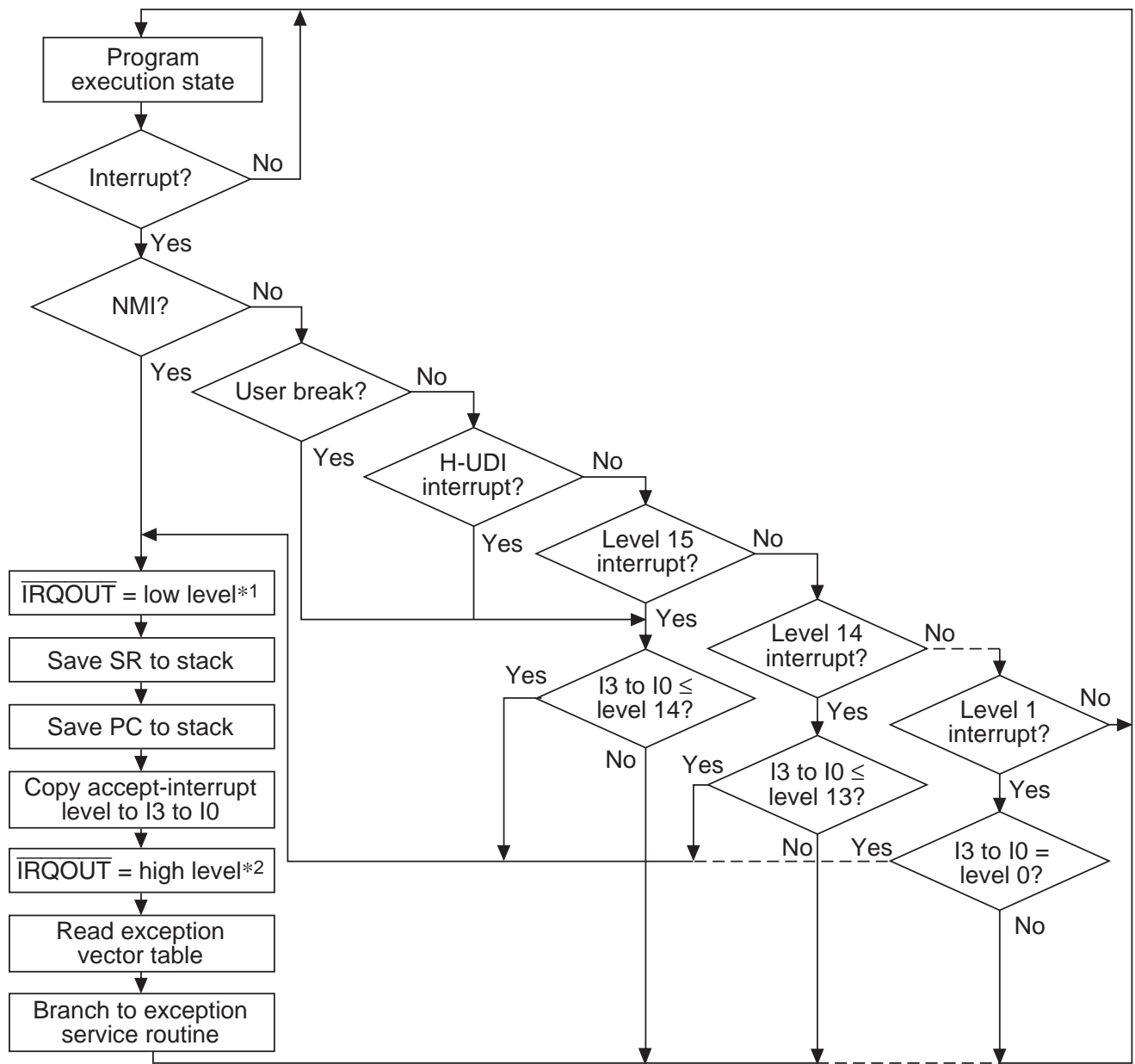
## 7.4 Interrupt Operation

### 7.4.1 Interrupt Sequence

The sequence of interrupt operations is explained below. Figure 7.2 is a flowchart of the operations.

1. The interrupt request sources send interrupt request signals to the interrupt controller.
2. The interrupt controller selects the highest priority interrupt in the interrupt requests sent, following the priority levels set in interrupt priority registers A–L (IPRA–IPRL). Lower-priority interrupts are ignored. They are held pending until interrupt requests designated as edge-detect type are accepted. For IRQ interrupts, however, withdrawal is possible by accessing the IRQ status register (ISR). See section 7.2.4, IRQ Interrupts, for details. Interrupts held pending due to edge detection are cleared by a power-on reset or a manual reset. If two of these interrupts have the same priority level or if multiple interrupts occur within a single module, the interrupt with the highest default priority or the highest priority within its IPR setting range (as indicated in table 7.3) is selected.
3. The interrupt controller compares the priority level of the selected interrupt request with the interrupt mask bits (I3–I0) in the CPU's status register (SR). If the request priority level is equal to or less than the level set in I3–I0, the request is ignored. If the request priority level is higher than the level in bits I3–I0, the interrupt controller accepts the interrupt and sends an interrupt request signal to the CPU.
4. When the interrupt controller accepts an interrupt, a low level is output from the  $\overline{\text{IRQOUT}}$  pin.
5. The CPU detects the interrupt request sent from the interrupt controller when it decodes the next instruction to be executed. Instead of executing the decoded instruction, the CPU starts interrupt exception processing (figure 7.4).
6. SR and PC are saved onto the stack.
7. The priority level of the accepted interrupt is copied to the interrupt mask level bits (I3 to I0) in the status register (SR).
8. When the accepted interrupt is sensed by level or is from an on-chip peripheral module, a high level is output from the  $\overline{\text{IRQOUT}}$  pin. When the accepted interrupt is sensed by edge, a high level is output from the  $\overline{\text{IRQOUT}}$  pin at the point when the CPU starts interrupt exception processing instead of instruction execution as noted in 5 above. However, if the interrupt controller accepts an interrupt with a higher priority than one it is in the process of accepting, the  $\overline{\text{IRQOUT}}$  pin will remain low.
9. The CPU reads the start address of the exception service routine from the exception vector table for the accepted interrupt, jumps to that address, and starts executing the program there. This jump is not a delay branch.





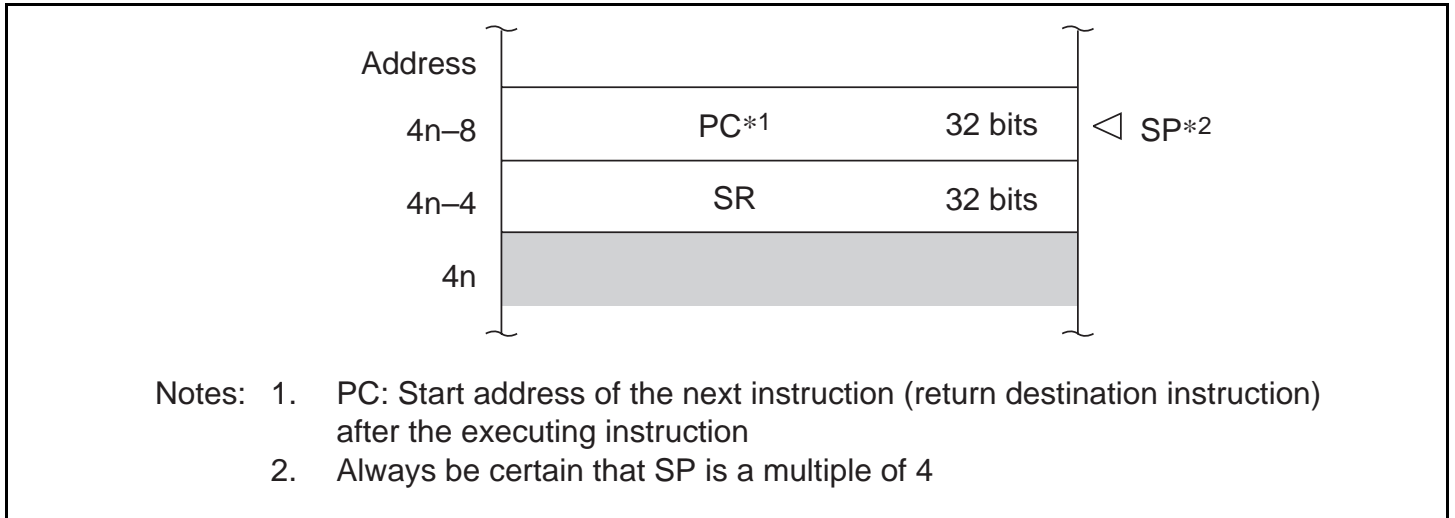
I3 to I0: Interrupt mask bits of status register

- Notes:
1. As  $\overline{\text{IRQOUT}}$  is synchronized with a peripheral clock  $P\phi$ , it may be output later than a CPU interrupt request.
  2. When the accepted interrupt is sensed by edge, the  $\overline{\text{IRQOUT}}$  pin becomes high level at the point when the CPU starts interrupt exception processing instead of instruction execution (before SR is saved to the stack).  
If the interrupt controller has accepted another interrupt with a higher priority and has output an interrupt request to the CPU, the  $\overline{\text{IRQOUT}}$  pin will remain low.

**Figure 7.2 Interrupt Sequence Flowchart**

## 7.4.2 Stack after Interrupt Exception Processing

Figure 7.3 shows the stack after interrupt exception processing.



**Figure 7.3 Stack after Interrupt Exception Processing**

## 7.5 Interrupt Response Time

Table 7.5 indicates the interrupt response time, which is the time from the occurrence of an interrupt request until the interrupt exception processing starts and fetching of the first instruction of the interrupt service routine begins. Figure 7.4 shows an example of pipeline operation when an IRQ interrupt is accepted.

**Table 7.5 Interrupt Response Time (Multiplication Ratio of 8)**

Item	Number of States			Notes
	Peripheral Module	NMI	IRQ	
Synchronizing input signal (synchronized with peripheral clock $P\phi$ ) with internal clock $\phi$ and DMAC activation judgment	0 or 6	1 to 4 [1 or 2]	6 to 9 [3 to 5]	For the number of states required for each interrupt, see the note (*) below. The values enclosed in [ ] are values for when the multiplication ratio is 4.
Compare identified interrupt priority with SR mask level	2	2	2	
Wait for completion of sequence currently being executed by CPU	$X (\geq 0)$			The longest sequence is for interrupt or address-error exception processing ( $X = 4 + m1 + m2 + m3 + m4$ ). If an interrupt-masking instruction follows, however, the time may be even longer.
Time from start of interrupt exception processing until fetch of first instruction of exception service routine starts	$5 + m1 + m2 + m3$			Performs the PC and SR saves and vector address fetch.
Interrupt response time	Total:	$(7 \text{ or } 13) + m1 + m2 + m3 + X$	$(8 \text{ to } 11) + m1 + m2 + m3 + X$	$(13 \text{ to } 16) + m1 + m2 + m3 + X$
	Minimum:	10	11	16
	Maximum:	$17 + 2(m1 + m2 + m3) + m4$	$15 + 2(m1 + m2 + m3) + m4$	$20 + 2(m1 + m2 + m3) + m4$

Note: \* Number of states needed for synchronization and DMAC activation judgment

The relations between numbers of states needed for synchronizing an input signal (synchronized with the peripheral clock  $P\phi$ ) with the internal clock  $\phi$  and DMAC activation judgment and vector numbers are shown below.

0 state: 9, 10, 12, 13, 14, 72, 74, 76, 78, 189, 193, and 224

6 states: Peripheral module interrupts other than the above. However, vector number 222 (HCAN0/RM0) is different from the others.

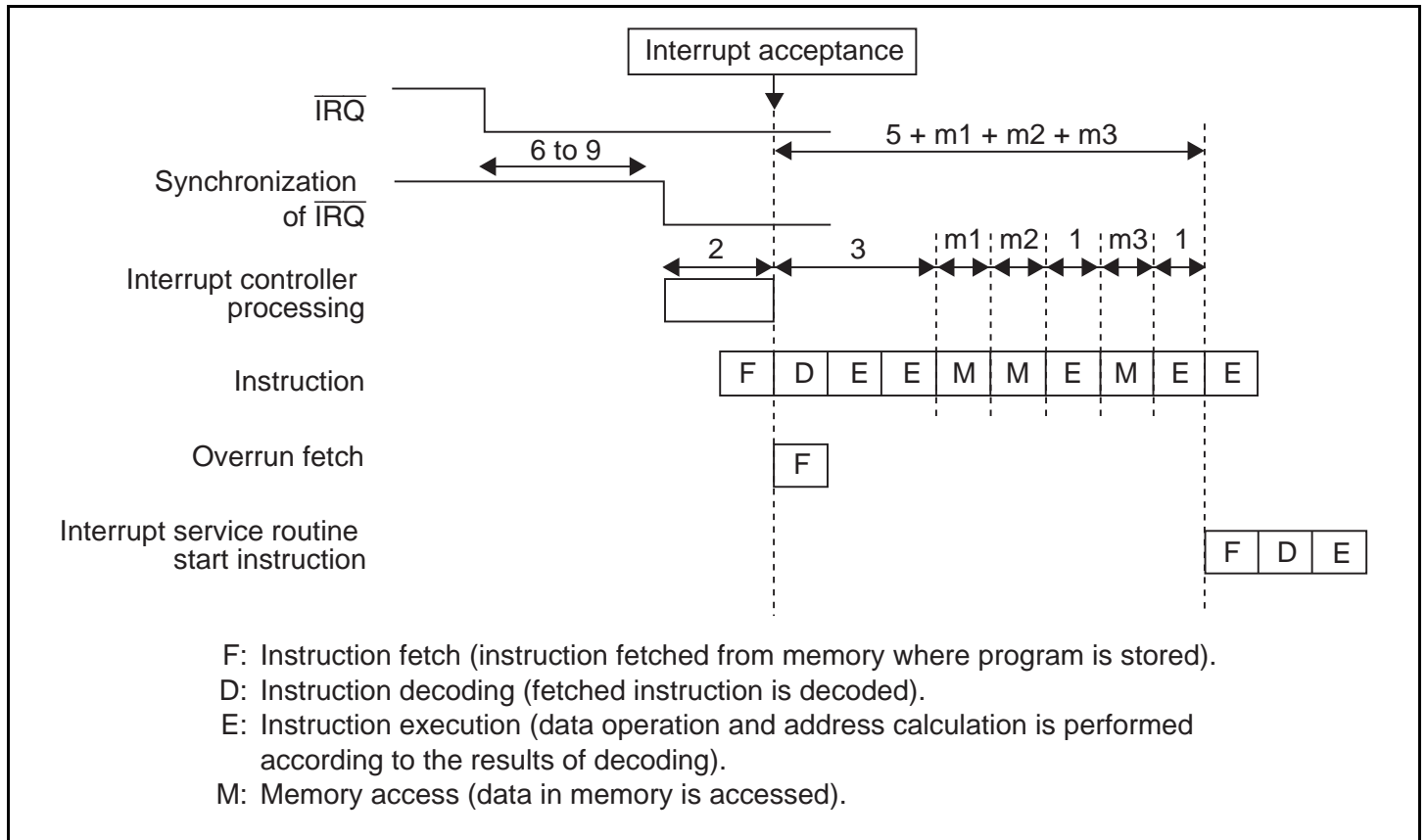
For an interrupt with vector number 222 (HCAN0/RM0), the needed states differ from other interrupts since the interrupt by HCAN0 mailbox 0 can activate the DMAC.

HCAN0 mailbox 0: 7 states

Other than above: 6 states

The same number of states is needed to cancel interrupt sources.

If the necessary number of states is not secured after flag clear of the interrupt source, the interrupt may occur again.



**Figure 7.4 Example of Pipeline Operation when an IRQ Interrupt is Accepted**

## 7.6 Data Transfer with Interrupt Request Signals

The following data transfer can be carried out using interrupt request signals:

- Activate DMAC only, without generating CPU interrupt

Among interrupt sources, those designated as DMAC activating sources are masked and not input to the INTC. The masking condition is as follows:

$$\text{Mask condition} = \text{DME} \cdot (\text{DE0} \cdot \text{source selection 0} + \text{DE1} \cdot \text{source selection 1} + \text{DE2} \cdot \text{source selection 2} + \text{DE3} \cdot \text{source selection 3})$$

### 7.6.1 Handling CPU Interrupt Sources, but Not DMAC Activating Sources

1. Either do not select the DMAC as a source, or clear the DME bit to 0.
2. Activating sources are applied to the CPU when interrupts occur.
3. The CPU clears interrupt sources with its interrupt processing routine and performs the necessary processing.

### 7.6.2 Handling DMAC Activating Sources but Not CPU Interrupt Sources

1. Select the DMAC as a source and set the DME bit to 1. CPU interrupt sources are masked regardless of the interrupt priority level register settings.
2. Activating sources are applied to the DMAC when interrupts occur.
3. The DMAC clears activating sources at the time of data transfer.

# Section 8 User Break Controller (UBC)

## 8.1 Overview

The user break controller (UBC) provides functions that simplify program debugging. Break conditions are set in the UBC and a user break interrupt is generated according to the conditions of the bus cycle generated by the CPU or DMAC. This function makes it easy to design an effective self-monitoring debugger, enabling the chip to easily debug programs without using a large in-circuit emulator.

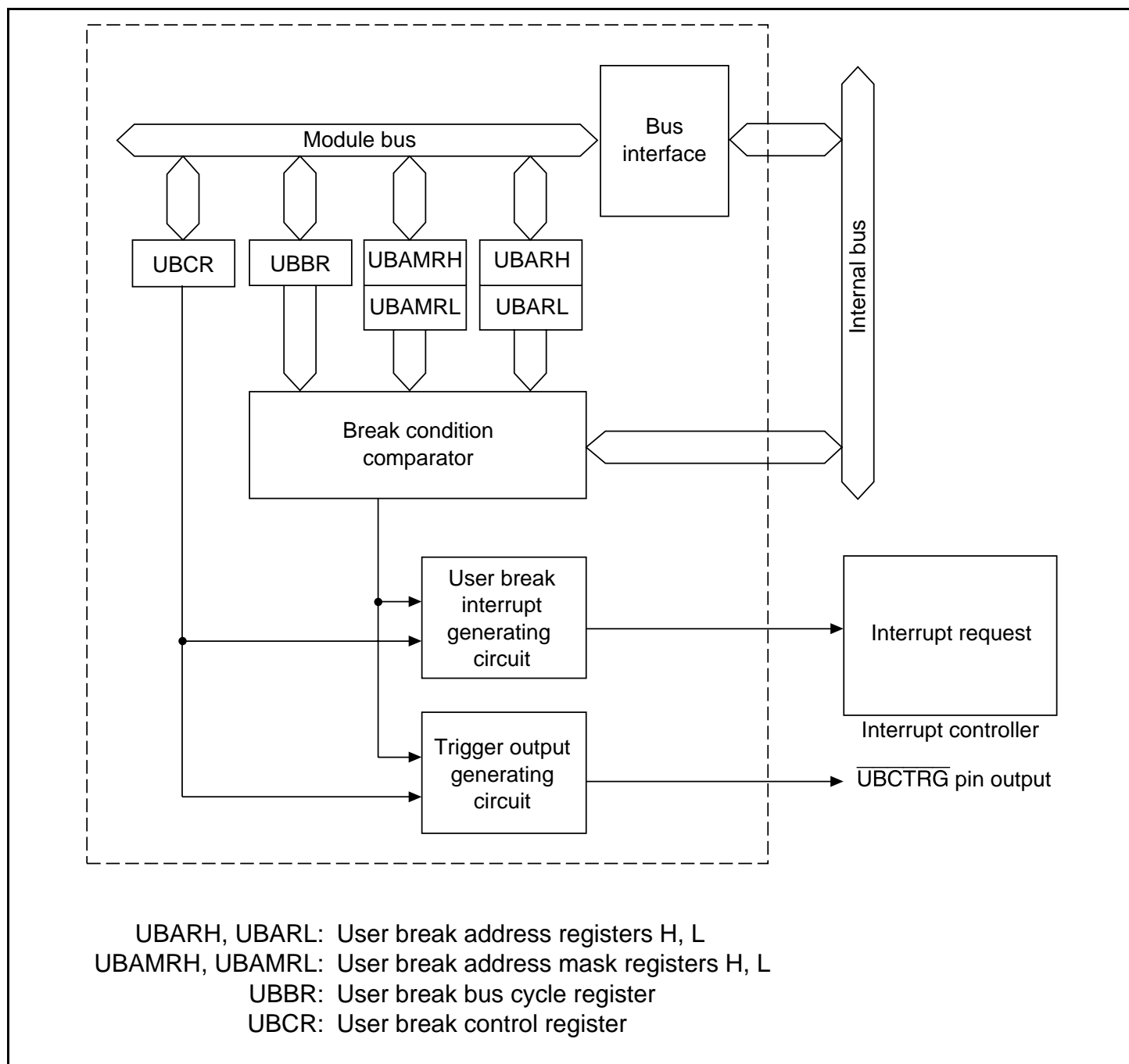
### 8.1.1 Features

The features of the user break controller are:

- The following break compare conditions can be set:
  - Address
  - CPU cycle/DMA cycle
  - Instruction fetch or data access
  - Read or write
  - Operand size: byte/word/longword
- User break interrupt generated upon satisfying break conditions  
A user-designed user break interrupt exception processing routine can be run.
- Select either to break in the CPU instruction fetch cycle before the instruction is executed or after.
- Satisfaction of a break condition can be output to the  $\overline{\text{UBCTR}}\overline{\text{G}}$  pin.

## 8.1.2 Block Diagram

Figure 8.1 shows a block diagram of the UBC.



**Figure 8.1 User Break Controller Block Diagram**

### 8.1.3 Register Configuration

The UBC has the six registers shown in table 8.1. Break conditions are established using these registers.

**Table 8.1 Register Configuration**

Name	Abbr.	R/W	Initial Value	Address*	Access Size
User break address register H	UBARH	R/W	H'0000	H'FFFEC00	8, 16, 32
User break address register L	UBARL	R/W	H'0000	H'FFFEC02	8, 16, 32
User break address mask register H	UBAMRH	R/W	H'0000	H'FFFEC04	8, 16, 32
User break address mask register L	UBAMRL	R/W	H'0000	H'FFFEC06	8, 16, 32
User break bus cycle register	UBBR	R/W	H'0000	H'FFFEC08	8, 16, 32
User break control register	UBCR	R/W	H'0000	H'FFFEC0A	8, 16, 32

Note: \* In register access, four cycles are required for byte access and word access, and eight cycles for longword access.

## 8.2 Register Descriptions

### 8.2.1 User Break Address Register (UBAR)

#### UBARH:

Bit:	15	14	13	12	11	10	9	8
	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24
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:	7	6	5	4	3	2	1	0
	UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16
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



## UBARL:

Bit:	15	14	13	12	11	10	9	8
	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8
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:	7	6	5	4	3	2	1	0
	UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0
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

The user break address register (UBAR) consists of user break address register H (UBARH) and user break address register L (UBARL). Both are 16-bit readable/writable registers. UBARH stores the upper bits (bits 31 to 16) of the address of the break condition, while UBARL stores the lower bits (bits 15 to 0). UBARH and UBARL are initialized to H'0000 by a power-on reset and in module standby mode. They are not initialized in software standby mode.

- UBARH Bits 15 to 0—User Break Address 31 to 16 (UBA31 to UBA16): These bits store the upper bit values (bits 31 to 16) of the address of the break condition.
- UBARL Bits 15 to 0—User Break Address 15 to 0 (UBA15 to UBA0): These bits store the lower bit values (bits 15 to 0) of the address of the break condition.

## 8.2.2 User Break Address Mask Register (UBAMR)

### UBAMRH:

Bit:	15	14	13	12	11	10	9	8
	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24
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:	7	6	5	4	3	2	1	0
	UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16
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

## UBAMRL:

Bit:	15	14	13	12	11	10	9	8
	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8
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:	7	6	5	4	3	2	1	0
	UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0
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

The user break address mask register (UBAMR) consists of user break address mask register H (UBAMRH) and user break address mask register L (UBAMRL). Both are 16-bit readable/writable registers. UBAMRH designates whether to mask any of the break address bits established in UBARH, and UBAMRL designates whether to mask any of the break address bits established in UBARL. UBAMRH and UBAMRL are initialized to H'0000 by a power-on reset and in module standby mode. They are not initialized in software standby mode.

- UBAMRH Bits 15 to 0—User Break Address Mask 31 to 16 (UBM31 to UBM16): These bits designate whether to mask the corresponding break address 31 to 16 bits (UBA31 to UBA16) established in UBARH.
- UBAMRL Bits 15 to 0—User Break Address Mask 15 to 0 (UBM15 to UBM0): These bits designate whether to mask the corresponding break address 15 to 0 bits (UBA15 to UBA0) established in UBARL.

Bits 15–0: UBMn	Description
0	Break address UBA <sub>n</sub> is included in the break conditions (Initial value)
1	Break address UBA <sub>n</sub> is not included in the break conditions

Note: n = 31 to 0

### 8.2.3 User Break Bus Cycle Register (UBBR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0
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

The user break bus cycle register (UBBR) is a 16-bit readable/writable register that selects from among the following four break conditions:

1. CPU cycle/DMA cycle
2. Instruction fetch/data access
3. Read/write
4. Operand size (byte, word, longword)

UBBR is initialized to H'0000 by a power on reset and in module standby mode. It is not initialized in software standby mode.

- Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 7 and 6—CPU Cycle/DMA Cycle Select (CP1, CP0): These bits designate break conditions for CPU cycles or DMA cycles.

Bit 7: CP1	Bit 6: CP0	Description
0	0	No user break interrupt occurs (Initial value)
	1	Break on CPU cycles
1	0	Break on DMA cycles
	1	Break on both CPU and DMA cycles

- Bits 5 and 4—Instruction Fetch/Data Access Select (ID1, ID0): These bits select whether to break on instruction fetch and/or data access cycles.

Bit 5: ID1	Bit 4: ID0	Description
0	0	No user break interrupt occurs (Initial value)
	1	Break on instruction fetch cycles
1	0	Break on data access cycles
	1	Break on both instruction fetch and data access cycles

- Bits 3 and 2—Read/Write Select (RW1, RW0): These bits select whether to break on read and/or write cycles.

Bit 3: RW1	Bit 2: RW0	Description
0	0	No user break interrupt occurs (Initial value)
	1	Break on read cycles
1	0	Break on write cycles
	1	Break on both read and write cycles

- Bits 1 and 0—Operand Size Select (SZ1, SZ0): These bits select operand size as a break condition.

Bit 1: SZ1	Bit 0: SZ0	Description
0	0	Operand size is not a break condition (Initial value)
	1	Break on byte access
1	0	Break on word access
	1	Break on longword access

Note: When breaking on an instruction fetch, clear the SZ0 bit to 0. All instructions are considered to be word-size accesses (even when there are instructions in on-chip memory and two instruction fetches are performed simultaneously in one bus cycle).

Operand size is word for instructions or determined by the operand size specified for the CPU/DMAC data access. It is not determined by the bus width of the space being accessed.

## 8.2.4 User Break Control Register (UBCR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	CKS1	CKS0	UBID
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

The user break control register (UBCR) is a 16-bit readable/writable register that (1) enables or disables user break interrupts and (2) sets the pulse width of the  $\overline{\text{UBCTR}}\overline{\text{G}}$  signal output in the event of a break condition match.

UBCR is initialized to H'0000 by a power-on reset and in module standby mode. It is not initialized in software standby mode.

- Bits 15 to 3—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 2 and 1—Clock Select 1 and 0 (CKS1, CKS0): These bits specify the pulse width of the  $\overline{\text{UBCTR}}\overline{\text{G}}$  signal output in the event of a condition match.

Bit 2: CKS1	Bit 1: CKS0	Description
0	0	When the internal clock is four times an input clock, $\overline{\text{UBCTR}}\overline{\text{G}}$ pulse width is $\phi/2$ When the internal clock is eight times an input clock, $\overline{\text{UBCTR}}\overline{\text{G}}$ pulse width is $\phi/4$ (Initial value)
	1	$\overline{\text{UBCTR}}\overline{\text{G}}$ pulse width is $\phi/4$
1	0	$\overline{\text{UBCTR}}\overline{\text{G}}$ pulse width is $\phi/8$
	1	$\overline{\text{UBCTR}}\overline{\text{G}}$ pulse width is $\phi/16$

Notes:  $\phi$ : Internal clock

See section 8.5.7, Internal Clock ( $\phi$ ) Multiplication Ratio and  $\overline{\text{UBCTR}}\overline{\text{G}}$  Pulse Width.

- Bit 0—User Break Disable (UBID): Enables or disables user break interrupt request generation in the event of a user break condition match.

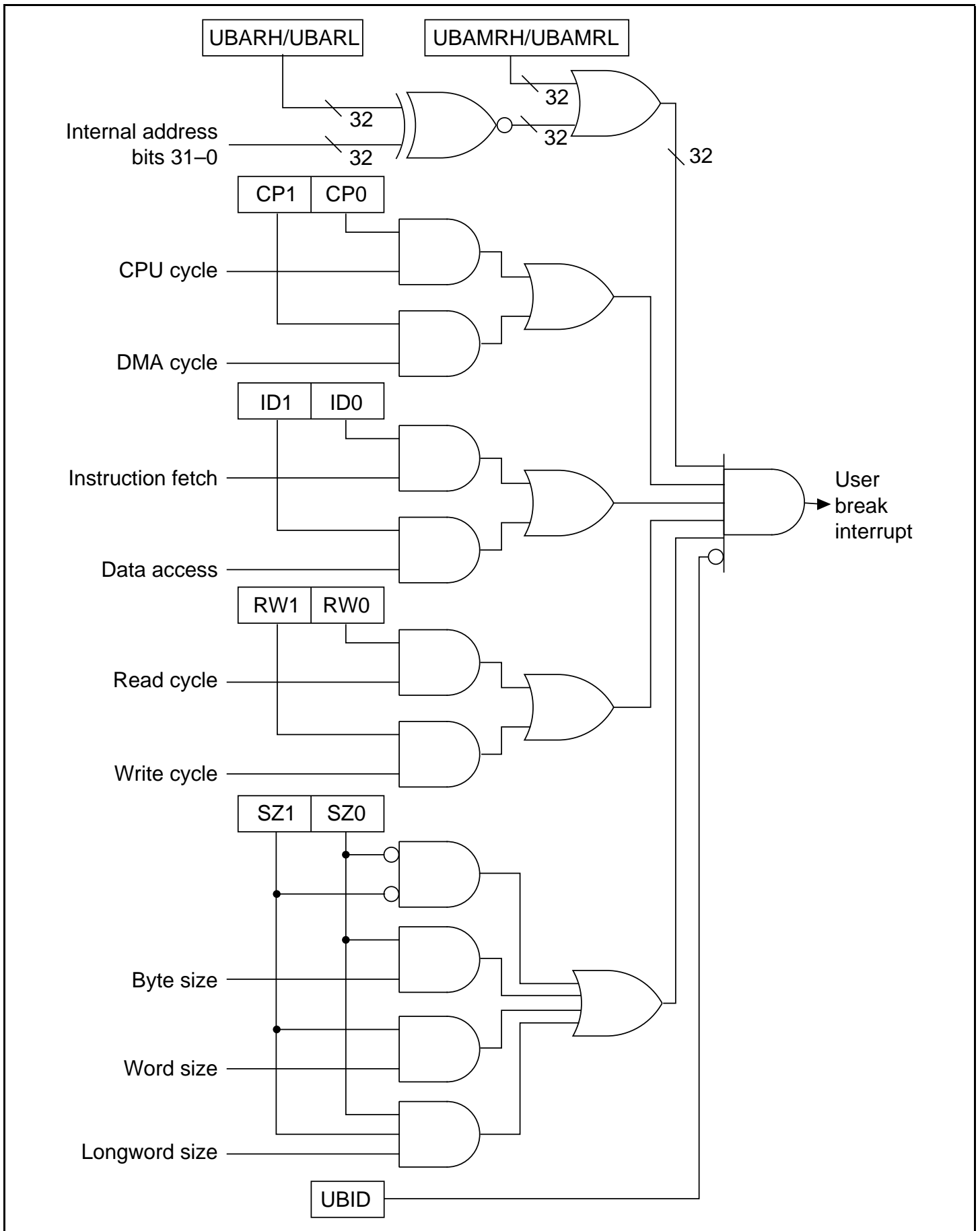
Bit 0: UBID	Description
0	User break interrupt request is enabled (Initial value)
1	User break interrupt request is disabled

## 8.3 Operation

### 8.3.1 Flow of the User Break Operation

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

1. The user break addresses are set in the user break address register (UBAR), the desired masked bits in the addresses are set in the user break address mask register (UBAMR) and the breaking bus cycle type is set in the user break bus cycle register (UBBR). If even one of the three groups of the UBBR's CPU cycle/DMA cycle select bits (CP1, CP0), instruction fetch/data access select bits (ID1, ID0), and read/write select bits (RW1, RW0) is set to 00 (no user break generated), no user break interrupt will be generated even if all other conditions are in agreement. When using user break interrupts, always be certain to establish bit conditions for all of these three groups.
2. The UBC uses the method shown in figure 8.2 to judge whether set conditions have been fulfilled. When the set conditions are satisfied, the UBC sends a user break interrupt request signal to the interrupt controller (INTC). At the same time, a condition match signal is output at the  $\overline{\text{UBCTR}}\overline{\text{G}}$  pin with the pulse width set in bits CKS1 and CKS0.
3. The interrupt controller checks the accepted user break interrupt request signal's priority level. The user break interrupt has priority level 15, so it is accepted only if the interrupt mask level in bits I3–I0 in the status register (SR) is 14 or lower. When the I3–I0 bit level is 15, the user break interrupt cannot be accepted but it is held pending until user break interrupt exception processing can be carried out. Consequently, user break interrupts within NMI exception service routines cannot be accepted, since the I3–I0 bit level is 15. However, if the I3–I0 bit level is changed to 14 or lower at the start of the NMI exception service routine, user break interrupts become acceptable thereafter. Section 7, Interrupt Controller (INTC), describes the handling of priority levels in greater detail.
4. The INTC sends the user break interrupt request signal to the CPU, which begins user break interrupt exception processing upon receipt. See section 7.4, Interrupt Operation, for details on interrupt exception processing.



**Figure 8.2 Break Condition Judgment Method**

### 8.3.2 Break on On-Chip Memory Instruction Fetch Cycle

On-chip memory (on-chip ROM and/or RAM) is always accessed as 32 bits in one bus cycle. Therefore, two instructions can be retrieved in one bus cycle when fetching instructions from on-chip memory. At such times, only one bus cycle is generated, but by setting the start addresses of both instructions in the user break address register (UBAR) it is possible to cause independent breaks. In other words, when wanting to effect a break using the latter of two addresses retrieved in one bus cycle, set the start address of that instruction in UBAR. The break will occur after execution of the former instruction.

### 8.3.3 Program Counter (PC) Values Saved

**Break on Instruction Fetch:** The program counter (PC) value saved to the stack in user break interrupt exception processing is the address that matches the break condition. The user break interrupt is generated before the fetched instruction is executed. If a break condition is set in an instruction fetch cycle placed immediately after a delayed branch instruction (delay slot), or on an instruction that follows an interrupt-disabled instruction, however, the user break interrupt is not accepted immediately, but the break condition establishing instruction is executed. The user break interrupt is accepted after execution of the instruction that has accepted the interrupt. In this case, the PC value saved is the start address of the instruction that will be executed after the instruction that has accepted the interrupt.

**Break on Data Access (CPU/DMA):** The program counter (PC) value is the top address of the next instruction after the last instruction executed before the user break exception processing started. When data access (CPU/DMA) is set as a break condition, the place where the break will occur cannot be specified exactly. The break will occur at the instruction fetched close to where the data access that is to receive the break occurs.

## 8.4 Examples of Use

### 8.4.1 Break on CPU Instruction Fetch Cycle

1. Register settings:      UBARH = H'0000  
                              UBARL = H'0404  
                              UBBR = H'0054  
                              UBCR = H'0000  
  
Conditions set:            Address: H'00000404  
                              Bus cycle: CPU, instruction fetch, read  
                              (operand size not included in conditions)  
                              Interrupt requests enabled



A user break interrupt will occur before the instruction at address H'00000404. If it is possible for the instruction at H'00000402 to accept an interrupt, the user break exception processing will be executed after execution of that instruction. The instruction at H'00000404 is not executed. The PC value saved is H'00000404.

2. Register settings:      UBARH = H'0015  
                              UBARL = H'389C  
                              UBBR = H'0058  
                              UBCR = H'0000  
  
Conditions set:            Address: H'0015389C  
                              Bus cycle: CPU, instruction fetch, write  
                              (operand size not included in conditions)  
                              Interrupt requests enabled

A user break interrupt does not occur because the instruction fetch cycle is not a write cycle.

3. Register settings:      UBARH = H'0003  
                              UBARL = H'0147  
                              UBBR = H'0054  
                              UBCR = H'0000  
  
Conditions set:            Address: H'00030147  
                              Bus cycle: CPU, instruction fetch, read  
                              (operand size not included in conditions)  
                              Interrupt requests enabled

A user break interrupt does not occur because the instruction fetch was performed for an even address. However, if the first instruction fetch address after the branch is an odd address set by these conditions, user break interrupt exception processing will be carried out after address error exception processing.

#### 8.4.2 Break on CPU Data Access Cycle

1. Register settings:      UBARH = H'0012  
                              UBARL = H'3456  
                              UBBR = H'006A  
                              UBCR = H'0000  
  
Conditions set:            Address: H'00123456  
                              Bus cycle: CPU, data access, write, word  
                              Interrupt requests enabled

A user break interrupt occurs when word data is written into address H'00123456.

2. Register settings:      UBARH = H'00A8  
                              UBARL = H'0391  
                              UBBR = H'0066  
                              UBCR = H'0000

Conditions set:           Address: H'00A80391  
                              Bus cycle: CPU, data access, read, word  
                              Interrupt requests enabled

A user break interrupt does not occur because the word access was performed on an even address.

### 8.4.3     Break on DMA Cycle

1. Register settings:       UBARH = H'0076  
                              UBARL = H'BCDC  
                              UBBR = H'00A7  
                              UBCR = H'0000

Conditions set:           Address: H'0076BCDC  
                              Bus cycle: DMA, data access, read, longword  
                              Interrupt requests enabled

A user break interrupt occurs when longword data is read from address H'0076BCDC.

2. Register settings:       UBARH = H'0023  
                              UBARL = H'45C8  
                              UBBR = H'0094  
                              UBCR = H'0000

Conditions set:           Address: H'002345C8  
                              Bus cycle: DMA, instruction fetch, read  
                              (operand size not included in conditions)  
                              Interrupt requests enabled

A user break interrupt does not occur because no instruction fetch is performed in the DMA cycle.

## 8.5 Usage Notes

### 8.5.1 Simultaneous Fetching of Two Instructions

Two instructions may be simultaneously fetched from on-chip memory. If a break condition is set on the second of these two instructions but the contents of the UBC break condition registers are changed so as to alter the break condition immediately after the first of the two instructions is fetched, a user break interrupt will still occur when the second instruction is fetched.

### 8.5.2 Instruction Fetches at Branches

When a conditional branch instruction or TRAPA instruction causes a branch, the order of instruction fetching and execution is as follows:

1. When branching with a conditional branch instruction: BT and BF instructions  
When branching with a TRAPA instruction: TRAPA instruction  
Instruction fetch order: Branch instruction fetch → next instruction overrun fetch → overrun fetch of instruction after next → branch destination instruction fetch  
Instruction execution order: Branch instruction execution → branch destination instruction execution
2. When branching with a delayed conditional branch instruction: BT/S and BF/S instructions  
Instruction fetch order: Branch instruction fetch → next instruction fetch (delay slot) → overrun fetch of instruction after next → branch destination instruction fetch  
Instruction execution order: Branch instruction execution → delay slot instruction execution → branch destination instruction execution

Thus, when a conditional branch instruction or TRAPA instruction causes a branch, the branch destination instruction will be fetched after an overrun fetch of the next instruction or the instruction after next. However, as the instruction that is the object of the break does not break until fetching and execution of the instruction have been confirmed, the overrun fetches described above do not become objects of a break.

If data accesses are also included as break conditions in addition to instruction fetch breaks, a break will occur because the instruction overrun fetch is also regarded as satisfying the data break condition.

### 8.5.3 Contention between User Break and Exception Processing

If a user break is set for the fetch of a particular instruction, and exception processing with higher priority than a user break is in contention and is accepted in the decode stage for that instruction (or the next instruction), user break exception processing may not be performed after completion of the higher-priority exception service routine (on return by RTE).

Thus, if a user break condition is applied to the branch destination instruction fetch after a branch (BRA, BRAF, BT, BF, BT/S, BF/S, BSR, BSRF, JMP, JSR, RTS, RTE, exception processing), and that branch instruction accepts exception processing with higher priority than a user break interrupt, user break exception processing is not performed after completion of the higher-priority exception service routine.

Therefore, a user break condition should not be set for the fetch of the branch destination instruction after a branch.

### 8.5.4 Break at Non-Delay Branch Instruction Jump Destination

When a branch instruction with no delay slot (including exception processing) jumps to the jump destination instruction on execution of the branch, a user break will not be generated even if a user break condition has been set for the first jump destination instruction fetch.

### 8.5.5 User Break Trigger Output

Information on internal bus condition matches monitored by the UBC is output as  $\overline{\text{UBCTR}}\overline{\text{G}}$ . The trigger width can be set with clock select bits 1 and 0 (CKS1, CKS0) in the user break control register (UBCR).

If a condition matches occurs again during trigger output, the  $\overline{\text{UBCTR}}\overline{\text{G}}$  pin continues to output a low level, and outputs a pulse of the length set in bits CKS1 and CKS0 from the cycle in which the last condition match occurs.

The trigger output conditions differ from those in the case of a user break interrupt when a CPU instruction fetch condition is satisfied. When a condition occurs in an overrun fetch instruction as described in section 8.5.2, Instruction Fetch at Branches, a user break interrupt is not requested but a trigger is output from the  $\overline{\text{UBCTR}}\overline{\text{G}}$  pin.

In other CPU data accesses and DMAC bus cycles, pulse output is performed under conditions similar to user break interrupt conditions.

Setting the user break interrupt disable (UBID) bit to 1 in UBCR enables trigger output to be monitored externally without requesting a user break interrupt.

## 8.5.6 Module Standby

After a power-on reset the UBC is in the module standby state, in which the clock supply is halted. When using the UBC, the module standby state must be cleared before making UBC register settings. Module standby is controlled by the System Control Register 2 (SYSCR2). See section 25.2.3, System Control Register 2 (SYSCR2), for further details.

## 8.5.7 Internal Clock ( $\phi$ ) Multiplication Ratio and $\overline{\text{UBCTR}}\overline{\text{G}}$ Pulse Width

The user break controller operates in synchronization with an internal clock ( $\phi$ ) which is four or eight times an input clock. Even when the same kind of clock is selected by Clock Select 1 and 0 (CKS1 and CKS0) in the user break control register (UBCR), the output pulse width of  $\overline{\text{UBCTR}}\overline{\text{G}}$  is changed according to the internal clock ( $\phi$ ) multiplication ratio (an internal clock is eight or four times an input clock). When the multiplication ratio is changed during the  $\overline{\text{UBCTR}}\overline{\text{G}}$  pulse output, pulse width of  $\overline{\text{UBCTR}}\overline{\text{G}}$  is changed simultaneously.

# Section 9 Bus State Controller (BSC)

## 9.1 Overview

The bus state controller (BSC) divides up the address spaces and outputs control for various types of memory. This enables memories like SRAM and ROM to be linked directly to the chip without external circuitry, simplifying system design and enabling high-speed data transfer to be achieved in a compact system.

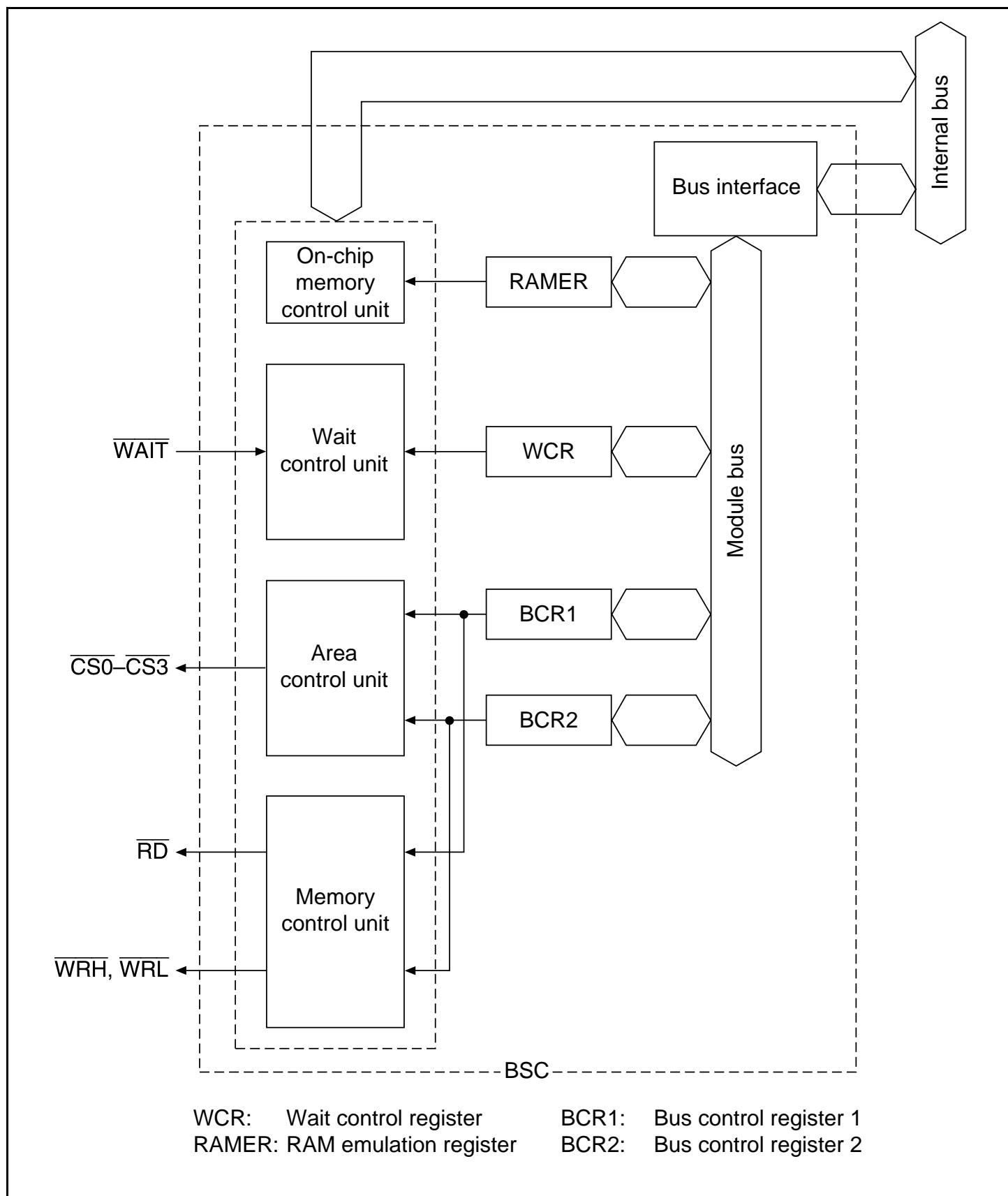
### 9.1.1 Features

The BSC has the following features:

- Address space is divided into four spaces
  - A maximum linear 2 Mbytes for on-chip ROM effective mode, and a maximum 4 Mbytes for on-chip ROM disabled mode, for address space CS0
  - A maximum linear 4 Mbytes for each of address spaces CS1–CS3
  - Bus width can be selected for each space (8 or 16 bits)
  - Wait states can be inserted by software for each space
  - Wait state insertion with  $\overline{\text{WAIT}}$  pin in external memory space access
  - Outputs control signals for each space according to the type of memory connected
- On-chip ROM and RAM interfaces
  - On-chip RAM access of 32 bits in 1 state
  - On-chip ROM access of 32 bits in 1 state for a read and 2 states for a write

## 9.1.2 Block Diagram

Figure 9.1 shows the BSC block diagram.



**Figure 9.1 BSC Block Diagram**

### 9.1.3 Pin Configuration

Table 9.1 shows the bus state controller pin configuration.

**Table 9.1 Pin Configuration**

Name	Abbr.	I/O	Description
Address bus	A21–A0	O	Address output
Data bus	D15–D0	I/O	16-bit data bus
Chip select	$\overline{CS0}$ – $\overline{CS3}$	O	Chip select signals indicating the area being accessed
Read	$\overline{RD}$	O	Strobe that indicates the read cycle for ordinary space/multiplex I/O
Upper write	$\overline{WRH}$	O	Strobe that indicates a write cycle to the upper 8 bits (D15–D8)
Lower write	$\overline{WRL}$	O	Strobe that indicates a write cycle to the lower 8 bits (D7–D0)
Wait	$\overline{WAIT}$	I	Wait state request signal
Bus request	$\overline{BREQ}$	I	Bus release request input
Bus acknowledge	$\overline{BACK}$	O	Bus use enable output

Notes: 1. When an 8-bit bus width is selected for external space,  $\overline{WRL}$  is enabled.  
2. When a 16-bit bus width is selected for external space,  $\overline{WRH}$  and  $\overline{WRL}$  are enabled.

### 9.1.4 Register Configuration

The BSC has four registers. These registers are used to control wait states, bus width, and interfaces with memories like ROM and SRAM, as well as refresh control. The register configurations are listed in table 9.2.

All registers are 16 bits. All BSC registers are all initialized by a power-on reset and in hardware standby mode. Values are retained in a manual reset and in software standby mode.

**Table 9.2 Register Configuration**

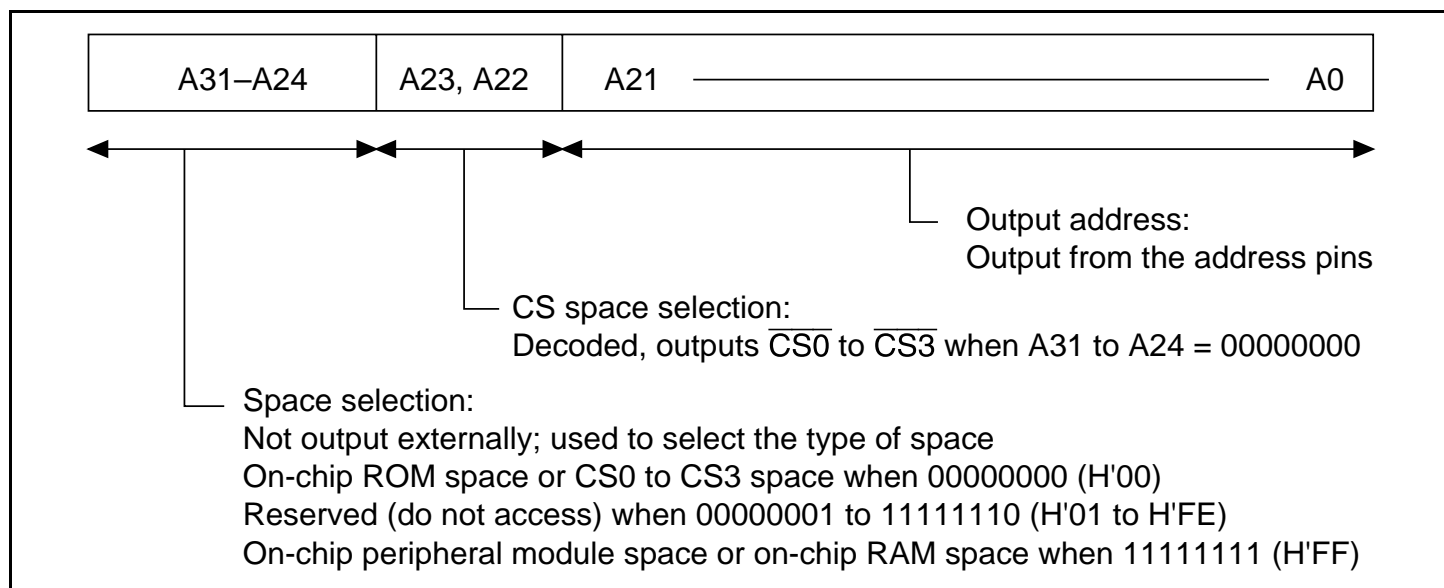
Name	Abbr.	R/W	Initial Value	Address	Access Size
Bus control register 1	BCR1	R/W	H'000F	H'FFFFFFE20	8, 16, 32
Bus control register 2	BCR2	R/W	H'FFFF	H'FFFFFFE22	8, 16, 32
Wait state control register	WCR	R/W	H'7777	H'FFFFFFE24	8, 16, 32
RAM emulation register	RAMER	R/W	H'0000	H'FFFFFFE26	8, 16, 32

Note: In register access, four cycles are required for byte access and word access, and eight cycles for longword access.



## 9.1.5 Address Map

Figure 9.2 shows the address format used by the SH7058.



**Figure 9.2 Address Format**

This chip uses 32-bit addresses:

- Bits A31 to A24 are used to select the type of space and are not output externally.
- Bits A23 and A22 are decoded and output as chip select signals ( $\overline{CS0}$  to  $\overline{CS3}$ ) for the corresponding areas when bits A31 to A24 are 00000000.
- A21 to A0 are output externally.

Table 9.3 shows the address map.

**Table 9.3 Address Map**

- On-chip ROM enabled mode

<b>Address</b>	<b>Space</b>	<b>Memory</b>	<b>Size</b>	<b>Bus Width</b>
H'0000 0000 to H'000F FFFF	On-chip ROM	On-chip ROM	1 MB	32 bits
H'0010 0000 to H'001F FFFF	Reserved	Reserved		
H'0020 0000 to H'003F FFFF	CS0 space	External space	2 MB	8, 16 bits* <sup>1</sup>
H'0040 0000 to H'007F FFFF	CS1 space	External space	4 MB	8, 16 bits* <sup>1</sup>
H'0080 0000 to H'00BF FFFF	CS2 space	External space	4 MB	8, 16 bits* <sup>1</sup>
H'00C0 0000 to H'00FF FFFF	CS3 space	External space	4 MB	8, 16 bits* <sup>1</sup>
H'0100 0000 to H'FFFE FFFF	Reserved	Reserved		
H'FFFF 0000 to H'FFFF BFFF	On-chip RAM	On-chip RAM	48 kB	32 bits
H'FFFF C000 to H'FFFF FFFF	On-chip peripheral module	On-chip peripheral module	16 kB	8, 16 bits

- On-chip ROM disabled mode

<b>Address</b>	<b>Space</b>	<b>Memory</b>	<b>Size</b>	<b>Bus Width</b>
H'0000 0000 to H'003F FFFF	CS0 space	External space	4 MB	8, 16 bits* <sup>2</sup>
H'0040 0000 to H'007F FFFF	CS1 space	External space	4 MB	8, 16 bits* <sup>1</sup>
H'0080 0000 to H'00BF FFFF	CS2 space	External space	4 MB	8, 16 bits* <sup>1</sup>
H'00C0 0000 to H'00FF FFFF	CS3 space	External space	4 MB	8, 16 bits* <sup>1</sup>
H'0100 0000 to H'FFFE FFFF	Reserved	Reserved		
H'FFFF 0000 to H'FFFF BFFF	On-chip RAM	On-chip RAM	48 kB	32 bits
H'FFFF C000 to H'FFFF FFFF	On-chip peripheral module	On-chip peripheral module	16 kB	8, 16 bits

Notes: 1. Selected by on-chip register (BCR1) settings.

2. Selected by the mode pin.

Do not access reserved spaces. Operation cannot be guaranteed if they are accessed.

- Number of Access Cycles for On-Chip Peripheral Module Registers

Module Name	Bus Width	Number of Access Cycles	
		Multiplication Ratio of 4	Multiplication Ratio of 8
ROM	8	Byte: 4	Byte: 4
UBC, WDT, BSC, DMAC, and INTC	16	Byte and word: 4, longword: 8	Byte and word: 4, longword: 8
SCI	8	Byte: 4 or 5, word: 8 or 9	Byte: 8 to 11, word: 16 to 19
ATU, APC, CMT, PFC, I/O ports, H-UDI, CPG, and power-down	16	Byte and word: 4 or 5, longword: 8 or 9	Byte and word: 8 to 11, longword: 16 to 19
AD and MTAD	8	Byte: 6 or 7, word: 12 or 13	Byte: 12 to 15, word: 24 to 27
HCAN	16	Byte and word: 6 or 7 + wait, longword: 12 or 13 + wait	Byte and word: 12 to 15 + wait, longword: 24 to 27 + wait

## 9.2 Description of Registers

### 9.2.1 Bus Control Register 1 (BCR1)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	A3SZ	A2SZ	A1SZ	A0SZ
Initial value:	0	0	0	0	1	1	1	1
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

BCR1 is a 16-bit readable/writable register that specifies the bus size of the CS spaces.

Write bits 15–0 of BCR1 during the initialization stage after a power-on reset, and do not change the values thereafter. In on-chip ROM enabled mode, do not access any of the CS spaces until after completion of register initialization. In on-chip ROM disabled mode, do not access any CS space other than CS0 until after completion of register initialization.

BCR1 is initialized to H'000F by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

- Bits 15–4—Reserved: The write value should always be 0. Operation cannot be guaranteed if 1 is written to these bits.
- Bit 3—CS3 Space Size Specification (A3SZ): Specifies the CS3 space bus size. A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 3: A3SZ	Description
0	Byte (8-bit) size
1	Word (16-bit) size (Initial value)

- Bit 2—CS2 Space Size Specification (A2SZ): Specifies the CS2 space bus size. A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 2: A2SZ	Description
0	Byte (8-bit) size
1	Word (16-bit) size (Initial value)

- Bit 1—CS1 Space Size Specification (A1SZ): Specifies the CS1 space bus size. A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 1: A1SZ	Description
0	Byte (8-bit) size
1	Word (16-bit) size (Initial value)

- Bit 0—CS0 Space Size Specification (A0SZ): Specifies the CS0 space bus size. A 0 setting specifies byte (8-bit) size, and a 1 setting specifies word (16-bit) size.

Bit 0: A0SZ	Description
0	Byte (8-bit) size
1	Word (16-bit) size (Initial value)

Note: A0SZ is valid only in on-chip ROM enabled mode. In on-chip ROM disabled mode, the CS0 space bus size is specified by the mode pin.

## 9.2.2 Bus Control Register 2 (BCR2)

Bit:	15	14	13	12	11	10	9	8
	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	7	6	5	4	3	2	1	0
	CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCR2 is a 16-bit readable/writable register that specifies the number of idle cycles and  $\overline{CS}$  signal assert extension of each CS space.

BCR2 is initialized to H'FFFF by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

- Bits 15–8—Idles between Cycles (IW31, IW30, IW21, IW20, IW11, IW10, IW01, IW00):  
These bits specify idle cycles inserted between consecutive accesses when the second one is to a different CS area after a read. Idles are used to prevent data conflict between ROM (and other memories, which are slow to turn the read data buffer off), fast memories, and I/O interfaces. Even when access is to the same area, idle cycles must be inserted when a read access is followed immediately by a write access. The idle cycles to be inserted comply with the area specification of the previous access. Refer to section 9.4, Waits between Access Cycles, for details.  
IW31, IW30 specify the idle between cycles for CS3 space; IW21, IW20 specify the idle between cycles for CS2 space; IW11, IW10 specify the idle between cycles for CS1 space and IW01, IW00 specify the idle between cycles for CS0 space.

Bit 15: IW31	Bit 14: IW30	Description
0	0	No CS3 space idle cycle
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (Initial value)

Bit 13: IW21	Bit 12: IW20	Description
0	0	No CS2 space idle cycle
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (Initial value)

Bit 11: IW11	Bit 10: IW10	Description
0	0	No CS1 space idle cycle
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (Initial value)

Bit 9: IW01	Bit 8: IW00	Description
0	0	No CS0 space idle cycle
	1	Inserts one idle cycle
1	0	Inserts two idle cycles
	1	Inserts three idle cycles (Initial value)

- Bits 7–4—Idle Specification for Continuous Access (CW3, CW2, CW1, CW0): The continuous access idle specification makes insertions to clearly delineate the bus intervals by once negating the CS<sub>n</sub> signal when performing consecutive accesses to the same CS space. When a write immediately follows a read, the number of idle cycles inserted is the larger of the two values specified by IW and CW. Refer to section 9.4, Waits between Access Cycles, for details.

CW3 specifies the continuous access idles for CS3 space; CW2 specifies the continuous access idles for CS2 space; CW1 specifies the continuous access idles for CS1 space and CW0 specifies the continuous access idles for CS0 space.

Bit 7: CW3	Description
0	No CS3 space continuous access idle cycles
1	One CS3 space continuous access idle cycle (Initial value)

Bit 6: CW2	Description
0	No CS2 space continuous access idle cycles
1	One CS2 space continuous access idle cycle (Initial value)

Bit 5: CW1	Description
0	No CS1 space continuous access idle cycles
1	One CS1 space continuous access idle cycle (Initial value)

Bit 4: CW0	Description
0	No CS0 space continuous access idle cycles
1	One CS0 space continuous access idle cycle (Initial value)

- Bits 3–0— $\overline{CS}$  Assert Extension Specification (SW3, SW2, SW1, SW0): The  $\overline{CS}$  assert cycle extension specification is for making insertions to prevent extension of the  $\overline{RD}$  signal,  $\overline{WRH}$  signal, or  $\overline{WRL}$  signal assert period beyond the length of the  $\overline{CS}_n$  signal assert period. Extended cycles insert one cycle before and after each bus cycle, which simplifies interfaces with external devices and also has the effect of extending the write data hold time. Refer to section 9.3.3,  $\overline{CS}$  Assert Period Extension, for details.  
SW3 specifies the  $\overline{CS}$  assert extension for CS3 space access; SW2 specifies the  $\overline{CS}$  assert extension for CS2 space access; SW1 specifies the  $\overline{CS}$  assert extension for CS1 space access and SW0 specifies the  $\overline{CS}$  assert extension for CS0 space access.

Bit 3: SW3	Description
0	No CS3 space $\overline{CS}$ assert extension
1	CS3 space $\overline{CS}$ assert extension (Initial value)

Bit 2: SW2	Description
0	No CS2 space $\overline{CS}$ assert extension
1	CS2 space $\overline{CS}$ assert extension (Initial value)

Bit 1: SW1	Description
0	No CS1 space $\overline{CS}$ assert extension
1	CS1 space $\overline{CS}$ assert extension (Initial value)

Bit 0: SW0	Description
0	No CS0 space $\overline{CS}$ assert extension
1	CS0 space $\overline{CS}$ assert extension (Initial value)

### 9.2.3 Wait Control Register (WCR)

Bit:	15	14	13	12	11	10	9	8
	—	W32	W31	W30	—	W22	W21	W20
Initial value:	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	W12	W11	W10	—	W02	W01	W00
Initial value:	0	1	1	1	0	1	1	1
R/W:	R	R/W	R/W	R/W	R	R/W	R/W	R/W

WCR is a 16-bit readable/writable register that specifies the number of wait cycles for each CS space.

WCR is initialized to H'7777 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

- Bit 15—Reserved
- Bits 14–12—CS3 Space Wait Specification (W32, W31, W30): These bits specify the number of waits for CS3 space access.

Bit 14: W32	Bit 13: W31	Bit 12: W30	Description
0	0	0	No wait (external wait input disabled)
0	0	1	1 wait external wait input enabled
...			
1	1	1	7 wait external wait input enabled (Initial value)

- Bit 11—Reserved
- Bits 10–8—CS2 Space Wait Specification (W22, W21, W20): These bits specify the number of waits for CS2 space access.

Bit 10: W22	Bit 9: W21	Bit 8: W20	Description
0	0	0	No wait (external wait input disabled)
0	0	1	1 wait external wait input enabled
...			
1	1	1	7 wait external wait input enabled (Initial value)



- Bit 7—Reserved
- Bits 6–4—CS1 Space Wait Specification (W12, W11, W10): These bits specify the number of waits for CS1 space access.

Bit 6: W12	Bit 5: W11	Bit 4: W10	Description
0	0	0	No wait (external wait input disabled)
0	0	1	1 wait external wait input enabled
...			
1	1	1	7 wait external wait input enabled (Initial value)

- Bit 3—Reserved
- Bits 2–0—CS0 Space Wait Specification (W02, W01, W00): These bits specify the number of waits for CS0 space access.

Bit 2: W02	Bit 1: W01	Bit 0: W00	Description
0	0	0	No wait (external wait input disabled)
0	0	1	1 wait external wait input enabled
...			
1	1	1	7 wait external wait input enabled (Initial value)

#### 9.2.4 RAM Emulation Register (RAMER)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

The RAM emulation register (RAMER) is a 16-bit readable/writable register that selects the RAM area to be used when emulating realtime programming of flash memory.

RAMER is initialized to H'0000 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

Note: To ensure correct operation of the RAM emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Operation cannot be guaranteed if such an access is made.

- Bits 15 to 4—Reserved: Only 0 should be written to these bits. Operation cannot be guaranteed if 1 is written.
- Bit 3—RAM Select (RAMS): Used together with bits 2 to 0 to select or deselect flash memory emulation by RAM (table 9.4).

When 1 is written to this bit, all flash memory blocks are write/erase-protected.

This bit is ignored in modes with on-chip ROM disabled.

- Bits 2 to 0—RAM Area Specification (RAM2 to RAM0): These bits are used together with the RAMS bit to designate the flash memory area to be overlapped onto RAM (table 9.4).

**Table 9.4 RAM Area Setting Method**

RAM Area	Bit 3: RAMS	Bit 2: RAM2	Bit 1: RAM1	Bit 0: RAM0
H'FFFF0000 to H'FFFF0FFF	0	*	*	*
H'00000000 to H'00000FFF	1	0	0	0
H'00001000 to H'00001FFF	1	0	0	1
H'00002000 to H'00002FFF	1	0	1	0
H'00003000 to H'00003FFF	1	0	1	1
H'00004000 to H'00004FFF	1	1	0	0
H'00005000 to H'00005FFF	1	1	0	1
H'00006000 to H'00006FFF	1	1	1	0
H'00007000 to H'00007FFF	1	1	1	1

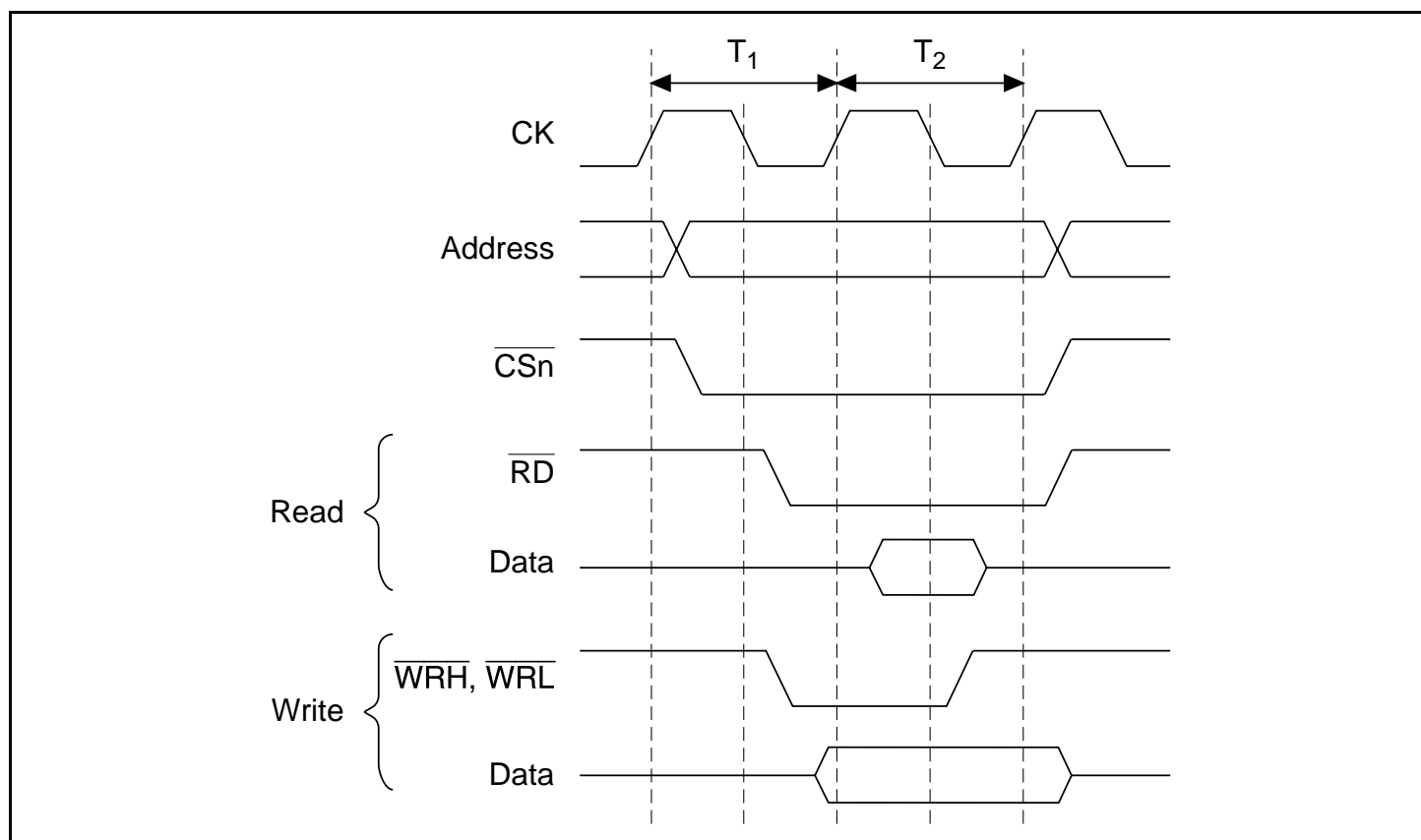
\*: Don't care

## 9.3 Accessing External Space

A strobe signal is output in external space accesses to provide primarily for SRAM or ROM direct connections.

### 9.3.1 Basic Timing

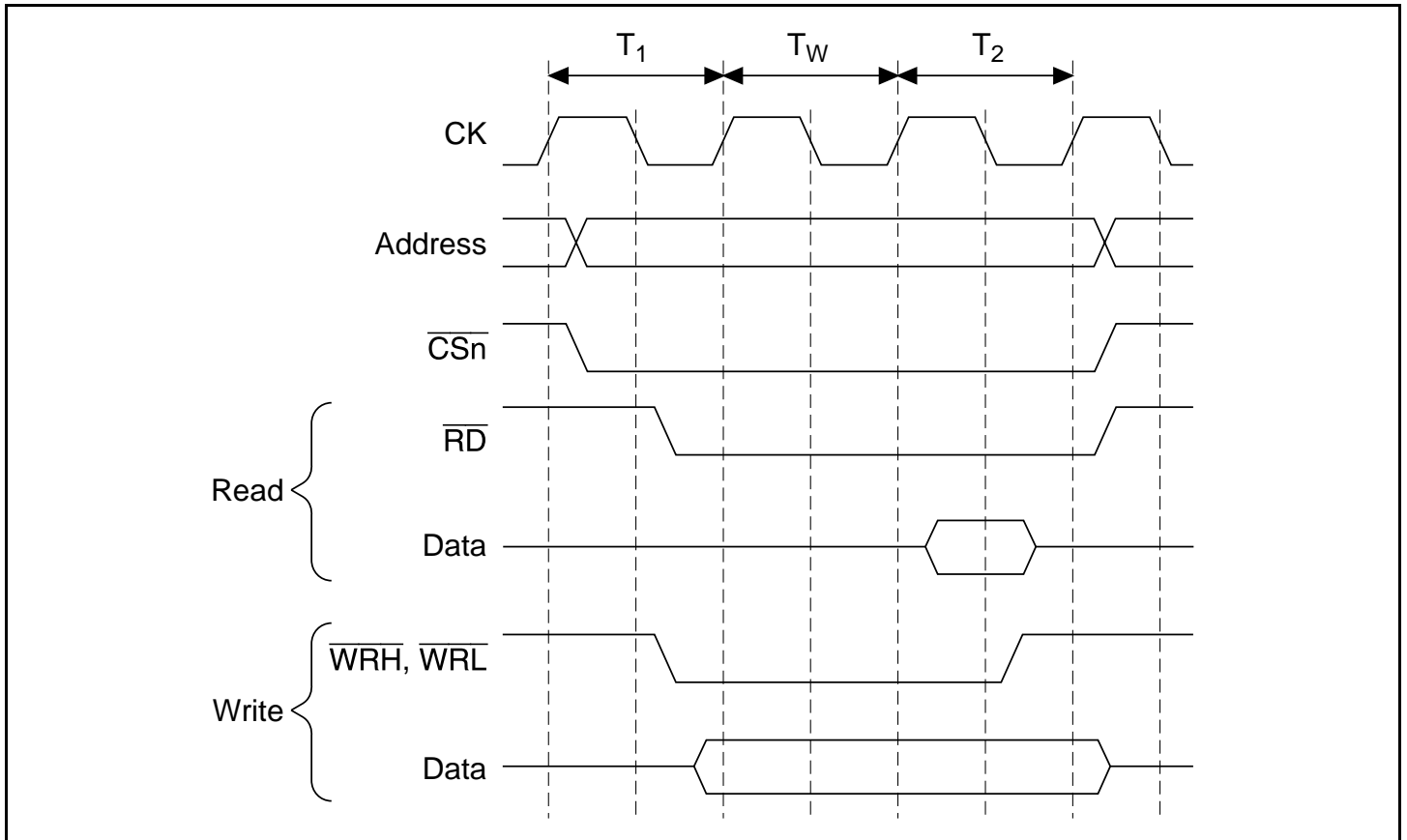
Figure 9.3 shows the basic timing of external space access. External access bus cycles are performed in 2 states.



**Figure 9.3 Basic Timing of External Space Access**

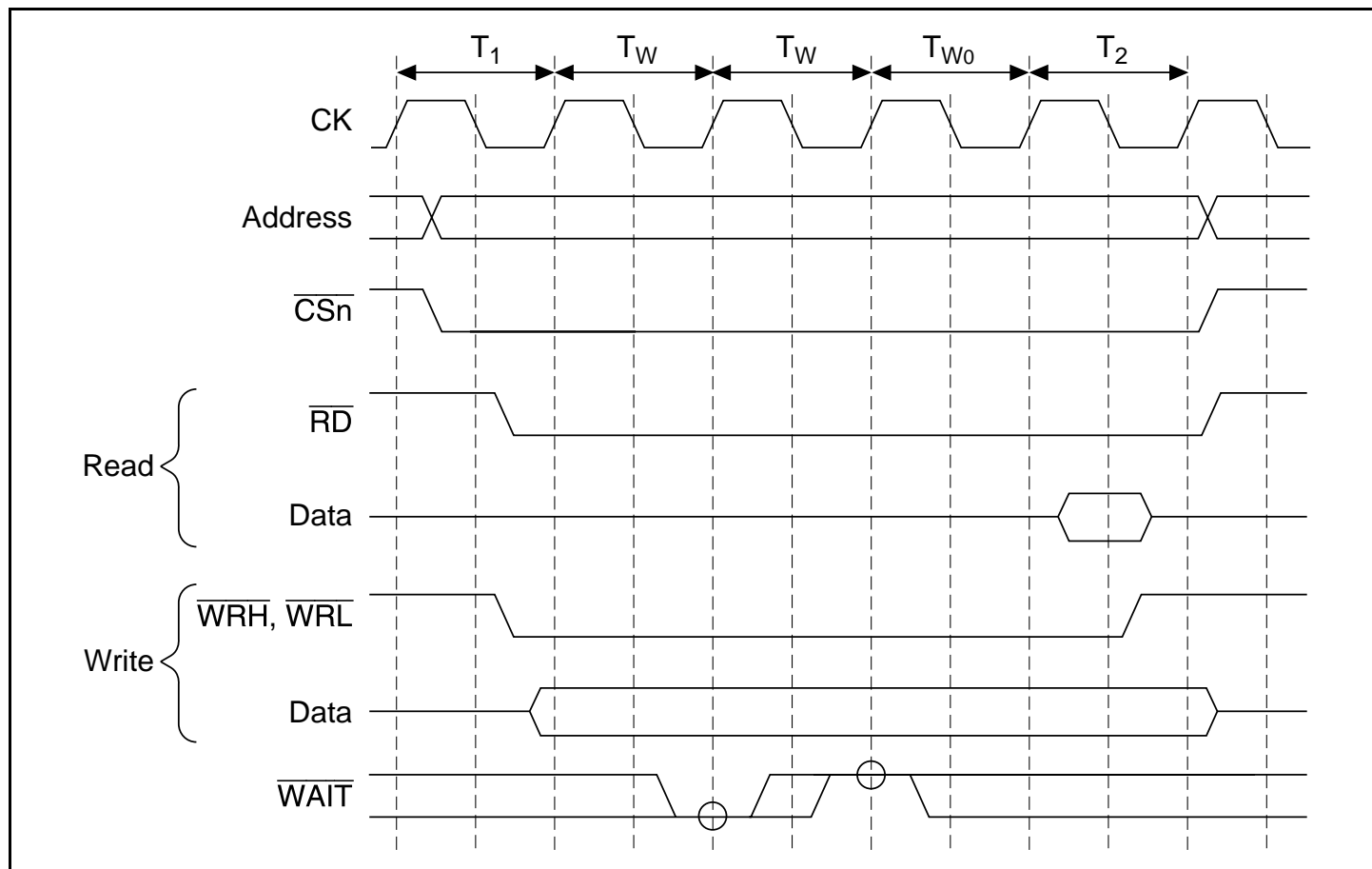
### 9.3.2 Wait State Control

The number of wait states inserted into external space access states can be controlled using the WCR settings (figure 9.4). The specified number of  $T_w$  cycles are inserted as software cycles at the timing shown in figure 9.4.



**Figure 9.4 Wait State Timing of External Space Access (Software Wait Only)**

When the wait is specified by software using WCR, the wait input  $\overline{\text{WAIT}}$  signal from outside is sampled. Figure 9.5 shows the  $\overline{\text{WAIT}}$  signal sampling. The  $\overline{\text{WAIT}}$  signal is sampled at the clock rise one cycle before the clock rise when the  $T_w$  state shifts to the  $T_2$  state. When using external waits, use a WCR setting of 1 state or more when extending  $\overline{\text{CS}}$  assertion, and 2 states or more otherwise.



**Figure 9.5 Wait State Timing of External Space Access (Two Software Wait States +  $\overline{\text{WAIT}}$  Signal Wait State)**

### 9.3.3 $\overline{\text{CS}}$ Assert Period Extension

Idle cycles can be inserted to prevent extension of the  $\overline{\text{RD}}$ ,  $\overline{\text{WRH}}$ , or  $\overline{\text{WRL}}$  signal assert period beyond the length of the  $\overline{\text{CSn}}$  signal assert period by setting the SW3–SW0 bits of BCR2. This allows for flexible interfaces with external circuitry. The timing is shown in figure 9.6.  $T_h$  and  $T_f$  cycles are added respectively before and after the ordinary cycle. Only  $\overline{\text{CSn}}$  is asserted in these cycles;  $\overline{\text{RD}}$ ,  $\overline{\text{WRH}}$ , and  $\overline{\text{WRL}}$  signals are not. Further, data is extended up to the  $T_f$  cycle, which is effective for gate arrays and the like, which have slower write operations.

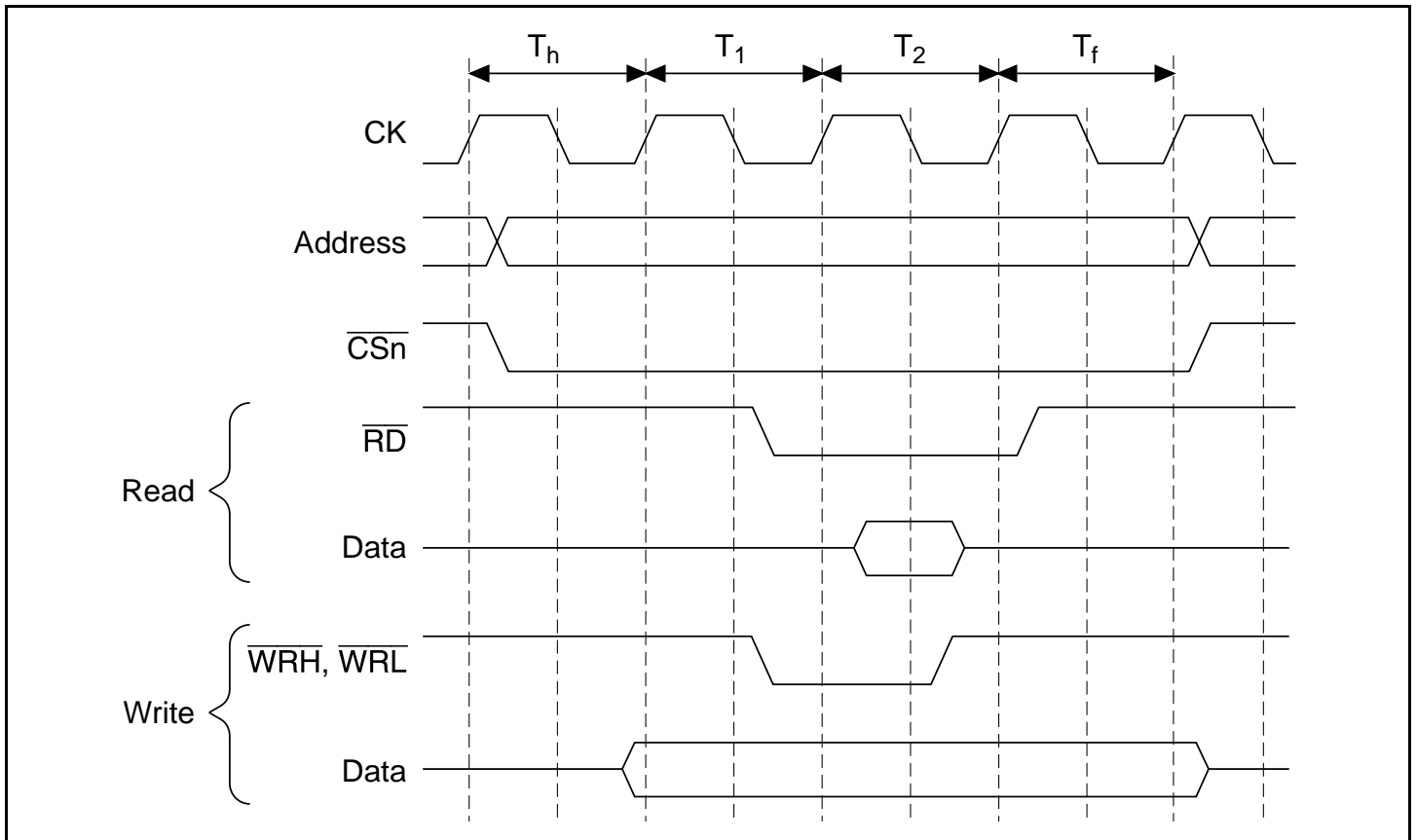


Figure 9.6  $\overline{\text{CS}}$  Assert Period Extension Function

## 9.4 Waits between Access Cycles

When a read from a slow device is completed, data buffers may not go off in time to prevent data conflicts with the next access. If there is a data conflict during memory access, the problem can be solved by inserting a wait in the access cycle.

To enable detection of bus cycle starts, waits can be inserted between access cycles during continuous accesses of the same CS space by negating the  $\overline{CSn}$  signal once.

### 9.4.1 Prevention of Data Bus Conflicts

For the two cases of write cycles after read cycles, and read cycles for a different area after read cycles, waits are inserted so that the number of idle cycles specified by the IW31 to IW00 bits of BCR2 occur. When idle cycles already exist between access cycles, only the number of empty cycles remaining beyond the specified number of idle cycles are inserted.

Figure 9.7 shows an example of idles between cycles. In this example, one idle between CSn space cycles has been specified, so when a CSm space write immediately follows a CSn space read cycle, one idle cycle is inserted.

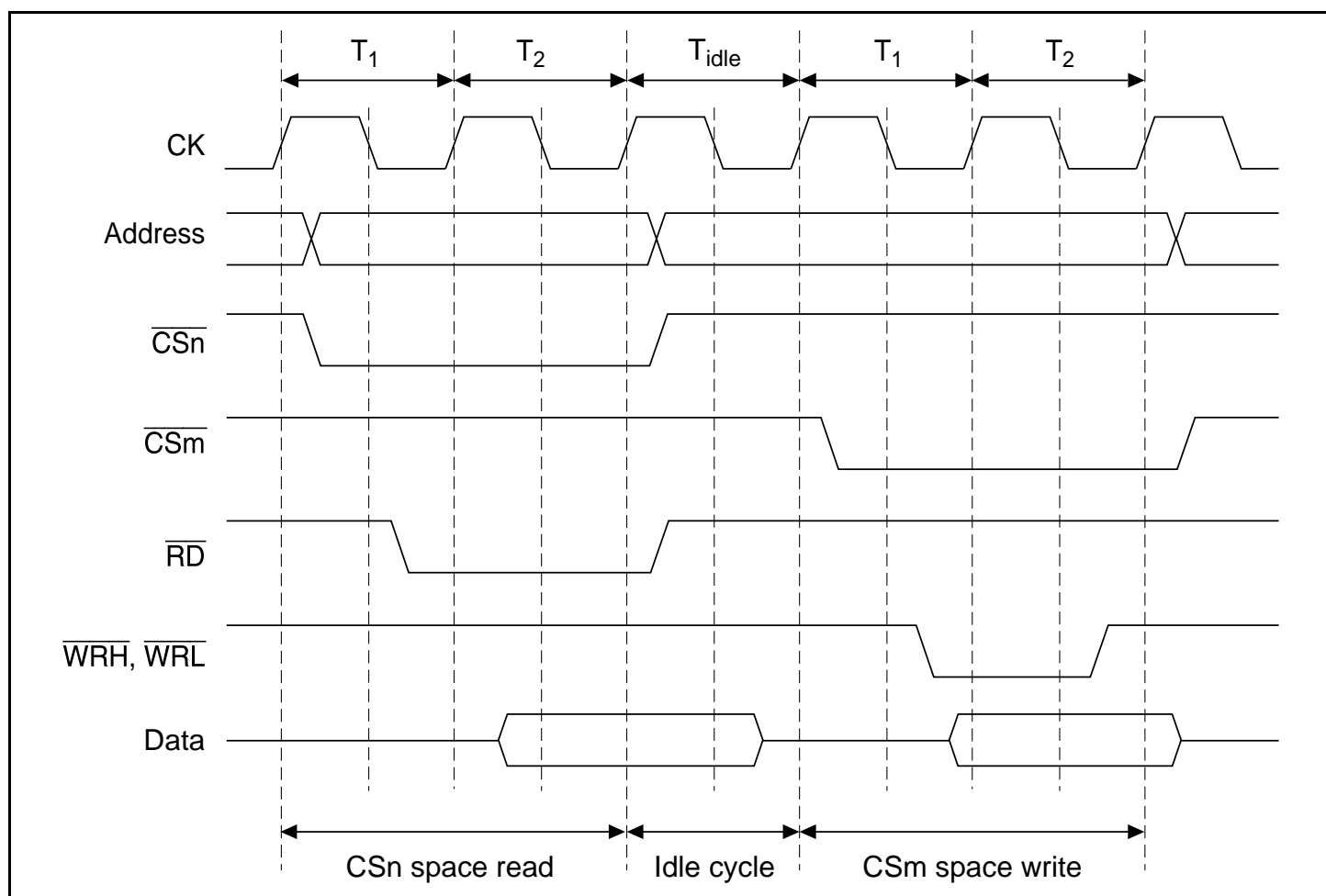
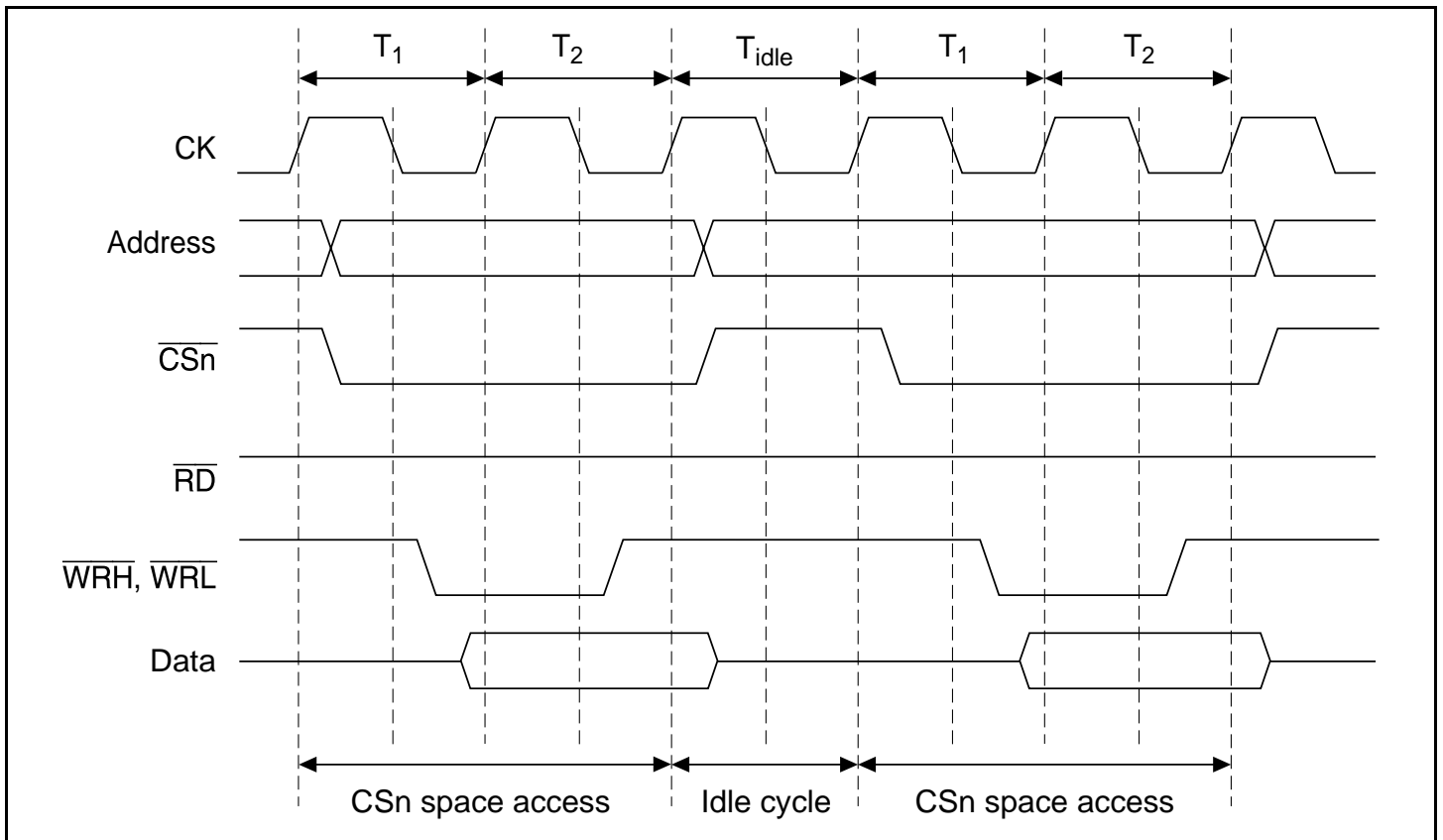


Figure 9.7 Idle Cycle Insertion Example

IW31 and IW30 specify the number of idle cycles required after a CS3 space read either to read other external spaces, or for this chip, to perform write accesses. In the same manner, IW21 and IW20 specify the number of idle cycles after a CS2 space read, IW11 and IW10, the number after a CS1 space read, and IW01 and IW00, the number after a CS0 space read. 0 to 3 idle cycles can be specified.

### 9.4.2 Simplification of Bus Cycle Start Detection

For consecutive accesses to the same CS space, waits are inserted to provide the number of idle cycles designated by bits CW3 to CW0 in BCR2. However, in the case of a write cycle after a read, the number of idle cycles inserted will be the larger of the two values designated by the IW and CW bits. When idle cycles already exist between access cycles, waits are not inserted. Figure 9.8 shows an example. A continuous access idle is specified for CSn space, and CSn space is consecutively write-accessed.



**Figure 9.8 Same Space Consecutive Access Idle Cycle Insertion Example**



## 9.5 Bus Arbitration

The SH7058 has a bus arbitration function that, when a bus release request is received from an external device, releases the bus to that device. It also has three internal bus masters, the CPU, DMAC, and AUD. The priority ranking for determining bus right transfer between these bus masters is:

Bus right request from external device > AUD > DMAC > CPU

Therefore, an external device that generates a bus request is given priority even if the request is made during a DMAC burst transfer.

The AUD does not acquire the bus during DMAC burst transfer, but at the end of the transfer. When the CPU has possession of the bus, the AUD has higher priority than the DMAC for bus acquisition.

A bus request by an external device should be input at the  $\overline{\text{BREQ}}$  pin. The signal indicating that the bus has been released is output from the  $\overline{\text{BACK}}$  pin.

Figure 9.9 shows the bus right release procedure.

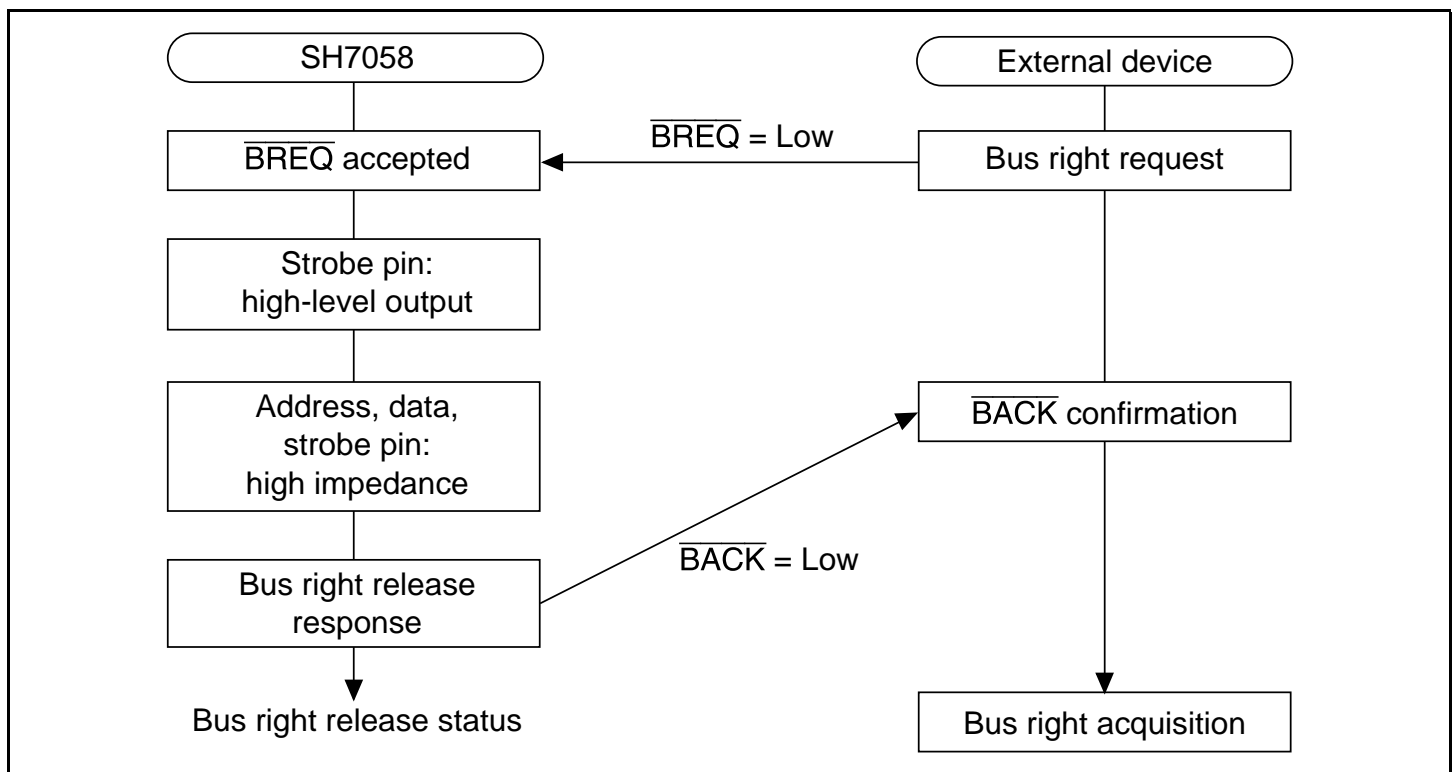
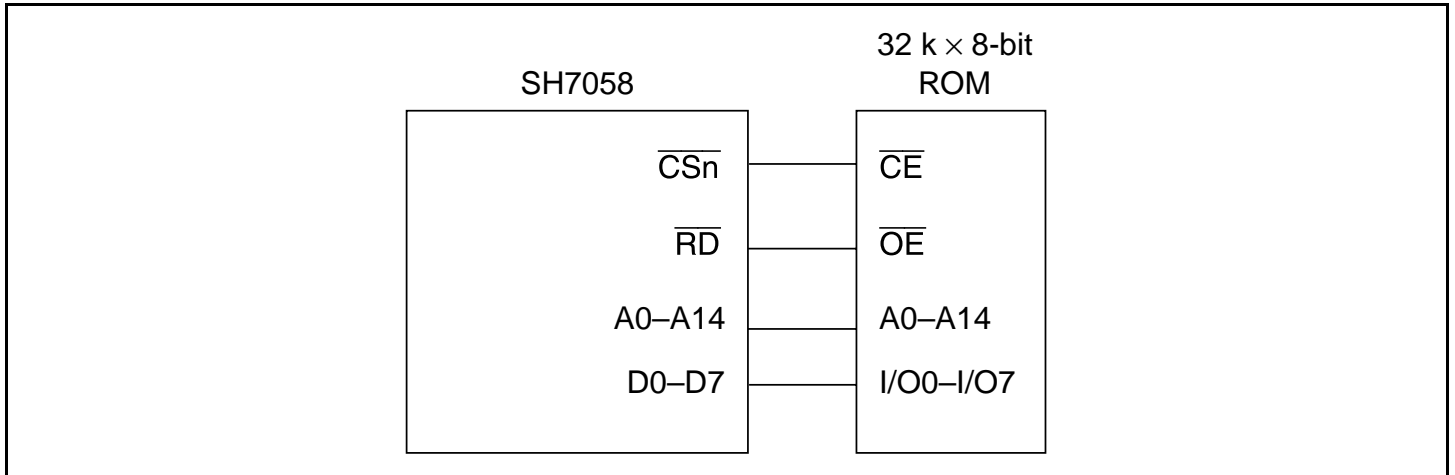


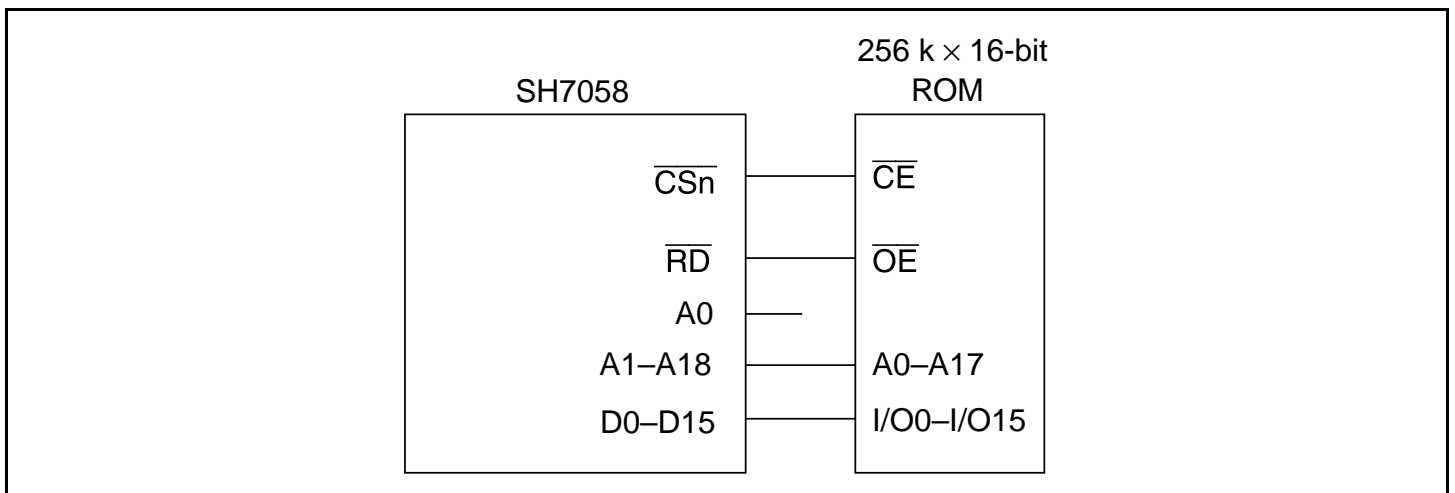
Figure 9.9 Bus Right Release Procedure

## 9.6 Memory Connection Examples

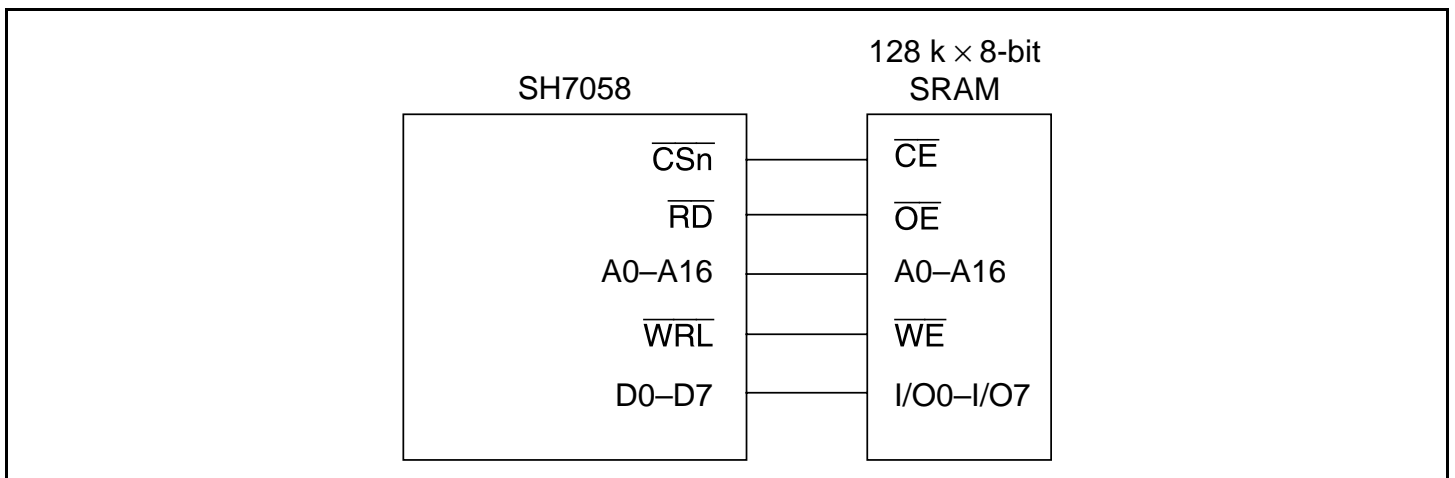
Figures 9.10–9.13 show examples of the memory connections.



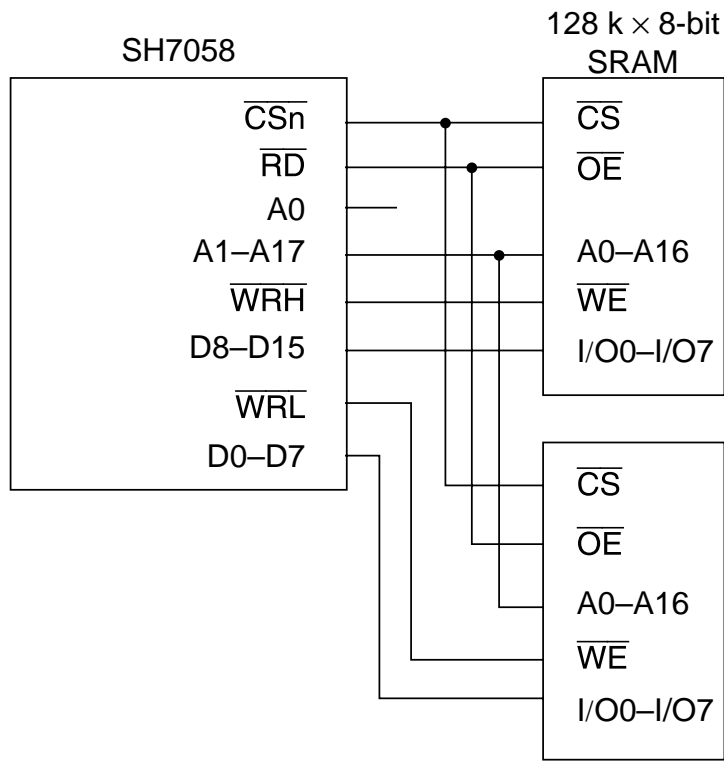
**Figure 9.10 Example of 8-Bit Data Bus Width ROM Connection**



**Figure 9.11 Example of 16-Bit Data Bus Width ROM Connection**



**Figure 9.12 Example of 8-Bit Data Bus Width SRAM Connection**



**Figure 9.13 Example of 16-Bit Data Bus Width SRAM Connection**

# Section 10 Direct Memory Access Controller (DMAC)

## 10.1 Overview

The SH7058 includes an on-chip four-channel direct memory access controller (DMAC). The DMAC can be used in place of the CPU to perform high-speed data transfers among external memories, memory-mapped external devices, and on-chip peripheral modules (except for the DMAC, BSC, and UBC). Using the DMAC reduces the burden on the CPU and increases the operating efficiency of the chip as a whole.

### 10.1.1 Features

The DMAC has the following features:

- Four channels
- 4-Gbyte address space in the architecture
- 8-, 16-, or 32-bit selectable data transfer length
- Maximum of 16 M (16,777,216) transfers
- Address modes

Both the transfer source and transfer destination are accessed by address. There are two transfer modes: direct address and indirect address.

— Direct address transfer mode: Values set in a DMAC internal register indicate the accessed address for both the transfer source and transfer destination. Two bus cycles are required for one data transfer.

— Indirect address transfer mode: The value stored at the location pointed to by the address set in the DMAC internal transfer source register is used as the address. Operation is otherwise the same as for direct access. This function can only be set for channel 3. Four bus cycles are required for one data transfer.

- Channel function: Dual address mode is supported on all channels.

Channel 2 has a source address reload function that reloads the source address every fourth transfer. Direct address transfer mode or indirect address transfer mode can be specified for channel 3.

- Reload function

Enables automatic reloading of the value set in the first source address register every fourth DMA transfer. This function can be executed on channel 2 only.

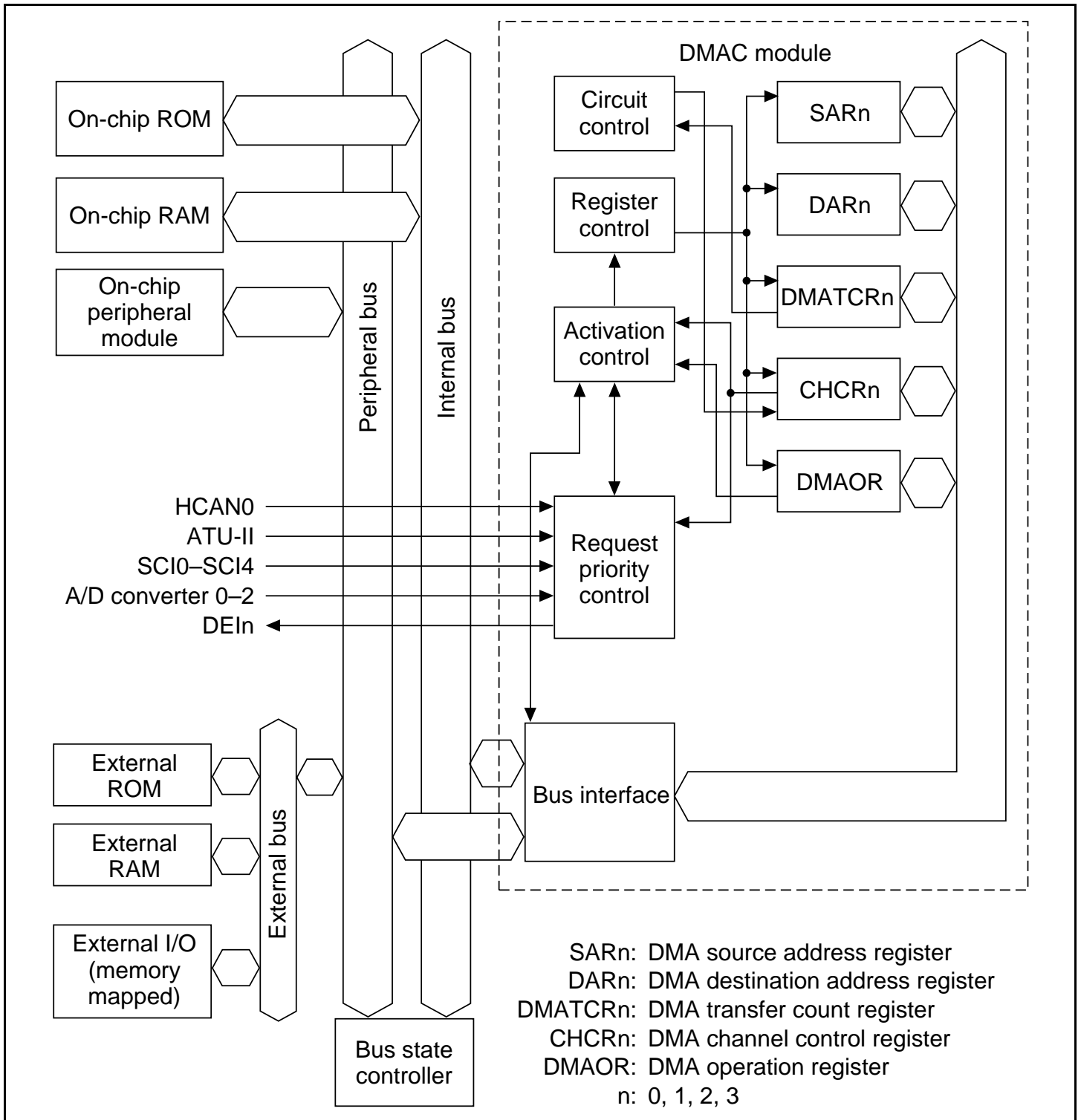
- Transfer requests

There are two DMAC transfer activation requests, as indicated below.

- Requests from on-chip peripheral modules: Transfer requests from on-chip modules such as the SCI or A/D. These can be received by all channels.
- Auto-request: The transfer request is generated automatically within the DMAC.
- Selectable bus modes: Cycle-steal mode or burst mode
- Fixed DMAC channel priority ranking
- CPU can be interrupted when the specified number of data transfers are complete.

## 10.1.2 Block Diagram

Figure 10.1 is a block diagram of the DMAC.



**Figure 10.1 DMAC Block Diagram**

### 10.1.3 Register Configuration

Table 10.1 summarizes the DMAC registers. The DMAC has a total of 17 registers. Each channel has four registers, and one overall DMAC control register is shared by all channels.

**Table 10.1 DMAC Registers**

Channel	Name	Abbr.	R/W	Initial Value	Address	Register Size	Access Size
0	DMA source address register 0	SAR0	R/W	Undefined	H'FFFEECC0	32 bits	16, 32* <sup>2</sup>
	DMA destination address register 0	DAR0	R/W	Undefined	H'FFFEECC4	32 bits	16, 32* <sup>2</sup>
	DMA transfer count register 0	DMATCR0	R/W	Undefined	H'FFFEECC8	32 bits	16, 32* <sup>2</sup>
	DMA channel control register 0	CHCR0	R/W* <sup>1</sup>	H'00000000	H'FFFEECCC	32 bits	16, 32* <sup>2</sup>
1	DMA source address register 1	SAR1	R/W	Undefined	H'FFFEECD0	32 bits	16, 32* <sup>2</sup>
	DMA destination address register 1	DAR1	R/W	Undefined	H'FFFEECD4	32 bits	16, 32* <sup>2</sup>
	DMA transfer count register 1	DMATCR1	R/W	Undefined	H'FFFEECD8	32 bits	16, 32* <sup>3</sup>
	DMA channel control register 1	CHCR1	R/W* <sup>1</sup>	H'00000000	H'FFFEECDC	32 bits	16, 32* <sup>2</sup>
2	DMA source address register 2	SAR2	R/W	Undefined	H'FFFEECE0	32 bits	16, 32* <sup>2</sup>
	DMA destination address register 2	DAR2	R/W	Undefined	H'FFFEECE4	32 bits	16, 32* <sup>2</sup>
	DMA transfer count register 2	DMATCR2	R/W	Undefined	H'FFFEECE8	32 bits	16, 32* <sup>3</sup>
	DMA channel control register 2	CHCR2	R/W* <sup>1</sup>	H'00000000	H'FFFEECEC	32 bits	16, 32* <sup>2</sup>

**Table 10.1 DMAC Registers (cont)**

Channel	Name	Abbr.	R/W	Initial Value	Address	Register Size	Access Size
3	DMA source address register 3	SAR3	R/W	Undefined	H'FFFFFFCF0	32 bits	16, 32* <sup>2</sup>
	DMA destination address register 3	DAR3	R/W	Undefined	H'FFFFFFCF4	32 bits	16, 32* <sup>2</sup>
	DMA transfer count register 3	DMATCR3	R/W	Undefined	H'FFFFFFCF8	32 bits	16, 32* <sup>3</sup>
	DMA channel control register 3	CHCR3	R/W* <sup>1</sup>	H'00000000	H'FFFFFFCFC	32 bits	16, 32* <sup>2</sup>
Shared	DMA operation register	DMAOR	R/W* <sup>1</sup>	H'0000	H'FFFFFFCB0	16 bits	16* <sup>4</sup>

- Notes:
- Word access to a register takes four cycles, and longword access eight cycles. Do not attempt to access an empty address, as operation cannot be guaranteed if this is done.
  - 1. Write 0 after reading 1 in bit 1 of CHCR0–CHCR3 and in bits 1 and 2 of DMAOR to clear flags. No other writes are allowed.
  - 2. For 16-bit access of SAR0–SAR3, DAR0–DAR3, and CHCR0–CHCR3, the 16-bit value on the side not accessed is held.
  - 3. DMATCR has a 24-bit configuration: bits 0–23. Writing to the upper 8 bits (bits 24–31) is invalid, and these bits always read 0.
  - 4. Do not use 32-bit access on DMAOR.

## 10.2 Register Descriptions

### 10.2.1 DMA Source Address Registers 0–3 (SAR0–SAR3)

Bit:	31	30	29	28	27	26	25	24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	23	22	21	...	...	2	1	0
Initial value:	—	—	—	...	...	—	—	—
R/W:	R/W	R/W	R/W	...	...	R/W	R/W	R/W



DMA source address registers 0–3 (SAR0–SAR3) are 32-bit readable/writable registers that specify the source address of a DMA transfer. These registers have a count function, and during a DMA transfer, they indicate the next source address.

Specify a 16-bit boundary when performing 16-bit data transfers, and a 32-bit boundary when performing 32-bit data transfers. Operation cannot be guaranteed if any other addresses are set.

The initial value after a power-on reset and in standby mode is undefined.

### 10.2.2 DMA Destination Address Registers 0–3 (DAR0–DAR3)

Bit:	31	30	29	28	27	26	25	24
Initial value:	—	—	—	—	—	—	—	—
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit:	23	22	21	...	...	2	1	0
Initial value:	—	—	—	...	...	—	—	—
R/W:	R/W	R/W	R/W	...	...	R/W	R/W	R/W

DMA destination address registers 0–3 (DAR0–DAR3) are 32-bit readable/writable registers that specify the destination address of a DMA transfer. These registers have a count function, and during a DMA transfer, they indicate the next destination address.

Specify a 16-bit boundary when performing 16-bit data transfers, and a 32-bit boundary when performing 32-bit data transfers. Operation cannot be guaranteed if any other addresses are set.

The value after a power-on reset and in standby mode is undefined.

### 10.2.3 DMA Transfer Count Registers 0–3 (DMATCR0–DMATCR3)

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

DMA transfer count registers 0–3 (DMATCR0–DMATCR3) are 24-bit read/write registers that specify the transfer count for the channel (byte count, word count, or longword count) in bits 23 to 0. Specifying H'000001 gives a transfer count of 1, while H'000000 gives the maximum setting, 16,777,216 transfers. During DMAC operation, these registers indicate the remaining number of transfers.

The upper 8 bits of DMATCR always read 0. The write value, also, should always be 0.

The value after a power-on reset and in standby mode is undefined.

## 10.2.4 DMA Channel Control Registers 0–3 (CHCR0–CHCR3)

Bit:	31	30	29	28	27	26	25	24
	—	—	—	DI	—	—	—	RO
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W* <sup>2</sup>	R	R	R	R/W* <sup>2</sup>

Bit:	23	22	21	20	19	18	17	16
	—	—	—	RS4	RS3	RS2	RS1	RS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W* <sup>1</sup>	R/W

Bit:	15	14	13	12	11	10	9	8
	—	—	SM1	SM0	—	—	DM1	DM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	TS1	TS0	TM	IE	TE	DE
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/(W)* <sup>1</sup>	R/W

- Notes: 1. TE bit: Allows only a 0 write after reading 1.  
 2. The DI and RO bits may be absent, depending on the channel.

DMA channel control registers 0–3 (CHCR0–CHCR3) are 32-bit readable/writable registers that designate the operation and transmission of each channel. CHCR register bits are initialized to H'00000000 by a power-on reset and in standby mode.

- Bits 31–29, 27–25, 23–21, 15, 14, 11, 10, 7, 6—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 28—Direct/Indirect Select (DI): Specifies either direct address mode operation or indirect address mode operation for the channel 3 source address. This bit is valid only in CHCR3. This bit is always read as 0 in CHCR0–CHCR2, and the write value should always be 0.

Bit 28: DI	Description	
0	Direct access mode operation for channel 3	(Initial value)
1	Indirect access mode operation for channel 3	

- Bit 24—Source Address Reload (RO): Selects whether to reload the source address initial value during channel 2 transfer. This bit is valid only for channel 2. This bit is always read as 0 in CHCR0, CHCR1, and CHCR3, and the write value should always be 0.

Bit 24: RO	Description	
0	Does not reload source address	(Initial value)
1	Reloads source address	

- Bits 20–16—Resource Select 4–0 (RS4–RS0): These bits specify the transfer request source.

Bit 20: RS4	Bit 19: RS3	Bit 18: RS2	Bit 17: RS1	Bit 16: RS0	Description
0	0	0	0	0	No request* (Initial value)
				1	SCI0 transmission
				0	SCI0 reception
				1	SCI1 transmission
				0	SCI1 reception
				1	SCI2 transmission
				0	SCI2 reception
				1	SCI3 transmission
				0	SCI3 reception
				1	SCI4 transmission
				0	SCI4 reception
				1	On-chip A/D0
				0	On-chip A/D1
				1	On-chip A/D2
				0	No request*
				1	HCAN0 (RM0)
1	0	0	0	0	No request*
				1	ATU-II (IC10A)
				0	ATU-II (IC10B)
				1	ATU-II (IC10C)
				0	ATU-II (IC10D)
				1	ATU-II (CMI6A)
				0	ATU-II (CMI6B)
				1	ATU-II (CMI6C)
				0	ATU-II (CMI6D)
				1	ATU-II (CMI7A)
				0	ATU-II (CMI7B)
				1	ATU-II (CMI7C)
				0	ATU-II (CMI7D)
				1	No request*
				0	No request*
				1	Auto-request

Note: \* Refer to no. 12 in section 10.5, Usage Notes.

- Bits 13 and 12—Source Address Mode 1, 0 (SM1, SM0): These bits specify increment/decrement of the DMA transfer source address.

Bit 13: SM1	Bit 12: SM0	Description
0	0	Source address fixed (Initial value)
0	1	Source address incremented (+1 during 8-bit transfer, +2 during 16-bit transfer, +4 during 32-bit transfer)
1	0	Source address decremented (−1 during 8-bit transfer, −2 during 16-bit transfer, −4 during 32-bit transfer)
1	1	Setting prohibited

When the transfer source is specified at an indirect address, specify in source address register 3 (SAR3) the actual storage address of the data to be transferred as the data storage address (indirect address).

During indirect address mode, SAR3 obeys the SM1/SM0 setting for increment/decrement. In this case, SAR3's increment/decrement is fixed at +4/−4 or 0, irrespective of the transfer data size specified by TS1 and TS0.

- Bits 9 and 8—Destination Address Mode 1, 0 (DM1, DM0): These bits specify increment/decrement of the DMA transfer source address.

Bit 9: DM1	Bit 8: DM0	Description
0	0	Destination address fixed (Initial value)
0	1	Destination address incremented (+1 during 8-bit transfer, +2 during 16-bit transfer, +4 during 32-bit transfer)
1	0	Destination address decremented (−1 during 8-bit transfer, −2 during 16-bit transfer, −4 during 32-bit transfer)
1	1	Setting prohibited

- Bits 5 and 4—Transfer Size 1, 0 (TS1, TS0): These bits specify the size of the data for transfer.

Bit 5: TS1	Bit 4: TS0	Description
0	0	Specifies byte size (8 bits) (Initial value)
0	1	Specifies word size (16 bits)
1	0	Specifies longword size (32 bits)
1	1	Setting prohibited

- Bit 3—Transfer Mode (TM): Specifies the bus mode for data transfer.

Bit 3: TM	Description
0	Cycle-steal mode (Initial value)
1	Burst mode

- Bit 2—Interrupt Enable (IE): When this bit is set to 1, interrupt requests are generated after the number of data transfers specified in DMATCR (when TE = 1).

Bit 2: IE	Description
0	Interrupt request not generated on completion of DMATCR-specified number of transfers (Initial value)
1	Interrupt request enabled on completion of DMATCR-specified number of transfers

- Bit 1—Transfer End (TE): This bit is set to 1 after the number of data transfers specified by DMATCR. At this time, if the IE bit is set to 1, an interrupt request is generated.

If data transfer ends before TE is set to 1 (for example, due to an NMI or address error, or clearing of the DE bit or DME bit of DMAOR) TE is not set to 1. With this bit set to 1, data transfer is disabled even if the DE bit is set to 1.

Bit 1: TE	Description
0	DMATCR-specified number of transfers not completed (Initial value) [Clearing condition] 0 write after TE = 1 read, power-on reset, standby mode
1	DMATCR-specified number of transfers completed

- Bit 0—DMAC Enable (DE): DE enables operation in the corresponding channel.

Bit 0: DE	Description
0	Operation of the corresponding channel disabled (Initial value)
1	Operation of the corresponding channel enabled

Transfer is initiated if this bit is set to 1 when auto-request is specified (RS4–RS0 settings). With an on-chip module request, when a transfer request occurs after this bit is set to 1, transfer is initiated. If this bit is cleared during a data transfer, transfer is suspended.

If the DE bit has been set, but TE = 1, then if the DME bit of DMAOR is 0, and the NMIF or AE bit of DMAOR is 1, the transfer enable state is not entered.

## 10.2.5 DMAC Operation Register (DMAOR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	AE	NMIF	DME
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/(W)*	R/(W)*	R/W

Note: \* Only a 0 write is valid after 1 is read at the AE and NMIF bits.

DMAOR is a 16-bit readable/writable register that controls the overall operation of the DMAC.

Register values are initialized to H'0000 by a power-on reset and in standby mode.

- Bits 15–3—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 2—Address Error Flag (AE): Indicates that an address error has occurred during DMA transfer. If this bit is set during a data transfer, transfers on all channels are suspended. The CPU cannot write a 1 to the AE bit. Clearing is effected by a 0 write after a 1 read.

Bit 2: AE	Description
0	No address error, DMA transfer enabled (Initial value) [Clearing condition] Write AE = 0 after reading AE = 1
1	Address error, DMA transfer disabled [Setting condition] Address error due to DMAC



- Bit 1—NMI Flag (NMIF): Indicates input of an NMI. This bit is set irrespective of whether the DMAC is operating or suspended. If this bit is set during a data transfer, transfers on all channels are suspended. The CPU is unable to write a 1 to the NMIF. Clearing is effected by a 0 write after a 1 read.

Bit 1: NMIF	Description	
0	No NMI interrupt, DMA transfer enabled [Clearing condition] Write NMIF = 0 after reading NMIF = 1	(Initial value)
1	NMI has occurred, DMC transfer disabled [Setting condition] NMI interrupt occurrence	

- Bit 0—DMAC Master Enable (DME): This bit enables activation of the entire DMAC. When the DME bit and DE bit of the CHCR register for the corresponding channel are set to 1, that channel is transfer-enabled. If this bit is cleared during a data transfer, transfers on all channels are suspended.

Even when the DME bit is set, when the TE bit of CHCR is 1, or its DE bit is 0, transfer is disabled if the NMIF or AE bit in DMAOR is set to 1.

Bit 0: DME	Description	
0	Operation disabled on all channels	(Initial value)
1	Operation enabled on all channels	

## 10.3 Operation

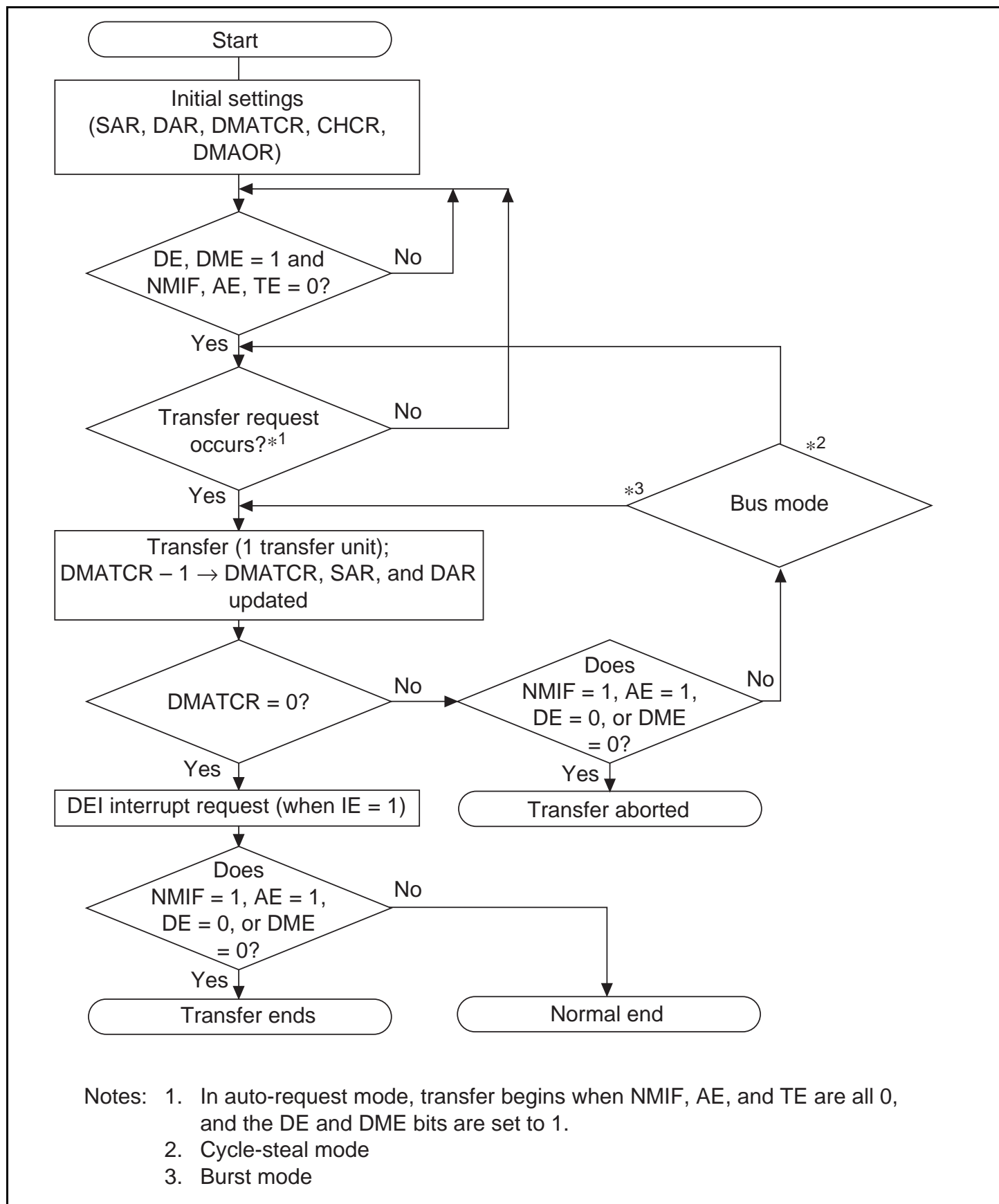
When there is a DMA transfer request, the DMAC starts the transfer according to the channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in two modes: auto-request and on-chip peripheral module request. Transfer is performed only in dual address mode, and either direct or indirect address transfer mode can be used. The bus mode can be either burst or cycle-steal.

### 10.3.1 DMA Transfer Flow

After the DMA source address registers (SAR), DMA destination address registers (DAR), DMA transfer count register (DMATCR), DMA channel control registers (CHCR), and DMA operation register (DMAOR) are set to the desired transfer conditions, the DMAC transfers data according to the following procedure:

1. The DMAC checks to see if transfer is enabled ( $DE = 1$ ,  $DME = 1$ ,  $TE = 0$ ,  $NMIF = 0$ ,  $AE = 0$ ).
2. When a transfer request comes and transfer has been enabled, the DMAC transfers 1 transfer unit of data (determined by the TS0 and TS1 setting). 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 upon each transfer. The actual transfer flows vary by address mode and bus mode.
3. When the specified number of transfers have been completed (when DMATCR reaches 0), the transfer ends normally. If the IE bit of CHCR is set to 1 at this time, a DEI interrupt is sent to the CPU.
4. When an address error occurs in the DMAC or an NMI interrupt is generated, the transfer is aborted. Transfer is also aborted when the DE bit of CHCR or the DME bit of DMAOR is cleared to 0.

Figure 10.2 is a flowchart of this procedure.



**Figure 10.2 DMAC Transfer Flowchart**

### 10.3.2 DMA Transfer Requests

DMA transfer requests are generated in either the data transfer source or destination. Transfers can be requested in two modes: auto-request and on-chip peripheral module request. The request mode is selected in the RS4–RS0 bits of DMA channel control registers 0–3 (CHCR0–CHCR3).

**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 of CHCR0–CHCR3 and the DME bit of DMAOR are set to 1, the transfer begins (so long as the TE bits of CHCR0–CHCR3 and the NMIF and AE bits of DMAOR are all 0).

**On-Chip Peripheral Module Request Mode:** In this mode a transfer is performed at the transfer request signal (interrupt request signal) of an on-chip peripheral module. As indicated in table 10.2, there are 26 transfer request signals: 12 from the advanced timer unit (ATU-II), which are compare match or input capture interrupts; the receive data full interrupts (RXI) and transmit data empty interrupts (TXI) of the five serial communication interfaces (SCI); the receive interrupt of HCAN0; and the A/D conversion end interrupts (ADI) of the three A/D converters. When DMA transfers are enabled (DE = 1, DME = 1, TE = 0, NMIF = 0, AE = 0), a transfer is performed upon the input of a transfer request signal.

When the transfer request is set to RXI (transfer request because the SCI's receive data register is full), the transfer source must be the SCI's receive data register (RDR). When the transfer request is set to TXI (transfer request because the SCI's transmit data register is empty), the transfer destination must be the SCI's transmit data register (TDR). If the transfer request is set to the A/D converter, the data transfer source must be the A/D converter register; if set to HCAN0, the transfer source must be HCAN0 message data.

In on-chip peripheral module request mode, when the DMAC accepts the transfer request, the next transfer request is ignored until a single transfer ends in cycle steal mode or all transfers end in burst mode. Only when the address reload function is used, the next transfer request is accepted after the fourth transfer.

**Table 10.2 Selecting On-Chip Peripheral Module Request Modes with the RS Bits**

RS4	RS3	RS2	RS1	RS0	DMAC Transfer Request Source	DMAC Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode
0	0	0	0	1	SCI0 transmit block	TXI0 (SCI0 transmit-data-empty transfer request)	Don't care*	TDR0	Burst/cycle-steal
			1	0	SCI0 receive block	RXI0 (SCI0 receive-data-full transfer request)	RDR0	Don't care*	Burst/cycle-steal
				1	SCI1 transmit block	TXI1 (SCI1 transmit-data-empty transfer request)	Don't care*	TDR1	Burst/cycle-steal
	1	0	0	0	SCI1 receive block	RXI1 (SCI1 receive-data-full transfer request)	RDR1	Don't care*	Burst/cycle-steal
				1	SCI2 transmit block	TXI2 (SCI2 transmit-data-empty transfer request)	Don't care*	TDR2	Burst/cycle-steal
			1	0	SCI2 receive block	RXI2 (SCI2 receive-data-full transfer request)	RDR2	Don't care*	Burst/cycle-steal
				1	SCI3 transmit block	TXI3 (SCI3 transmit-data-empty transfer request)	Don't care*	TDR3	Burst/cycle-steal
1	0	0	0	0	SCI3 receive block	RXI3 (SCI3 receive-data-full transfer request)	RDR3	Don't care*	Burst/cycle-steal
				1	SCI4 transmit block	TXI4 (SCI4 transmit-data-empty transfer request)	Don't care*	TDR4	Burst/cycle-steal
			1	0	SCI4 receive block	RXI4 (SCI4 receive-data-full transfer request)	RDR4	Don't care*	Burst/cycle-steal
				1	A/D0	ADI0 (A/D0 conversion end interrupt)	ADDR0–ADDR11	Don't care*	Burst/cycle-steal
	1	0	0	0	A/D1	ADI1 (A/D1 conversion end interrupt)	ADDR12–ADDR23	Don't care*	Burst/cycle-steal
				1	A/D2	ADI2 (A/D2 conversion end interrupt)	ADDR24–ADDR31	Don't care*	Burst/cycle-steal
			1	1	HCAN0	RM0 (HCAN0 receive interrupt)	MB0–MB15	Don't care*	Burst/cycle-steal

**Table 10.2 Selecting On-Chip Peripheral Module Request Modes with the RS Bits (cont)**

RS4	RS3	RS2	RS1	RS0	DMAC Transfer Request Source	DMAC Transfer Request Signal	Transfer Source	Transfer Destination	Bus Mode		
1	0	0	0	1	ATU-II	ICI0A (ICR0A input capture generation)	Don't care*	Don't care*	Burst/cycle-steal		
				1	0	ATU-II	ICI0B (ICR0B input capture generation)	Don't care*	Don't care*	Burst/cycle-steal	
				1	1	ATU-II	ICI0C (ICR0C input capture generation)	Don't care*	Don't care*	Burst/cycle-steal	
		1	0	0	0	ATU-II	ICI0D (ICR0D input capture generation)	Don't care*	Don't care*	Burst/cycle-steal	
					1	1	ATU-II	CMI6A (CYLR6A compare-match generation)	Don't care*	Don't care*	Burst/cycle-steal
					1	0	ATU-II	CMI6B (CYLR6B compare-match generation)	Don't care*	Don't care*	Burst/cycle-steal
	1	0	0	0	0	ATU-II	CMI6C (CYLR6C compare-match generation)	Don't care*	Don't care*	Burst/cycle-steal	
					1	0	ATU-II	CMI6D (CYLR6D compare-match generation)	Don't care*	Don't care*	Burst/cycle-steal
					1	1	ATU-II	CMI7A (CYLR7A compare-match generation)	Don't care*	Don't care*	Burst/cycle-steal
					1	0	ATU-II	CMI7B (CYLR7B compare-match generation)	Don't care*	Don't care*	Burst/cycle-steal
					1	1	ATU-II	CMI7C (CYLR7C compare-match generation)	Don't care*	Don't care*	Burst/cycle-steal
					1	0	0	ATU-II	CMI7D (CYLR7D compare-match generation)	Don't care*	Don't care*

**Legend:**

SCI0, SCI1, SCI2, SCI3, SCI4: Serial communication interface channels 0–4

A/D0, A/D1, A/D2: A/D converter channels 0–2

HCAN0: Controller area network-II channel 0

ATU-II: Advanced timer unit

TDR0, TDR1, TDR2, TDR3, TDR4: SCI0–SCI4 transmit data registers

RDR0, RDR1, RDR2, RDR3, RDR4: SCI0–SCI4 receive data registers

ADDR0–ADDR11: A/D0 data registers

ADDR12–ADDR23: A/D1 data registers

ADDR24–ADDR31: A/D2 data registers

MB0–MB15: HCAN0 message data

Note: \* External memory, memory-mapped external device, on-chip memory, on-chip peripheral module (excluding DMAC, BSC, and UBC)

**10.3.3 Channel Priority**

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to the following priority order:

- CH0 > CH1 > CH2 > CH3

**10.3.4 DMA Transfer Types**

The DMAC supports the transfers shown in table 10.3. It operates in dual address mode, in which both the transfer source and destination addresses are output. The dual address mode consists of a direct address mode, in which the output address value is the object of a direct data transfer, and an indirect address mode, in which the output address value is not the object of the data transfer, but the value stored at the output address becomes the transfer object address. The actual transfer operation timing varies with the bus mode. The DMAC has two bus modes: cycle-steal mode and burst mode.

**Table 10.3 Supported DMA Transfers**

Transfer Source	Transfer Destination			
	External Memory	Memory-Mapped External Device	On-Chip Memory	On-Chip Peripheral Module
External memory	Supported	Supported	Supported	Supported
Memory-mapped external device	Supported	Supported	Supported	Supported
On-chip memory	Supported	Supported	Supported	Supported
On-chip peripheral module	Supported	Supported	Supported	Supported

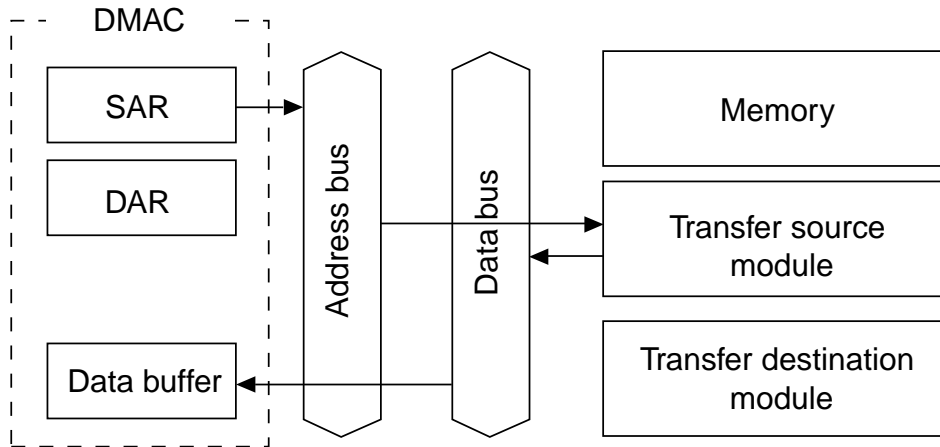
### 10.3.5 Dual Address Mode

Dual address mode is used for access of both the transfer source and destination by address. Transfer source and destination can be accessed either internally or externally. Dual address mode is subdivided into two other modes: direct address transfer mode and indirect address transfer mode.

**Direct Address Transfer Mode:** Data is read from the transfer source during the data read cycle, and written to the transfer destination during the write cycle, so transfer is conducted in two bus cycles. At this time, the transfer data is temporarily stored in the DMAC. With the kind of external memory transfer shown in figure 10.3, data is read from one of the memories by the DMAC during a read cycle, then written to the other external memory during the subsequent write cycle. Figure 10.4 shows the timing for this operation.

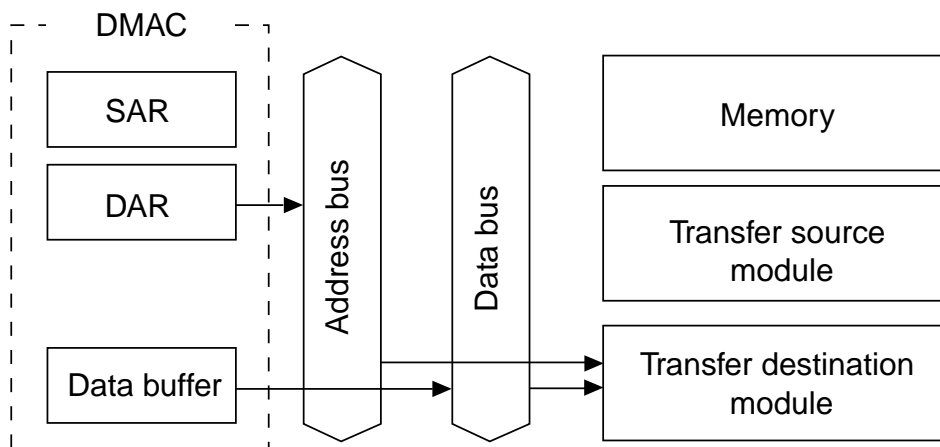


### 1st bus cycle



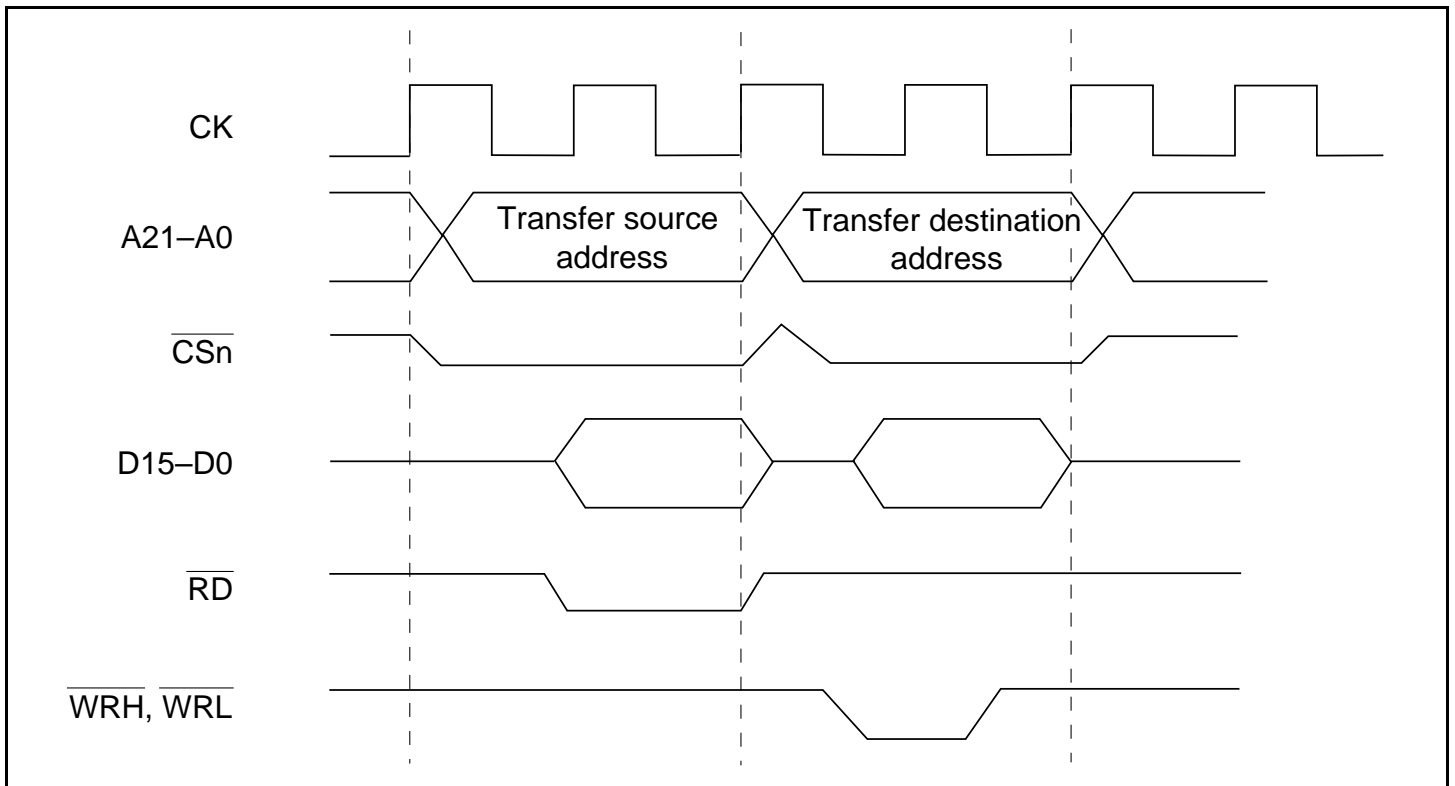
The SAR value is taken as the address, and data is read from the transfer source module and stored temporarily in the DMAC.

### 2nd bus cycle



The DAR value is taken as the address, and data stored in the DMAC's data buffer is written to the transfer destination module.

**Figure 10.3 Direct Address Operation in Dual Address Mode**



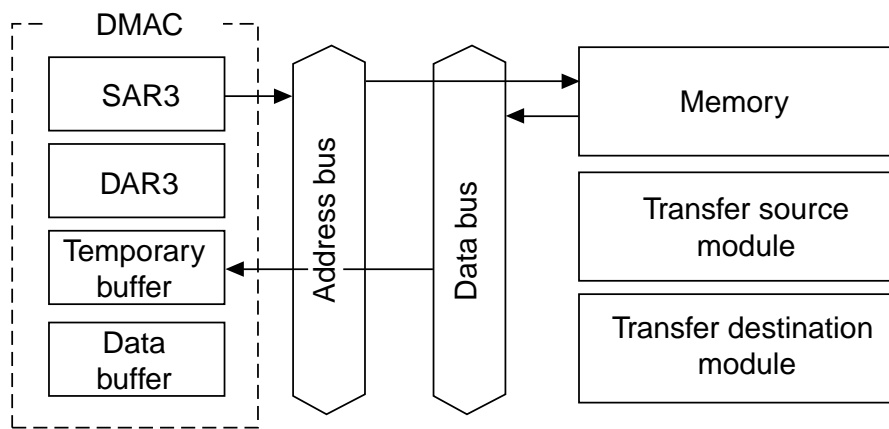
**Figure 10.4 Direct Address Transfer Timing in Dual Address Mode**

**Indirect Address Transfer Mode:** In this mode the memory address storing the data actually to be transferred is specified in the DMAC internal transfer source address register (SAR3). Therefore, in indirect address transfer mode, the DMAC internal transfer source address register value is read first. This value is first stored in the DMAC. Next, the read value is output as the address, and the value stored at that address is again stored in the DMAC. Finally, the subsequent read value is written to the address specified by the transfer destination address register, ending one cycle of DMAC transfer.

In indirect address mode (figure 10.5), the transfer destination, transfer source, and indirect address storage destination are all 16-bit external memory locations, and transfer in this example is conducted in 16-bit or 8-bit units. Timing for this transfer example is shown in figure 10.6.

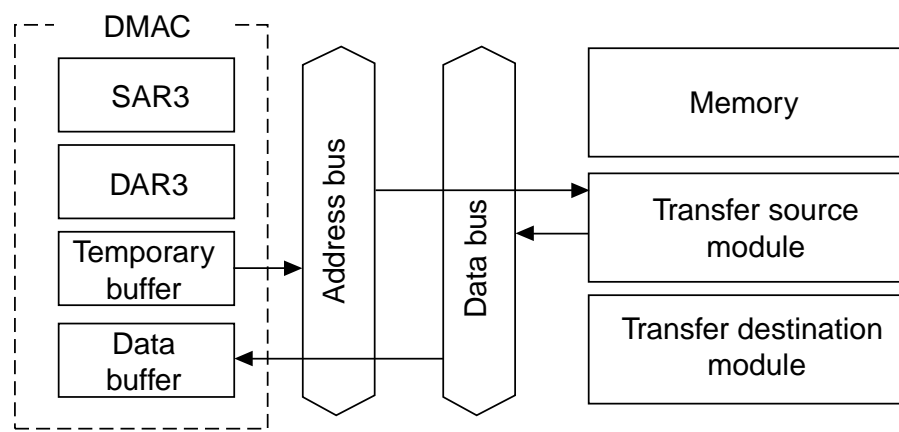
In indirect address mode, one NOP cycle (figure 10.6) is required until the data read as the indirect address is output to the address bus. When transfer data is 32-bit, the third and fourth bus cycles each need to be doubled, giving a required total of six bus cycles and one NOP cycle for the whole operation.

### 1st and 2nd bus cycles



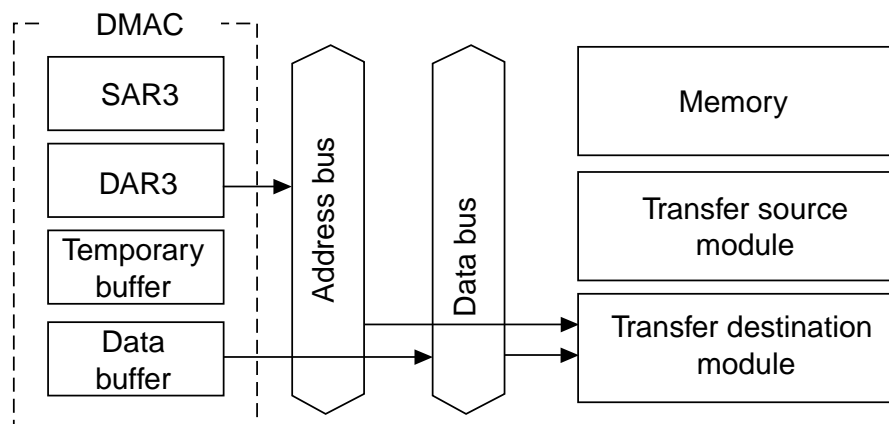
The SAR3 value is taken as the address, memory data is read, and the value is stored in the temporary buffer. Since the value read at this time is used as the address, it must be 32 bits. If data bus is 16 bits wide when accessed to an external memory space, two bus cycles are necessary.

### 3rd bus cycle



The value in the temporary buffer is taken as the address, and data is read from the transfer source module to the data buffer.

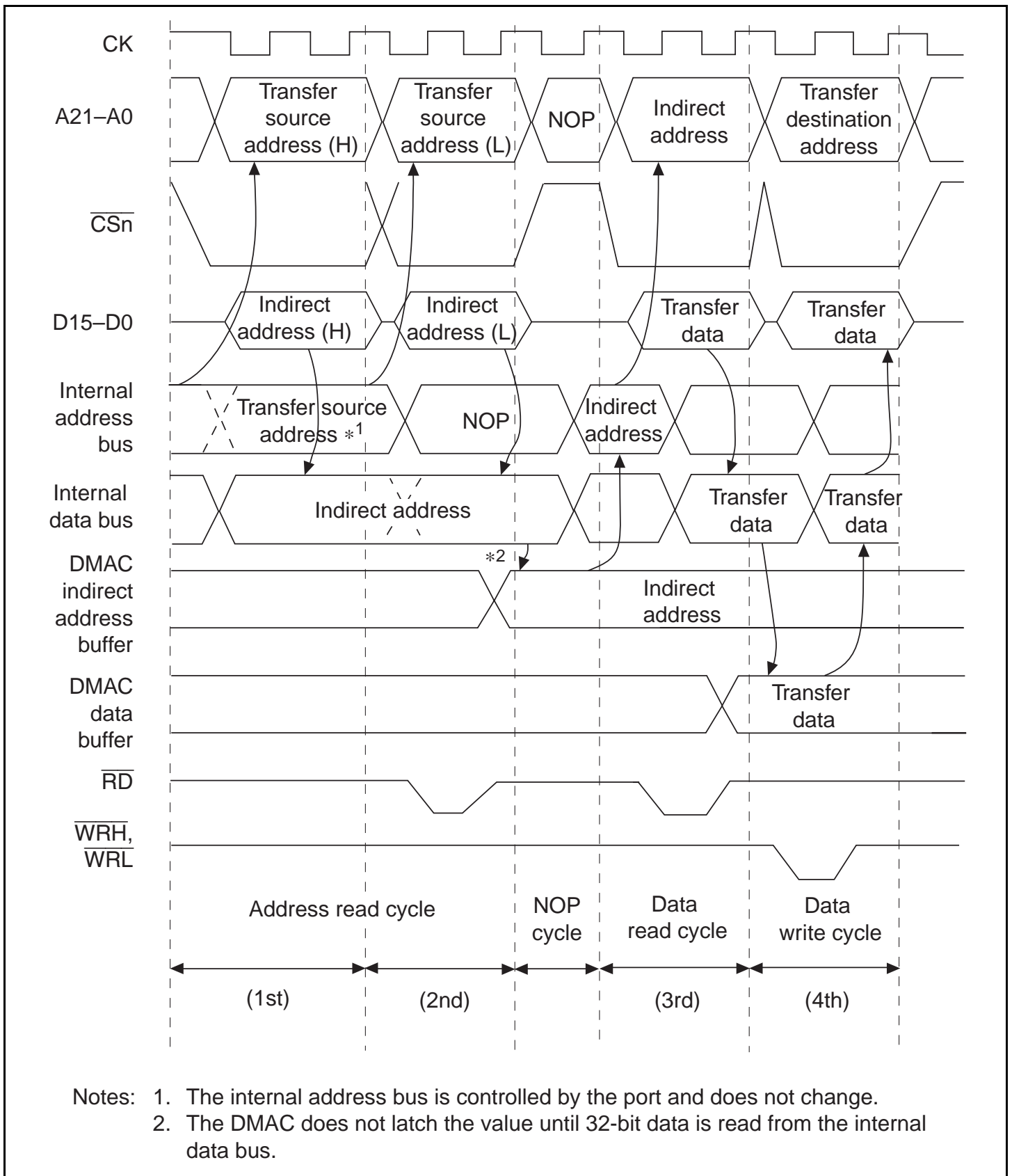
### 4th bus cycle



The DAR3 value is taken as the address, and the value in the data buffer is written to the transfer destination module.

Note: Memory, transfer source, and transfer destination modules are shown here. In practice, any connection can be made as long as it is within the address space.

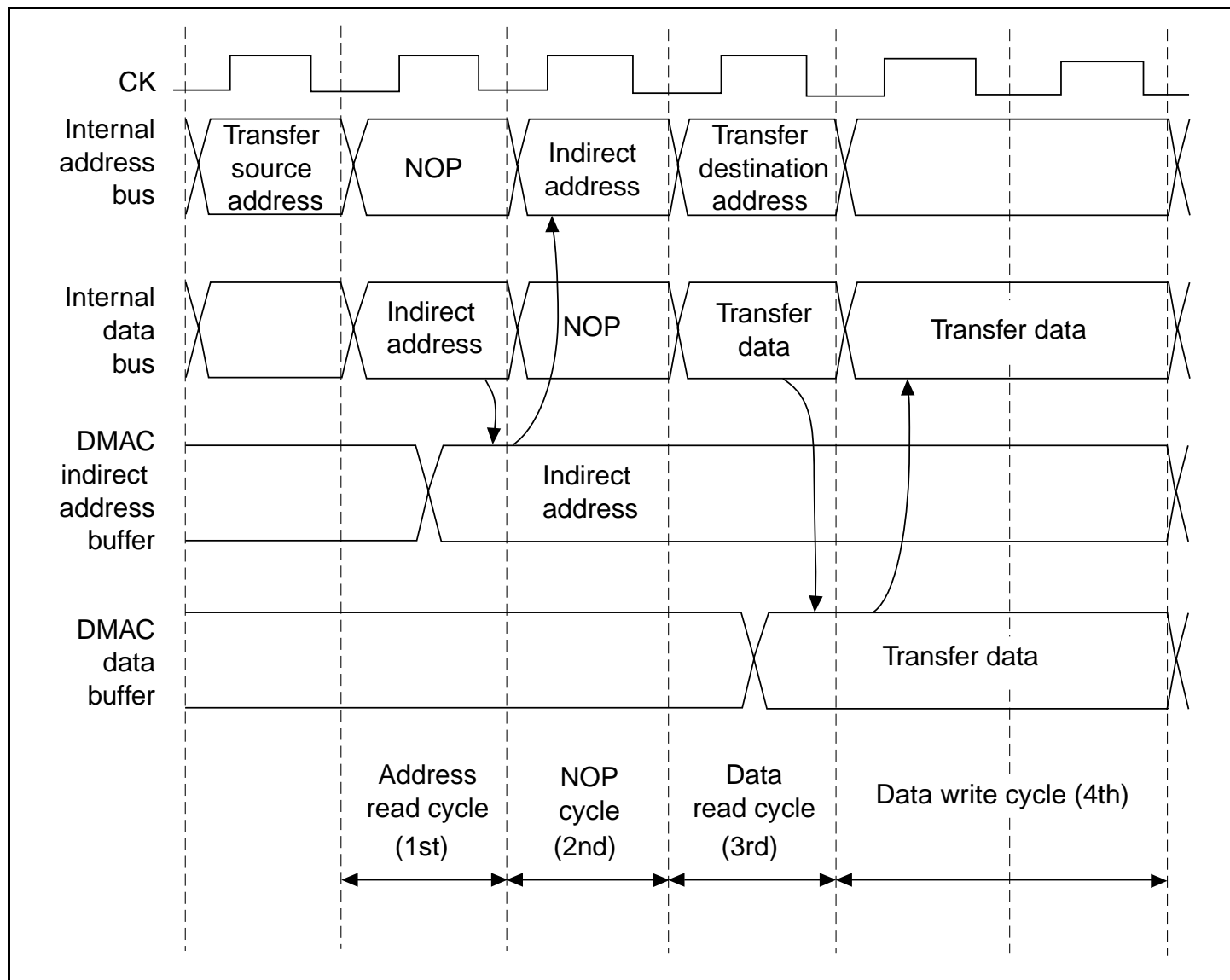
**Figure 10.5 Dual Address Mode and Indirect Address Operation (16-Bit-Width External Memory Space)**



**Figure 10.6 Dual Address Mode and Indirect Address Transfer Timing Example 1**  
**External Memory Space → External Memory Space**  
**(External memory space has 16-bit width)**

Figure 10.7 shows an example of timing in indirect address mode when transfer source and indirect address storage locations are in internal memory, the transfer destination is an on-chip peripheral module with 2-cycle access space, and transfer data is 8-bit.

Since the indirect address storage destination and the transfer source are in internal memory, these can be accessed in one cycle. The transfer destination is 2-cycle access space, so two data write cycles are required. One NOP cycle is required until the data read as the indirect address is output to the address bus.



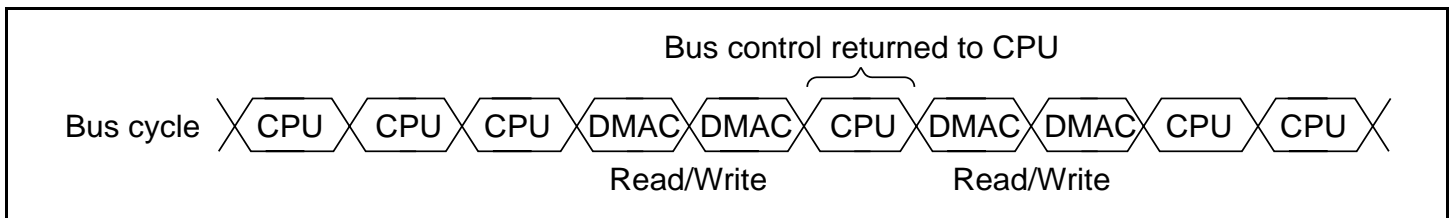
**Figure 10.7 Dual Address Mode and Indirect Address Transfer Timing Example 2  
Internal Memory Space → Internal Memory Space**

### 10.3.6 Bus Modes

Select the appropriate bus mode in the TM bits of CHCR0–CHCR3. There are two bus modes: cycle-steal and burst.

**Cycle-Steal Mode:** In cycle-steal mode, the bus right is given to another bus master after each one-transfer-unit (8-bit, 16-bit, or 32-bit) DMAC transfer. When the next transfer request occurs, the bus right is obtained from the other bus master and a transfer is performed for one transfer unit. When that transfer ends, the bus right is passed to the other bus master. This is repeated until the transfer end conditions are satisfied.

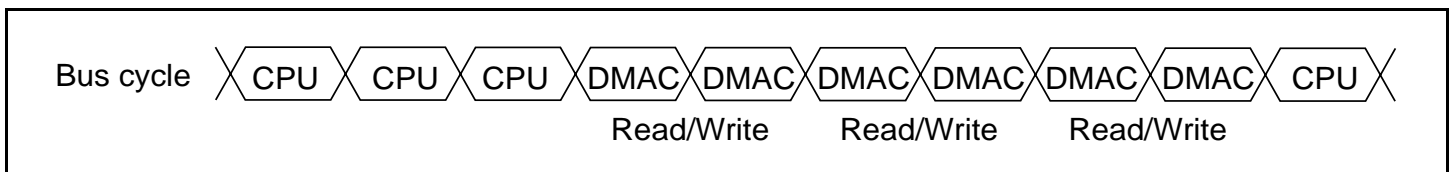
Cycle-steal mode can be used with all categories of transfer destination, transfer source and transfer request. Figure 10.8 shows an example of DMA transfer timing in cycle-steal mode.



**Figure 10.8 DMA Transfer Timing Example in Cycle-Steal Mode**

**Burst Mode:** Once the bus right is obtained, transfer is performed continuously until the transfer end condition is satisfied.

Figure 10.9 shows an example of DMA transfer timing in burst mode.



**Figure 10.9 DMA Transfer Timing Example in Burst Mode**

### 10.3.7 Relationship between Request Modes and Bus Modes by DMA Transfer Category

Table 10.4 shows the relationship between request modes and bus modes by DMA transfer category.

**Table 10.4 Relationship between Request Modes and Bus Modes by DMA Transfer Category**

Address Mode	Transfer Category	Request Mode	Bus Mode	Transfer Size (Bits)	Usable Channels
Dual	External memory and external memory	Any* <sup>1</sup>	B/C	8/16/32	0–3
	External memory and memory-mapped external device	Any* <sup>1</sup>	B/C	8/16/32	0–3
	Memory-mapped external device and memory-mapped external device	Any* <sup>1</sup>	B/C	8/16/32	0–3
	External memory and on-chip memory	Any* <sup>1</sup>	B/C	8/16/32	0–3
	External memory and on-chip peripheral module	Any* <sup>2</sup>	B/C* <sup>3</sup>	8/16/32* <sup>4</sup>	0–3
	Memory-mapped external device and on-chip memory	Any* <sup>1</sup>	B/C	8/16/32	0–3
	Memory-mapped external device and on-chip peripheral module	Any* <sup>2</sup>	B/C* <sup>3</sup>	8/16/32* <sup>4</sup>	0–3
	On-chip memory and on-chip memory	Any* <sup>1</sup>	B/C	8/16/32	0–3
	On-chip memory and on-chip peripheral module	Any* <sup>2</sup>	B/C* <sup>3</sup>	8/16/32* <sup>4</sup>	0–3
	On-chip peripheral module and on-chip peripheral module	Any* <sup>2</sup>	B/C* <sup>3</sup>	8/16/32* <sup>4</sup>	0–3

B: Burst, C: Cycle-steal

- Notes:
1. Auto-request or on-chip peripheral module request enabled. However, in the case of an on-chip peripheral module request, it is not possible to specify the SCI, HCAN0, or A/D converter for the transfer request source.
  2. Auto-request or on-chip peripheral module request possible. However, if the transfer request source is also the SCI, HCAN0, or A/D converter, the transfer source or transfer destination must be same as the transfer source.
  3. When the transfer request source is the SCI, only cycle-steal mode is possible.
  4. Access size permitted by the on-chip peripheral module register that is the transfer source or transfer destination.

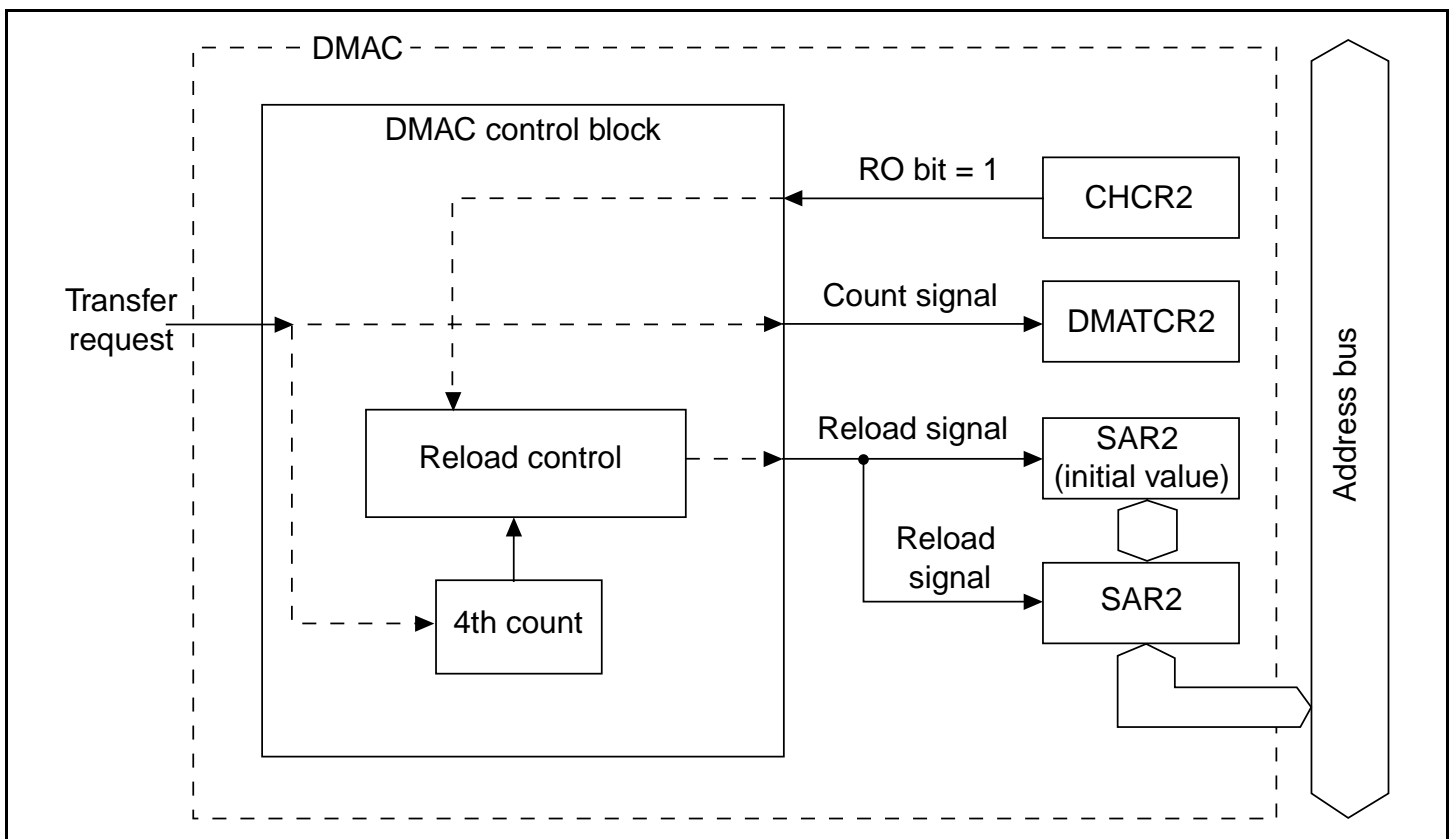
### 10.3.8 Bus Mode and Channel Priorities

If, for example, a transfer request is issued for channel 0 while transfer is in progress on lower-priority channel 1 in burst mode, transfer is started immediately on channel 0.

In this case, if channel 0 is set to burst mode, channel 1 transfer is continued after completion of all transfers on channel 0. If channel 0 is set to cycle-steal mode, channel 1 transfer is continued only if a channel 0 transfer request has not been issued; if a transfer request is issued, channel 0 transfer is started immediately.

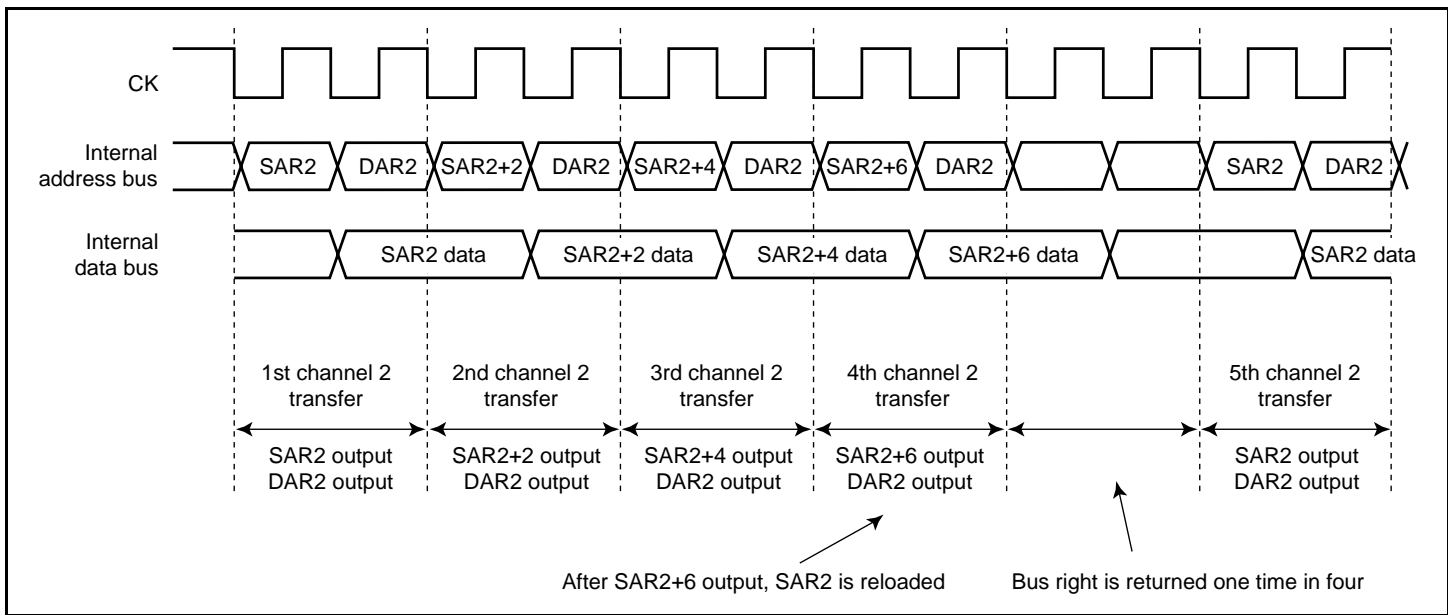
### 10.3.9 Source Address Reload Function

Channel 2 has a source address reload function. This returns to the first value set in the source address register (SAR2) every four transfers by setting the RO bit of CHCR2 to 1. Figure 10.10 illustrates this operation. Figure 10.11 is a timing chart for use of channel 2 only with the following transfer conditions set: burst mode, auto-request, 16-bit transfer data size, SAR2 incremented, DAR2 fixed, reload function on.



**Figure 10.10 Source Address Reload Function**





**Figure 10.11 Source Address Reload Function Timing Chart**

The reload function can be executed whether the transfer data size is 8, 16, or 32 bits.

DMATCR2, which specifies the number of transfers, is decremented by 1 at the end of every single-transfer-unit transfer, regardless of whether the reload function is on or off. Therefore, when using the reload function in the on state, a multiple of 4 must be specified in DMATCR2. Operation will not be guaranteed if any other value is set. Also, the counter which counts the occurrence of four transfers for address reloading is reset by clearing of the DME bit in DMAOR or the DE bit in CHCR2, setting of the transfer end flag (the TE bit in CHCR2), NMI input, and setting of the AE flag (address error generation in DMAC transfer), as well as by a reset and in software standby mode, but SAR2, DAR2, DMATCR2, and other registers are not reset. Consequently, when one of these sources occurs, there is a mixture of initialized counters and uninitialized registers in the DMAC, and incorrect operation may result if a restart is executed in this state. Therefore, when one of the above sources, other than TE setting, occurs during use of the address reload function, SAR, DAR2, and DMATCR2 settings must be carried out before re-execution.

### 10.3.10 DMA Transfer Ending Conditions

The DMA transfer ending conditions vary for individual channels ending and for all channels ending together.

**Individual Channel Ending Conditions:** There are two ending conditions. A transfer ends when the value of the channel's DMA transfer count register (DMATCR) is 0, or when the DE bit of the channel's CHCR is cleared to 0.

- When DMATCR is 0: When the DMATCR value becomes 0 and the corresponding channel's DMA transfer ends, the transfer end flag bit (TE) is set in CHCR. If the IE (interrupt enable) bit has been set, a DMAC interrupt (DEI) request is sent to the CPU.
- When DE of CHCR is 0: Software can halt a DMA transfer by clearing the DE bit in the channel's CHCR. The TE bit is not set when this happens.

**Conditions for Ending on All Channels Simultaneously:** Transfers on all channels end when the NMIF (NMI flag) bit or AE (address error flag) bit is set to 1 in DMAOR, or when the DME bit in DMAOR is cleared to 0.

- When the NMIF or AE bit is set to 1 in DMAOR: When an NMI interrupt or DMAC address error occurs, the NMIF or AE bit is set to 1 in DMAOR and all channels stop their transfers. The DMAC obtains the bus right, and if these flags are set to 1 during execution of a transfer, DMAC halts operation when the transfer processing currently being executed ends, and transfers the bus right to the other bus master. Consequently, even if the NMIF or AE bit is set to 1 during a transfer, the DMA source address register (SAR), designation address register (DAR), and transfer count register (DMATCR) are all updated. The TE bit is not set. To resume the transfers after NMI interrupt or address error processing, the NMIF or AE flag must be cleared. To avoid restarting a transfer on a particular channel, clear its DE bit to 0 in CHCR.

When the processing of a one-unit transfer is complete: In a dual address mode direct address transfer, even if an address error occurs or the NMI flag is set during read processing, the transfer will not be halted until after completion of the following write processing. In such a case, SAR, DAR, and DMATCR values are updated. In the same manner, the transfer is not halted in indirect address transfers until after the final write processing has ended.

- When DME is cleared to 0 in DMAOR: Clearing the DME bit to 0 in DMAOR aborts the transfers on all channels. The TE bit is not set.

### 10.3.11 DMAC Access from CPU

The space addressed by the DMAC is 4-cycle space. Therefore, when the CPU becomes the bus master and accesses the DMAC, a minimum of four internal clock cycles ( $\phi$ ) are required for one bus cycle. Also, since the DMAC is located in word space, while a word-size access to the DMAC is completed in one bus cycle, a longword-size access is automatically divided into two word accesses, requiring two bus cycles (eight basic clock cycles). These two bus cycles are executed consecutively; a different bus cycle is never inserted between the two word accesses. This applies to both write accesses and read accesses.

## 10.4 Examples of Use

### 10.4.1 Example of DMA Transfer between On-Chip SCI and External Memory

In this example, on-chip serial communication interface channel 0 (SCI0) receive data is transferred to external memory using DMAC channel 0.

Table 10.5 indicates the transfer conditions and the set values of each of the registers.

**Table 10.5 Transfer Conditions and Register Set Values for Transfer between On-chip SCI and External Memory**

<b>Transfer Conditions</b>	<b>Register</b>	<b>Value</b>
Transfer source: RDR0 of on-chip SCI0	SAR0	H'FFFFFF05
Transfer destination: external memory	DAR0	H'00400000
Transfer count: 64 times	DMATCR0	H'00000040
Transfer source address: fixed	CHCR0	H'00020105
Transfer destination address: incremented		
Transfer request source: SCI0 (RDR0)		
Bus mode: cycle-steal		
Transfer unit: byte		
Interrupt request generation at end of transfer		
DMAC master enable on	DMAOR	H'0001

### 10.4.2 Example of DMA Transfer between A/D Converter and On-Chip Memory (Address Reload On)

In this example, on-chip A/D converter channel 0 is the transfer source and on-chip memory is the transfer destination, and the address reload function is on.

Table 10.6 indicates the transfer conditions and the set values of each of the registers.

**Table 10.6 Transfer Conditions and Register Set Values for Transfer between A/D Converter and On-Chip Memory**

<b>Transfer Conditions</b>	<b>Register</b>	<b>Value</b>
Transfer source: on-chip A/D converter ch1 (A/D1)	SAR2	H'FFFFFF820
Transfer destination: on-chip memory	DAR2	H'FFFF6000
Transfer count: 128 times (reload count 32 times)	DMATCR2	H'00000080
Transfer source address: incremented	CHCR2	H'010C110D
Transfer destination address: incremented		
Transfer request source: A/D converter ch1 (A/D1)		
Bus mode: burst		
Transfer unit: byte		
Interrupt request generation at end of transfer		
DMAC master enable on	DMAOR	H'0001

When address reload is on, the SAR2 value returns to its initially set value every four transfers. In the above example, when a transfer request is input from the A/D1, the byte-size data is first read in from the H'FFFFFF820 register of on-chip A/D1 and that data is written to internal address H'FFFF6000. Because a byte-size transfer was performed, the SAR2 and DAR2 values at this point are H'FFFFFF821 and H'FFFF6001, respectively. Also, because this is a burst transfer, the bus right remains secured, so continuous data transfer is possible.

When four transfers are completed, if address reload is off, execution continues with the fifth and sixth transfers and the SAR2 value continues to increment from H'FFFFFF824 to H'FFFFFF825 to H'FFFFFF826 and so on. However, when address reload is on, DMAC transfer is halted upon completion of the fourth transfer and the bus right request signal to the CPU is cleared. At this time, the value stored in SAR2 is not H'FFFFFF823 → H'FFFFFF824, but H'FFFFFF823 → H'FFFFFF820, a return to the initially set address. The DAR2 value always continues to be decremented regardless of whether address reload is on or off.

The DMAC internal status, due to the above operation after completion of the fourth transfer, is indicated in table 10.7 for both address reload on and off.

**Table 10.7 DMAC Internal Status**

Item	Address Reload On	Address Reload Off
SAR2	H'FFFFFF820	H'FFFFFF824
DAR2	H'FFFF6004	H'FFFF6004
DMATCR2	H'0000007C	H'0000007C
Bus right	Released	Retained
DMAC operation	Halted	Processing continues
Interrupts	Not issued	Not issued
Transfer request source flag clear	Executed	Not executed

- Notes:
1. Interrupts are executed until the DMATCR2 value becomes 0, and if the IE bit of CHCR2 is set to 1, are issued regardless of whether address reload is on or off.
  2. If transfer request source flag clears are executed until the DMATCR2 value becomes 0, they are executed regardless of whether address reload is on or off.
  3. Designate burst mode when using the address reload function. There are cases where abnormal operation will result if it is used in cycle-steal mode.
  4. Designate a multiple of four for the DMATCR2 value when using the address reload function. There are cases where abnormal operation will result if anything else is designated.

To execute transfers after the fifth transfer when address reload is on, have the transfer request source issue another transfer request signal.

### 10.4.3 Example of DMA Transfer between External Memory and SCI1 Transmitting Side (Indirect Address on)

In this example, DMAC channel 3 is used, indirect address designated external memory is the transfer source, and the SCI1 transmitting side is the transfer destination.

Table 10.8 indicates the transfer conditions and the set values of each of the registers.

**Table 10.8 Transfer Conditions and Register Set Values for Transfer between External Memory and SCI1 Transmitting Side**

<b>Transfer Conditions</b>	<b>Register</b>	<b>Value</b>
Transfer source: external memory	SAR3	H'00400000
Value stored in address H'00400000	—	H'00450000
Value stored in address H'00450000	—	H'55
Transfer destination: on-chip SCI TDR1	DAR3	H'FFFFFF0B
Transfer count: 10 times	DMATCR3	H'0000000A
Transfer source address: incremented	CHCR3	H'10031001
Transfer destination address: fixed		
Transfer request source: SCI1 (TDR1)		
Bus mode: cycle-steal		
Transfer unit: byte		
Interrupt request not generated at end of transfer		
DMAC master enable on	DMAOR	H'0001

When indirect address mode is on, the data stored in the address set in SAR is not used as the transfer source data. In the case of indirect addressing, the value stored in the SAR address is read, then that value is used as the address and the data read from that address is used as the transfer source data, then that data is stored in the address designated by DAR.

In the table 10.8 example, when a transfer request from TDR1 of SCI1 is generated, a read of the address located at H'00400000, which is the value set in SAR3, is performed first. The data H'00450000 is stored at this H'00400000 address, and the DMAC first reads this H'00450000 value. It then uses this read value of H'00450000 as an address and reads the value of H'55 that is stored in the H'00450000 address. It then writes the value H'55 to address H'FFFFFF0B designated by DAR3 to complete one indirect address transfer.

With indirect addressing, the first executed data read from the address set in SAR3 always results in a longword size transfer regardless of the TS0 and TS1 bit designations for transfer data size. However, the transfer source address fixed and increment or decrement designations are according to the SM0 and SM1 bits. Consequently, despite the fact that the transfer data size designation is byte in this example, the SAR3 value at the end of one transfer is H'00400004. The write operation is exactly the same as an ordinary dual address transfer write operation.

## 10.5 Usage Notes

1. Only word (16-bit) access can be used on the DMA operation register (DMAOR). All other registers can be accessed in word (16-bit) or longword (32-bit) units.
2. When rewriting the RS0–RS4 bits of CHCR0–CHCR3, first clear the DE bit to 0 (clear the DE bit to 0 before modifying CHCR).
3. When an NMI interrupt is input, the NMIF bit of DMAOR is set even when the DMAC is not operating.
4. Clear the DME bit of DMAOR to 0 and make certain that any transfer request processing accepted by the DMAC has been completed before entering standby mode.
5. Do not access the DMAC, BSC, or UBC on-chip peripheral modules from the DMAC.
6. When activating the DMAC, make the CHCR settings as the final step. Abnormal operation may result if any other registers are set last.
7. After the DMATCR count becomes 0 and the DMA transfer ends normally, always write 0 to DMATCR, even when executing the maximum number of transfers on the same channel. Abnormal operation may result if this is not done.
8. Designate burst mode as the transfer mode when using the address reload function. Abnormal operation may result in cycle-steal mode.
9. Designate a multiple of four for the DMATCR value when using the address reload function, otherwise abnormal operation may result.
10. Do not access empty DMAC register addresses. Operation cannot be guaranteed when empty addresses are accessed.
11. If DMAC transfer is aborted by NMIF or AE setting, or DME or DE clearing, during DMAC execution with address reload on, the SAR2, DAR2, and DMATCR2 settings should be made before re-executing the transfer. The DMAC may not operate correctly if this is not done.
12. Do not set the DE bit to 1 while bits RS0 to RS4 in CHCR0 to CHCR3 are still set to “no request.”

# Section 11 Advanced Timer Unit-II (ATU-II)

## 11.1 Overview

The SH7058 has an on-chip advanced timer unit-II (ATU-II) with one 32-bit timer channel and eleven 16-bit timer channels.

### 11.1.1 Features

ATU-II features are summarized below.

- Capability to process up to 65 pulse inputs and outputs
- Prescaler
  - Input clock to channels 0 and 10 scaled in 1 stage, input clock to channels 1 to 8 and 11 scaled in 2 stages
  - 1/1 to 1/32 clock scaling possible in initial stage for channels 0 to 8, 10, and 11
  - 1/1, 1/2, 1/4, 1/8, 1/16, or 1/32 scaling possible in second stage for channels 1 to 8 and 11
  - External clock TCLKA, TCLKB selection also possible for channels 1 to 5 and 11
  - TI10, TI10 multiplication (compensation) selection possible for channels 1 to 5: AGCK, AGCKM
- Channel 0 has four 32-bit input capture lines, allowing the following operations:
  - Rising-edge, falling-edge, or both-edge detection selectable
  - DMAC can be activated at capture timing
  - Channel 10 compare-match signal can be captured as a trigger
  - Interval interrupt generation function generates three interval interrupts as selected. CPU interruption or A/D converter (AD0, 1, 2) activation possible
  - Capture interrupt and counter overflow interrupt can be generated
- Channel 1 has one 16-bit output compare register, eight general registers, and one dedicated input capture register. The output compare register can also be selected for one-shot pulse offset in combination with the channel 8 down-counter.
  - General registers (GR1A–H) can be used as input capture or output compare registers
  - Waveform output by means of compare-match: Selection of 0 output, 1 output, or toggle output
  - Input capture function: Rising-edge, falling-edge, or both-edge detection
  - Channel 0 input signal (TI0A) can be captured as trigger
  - Provision for forcible cutoff of channel 8 down-counters (DCNT8A–H)
  - Compare-match interrupts/capture interrupts and counter overflow interrupts can be generated



- Channel 2 has eight 16-bit output compare registers, eight general registers, and one dedicated input capture register. The output compare registers can also be selected for one-shot pulse offset in combination with the channel 8 down-counter.
  - General registers (GR2A–H) can be used as input capture or output compare registers
  - Waveform output by means of compare-match: Selection of 0 output, 1 output, or toggle output
  - Input capture function: Rising-edge, falling-edge, or both-edge detection
  - Channel 0 input signal (TI0A) can be captured as trigger
  - Provision for forcible cutoff of channel 8 down-counters (DCNT8I–P)
  - Compare-match interrupts/capture interrupts and counter overflow interrupts can be generated
- Channels 3 to 5 each have four general registers, allowing the following operations:
  - Selection of input capture, output compare, PWM mode
  - Waveform output by means of compare-match: Selection of 0 output, 1 output, or toggle output
  - Input capture function: Rising-edge, falling-edge, or both-edge detection
  - Channel 9 compare-match signal can be captured as trigger (channel 3 only)
  - Compare-match interrupts/capture interrupts can be generated
- Channels 6 and 7 have four 16-bit duty registers, four cycle registers, and four buffer registers, allowing the following operations:
  - Any cycle and duty from 0 to 100% can be set
  - Duty buffer register value transferred to duty register every cycle
  - Interrupts can be generated every cycle
  - Complementary PWM output can be set (channel 6 only)
- Channel 8 has sixteen 16-bit down-counters for one-shot pulse output, allowing the following operations:
  - One-shot pulse generation by down-counter
  - Down-counter can be rewritten during count
  - Interrupt can be generated at end of down-count
  - Offset one-shot pulse function available
  - Can be linked to channel 1 and 2 output compare functions
  - Reload function can be set to eight 16-bit down-counters (DCNT8I to DCNT8P)
- Channel 9 has six event counters and six general registers, allowing the following operations:
  - Event counters can be cleared by compare-match
  - Rising-edge, falling-edge, or both-edge detection available for external input
  - Compare-match signal can be input to channel 3
- Channel 10 has a 32-bit output compare and input capture register, free-running counter, 16-bit free-running counter, output compare/input capture register, reload register, 8-bit event

counter, and output compare register, and 16-bit reload counter, allowing the following operations:

- Capture on external input pin edge input
- Reload count possible with 1/32, 1/64, 1/128, or 1/256 times the captured value
- Internal clock generated by reload counter underflow can be used as 16-bit free-running counter input
- Channel 1 and 2 free-running counter clearing capability
- Channel 11 has one 16-bit free-running counter and two 16-bit general registers, allowing the following operations:
  - Two general registers can be used for input capture/output compare
  - Waveform output at compare-match: Selection of 0, 1, or toggle output
  - Input capture function: Selection of rising edge, falling edge, or both edge detection
  - Compare-match signal can be output to APC by using a general register as an output compare register
- High-speed access to internal 16-bit bus
  - High-speed access to 16-bit bus for 16-bit registers: timer counters, compare registers, and capture registers
- 75 interrupt sources
  - Four input capture interrupt requests, one overflow interrupt request, and one interval interrupt request for channel 0
  - Sixteen dual input capture/compare-match interrupt requests and two counter overflow interrupt requests for channels 1 and 2
  - Twelve dual input capture/compare-match interrupt requests and three overflow interrupt requests for channels 3 to 5
  - Eight compare-match interrupts for channels 6 and 7
  - Sixteen one-shot end interrupt requests for channel 8
  - Six compare-match interrupts for channel 9
  - Two compare-match interrupts and one dual-function input capture/compare-match interrupt for channel 10
  - Two dual input capture/compare-match interrupt requests and one overflow interrupt request for channel 11
- Direct memory access controller (DMAC) activation
  - The DMAC can be activated by a channel 0 input capture interrupt (ICI0A–D)
  - The DMAC can be activated by a channel 6 cycle register 6 compare-match interrupt (CMI6A–D)
  - The DMAC can be activated by a channel 7 cycle register 7 compare-match interrupt (CMI7A–D)
- A/D converter activation

- The A/D converter can be activated by detection of 1 in bits ITVA6–13 of the channel 0 interval interrupt request registers (ITVRR1, ITVRR2A, ITVRR2B)

Table 11.1 lists the functions of the ATU-II.

**Table 11.1 ATU-II Functions**

Item	Channel 0	Channel 1	Channel 2	Channels 3–5	
Counter configuration	Clock sources	$\phi-\phi/32$	$(\phi-\phi/32) \times (1/2^n)$ (n = 0–5) TCLKA, TCLKB, AGCK, AGCKM	$(\phi-\phi/32) \times (1/2^n)$ (n = 0–5) TCLKA, TCLKB, AGCK, AGCKM	$(\phi-\phi/32) \times (1/2^n)$ (n = 0–5) TCLKA, TCLKB, AGCK, AGCKM
	Counters	TCNT0H, TCNT0L	TCNT1A, TCNT1B	TCNT2A, TCNT2B	TCNT3–5
	General registers	—	GR1A–H	GR2A–H	GR3A–D, GR4A–D, GR5A–D
	Dedicated input capture	ICR0AH, ICR0AL, ICR0BH, ICR0BL, ICR0CH, ICR0CL, ICR0DH, ICR0DL	OSBR1	OSBR2	—
	Dedicated output compare	—	OCR1	OCR2A–2H	—
	PWM output	—	—	—	Duty: GR3A–C, GR4A–C, GR5A–C Cycle: GR3D, GR4D, GR5D
Input pins	TIOA–D	—	—	—	
I/O pins	—	TIO1A–H	TIO2A–H	TIO3A–D, TIO4A–D, TIO5A–D	
Output pins	—	—	—	—	
Counter clearing function	—	—	—	O	
Interrupt sources	6 sources Interval $\times$ 1, input capture $\times$ 4, overflow $\times$ 1	9 sources Dual input capture/ compare-match $\times$ 8, overflow $\times$ 1	9 sources Dual input capture/ compare-match $\times$ 8, overflow $\times$ 1*  (* Same vector)	15 sources Dual input capture/ compare-match $\times$ 12, overflow $\times$ 3	
Inter-channel and inter-module connection signals	A/D converter activation by interval interrupt request, DMAC activation by input capture interrupt, channel 10 compare-match signal capture trigger input	Compare-match signal trigger output to channel 8 one-shot pulse output down-counter  Channel 10 compare-match signal counter clear input	Compare-match signal trigger output to channel 8 one-shot pulse output down-counter  Channel 10 compare-match signal counter clear input	Channel 9 compare-match signal input to capture trigger  (Channel 3 only)	

**Table 11.1 ATU-II Functions (cont)**

Item		Channels 6, 7	Channel 8	Channel 9	Channel 10	Channel 11
Counter configuration	Clock sources	$(\phi-\phi/32) \times (1/2^n)$ (n = 0–5)	$(\phi-\phi/32) \times (1/2^n)$ (n = 0–5)	—	$(\phi-\phi/32)$	$(\phi-\phi/32) \times (1/2^n)$ (n = 0-5) TCLKA, TCLKB
	Counters	TCNT6A–D, TCNT7A–D	DCNT8A–P	ECNT9A–F	TCNT10AH, TCNT10AL, TCNT10B–H	TCNT11
	General registers	—	—	—	—	GR11A, GR11B
	Dedicated input capture	—	—	—	ICR10AH, ICR10AL	—
	Dedicated output compare	—	—	GR9A–F	GR10G, OCR/0AH, OCR/0AL, OCR/0B, NCR10, TCCLR10	—
	PWM output	CYLR6A–D, CYLR7A–D, DTR6A–D, DTR7A–D, BFR6A–D, BFR7A–D	—	—	—	—
Input pins	—	—	TI9A–F	TI10	—	
I/O pins	—	—	—	—	TIO11A, TIO11B	
Output pins	TO6A–D, TO7A–D	TO8A–P	—	—	—	
Counter clearing function	O	—	O	O	—	
Interrupt sources	8 sources Compare-match × 8	16 sources Underflow × 16	6 sources Compare-match × 6	3 sources Compare-match × 2, dual input capture/compare - match × 1	3 sources Dual input capture/compare overflow × 1	
Inter-channel and inter-module connection signals	DMAC activation compare-match signal output	Channel 1 and 2 compare-match signal trigger input to one-shot pulse output down-counter	Compare-match signal channel 3 capture trigger output	Compare-match signal channel 0 capture trigger output Channel 1 and 2 counter clear output	Compare-match signal output to APC	

O: Available

—: Not available

## 11.1.2 Pin Configuration

Table 11.2 shows the pin configuration of the ATU-II. When these external pin functions are used, the pin function controller (PFC) should also be set in accordance with the ATU-II settings. If there are a number of pins with the same function, make settings so that only one of the pins is used. For details, see section 21, Pin Function Controller (PEC).

**Table 11.2 ATU-II Pins**

Channel	Name	Abbreviation	I/O	Function
Common	Clock input A	TCLKA	Input	External clock A input pin
	Clock input B	TCLKB	Input	External clock B input pin
0	Input capture 0A	TIOA	Input	ICR0AH, ICR0AL input capture input pin
	Input capture 0B	TIOB	Input	ICR0BH, ICR0BL input capture input pin
	Input capture 0C	TIOC	Input	ICR0CH, ICR0CL input capture input pin
	Input capture 0D	TIOD	Input	ICR0DH, ICR0DL input capture input pin
1	Input capture/output compare 1A	TIO1A	Input/output	GR1A output compare output/input capture input
	Input capture/output compare 1B	TIO1B	Input/output	GR1B output compare output/input capture input
	Input capture/output compare 1C	TIO1C	Input/output	GR1C output compare output/input capture input
	Input capture/output compare 1D	TIO1D	Input/output	GR1D output compare output/input capture input
	Input capture/output compare 1E	TIO1E	Input/output	GR1E output compare output/input capture input
	Input capture/output compare 1F	TIO1F	Input/output	GR1F output compare output/input capture input
	Input capture/output compare 1G	TIO1G	Input/output	GR1G output compare output/input capture input
	Input capture/output compare 1H	TIO1H	Input/output	GR1H output compare output/input capture input

**Table 11.2 ATU-II Pins (cont)**

<b>Channel</b>	<b>Name</b>	<b>Abbreviation</b>	<b>I/O</b>	<b>Function</b>
2	Input capture/output compare 2A	TIO2A	Input/output	GR2A output compare output/input capture input
	Input capture/output compare 2B	TIO2B	Input/output	GR2B output compare output/input capture input
	Input capture/output compare 2C	TIO2C	Input/output	GR2C output compare output/input capture input
	Input capture/output compare 2D	TIO2D	Input/output	GR2D output compare output/input capture input
	Input capture/output compare 2E	TIO2E	Input/output	GR2E output compare output/input capture input
	Input capture/output compare 2F	TIO2F	Input/output	GR2F output compare output/input capture input
	Input capture/output compare 2G	TIO2G	Input/output	GR2G output compare output/input capture input
	Input capture/output compare 2H	TIO2H	Input/output	GR2H output compare output/input capture input
3	Input capture/output compare 3A	TIO3A	Input/output	GR3A output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 3B	TIO3B	Input/output	GR3B output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 3C	TIO3C	Input/output	GR3C output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 3D	TIO3D	Input/output	GR3D output compare output/input capture input
4	Input capture/output compare 4A	TIO4A	Input/output	GR4A output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 4B	TIO4B	Input/output	GR4B output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 4C	TIO4C	Input/output	GR4C output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 4D	TIO4D	Input/output	GR4D output compare output/input capture input

**Table 11.2 ATU-II Pins (cont)**

<b>Channel</b>	<b>Name</b>	<b>Abbreviation</b>	<b>I/O</b>	<b>Function</b>
5	Input capture/output compare 5A	TIO5A	Input/output	GR5A output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 5B	TIO5B	Input/output	GR5B output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 5C	TIO5C	Input/output	GR5C output compare output/input capture input/PWM output pin (PWM mode)
	Input capture/output compare 5D	TIO5D	Input/output	GR5D output compare output/input capture input
6	Output compare 6A	TO6A	Output	PWM output pin
	Output compare 6B	TO6B	Output	PWM output pin
	Output compare 6C	TO6C	Output	PWM output pin
	Output compare 6D	TO6D	Output	PWM output pin
7	Output compare 7A	TO7A	Output	PWM output pin
	Output compare 7B	TO7B	Output	PWM output pin
	Output compare 7C	TO7C	Output	PWM output pin
	Output compare 7D	TO7D	Output	PWM output pin
8	One-shot pulse 8A	TO8A	Output	One-shot pulse output pin
	One-shot pulse 8B	TO8B	Output	One-shot pulse output pin
	One-shot pulse 8C	TO8C	Output	One-shot pulse output pin
	One-shot pulse 8D	TO8D	Output	One-shot pulse output pin
	One-shot pulse 8E	TO8E	Output	One-shot pulse output pin
	One-shot pulse 8F	TO8F	Output	One-shot pulse output pin
	One-shot pulse 8G	TO8G	Output	One-shot pulse output pin
	One-shot pulse 8H	TO8H	Output	One-shot pulse output pin
	One-shot pulse 8I	TO8I	Output	One-shot pulse output pin
	One-shot pulse 8J	TO8J	Output	One-shot pulse output pin
	One-shot pulse 8K	TO8K	Output	One-shot pulse output pin
	One-shot pulse 8L	TO8L	Output	One-shot pulse output pin
	One-shot pulse 8M	TO8M	Output	One-shot pulse output pin
	One-shot pulse 8N	TO8N	Output	One-shot pulse output pin

**Table 11.2 ATU-II Pins (cont)**

<b>Channel</b>	<b>Name</b>	<b>Abbreviation</b>	<b>I/O</b>	<b>Function</b>
8	One-shot pulse 8O	TO8O	Output	One-shot pulse output pin
	One-shot pulse 8P	TO8P	Output	One-shot pulse output pin
9	Event input 9A	TI9A	Input	GR9A event input
	Event input 9B	TI9B	Input	GR9B event input
	Event input 9C	TI9C	Input	GR9C event input
	Event input 9D	TI9D	Input	GR9D event input
	Event input 9E	TI9E	Input	GR9E event input
	Event input 9F	TI9F	Input	GR9F event input
10	Input capture	TI10	Input	ICR10AH, ICR10AL input capture input
11	Input capture/output compare 11A	TIO11A	Input/output	GR11A output compare output/input capture input
	Input capture/output compare 11B	TIO11B	Input/output	GR11B output compare output/input capture input



### 11.1.3 Register Configuration

Table 11.3 summarizes the ATU-II registers.

**Table 11.3 ATU-II Registers**

Channel	Name	Abbrevia- tion	R/W	Initial Value	Address	Access Size (Bits)	Section No.	
Common	Timer start register 1	TSTR1	R/W	H'00	H'FFFFFF401	8, 16, 32	11.2.1	
	Timer start register 2	TSTR2	R/W	H'00	H'FFFFFF400			
	Timer start register 3	TSTR3	R/W	H'00	H'FFFFFF402			
		Prescaler register 1	PSCR1	R/W	H'00	H'FFFFFF404	8	11.2.2
		Prescaler register 2	PSCR2	R/W	H'00	H'FFFFFF406		
		Prescaler register 3	PSCR3	R/W	H'00	H'FFFFFF408		
		Prescaler register 4	PSCR4	R/W	H'00	H'FFFFFF40A		
0	Free-running counter 0H	TCNT0H	R/W	H'0000	H'FFFFFF430	32	11.2.15	
	Free-running counter 0L	TCNT0L	R/W	H'0000				
	Input capture register 0AH	ICR0AH	R	H'0000	H'FFFFFF434	11.2.19		
	Input capture register 0AL	ICR0AL	R	H'0000				
	Input capture register 0BH	ICR0BH	R	H'0000	H'FFFFFF438			
	Input capture register 0BL	ICR0BL	R	H'0000				
	Input capture register 0CH	ICR0CH	R	H'0000	H'FFFFFF43C			
	Input capture register 0CL	ICR0CL	R	H'0000				
	Input capture register 0DH	ICR0DH	R	H'0000	H'FFFFFF420			
	Input capture register 0DL	ICR0DL	R	H'0000				
	Timer interval interrupt request register 1	ITVRR1	R/W	H'00	H'FFFFFF424	8	11.2.7	
	Timer interval interrupt request register 2A	ITVRR2A	R/W	H'00	H'FFFFFF426			

**Table 11.3 ATU-II Registers (cont)**

Channel	Name	Abbrevia- tion	R/W	Initial Value	Address	Access Size (Bits)	Section No.
0	Timer interval interrupt request register 2B	ITVRR2B	R/W	H'00	H'FFFFFF428	8	11.2.7
	Timer I/O control register	TIOR0	R/W	H'00	H'FFFFFF42A		11.2.4
	Timer status register 0	TSR0	R/(W)*	H'0000	H'FFFFFF42C	16	11.2.5
	Timer interrupt enable register 0	TIER0	R/W	H'0000	H'FFFFFF42E		11.2.6
1	Free-running counter 1A	TCNT1A	R/W	H'0000	H'FFFFFF440	16	11.2.15
	Free-running counter 1B	TCNT1B	R/W	H'0000	H'FFFFFF442		
	General register 1A	GR1A	R/W	H'FFFF	H'FFFFFF444		11.2.20
	General register 1B	GR1B	R/W	H'FFFF	H'FFFFFF446		
	General register 1C	GR1C	R/W	H'FFFF	H'FFFFFF448		
	General register 1D	GR1D	R/W	H'FFFF	H'FFFFFF44A		
	General register 1E	GR1E	R/W	H'FFFF	H'FFFFFF44C		
	General register 1F	GR1F	R/W	H'FFFF	H'FFFFFF44E		
	General register 1G	GR1G	R/W	H'FFFF	H'FFFFFF450		
	General register 1H	GR1H	R/W	H'FFFF	H'FFFFFF452		
	Output compare register 1	OCR1	R/W	H'FFFF	H'FFFFFF454		11.2.18
	Offset base register 1	OSBR1	R	H'0000	H'FFFFFF456		11.2.21
	Timer I/O control register 1A	TIOR1A	R/W	H'00	H'FFFFFF459	8, 16	11.2.4
	Timer I/O control register 1B	TIOR1B	R/W	H'00	H'FFFFFF458		
	Timer I/O control register 1C	TIOR1C	R/W	H'00	H'FFFFFF45B		
	Timer I/O control register 1D	TIOR1D	R/W	H'00	H'FFFFFF45A		
	Timer control register 1A	TCR1A	R/W	H'00	H'FFFFFF45D		11.2.3
	Timer control register 1B	TCR1B	R/W	H'00	H'FFFFFF45C		

**Table 11.3 ATU-II Registers (cont)**

Channel	Name	Abbrevia- tion	R/W	Initial Value	Address	Access Size (Bits)	Section No.
1	Timer status register 1A	TSR1A	R/(W)*	H'0000	H'FFFFFF45E	16	11.2.5
	Timer status register 1B	TSR1B	R/(W)*	H'0000	H'FFFFFF460		
	Timer interrupt enable register 1A	TIER1A	R/W	H'0000	H'FFFFFF462		11.2.6
	Timer interrupt enable register 1B	TIER1B	R/W	H'0000	H'FFFFFF464		
	Trigger mode register	TRGMDR	R/W	H'00	H'FFFFFF466	8	11.2.8
2	Free-running counter 2A	TCNT2A	R/W	H'0000	H'FFFFFF600	16	11.2.15
	Free-running counter 2B	TCNT2B	R/W	H'0000	H'FFFFFF602		
	General register 2A	GR2A	R/W	H'FFFF	H'FFFFFF604		11.2.20
	General register 2B	GR2B	R/W	H'FFFF	H'FFFFFF606		
	General register 2C	GR2C	R/W	H'FFFF	H'FFFFFF608		
	General register 2D	GR2D	R/W	H'FFFF	H'FFFFFF60A		
	General register 2E	GR2E	R/W	H'FFFF	H'FFFFFF60C		
	General register 2F	GR2F	R/W	H'FFFF	H'FFFFFF60E		
	General register 2G	GR2G	R/W	H'FFFF	H'FFFFFF610		
	General register 2H	GR2H	R/W	H'FFFF	H'FFFFFF612		
	Output compare register 2A	OCR2A	R/W	H'FFFF	H'FFFFFF614		11.2.18
	Output compare register 2B	OCR2B	R/W	H'FFFF	H'FFFFFF616		
	Output compare register 2C	OCR2C	R/W	H'FFFF	H'FFFFFF618		
	Output compare register 2D	OCR2D	R/W	H'FFFF	H'FFFFFF61A		
	Output compare register 2E	OCR2E	R/W	H'FFFF	H'FFFFFF61C		
Output compare register 2F	OCR2F	R/W	H'FFFF	H'FFFFFF61E			

**Table 11.3 ATU-II Registers (cont)**

Channel	Name	Abbrevia- tion	R/W	Initial Value	Address	Access Size (Bits)	Section No.
2	Output compare register 2G	OCR2G	R/W	H'FFFF	H'FFFFFF620	16	11.2.18
	Output compare register 2H	OCR2H	R/W	H'FFFF	H'FFFFFF622		
	Offset base register 2	OSBR2	R	H'0000	H'FFFFFF624		11.2.21
	Timer I/O control register 2A	TIOR2A	R/W	H'00	H'FFFFFF627	8, 16	11.2.4
	Timer I/O control register 2B	TIOR2B	R/W	H'00	H'FFFFFF626		
	Timer I/O control register 2C	TIOR2C	R/W	H'00	H'FFFFFF629		
	Timer I/O control register 2D	TIOR2D	R/W	H'00	H'FFFFFF628		
	Timer control register 2A	TCR2A	R/W	H'00	H'FFFFFF62B		11.2.3
	Timer control register 2B	TCR2B	R/W	H'00	H'FFFFFF62A		
	Timer status register 2A	TSR2A	R/(W)*	H'0000	H'FFFFFF62C	16	11.2.5
	Timer status register 2B	TSR2B	R/(W)*	H'0000	H'FFFFFF62E		
	Timer interrupt enable register 2A	TIER2A	R/W	H'0000	H'FFFFFF630		11.2.6
Timer interrupt enable register 2B	TIER2B	R/W	H'0000	H'FFFFFF632			
3–5	Timer status register 3	TSR3	R/(W)*	H'0000	H'FFFFFF480	16	11.2.5
	Timer interrupt enable register 3	TIER3	R/W	H'0000	H'FFFFFF482		11.2.6
	Timer mode register	TMDR	R/W	H'00	H'FFFFFF484	8	11.2.9
3	Free-running counter 3	TCNT3	R/W	H'0000	H'FFFFFF4A0	16	11.2.15
	General register 3A	GR3A	R/W	H'FFFF	H'FFFFFF4A2		11.2.20
	General register 3B	GR3B	R/W	H'FFFF	H'FFFFFF4A4		
	General register 3C	GR3C	R/W	H'FFFF	H'FFFFFF4A6		
	General register 3D	GR3D	R/W	H'FFFF	H'FFFFFF4A8		

**Table 11.3 ATU-II Registers (cont)**

Channel	Name	Abbrevia- tion	R/W	Initial Value	Address	Access Size (Bits)	Section No.
3	Timer I/O control register 3A	TIOR3A	R/W	H'00	H'FFFFFF4AB	8, 16	11.2.4
	Timer I/O control register 3B	TIOR3B	R/W	H'00	H'FFFFFF4AA		
	Timer control register 3	TCR3	R/W	H'00	H'FFFFFF4AC	8	11.2.3
4	Free-running counter 4	TCNT4	R/W	H'0000	H'FFFFFF4C0	16	11.2.15
	General register 4A	GR4A	R/W	H'FFFF	H'FFFFFF4C2		11.2.20
	General register 4B	GR4B	R/W	H'FFFF	H'FFFFFF4C4		
	General register 4C	GR4C	R/W	H'FFFF	H'FFFFFF4C6		
	General register 4D	GR4D	R/W	H'FFFF	H'FFFFFF4C8		
	Timer I/O control register 4A	TIOR4A	R/W	H'00	H'FFFFFF4CB	8, 16	11.2.4
	Timer I/O control register 4B	TIOR4B	R/W	H'00	H'FFFFFF4CA		
	Timer control register 4	TCR4	R/W	H'00	H'FFFFFF4CC	8	11.2.3
5	Free-running counter 5	TCNT5	R/W	H'0000	H'FFFFFF4E0	16	11.2.15
	General register 5A	GR5A	R/W	H'FFFF	H'FFFFFF4E2		11.2.20
	General register 5B	GR5B	R/W	H'FFFF	H'FFFFFF4E4		
	General register 5C	GR5C	R/W	H'FFFF	H'FFFFFF4E6		
	General register 5D	GR5D	R/W	H'FFFF	H'FFFFFF4E8		
	Timer I/O control register 5A	TIOR5A	R/W	H'00	H'FFFFFF4EB	8, 16	11.2.4
	Timer I/O control register 5B	TIOR5B	R/W	H'00	H'FFFFFF4EA		
	Timer control register 5	TCR5	R/W	H'00	H'FFFFFF4EC	8	11.2.3
6	Free-running counter 6A	TCNT6A	R/W	H'0001	H'FFFFFF500	16	11.2.15
	Free-running counter 6B	TCNT6B	R/W	H'0001	H'FFFFFF502		
	Free-running counter 6C	TCNT6C	R/W	H'0001	H'FFFFFF504		
	Free-running counter 6D	TCNT6D	R/W	H'0001	H'FFFFFF506		

**Table 11.3 ATU-II Registers (cont)**

Channel	Name	Abbrevia- tion	R/W	Initial Value	Address	Access Size (Bits)	Section No.
6	Cycle register 6A	CYLR6A	R/W	H'FFFF	H'FFFFFF508	16	11.2.22
	Cycle register 6B	CYLR6B	R/W	H'FFFF	H'FFFFFF50A		
	Cycle register 6C	CYLR6C	R/W	H'FFFF	H'FFFFFF50C		
	Cycle register 6D	CYLR6D	R/W	H'FFFF	H'FFFFFF50E		
	Buffer register 6A	BFR6A	R/W	H'FFFF	H'FFFFFF510		11.2.23
	Buffer register 6B	BFR6B	R/W	H'FFFF	H'FFFFFF512		
	Buffer register 6C	BFR6C	R/W	H'FFFF	H'FFFFFF514		
	Buffer register 6D	BFR6D	R/W	H'FFFF	H'FFFFFF516		
	Duty register 6A	DTR6A	R/W	H'FFFF	H'FFFFFF518		11.2.24
	Duty register 6B	DTR6B	R/W	H'FFFF	H'FFFFFF51A		
	Duty register 6C	DTR6C	R/W	H'FFFF	H'FFFFFF51C		
	Duty register 6D	DTR6D	R/W	H'FFFF	H'FFFFFF51E		
	Timer control register 6A	TCR6A	R/W	H'00	H'FFFFFF521	8, 16	11.2.3
	Timer control register 6B	TCR6B	R/W	H'00	H'FFFFFF520		
	Timer status register 6	TSR6	R/(W)*	H'0000	H'FFFFFF522	16	11.2.5
	Timer interrupt enable register 6	TIER6	R/W	H'0000	H'FFFFFF524		11.2.6
PWM mode register	PMDR	R/W	H'00	H'FFFFFF526	8	11.2.10	
7	Free-running counter 7A	TCNT7A	R/W	H'0001	H'FFFFFF580	16	11.2.15
	Free-running counter 7B	TCNT7B	R/W	H'0001	H'FFFFFF582		
	Free-running counter 7C	TCNT7C	R/W	H'0001	H'FFFFFF584		
	Free-running counter 7D	TCNT7D	R/W	H'0001	H'FFFFFF586		
	Cycle register 7A	CYLR7A	R/W	H'FFFF	H'FFFFFF588		11.2.22
	Cycle register 7B	CYLR7B	R/W	H'FFFF	H'FFFFFF58A		
	Cycle register 7C	CYLR7C	R/W	H'FFFF	H'FFFFFF58C		
	Cycle register 7D	CYLR7D	R/W	H'FFFF	H'FFFFFF58E		

**Table 11.3 ATU-II Registers (cont)**

Channel	Name	Abbrevia- tion	R/W	Initial Value	Address	Access Size (Bits)	Section No.
7	Buffer register 7A	BFR7A	R/W	H'FFFF	H'FFFFFF590	16	11.2.23
	Buffer register 7B	BFR7B	R/W	H'FFFF	H'FFFFFF592		
	Buffer register 7C	BFR7C	R/W	H'FFFF	H'FFFFFF594		
	Buffer register 7D	BFR7D	R/W	H'FFFF	H'FFFFFF596		
	Duty register 7A	DTR7A	R/W	H'FFFF	H'FFFFFF598		11.2.24
	Duty register 7B	DTR7B	R/W	H'FFFF	H'FFFFFF59A		
	Duty register 7C	DTR7C	R/W	H'FFFF	H'FFFFFF59C		
	Duty register 7D	DTR7D	R/W	H'FFFF	H'FFFFFF59E		
	Timer control register 7A	TCR7A	R/W	H'00	H'FFFFFF5A1	8, 16	11.2.3
	Timer control register 7B	TCR7B	R/W	H'00	H'FFFFFF5A0		
	Timer status register 7	TSR7	R/(W)*	H'0000	H'FFFFFF5A2	16	11.2.5
	Timer interrupt enable register 7	TIER7	R/W	H'0000	H'FFFFFF5A4		11.2.6
8	Down-counter 8A	DCNT8A	R/W	H'0000	H'FFFFFF640	16	11.2.16
	Down-counter 8B	DCNT8B	R/W	H'0000	H'FFFFFF642		
	Down-counter 8C	DCNT8C	R/W	H'0000	H'FFFFFF644		
	Down-counter 8D	DCNT8D	R/W	H'0000	H'FFFFFF646		
	Down-counter 8E	DCNT8E	R/W	H'0000	H'FFFFFF648		
	Down-counter 8F	DCNT8F	R/W	H'0000	H'FFFFFF64A		
	Down-counter 8G	DCNT8G	R/W	H'0000	H'FFFFFF64C		
	Down-counter 8H	DCNT8H	R/W	H'0000	H'FFFFFF64E		
	Down-counter 8I	DCNT8I	R/W	H'0000	H'FFFFFF650		
	Down-counter 8J	DCNT8J	R/W	H'0000	H'FFFFFF652		
	Down-counter 8K	DCNT8K	R/W	H'0000	H'FFFFFF654		
	Down-counter 8L	DCNT8L	R/W	H'0000	H'FFFFFF656		
	Down-counter 8M	DCNT8M	R/W	H'0000	H'FFFFFF658		
	Down-counter 8N	DCNT8N	R/W	H'0000	H'FFFFFF65A		
	Down-counter 8O	DCNT8O	R/W	H'0000	H'FFFFFF65C		
	Down-counter 8P	DCNT8P	R/W	H'0000	H'FFFFFF65E		

**Table 11.3 ATU-II Registers (cont)**

Channel	Name	Abbrevia- tion	R/W	Initial Value	Address	Access Size (Bits)	Section No.
8	Reload register 8	RLDR8	R/W	H'0000	H'FFFFFF660	16	11.2.25
	Timer connection register	TCNR	R/W	H'0000	H'FFFFFF662		11.2.12
	One-shot pulse terminate register	OTR	R/W	H'0000	H'FFFFFF664		11.2.13
	Down-count start register	DSTR	R/W	H'0000	H'FFFFFF666		11.2.11
	Timer control register 8	TCR8	R/W	H'00	H'FFFFFF668	8	11.2.3
	Timer status register 8	TSR8	R/(W)*	H'0000	H'FFFFFF66A	16	11.2.5
	Timer interrupt enable register 8	TIER8	R/W	H'0000	H'FFFFFF66C		11.2.6
	Reload enable register	RLDENR	R/W	H'00	H'FFFFFF66E	8	11.2.14
9	Event counter 9A	ECNT9A	R/W	H'00	H'FFFFFF680	8	11.2.17
	Event counter 9B	ECNT9B	R/W	H'00	H'FFFFFF682		
	Event counter 9C	ECNT9C	R/W	H'00	H'FFFFFF684		
	Event counter 9D	ECNT9D	R/W	H'00	H'FFFFFF686		
	Event counter 9E	ECNT9E	R/W	H'00	H'FFFFFF688		
	Event counter 9F	ECNT9F	R/W	H'00	H'FFFFFF68A		
	General register 9A	GR9A	R/W	H'FF	H'FFFFFF68C		11.2.20
	General register 9B	GR9B	R/W	H'FF	H'FFFFFF68E		
	General register 9C	GR9C	R/W	H'FF	H'FFFFFF690		
	General register 9D	GR9D	R/W	H'FF	H'FFFFFF692		
	General register 9E	GR9E	R/W	H'FF	H'FFFFFF694		
	General register 9F	GR9F	R/W	H'FF	H'FFFFFF696		
	Timer control register 9A	TCR9A	R/W	H'00	H'FFFFFF698		11.2.3
	Timer control register 9B	TCR9B	R/W	H'00	H'FFFFFF69A		
	Timer control register 9C	TCR9C	R/W	H'00	H'FFFFFF69C		
	Timer status register 9	TSR9	R/(W)*	H'0000	H'FFFFFF69E	16	11.2.5
	Timer interrupt enable register 9	TIER9	R/W	H'0000	H'FFFFFF6A0		11.2.6



**Table 11.3 ATU-II Registers (cont)**

Channel	Name	Abbrevia- tion	R/W	Initial Value	Address	Access Size (Bits)	Section No.
10	Free-running counter 10AH	TCNT10AH	R/W	H'0000	H'FFFFFF6C0	32	11.2.26
	Free-running counter 10AL	TCNT10AL	R/W	H'0001			
	Event counter 10B	TCNT10B	R/W	H'00	H'FFFFFF6C4	8	
	Reload counter 10C	TCNT10C	R/W	H'0001	H'FFFFFF6C6	16	
	Correction counter 10D	TCNT10D	R/W	H'00	H'FFFFFF6C8	8	
	Correction angle counter 10E	TCNT10E	R/W	H'0000	H'FFFFFF6CA	16	
	Correction angle counter 10F	TCNT10F	R/W	H'0001	H'FFFFFF6CC		
	Free-running counter 10G	TCNT10G	R/W	H'0000	H'FFFFFF6CE		
	Input capture register 10AH	ICR10AH	R	H'0000	H'FFFFFF6D0	32	
	Input capture register 10AL	ICR10AL	R	H'0000			
	Output compare register 10AH	OCR10AH	R/W	H'FFFF	H'FFFFFF6D4		
	Output compare register 10AL	OCR10AL	R/W	H'FFFF			
	Output compare register 10B	OCR10B	R/W	H'FF	H'FFFFFF6D8	8	
	Reload register 10C	RLD10C	R/W	H'0000	H'FFFFFF6DA	16	
	General register 10G	GR10G	R/W	H'FFFF	H'FFFFFF6DC		
	Noise canceler counter 10H	TCNT10H	R/W	H'00	H'FFFFFF6DE	8	
	Noise canceler register 10	NCR10	R/W	H'FF	H'FFFFFF6E0		
	Timer I/O control register 10	TIOR10	R/W	H'00	H'FFFFFF6E2		
	Timer control register 10	TCR10	R/W	H'00	H'FFFFFF6E4		

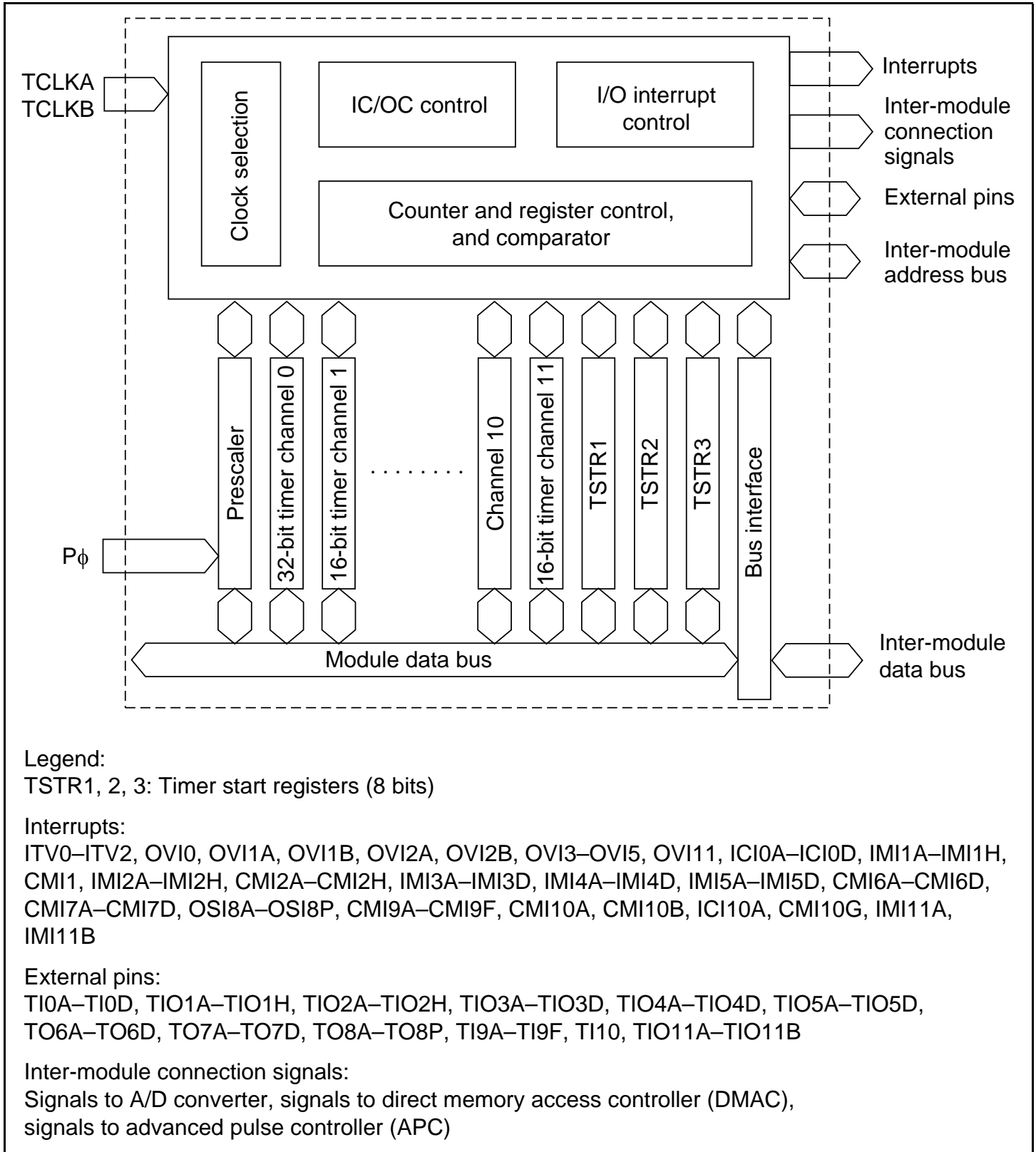
**Table 11.3 ATU-II Registers (cont)**

Channel	Name	Abbrevia- tion	R/W	Initial Value	Address	Access Size (Bits)	Section No.
10	Correction counter clear register 10	TCCLR10	R/W	H'0000	H'FFFFFF6E6	16	11.2.26
	Timer status register 10	TSR10	R/(W)*	H'0000	H'FFFFFF6E8		
	Timer interrupt enable register 10	TIER10	R/W	H'0000	H'FFFFFF6EA		
11	Free-running counter 11	TCNT11	R/W	H'0000	H'FFFFFF5C0	16	11.2.15
	General register 11A	GR11A	R/W	H'FFFF	H'FFFFFF5C2		11.2.20
	General register 11B	GR11B	R/W	H'FFFF	H'FFFFFF5C4		
	Timer I/O control register 11	TIOR11	R/W	H'00	H'FFFFFF5C6	8	11.2.4
	Timer control register 11	TCR11	R/W	H'00	H'FFFFFF5C8		11.2.3
	Timer status register 11	TSR11	R/(W)*	H'0000	H'FFFFFF5CA	16	11.2.5
	Timer interrupt enable register 11	TIER11	R/W	H'0000	H'FFFFFF5CC		11.2.6

Note: \* Only a 0 write after a read is enabled.

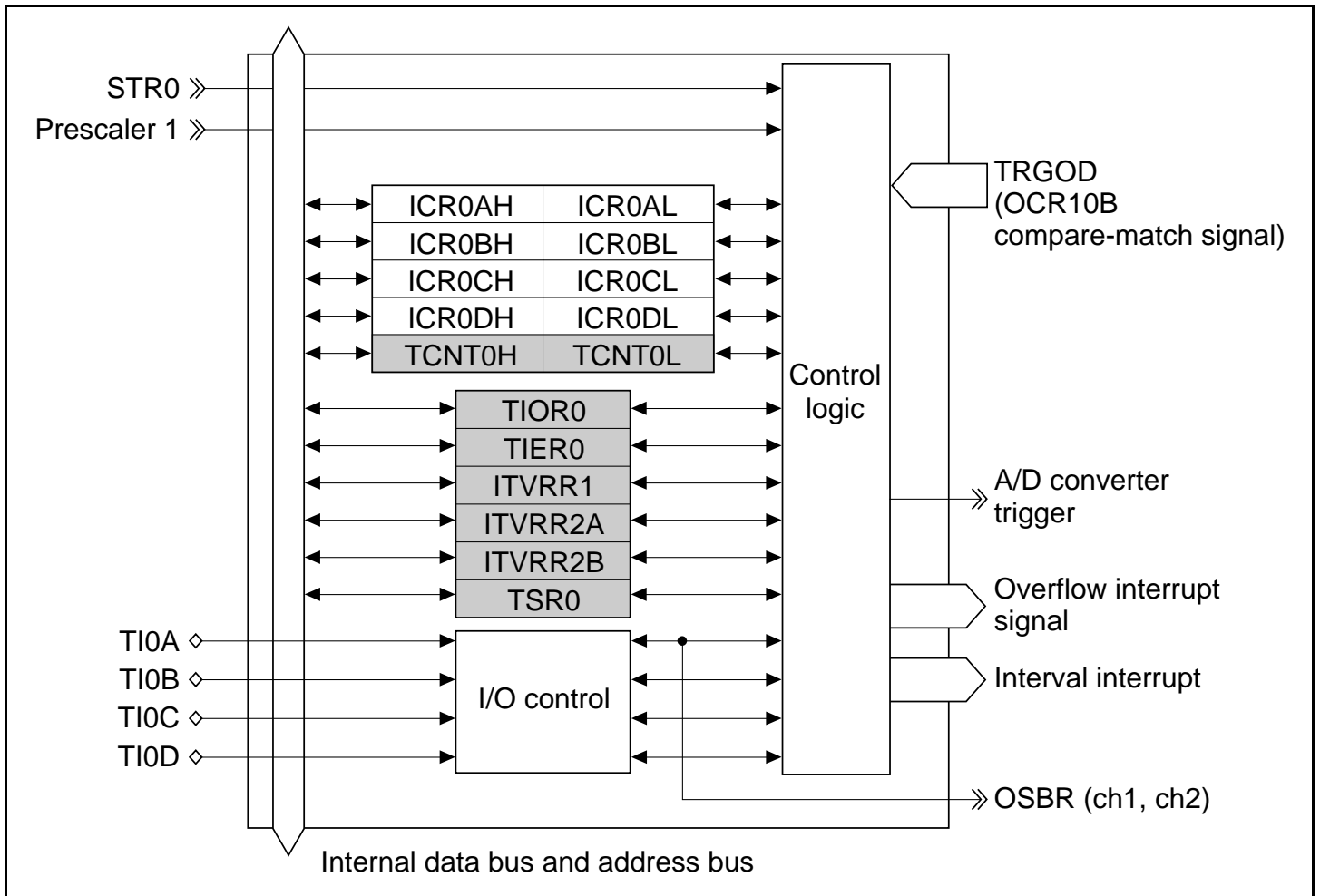
## 11.1.4 Block Diagrams

**Overall Block Diagram of ATU-II:** Figure 11.1 shows an overall block diagram of the ATU-II.



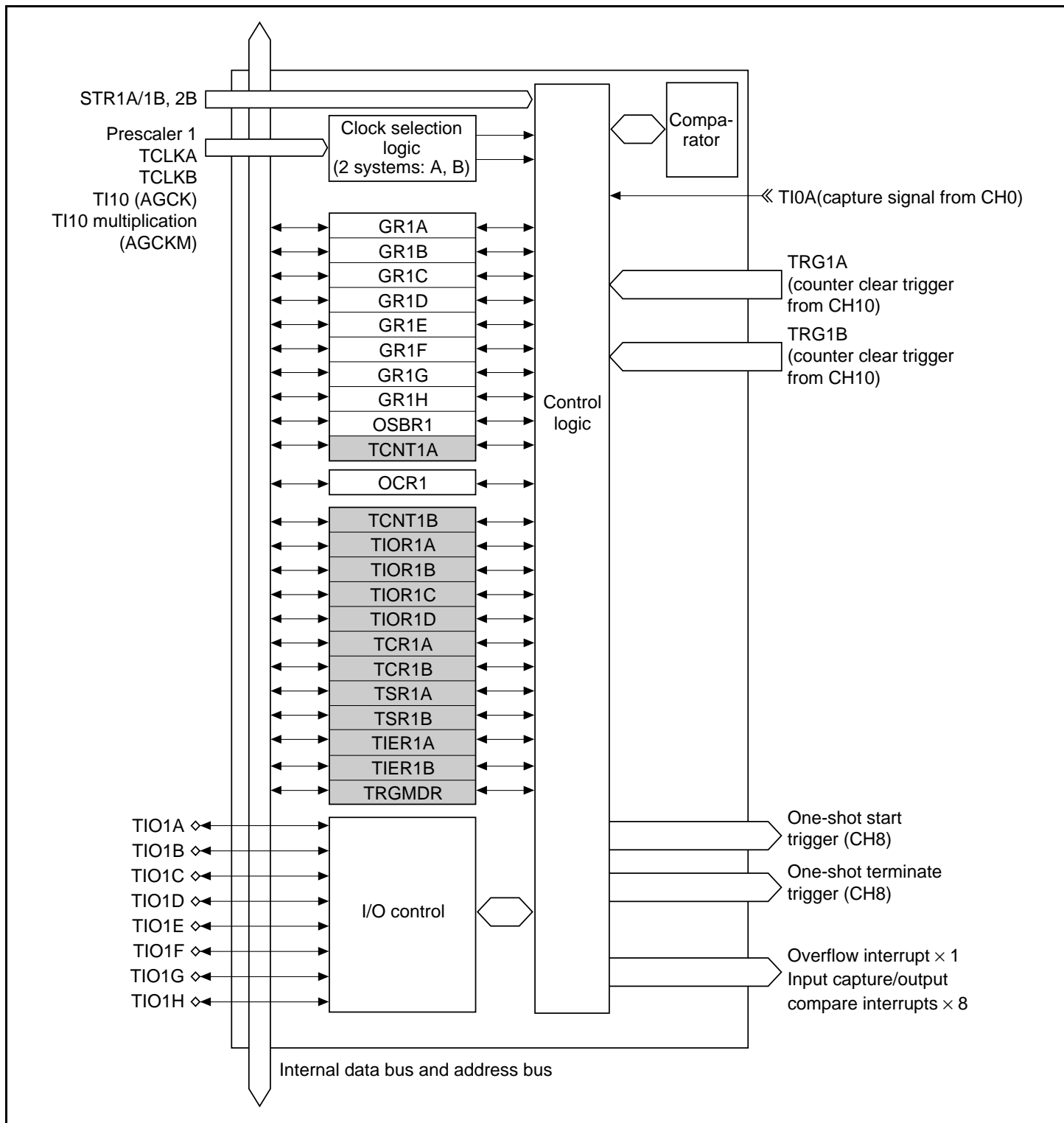
**Figure 11.1 Overall Block Diagram of ATU-II**

**Block Diagram of Channel 0:** Figure 11.2 shows a block diagram of ATU-II channel 0.



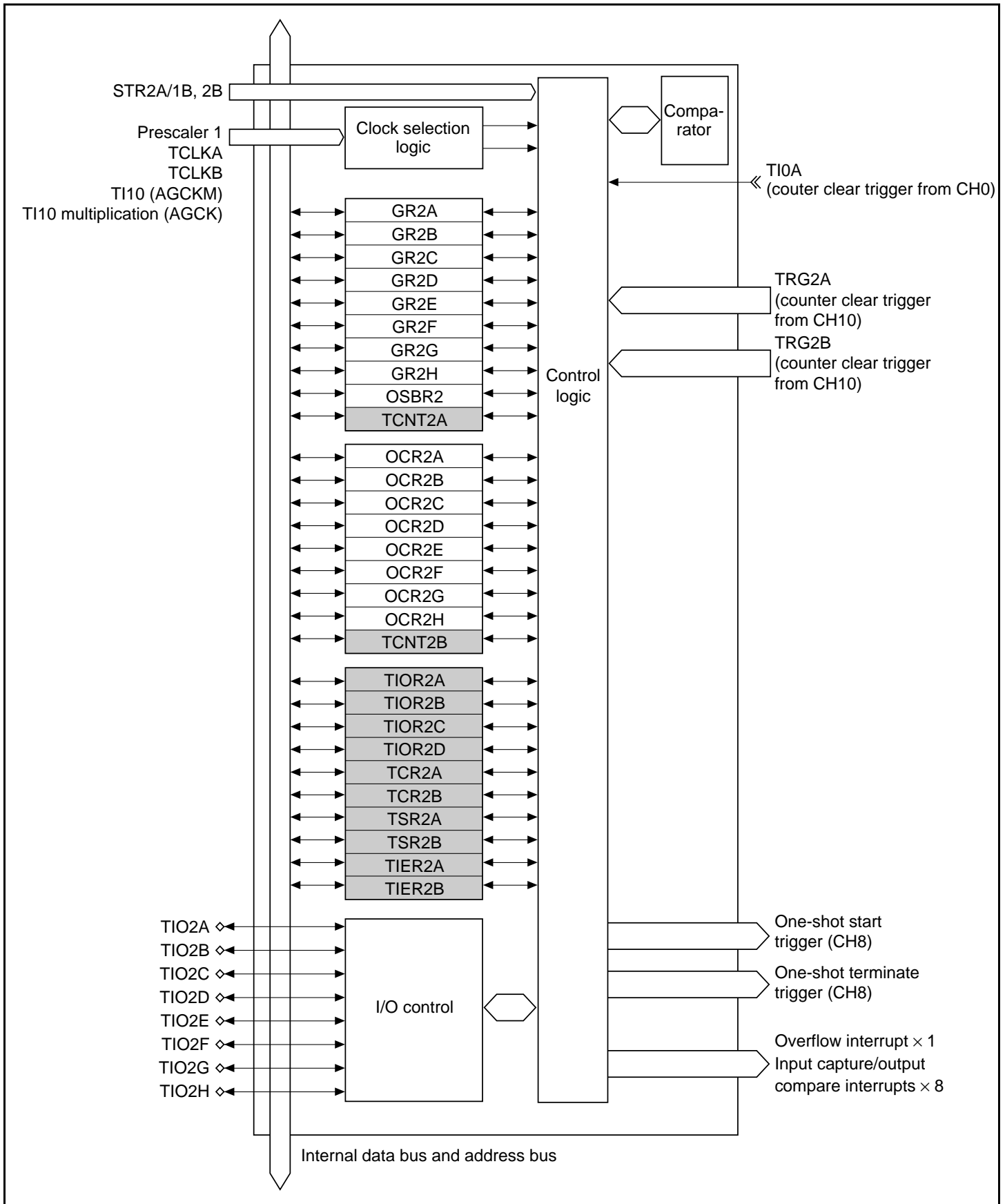
**Figure 11.2 Block Diagram of Channel 0**

**Block Diagram of Channel 1:** Figure 11.3 shows a block diagram of ATU-II channel 1.



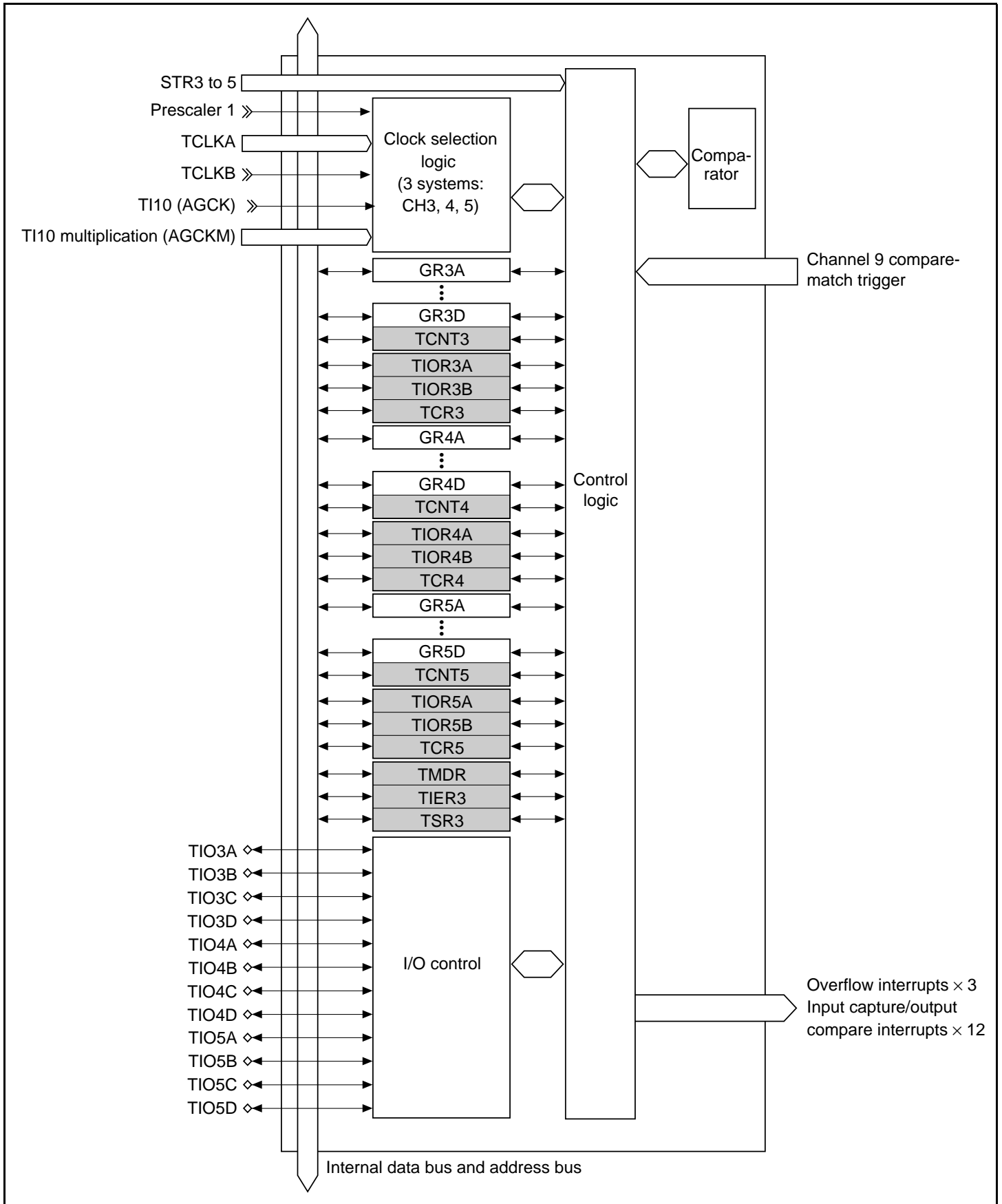
**Figure 11.3 Block Diagram of Channel 1**

**Block Diagram of Channel 2:** Figure 11.4 shows a block diagram of ATU-II channel 2.



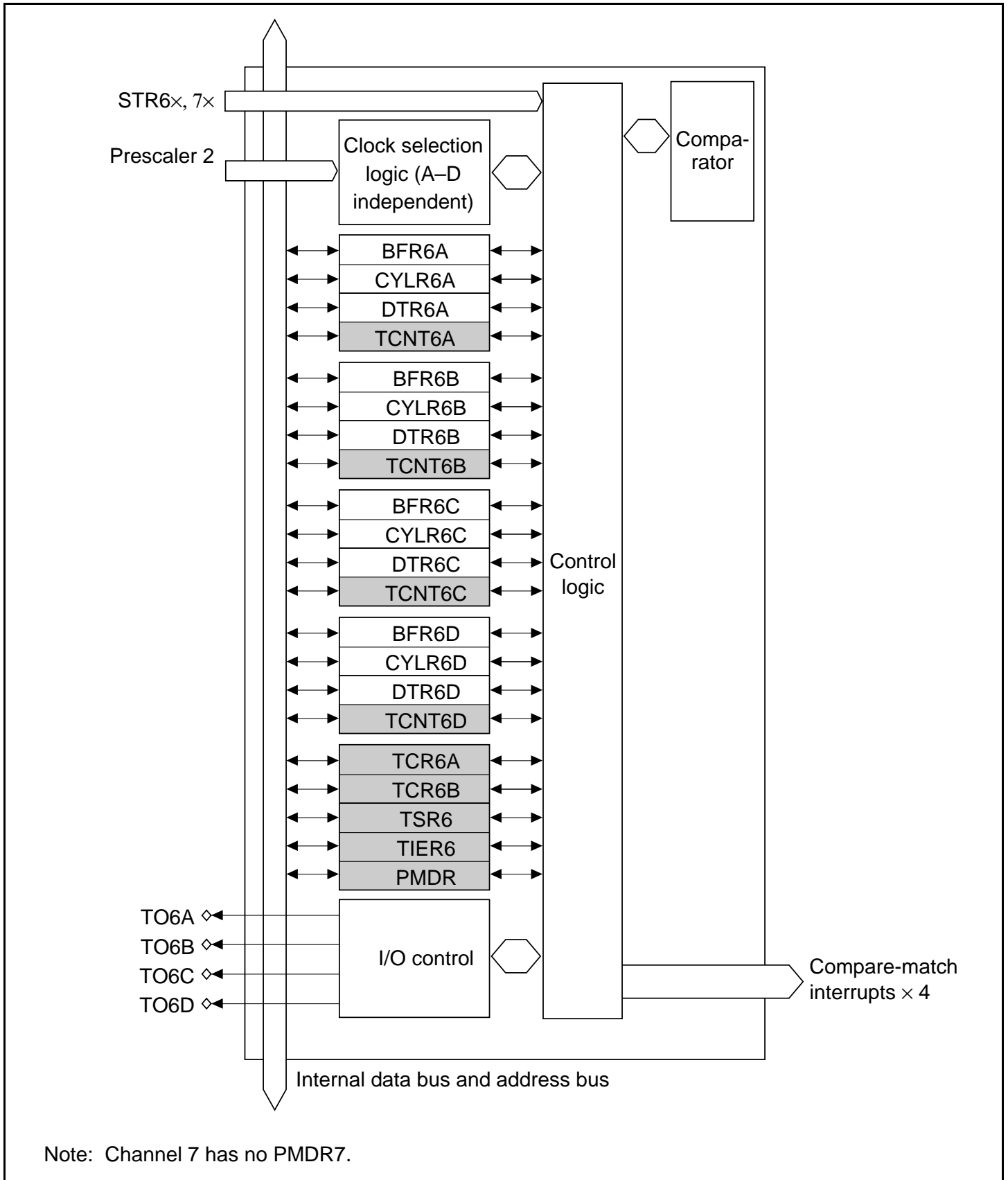
**Figure 11.4 Block Diagram of Channel 2**

**Block Diagram of Channels 3 to 5:** Figure 11.5 shows a block diagram of ATU-II channels 3, 4, and 5.



**Figure 11.5 Block Diagram of Channels 3 to 5**

**Block Diagram of Channels 6 and 7:** Figure 11.6 shows a block diagram of ATU-II channels 6 and 7.



**Figure 11.6 Block Diagram of Channel 6 (Same Configuration for Channel 7)**



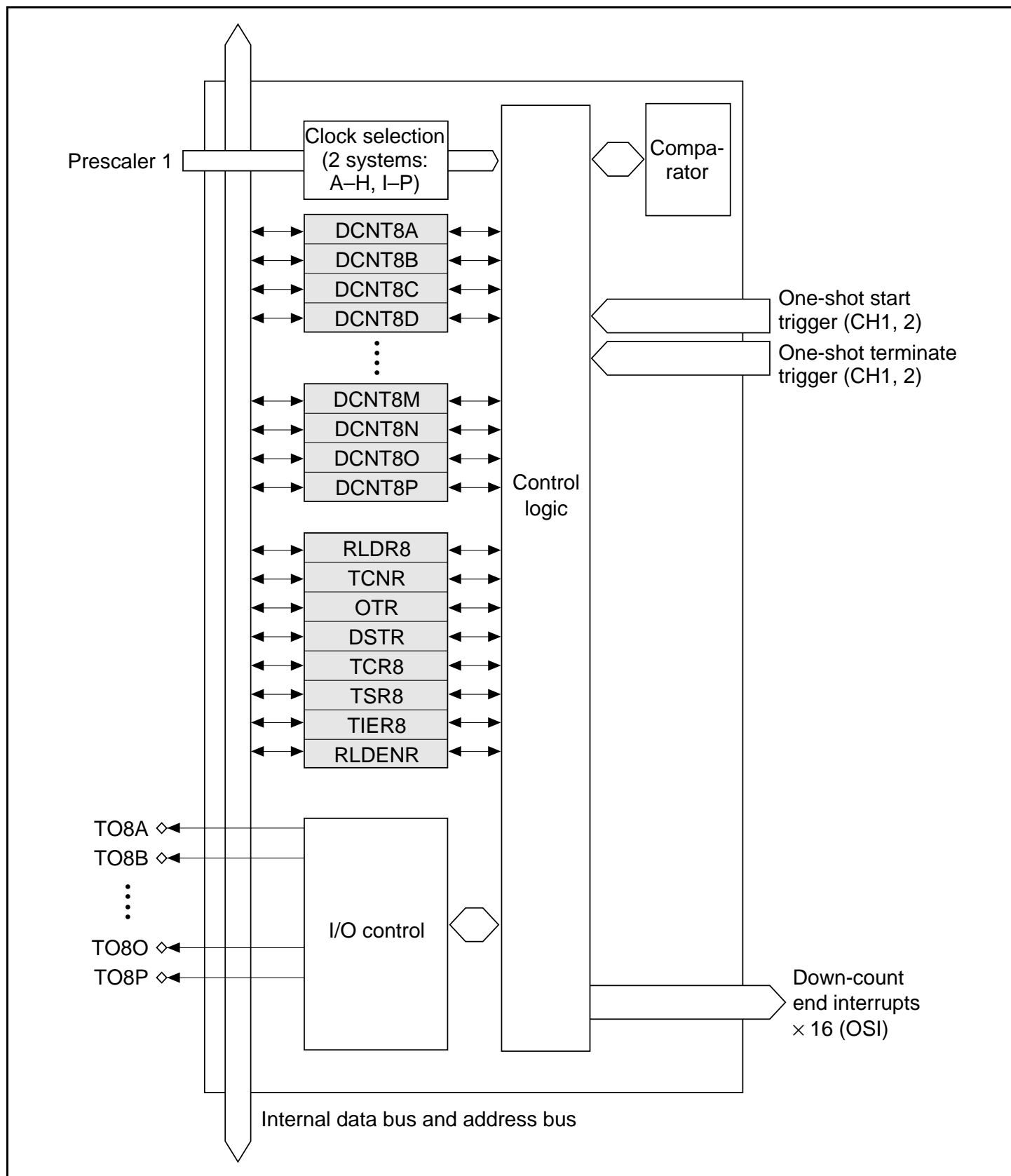
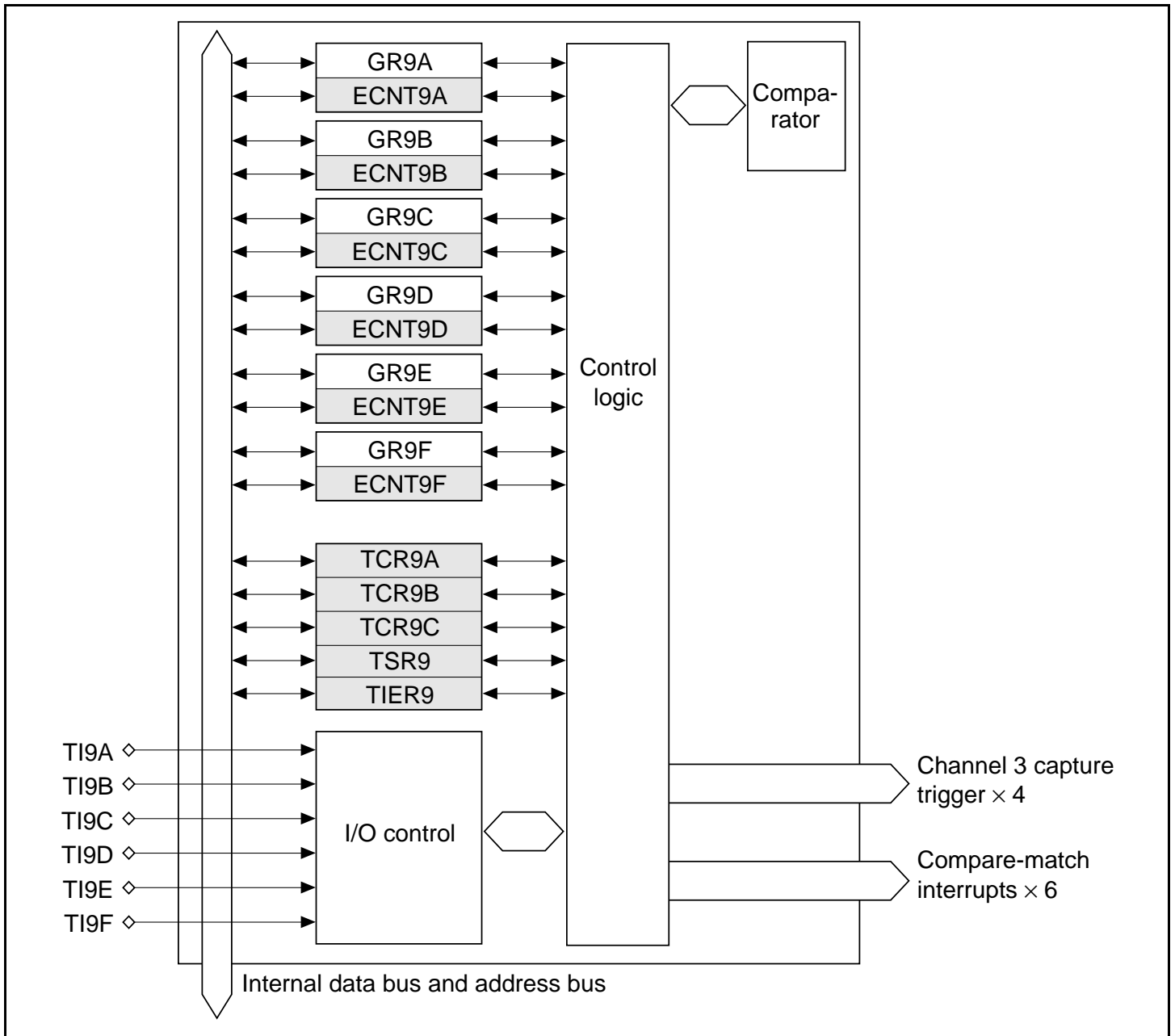


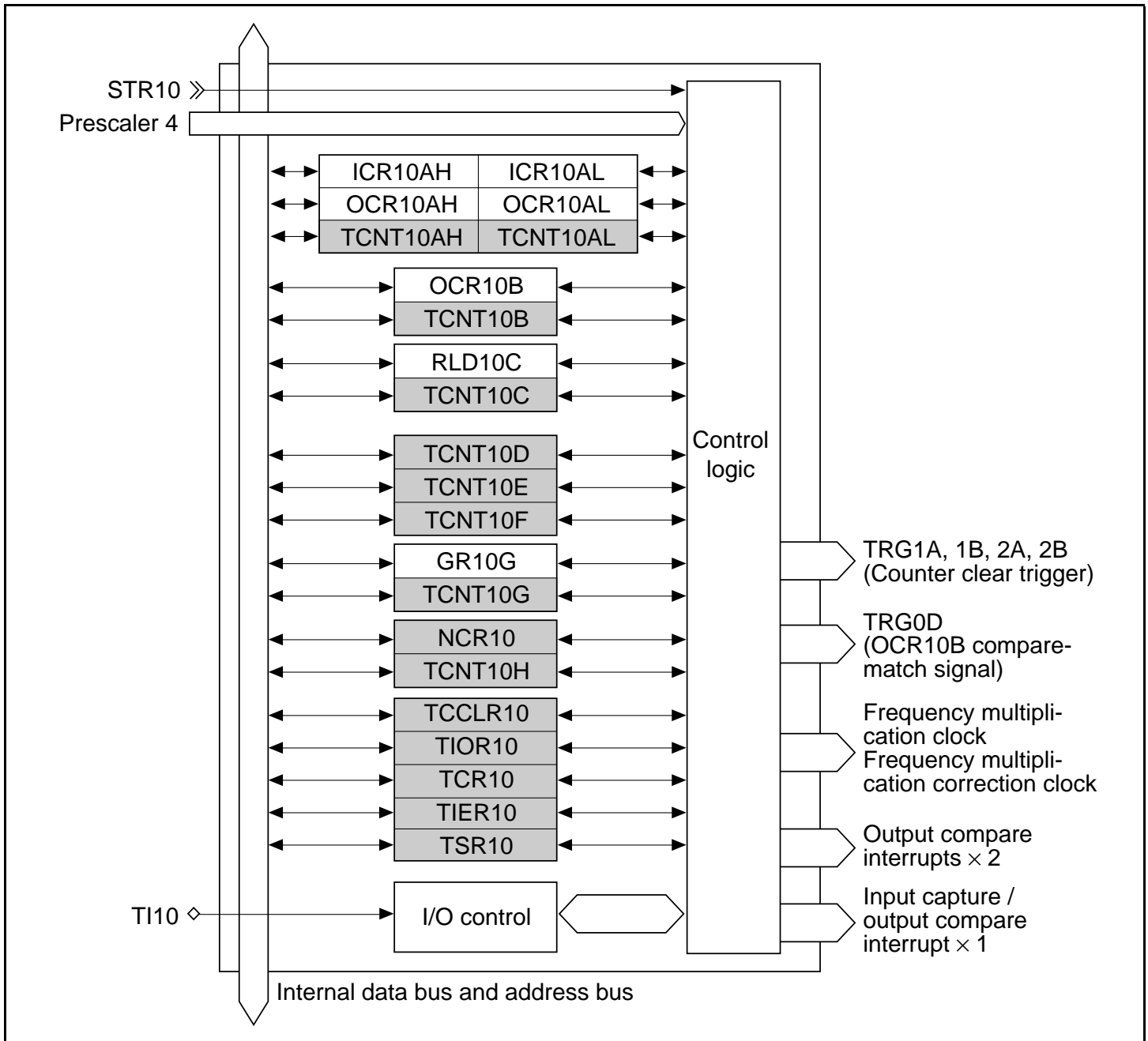
Figure 11.7 Block Diagram of Channel 8

**Block Diagram of Channel 9:** Figure 11.8 shows a block diagram of ATU-II channel 9.



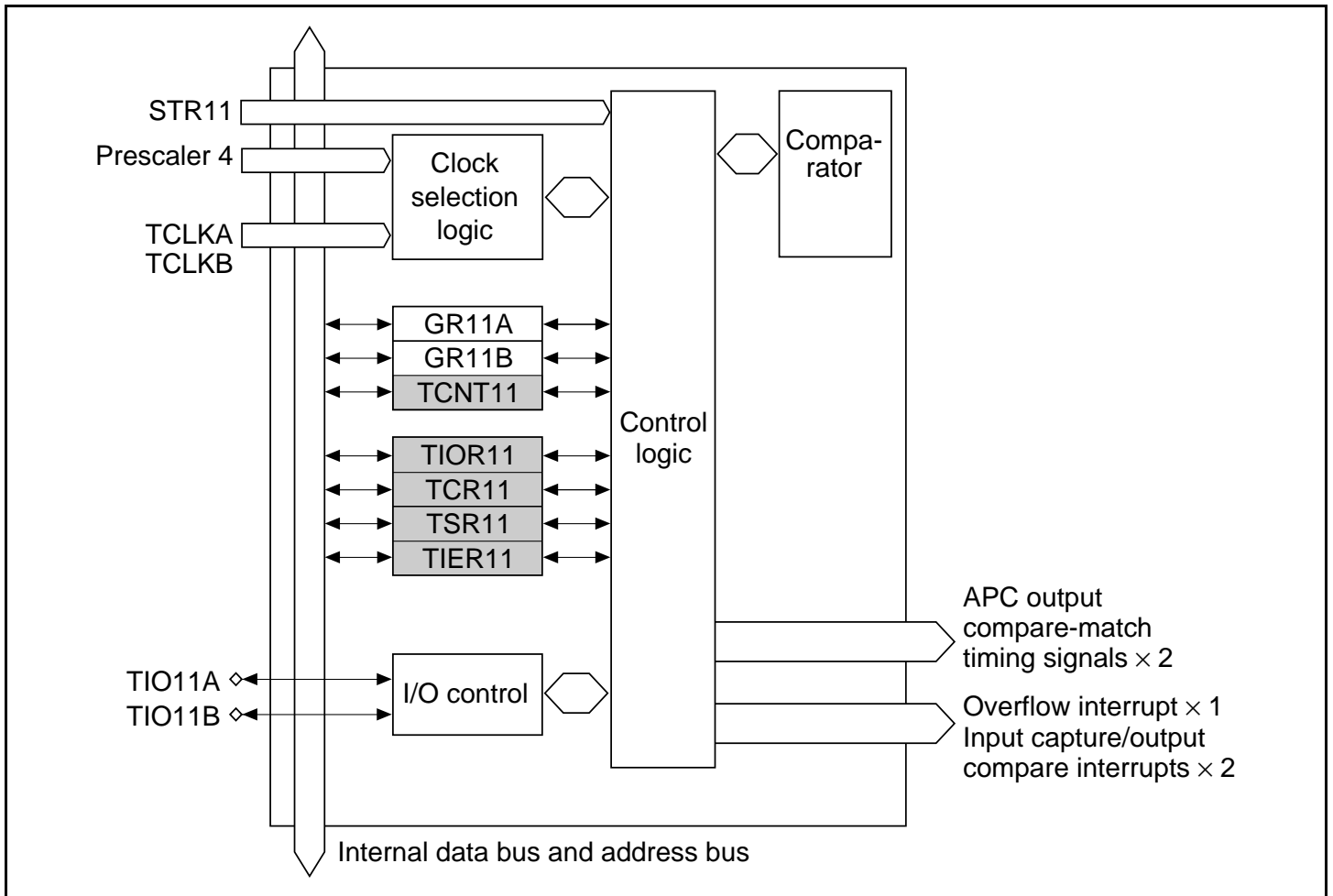
**Figure 11.8 Block Diagram of Channel 9**

**Block Diagram of Channel 10:** Figure 11.9 shows a block diagram of ATU-II channel 10.



**Figure 11.9 Block Diagram of Channel 10**

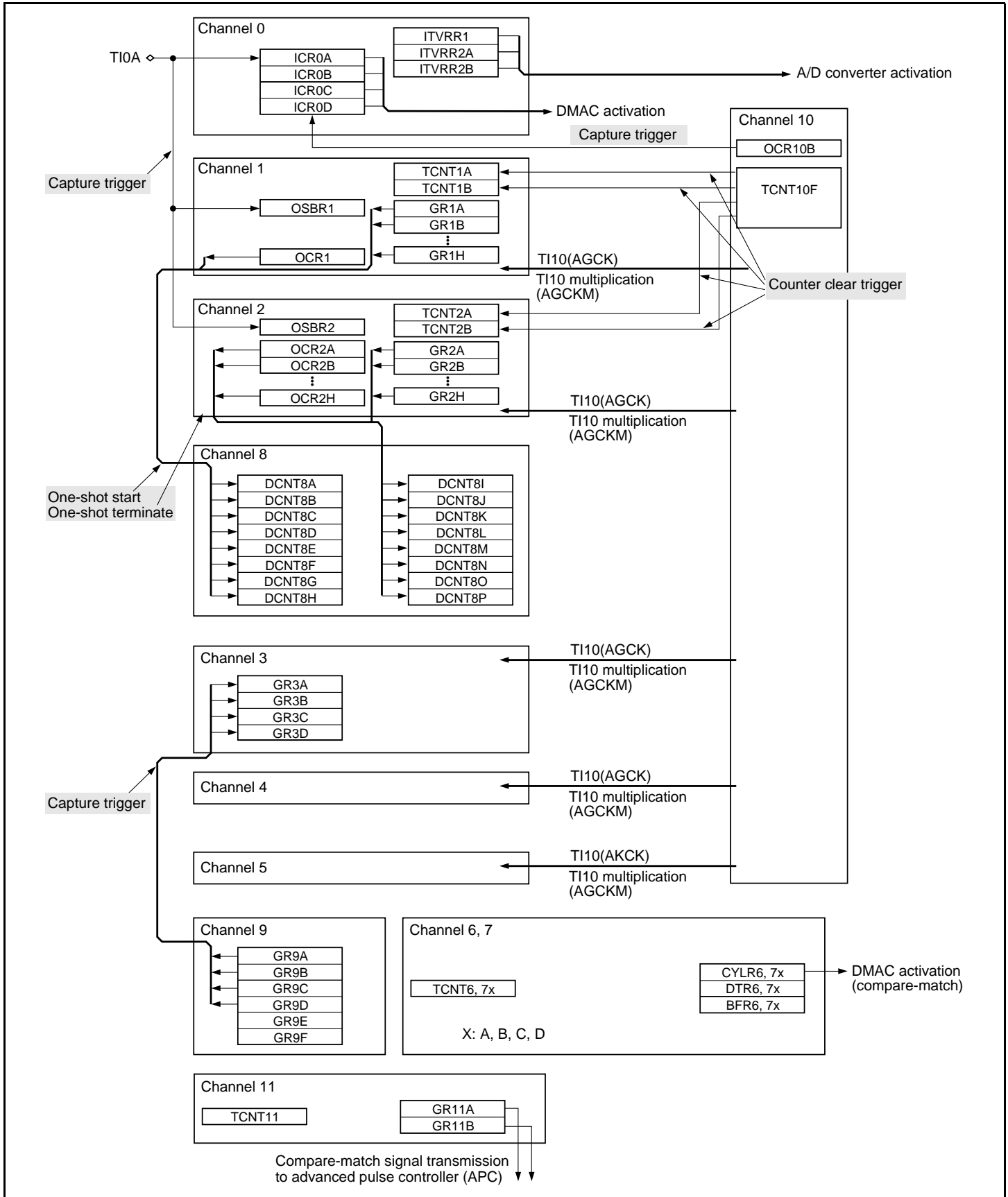
**Block Diagram of Channel 11:** Figure 11.10 shows a block diagram of ATU-II channel 11.



**Figure 11.10 Block Diagram of Channel 11**

## 11.1.5 Inter-Channel and Inter-Module Signal Communication Diagram

Figure 11.11 shows the connections between channels and between modules in the ATU-II.



**Figure 11.11 Inter-Module Communication Signals**

## 11.1.6 Prescaler Diagram

Figure 11.12 shows a diagram of the ATU-II prescalers.

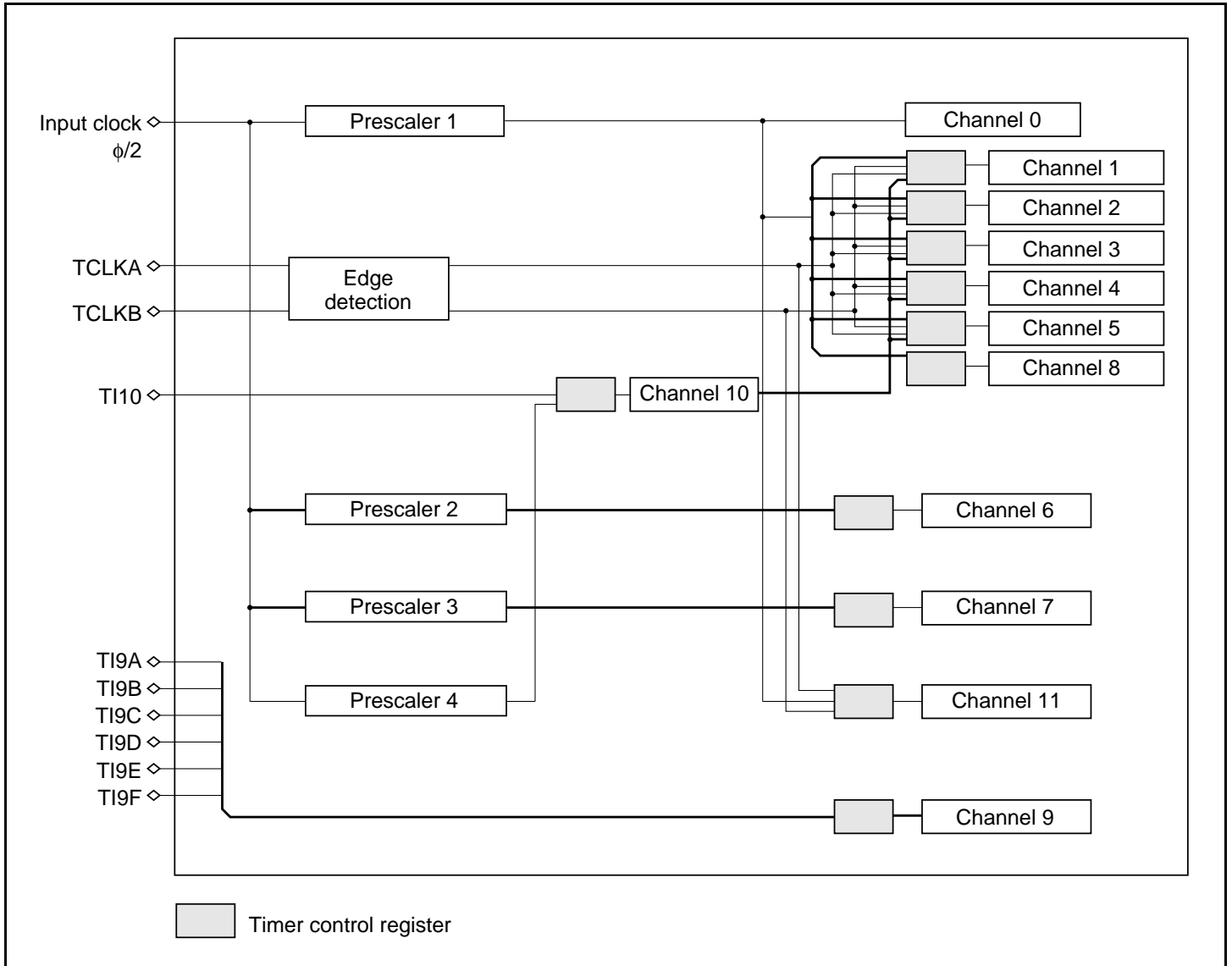


Figure 11.12 Prescaler Diagram

## 11.2 Register Descriptions

### 11.2.1 Timer Start Registers (TSTR)

The timer start registers (TSTR) are 8-bit registers. The ATU-II has three TSTR registers.

Channel	Abbreviation	Function
0, 1, 2, 3, 4, 5, 10	TSTR1	Free-running counter operation/stop setting
6, 7	TSTR2	
11	TSTR3	

#### Timer Start Register 1 (TSTR1)

Bit:	7	6	5	4	3	2	1	0
	STR10	STR5	STR4	STR3	STR1B, 2B	STR2A	STR1A	STR0
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

TSTR1 is an 8-bit readable/writable register that starts and stops the free-running counter (TCNT) in channels 0 to 5 and 10.

TSTR1 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Counter Start 10 (STR10): Starts and stops channel 10 counters (TCNT10A, 10C, 10D, 10E, 10F, and 10G). TCNT10B and 10H are not stopped.

Bit 7: STR10	Description
0	TCNT10 is halted (Initial value)
1	TCNT10 counts

- Bit 6—Counter Start 5 (STR5): Starts and stops free-running counter 5 (TCNT5).

Bit 6: STR5	Description
0	TCNT5 is halted (Initial value)
1	TCNT5 counts

- Bit 5—Counter Start 4 (STR4): Starts and stops free-running counter 4 (TCNT4).

Bit 5: STR4	Description
0	TCNT4 is halted (Initial value)
1	TCNT4 counts

- Bit 4—Counter Start 3 (STR3): Starts and stops free-running counter 3 (TCNT3).

Bit 4: STR3	Description
0	TCNT3 is halted (Initial value)
1	TCNT3 counts

- Bit 3—Counter Start 1B, 2B (STR1B, STR2B): Starts and stops free-running counters 1B and 2B (TCNT1B, TCNT2B).

Bit 3: STR1B, STR2B	Description
0	TCNT1B and TCNT2B are halted (Initial value)
1	TCNT1B and TCNT2B count

- Bit 2—Counter Start 2A (STR2A): Starts and stops free-running counter 2A (TCNT2A).

Bit 2: STR2A	Description
0	TCNT2A is halted (Initial value)
1	TCNT2A counts

- Bit 1—Counter Start 1A (STR1A): Starts and stops free-running counter 1A (TCNT1A).

Bit 1: STR1A	Description
0	TCNT1A is halted (Initial value)
1	TCNT1A counts

- Bit 0—Counter Start 0 (STR0): Starts and stops free-running counter 0 (TCNT0).

Bit 0: STR0	Description
0	TCNT0 is halted (Initial value)
1	TCNT0 counts



## Timer Start Register 2 (TSTR2)

Bit:	7	6	5	4	3	2	1	0
	STR7D	STR7C	STR7B	STR7A	STR6D	STR6C	STR6B	STR6A
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

TSTR2 is an 8-bit readable/writable register that starts and stops the free-running counter (TCNT) in channels 6 and 7.

TSTR2 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Counter Start 7D (STR7D): Starts and stops free-running counter 7D (TCNT7D).

Bit 7: STR7D	Description
0	TCNT7D is halted (Initial value)
1	TCNT7D counts

- Bit 6—Counter Start 7C (STR7C): Starts and stops free-running counter 7C (TCNT7C).

Bit 6: STR7C	Description
0	TCNT7C is halted (Initial value)
1	TCNT7C counts

- Bit 5—Counter Start 7B (STR7B): Starts and stops free-running counter 7B (TCNT7B).

Bit 5: STR7B	Description
0	TCNT7B is halted (Initial value)
1	TCNT7B counts

- Bit 4—Counter Start 7A (STR7A): Starts and stops free-running counter 7A (TCNT7A).

Bit 4: STR7A	Description
0	TCNT7A is halted (Initial value)
1	TCNT7A counts

- Bit 3—Counter Start 6D (STR6D): Starts and stops free-running counter 6D (TCNT6D).

Bit 3: STR6D	Description
0	TCNT6D is halted (Initial value)
1	TCNT6D counts

- Bit 2—Counter Start 6C (STR6C): Starts and stops free-running counter 6C (TCNT6C).

Bit 2: STR6C	Description
0	TCNT6C is halted (Initial value)
1	TCNT6C counts

- Bit 1—Counter Start 6B (STR6B): Starts and stops free-running counter 6B (TCNT6B).

Bit 1: STR6B	Description
0	TCNT6B is halted (Initial value)
1	TCNT6B counts

- Bit 0—Counter Start 6A (STR6A): Starts and stops free-running counter 6A (TCNT6A).

Bit 0: STR6A	Description
0	TCNT6A is halted (Initial value)
1	TCNT6A counts

### Timer Start Register 3 (TSTR3)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	STR11
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

TSTR3 is an 8-bit readable/writable register that starts and stops the free-running counter (TCNT11) in channel 11.

TSTR3 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always be 0.

- Bit 0—Counter Start 11 (STR11): Starts and stops free-running counter 11 (TCNT11).

Bit 0: STR11	Description
0	TCNT11 is halted (Initial value)
1	TCNT11 counts

## 11.2.2 Prescaler Registers (PSCR)

The prescaler registers (PSCR) are 8-bit registers. The ATU-II has four PSCR registers.

Channel	Abbreviation	Function
0, 1, 2, 3, 4, 5, 8, 11	PSCR1	Prescaler setting for respective channels
6	PSCR2	
7	PSCR3	
10	PSCR4	

PSCR<sub>x</sub> is an 8-bit writable register that enables the first-stage counter clock  $\phi'$  input to each channel to be set to any value from  $P\phi/1$  to  $P\phi/32$ .

Bit:	7	6	5	4	3	2	1	0
	—	—	—	PSCxE	PSCxD	PSCxC	PSCxB	PSCxA
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

$x = 1$  to 4

Input counter clock  $\phi'$  is determined by setting PSCxA to PSCxE:  $\phi'$  is  $P\phi/1$  when the set value is H'00, and  $P\phi/32$  when H'1F.

PSCR<sub>x</sub> is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

The internal clock  $\phi'$  set with this register can undergo further second-stage scaling to create clock  $\phi''$  for channels 1 to 8 and 11, the setting being made in the timer control register (TCR).

- Bits 7 to 5—Reserved: These bits cannot be modified.
- Bits 4 to 0—Prescaler (PSCxE, PSCxD, PSCxC, PSCxB, PSCxA): These bits specify frequency division of first-stage counter clock  $\phi'$  input to the corresponding channel.

### 11.2.3 Timer Control Registers (TCR)

The timer control registers (TCR) are 8-bit registers. The ATU-II has 16 TCR registers: two each for channels 1 and 2, one each for channels 3, 4, 5, 8, and 11, two each for channels 6 and 7, and three for channel 9. For details of channel 10, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
1	TCR1A, TCR1B	Internal clock/external clock/TI10 input clock selection
2	TCR2A, TCR2B	
3	TCR3	
4	TCR4	
5	TCR5	
6	TCR6A, TCR6B	Internal clock selection
7	TCR7A, TCR7B	
8	TCR8	
9	TCR9A, TCR9B, TCR9C	External clock selection/setting of channel 3 trigger in event of compare-match
11	TCR11	Internal clock/external clock selection

Each TCR is an 8-bit readable/writable register that selects whether an internal clock or external clock is used for channels 1 to 5 and 11. For channels 6 to 8, TCR selects an internal clock, and for channel 9, an external clock.

When an internal clock is selected, TCR selects the value of  $\phi''$  further scaled from clock  $\phi'$  scaled with prescaler register (PSCR). Scaled clock  $\phi''$  can be selected, for channels 1 to 8 and 11 only, from  $\phi'$ ,  $\phi'/2$ ,  $\phi'/4$ ,  $\phi'/8$ ,  $\phi'/16$ , and  $\phi'/32$  (only  $\phi'$  is available for channel 0). Edge detection is performed on the rising edge.

When an external clock is selected, TCR selects whether TCLKA, TCLKB (channels 1 to 5 and 11 only), TI10 pin input (channels 1 to 5 only), or a TI10 pin input multiplied clock (channels 1 to 5 only) is used, and also performs edge selection.

Each TCR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

## Timer Control Registers 1A, 1B, 2A, 2B (TCR1A, TCR1B, TCR2A, TCR2B)

### TCR1A, TCR2A

Bit:	7	6	5	4	3	2	1	0
	—	—	CKEGA1	CKEGA0	CKSELA3	CKSELA2	CKSELA1	CKSELA0
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

### TCR1B, TCR2B

Bit:	7	6	5	4	3	2	1	0
	—	—	CKEGB1	CKEGB0	CKSELB3	CKSELB2	CKSELB1	CKSELB0
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

- Bits 7 and 6—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 5 and 4—Clock Edge 1 and 0 (CKEG<sub>x</sub>1, CKEG<sub>x</sub>0): These bits select the count edge(s) for external clock TCLKA and TCLKB input.

Bit 5: CKEG <sub>x</sub> 1	Bit 4: CKEG <sub>x</sub> 0	Description
0	0	Rising edges counted (Initial value)
	1	Falling edges counted
1	0	Both rising and falling edges counted
	1	Count disabled

x = A or B

- Bits 3 to 0—Clock Select A3 to A0, B3 to B0 (CKSELA3 to CKSELA0, CKSELB3 to CKSELB0): These bits select whether an internal clock or external clock is used.  
When an internal clock is selected, scaled clock  $\phi''$  is selected from  $\phi'$ ,  $\phi'/2$ ,  $\phi'/4$ ,  $\phi'/8$ ,  $\phi'/16$ , and  $\phi'/32$ .  
When an external clock is selected, TCLKA, TCLKB, TI10 pin input, or a TI10 pin input multiplied clock is selected.  
When TI10 pin input and TI10 pin input clock multiplication are selected, set CKEG1 and CKEG0 in TCR10 so that TI10 input is possible.

Bit 3: CKSELx3	Bit 2: CKSELx2	Bit 1: CKSELx1	Bit 0: CKSELx0	Description
0	0	0	0	Internal clock $\phi''$ : counting on $\phi'$ (Initial value)
			1	Internal clock $\phi''$ : counting on $\phi'/2$
		1	0	Internal clock $\phi''$ : counting on $\phi'/4$
			1	Internal clock $\phi''$ : counting on $\phi'/8$
	1	0	0	Internal clock $\phi''$ : counting on $\phi'/16$
			1	Internal clock $\phi''$ : counting on $\phi'/32$
		1	0	External clock: counting on TCLKA pin input
			1	External clock: counting on TCLKB pin input
1	0	0	Counting on TI10 pin input (AGCK)	
		1	Counting on multiplied (corrected)(AGCKM) TI10 pin input clock	
	1	*	Setting prohibited	
		*	Setting prohibited	

x = A or B

\*: Don't care

### Timer Control Registers 3 to 5 (TCR3, TCR4, TCR5)

Bit:	7	6	5	4	3	2	1	0
	—	—	CKEG1	CKEG0	CKSEL3	CKSEL2	CKSEL1	CKSEL0
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

- Bits 7 and 6—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 5 and 4—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select the count edge(s) for external clock TCLKA and TCLKB input.

Bit 5: CKEG1	Bit 4: CKEG0	Description
0	0	Rising edges counted (Initial value)
	1	Falling edges counted
1	0	Both rising and falling edges counted
	1	Count disabled

- Bits 3 to 0—Clock Select 3 to 0 (CKSEL3 to CKSEL0): These bits select whether an internal clock or external clock is used.

When an internal clock is selected, scaled clock  $\phi''$  is selected from  $\phi'$ ,  $\phi'/2$ ,  $\phi'/4$ ,  $\phi'/8$ ,  $\phi'/16$ , and  $\phi'/32$ .

When an external clock is selected, TCLKA, TCLKB, TI10 pin input, or a TI10 pin input multiplied clock is selected.

When TI10 pin input and TI10 pin input clock multiplication are selected, set CKEG1 and CKEG0 in TCR10 so that TI10 input is possible.

Bit 3: CKSEL3	Bit 2: CKSEL2	Bit 1: CKSEL1	Bit 0: CKSEL0	Description
0	0	0	0	Internal clock $\phi''$ : counting on $\phi'$ (Initial value)
			1	Internal clock $\phi''$ : counting on $\phi'/2$
		1	0	Internal clock $\phi''$ : counting on $\phi'/4$
			1	Internal clock $\phi''$ : counting on $\phi'/8$
	1	0	0	Internal clock $\phi''$ : counting on $\phi'/16$
			1	Internal clock $\phi''$ : counting on $\phi'/32$
		1	0	External clock: counting on TCLKA pin input
			1	External clock: counting on TCLKB pin input
1	0	0	0	Counting on TI10 pin input (AGCK)
			1	Counting on multiplied (corrected)(AGCKM) TI10 pin input clock
		1	*	Setting prohibited
	1	*	*	Setting prohibited

\*: Don't care

## Timer Control Registers 6A, 6B, 7A, 7B (TCR6A, TCR6B, TCR7A, TCR7B)

### TCR6A, TCR7A

Bit:	7	6	5	4	3	2	1	0
	—	CKSELB2	CKSELB1	CKSELB0	—	CKSELA2	CKSELA1	CKSELA0
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

### TCR6B, TCR7B

Bit:	7	6	5	4	3	2	1	0
	—	CKSELD2	CKSELD1	CKSELD0	—	CKSELC2	CKSELC1	CKSELC0
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

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bits 6 to 4—Clock Select B2 to B0, D2 to D0 (CKSELB2 to CKSELB0, CKSELD2 to CKSELD0): These bits select clock  $\phi''$ , scaled from the internal clock source, from  $\phi'$ ,  $\phi'/2$ ,  $\phi'/4$ ,  $\phi'/8$ ,  $\phi'/16$ , and  $\phi'/32$ .

Bit 6: CKSELx2	Bit 5: CKSELx1	Bit 4: CKSELx0	Description
0	0	0	Internal clock $\phi''$ : counting on $\phi'$ (Initial value)
		1	Internal clock $\phi''$ : counting on $\phi'/2$
	1	0	Internal clock $\phi''$ : counting on $\phi'/4$
		1	Internal clock $\phi''$ : counting on $\phi'/8$
1	0	0	Internal clock $\phi''$ : counting on $\phi'/16$
		1	Internal clock $\phi''$ : counting on $\phi'/32$
	1	0	Setting prohibited
		1	Setting prohibited

x = B or D

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bits 2 to 0—Clock Select A2 to A0, C2 to C0 (CKSELA2 to CKSELA0, CKSELC2 to CKSELC0): These bits select clock  $\phi''$ , scaled from the internal clock source, from  $\phi'$ ,  $\phi'/2$ ,  $\phi'/4$ ,  $\phi'/8$ ,  $\phi'/16$ , and  $\phi'/32$ .



Bit 2: CKSELx2	Bit 1 CKSELx1	Bit 0 CKSELx0	Description
0	0	0	Internal clock $\phi''$ : counting on $\phi'$ (Initial value)
		1	Internal clock $\phi''$ : counting on $\phi'/2$
	1	0	Internal clock $\phi''$ : counting on $\phi'/4$
		1	Internal clock $\phi''$ : counting on $\phi'/8$
1	0	0	Internal clock $\phi''$ : counting on $\phi'/16$
		1	Internal clock $\phi''$ : counting on $\phi'/32$
	1	0	Setting prohibited
		1	Setting prohibited

x = A or C

### Timer Control Register 8 (TCR8)

Bit:	7	6	5	4	3	2	1	0
	—	CKSELB2	CKSELB1	CKSELB0	—	CKSELA2	CKSELA1	CKSELA0
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

The CKSELAx bits relate to DCNT8A to DCNT8H, and the CKSELBx bits relate to DCNT8I to DCNT8P.

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bits 6 to 4—Clock Select B2 to B0 (CKSELB2 to CKSELB0): These bits, relating to counters DCNT8I to DCNT8P, select clock  $\phi''$ , scaled from the internal clock source, from  $\phi'$ ,  $\phi'/2$ ,  $\phi'/4$ ,  $\phi'/8$ ,  $\phi'/16$ , and  $\phi'/32$ .

Bit 6: CKSELB2	Bit 5: CKSELB1	Bit 4: CKSELB0	Description
0	0	0	Internal clock $\phi''$ : counting on $\phi'$ (Initial value)
		1	Internal clock $\phi''$ : counting on $\phi'/2$
	1	0	Internal clock $\phi''$ : counting on $\phi'/4$
		1	Internal clock $\phi''$ : counting on $\phi'/8$
1	0	0	Internal clock $\phi''$ : counting on $\phi'/16$
		1	Internal clock $\phi''$ : counting on $\phi'/32$
	1	0	Setting prohibited
		1	Setting prohibited

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.

- Bits 2 to 0—Clock Select A2 to A0 (CKSELA2 to CKSELA0): These bits, relating to counters DCNT8A to DCNT8H, select clock  $\phi''$ , scaled from the internal clock source, from  $\phi'$ ,  $\phi'/2$ ,  $\phi'/4$ ,  $\phi'/8$ ,  $\phi'/16$ , and  $\phi'/32$ .

Bit 2: CKSELA2	Bit 1: CKSELA1	Bit 0: CKSELA0	Description
0	0	0	Internal clock $\phi''$ : counting on $\phi'$ (Initial value)
		1	Internal clock $\phi''$ : counting on $\phi'/2$
	1	0	Internal clock $\phi''$ : counting on $\phi'/4$
		1	Internal clock $\phi''$ : counting on $\phi'/8$
1	0	0	Internal clock $\phi''$ : counting on $\phi'/16$
		1	Internal clock $\phi''$ : counting on $\phi'/32$
	1	0	Setting prohibited
		1	Setting prohibited

## Timer Control Registers 9A, 9B, 9C (TCR9A, TCR9B, TCR9C)

### TCR9A

Bit:	7	6	5	4	3	2	1	0
	—	TRG3BEN	EGSELB1	EGSELB0	—	TRG3AEN	EGSELA1	EGSELA0
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

### TCR9B

Bit:	7	6	5	4	3	2	1	0
	—	TRG3DEN	EGSELD1	EGSELD0	—	TRG3CEN	EGSELC1	EGSELC0
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

### TCR9C

Bit:	7	6	5	4	3	2	1	0
	—	—	EGSELF1	EGSELF0	—	—	EGSELE1	EGSELE0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 6—Trigger Channel 3BEN, 3DEN (TRG3BEN, TRG3DEN): These bits select the channel 9 event counter compare-match signal channel 3 input capture trigger.

Bit 6: TRG3xEN	Description
0	Channel 3 input capture trigger in event of channel 9 compare-match (ECNT9x = GR9x) is disabled (Initial value)
1	Channel 3 input capture trigger in event of channel 9 compare-match (ECNT9x = GR9x) is enabled

x = B or D

- Bits 5 and 4—Edge Select B1, B0, D1, D0, F1, F0 (EGSELB1, EGSELB0, EGSELD1, EGSELD0, EGSELF1, EGSELF0): These bits select the event counter counted edge(s).

Bit 5: EGSELx1	Bit 4: EGSELx0	Description
0	0	Count disabled (Initial value)
	1	Rising edges counted
1	0	Falling edges counted
	1	Both rising and falling edges counted

x = B, D, or F

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—Trigger Channel 3AEN, 3CEN (TRG3AEN, TRG3CEN): These bits select the channel 9 event counter compare-match signal channel 3 input capture trigger.

Bit 2: TRG3xEN	Description
0	Channel 3 input capture trigger in event of channel 9 compare-match (ECNT9x = GR9x) is disabled (Initial value)
1	Channel 3 input capture trigger in event of channel 9 compare-match (ECNT9x = GR9x) is enabled

x = A or C

- Bits 1 and 0—Edge Select A1, A0, C1, C0, E1, E0 (EGSELA1, EGSELA0, EGSELC1, EGSELC0, EGSELE1, EGSELE0): These bits select the event counter counted edge(s).

Bit 1: EGSELx1	Bit 0: EGSELx0	Description
0	0	Count disabled (Initial value)
	1	Rising edges counted
1	0	Falling edges counted
	1	Both rising and falling edges counted

x = A, C, or E

### Timer Control Register 11 (TCR11)

Bit:	7	6	5	4	3	2	1	0
	—	—	CKEG1	CKEG0	—	CKSELA2	CKSELA1	CKSELA0
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

- Bits 7, 6, and 3—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 5 and 4—Edge Select: These bits select the event counter counted edge(s).

Bit 5: CKEG1	Bit 4: CKEG0	Description
0	0	Rising edges counted (Initial value)
	1	Falling edges counted
1	0	Both rising and falling edges counted
	1	Count disabled

- Bits 2 to 0—Clock Select A2 to A0 (CKSELA2 to CKSELA0): These bits select clock  $\phi''$ , scaled from the internal clock source, from  $\phi'$ ,  $\phi'/2$ ,  $\phi'/4$ ,  $\phi'/8$ ,  $\phi'/16$ , and  $\phi'/32$ .

Bit 2: CKSELA2	Bit 1: CKSELA1	Bit 0: CKSELA0	Description
0	0	0	Internal clock $\phi''$ : counting on $\phi'$ (Initial value)
		1	Internal clock $\phi''$ : counting on $\phi'/2$
	1	0	Internal clock $\phi''$ : counting on $\phi'/4$
		1	Internal clock $\phi''$ : counting on $\phi'/8$
1	0	0	Internal clock $\phi''$ : counting on $\phi'/16$
		1	Internal clock $\phi''$ : counting on $\phi'/32$
	1	0	External clock: counting on TCLKA pin input
		1	External clock: counting on TCLKB pin input

## 11.2.4 Timer I/O Control Registers (TIOR)

The timer I/O control registers (TIOR) are 8-bit registers. The ATU-II has 16 TIOR registers: one for channel 0, four each for channels 1 and 2, two each for channels 3 to 5, and one for channel 11. For details of channel 10, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
0	TIOR0	ICR0 edge detection setting
1	TIOR1A–1D	GR input capture/compare-match switching, edge detection/output value setting
2	TIOR2A–2D	
3	TIOR3A, TIOR3B	GR input capture/compare-match switching, edge detection/output value setting, TCNT3 to TCNT5 clear enable/disable setting
4	TIOR4A, TIOR4B	
5	TIOR5A, TIOR5B	
11	TIOR11	GR input capture/compare-match switching, edge detection/output value setting

Each TIOR is an 8-bit readable/writable register used to select the functions of dedicated input capture registers and general registers.

For dedicated input capture registers (ICR), TIOR performs edge detection setting.

For general registers (GR), TIOR selects use as an input capture register or output compare register, and performs edge detection setting. For channels 3 to 5, TIOR also selects enabling or disabling of free-running counter (TCNT) clearing in the event of a compare-match.

### Timer I/O Control Register 0 (TIOR0)

Bit:	7	6	5	4	3	2	1	0
	IO0D1	IO0D0	IO0C1	IO0C0	IO0B1	IO0B0	IO0A1	IO0A0
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

TIOR0 specifies edge detection for input capture registers ICR0A to ICR0D.

TIOR0 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bits 7 and 6—I/O Control 0D1 and 0D0 (IO0D1, IO0D0): These bits select TI0D pin input capture signal edge detection.

Bit 7: IO0D1	Bit 6: IO0D0	Description
0	0	Input capture disabled (input capture possible in TCNT10B compare-match) (Initial value)
	1	Input capture in ICR0D on rising edge
1	0	Input capture in ICR0D on falling edge
	1	Input capture in ICR0D on both rising and falling edges

- Bits 5 and 4—I/O Control 0C1 and 0C0 (IO0C1, IO0C0): These bits select TI0C pin input capture signal edge detection.

Bit 5: IO0C1	Bit 4: IO0C0	Description
0	0	Input capture disabled (Initial value)
	1	Input capture in ICR0C on rising edge
1	0	Input capture in ICR0C on falling edge
	1	Input capture in ICR0C on both rising and falling edges

- Bits 3 and 2—I/O Control 0B1 and 0B0 (IO0B1, IO0B0): These bits select TI0B pin input capture signal edge detection.

Bit 3: IO0B1	Bit 2: IO0B0	Description
0	0	Input capture disabled (Initial value)
	1	Input capture in ICR0B on rising edge
1	0	Input capture in ICR0B on falling edge
	1	Input capture in ICR0B on both rising and falling edges

- Bits 1 and 0—I/O Control 0A1 and 0A0 (IO0A1, IO0A0): These bits select TI0A pin input capture signal edge detection.

Bit 1: IO0A1	Bit 0: IO0A0	Description
0	0	Input capture disabled (Initial value)
	1	Input capture in ICR0A on rising edge
1	0	Input capture in ICR0A on falling edge
	1	Input capture in ICR0A on both rising and falling edges

## Timer I/O Control Registers 1A to 1D (TIOR1A to TIOR1D)

### TIOR1A

Bit:	7	6	5	4	3	2	1	0
	—	IO1B2	IO1B1	IO1B0	—	IO1A2	IO1A1	IO1A0
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

### TIOR1B

Bit:	7	6	5	4	3	2	1	0
	—	IO1D2	IO1D1	IO1D0	—	IO1C2	IO1C1	IO1C0
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

### TIOR1C

Bit:	7	6	5	4	3	2	1	0
	—	IO1F2	IO1F1	IO1F0	—	IO1E2	IO1E1	IO1E0
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

### TIOR1D

Bit:	7	6	5	4	3	2	1	0
	—	IO1H2	IO1H1	IO1H0	—	IO1G2	IO1G1	IO1G0
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

Registers TIOR1A to TIOR1D specify whether general registers GR1A to GR1H are used as input capture or compare-match registers, and also perform edge detection and output value setting.

Each TIOR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.

- Bits 6 to 4—I/O Control 1B2 to 1B0, 1D2 to 1D0, 1F2 to 1F0, 1H2 to 1H0 (IO1B2 to IO1B0, IO1D2 to IO1D0, IOF12 to IO1F0, IO1H2 to IO1H0): These bits select the general register (GR) function.

Bit 6: IO1x2	Bit 5: IO1x1	Bit 4: IO1x0	Description
0	0	0	GR is an output compare register Compare-match disabled; pin output undefined (Initial value)
		1	0 output on GR compare-match
	1	0	1 output on GR compare-match
		1	Toggle output on GR compare-match
1	0	0	GR is an input capture register Input capture disabled (GR cannot be written to)
		1	Input capture in GR on rising edge at TIO1x pin (GR cannot be written to)
	1	0	Input capture in GR on falling edge at TIO1x pin (GR cannot be written to)
		1	Input capture in GR on both rising and falling edges at TIO1x pin (GR cannot be written to)

x = B, D, F, or H

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.



- Bits 2 to 0—I/O Control 1A2 to 1A0, 1C2 to 1C0, 1E2 to 1E0, 1G2 to 1G0 (IO1A2 to IO1A0, IO1C2 to IO1C0, IO1E2 to IO1E0, IO1G2 to IO1G0): These bits select the general register (GR) function.

Bit 2: IO1x2	Bit 1: IO1x1	Bit 0: IO1x0	Description
0	0	0	GR is an output compare register
		1	Compare-match disabled; pin output undefined (Initial value)
		1	0 output on GR compare-match
	1	0	1 output on GR compare-match
		1	Toggle output on GR compare-match
1	0	0	GR is an input capture register
		1	Input capture disabled
		1	Input capture in GR on rising edge at TIO1x pin (GR cannot be written to)
	1	0	Input capture in GR on falling edge at TIO1x pin (GR cannot be written to)
		1	Input capture in GR on both rising and falling edges at TIO1x pin (GR cannot be written to)

x = A, C, E, or G

## Timer I/O Control Registers 2A to 2D (TIOR2A to TIOR2D)

### TIOR2A

Bit:	7	6	5	4	3	2	1	0
	—	IO2B2	IO2B1	IO2B0	—	IO2A2	IO2A1	IO2A0
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

### TIOR2B

Bit:	7	6	5	4	3	2	1	0
	—	IO2D2	IO2D1	IO2D0	—	IO2C2	IO2C1	IO2C0
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

### TIOR2C

Bit:	7	6	5	4	3	2	1	0
	—	IO2F2	IO2F1	IO2F0	—	IO2E2	IO2E1	IO2E0
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

### TIOR2D

Bit:	7	6	5	4	3	2	1	0
	—	IO2H2	IO2H1	IO2H0	—	IO2G2	IO2G1	IO2G0
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

Registers TIOR2A to TIOR2D specify whether general registers GR2A to GR2H are used as input capture or compare-match registers, and also perform edge detection and output value setting.

Each TIOR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bits 6 to 4—I/O Control 2B2 to 2B0, 2D2 to 2D0, 2F2 to 2F0, 2H2 to 2H0 (IO2B2 to IO2B0, IO2D2 to IO2D0, IO2F2 to IO2F0, IO2H2 to IO2H0): These bits select the general register (GR) function.

Bit 6: IO2x2	Bit 5: IO2x1	Bit 4: IO2x0	Description
0	0	0	GR is an output compare register
		1	Compare-match disabled; pin output undefined (Initial value)
	1	0	0 output on GR compare-match
		1	1 output on GR compare-match
1	0	0	GR is an input capture register
		1	Input capture disabled
	1	0	Input capture in GR on rising edge at TIO2x pin (GR cannot be written to)
		1	Input capture in GR on falling edge at TIO2x pin (GR cannot be written to)
		1	Input capture in GR on both rising and falling edges at TIO2x pin (GR cannot be written to)

x = B, D, F, or H

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.

- Bits 2 to 0—I/O Control 2A2 to 2A0, 2C2 to 2C0, 2E2 to 2E0, 2G2 to 2G0 (IO2A2 to IO2A0, IO2C2 to IO2C0, IO2E2 to IO2E0, IO2G2 to IO2G0): These bits select the general register (GR) function.

Bit 2: IO2x2	Bit 1: IO2x1	Bit 0: IO2x0	Description
0	0	0	GR is an output compare register
		1	
	1	0	
		1	
1	0	0	GR is an input capture register
		1	
	1	0	
		1	

x = A, C, E, or G

## Timer I/O Control Registers 3A, 3B, 4A, 4B, 5A, 5B (TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B)

### TIOR3A, TIOR4A, TIOR5A

Bit:	7	6	5	4	3	2	1	0
	CCIxB	IOxB2	IOxB1	IOxB0	CCIxA	IOxA2	IOxA1	IOxA0
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

x = 3 to 5

### TIOR3B, TIOR4B, TIOR5B

Bit:	7	6	5	4	3	2	1	0
	CCIxD	IOxD2	IOxD1	IOxD0	CCIxC	IOxC2	IOxC1	IOxC0
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

x = 3 to 5

TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, and TIOR5B specify whether general registers GR3A to GR3D, GR4A to GR4D, and GR5A to GR5D are used as input capture or compare-match registers, and also perform edge detection and output value setting. They also select enabling or disabling of free-running counter (TCNT3 to TCNT5) clearing on compare-match.

Each TIOR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Clear Counter Enable Flag 3B, 4B, 5B, 3D, 4D, 5D (CCI3B, CCI4B, CCI5B, CCI3D, CCI4D, CCI5D): These bits select enabling or disabling of free-running counter (TCNT) clearing.

Bit 7: CCIxx	Description
0	TCNT clearing disabled (Initial value)
1	TCNT cleared on GR compare-match

xx = 3B, 4B, 5B, 3D, 4D, or 5D

TCNT is cleared on compare-match only when GR is functioning as an output compare register.

- Bits 6 to 4—I/O Control 3B2 to 3B0, 4B2 to 4B0, 5B2 to 5B0, 3D2 to 3D0, 4D2 to 4D0, 5D2 to 5D0 (IO3B2 to IO3B0, IO4B2 to IO4B0, IO5B2 to IO5B0, IO3D2 to IO3D0, IO4D2 to IO4D0, IO5D2 to IO5D0): These bits select the general register (GR) function.

Bit 6: IOxx2	Bit 5: IOxx1	Bit 4: IOxx0	Description	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register	Input capture disabled (In channel 3 only, GR cannot be written to)
		1	(input capture by channel 3 and 9 compare-match enabled)	Input capture in GR on rising edge at TIOxx pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIOxx pin (GR cannot be written to)
		1		Input capture in GR on both rising and falling edges at TIOxx pin (GR cannot be written to)

xx = 3B, 4B, 5B, 3D, 4D, or 5D

- Bit 3—Clear Counter Enable Flag 3A, 4A, 5A, 3C, 4C, 5C (CCI3A, CCI4A, CCI5A, CCI3C, CCI4C, CCI5C): These bits select enabling or disabling of free-running counter (TCNT) clearing.

Bit 3: CCIxx	Description
0	TCNT clearing disabled (Initial value)
1	TCNT cleared on GR compare-match

xx = 3A, 4A, 5A, 3C, 4C, or 5C

TCNT is cleared on compare-match only when GR is functioning as an output compare register.

- Bits 2 to 0—I/O Control 3A2 to 3A0, 4A2 to 4A0, 5A2 to 5A0, 3C2 to 3C0, 4C2 to 4C0, 5C2 to 5C0 (IO3A2 to IO3A0, IO4A2 to IO4A0, IO5A2 to IO5A0, IO3C2 to IO3C0, IO4C2 to IO4C0, IO5C2 to IO5C0): These bits select the general register (GR) function.

Bit 2: IOxx2	Bit 1: IOxx1	Bit 0: IOxx0	Description
0	0	0	GR is an output compare register
		1	
	1	0	
		1	
1	0	0	GR is an input capture register
		1	(input capture by channel 3 and 9 compare-match enabled)
	1	0	
		1	

xx = 3A, 4A, 5A, 3C, 4C, or 5C

## Timer I/O Control Register 11 (TIOR11)

### TIOR11

Bit:	7	6	5	4	3	2	1	0
	—	IO11B2	IO11B1	IO11B0	—	IO11A2	IO11A1	IO11A0
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

TIOR11 specifies whether general registers GR11A and GR11B are used as input capture or compare-match registers, and also performs edge detection and output value setting.

TIOR11 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.

- Bits 6 to 4—I/O Control 11B2 to 11B0 (IO11B2 to IO11B0): These bits select the general register (GR) function.

Bit 6: IO11B2	Bit 5: IO11B1	Bit 4: IO11B0	Description	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register	Input capture disabled
		1		Input capture in GR on rising edge at TIO11B pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIO11B pin (GR cannot be written to)
		1		Input capture in GR on both rising and falling edges at TIO11B pin (GR cannot be written to)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bits 2 to 0—I/O Control 11A2 to 11A0 (IO11A2 to IO11A0): These bits select the general register (GR) function.

Bit 2: IO11A2	Bit 1: IO11A1	Bit 0: IO11A0	Description	
0	0	0	GR is an output compare register	Compare-match disabled; pin output undefined (Initial value)
		1		0 output on GR compare-match
	1	0		1 output on GR compare-match
		1		Toggle output on GR compare-match
1	0	0	GR is an input capture register	Input capture disabled
		1		Input capture in GR on rising edge at TIO11A pin (GR cannot be written to)
	1	0		Input capture in GR on falling edge at TIO11A pin (GR cannot be written to)
		1		Input capture in GR on both rising and falling edges at TIO11A pin (GR cannot be written to)



## 11.2.5 Timer Status Registers (TSR)

The timer status registers (TSR) are 16-bit registers. The ATU-II has 11 TSR registers: one each for channels 0, 6 to 9, and 11, two each for channels 1 and 2, and one for channels 3 to 5. For details of channel 10, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
0	TSR0	Indicates input capture, interval interrupt, and overflow status
1	TSR1A, TSR1B	Indicate input capture, compare-match, and overflow status
2	TSR2A, TSR2B	
3	TSR3	Indicates input capture, compare-match, and overflow status
4		
5		
6	TSR6	Indicate cycle register compare-match status
7	TSR7	
8	TSR8	Indicates down-counter output end (low) status
9	TSR9	Indicates event counter compare-match status
11	TSR11	Indicates input capture, compare-match, and overflow status

The TSR registers are 16-bit readable/writable registers containing flags that indicate free-running counter (TCNT) overflow, channel 0 input capture or interval interrupt generation, channel 3, 4, 5, and 11 general register input capture or compare-match, channel 6 and 7 compare-matches, channel 8 down-counter output end, and channel 9 event counter compare-matches.

Each flag is an interrupt source, and issues an interrupt request to the CPU if the interrupt is enabled by the corresponding bit in the timer interrupt enable register (TIER).

Each TSR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

## Timer Status Register 0 (TSR0)

TSR0 indicates the status of channel 0 interval interrupts, input capture, and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	IIF2B	IIF2A	IIF1	OVF0	ICF0D	ICF0C	ICF0B	ICF0A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: \* Only 0 can be written to clear the flag.

- Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 7—Interval Interrupt Flag 2B (IIF2B): Status flag that indicates the generation of an interval interrupt.

Bit 7: IIF2B	Description
0	[Clearing condition] (Initial value) When IIF2B is read while set to 1, then 0 is written to IIF2B
1	[Setting condition] When interval interrupt selected by ITVRR2B is generated

- Bit 6—Interval Interrupt Flag 2A (IIF2A): Status flag that indicates the generation of an interval interrupt.

Bit 6: IIF2A	Description
0	[Clearing condition] (Initial value) When IIF2A is read while set to 1, then 0 is written to IIF2A
1	[Setting condition] When interval interrupt selected by ITVRR2A is generated

- Bit 5—Interval Interrupt Flag 1 (IIF1): Status flag that indicates the generation of an interval interrupt.

Bit 5: IIF1	Description
0	[Clearing condition] (Initial value) When IIF1 is read while set to 1, then 0 is written to IIF1
1	[Setting condition] When interval interrupt selected by ITVRR1 is generated

- Bit 4—Overflow Flag 0 (OVF0): Status flag that indicates TCNT0 overflow.

Bit 4: OVF0	Description
0	[Clearing condition] (Initial value) When OVF0 is read while set to 1, then 0 is written to OVF0
1	[Setting condition] When the TCNT0 value overflows (from H'FFFFFFFF to H'00000000)

- Bit 3—Input Capture Flag 0D (ICF0D): Status flag that indicates ICR0D input capture.

Bit 3: ICF0D	Description
0	[Clearing condition] (Initial value) When ICF0D is read while set to 1, then 0 is written to ICF0D
1	[Setting condition] When the TCNT0 value is transferred to the input capture register by an input capture signal. Also set by input capture with a channel 10 compare match as the trigger

- Bit 2—Input Capture Flag 0C (ICF0C): Status flag that indicates ICR0C input capture.

Bit 2: ICF0C	Description
0	[Clearing condition] (Initial value) When ICF0C is read while set to 1, then 0 is written to ICF0C
1	[Setting condition] When the TCNT0 value is transferred to the input capture register by an input capture signal

- Bit 1—Input Capture Flag 0B (ICF0B): Status flag that indicates ICR0B input capture.

Bit 1: ICF0B	Description
0	[Clearing condition] (Initial value) When ICF0B is read while set to 1, then 0 is written to ICF0B
1	[Setting condition] When the TCNT0 value is transferred to the input capture register by an input capture signal

- Bit 0—Input Capture Flag 0A (ICF0A): Status flag that indicates ICR0A input capture.

Bit 0: ICF0A	Description
0	[Clearing condition] (Initial value) When ICF0A is read while set to 1, then 0 is written to ICF0A
1	[Setting condition] When the TCNT0 value is transferred to the input capture register by an input capture signal

### Timer Status Registers 1A and 1B (TSR1A, TSR1B)

**TSR1A:** TSR1A indicates the status of channel 1 input capture, compare-match, and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVF1A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	IMF1H	IMF1G	IMF1F	IMF1E	IMF1D	IMF1C	IMF1B	IMF1A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: \* Only 0 can be written, to clear the flag.

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.

- Bit 8—Overflow Flag 1A (OVF1A): Status flag that indicates TCNT1A overflow.

Bit 8: OVF1A	Description
0	[Clearing condition] (Initial value) When OVF1A is read while set to 1, then 0 is written to OVF1A
1	[Setting condition] When the TCNT1A value overflows (from H'FFFF to H'0000)

- Bit 7—Input Capture/Compare-Match Flag 1H (IMF1H): Status flag that indicates GR1H input capture or compare-match.

Bit 7: IMF1H	Description
0	[Clearing condition] (Initial value) When IMF1H is read while set to 1, then 0 is written to IMF1H
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT1A value is transferred to GR1H by an input capture signal while GR1H is functioning as an input capture register</li> <li>• When TCNT1A = GR1H while GR1H is functioning as an output compare register</li> </ul>

- Bit 6—Input Capture/Compare-Match Flag 1G (IMF1G): Status flag that indicates GR1G input capture or compare-match.

Bit 6: IMF1G	Description
0	[Clearing condition] (Initial value) When IMF1G is read while set to 1, then 0 is written to IMF1G
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT1A value is transferred to GR1G by an input capture signal while GR1G is functioning as an input capture register</li> <li>• When TCNT1A = GR1G while GR1G is functioning as an output compare register</li> </ul>

- Bit 5—Input Capture/Compare-Match Flag 1F (IMF1F): Status flag that indicates GR1F input capture or compare-match.

Bit 5: IMF1F	Description
0	[Clearing condition] (Initial value) When IMF1F is read while set to 1, then 0 is written to IMF1F
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT1A value is transferred to GR1F by an input capture signal while GR1F is functioning as an input capture register</li> <li>• When TCNT1A = GR1F while GR1F is functioning as an output compare register</li> </ul>

- Bit 4—Input Capture/Compare-Match Flag 1E (IMF1E): Status flag that indicates GR1E input capture or compare-match.

Bit 4: IMF1E	Description
0	[Clearing condition] (Initial value) When IMF1E is read while set to 1, then 0 is written to IMF1E
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT1A value is transferred to GR1E by an input capture signal while GR1E is functioning as an input capture register</li> <li>• When TCNT1A = GR1E while GR1E is functioning as an output compare register</li> </ul>

- Bit 3—Input Capture/Compare-Match Flag 1D (IMF1D): Status flag that indicates GR1D input capture or compare-match.

Bit 3: IMF1D	Description
0	[Clearing condition] (Initial value) When IMF1D is read while set to 1, then 0 is written to IMF1D
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT1A value is transferred to GR1D by an input capture signal while GR1D is functioning as an input capture register</li> <li>• When TCNT1A = GR1D while GR1D is functioning as an output compare register</li> </ul>

- Bit 2—Input Capture/Compare-Match Flag 1C (IMF1C): Status flag that indicates GR1C input capture or compare-match.

Bit 2: IMF1C	Description
0	[Clearing condition] (Initial value) When IMF1C is read while set to 1, then 0 is written to IMF1C
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT1A value is transferred to GR1C by an input capture signal while GR1C is functioning as an input capture register</li> <li>• When TCNT1A = GR1C while GR1C is functioning as an output compare register</li> </ul>

- Bit 1—Input Capture/Compare-Match Flag 1B (IMF1B): Status flag that indicates GR1B input capture or compare-match.

Bit 1: IMF1B	Description
0	[Clearing condition] (Initial value) When IMF1B is read while set to 1, then 0 is written to IMF1B
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT1A value is transferred to GR1B by an input capture signal while GR1B is functioning as an input capture register</li> <li>• When TCNT1A = GR1B while GR1B is functioning as an output compare register</li> </ul>

- Bit 0—Input Capture/Compare-Match Flag 1A (IMF1A): Status flag that indicates GR1A input capture or compare-match.

Bit 0: IMF1A	Description
0	[Clearing condition] (Initial value) When IMF1A is read while set to 1, then 0 is written to IMF1A
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT1A value is transferred to GR1A by an input capture signal while GR1A is functioning as an input capture register</li> <li>• When TCNT1A = GR1A while GR1A is functioning as an output compare register</li> </ul>

**TSR1B:** TSR1B indicates the status of channel 1 compare-match and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVF1B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*
Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CMF1
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.

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Flag 1B (OVF1B): Status flag that indicates TCNT1B overflow.

**Bit 8: OVF1B**      **Description**

0	[Clearing condition] When OVF1B is read while set to 1, then 0 is written to OVF1B	(Initial value)
1	[Setting condition] When the TCNT1B value overflows (from H'FFFF to H'0000)	

- Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 0—Compare-Match Flag 1 (CMF1): Status flag that indicates OCR1 compare-match.

**Bit 0: CMF1**      **Description**

0	[Clearing condition] When CMF1 is read while set to 1, then 0 is written to CMF1	(Initial value)
1	[Setting condition] When TCNT1B = OCR1	



## Timer Status Registers 2A and 2B (TSR2A, TSR2B)

**TSR2A:** TSR2A indicates the status of channel 2 input capture, compare-match, and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVF2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	IMF2H	IMF2G	IMF2F	IMF2E	IMF2D	IMF2C	IMF2B	IMF2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: \* Only 0 can be written to clear the flag.

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Flag 2A (OVF2A): Status flag that indicates TCNT2A overflow.

Bit 8: OVF2A	Description
0	[Clearing condition] (Initial value) When OVF2A is read while set to 1, then 0 is written to OVF2A
1	[Setting condition] When the TCNT2A value overflows (from H'FFFF to H'0000)

- Bit 7—Input Capture/Compare-Match Flag 2H (IMF2H): Status flag that indicates GR2H input capture or compare-match.

Bit 7: IMF2H	Description
0	[Clearing condition] (Initial value) When IMF2H is read while set to 1, then 0 is written to IMF2H
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT2A value is transferred to GR2H by an input capture signal while GR2H is functioning as an input capture register</li> <li>• When TCNT2A = GR2H while GR2H is functioning as an output compare register</li> </ul>

- Bit 6—Input Capture/Compare-Match Flag 2G (IMF2G): Status flag that indicates GR2G input capture or compare-match.

Bit 6: IMF2G	Description
0	[Clearing condition] (Initial value) When IMF2G is read while set to 1, then 0 is written to IMF2G
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT2A value is transferred to GR2G by an input capture signal while GR2G is functioning as an input capture register</li> <li>• When TCNT2A = GR2G while GR2G is functioning as an output compare register</li> </ul>

- Bit 5—Input Capture/Compare-Match Flag 2F (IMF2F): Status flag that indicates GR2F input capture or compare-match.

Bit 5: IMF2F	Description
0	[Clearing condition] (Initial value) When IMF2F is read while set to 1, then 0 is written to IMF2F
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT2A value is transferred to GR2F by an input capture signal while GR2F is functioning as an input capture register</li> <li>• When TCNT2A = GR2F while GR2F is functioning as an output compare register</li> </ul>

- Bit 4—Input Capture/Compare-Match Flag 2E (IMF2E): Status flag that indicates GR2E input capture or compare-match.

Bit 4: IMF2E	Description
0	[Clearing condition] (Initial value) When IMF2E is read while set to 1, then 0 is written to IMF2E
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT2A value is transferred to GR2E by an input capture signal while GR2E is functioning as an input capture register</li> <li>• When TCNT2A = GR2E while GR2E is functioning as an output compare register</li> </ul>

- Bit 3—Input Capture/Compare-Match Flag 2D (IMF2D): Status flag that indicates GR2D input capture or compare-match.

Bit 3: IMF2D	Description
0	[Clearing condition] (Initial value) When IMF2D is read while set to 1, then 0 is written to IMF2D
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT2A value is transferred to GR2D by an input capture signal while GR2D is functioning as an input capture register</li> <li>• When TCNT2A = GR2D while GR2D is functioning as an output compare register</li> </ul>

- Bit 2—Input Capture/Compare-Match Flag 2C (IMF2C): Status flag that indicates GR2C input capture or compare-match.

Bit 2: IMF2C	Description
0	[Clearing condition] (Initial value) When IMF2C is read while set to 1, then 0 is written to IMF2C
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT2A value is transferred to GR2C by an input capture signal while GR2C is functioning as an input capture register</li> <li>• When TCNT2A = GR2C while GR2C is functioning as an output compare register</li> </ul>

- Bit 1—Input Capture/Compare-Match Flag 2B (IMF2B): Status flag that indicates GR2B input capture or compare-match.

Bit 1: IMF2B	Description
0	[Clearing condition] (Initial value) When IMF2B is read while set to 1, then 0 is written to IMF2B
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT2A value is transferred to GR2B by an input capture signal while GR2B is functioning as an input capture register</li> <li>• When TCNT2A = GR2B while GR2B is functioning as an output compare register</li> </ul>

- Bit 0—Input Capture/Compare-Match Flag 2A (IMF2A): Status flag that indicates GR2A input capture or compare-match.

Bit 0: IMF2A	Description
0	[Clearing condition] (Initial value) When IMF2A is read while set to 1, then 0 is written to IMF2A
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT2A value is transferred to GR2A by an input capture signal while GR2A is functioning as an input capture register</li> <li>• When TCNT2A = GR2A while GR2A is functioning as an output compare register</li> </ul>

**TSR2B:** TSR2B indicates the status of channel 2 compare-match and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVF2B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	CMF2H	CMF2G	CMF2F	CMF2E	CMF2D	CMF2C	CMF2B	CMF2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: \* Only 0 can be written to clear the flag.

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Flag 2B (OVF2B): Status flag that indicates TCNT2B overflow.

Bit 8: OVF2B	Description
0	[Clearing condition] (Initial value) When OVF2B is read while set to 1, then 0 is written to OVF2B
1	[Setting condition] When the TCNT2B value overflows (from H'FFFF to H'0000)

- Bit 7—Compare-Match Flag 2H (CMF2H): Status flag that indicates OCR2H compare-match.

Bit 7: CMF2H	Description
0	[Clearing condition] (Initial value) When CMF2H is read while set to 1, then 0 is written to CMF2H
1	[Setting condition] When TCNT2B = OCR2H

- Bit 6—Compare-Match Flag 2G (CMF2G): Status flag that indicates OCR2G compare-match.

Bit 6: CMF2G	Description
0	[Clearing condition] (Initial value) When CMF2G is read while set to 1, then 0 is written to CMF2G
1	[Setting condition] When TCNT2B = OCR2G

- Bit 5—Compare-Match Flag 2F (CMF2F): Status flag that indicates OCR2F compare-match.

Bit 5: CMF2F	Description
0	[Clearing condition] (Initial value) When CMF2F is read while set to 1, then 0 is written to CMF2F
1	[Setting condition] When TCNT2B = OCR2F

- Bit 4—Compare-Match Flag 2E (CMF2E): Status flag that indicates OCR2E compare-match.

Bit 4: CMF2E	Description
0	[Clearing condition] (Initial value) When CMF2E is read while set to 1, then 0 is written to CMF2E
1	[Setting condition] When TCNT2B = OCR2E

- Bit 3—Compare-Match Flag 2D (CMF2D): Status flag that indicates OCR2D compare-match.

Bit 3: CMF2D	Description
0	[Clearing condition] (Initial value) When CMF2D is read while set to 1, then 0 is written to CMF2D
1	[Setting condition] When TCNT2B = OCR2D

- Bit 2—Compare-Match Flag 2C (CMF2C): Status flag that indicates OCR2C compare-match.

Bit 2: CMF2C	Description
0	[Clearing condition] (Initial value) When CMF2C is read while set to 1, then 0 is written to CMF2C
1	[Setting condition] When TCNT2B = OCR2C

- Bit 1—Compare-Match Flag 2B (CMF2B): Status flag that indicates OCR2B compare-match.

Bit 1: CMF2B	Description
0	[Clearing condition] (Initial value) When CMF2B is read while set to 1, then 0 is written to CMF2B
1	[Setting condition] When TCNT2B = OCR2B

- Bit 0—Compare-Match Flag 2A (CMF2A): Status flag that indicates OCR2A compare-match.

Bit 0: CMF2A	Description
0	[Clearing condition] (Initial value) When CMF2A is read while set to 1, then 0 is written to CMF2A
1	[Setting condition] When TCNT2B = OCR2A

### Timer Status Register 3 (TSR3)

TSR3 indicates the status of channel 3 to 5 input capture, compare-match, and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	OVF5	IMF5D	IMF5C	IMF5B	IMF5A	OVF4	IMF4D
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	IMF4C	IMF4B	IMF4A	OVF3	IMF3D	IMF3C	IMF3B	IMF3A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: \* Only 0 can be written to clear the flag.

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—Overflow Flag 5 (OVF5): Status flag that indicates TCNT5 overflow.

Bit 14: OVF5	Description
0	[Clearing condition] (Initial value) When OVF5 is read while set to 1, then 0 is written to OVF5
1	[Setting condition] When the TCNT5 value overflows (from H'FFFF to H'0000)

- Bit 13—Input Capture/Compare-Match Flag 5D (IMF5D): Status flag that indicates GR5D input capture or compare-match.

Bit 13: IMF5D	Description
0	[Clearing condition] (Initial value) When IMF5D is read while set to 1, then 0 is written to IMF5D
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT5 value is transferred to GR5D by an input capture signal while GR5D is functioning as an input capture register</li> <li>• When TCNT5 = GR5D while GR5D is functioning as an output compare register</li> <li>• When TCNT5 = GR5D while GR5D is functioning as a cycle register in PWM mode</li> </ul>

- Bit 12—Input Capture/Compare-Match Flag 5C (IMF5C): Status flag that indicates GR5C input capture or compare-match. The flag is not set in PWM mode.

Bit 12: IMF5C	Description
0	[Clearing condition] (Initial value) When IMF5C is read while set to 1, then 0 is written to IMF5C
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT5 value is transferred to GR5C by an input capture signal while GR5C is functioning as an input capture register</li> <li>• When TCNT5 = GR5C while GR5C is functioning as an output compare register</li> </ul>

- Bit 11—Input Capture/Compare-Match Flag 5B (IMF5B): Status flag that indicates GR5B input capture or compare-match. The flag is not set in PWM mode.

Bit 11: IMF5B	Description
0	[Clearing condition] (Initial value) When IMF5B is read while set to 1, then 0 is written to IMF5B
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT5 value is transferred to GR5B by an input capture signal while GR5B is functioning as an input capture register</li> <li>• When TCNT5 = GR5B while GR5B is functioning as an output compare register</li> </ul>

- Bit 10—Input Capture/Compare-Match Flag 5A (IMF5A): Status flag that indicates GR5A input capture or compare-match. The flag is not set in PWM mode.

Bit 10: IMF5A	Description
0	[Clearing condition] (Initial value) When IMF5A is read while set to 1, then 0 is written to IMF5A
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT5 value is transferred to GR5A by an input capture signal while GR5A is functioning as an input capture register</li> <li>• When TCNT5 = GR5A while GR5A is functioning as an output compare register</li> </ul>

- Bit 9—Overflow Flag 4 (OVF4): Status flag that indicates TCNT4 overflow.

Bit 9: OVF4	Description
0	[Clearing condition] (Initial value) When OVF4 is read while set to 1, then 0 is written to OVF4
1	[Setting condition] When the TCNT4 value overflows (from H'FFFF to H'0000)



- Bit 8—Input Capture/Compare-Match Flag 4D (IMF4D): Status flag that indicates GR4D input capture or compare-match.

Bit 8: IMF4D	Description
0	[Clearing condition] (Initial value) When IMF4D is read while set to 1, then 0 is written to IMF4D
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT4 value is transferred to GR4D by an input capture signal while GR4D is functioning as an input capture register</li> <li>• When TCNT4 = GR4D while GR4D is functioning as an output compare register</li> <li>• When TCNT4 = GR4D while GR4D is functioning as a PWM mode synchronous register</li> </ul>

- Bit 7—Input Capture/Compare-Match Flag 4C (IMF4C): Status flag that indicates GR4C input capture or compare-match. The flag is not set in PWM mode.

Bit 7: IMF4C	Description
0	[Clearing condition] (Initial value) When IMF4C is read while set to 1, then 0 is written to IMF4C
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT4 value is transferred to GR4C by an input capture signal while GR4C is functioning as an input capture register</li> <li>• When TCNT4 = GR4C while GR4C is functioning as an output compare register</li> </ul>

- Bit 6—Input Capture/Compare-Match Flag 4B (IMF4B): Status flag that indicates GR4B input capture or compare-match. The flag is not set in PWM mode.

Bit 6: IMF4B	Description
0	[Clearing condition] (Initial value) When IMF4B is read while set to 1, then 0 is written to IMF4B
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT4 value is transferred to GR4B by an input capture signal while GR4B is functioning as an input capture register</li> <li>• When TCNT4 = GR4B while GR4B is functioning as an output compare register</li> </ul>

- Bit 5—Input Capture/Compare-Match Flag 4A (IMF4A): Status flag that indicates GR4A input capture or compare-match. The flag is not set in PWM mode.

Bit 5: IMF4A	Description
0	[Clearing condition] (Initial value) When IMF4A is read while set to 1, then 0 is written to IMF4A
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT4 value is transferred to GR4A by an input capture signal while GR4A is functioning as an input capture register</li> <li>• When TCNT4 = GR4A while GR4A is functioning as an output compare register</li> </ul>

- Bit 4—Overflow Flag 3 (OVF3): Status flag that indicates TCNT3 input capture or compare-match.

Bit 4: OVF3	Description
0	[Clearing condition] (Initial value) When OVF3 is read while set to 1, then 0 is written to OVF3
1	[Setting condition] When the TCNT3 value overflows (from H'FFFF to H'0000)

- Bit 3—Input Capture/Compare-Match Flag 3D (IMF3D): Status flag that indicates GR5D input capture or compare-match.

Bit 3: IMF3D	Description
0	[Clearing condition] (Initial value) When IMF3D is read while set to 1, then 0 is written to IMF3D
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT3 value is transferred to GR3D by an input capture signal while GR3D is functioning as an input capture register. However, IMF3D is not set by input capture with a channel 9 compare match as the trigger</li> <li>• When TCNT3 = GR3D while GR3D is functioning as an output compare register</li> <li>• When TCNT3 = GR3D while GR3D is functioning as a synchronous register in PWM mode</li> </ul>

- Bit 2—Input Capture/Compare-Match Flag 3C (IMF3C): Status flag that indicates GR3C input capture or compare-match. The flag is not set in PWM mode.

Bit 2: IMF3C	Description
0	[Clearing condition] (Initial value) When IMF3C is read while set to 1, then 0 is written to IMF3C
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT3 value is transferred to GR3C by an input capture signal while GR3C is functioning as an input capture register. However, IMF3C is not set by input capture with a channel 9 compare match as the trigger</li> <li>• When TCNT3 = GR3C while GR3C is functioning as an output compare register</li> </ul>

- Bit 1—Input Capture/Compare-Match Flag 3B (IMF3B): Status flag that indicates GR3B input capture or compare-match. The flag is not set in PWM mode.

Bit 1: IMF3B	Description
0	[Clearing condition] (Initial value) When IMF3B is read while set to 1, then 0 is written to IMF3B
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT3 value is transferred to GR3B by an input capture signal while GR3B is functioning as an input capture register. However, IMF3B is not set by input capture with a channel 9 compare match as the trigger</li> <li>• When TCNT3 = GR3B while GR3B is functioning as an output compare register</li> </ul>

- Bit 0—Input Capture/Compare-Match Flag 3A (IMF3A): Status flag that indicates GR3A input capture or compare-match. The flag is not set in PWM mode.

Bit 0: IMF3A	Description
0	[Clearing condition] (Initial value) When IMF3A is read while set to 1, then 0 is written to IMF3A
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT3 value is transferred to GR3A by an input capture signal while GR3A is functioning as an input capture register. However, IMF3A is not set by input capture with a channel 9 compare match as the trigger</li> <li>• When TCNT3 = GR3A while GR3A is functioning as an output compare register</li> </ul>

## Timer Status Registers 6 and 7 (TSR6, TSR7)

TSR6 and TRS7 indicate the channel 6 and 7 free-running counter up-count and down-count status, and cycle register compare status.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	UDxD	UDxC	UDxB	UDxA	CMFxD	CMFxC	CMFxB	CMFxA
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: \* Only 0 can be written to clear the flag.  
x = 6 or 7

UDxA to UDxD relate to TSR6 only. Bits relating to TSR7 always read 0.

- Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 7—Count-Up/Count-Down Flag 6D (UD6D): Status flag that indicates the TCNT6D count operation.

Bit 7: UD6D	Description
0	Free-running counter TCNT6D operates as an up-counter
1	Free-running counter TCNT6D operates as a down-counter

- Bit 6—Count-Up/Count-Down Flag 6C (UD6C): Status flag that indicates the TCNT6C count operation.

Bit 6: UD6C	Description
0	Free-running counter TCNT6C operates as an up-counter
1	Free-running counter TCNT6C operates as a down-counter

- Bit 5—Count-Up/Count-Down Flag 6B (UD6B): Status flag that indicates the TCNT6B count operation.

Bit 5: UD6B	Description
0	Free-running counter TCNT6B operates as an up-counter
1	Free-running counter TCNT6B operates as a down-counter

- Bit 4—Count-Up/Count-Down Flag 6A (UD6A): Status flag that indicates the TCNT6A count operation.

Bit 4: UD6A	Description
0	Free-running counter TCNT6A operates as an up-counter
1	Free-running counter TCNT6A operates as a down-counter

- Bit 3—Cycle Register Compare-Match Flag 6D/7D (CMF6D/CMF7D): Status flag that indicates CYLRxD compare-match.

Bit 3: CMFxD	Description
0	[Clearing condition] (Initial value) When CMFxD is read while set to 1, then 0 is written to CMFxD
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When TCNTxD = CYLRxD (in non-complementary PWM mode)</li> <li>• When TCNT6D = H'0000 in a down-count (in complementary PWM mode)</li> </ul>

x = 6 or 7

- Bit 2—Cycle Register Compare-Match Flag 6C/7C (CMF6C/CMF7C): Status flag that indicates CYLRxC compare-match.

Bit 2: CMFxC	Description
0	[Clearing condition] (Initial value) When CMFxC is read while set to 1, then 0 is written to CMFxC
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When TCNTxC = CYLRxC (in non-complementary PWM mode)</li> <li>• When TCNT6C = H'0000 in a down-count (in complementary PWM mode)</li> </ul>

x = 6 or 7

- Bit 1—Cycle Register Compare-Match Flag 6B/7B (CMF6B/CMF7B): Status flag that indicates CYLRxB compare-match.

Bit 1: CMFxB	Description
0	[Clearing condition] (Initial value) When CMFxB is read while set to 1, then 0 is written to CMFxB
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When TCNTxB = CYLRxB (in non-complementary PWM mode)</li> <li>• When TCNT6B = H'0000 in a down-count (in complementary PWM mode)</li> </ul>

x = 6 or 7

- Bit 0—Cycle Register Compare-Match Flag 6A/7A (CMF6A/CMF7A): Status flag that indicates CYLRxA compare-match.

Bit 0: CMFxA	Description
0	[Clearing condition] (Initial value) When CMFxA is read while set to 1, then 0 is written to CMFxA
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When TCNTxA = CYLRxA (in non-complementary PWM mode)</li> <li>• When TCNT6A = H'0000 in a down-count (in complementary PWM mode)</li> </ul>

x = 6 or 7

### Timer Status Register 8 (TSR8)

TSR8 indicates the channel 8 one-shot pulse status.

Bit:	15	14	13	12	11	10	9	8
	OSF8P	OSF8O	OSF8N	OSF8M	OSF8L	OSF8K	OSF8J	OSF8I
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:	7	6	5	4	3	2	1	0
	OSF8H	OSF8G	OSF8F	OSF8E	OSF8D	OSF8C	OSF8B	OSF8A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: \* Only 0 can be written to clear the flag.

- Bit 15—One-Shot Pulse Flag 8P (OSF8P): Status flag that indicates a DCNT8P one-shot pulse.

Bit 15: OSF8P	Description
0	[Clearing condition] (Initial value) When OSF8P is read while set to 1, then 0 is written to OSF8P
1	[Setting condition] When DCNT8P underflows

- Bit 14—One-Shot Pulse Flag 8O (OSF8O): Status flag that indicates a DCNT8O one-shot pulse.

Bit 14: OSF8O	Description
0	[Clearing condition] (Initial value) When OSF8O is read while set to 1, then 0 is written to OSF8O
1	[Setting condition] When DCNT8O underflows

- Bit 13—One-Shot Pulse Flag 8N (OSF8N): Status flag that indicates a DCNT8N one-shot pulse.

Bit 13: OSF8N	Description
0	[Clearing condition] (Initial value) When OSF8N is read while set to 1, then 0 is written to OSF8N
1	[Setting condition] When DCNT8N underflows

- Bit 12—One-Shot Pulse Flag 8M (OSF8M): Status flag that indicates a DCNT8M one-shot pulse.

Bit 12: OSF8M	Description
0	[Clearing condition] (Initial value) When OSF8M is read while set to 1, then 0 is written to OSF8M
1	[Setting condition] When DCNT8M underflows

- Bit 11—One-Shot Pulse Flag 8L (OSF8L): Status flag that indicates a DCNT8L one-shot pulse.

Bit 11: OSF8L	Description
0	[Clearing condition] (Initial value) When OSF8L is read while set to 1, then 0 is written to OSF8L
1	[Setting condition] When DCNT8L underflows

- Bit 10—One-Shot Pulse Flag 8K (OSF8K): Status flag that indicates a DCNT8K one-shot pulse.

Bit 10: OSF8K	Description
0	[Clearing condition] (Initial value) When OSF8K is read while set to 1, then 0 is written to OSF8K
1	[Setting condition] When DCNT8K underflows

- Bit 9—One-Shot Pulse Flag 8J (OSF8J): Status flag that indicates a DCNT8J one-shot pulse.

Bit 9: OSF8J	Description
0	[Clearing condition] (Initial value) When OSF8J is read while set to 1, then 0 is written to OSF8J
1	[Setting condition] When DCNT8J underflows

- Bit 8—One-Shot Pulse Flag 8I (OSF8I): Status flag that indicates a DCNT8I one-shot pulse.

Bit 8: OSF8I	Description
0	[Clearing condition] (Initial value) When OSF8I is read while set to 1, then 0 is written to OSF8I
1	[Setting condition] When DCNT8I underflows



- Bit 7—One-Shot Pulse Flag 8H (OSF8H): Status flag that indicates a DCNT8H one-shot pulse.

Bit 7: OSF8H	Description
0	[Clearing condition] (Initial value) When OSF8H is read while set to 1, then 0 is written to OSF8H
1	[Setting condition] When DCNT8H underflows

- Bit 6—One-Shot Pulse Flag 8G (OSF8G): Status flag that indicates a DCNT8G one-shot pulse.

Bit 6: OSF8G	Description
0	[Clearing condition] (Initial value) When OSF8G is read while set to 1, then 0 is written to OSF8G
1	[Setting condition] When DCNT8G underflows

- Bit 5—One-Shot Pulse Flag 8F (OSF8F): Status flag that indicates a DCNT8F one-shot pulse.

Bit 5: OSF8F	Description
0	[Clearing condition] (Initial value) When OSF8F is read while set to 1, then 0 is written to OSF8F
1	[Setting condition] When DCNT8F underflows

- Bit 4—One-Shot Pulse Flag 8E (OSF8E): Status flag that indicates a DCNT8E one-shot pulse.

Bit 4: OSF8E	Description
0	[Clearing condition] (Initial value) When OSF8E is read while set to 1, then 0 is written to OSF8E
1	[Setting condition] When DCNT8E underflows

- Bit 3—One-Shot Pulse Flag 8D (OSF8D): Status flag that indicates a DCNT8D one-shot pulse.

Bit 3: OSF8D	Description
0	[Clearing condition] (Initial value) When OSF8D is read while set to 1, then 0 is written to OSF8D
1	[Setting condition] When DCNT8D underflows

- Bit 2—One-Shot Pulse Flag 8C (OSF8C): Status flag that indicates a DCNT8C one-shot pulse.

Bit 2: OSF8C	Description
0	[Clearing condition] (Initial value) When OSF8C is read while set to 1, then 0 is written to OSF8C
1	[Setting condition] When DCNT8C underflows

- Bit 1—One-Shot Pulse Flag 8B (OSF8B): Status flag that indicates a DCNT8B one-shot pulse.

Bit 1: OSF8B	Description
0	[Clearing condition] (Initial value) When OSF8B is read while set to 1, then 0 is written to OSF8B
1	[Setting condition] When DCNT8B underflows

- Bit 0—One-Shot Pulse Flag 8A (OSF8A): Status flag that indicates a DCNT8A one-shot pulse.

Bit 0: OSF8A	Description
0	[Clearing condition] (Initial value) When OSF8A is read while set to 1, then 0 is written to OSF8A
1	[Setting condition] When DCNT8A underflows

## Timer Status Register 9 (TSR9)

TSR9 indicates the channel 9 event counter compare-match status.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	CMF9F	CMF9E	CMF9D	CMF9C	CMF9B	CMF9A
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)*

Note: \* Only 0 can be written to clear the flag.

- Bits 15 to 6—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 5—Compare-Match Flag 9F (CMF9F): Status flag that indicates GR9F compare-match.

Bit 5: CMF9F	Description
0	[Clearing condition] (Initial value) When CMF9F is read while set to 1, then 0 is written to CMF9F
1	[Setting condition] When the next edge is input while ECNT9F = GR9F

- Bit 4—Compare-Match Flag 9E (CMF9E): Status flag that indicates GR9E compare-match.

Bit 4: CMF9E	Description
0	[Clearing condition] (Initial value) When CMF9E is read while set to 1, then 0 is written to CMF9E
1	[Setting condition] When the next edge is input while ECNT9E = GR9E

- Bit 3—Compare-Match Flag 9D (CMF9D): Status flag that indicates GR9D compare-match.

Bit 3: CMF9D	Description
0	[Clearing condition] (Initial value) When CMF9D is read while set to 1, then 0 is written to CMF9D
1	[Setting condition] When the next edge is input while ECNT9D = GR9D

- Bit 2—Compare-Match Flag 9C (CMF9C): Status flag that indicates GR9C compare-match.

Bit 2: CMF9C	Description
0	[Clearing condition] (Initial value) When CMF9C is read while set to 1, then 0 is written to CMF9C
1	[Setting condition] When the next edge is input while ECNT9C = GR9C

- Bit 1—Compare-Match Flag 9B (CMF9B): Status flag that indicates GR9B compare-match.

Bit 1: CMF9B	Description
0	[Clearing condition] (Initial value) When CMF9B is read while set to 1, then 0 is written to CMF9B
1	[Setting condition] When the next edge is input while ECNT9B = GR9B

- Bit 0—Compare-Match Flag 9A (CMF9A): Status flag that indicates GR9A compare-match.

Bit 0: CMF9A	Description
0	[Clearing condition] (Initial value) When CMF9A is read while set to 1, then 0 is written to CMF9A
1	[Setting condition] When the next edge is input while ECNT9A = GR9A

## Timer Status Register 11 (TSR11)

TSR11 indicates the status of channel 11 input capture, compare-match, and overflow.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVF11
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/(W)*
Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IMF11B	IMF11A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/(W)*	R/(W)*

Note: \* Only 0 can be written to clear the flag.

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Flag 11 (OVF11): Status flag that indicates TCNT11 overflow.

Bit 8: OVF11	Description
0	[Clearing condition] (Initial value) When OVF11 is read while set to 1, then 0 is written to OVF11
1	[Setting condition] When the TCNT11 value overflows (from H'FFFF to H'0000)

- Bits 7 to 2—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 1—Input Capture/Compare-Match Flag 11B (IMF11B): Status flag that indicates GR11B input capture or compare-match.

Bit 1: IMF11B	Description
0	[Clearing condition] (Initial value) When IMF11B is read while set to 1, then 0 is written to IMF11B
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT11 value is transferred to GR11B by an input capture signal while GR11B is functioning as an input capture register</li> <li>• When TCNT11 = GR11B while GR11B is functioning as an output compare register</li> </ul>

- Bit 0—Input Capture/Compare-Match Flag 11A (IMF11A): Status flag that indicates GR11A input capture or compare-match.

Bit 0: IMF11A	Description
0	[Clearing condition] (Initial value) When IMF11A is read while set to 1, then 0 is written to IMF11A
1	[Setting conditions] <ul style="list-style-type: none"> <li>• When the TCNT11 value is transferred to GR11A by an input capture signal while GR11A is functioning as an input capture register</li> <li>• When TCNT11 = GR11A while GR11A is functioning as an output compare register</li> </ul>

### 11.2.6 Timer Interrupt Enable Registers (TIER)

The timer interrupt enable registers (TIER) are 16-bit registers. The ATU-II has 11 TIER registers: one each for channels 0, 6 to 9, and 11, two each for channels 1 and 2, and one for channels 3 to 5. For details of channel 10, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
0	TIER0	Controls input capture, and overflow interrupt request enabling/disabling.
1	TIER1A, TIER1B	Control input capture, compare-match, and overflow interrupt request enabling/disabling.
2	TIER2A, TIER2B	
3	TIER3	Controls input capture, compare-match, and overflow interrupt request enabling/disabling.
4		
5		
6	TIER6	Control cycle register compare-match interrupt request enabling/disabling.
7	TIER7	
8	TIER8	Controls down-counter output end (low) interrupt request enabling/disabling.
9	TIER9	Controls event counter compare-match interrupt request enabling/disabling.
11	TIER11	Controls input capture, compare-match, and overflow interrupt request enabling/disabling.

The TIER registers are 16-bit readable/writable registers that control enabling/disabling of free-running counter (TCNT) overflow interrupt requests, channel 0 input capture interrupt requests, channel 1 to 5 and 11 general register input capture/compare-match interrupt requests, channel 6 and 7 compare-match interrupt requests, channel 8 down-counter output end interrupt requests, and channel 9 event counter compare-match interrupt requests.

Each TIER is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

### Timer Interrupt Enable Register 0 (TIER0)

TIER0 controls enabling/disabling of channel 0 input capture and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	—	—	—	OVE0	ICE0D	ICE0C	ICE0B	ICE0A
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

- Bits 15 to 5—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 4—Overflow Interrupt Enable 0 (OVE0): Enables or disables interrupt requests by the overflow flag (OVF0) in TSR0 when OVF0 is set to 1.

Bit 4: OVE0	Description
0	OVI0 interrupt requested by OVF0 is disabled (Initial value)
1	OVI0 interrupt requested by OVF0 is enabled

- Bit 3—Input Capture Interrupt Enable 0D (ICE0D): Enables or disables interrupt requests by the input capture flag (ICF0D) in TSR0 when ICF0D is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 3: ICE0D	Description
0	ICI0D interrupt requested by ICF0D is disabled (Initial value)
1	ICI0D interrupt requested by ICF0D is enabled

- Bit 2—Input Capture Interrupt Enable 0C (ICE0C): Enables or disables interrupt requests by the input capture flag (ICF0C) in TSR0 when ICF0C is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 2: ICE0C	Description
0	ICI0C interrupt requested by ICF0C is disabled (Initial value)
1	ICI0C interrupt requested by ICF0C is enabled

- Bit 1—Input Capture Interrupt Enable 0B (ICE0B): Enables or disables interrupt requests by the input capture flag (ICF0B) in TSR0 when ICF0B is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 1: ICE0B	Description
0	ICI0B interrupt requested by ICF0B is disabled (Initial value)
1	ICI0B interrupt requested by ICF0B is enabled

- Bit 0—Input Capture Interrupt Enable 0A (ICE0A): Enables or disables interrupt requests by the input capture flag (ICF0A) in TSR0 when ICF0A is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 0: ICE0A	Description
0	ICI0A interrupt requested by ICF0A is disabled (Initial value)
1	ICI0A interrupt requested by ICF0A is enabled

### Timer Interrupt Enable Registers 1A and 1B (TIER1A, TIER1B)

**TIER1A:** TIER1A controls enabling/disabling of channel 1 input capture, compare-match, and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
Bit name:	—	—	—	—	—	—	—	OVE1A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
Bit name:	IME1H	IME1G	IME1F	IME1E	IME1D	IME1C	IME1B	IME1A
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



- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Interrupt Enable 1A (OVE1A): Enables or disables interrupt requests by OVF1A in TSR1A when OVF1A is set to 1.

Bit 8: OVE1A	Description
0	OV1A interrupt requested by OVF1A is disabled (Initial value)
1	OV1A interrupt requested by OVF1A is enabled

- Bit 7—Input Capture/Compare-Match Interrupt Enable 1H (IME1H): Enables or disables interrupt requests by IMF1H in TSR1A when IMF1H is set to 1.

Bit 7: IME1H	Description
0	IMI1H interrupt requested by IMF1H is disabled (Initial value)
1	IMI1H interrupt requested by IMF1H is enabled

- Bit 6—Input Capture/Compare-Match Interrupt Enable 1G (IME1G): Enables or disables interrupt requests by IMF1G in TSR1A when IMF1G is set to 1.

Bit 6: IME1G	Description
0	IMI1G interrupt requested by IMF1G is disabled (Initial value)
1	IMI1G interrupt requested by IMF1G is enabled

- Bit 5—Input Capture/Compare-Match Interrupt Enable 1F (IME1F): Enables or disables interrupt requests by IMF1F in TSR1A when IMF1F is set to 1.

Bit 5: IME1F	Description
0	IMI1F interrupt requested by IMF1F is disabled (Initial value)
1	IMI1F interrupt requested by IMF1F is enabled

- Bit 4—Input Capture/Compare-Match Interrupt Enable 1E (IME1E): Enables or disables interrupt requests by IMF1E in TSR1A when IMF1E is set to 1.

Bit 4: IME1E	Description
0	IMI1E interrupt requested by IMF1E is disabled (Initial value)
1	IMI1E interrupt requested by IMF1E is enabled

- Bit 3—Input Capture/Compare-Match Interrupt Enable 1D (IME1D): Enables or disables interrupt requests by IMF1D in TSR1A when IMF1D is set to 1.

Bit 3: IME1D	Description
0	IMI1D interrupt requested by IMF1D is disabled (Initial value)
1	IMI1D interrupt requested by IMF1D is enabled

- Bit 2—Input Capture/Compare-Match Interrupt Enable 1C (IME1C): Enables or disables interrupt requests by IMF1C in TSR1A when IMF1C is set to 1.

Bit 2: IME1C	Description
0	IMI1C interrupt requested by IMF1C is disabled (Initial value)
1	IMI1C interrupt requested by IMF1C is enabled

- Bit 1—Input Capture/Compare-Match Interrupt Enable 1B (IME1B): Enables or disables interrupt requests by IMF1B in TSR1A when IMF1B is set to 1.

Bit 1: IME1B	Description
0	IMI1B interrupt requested by IMF1B is disabled (Initial value)
1	IMI1B interrupt requested by IMF1B is enabled

- Bit 0—Input Capture/Compare-Match Interrupt Enable 1A (IME1A): Enables or disables interrupt requests by IMF1A in TSR1A when IMF1A is set to 1.

Bit 0: IME1A	Description
0	IMI1A interrupt requested by IMF1A is disabled (Initial value)
1	IMI1A interrupt requested by IMF1A is enabled

**TIER1B:** TIER1B controls enabling/disabling of channel 1 compare-match and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVE1B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W
Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CME1
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Interrupt Enable 1B (OVE1B): Enables or disables interrupt requests by OVF1B in TSR1B when OVF1B is set to 1.

Bit 8: OVE1B	Description
0	OV11B interrupt requested by OVF1B is disabled (Initial value)
1	OV11B interrupt requested by OVF1B is enabled

- Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 0—Compare-Match Interrupt Enable 1 (CME1): Enables or disables interrupt requests by CMF1 in TSR1B when CMF1 is set to 1.

Bit 0: CME1	Description
0	CMI1 interrupt requested by CMF1 is disabled (Initial value)
1	CMI1 interrupt requested by CMF1 is enabled

## Timer Interrupt Enable Registers 2A and 2B (TIER2A, TIER2B)

**TIER2A:** TIER2A controls enabling/disabling of channel 2 input capture, compare-match, and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVE2A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	IME2H	IME2G	IME2F	IME2E	IME2D	IME2C	IME2B	IME2A
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

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Interrupt Enable 2A (OVE2A): Enables or disables interrupt requests by OVF2A in TSR2A when OVF2A is set to 1.

Bit 8: OVE2A	Description
0	OVI2A interrupt requested by OVF2A is disabled (Initial value)
1	OVI2A interrupt requested by OVF2A is enabled

- Bit 7—Input Capture/Compare-Match Interrupt Enable 2H (IME2H): Enables or disables interrupt requests by IMF2H in TSR2A when IMF2H is set to 1.

Bit 7: IME2H	Description
0	IMI2H interrupt requested by IMF2H is disabled (Initial value)
1	IMI2H interrupt requested by IMF2H is enabled

- Bit 6—Input Capture/Compare-Match Interrupt Enable 2G (IME2G): Enables or disables interrupt requests by IMF2G in TSR2A when IMF2G is set to 1.

Bit 6: IME2G	Description
0	IMI2G interrupt requested by IMF2G is disabled (Initial value)
1	IMI2G interrupt requested by IMF2G is enabled

- Bit 5—Input Capture/Compare-Match Interrupt Enable 2F (IME2F): Enables or disables interrupt requests by IMF2F in TSR2A when IMF2F is set to 1.

Bit 5: IME2F	Description
0	IMI2F interrupt requested by IMF2F is disabled (Initial value)
1	IMI2F interrupt requested by IMF2F is enabled

- Bit 4—Input Capture/Compare-Match Interrupt Enable 2E (IME2E): Enables or disables interrupt requests by IMF2E in TSR2A when IMF2E is set to 1.

Bit 4: IME2E	Description
0	IMI2E interrupt requested by IMF2E is disabled (Initial value)
1	IMI2E interrupt requested by IMF2E is enabled

- Bit 3—Input Capture/Compare-Match Interrupt Enable 2D (IME2D): Enables or disables interrupt requests by IMF2D in TSR2A when IMF2D is set to 1.

Bit 3: IME2D	Description
0	IMI2D interrupt requested by IMF2D is disabled (Initial value)
1	IMI2D interrupt requested by IMF2D is enabled

- Bit 2—Input Capture/Compare-Match Interrupt Enable 2C (IME2C): Enables or disables interrupt requests by IMF2C in TSR2A when IMF2C is set to 1.

Bit 2: IME2C	Description
0	IMI2C interrupt requested by IMF2C is disabled (Initial value)
1	IMI2C interrupt requested by IMF2C is enabled

- Bit 1—Input Capture/Compare-Match Interrupt Enable 2B (IME2B): Enables or disables interrupt requests by IMF2B in TSR2A when IMF2B is set to 1.

Bit 1: IME2B	Description
0	IMI2B interrupt requested by IMF2B is disabled (Initial value)
1	IMI2B interrupt requested by IMF2B is enabled

- Bit 0—Input Capture/Compare-Match Interrupt Enable 2A (IME2A): Enables or disables interrupt requests by IMF2A in TSR2A when IMF2A is set to 1.

Bit 0: IME2A	Description
0	IMI2A interrupt requested by IMF2A is disabled (Initial value)
1	IMI2A interrupt requested by IMF2A is enabled

**TIER2B:** TIER2B controls enabling/disabling of channel 2 compare-match and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVE2B
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	CME2H	CME2G	CME2F	CME2E	CME2D	CME2C	CME2B	CME2A
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

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—Overflow Interrupt Enable 2B (OVE2B): Enables or disables interrupt requests by OVF2B in TSR2B when OVF2B is set to 1.

Bit 8: OVE2B	Description
0	OVI2B interrupt requested by OVF2B is disabled (Initial value)
1	OVI2B interrupt requested by OVF2B is enabled

- Bit 7—Compare-Match Interrupt Enable 2H (CME2H): Enables or disables interrupt requests by CMF2F in TSR2B when CMF2H is set to 1.

Bit 7: CME2H	Description
0	CMI2H interrupt requested by CMF2H is disabled (Initial value)
1	CMI2H interrupt requested by CMF2H is enabled

- Bit 6—Compare-Match Interrupt Enable 2G (CME2G): Enables or disables interrupt requests by CMF2G in TSR2B when CMF2G is set to 1.

Bit 6: CME2G	Description
0	CMI2G interrupt requested by CMF2G is disabled (Initial value)
1	CMI2G interrupt requested by CMF2G is enabled

- Bit 5—Compare-Match Interrupt Enable 2F (CME2F): Enables or disables interrupt requests by CMF2F in TSR2B when CMF2F is set to 1.

Bit 5: CME2F	Description
0	CMI2F interrupt requested by CMF2F is disabled (Initial value)
1	CMI2F interrupt requested by CMF2F is enabled

- Bit 4—Compare-Match Interrupt Enable 2E (CME2E): Enables or disables interrupt requests by CMF2E in TSR2B when CMF2E is set to 1.

Bit 4: CME2E	Description
0	CMI2E interrupt requested by CMF2E is disabled (Initial value)
1	CMI2E interrupt requested by CMF2E is enabled

- Bit 3—Compare-Match Interrupt Enable 2D (CME2D): Enables or disables interrupt requests by CMF2D in TSR2B when CMF2D is set to 1.

Bit 3: CME2D	Description
0	CMI2D interrupt requested by CMF2D is disabled (Initial value)
1	CMI2D interrupt requested by CMF2D is enabled

- Bit 2—Compare-Match Interrupt Enable 2C (CME2C): Enables or disables interrupt requests by CMF2C in TSR2B when CMF2C is set to 1.

Bit 2: CME2C	Description
0	CMI2C interrupt requested by CMF2C is disabled (Initial value)
1	CMI2C interrupt requested by CMF2C is enabled

- Bit 1—Compare-Match Interrupt Enable 2B (CME2BB): Enables or disables interrupt requests by CMF2B in TSR2B when CMF2B is set to 1.

Bit 1: CME2B	Description
0	CMI2B interrupt requested by CMF2B is disabled (Initial value)
1	CMI2B interrupt requested by CMF2B is enabled

- Bit 0—Compare-Match Interrupt Enable 2A (CME2A): Enables or disables interrupt requests by CMF2A in TSR2B when CMF2A is set to 1.

Bit 0: CME2A	Description
0	CMI2A interrupt requested by CMF2A is disabled (Initial value)
1	CMI2A interrupt requested by CMF2A is enabled

### Timer Interrupt Enable Register 3 (TIER3)

TIER3 controls enabling/disabling of channel 3 to 5 input capture, compare-match, and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	OVE5	IME5D	IME5C	IME5B	IME5A	OVE4	IME4D
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	IME4C	IME4B	IME4A	OVE3	IME3D	IME3C	IME3B	IME3A
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 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—Overflow Interrupt Enable 5 (OVE5): Enables or disables interrupt requests by OVF5 in TSR3 when OVF5 is set to 1.

Bit 14: OVE5	Description
0	OVI5 interrupt requested by OVF5 is disabled (Initial value)
1	OVI5 interrupt requested by OVF5 is enabled



- Bit 13—Input Capture/Compare-Match Interrupt Enable 5D (IME5D): Enables or disables interrupt requests by IMF5D in TSR3 when IMF5D is set to 1.

Bit 13: IME5D	Description
0	IMI5D interrupt requested by IMF5D is disabled (Initial value)
1	IMI5D interrupt requested by IMF5D is enabled

- Bit 12—Input Capture/Compare-Match Interrupt Enable 5C (IME5C): Enables or disables interrupt requests by IMF5C in TSR3 when IMF5C is set to 1.

Bit 12: IME5C	Description
0	IMI5C interrupt requested by IMF5C is disabled (Initial value)
1	IMI5C interrupt requested by IMF5C is enabled

- Bit 11—Input Capture/Compare-Match Interrupt Enable 5B (IME5B): Enables or disables interrupt requests by IMF5B in TSR3 when IMF5B is set to 1.

Bit 11: IME5B	Description
0	IMI5B interrupt requested by IMF5B is disabled (Initial value)
1	IMI5B interrupt requested by IMF5B is enabled

- Bit 10—Input Capture/Compare-Match Interrupt Enable 5A (IME5A): Enables or disables interrupt requests by IMF5A in TSR3 when IMF5A is set to 1.

Bit 10: IME5A	Description
0	IMI5A interrupt requested by IMF5A is disabled (Initial value)
1	IMI5A interrupt requested by IMF5A is enabled

- Bit 9—Overflow Interrupt Enable 4 (OVE4): Enables or disables interrupt requests by OVF4 in TSR3 when OVF4 is set to 1.

Bit 9: OVE4	Description
0	OVI4 interrupt requested by OVF4 is disabled (Initial value)
1	OVI4 interrupt requested by OVF4 is enabled

- Bit 8—Input Capture/Compare-Match Interrupt Enable 4D (IME4D): Enables or disables interrupt requests by IMF4D in TSR3 when IMF4D is set to 1.

Bit 8: IME4D	Description
0	IMI4D interrupt requested by IMF4D is disabled (Initial value)
1	IMI4D interrupt requested by IMF4D is enabled

- Bit 7—Input Capture/Compare-Match Interrupt Enable 4C (IME4C): Enables or disables interrupt requests by IMF4C in TSR3 when IMF4C is set to 1.

Bit 7: IME4C	Description
0	IMI4C interrupt requested by IMF4C is disabled (Initial value)
1	IMI4C interrupt requested by IMF4C is enabled

- Bit 6—Input Capture/Compare-Match Interrupt Enable 4B (IME4B): Enables or disables interrupt requests by IMF4B in TSR3 when IMF4B is set to 1.

Bit 6: IME4B	Description
0	IMI4B interrupt requested by IMF4B is disabled (Initial value)
1	IMI4B interrupt requested by IMF4B is enabled

- Bit 5—Input Capture/Compare-Match Interrupt Enable 4A (IME4A): Enables or disables interrupt requests by IMF4A in TSR3 when IMF4A is set to 1.

Bit 5: IME4A	Description
0	IMI4A interrupt requested by IMF4A is disabled (Initial value)
1	IMI4A interrupt requested by IMF4A is enabled

- Bit 4—Overflow Interrupt Enable 3 (OVE3): Enables or disables interrupt requests by OVF3 in TSR3 when OVF3 is set to 1.

Bit 4: OVE3	Description
0	OVI3 interrupt requested by OVF3 is disabled (Initial value)
1	OVI3 interrupt requested by OVF3 is enabled

- Bit 3—Input Capture/Compare-Match Interrupt Enable 3D (IME3D): Enables or disables interrupt requests by IMF3D in TSR3 when IMF3D is set to 1.

Bit 3: IME3D	Description
0	IMI3D interrupt requested by IMF3D is disabled (Initial value)
1	IMI3D interrupt requested by IMF3D is enabled

- Bit 2—Input Capture/Compare-Match Interrupt Enable 3C (IME3C): Enables or disables interrupt requests by IMF3C in TSR3 when IMF3C is set to 1.

Bit 2: IME3C	Description
0	IMI3C interrupt requested by IMF3C is disabled (Initial value)
1	IMI3C interrupt requested by IMF3C is enabled

- Bit 1—Input Capture/Compare-Match Interrupt Enable 3B (IME3B): Enables or disables interrupt requests by IMF3B in TSR3 when IMF3B is set to 1.

Bit 1: IME3B	Description
0	IMI3B interrupt requested by IMF3B is disabled (Initial value)
1	IMI3B interrupt requested by IMF3B is enabled

- Bit 0—Input Capture/Compare-Match Interrupt Enable 3A (IME3A): Enables or disables interrupt requests by IMF3A in TSR3 when IMF3A is set to 1.

Bit 0: IME3A	Description
0	IMI3A interrupt requested by IMF3A is disabled (Initial value)
1	IMI3A interrupt requested by IMF3A is enabled

## Timer Interrupt Enable Registers 6 and 7 (TIER6, TIER7)

TIER6 and TIER7 control enabling/disabling of channel 6 and 7 cycle register compare interrupt requests.

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

x = 6 or 7

- Bits 15 to 4—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 3—Cycle Register Compare-Match Interrupt Enable 6D/7D (CME6D/CME7D): Enables or disables interrupt requests by CMFxD in TSR6 or TSR7 when CMFxD is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 3: CMExD	Description
0	CMIxD interrupt requested by CMFxD is disabled (Initial value)
1	CMIxD interrupt requested by CMFxD is enabled

x = 6 or 7

- Bit 2—Cycle Register Compare-Match Interrupt Enable 6C/7C (CME6C/CME7C): Enables or disables interrupt requests by CMFxC in TSR6 or TSR7 when CMFxC is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 2: CMExC	Description
0	CMIxC interrupt requested by CMFxC is disabled (Initial value)
1	CMIxC interrupt requested by CMFxC is enabled

x = 6 or 7

- Bit 1—Cycle Register Compare-Match Interrupt Enable 6B/7B (CME6B/CME7B): Enables or disables interrupt requests by CMFxB in TSR6 or TSR7 when CMFxB is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 1: CMExB	Description
0	CMIxB interrupt requested by CMFxB is disabled (Initial value)
1	CMIxB interrupt requested by CMFxB is enabled

x = 6 or 7

- Bit 0—Cycle Register Compare-Match Interrupt Enable 6A/7A (CME6A/CME7A): Enables or disables interrupt requests by CMFxA in TSR6 or TSR7 when CMFxA is set to 1. Setting the DMAC while interrupt requests are enabled allows the DMAC to be activated by an interrupt request.

Bit 0: CMExA	Description
0	CMIxA interrupt requested by CMFxA is disabled (Initial value)
1	CMIxA interrupt requested by CMFxA is enabled

x = 6 or 7

### Timer Interrupt Enable Register 8 (TIER8)

TIER8 controls enabling/disabling of channel 8 one-shot pulse interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	OSE8P	OSE8O	OSE8N	OSE8M	OSE8L	OSE8K	OSE8J	OSE8I
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:	7	6	5	4	3	2	1	0
	OSE8H	OSE8G	OSE8F	OSE8E	OSE8D	OSE8C	OSE8B	OSE8A
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 15—One-Shot Pulse Interrupt Enable 8P (OSE8P): Enables or disables interrupt requests by OSF8P in TSR8 when OSF8P is set to 1.

Bit 15: OSE8P	Description
0	OSI8P interrupt requested by OSF8P is disabled (Initial value)
1	OSI8P interrupt requested by OSF8P is enabled

- Bit 14—One-Shot Pulse Interrupt Enable 8O (OSE8O): Enables or disables interrupt requests by OSF8O in TSR8 when OSF8O is set to 1.

Bit 14: OSE8O	Description
0	OSI8O interrupt requested by OSF8O is disabled (Initial value)
1	OSI8O interrupt requested by OSF8O is enabled

- Bit 13—One-Shot Pulse Interrupt Enable 8N (OSE8N): Enables or disables interrupt requests by OSF8N in TSR8 when OSF8N is set to 1.

Bit 13: OSE8N	Description
0	OSI8N interrupt requested by OSF8N is disabled (Initial value)
1	OSI8N interrupt requested by OSF8N is enabled

- Bit 12—One-Shot Pulse Interrupt Enable 8M (OSE8M): Enables or disables interrupt requests by OSF8M in TSR8 when OSF8M is set to 1.

Bit 12: OSE8M	Description
0	OSI8M interrupt requested by OSF8M is disabled (Initial value)
1	OSI8M interrupt requested by OSF8M is enabled

- Bit 11—One-Shot Pulse Interrupt Enable 8L (OSE8L): Enables or disables interrupt requests by OSF8L in TSR8 when OSF8L is set to 1.

Bit 11: OSE8L	Description
0	OSI8L interrupt requested by OSF8L is disabled (Initial value)
1	OSI8L interrupt requested by OSF8L is enabled

- Bit 10—One-Shot Pulse Interrupt Enable 8K (OSE8K): Enables or disables interrupt requests by OSF8K in TSR8 when OSF8K is set to 1.

Bit 10: OSE8K	Description
0	OSI8K interrupt requested by OSF8K is disabled (Initial value)
1	OSI8K interrupt requested by OSF8K is enabled

- Bit 9—One-Shot Pulse Interrupt Enable 8J (OSE8J): Enables or disables interrupt requests by OSF8J in TSR8 when OSF8J is set to 1.

Bit 9: OSE8J	Description
0	OSI8J interrupt requested by OSF8J is disabled (Initial value)
1	OSI8J interrupt requested by OSF8J is enabled

- Bit 8—One-Shot Pulse Interrupt Enable 8I (OSE8I): Enables or disables interrupt requests by OSF8I in TSR8 when OSF8I is set to 1.

Bit 8: OSE8I	Description
0	OSI8I interrupt requested by OSF8I is disabled (Initial value)
1	OSI8I interrupt requested by OSF8I is enabled

- Bit 7—One-Shot Pulse Interrupt Enable 8H (OSE8H): Enables or disables interrupt requests by OSF8H in TSR8 when OSF8H is set to 1.

Bit 7: OSE8H	Description
0	OSI8H interrupt requested by OSF8H is disabled (Initial value)
1	OSI8H interrupt requested by OSF8H is enabled

- Bit 6—One-Shot Pulse Interrupt Enable 8G (OSE8G): Enables or disables interrupt requests by OSF8G in TSR8 when OSF8G is set to 1.

Bit 6: OSE8G	Description
0	OSI8G interrupt requested by OSF8G is disabled (Initial value)
1	OSI8G interrupt requested by OSF8G is enabled

- Bit 5—One-Shot Pulse Interrupt Enable 8F (OSE8F): Enables or disables interrupt requests by OSF8F in TSR8 when OSF8F is set to 1.

Bit 5: OSE8F	Description
0	OSI8F interrupt requested by OSF8F is disabled (Initial value)
1	OSI8F interrupt requested by OSF8F is enabled

- Bit 4—One-Shot Pulse Interrupt Enable 8E (OSE8E): Enables or disables interrupt requests by OSF8E in TSR8 when OSF8E is set to 1.

Bit 4: OSE8E	Description
0	OSI8E interrupt requested by OSF8E is disabled (Initial value)
1	OSI8E interrupt requested by OSF8E is enabled

- Bit 3—One-Shot Pulse Interrupt Enable 8D (OSE8D): Enables or disables interrupt requests by OSF8D in TSR8 when OSF8D is set to 1.

Bit 3: OSE8D	Description
0	OSI8D interrupt requested by OSF8D is disabled (Initial value)
1	OSI8D interrupt requested by OSF8D is enabled

- Bit 2—One-Shot Pulse Interrupt Enable 8C (OSE8C): Enables or disables interrupt requests by OSF8C in TSR8 when OSF8C is set to 1.

Bit 2: OSE8C	Description
0	OSI8C interrupt requested by OSF8C is disabled (Initial value)
1	OSI8C interrupt requested by OSF8C is enabled

- Bit 1—One-Shot Pulse Interrupt Enable 8B (OSE8B): Enables or disables interrupt requests by OSF8B in TSR8 when OSF8B is set to 1.

Bit 1: OSE8B	Description
0	OSI8B interrupt requested by OSF8B is disabled (Initial value)
1	OSI8B interrupt requested by OSF8B is enabled

- Bit 0—One-Shot Pulse Interrupt Enable 8A (OSE8A): Enables or disables interrupt requests by OSF8A in TSR8 when OSF8A is set to 1.

Bit 0: OSE8A	Description
0	OSI8A interrupt requested by OSF8A is disabled (Initial value)
1	OSI8A interrupt requested by OSF8A is enabled



## Timer Interrupt Enable Register 9 (TIER9)

TIER9 controls enabling/disabling of channel 9 event counter compare-match interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	CME9F	CME9E	CME9D	CME9C	CME9B	CME9A
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

- Bits 15 to 6—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 5—Compare-Match Interrupt Enable 9F (CME9F): Enables or disables interrupt requests by CMF9F in TSR9 when CMF9F is set to 1.

Bit 5: CME9F	Description
0	CMI9F interrupt requested by CMF9F is disabled (Initial value)
1	CMI9F interrupt requested by CMF9F is enabled

- Bit 4—Compare-Match Interrupt Enable 9E (CME9E): Enables or disables interrupt requests by CMF9E in TSR9 when CMF9E is set to 1.

Bit 4: CME9E	Description
0	CMI9E interrupt requested by CMF9E is disabled (Initial value)
1	CMI9E interrupt requested by CMF9E is enabled

- Bit 3—Compare-Match Interrupt Enable 9D (CME9D): Enables or disables interrupt requests by CMF9D in TSR9 when CMF9D is set to 1.

Bit 3: CME9D	Description
0	CMI9D interrupt requested by CMF9D is disabled (Initial value)
1	CMI9D interrupt requested by CMF9D is enabled

- Bit 2—Compare-Match Interrupt Enable 9C (CME9C): Enables or disables interrupt requests by CMF9C in TSR9 when CMF9C is set to 1.

Bit 2: CME9C	Description
0	CMI9C interrupt requested by CMF9C is disabled (Initial value)
1	CMI9C interrupt requested by CMF9C is enabled

- Bit 1—Compare-Match Interrupt Enable 9B (CME9B): Enables or disables interrupt requests by CMF9B in TSR9 when CMF9B is set to 1.

Bit 1: CME9B	Description
0	CMI9B interrupt requested by CMF9B is disabled (Initial value)
1	CMI9B interrupt requested by CMF9B is enabled

- Bit 0—Compare-Match Interrupt Enable 9A (CME9A): Enables or disables interrupt requests by CMF9A in TSR9 when CMF9A is set to 1.

Bit 0: CME9A	Description
0	CMI9A interrupt requested by CMF9A is disabled (Initial value)
1	CMI9A interrupt requested by CMF9A is enabled

### Timer Interrupt Enable Register 11 (TIER11)

TIER11 controls enabling/disabling of channel 11 input capture, compare-match, and overflow interrupt requests.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	OVE11
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IME11B	IME11A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.

- Bit 8—Overflow Interrupt Enable 11 (OVE11): Enables or disables interrupt requests by OVF11 in TSR11 when OVF11 is set to 1.

Bit 8: OVE11	Description
0	OVI11 interrupt requested by OVF11 is disabled (Initial value)
1	OVI11 interrupt requested by OVF11 is enabled

- Bits 7 to 2—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 1—Input Capture/Compare-Match Interrupt Enable 11B (IME11B): Enables or disables interrupt requests by IMF11B in TSR11 when IMF11B is set to 1.

Bit 1: IME11B	Description
0	IMI11B interrupt requested by IMF11B is disabled (Initial value)
1	IMI11B interrupt requested by IMF11B is enabled

- Bit 0—Input Capture/Compare-Match Interrupt Enable 11A (IME11A): Enables or disables interrupt requests by IMF11A in TSR11 when IMF11A is set to 1.

Bit 0: IME11A	Description
0	IMI11A interrupt requested by IMF11A is disabled (Initial value)
1	IMI11A interrupt requested by IMF11A is enabled

### 11.2.7 Interval Interrupt Request Registers (ITVRR)

The interval interrupt request registers (ITVRR) are 8-bit registers. The ATU-II has three ITVRR registers in channel 0.

Channel	Abbreviation	Function
0	ITVRR1	TCNT0 bit 6 to 9 interval interrupt generation and A/D2 converter activation
	ITVRR2A	TCNT0 bit 10 to 13 interval interrupt generation and A/D0 converter activation
	ITVRR2B	TCNT0 bit 10 to 13 interval interrupt generation and A/D1 converter activation

## Interval Interrupt Request Register 1 (ITVRR1)

Bit:	7	6	5	4	3	2	1	0
	ITVA9	ITVA8	ITVA7	ITVA6	ITVE9	ITVE8	ITVE7	ITVE6
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

ITVRR1 is an 8-bit readable/writable register that detects the rise of bits corresponding to the channel 0 free-running counter (TCNT0) and controls cyclic interrupt output and A/D2 converter activation.

ITVRR1 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—A/D2 Converter Interval Activation Bit 9 (ITVA9): A/D2 converter activation setting bit corresponding to bit 9 in TCNT0. The rise of bit 9 in TCNT0 is ANDed with ITVA9, and the result is output to the A/D2 converter as an activation signal.

Bit 7: ITVA9	Description
0	A/D2 converter activation by rise of TCNT0 bit 9 is disabled (Initial value)
1	A/D2 converter activation by rise of TCNT0 bit 9 is enabled

- Bit 6—A/D2 Converter Interval Activation Bit 8 (ITVA8): A/D2 converter activation setting bit corresponding to bit 8 in TCNT0. The rise of bit 8 in TCNT0 is ANDed with ITVA8, and the result is output to the A/D2 converter as an activation signal.

Bit 6: ITVA8	Description
0	A/D2 converter activation by rise of TCNT0 bit 8 is disabled (Initial value)
1	A/D2 converter activation by rise of TCNT0 bit 8 is enabled

- Bit 5—A/D2 Converter Interval Activation Bit 7 (ITVA7): A/D2 converter activation setting bit corresponding to bit 7 in TCNT0. The rise of bit 7 in TCNT0 is ANDed with ITVA7, and the result is output to the A/D2 converter as an activation signal.

Bit 5: ITVA7	Description
0	A/D2 converter activation by rise of TCNT0 bit 7 is disabled (Initial value)
1	A/D2 converter activation by rise of TCNT0 bit 7 is enabled

- Bit 4—A/D2 Converter Interval Activation Bit 6 (ITVA6): A/D2 converter activation setting bit corresponding to bit 6 in TCNT0. The rise of bit 6 in TCNT0 is ANDed with ITVA6, and the result is output to the A/D2 converter as an activation signal.

Bit 4: ITVA6	Description
0	A/D2 converter activation by rise of TCNT0 bit 6 is disabled (Initial value)
1	A/D2 converter activation by rise of TCNT0 bit 6 is enabled

- Bit 3—Interval Interrupt Bit 9 (ITVE9): INTC interval interrupt setting bit corresponding to bit 9 in TCNT0. The rise of bit 9 in TCNT0 is ANDed with ITVE9, the result is stored in IIF1 in TSR0, and an interrupt request is sent to the CPU.

Bit 3: ITVE9	Description
0	Interrupt request (ITV1) by rise of TCNT0 bit 9 is disabled (Initial value)
1	Interrupt request (ITV1) by rise of TCNT0 bit 9 is enabled

- Bit 2—Interval Interrupt Bit 8 (ITVE8): INTC interval interrupt setting bit corresponding to bit 8 in TCNT0. The rise of bit 8 in TCNT0 is ANDed with ITVE8, the result is stored in IIF1 in TSR0, and an interrupt request is sent to the CPU.

Bit 2: ITVE8	Description
0	Interrupt request (ITV1) by rise of TCNT0 bit 8 is disabled (Initial value)
1	Interrupt request (ITV1) by rise of TCNT0 bit 8 is enabled

- Bit 1—Interval Interrupt Bit 7 (ITVE7): INTC interval interrupt setting bit corresponding to bit 7 in TCNT0. The rise of bit 7 in TCNT0 is ANDed with ITVE7, the result is stored in IIF1 in TSR0, and an interrupt request is sent to the CPU.

Bit 1: ITVE7	Description
0	Interrupt request (ITV1) by rise of TCNT0 bit 7 is disabled (Initial value)
1	Interrupt request (ITV1) by rise of TCNT0 bit 7 is enabled

- Bit 0—Interval Interrupt Bit 6 (ITVE6): INTC interval interrupt setting bit corresponding to bit 6 in TCNT0. The rise of bit 6 in TCNT0 is ANDed with ITVE6, the result is stored in IIF1 in TSR0, and an interrupt request is sent to the CPU.

Bit 0: ITVE6	Description
0	Interrupt request (ITV1) by rise of TCNT0 bit 6 is disabled (Initial value)
1	Interrupt request (ITV1) by rise of TCNT0 bit 6 is enabled

## Interval Interrupt Request Registers 2A and 2B (ITVRR2A, ITVRR2B)

Bit:	7	6	5	4	3	2	1	0
	ITVA13x	ITVA12x	ITVA11x	ITVA10x	ITVE13x	ITVE12x	ITVE11x	ITVE10x
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

x = A or B

- Bit 7—A/D0 / A/D1 Converter Interval Activation Bit 13A/13B (ITVA13A/ITVA13B): A/D0 or A/D1 (ITVRR2A: A/D0; ITVRR2B: A/D1) converter activation setting bit corresponding to bit 13 in TCNT0. The rise of bit 13 in TCNT0 is ANDed with ITVA13x, and the result is output to the A/D0 or A/D1 converter as an activation signal.

### Bit 7: ITVA13x Description

0	A/D0 or A/D1 converter activation by rise of TCNT0 bit 13 is disabled (Initial value)
1	A/D0 or A/D1 converter activation by rise of TCNT0 bit 13 is enabled

x = A or B

- Bit 6—A/D0 / A/D1 Converter Interval Activation Bit 12A/12B (ITVA12A/ITVA12B): A/D0 or A/D1 (ITVRR2A: A/D0; ITVRR2B: A/D1) converter activation setting bit corresponding to bit 12 in TCNT0. The rise of bit 12 in TCNT0 is ANDed with ITVA12x, and the result is output to the A/D0 or A/D1 converter as an activation signal.

### Bit 6: ITVA12x Description

0	A/D0 or A/D1 converter activation by rise of TCNT0 bit 12 is disabled (Initial value)
1	A/D0 or A/D1 converter activation by rise of TCNT0 bit 12 is enabled

x = A or B

- Bit 5—A/D0 / A/D1 Converter Interval Activation Bit 11A/11B (ITVA11A/ITVA11B): A/D0 or A/D1 (ITVRR2A: A/D0; ITVRR2B: A/D1) converter activation setting bit corresponding to bit 11 in TCNT0. The rise of bit 11 in TCNT0 is ANDed with ITVA11x, and the result is output to the A/D0 or A/D1 converter as an activation signal.

### Bit 5: ITVA11x Description

0	A/D0 or A/D1 converter activation by rise of TCNT0 bit 11 is disabled (Initial value)
1	A/D0 or A/D1 converter activation by rise of TCNT0 bit 11 is enabled

x = A or B

- Bit 4—A/D0 / A/D1 Converter Interval Activation Bit 10A/10B (ITVA10A/ITVA10B): A/D0 or A/D1 (ITVRR2A: A/D0; ITVRR2B: A/D1) converter activation setting bit corresponding to bit 10 in TCNT0. The rise of bit 10 in TCNT0 is ANDed with ITVA10x, and the result is output to the A/D0 or A/D1 converter as an activation signal.

Bit 4: ITVA10x	Description
0	A/D0 or A/D1 converter activation by rise of TCNT0 bit 10 is disabled (Initial value)
1	A/D0 or A/D1 converter activation by rise of TCNT0 bit 10 is enabled

x = A or B

- Bit 3—Interval Interrupt Bit 13A/13B (ITVE13A/ITVE13B): INTC interval interrupt setting bit corresponding to bit 13 in TCNT0. The rise of bit 13 in TCNT0 is ANDed with ITVE13x, the result is stored in IIF2x in TSR0, and an interrupt request is sent to the CPU.

Bit 3: ITVE13x	Description
0	Interrupt request (ITV2x) by rise of TCNT0 bit 13 is disabled (Initial value)
1	Interrupt request (ITV2x) by rise of TCNT0 bit 13 is enabled

x = A or B

- Bit 2—Interval Interrupt Bit 12A/12B (ITVE12A/ITVE12B): INTC interval interrupt setting bit corresponding to bit 12 in TCNT0. The rise of bit 12 in TCNT0 is ANDed with ITVE12x, the result is stored in IIF2x in TSR0, and an interrupt request is sent to the CPU.

Bit 2: ITVE12x	Description
0	Interrupt request (ITV2x) by rise of TCNT0 bit 12 is disabled (Initial value)
1	Interrupt request (ITV2x) by rise of TCNT0 bit 12 is enabled

x = A or B

- Bit 1—Interval Interrupt Bit 11A/11B (ITVE11A/ITVE11B): INTC interval interrupt setting bit corresponding to bit 11 in TCNT0. The rise of bit 11 in TCNT0 is ANDed with ITVE11x, the result is stored in IIF2x in TSR0, and an interrupt request is sent to the CPU.

Bit 1: ITVE11x	Description
0	Interrupt request (ITV2x) by rise of TCNT0 bit 11 is disabled (Initial value)
1	Interrupt request (ITV2x) by rise of TCNT0 bit 11 is enabled

x = A or B

- Bit 0—Interval Interrupt Bit 10 (ITVE10): INTC interval interrupt setting bit corresponding to bit 10 in TCNT0. The rise of bit 10 in TCNT0 is ANDed with ITVE10x, the result is stored in IIF2x in TSR0, and an interrupt request is sent to the CPU.

Bit 0: ITVE10x	Description
0	Interrupt request (ITV2x) by rise of TCNT0 bit 10 is disabled (Initial value)
1	Interrupt request (ITV2x) by rise of TCNT0 bit 10 is enabled

x = A or B

For details, see section 11.3.7, Interval Timer Operation.

### 11.2.8 Trigger Mode Register (TRGMDR)

The trigger mode register (TRGMDR) is an 8-bit register. The ATU-II has one TRGMDR register.

Bit:	7	6	5	4	3	2	1	0
	TRGMD	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

TRGMDR is an 8-bit readable/writable register that selects whether a channel 1 compare-match is used as a channel 8 one-shot pulse start trigger or as a one-shot pulse terminate trigger when channel 1 and channel 8 are used in combination.

TRGMDR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Trigger Mode Selection Register (TRGMD): Selects the channel 8 one-shot pulse start trigger/one-shot pulse terminate trigger setting.

Bit 7: TRGMD	Description
0	One-shot pulse start trigger (TCNT1B = OCR1) (Initial value) One-shot pulse terminate trigger (TCNT1A = GR1A–GR1H)
1	One-shot pulse start trigger (TCNT1A = GR1A–GR1H) One-shot pulse terminate trigger (TCNT1B = OCR1)

- Bits 6 to 0—Reserved: These bits are always read as 0. The write value should always be 0.



## 11.2.9 Timer Mode Register (TMDR)

The timer mode register (TMDR) is an 8-bit register. The ATU-II has one TDR register.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	T5PWM	T4PWM	T3PWM
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W

TMDR is an 8-bit readable/writable register that specifies whether channels 3 to 5 are used in input capture/output compare mode or PWM mode.

TMDR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bits 7 to 3—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 2—PWM Mode 5 (T5PWM): Selects whether channel 5 operates in input capture/output compare mode or PWM mode.

Bit 2: T5PWM	Description
0	Channel 5 operates in input capture/output compare mode (Initial value)
1	Channel 5 operates in PWM mode

When bit T5PWM is set to 1 to select PWM mode, pins TIO5A to TIO5C become PWM output pins, general register 5D (GR5D) functions as a cycle register, and general registers 5A to 5C (GR5A to GR5C) function as duty registers. Settings in the timer I/O control registers (TIOR5A, TIOR5B) are invalid, and general registers 5A to 5D (GR5A to GR5D) can be written to. Do not use the TIO5D pin as a timer output.

- Bit 1—PWM Mode 4 (T4PWM): Selects whether channel 4 operates in input capture/output compare mode or PWM mode.

Bit 1: T4PWM	Description
0	Channel 4 operates in input capture/output compare mode (Initial value)
1	Channel 4 operates in PWM mode

When bit T4PWM is set to 1 to select PWM mode, pins TIO4A to TIO4C become PWM output pins, general register 4D (GR4D) functions as a cycle register, and general registers 4A to 4C (GR4A to GR4C) function as duty registers. Settings in the timer I/O control registers (TIOR4A, TIOR4B) are invalid, and general registers 4A to 4D (GR4A to GR4D) can be written to. Do not use the TIO4D pin as a timer output.

- Bit 0—PWM Mode 3 (T3PWM): Selects whether channel 3 operates in input capture/output compare mode or PWM mode.

Bit 0: T3PWM	Description
0	Channel 3 operates in input capture/output compare mode (Initial value)
1	Channel 3 operates in PWM mode

When bit T3PWM is set to 1 to select PWM mode, pins TIO3A to TIO3C become PWM output pins, general register 3D (GR3D) functions as a cycle register, and general registers 3A to 3C (GR3A to GR3C) function as duty registers. Settings in the timer I/O control registers (TIOR3A, TIOR3B) are invalid, and general registers 3A to 3D (GR3A to GR3D) can be written to. Do not use the TIO3D pin as a timer output.

### 11.2.10 PWM Mode Register (PMDR)

The PWM mode register (PMDR) is an 8-bit register. The ATU-II has one PMDR register.

Bit:	7	6	5	4	3	2	1	0
	DTSELD	DTSELC	DTSELB	DTSELA	CNTSELD	CNTSELC	CNTSELB	CNTSELA
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

PMDR is an 8-bit readable/writable register that selects whether channel 6 PWM output is set to on-duty/off-duty, or to non-complementary PWM mode/complementary PWM mode.

PMDR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 7—Duty Selection Register D (DTSELD): Selects whether channel 6D TO6D output PWM is set to on-duty or to off-duty.

Bit 7: DTSELD	Description
0	TO6D PWM output is on-duty (Initial value)
1	TO6D PWM output is off-duty

- Bit 6—Duty Selection Register C (DTSEL C): Selects whether channel 6C TO6C output PWM is set to on-duty or to off-duty.

Bit 6: DTSEL C	Description
0	TO6C PWM output is on-duty (Initial value)
1	TO6C PWM output is off-duty

- Bit 5—Duty Selection Register B (DTSEL B): Selects whether channel 6B TO6B output PWM is set to on-duty or to off-duty.

Bit 5: DTSEL B	Description
0	TO6B PWM output is on-duty (Initial value)
1	TO6B PWM output is off-duty

- Bit 4—Duty Selection Register A (DTSEL A): Selects whether channel 6A TO6A output PWM is set to on-duty or to off-duty.

Bit 4: DTSEL A	Description
0	TO6A PWM output is on-duty (Initial value)
1	TO6A PWM output is off-duty

- Bit 3—Counter Selection Register D (CNTSEL D): Selects whether channel 6D PWM is set to non-complementary PWM mode or to complementary PWM mode.

Bit 3: CNTSEL D	Description
0	TCNT6D is set to non-complementary PWM mode (Initial value)
1	TCNT6D is set to complementary PWM mode

- Bit 2—Counter Selection Register C (CNTSEL C): Selects whether channel 6C PWM is set to non-complementary PWM mode or to complementary PWM mode.

Bit 2: CNTSEL C	Description
0	TCNT6C is set to non-complementary PWM mode (Initial value)
1	TCNT6C is set to complementary PWM mode

- Bit 1—Counter Selection Register B (CNTSELB): Selects whether channel 6B PWM is set to non-complementary PWM mode or to complementary PWM mode.

Bit 1: CNTSELB	Description
0	TCNT6B is set to non-complementary PWM mode (Initial value)
1	TCNT6B is set to complementary PWM mode

- Bit 0—Counter Selection Register A (CNTSELA): Selects whether channel 6A PWM is set to non-complementary PWM mode or to complementary PWM mode.

Bit 0: CNTSELA	Description
0	TCNT6A is set to non-complementary PWM mode (Initial value)
1	TCNT6A is set to complementary PWM mode

### 11.2.11 Down-Count Start Register (DSTR)

The down-count start register (DSTR) is a 16-bit register. The ATU-II has one DSTR register in channel 8.

Bit:	15	14	13	12	11	10	9	8
	DST8P	DST8O	DST8N	DST8M	DST8L	DST8K	DST8J	DST8I
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:	7	6	5	4	3	2	1	0
	DST8H	DST8G	DST8F	DST8E	DST8D	DST8C	DST8B	DST8A
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Note: \* Only 1 can be written.

DSTR is a 16-bit readable/writable register that starts the channel 8 down-counter (DCNT).

When the one-shot pulse function is used, a value of 1 can be set in a DST8x bit at any time by the user program, except when the corresponding DCNT8x value is H'0000. The DST8x bits are cleared to 0 automatically when the DCNT value overflows.

When the offset one-shot pulse function is used, DST8x is automatically set to 1 (except when the DCNT8x value is H'0000) when a compare-match occurs between the channel 1 or 2 free-running counter (TCNT) and a general register (GR) or the output compare register (OCR1) while the corresponding timer connection register (TCNR) bit is set to 1. As regards DST8I to DST8P, if the

RLDEN bit in the reload enable register (RLDENR) is set to 1 and the reload register (RLDR8) value is not H'0000, a reload is performed into the corresponding DCNT8x, and the DST8x bit is set to 1. DST8x is automatically cleared to 0 when the DCNT8x value underflows, or by input of a channel 1 or 2 one-shot terminate trigger signal set in the trigger mode register (TRGMDR) while the corresponding one-shot pulse terminate register (OTR) bit is set to 1, whichever occurs first.

DCNT8x is cleared to H'0000 when underflow occurs.

DSTR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 11.3.5, One-Shot Pulse Function, and 11.3.6, Offset One-Shot Pulse Function and Output Cutoff Function.

- Bit 15—Down-Count Start 8P (DST8P): Starts down-counter 8P (DCNT8P).

Bit 15: DST8P	Description
0	DCNT8P is halted (Initial value) [Clearing condition] When the DCNT8P value underflows, or on channel 2 (GR2H) compare-match
1	DCNT8P counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8P ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR2H compare-match (DCNT8P ≠ H'0000 or reload possible) or by user program (DCNT8P ≠ H'0000)</li> </ul>

- Bit 14—Down-Count Start 8O (DST8O): Starts down-counter 8O (DCNT8O).

Bit 14: DST8O	Description
0	DCNT8O is halted (Initial value) [Clearing condition] When the DCNT8O value underflows, or on channel 2 (GR2G) compare-match
1	DCNT8O counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8O ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR2G compare-match (DCNT8O ≠ H'0000 or reload possible) or by user program (DCNT8O ≠ H'0000)</li> </ul>

- Bit 13—Down-Count Start 8N (DST8N): Starts down-counter 8N (DCNT8N).

Bit 13: DST8N	Description
0	DCNT8N is halted (Initial value) [Clearing condition] When the DCNT8N value underflows, or on channel 2 (GR2F) compare-match
1	DCNT8N counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8N ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR2F compare-match (DCNT8N ≠ H'0000 or reload possible) or by user program (DCNT8N ≠ H'0000)</li> </ul>

- Bit 12—Down-Count Start 8M (DST8M): Starts down-counter 8M (DCNT8M).

Bit 12: DST8M	Description
0	DCNT8M is halted (Initial value) [Clearing condition] When the DCNT8M value underflows, or on channel 2 (GR2E) compare-match
1	DCNT8M counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8M ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR2E compare-match (DCNT8M ≠ H'0000 or reload possible) or by user program (DCNT8M ≠ H'0000)</li> </ul>

- Bit 11—Down-Count Start 8L (DST8L): Starts down-counter 8L (DCNT8L).

Bit 11: DST8L	Description
0	DCNT8L is halted (Initial value) [Clearing condition] When the DCNT8L value underflows, or on channel 2 (GR2D) compare-match
1	DCNT8L counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8L ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR2D compare-match (DCNT8L ≠ H'0000 or reload possible) or by user program (DCNT8L ≠ H'0000)</li> </ul>

- Bit 10—Down-Count Start 8K (DST8K): Starts down-counter 8K (DCNT8K).

Bit 10: DST8K	Description
0	DCNT8K is halted (Initial value) [Clearing condition] When the DCNT8K value underflows, or on channel 2 (GR2C) compare-match
1	DCNT8K counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8K ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR2C compare-match (DCNT8K ≠ H'0000 or reload possible) or by user program (DCNT8K ≠ H'0000)</li> </ul>

- Bit 9—Down-Count Start 8J (DST8J): Starts down-counter 8J (DCNT8J).

Bit 9: DST8J	Description
0	DCNT8J is halted (Initial value) [Clearing condition] When the DCNT8J value underflows, or on channel 2 (GR2B) compare-match
1	DCNT8J counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8J ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR2B compare-match (DCNT8J ≠ H'0000 or reload possible) or by user program (DCNT8J ≠ H'0000)</li> </ul>

- Bit 8—Down-Count Start 8I (DST8I): Starts down-counter 8I (DCNT8I).

Bit 8: DST8I	Description
0	DCNT8I is halted (Initial value) [Clearing condition] When the DCNT8I value underflows, or on channel 2 (GR2A) compare-match
1	DCNT8I counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8I ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR2A compare-match (DCNT8I ≠ H'0000 or reload possible) or by user program (DCNT8I ≠ H'0000)</li> </ul>

- Bit 7—Down-Count Start 8H (DST8H): Starts down-counter 8H (DCNT8H).

Bit 7: DST8H	Description
0	DCNT8H is halted (Initial value) [Clearing condition] When the DCNT8H value underflows, or on channel 1 (GR1H or OCR1) compare-match
1	DCNT8H counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8H ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR1 compare-match or GR1H compare-match, or by user program (DCNT8H ≠ H'0000)</li> </ul>

- Bit 6—Down-Count Start 8G (DST8G): Starts down-counter 8G (DCNT8G).

Bit 6: DST8G	Description
0	DCNT8G is halted (Initial value) [Clearing condition] When the DCNT8G value underflows, or on channel 1 (GR1G or OCR1) compare-match
1	DCNT8G counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8G ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR1 compare-match or GR1G compare-match, or by user program (DCNT8G ≠ H'0000)</li> </ul>



- Bit 5—Down-Count Start 8F (DST8F): Starts down-counter 8F (DCNT8F).

Bit 5: DST8F	Description
0	DCNT8F is halted (Initial value) [Clearing condition] When the DCNT8F value underflows, or on channel 1 (GR1F or OCR1) compare-match
1	DCNT8F counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8F ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR1 compare-match or GR1F compare-match, or by user program (DCNT8F ≠ H'0000)</li> </ul>

- Bit 4—Down-Count Start 8E (DST8E): Starts down-counter 8E (DCNT8E).

Bit 4: DST8E	Description
0	DCNT8E is halted (Initial value) [Clearing condition] When the DCNT8E value underflows, or on channel 1 (GR1E or OCR1) compare-match
1	DCNT8E counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8E ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR1 compare-match or GR1E compare-match, or by user program (DCNT8E ≠ H'0000)</li> </ul>

- Bit 3—Down-Count Start 8D (DST8D): Starts down-counter 8D (DCNT8D).

Bit 3: DST8D	Description
0	DCNT8D is halted (Initial value) [Clearing condition] When the DCNT8D value underflows, or on channel 1 (GR1D or OCR1) compare-match
1	DCNT8D counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8D ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR1 compare-match or GR1D compare-match, or by user program (DCNT8D ≠ H'0000)</li> </ul>

- Bit 2—Down-Count Start 8C (DST8C): Starts down-counter 8C (DCNT8C).

Bit 2: DST8C	Description
0	DCNT8C is halted (Initial value) [Clearing condition] When the DCNT8C value underflows, or on channel 1 (GR1C or OCR1) compare-match
1	DCNT8C counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8C ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR1 compare-match or GR1C compare-match, or by user program (DCNT8C ≠ H'0000)</li> </ul>

- Bit 1—Down-Count Start 8B (DST8B): Starts down-counter 8B (DCNT8B).

Bit 1: DST8B	Description
0	DCNT8B is halted (Initial value) [Clearing condition] When the DCNT8B value underflows, or on channel 1 (GR1B or OCR1) compare-match
1	DCNT8B counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8B ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR1 compare-match or GR1B compare-match, or by user program (DCNT8B ≠ H'0000)</li> </ul>

- Bit 0—Down-Count Start 8A (DST8A): Starts down-counter 8A (DCNT8A).

Bit 0: DST8A	Description
0	DCNT8A is halted (Initial value) [Clearing condition] When the DCNT8A value underflows, or on channel 1 (GR1A or OCR1) compare-match
1	DCNT8A counts [Setting conditions] <ul style="list-style-type: none"> <li>• One-shot pulse function: Set by user program (DCNT8A ≠ H'0000)</li> <li>• Offset one-shot pulse function: Set on OCR1 compare-match or GR1A compare-match, or by user program (DCNT8A ≠ H'0000)</li> </ul>

## 11.2.12 Timer Connection Register (TCNR)

The timer connection register (TCNR) is a 16-bit register. The ATU-II has one TCNR register in channel 8.

Bit:	15	14	13	12	11	10	9	8
	CN8P	CN8O	CN8N	CN8M	CN8L	CN8K	CN8J	CN8I
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:	7	6	5	4	3	2	1	0
	CN8H	CN8G	CN8F	CN8E	CN8D	CN8C	CN8B	CN8A
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

TCNR is a 16-bit readable/writable register that enables or disables connection between the channel 8 down-count start register (DSTR) and channel 1 and 2 compare-match signals (down-count start triggers). Channel 1 down-count start triggers A to H are channel 1 OCR1 compare-match signals or GR1x compare-match signals (set in TRGMDR). Channel 2 down-count start triggers A to H are channel 2 OCR2x compare-match signals. When GR1x compare-matches are used, set TIOR1A to TIOR1D to allow compare-matches.

TCNR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 11.3.5, One-Shot Pulse Function, and 11.3.6, Offset One-Shot Pulse Function and Output Cutoff Function.

- Bit 15—Connection Flag 8P (CN8P): Enables or disables connection between DST8P and the channel 2 down-count start trigger.

Bit 15: CN8P	Description
0	Connection between DST8P and channel 2 down-count start trigger H is disabled (Initial value)
1	Connection between DST8P and channel 2 down-count start trigger H is enabled

- Bit 14—Connection Flag 8O (CN8O): Enables or disables connection between DST8O and the channel 2 down-count start trigger.

Bit 14: CN8O	Description
0	Connection between DST8O and channel 2 down-count start trigger G is disabled (Initial value)
1	Connection between DST8O and channel 2 down-count start trigger G is enabled

- Bit 13—Connection Flag 8N (CN8N): Enables or disables connection between DST8N and the channel 2 down-count start trigger.

Bit 13: CN8N	Description
0	Connection between DST8N and channel 2 down-count start trigger F is disabled (Initial value)
1	Connection between DST8N and channel 2 down-count start trigger F is enabled

- Bit 12—Connection Flag 8M (CN8M): Enables or disables connection between DST8M and the channel 2 down-count start trigger.

Bit 12: CN8M	Description
0	Connection between DST8M and channel 2 down-count start trigger E is disabled (Initial value)
1	Connection between DST8M and channel 2 down-count start trigger E is enabled

- Bit 11—Connection Flag 8L (CN8L): Enables or disables connection between DST8L and the channel 2 down-count start trigger.

Bit 11: CN8L	Description
0	Connection between DST8L and channel 2 down-count start trigger D is disabled (Initial value)
1	Connection between DST8L and channel 2 down-count start trigger D is enabled

- Bit 10—Connection Flag 8K (CN8K): Enables or disables connection between DST8K and the channel 2 down-count start trigger.

Bit 10: CN8K	Description
0	Connection between DST8K and channel 2 down-count start trigger C is disabled (Initial value)
1	Connection between DST8K and channel 2 down-count start trigger C is enabled

- Bit 9—Connection Flag 8J (CN8J): Enables or disables connection between DST8J and the channel 2 down-count start trigger.

Bit 9: CN8J	Description
0	Connection between DST8J and channel 2 down-count start trigger B is disabled (Initial value)
1	Connection between DST8J and channel 2 down-count start trigger B is enabled

- Bit 8—Connection Flag 8I (CN8I): Enables or disables connection between DST8I and the channel 2 down-count start trigger.

Bit 8: CN8I	Description
0	Connection between DST8I and channel 2 down-count start trigger A is disabled (Initial value)
1	Connection between DST8I and channel 2 down-count start trigger A is enabled

- Bit 7—Connection Flag 8H (CN8H): Enables or disables connection between DST8H and the channel 1 down-count start trigger.

Bit 7: CN8H	Description
0	Connection between DST8H and channel 1 down-count start trigger H is disabled (Initial value)
1	Connection between DST8H and channel 1 down-count start trigger H is enabled

- Bit 6—Connection Flag 8G (CN8G): Enables or disables connection between DST8G and the channel 1 down-count start trigger.

Bit 6: CN8G	Description
0	Connection between DST8G and channel 1 down-count start trigger G is disabled (Initial value)
1	Connection between DST8G and channel 1 down-count start trigger G is enabled

- Bit 5—Connection Flag 8F (CN8F): Enables or disables connection between DST8F and the channel 1 down-count start trigger.

Bit 5: CN8F	Description
0	Connection between DST8F and channel 1 down-count start trigger F is disabled (Initial value)
1	Connection between DST8F and channel 1 down-count start trigger F is enabled

- Bit 4—Connection Flag 8E (CN8E): Enables or disables connection between DST8E and the channel 1 down-count start trigger.

Bit 4: CN8E	Description
0	Connection between DST8E and channel 1 down-count start trigger E is disabled (Initial value)
1	Connection between DST8E and channel 1 down-count start trigger E is enabled

- Bit 3—Connection Flag 8D (CN8D): Enables or disables connection between DST8D and the channel 1 down-count start trigger.

Bit 3: CN8D	Description
0	Connection between DST8D and channel 1 down-count start trigger D is disabled (Initial value)
1	Connection between DST8D and channel 1 down-count start trigger D is enabled

- Bit 2—Connection Flag 8C (CN8C): Enables or disables connection between DST8C and the channel 1 down-count start trigger.

Bit 2: CN8C	Description
0	Connection between DST8C and channel 1 down-count start trigger C is disabled (Initial value)
1	Connection between DST8C and channel 1 down-count start trigger C is enabled

- Bit 1—Connection Flag 8B (CN8B): Enables or disables connection between DST8B and the channel 1 down-count start trigger.

Bit 1: CN8B	Description
0	Connection between DST8B and channel 1 down-count start trigger B is disabled (Initial value)
1	Connection between DST8B and channel 1 down-count start trigger B is enabled

- Bit 0—Connection Flag 8A (CN8A): Enables or disables connection between DST8A and the channel 1 down-count start trigger.

Bit 0: CN8A	Description
0	Connection between DST8A and channel 1 down-count start trigger A is disabled (Initial value)
1	Connection between DST8A and channel 1 down-count start trigger A is enabled

### 11.2.13 One-Shot Pulse Terminate Register (OTR)

The one-shot pulse terminate register (OTR) is a 16-bit register. The ATU-II has one OTR register in channel 8.

Bit:	15	14	13	12	11	10	9	8
	OTEP	OTEO	OTEN	OTEM	OTEL	OTEK	OTEJ	OTEI
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:	7	6	5	4	3	2	1	0
	OTEH	OTEG	OTEF	OTEE	OTED	OTEC	OTEB	OTEA
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

OTR is a 16-bit readable/writable register that enables or disables forced termination of channel 8 one-shot pulse output by channel 1 and 2 compare-match signals. When one-shot pulse output is forcibly terminated, the corresponding DSTR bit and down-counter are cleared, and the corresponding TSR8 bit is set. The channel 1 one-shot pulse terminate signal is generated by GR1A to GR1H compare-matches and OCR1 compare-match (see TRGMDR). The channel 2 one-shot pulse terminate signal is generated by GR2A to GR2H compare-matches. To generate the terminate signal with GR1A to GR1H and GR2A to GR2H, select the respective compare-matches in TIOR1A to TIOR1D.

OTR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

- Bit 15—One-Shot Pulse Terminate Enable P (OTEP): Enables or disables forced termination of output by channel 2 down-counter terminate trigger H.

Bit 15: OTEP	Description
0	Forced termination of TO8P by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8P by down-counter terminate trigger is enabled



- Bit 14—One-Shot Pulse Terminate Enable O (OTEO): Enables or disables forced termination of output by channel 2 down-counter terminate trigger G.

Bit 14: OTEO	Description
0	Forced termination of TO8O by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8O by down-counter terminate trigger is enabled

- Bit 13—One-Shot Pulse Terminate Enable N (OTEN): Enables or disables forced termination of output by channel 2 down-counter terminate trigger F.

Bit 13: OTEN	Description
0	Forced termination of TO8N by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8N by down-counter terminate trigger is enabled

- Bit 12—One-Shot Pulse Terminate Enable M (OTEM): Enables or disables forced termination of output by channel 2 down-counter terminate trigger E.

Bit 12: OTEM	Description
0	Forced termination of TO8M by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8M by down-counter terminate trigger is enabled

- Bit 11—One-Shot Pulse Terminate Enable L (OTEL): Enables or disables forced termination of output by channel 2 down-counter terminate trigger D.

Bit 11: OTEL	Description
0	Forced termination of TO8L by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8L by down-counter terminate trigger is enabled

- Bit 10—One-Shot Pulse Terminate Enable K (OTEK): Enables or disables forced termination of output by channel 2 down-counter terminate trigger C.

Bit 10: OTEK	Description
0	Forced termination of TO8K by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8K by down-counter terminate trigger is enabled

- Bit 9—One-Shot Pulse Terminate Enable J (OTEJ): Enables or disables forced termination of output by channel 2 down-counter terminate trigger B.

Bit 9: OTEJ	Description
0	Forced termination of TO8J by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8J by down-counter terminate trigger is enabled

- Bit 8—One-Shot Pulse Terminate Enable I (OTEI): Enables or disables forced termination of output by channel 2 down-counter terminate trigger A.

Bit 8: OTEI	Description
0	Forced termination of TO8I by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8I by down-counter terminate trigger is enabled

- Bit 7—One-Shot Pulse Terminate Enable H (OTEH): Enables or disables forced termination of output by channel 1 down-counter terminate trigger H.

Bit 7: OTEH	Description
0	Forced termination of TO8H by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8H by down-counter terminate trigger is enabled

- Bit 6—One-Shot Pulse Terminate Enable G (OTEG): Enables or disables forced termination of output by channel 1 down-counter terminate trigger G.

Bit 6: OTEG	Description
0	Forced termination of TO8G by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8G by down-counter terminate trigger is enabled

- Bit 5—One-Shot Pulse Terminate Enable F (OTEF): Enables or disables forced termination of output by channel 1 down-counter terminate trigger F.

Bit 5: OTEF	Description
0	Forced termination of TO8F by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8F by down-counter terminate trigger is enabled

- Bit 4—One-Shot Pulse Terminate Enable E (OTEE): Enables or disables forced termination of output by channel 1 down-counter terminate trigger E.

Bit 4: OTEE	Description
0	Forced termination of TO8E by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8E by down-counter terminate trigger is enabled

- Bit 3—One-Shot Pulse Terminate Enable D (OTED): Enables or disables forced termination of output by channel 1 down-counter terminate trigger D.

Bit 3: OTED	Description
0	Forced termination of TO8D by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8D by down-counter terminate trigger is enabled

- Bit 2—One-Shot Pulse Terminate Enable C (OTEC): Enables or disables forced termination of output by channel 1 down-counter terminate trigger C.

Bit 2: OTEC	Description
0	Forced termination of TO8C by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8C by down-counter terminate trigger is enabled

- Bit 1—One-Shot Pulse Terminate Enable B (OTEB): Enables or disables forced termination of output by channel 1 down-counter terminate trigger B.

Bit 1: OTEB	Description
0	Forced termination of TO8B by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8B by down-counter terminate trigger is enabled

- Bit 0—One-Shot Pulse Terminate Enable A (OTEAA): Enables or disables forced termination of output by channel 1 down-counter terminate trigger A.

Bit 0: OTEA	Description
0	Forced termination of TO8A by down-counter terminate trigger is disabled (Initial value)
1	Forced termination of TO8A by down-counter terminate trigger is enabled

### 11.2.14 Reload Enable Register (RLDENR)

The reload enable register (RLDENR) is an 8-bit register. The ATU-II has one RLDENR register in channel 8.

Bit:	7	6	5	4	3	2	1	0
	RLDEN	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

RLDENR is an 8-bit readable/writable register that enables or disables loading of the reload register8 (RLDR8) value into the down-counters (DCNT8I to DCNT8P). Loading is performed on generation of a channel 2 compare-match signal one-shot pulse start trigger. Reloading is not performed if there is no linkage with channel 2 (one-shot pulse function), or while the down-counter (DCNT8I to DCNT8P) is running.

RLDENR is initialized to H'00 by a power-on reset and in hardware standby mode and software standby mode.

- Bit 7—Reload Enable (RLDEN): Enables or disables loading of the RLDR value into DCNT8I to DCNT8P.

Bit 7: RLDEN	Description
0	Loading of reload register value into down-counters is disabled (Initial value)
1	Loading of reload register value into down-counters is enabled

- Bits 6 to 0—Reserved: These bits are always read as 0. The write value should always be 0.

## 11.2.15 Free-Running Counters (TCNT)

The free-running counters (TCNT) are 32- or 16-bit up- or up/down-counters. The ATU-II has 17 TCNT counters: one 32-bit TCNT in channel 0, and sixteen 16-bit TCNTs in each of channels 1 to 7 and 11. For details of the channel 10 free-running counters, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
0	TCNT0H, TCNT0L	32-bit up-counter (initial value H'00000000)
1	TCNT1A, TCNT1B	16-bit up-counters (initial value H'0000)
2	TCNT2A, TCNT2B	
3	TCNT3	
4	TCNT4	
5	TCNT5	
6	TCNT6A–D	16-bit up/down-counters (initial value H'0001)
7	TCNT7A–D	16-bit up-counters (initial value H'0001)
11	TCNT11	16-bit up-counter (initial value H'0000)

**Free-Running Counter 0 (TCNT0H, TCNT0L):** Free-running counter 0 (comprising TCNT0H and TCNT0L) is a 32-bit readable/writable register that counts on an input clock. The counter is started when the corresponding bit in the timer start register (TSTR1) is set to 1. The input clock is selected with prescaler register 1 (PSCR1).

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

When TCNT0 overflows (from H'FFFFFFFF to H'00000000), the OVF0 overflow flag in the timer status register (TSR0) is set to 1.

TCNT0 can only be accessed by a longword read or write. Word reads or writes should not be used.

TCNT0 is initialized to H'00000000 by a power-on reset, and in hardware standby mode and software standby mode.

**Free-Running Counters 1A, 1B, 2A, 2B, 3, 4, 5, 11 (TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, TCNT5, TCNT11):** Free-running counters 1A, 1B, 2A, 2B, 3, 4, 5, and 11 (TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, TCNT5, TCNT11) are 16-bit readable/writable registers that count on an input clock. Counting is started when the corresponding bit in the timer start register (TSTR1 or TSTR3) is set to 1. The input clock is selected with prescaler register 1 (PSCR1) and the timer control register (TCR).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:																
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

The TCNT1A, TCNT1B, TCNT2A, and TCNT2B counters are cleared if incremented during counter clear trigger input from channel 10.

TCNT3 to TCNT5 counter clearing is performed by a compare-match with the corresponding general register, according to the setting in TIOR.

When one of counters TCNT1A/1B/2A/2B/3/4/5/11 overflows (from H'FFFF to H'0000), the overflow flag (OVF) for the corresponding channel in the timer status register (TSR) is set to 1.

TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, TCNT5, and TCNT11 can only be accessed by a word read or write.

TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, TCNT5, and TCNT11 are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, and TCNT5 can count on external clock (TCLKA or TCLKB) input.

TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, and TCNT5 can count on an external interrupt clock (TI10) (AGCK) generated in channel 10 and on a channel 10 multiplied clock (AGCKM).

**Free-Running Counters 6A to 6D and 7A to 7D (TCNT6A to TCNT6D, TCNT7A to TCNT7D):** Free-running counters 6A to 6D and 7A to 7D (TCNT6A to TCNT6D, TCNT7A to TCNT7D) are 16-bit readable/writable registers. Channel 6 and 7 counts are started by the timer start register (TSTR2).

The clock input to channels 6 and 7 is selected with prescaler registers 2 and 3 (PSCR2, PSCR3) and timer control registers 6 and 7 (TCR6, TCR7).

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	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

TCNT6A to TCNT6D (in non-complementary PWM mode) and TCNT7A to TCNT7D are cleared by a compare-match with the cycle register (CYLR).

TCNT6A to TCNT6D (in complementary PWM mode) count up and down between zero and the cycle register value.

TCNT6A to TCNT6D and TCNT7A to TCNT7D are connected to the CPU by an internal 16-bit bus, and can only be accessed by a word read or write.

TCNT6A to TCNT6D and TCNT7A to TCNT7D are initialized to H'0001 by a power-on reset, and in hardware standby mode and software standby mode.

### 11.2.16 Down-Counters (DCNT)

The DCNT registers are 16-bit down-counters. The ATU-II has 16 DCNT counters in channel 8.

Channel	Abbreviation	Function
8	DCNT8A, DCNT8B, DCNT8C, DCNT8D, DCNT8E, DCNT8F, DCNT8G, DCNT8H, DCNT8I, DCNT8J, DCNT8K, DCNT8L, DCNT8M, DCNT8N, DCNT8O, DCNT8P	16-bit down-counters

**Down-Counters 8A to 8P (DCNT8A to DCNT8P):** Down-counters 8A to 8P (DCNT8A to DCNT8P) are 16-bit readable/writable registers that count on an input clock. The input clock is selected with prescaler register 1 (PSCR1) and the timer control register (TCR).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name:																
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

When the one-shot pulse function is used, DCNT8x starts counting down when the corresponding DSTR bit is set to 1 by the user program after the DCNT8x value has been set. When the DCNT8x value underflows, DSTR and DCNT8x are automatically cleared to 0, and the count is stopped. At the same time, the corresponding channel 8 timer status register 8 (TSR8) status flag is set to 1.

When the offset one-shot pulse function is used, on compare-match with a channel 1 or 2 general register (GR) or output compare register (OCR) (the compare-match setting being made in the trigger mode register (TRGMDR) (for channel 1 only) ) when the corresponding timer connection register (TCNR) bit is 1, the corresponding down-count start register (DSTR) bit is automatically set to 1 and the down-count is started. When the DCNT8x value underflows, the corresponding DSTR bit and DCNT8x are automatically cleared to 0, the count is stopped, and the output is inverted, or, if a one-shot terminate register (OTR) setting has been made to forcibly terminate output by means of a trigger, DSTR is cleared to 0 by a channel 1 or 2 compare-match between GR and OCR, the count is forcibly terminated, and the output is inverted. The output is inverted for whichever is first. When the output is inverted, the corresponding channel 8 TSR8 status flag is set to 1.

The DCNT8x counters can only be accessed by a word read or write.

The DCNT8x counters are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 11.3.5, One-Shot Pulse Function, and 11.3.6, Offset One-Shot Pulse Function and Output Cutoff Function.



## 11.2.17 Event Counters (ECNT)

The event counters (ECNT) are 8-bit up-counters. The ATU-II has six ECNT counters in channel 9.

Channel	Abbreviation	Function
9	ECNT9A, ECNT9B, ECNT9C, ECNT9D, ECNT9E, ECNT9F	8-bit event counters

The ECNT counters are 8-bit readable/writable registers that count on detection of an input signal from input pins TI9A to TI9F. Rising edge, falling edge, or both rising and falling edges can be selected for edge detection.

Bit:	7	6	5	4	3	2	1	0
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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

When a compare-match with GR9 corresponding to an ECNT9x counter occurs, the compare-match flag (CMF9) in the timer status register (TSR9) is set to 1. When a compare-match with GR occurs, the ECNT9x counter is cleared automatically.

The ECNT9x counters can only be accessed by a byte read or write.

The ECNT9x counters are initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

## 11.2.18 Output Compare Registers (OCR)

The output compare registers (OCR) are 16-bit registers. The ATU-II has nine OCR registers: one in channel 1 and eight in channel 2. For details of the channel 10 free-running counters, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
1	OCR1	Output compare registers
2	OCR2A, OCR2B, OCR2C, OCR2D, OCR2E, OCR2F, OCR2G, OCR2H	

## Output Compare Registers 1 and 2A to 2H (OCR1, OCR2A to OCR2H)

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

The OCR registers are 16-bit readable/writable registers that have an output compare register function.

The OCR and free-running counter (TCNT1B, TCNT2B) values are constantly compared, and if the two values match, the CMF bit in the timer status register (TSR) is set to 1. If channels 1 and 2 and channel 8 are linked by the timer connection register (TCNR), the corresponding channel 8 down-counter (DCNT) is started at the same time.

The OCR registers can only be accessed by a word read or write.

The OCR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

### 11.2.19 Input Capture Registers (ICR)

The input capture registers (ICR) are 32-bit registers. The ATU-II has four 32-bit ICR registers in channel 0. For details of the channel 10 free-running counters, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
0	ICR0AH, ICR0AL, ICR0BH, ICR0BL, ICR0CH, ICR0CL, ICR0DH, ICR0DL	Dedicated input capture registers

## Input Capture Registers 0AH, 0AL to 0DH, 0DL (ICR0AH, ICR0AL to ICR0DH, ICR0DL)

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

The ICR registers are 32-bit read-only registers used exclusively for input capture.

These dedicated input capture registers store the TCNT0 value on detection of an input capture signal from an external source. The corresponding TSR0 bit is set to 1 at this time. The input capture signal edge to be detected is specified by timer I/O control register TIOR0. By setting the TRG0DEN bit in TCR10, ICR0DH and ICR0DL can also be used for input capture in a compare match between TCNT10B and OCR10B.

The ICR registers can only be accessed by a longword read. Word reads should not be used.

The ICR registers are initialized to H'00000000 by a power-on reset, and in hardware standby mode and software standby mode.

### 11.2.20 General Registers (GR)

The general registers (GR) are 16-bit registers. The ATU-II has 36 general registers: eight each in channels 1 and 2, four each in channels 3 to 5, six in channel 9, and two in channel 11. For details of the channel 10 free-running counters, see section 11.2.26, Channel 10 Registers.

Channel	Abbreviation	Function
1	GR1A–GR1H	Dual-purpose input capture and output compare registers
2	GR2A–GR2H	
3	GR3A–GR3D	
4	GR4A–GR4D	
5	GR5A–GR5D	
9	GR9A–GR9F	Dedicated output compare registers
11	GR11A, GR11B	Dual-purpose input capture and output compare registers

## General Registers 1A to 1H and 2A to 2H (GR1A to GR1H, GR2A to GR2H)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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

These GR registers are 16-bit readable/writable registers with both input capture and output compare functions. Function switching is performed by means of the timer I/O control registers (TIOR).

When a general register is used for input capture, it stores the TCNT1A or TCNT2A value on detection of an input capture signal from an external source. The corresponding IMF bit in TSR is set to 1 at this time. The input capture signal edge to be detected is specified by the corresponding TIOR.

When a general register is used for output compare, the GR value and free-running counter (TCNT1A, TCNT2A) value are constantly compared, and when both values match, the IMF bit in the timer status register (TSR) is set to 1. If connection of channels 1 and 2 and channel 8 is specified in the timer connection register (TCNR), the corresponding channel 8 down-counter (DCNT) is started. Compare-match output is specified by the corresponding TIOR.

The GR registers can only be accessed by a word read or write.

The GR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

## General Registers 3A to 3D, 4A to 4D, 5A to 5D, 11A and 11B (GR3A to GR3D, GR4A to GR4D, GR5A to GR5D, GR11A and GR11B)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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

These GR registers are 16-bit readable/writable registers with both input capture and output compare functions. Function switching is performed by means of the timer I/O control registers (TIOR).

When a general register is used for input capture, it stores the corresponding TCNT value on detection of an input capture signal from an external source. The corresponding IMF bit in TSR is set to 1 at this time. The input capture signal edge to be detected is specified by the corresponding

TIOR. GR3A to GR3D can also be used for input capture with a channel 9 compare-match as the trigger. In this case, the corresponding IMF bit in TSR is not set.

When a general register is used for output compare, the GR value and free-running counter (TCNT) value are constantly compared, and when both values match, the IMF bit in the timer status register (TSR) is set to 1. Compare-match output is specified by the corresponding TIOR.

GRIIA and GR11B compare-match signals are transmitted to the advanced pulse controller (APC). For details, see section 12, Advanced Pulse Controller (APC).

The GR registers can only be accessed by a word read or write.

The GR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

### General Registers 9A to 9F (GR9A to GR9F)

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

These GR registers are 8-bit readable/writable registers with a compare-match function.

The GR value and event counter (ECNT) value are constantly compared, and when both values match a compare-match signal is generated and the next edge is input, the corresponding CMF bit in TSR is set to 1.

In addition, channel 3 (GR3A to GR3D) input capture can be generated by GR9A to GR9D compare-matches. This function is set by TRG3xEN in the timer control register (TCR).

The GR registers can be accessed by a byte read or write.

The GR registers are initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode.

### 11.2.21 Offset Base Registers (OSBR)

The offset base registers (OSBR) are 16-bit registers. The ATU-II has two OSBR registers, one each in channels 1 and 2.

Channel	Abbreviation	Function
1	OSBR1	Dedicated input capture registers with the same input trigger signal as that for channel 0 ICR0A
2	OSBR2	

#### Offset Base Registers 1 and 2 (OSBR1, OSBR2)

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

OSBR1 and OSBR2 are 16-bit read-only registers used exclusively for input capture. Same as the channel 0 input capture register (ICR0A), OSBR1 and OSBR2 use the TIOA input as their trigger signal, and store the TCNT1A or TCNT2A value on detection of an edge.

The OSBR registers can only be accessed by a word read.

The OSBR registers are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

For details, see sections 11.3.8, Twin Capture Function (TRGMDR).

### 11.2.22 Cycle Registers (CYLR)

The cycle registers (CYLR) are 16-bit registers. The ATU-II has eight cycle registers, four each in channels 6 and 7.

Channel	Abbreviation	Function
6	CYLR6A– CYLR6D	16-bit PWM cycle registers
7	CYLR7A– CYLR7D	

## Cycle Registers (CYLR6A to CYLR6D, CYLR7A to CYLR7D)

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

The CYLR registers are 16-bit readable/writable registers used for PWM cycle storage.

The CYLR value is constantly compared with the corresponding free-running counter (TCNT6A to TCNT6D, TCNT7A to TCNT7D) value, and when the two values match, the corresponding timer start register (TSR) bit (CMF6A to CMF6D, CMF7A to CMF7D) is set to 1, and the free-running counter (TCNT6A to TCNT6D, TCNT7A to TCNT7D) is cleared. At the same time, the buffer register (BFR) value is transferred to the duty register (DTR). The corresponding output pins (TO6A to TO6D, TO7A to TO7D) go to 0 output when the BFR value is H'0000. In other cases, they go to 1 output.

The CYLR registers can only be accessed by a word read or write.

The CYLR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

For details of the CYLR, BFR, and DTR registers, see section 11.3.9, PWM Timer Function.

### 11.2.23 Buffer Registers (BFR)

The buffer registers (BFR) are 16-bit registers. The ATU-II has eight buffer registers, four each in channels 6 and 7.

Channel	Abbreviation	Function
6	BFR6A–BFR6D	16-bit PWM buffer registers Buffer register (BFR) value is transferred to duty register (DTR) on compare-match of corresponding cycle register (CYLR)
7	BFR7A–BFR7D	

### Buffer Registers (BFR6A to BFR6D, BFR7A to BFR7D)

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

The BFR registers are 16-bit readable/writable registers that store the value to be transferred to the duty register (DTR) in the event of a cycle register (CYLR) compare-match.

The BFR registers can only be accessed by a word read or write.

The BFR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

### 11.2.24 Duty Registers (DTR)

The duty registers (DTR) are 16-bit registers. The ATU-II has eight duty registers, four each in channels 6 and 7.

Channel	Abbreviation	Function
6	DTR6A–DTR6D	16-bit PWM duty registers
7	DTR7A–DTR7D	

#### Duty Registers (DTR6A to DTR6D, DTR7A to DTR7D)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
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

The DTR registers are 16-bit readable/writable registers used for PWM duty storage.

The DTR value is constantly compared with the corresponding free-running counter (TCNT6A to TCNT6D, TCNT7A to TCNT7D) value, and when the two values match, the corresponding channel output pin (TO6A to TO6D, TO7A to TO7D) goes to 0 output. Also, when CYLR and the corresponding the free-running counter match, the corresponding BFR value is loaded. Set a value in the range 0 to CYLR for DTR; do not set a value greater than CYLR.

The DTR registers can only be accessed by a word read or write.

The DTR registers are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.



## 11.2.25 Reload Register (RLDR)

The reload register is a 16-bit register. The ATU-II has one RLDR register in channel 8.

### Reload Register 8 (RLDR8)

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

RLDR8 is a 16-bit readable/writable register. When reload is enabled (by a setting in RL DENR) and DSTR8I to DSTR8P are set to 1 by the channel 2 compare-match signal one-shot pulse start trigger, the reload register value is transferred to DCNT8I to DCNT8P before the down-count is started. The reload register value is not transferred when the one-shot pulse function is used independently, without linkage to channel 2, or when down-counters DCNT8I to DCNT8P are running.

RLDR8 can only be accessed by a word read or write.

RLDR is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

## 11.2.26 Channel 10 Registers

### Counters (TCNT)

Channel 10 has seven TCNT counters: one 32-bit TCNT, four 16-bit TCNTs, and two 8-bit TCNTs.

The input clock is selected with prescaler register 4 (PSCR4). Count operations are performed by setting STR10 to 1 in timer start register 1 (TSTR1).

Channel	Abbreviation	Function
10	TCNT10AH, AL	32-bit free-running counter (initial value H'00000001)
	TCNT10B	8-bit event counter (initial value H'00)
	TCNT10C	16-bit reload counter (initial value H'0001)
	TCNT10D	8-bit correction counter (initial value H'00)
	TCNT10E	16-bit correction counter (initial value H'0000)
	TCNT10F	16-bit correction counter (initial value H'0001)
	TCNT10G	16-bit free-running counter (initial value H'0000)

**Free-Running Counter 10AH, AL (TCNT10AH, TCNT10AL):** Free-running counter 10AH, AL (comprising TCNT10AH and TCNT10AL) is a 32-bit readable/writable register that counts on an input clock and is cleared to initial value by input capture input (TI10) (AGCK).

	Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16																		
		<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>																																	
Initial value:		0	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	R/W																	

	Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																		
		<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>																																	
Initial value:		0	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	R/W																	

TCNT10A can only be accessed by a longword read or write. Word reads or writes should not be used.

TCNT10A is initialized to H'00000001 by a power-on reset, and in hardware standby mode and software standby mode.

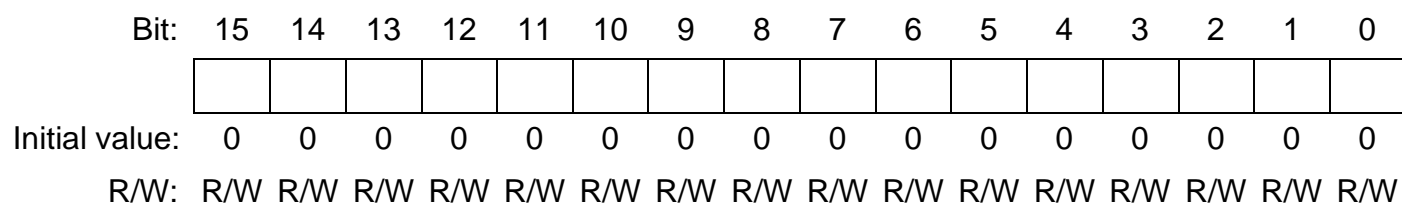
**Event Counter 10B (TCNT10B):** Event counter 10B (TCNT10B) is an 8-bit readable/writable register that counts on external clock input (TI10) (AGCK). For this operation, TI10 input must be set with bits CKEG1 and CKEG0 in TCR10. TI10 input will be counted even if halting of the count operation is specified by bit STR10 in TSTR1.

	Bit:	7	6	5	4	3	2	1	0										
		<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>																	
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										

TCNT10B can only be accessed by a byte read or write.

TCNT10B is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

**Reload Counter 10C (TCNT10C):** Reload counter 10C (TCNT10C) is a 16-bit readable/writable register.

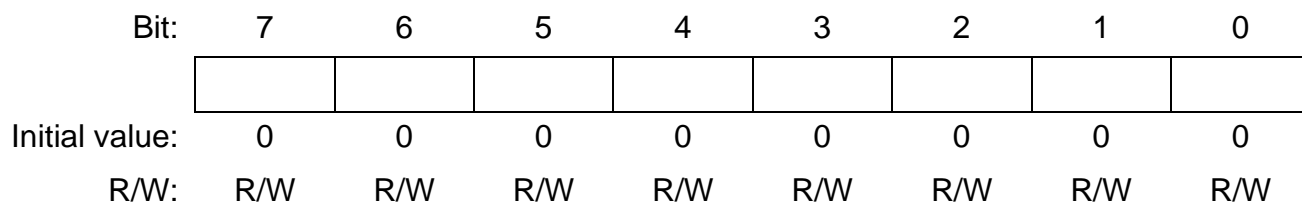


When TCNT10C = H'0001 in the down-count operation, the value in the reload register (RLD10C) is transferred to TCNT10C, and a multiplied clock (AGCK1) is generated.

TCNT10C is connected to the CPU via an internal 16-bit bus, and can only be accessed by a word read or write.

TCNT10C is initialized to H'0001 by a power-on reset, and in hardware standby mode and software standby mode.

**Correction Counter 10D (TCNT10D):** Correction counter 10D (TCNT10D) is an 8-bit readable/writable register that counts on external clock input (TI10) after transfer of the counter value to correction counter E (TCNT10E). Set TI10 input with bits CKEG1 and CKEG0 in TCR10. Transfer and counting will not be performed on TI10 input unless the count operation is enabled by bit STR10 in TSTR1.

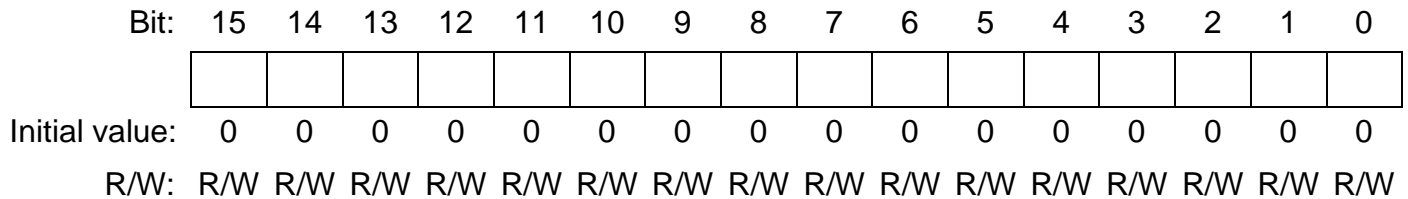


At the external clock input (TI10) (AGCK) timing, the value in this counter is shifted according to the multiplication factor set by bits PIM1 and PIM0 in timer I/O control register 10 (TIOR10) and transferred to correction counter E (TCNT10E).

TCNT10D can only be accessed by a byte read or write.

TCNT10D is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

**Correction Counter 10E (TCNT10E):** Correction counter 10E (TCNT10E) is a 16-bit readable/writable register that loads the TCNT10D shift value at the external input (TI10) timing, and counts on the multiplied clock (AGCK1) output by reload counter 10C (TCNT10C). However, if CCS in timer I/O control register 10 (TIOR10) is set to 1, when the TCNT10D shifted value is reached the count is halted.

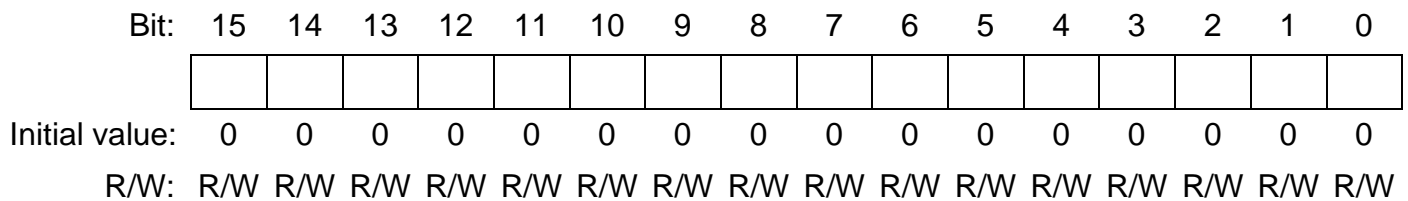


TCNT10E can only be accessed by a word read or write.

TCNT10E is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

**Correction Counter 10F (TCNT10F):** Correction counter 10F (TCNT10F) is a 16-bit readable/writable register that counts up on Pφ clock cycles if the counter value is smaller than the correction counter 10E (TCNT10E) value when the STR10 bit in TSTR1 has been set for counter operation. The count is halted by a match with the correction counter clear register (TCCLR10). If TI10 is input when TCNT10D = H'00, TCNT10F is initialized and correction is carried out. When TCNT10F = TCCLR10, TCNT10F is cleared to H'0001. While TCNT10F ≠ TCCLR10, TCNT10F is incremented automatically until it reaches the TCCLR10 value, and is then cleared to H'0001.

A corrected clock (AGCKM) is output following correction each time this counter is incremented.



TCNT10F is can only be accessed by a word read or write.

TCNT10F is initialized to H'0001 by a power-on reset, and in hardware standby mode and software standby mode.

**Free-Running Counter 10G (TCNT10G):** Free-running counter 10G (TCNT10G) is a 16-bit readable/writable register that counts up on the multiplied clock (AGCK1). TCNT10G is initialized to H'0000 by input from external input (TI10) (AGCK).

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

TCNT10G can only be accessed by a word read or write.

TCNT10G is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

## Registers

There are six registers in channel 10: a 32-bit ICR, 32-bit OCR, 16-bit GR, 16-bit RLD, 16-bit TCCLR, and 8-bit OCR.

Channel	Abbreviation	Function
10	ICR10AH, AL	32-bit input capture register (initial value H'00000000)
	OCR10AH, AL	32-bit output compare register (initial value H'FFFFFFFF)
	OCR10B	8-bit output compare register (initial value H'FF)
	RLD10C	16-bit reload register (initial value H'0000)
	GR10G	16-bit general register (initial value H'FFFF)
	TCCLR10	16-bit correction counter clear register (initial value H'0000)

**Input Capture Register 10AH, AL (ICR10AH, ICR10AL):** Input capture register 10AH, AL (comprising ICR10AH and ICR10AL) is a 32-bit read-only register to which the TCNT10AH, AL value is transferred on external input (TI10) (AGCK). At the same time, ICF10A in timer status register 10 (TSR10) is set to 1.

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

ICR10A is initialized to H'00000000 by a power-on reset, and in hardware standby mode and software standby mode.

**Output Compare Register 10AH, AL (OCR10AH, OCR10AL):** Output compare register 10AH, AL (comprising OCR10AH and OCR10AL) is a 32-bit readable/writable register that is constantly compared with free-running counter 10AH, AL (TCNT10AH, TCNT10AL). When both values match, CMF10A in timer status register 10 (TSR10) is set to 1.

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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

OCR10A is initialized to H'FFFFFFFF by a power-on reset, and in hardware standby mode and software standby mode.

**Output Compare Register 10B (OCR10B):** Output compare register 10B (OCR10B) is an 8-bit readable/writable register that is constantly compared with free-running counter 10B (TCNT10B). When AGCK is input with both values matching, CMF10B in timer status register 10 (TSR10) is set to 1.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

OCR10B is initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode.

**Reload Register 10C (RLD10C):** Reload register 10C (RLD10C) is a 16-bit readable/writable register. When STR10 in timer start register 1 (TSTR1) is 1 and RLDEN in the timer I/O control register (TIOR10) is 0, and the value of TCNT10A is captured into input capture register 10A (ICR10A), the ICR10A capture value is shifted according to the multiplication factor set by bits PIM1 and PIM0 in TIOR10 before being transferred to RLD10C. The contents of reload register 10C (RLD10C) are loaded when reload counter 10C (TCNT10C) reaches H'0001.

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

RLD10C is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

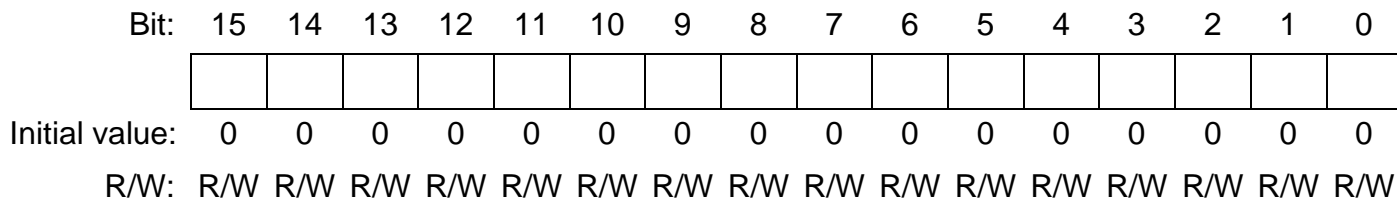
**General Register 10G (GR10G):** General register 10G (GR10G) is a 16-bit readable/writable register with an output compare function. Function switching is performed by means of timer I/O control register 10 (TIOR10). The GR10G value and free-running counter 10G (TCNT10G) value are constantly compared, and when AGCK is input with both values matching, CMF10G in timer status register 10 (TSR10) is set to 1.

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

GR10G is initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

**Correction Counter Clear Register 10 (TCCLR10):** Correction counter clear register 10 (TCCLR10) is a 16-bit readable/writable register.

TCCLR10 is constantly compared with TCNT10F, and when the two values match, TCNT10F halts. TCNT<sub>xx</sub> can be cleared at this time by setting TRG<sub>xx</sub>EN (xx = 1A, 1B, 2A, 2B) in TCR10. Then, when TCNT10D is H'00 and TI10 is input, TCNT10F is cleared to H'0001.



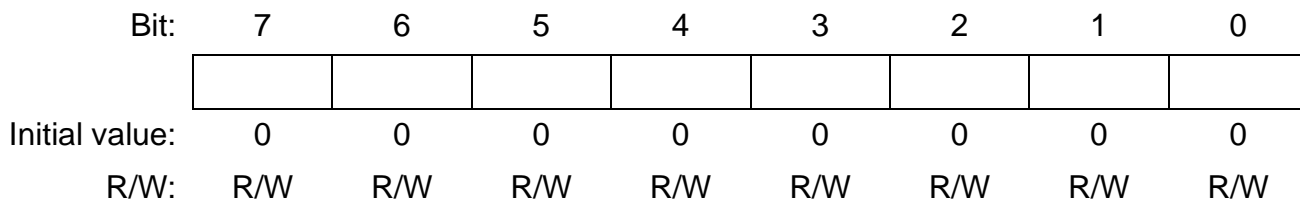
TCCLR10 is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

### Noise Canceler Registers

There are two 8-bit noise canceler registers in channel 10: TCNT10H and NCR10.

Channel	Abbreviation	Function	
10	TCNT10H	Noise canceler counter	(Initial value H'00)
	NCR10	Noise canceler compare-match register	(Initial value H'FF)

**Noise Canceler Counter 10H (TCNT10H):** Noise canceler counter 10H (TCNT10H) is an 8-bit readable/writable register. When the noise canceler function is enabled, TCNT10H starts counting up on  $P\phi \times 10$ , with the signal from external input (TI10) (AGCK) as a trigger. The counter operates even if STR10 is cleared to 0 in the timer start register (TSTR1). TI10 input is masked while the counter is running. When the count matches the noise canceler register (NCR10) value, the counter is cleared and TI10 input masking is released.



TCNT10H is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.



**Noise Canceler Register 10 (NCR10):** Noise canceler register 10 (NCR10) is an 8-bit readable/writable register used to set the upper count limit of noise canceler counter 10H (TCNT10H). TCNT10H is constantly compared with NCR10 during the count, and when a compare-match occurs the TCNT10H counter is halted and input signal masking is released.

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NCR10 is initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode.

### Channel 10 Control Registers

There are four control registers in channel 10.

Channel	Abbreviation	Function
10	TIOR10	Reload setting, counter correction setting, external input (TI10) edge interval multiplier setting GR compare-match setting (Initial value H'00)
	TCR10	TCCLR10 counter clear source Noise canceler function enabling/disabling selection External input (TI10) edge selection (Initial value H'00)
	TSR10	Input capture/compare-match status (Initial value H'0000)
	TIER10	Input capture/compare-match interrupt request enabling/disabling selection (Initial value H'0000)

**Timer I/O Control Register 10 (TIOR10):** TIOR10 is an 8-bit readable/writable register that selects the value for multiplication of the external input (TI10) edge interval. It also makes a setting for using the general register (GR10G) for output compare, and makes the edge detection setting.

TIOR10 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	RLDEN	CCS	PIM1	PIM0	—	IO10G2	IO10G1	IO10G0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

- Bit 7—Reload Enable (RLDEN): Enables or disables transfer of the input capture register 10A (ICR10A) value to reload register 10C (RLD10C).

Bit 7: RLDEN	Description
0	Transfer of ICR10A value to RLD10C on input capture is enabled (Initial value)
1	Transfer of ICR10A value to RLD10C on input capture is disabled

- Bit 6—Counter Clock Select (CCS): Selects the operation of correction counter 10E (TCNT10E). Set the multiplication factor with bits PIM1 and PIM0.

Bit 6: CCS	Description
0	TCNT10E count is not halted when TCNT10D x multiplication factor = TCNT10E* (Initial value)
1	TCNT10E count is halted when TCNT10D x multiplication factor = TCNT10E*

Note: \* When [TCNT10D × multiplication factor] matches the value of TCNT10E with bits 8 to 0 masked

- Bits 5 and 4—Pulse Interval Multiplier (PIM1, PIM0): These bits select the external input (TI10) cycle multiplier.

Bit 5: PIM1	Bit 4: PIM0	Description
0	0	Counting on external input cycle × 32 (Initial value)
	1	Counting on external input cycle × 64
1	0	Counting on external input cycle × 128
	1	Counting on external input cycle × 256

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.

- Bits 2 to 0—I/O Control 10G2 to 10G0 (IO10G2 to IO10G0): These bits select the function of general register 10G (GR10G).

Bit 2: IO10G2	Bit 1: IO10G1	Bit 0: IO10G0	Description
0	0	0	GR is an output compare register
		1	
	1	*	Cannot be used
1	*	*	Cannot be used

\*: Don't care

**Timer Control Register 10 (TCR10):** TCR10 is an 8-bit readable/writable register that selects the correction counter clear register (TCCLR10) compare-match counter clear source, enables or disables the noise canceler function, and selects the external input (TI10) edge.

TCR10 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	TRG2BEN	TRG1BEN	TRG2AEN	TRG1AEN	TRG0DEN	NCE	CKEG1	CKEG0
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 7—Trigger 2B Enable (TRG2BEN): Enables or disables counter clearing for channel 2 TCNT2B. When clearing is enabled, set the correction angle clock (AGCKM) as the TCNT2B count clock. If TCNT2B counts while clearing is enabled, TCNT2B will be cleared.

Bit 7: TRG2BEN	Description
0	Channel 2 counter B (TCNT2B) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is disabled (Initial value)
1	Channel 2 counter B (TCNT2B) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is enabled

- Bit 6—Trigger 1B Enable (TRG1BEN): Enables or disables counter clearing for channel 1 TCNT1B. When clearing is enabled, set the correction angle clock (AGCKM) as the TCNT1B count clock. If TCNT1B counts while clearing is enabled, TCNT1B will be cleared.

Bit 6: TRG1BEN	Description
0	Channel 1 counter B (TCNT1B) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is disabled (Initial value)
1	Channel 1 counter B (TCNT1B) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is enabled

- Bit 5—Trigger 2A Enable (TRG2AEN): Enables or disables counter clearing for channel 2 TCNT2A. When clearing is enabled, set the correction angle clock (AGCKM) as the TCNT2A count clock. If TCNT2A counts while clearing is enabled, TCNT2A will be cleared.

Bit 5: TRG2AEN	Description
0	Channel 2 counter 2A (TCNT2A) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is disabled (Initial value)
1	Channel 2 counter 2A (TCNT2A) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is enabled

- Bit 4—Trigger 1A Enable (TRG1AEN): Enables or disables counter clearing for channel 1 TCNT1A. When clearing is enabled, set the correction angle clock (AGCKM) as the TCNT1A count clock. If TCNT1A counts while clearing is enabled, TCNT1A will be cleared.

Bit 4: TRG1AEN	Description
0	Channel 1 counter 1A (TCNT1A) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is disabled (Initial value)
1	Channel 1 counter 1A (TCNT1A) clearing when correction counter clear register (TCCLR10) = correction counter (TCNT10F) is enabled

- Bit 3—Trigger 0D Enable (TRG0DEN): Enables or disables channel 0 ICR0D input capture signal requests.

Bit 3: TRG0DEN	Description
0	Capture requests for channel 0 input capture register (ICR0D) on event counter (TCNT10B) compare-match are disabled (Initial value)
1	Capture requests for channel 0 input capture register (ICR0D) on event counter (TCNT10B) compare-match are enabled

- Bit 2—Noise Canceler Enable (NCE): Enables or disables the noise canceler function.

Bit 2: NCE	Description
0	Noise canceler function is disabled (Initial value)
1	Noise canceler function is enabled

- Bits 1 and 0—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select the channel 10 external input (TI10) edge(s). The clock (AGCK) is generated by the detected edge(s).

Bit 1: CKEG1	Bit 0: CKEG0	Description
0	0	TI10 input disabled (Initial value)
	1	TI10 input rising edges detected
1	0	TI10 input falling edges detected
	1	TI10 input rising and falling edges both detected

**Timer Status Register 10 (TSR10):** TSR10 is a 16-bit readable/writable register that indicates the occurrence of channel 10 input capture or compare-match.

Each flag is an interrupt source, and issues an interrupt request to the CPU if the interrupt is enabled by the corresponding bit in timer interrupt enable register 10 (TIER10).

TSR10 is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	CMF10G	CMF10B	ICF10A	CMF10A
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: \* Only 0 can be written to clear the flag.

- Bits 15 to 4—Reserved: These bits are always read as 0. The write value should always be 0.

- Bit 3—Compare-Match Flag 10G (CMF10G): Status flag that indicates GR10G compare-match.

Bit 3: CMF10G	Description
0	[Clearing condition] (Initial value) When CMF10G is read while set to 1, then 0 is written to IMF10G
1	[Setting condition] When TCNT10G = GR10G

- Bit 2—Compare-Match Flag 10B (CMF10B): Status flag that indicates OCR10B compare-match.

Bit 2: CMF10B	Description
0	[Clearing condition] (Initial value) When CMF10B is read while set to 1, then 0 is written to CMF10B
1	[Setting condition] When TCNT10B is incremented while TCNT10B = OCR10B

- Bit 1—Input Capture Flag 10A (ICF10A): Status flag that indicates ICR10A input capture.

Bit 1: ICF10A	Description
0	[Clearing condition] (Initial value) When ICR10A is read while set to 1, then 0 is written to ICR10A
1	[Setting condition] When the TCNT10A value is transferred to ICR10A by an input capture signal

- Bit 0—Compare-Match Flag 10A (CMF10A): Status flag that indicates OCR10A compare-match.

Bit 0: CMF10A	Description
0	[Clearing condition] (Initial value) When CMF10A is read while set to 1, then 0 is written to CMF10A
1	[Setting condition] When TCNT10A = OCR10A

**Timer Interrupt Enable Register 10 (TIER10):** TIER10 is a 16-bit readable/writable register that controls enabling/disabling of channel 10 input capture and compare-match interrupt requests.

TIER10 is initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	IREG	CME10G	CME10B	ICE10A	CME10A
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

- Bits 15 to 5—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 4—Interrupt Enable Edge G (IREG): Specifies TSR10 CMF10G interrupt request timing.

Bit 4: IREG	Description
0	Interrupt is requested when CMF10G becomes 1 (Initial value)
1	Interrupt is requested by next external input (TI10) (AGCK) after CMF10G becomes 1

- Bit 3—Compare-Match Interrupt Enable 10G (CME10G): Enables or disables interrupt requests by CMF10G in TSR10 when CMF10G is set to 1.

Bit 3: CME10G	Description
0	CMI10G interrupt requested by CMF10G is disabled (Initial value)
1	CMI10G interrupt requested by CMF10G is enabled

- Bit 2—Compare-Match Interrupt Enable 10B (CME10B): Enables or disables interrupt requests by CMF10B in TSR10 when CMF10B is set to 1.

Bit 2: CME10B	Description
0	CMI10B interrupt requested by CMF10B is disabled (Initial value)
1	CMI10B interrupt requested by CMF10B is enabled

- Bit 1—Input Capture Interrupt Enable 10A (ICE10A): Enables or disables interrupt requests by ICF10A in TSR10 when ICF10A is set to 1.

<b>Bit 1: ICE10A</b>	<b>Description</b>
0	ICI10A interrupt requested by ICF10A is disabled (Initial value)
1	ICI10A interrupt requested by ICF10A is enabled

- Bit 0—Compare-Match Interrupt Enable 10A (CME10A): Enables or disables interrupt requests by CMF10A in TSR10 when CMF10A is set to 1.

<b>Bit 0: CME10A</b>	<b>Description</b>
0	CMI10A interrupt requested by CMF10A is disabled (Initial value)
1	CMI10A interrupt requested by CMF10A is enabled



## 11.3 Operation

### 11.3.1 Overview

The ATU-II has twelve timers of eight kinds in channels 0 to 11. It also has a built-in prescaler that generates input clocks, and it is possible to generate or select internal clocks of the required frequency independently of circuitry outside the ATU-II.

The operation of each channel and the prescaler is outlined below.

**Channel 0:** Channel 0 has a 32-bit free-running counter (TCNT0) and four 32-bit input capture registers (ICR0A to ICR0D). TCNT0 is an up-counter that performs free-running operation. An interrupt request can be generated on counter overflow. The four input capture registers (ICR0A to ICR0D) capture the free-running counter (TCNT0) value by means of input from the corresponding external signal input pin (TIOA to TIOD). For capture by means of input from an external signal input pin, rising edge, falling edge, or both edges can be selected in the timer I/O control register (TIOR0). In the case of input capture register 0D (ICR0D) only, capture can be performed by means of a compare-match between free-running counter 10B (TCNT10B) and compare-match register 10B (OCR10B), by making a setting in timer control register 10 (TCR10). In this case, capture is performed even if an input capture disable setting has been made for TIOR0. In each case, the DMAC can be activated or an interrupt requested when capture occurs.

Channel 0 also has three interval interrupt request registers (ITVRR1, ITVRR2A, and ITVRR2B). A/D converter (AD0 to AD2) activation can be selected by setting 1 in ITVA6 to ITVA13 in ITVRR, and an interrupt request to the CPU by setting 1 in ITVE6 to ITVE13. These operations are performed when the corresponding bit of bits 6 to 13 in TCNT0 changes to 1, enabling use as an interval timer function.

**Channel 1:** Channel 1 has two 16-bit free-running counters (TCNT1A and TCNT1B), eight 16-bit general registers (GR1A to GR1H), and a 16-bit output compare register (OCR1).

TCNT1A and TCNT1B are up-counters that perform free-running operation. When the clock generated in channel 10 (described below) is selected, these counters can be cleared at the count specified in channel 10. Each counter can generate an interrupt request when it overflows.

The eight general registers (GR1A to GR1H) can be used as input capture or output compare registers using the corresponding external signal I/O pin (TIO1A to TIO1H). When used for input capture, the free-running counter (TCNT1A) value is captured by means of input from the corresponding external signal I/O pin (TIO1A to TIO1H). Rising edge, falling edge, or both edges can be selected for the input capture signal in the timer I/O control registers (TIOR1A to TIOR1D). When used for output compare, compare-match with the free-running counter (TCNT1A) is performed. For the output from the external signal I/O pins by compare-match, 0 output, 1 output, or toggle output can be selected in the timer I/O control registers (TIOR1A to

TIOR1D). When used as output compare registers, a compare-match can be used as a one-shot pulse start/terminate trigger by setting the channel 8 timer connection register (TCNR) and one-shot pulse terminate register (OTR), and using these in combination with the down-counters (DCNT8A to DCNT8H). Start/terminate trigger selection is performed by means of the trigger mode register (TRGMDR).

In the case of the output compare register (OCR1), a TCNT1B compare-match can be used as a one-shot pulse start trigger, in the same way as the general registers, in combination with channel 8 down-counters DCNT8A to DCNT8H. An interrupt can be requested on the occurrence of the respective input capture or compare-match.

In addition, channel 1 has a 16-bit dedicated input capture register (OSBR1). The channel 0 TIOA input pin can also be used as the OSBR1 trigger input, enabling use of a twin-capture function.

**Channel 2:** Channel 2 has two 16-bit free-running counters (TCNT2A and TCNT2B), eight 16-bit general registers (GR2A to GR2H), and eight 16-bit output compare registers (OCR2A to OCR2H).

TCNT2A and TCNT2B are up-counters that perform free-running operation. When the clock generated in channel 10 (described below) is selected, these counters can be cleared at the count specified in channel 10. Each counter can generate an interrupt request when it overflows.

The eight general registers (GR2A to GR2H) can be used as input capture or output compare registers using the corresponding external signal I/O pin (TIO2A to TIO2H). When used for input capture, the free-running counter (TCNT2A) value is captured by means of input from the corresponding external signal I/O pin (TIO2A to TIO2H). Rising edge, falling edge, or both edges can be selected for the input capture signal in the timer I/O control registers (TIOR2A to TIOR2D). When used for output compare, compare-match with the free-running counter (TCNT2A) is performed. For the output from the external signal I/O pins by compare-match, 0 output, 1 output, or toggle output can be selected in the timer I/O control registers (TIOR2A to TIOR2D). When used as output compare registers, a compare-match can be used as a one-shot pulse terminate trigger by setting the channel 8 one-shot pulse terminate register (OTR), and using this in combination with the down-counters (DCNT8I to DCNT8P).

In the case of the output compare registers (OCR2A to OCR2H), a TCNT2B compare-match can be used as a one-shot pulse start trigger by setting the channel 8 timer connection register (TCNR), and using this in combination with the down-counters (DCNT8I to DCNT8P). An interrupt can be requested on the occurrence of the respective input capture or compare-match.

In addition, channel 2 has a 16-bit dedicated input capture register (OSBR2). The channel 0 TIOA input pin can also be used as the OSBR2 trigger input, enabling use of a twin-capture function.

**Channels 3 to 5:** Channels 3 to 5 each have a 16-bit free-running counter (TCNT3 to TCNT5) and four 16-bit general registers (GR3A to GR3D, GR4A to GR4D, GR5A to GR5D). TCNT3 to TCNT5 are up-counters that perform free-running operation. Channels 3 to 5 each have a 16-bit

free-running counter (TCNT3 to TCNT5) and four 16-bit general registers (GR3A to GR3D, GR4A to GR4D, GR5A to GR5D). TCNT3 to TCNT5 are up-counters that perform free-running operation. In addition, counter clearing can be performed by compare-match by making a setting in the timer I/O control register (TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B). Each counter can generate an interrupt request when it overflows.

The four general registers (GR3A to GR3D, GR4A to GR4D, GR5A to GR5D) each have corresponding external signal I/O pins (TIO3A to TIO3D, TIO4A to TIO4D, TIO5A to TIO5D), and can be used as input capture or output compare registers. When used for input capture, the free-running counter (TCNT3 to TCNT5) value is captured by means of input from the corresponding external signal I/O pin (TIO3A to TIO3D, TIO4A to TIO4D, TIO5A to TIO5D). Rising edge, falling edge, or both edges can be selected for the input capture signal in the timer I/O control registers (TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B). Also, in use for input capture, input capture can be performed using a compare-match between a channel 9 event counter (ECNT9A to ECNT9D), described later, and a general register (GR9A to GR9D) as the trigger (channel 3 only). In this case, capture is performed even if an input capture disable setting has been made for TIOR3A to TIOR3D. When used for output compare, compare-match with the free-running counter (TCNT3 to TCNT5) is performed. For the output from the external signal I/O pins by compare-match, 0 output, 1 output, or toggle output can be selected in the timer I/O control registers (TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B). An interrupt can be requested on the occurrence of the respective input capture or compare-match. However, in the case of input capture using channel 9 as a trigger, an interrupt request from channel 3 cannot be used.

By selecting PWM mode in the timer mode register (TMDR), PWM output can be obtained, with three outputs for each. In this case, GR3D, GR4D, and GR5D are automatically used as cycle registers, and GR3A to GR3C, GR4A to GR4C, GR5A to GR5C, as duty registers. TCNT3 to TCNT5 are cleared by the corresponding GR3D, GR4D, or GR5D compare-match.

**Channels 6 and 7:** Channels 6 and 7 each have 16-bit free-running counters (TCNT6A to TCNT6D, TCNT7A to TCNT7D), 16-bit cycle registers (CYLR6A to CYLR6D, CYLR7A to CYLR7D), 16-bit duty registers (DTR6A to DTR6D, DTR7A to DTR7D), and buffer registers (BFR6A to BFR6D, BFR7A to BFR7D). Channels 6 and 7 also each have external output pins (TO6A to TO6D, TO7A to TO7D), and can be used as buffered PWM timers. The TCNT registers are up-counters, and 0 is output to the corresponding external output pin when the TCNT value matches the DTR value (when  $DTR \neq CYLR$ ). When the TCNT value matches the CYLR value (when  $DTR \neq H'0000$ ), 1 is output to the external output pin, TCNT is initialized to H'0001, and the BFR value is transferred to DTR. Thus, the configuration of channels 6 and 7 enables them to perform waveform output with the CYLR value as the cycle and the DTR value as the duty, and to use BFR to absorb the time lag between setting of data in DTR and compare-match occurrence.

When  $DTR = CYLR$ , 1 is output continuously to the external output pin, giving a duty of 100%. When  $DTR = H'0000$ , 0 is output continuously to the external output pin, giving a duty of 0%. Do not set a value in DTR that will result in the condition  $DTR > CYLR$ . To set H'0000 to DTR, not

write H'0000 directly to DTR but set H'0000 to BFR and then transfer the value to DTR. Writing H'0000 directly to DTR may not give a duty of 0%.

In channel 6, TCNT can also be designated for complementary PWM output by means of the PWM mode register (PMDR). When the corresponding TSTR is set to 1, TCNT starts counting up, then switches to a down-count when the count matches the CYLR value. When TCNT reaches H'0000, it starts counting up again. When TCNT = DTR, the corresponding TO6A to TO6D output changes. Whether TCNT is counting up or down can be ascertained from the timer status register (TSR6).

DMAC activation and interrupt request generation, respectively, are possible when TCNT = CYLR in asynchronous PWM mode, and when TCNT = H'0000 in complementary PWM mode.

**Channel 8:** Channel 8 has sixteen 16-bit down-counters (DCNT8A to DCNT8P). The down-counters have corresponding external signal output pins, and can generate one-shot pulses. Setting a value in DCNT and setting the corresponding bit to 1 in the down-count start register (DSTR) starts DCNT operation and simultaneously outputs 1 to the external output pin. When DCNT counts down to H'0000, it stops and outputs 0 to the external output pin. An interrupt can be requested when DCNT underflows.

Down-counter operation can be coupled with the channel 1 or channel 2 output compare function by means of settings in the timer connection register (TCNR) and one-shot pulse terminate register (OTR), respectively, so that DCNT8I to DCNT8H count operations are started and stopped from channel 1, and DCNT8I to DCNT8P count operations from channel 2.

DCNT8I to DCNT8P have a reload register (RLDR), and a setting in the reload enable register (RLDEN) enables count operations to be started after reading the value from this register.

**Channel 9:** Channel 9 has six 8-bit event counters (ECNT9A to ECNT9F) and six 8-bit general registers (GR9A to GR9F). The event counters are up-counters, each with a corresponding external input pin (ECNT9A to ECNT9F). The event counter value is incremented by input from the corresponding external input pin. Incrementing on the rising edge, falling edge, or both edges can be selected by means of settings in the timer control registers (TCR9A to TCR9C). An event counter is cleared by edge input after a match with the corresponding general register. An interrupt can be requested when an event counter is cleared.

Timer control register (TCR9A, TCR9B) settings can be made to enable event counters ECNT9A to ECNT9D to send a compare-match signal to channel 3 when the count matches the corresponding general register (GR9A to GR9D), allowing input capture to be performed on channel 3. This enables the pulse input interval to be measured.

**Channel 10:** Channel 10 generates a multiplied clock based on external input, and supplies this to channels 1 to 5. Channel 10 is divided into three blocks: (1) an inter-edge measurement block, (2) a multiplied clock generation block, and (3) a multiplied clock correction block.

## (1) Inter-edge measurement block

This block has a 32-bit free-running counter (TCNT10A), 32-bit input capture register (ICR10A), 32-bit output compare register (OCR10A), 8-bit event counter (TCNT10B), 8-bit output compare register (OCR10B), 8-bit noise canceler counter (TCNT10H), and 8-bit noise canceler compare-match register (NCR10).

The 32-bit free-running counter (TCNT10A) is an up-counter that performs free-running operations. When input capture is performed by means of TI10 input, this counter is cleared to H'00000001. When free-running counter (TCNT10A) reaches the value set in the output compare register (OCR10A), a compare-match interrupt can be requested.

The input capture register (ICR10A) has an external signal input pin (TI10), and the free-running counter (TCNT10A) value can be captured by means of input from TI10. Rising edge, falling edge, or both edges can be selected by making a setting in bits CKEG1 and CKEG0 in the timer control register (TCR10). The TI10 input has a noise canceler function, which can be enabled by setting the NCE bit in the timer control register (TCR10). When the counter value is captured, TCNT10A is cleared to 0 and an interrupt can be requested. The captured value can be transferred to the multiplied clock generation block reload register (RLD10C).

The 8-bit event counter (TCNT10B) is an up-counter that is incremented by TI10 input. When the event counter (TCNT10B) value reaches the value set in the output compare register (OCR10B), a compare-match interrupt can be requested. By setting the TRG0DEN bit in the timer control register (TCR10), a capture request can also be issued for the channel 0 input capture register 0D (ICR0D) when compare-match occurs.

The 8-bit noise canceler counter (TCNT10H) and 8-bit noise canceler compare-match register (NCR10) are used to set the period for which the noise canceler functions. By setting a value in the noise canceler compare-match register (TCNT10H) and setting the NCE bit in the timer control register (TCR10), TI10 input is masked when it occurs. At the same time as TI10 input is masked, the noise canceler counter (TCNT10H) starts counting up on the Pφx10 clock. When the noise canceler counter (TCNT10H) value matches the noise canceler compare-match register (NCR10) value, the noise canceler counter (TCNT10H) is cleared to H'0000 and TI10 input masking is cleared.

## (2) Multiplied clock generation block

This block has 16-bit reload counters (TCNT10C, RLD10C), a 16-bit register free-running counter (TCNT10G), and a 16-bit general register (GR10G).

16-bit reload counter 10C (RLD10C) is captured by 32-bit input capture register 10A (ICR10A), and when RLDEN in the timer I/O control register (TIOR10) is 0, the value captured in input capture register 10A is transferred to the multiplied clock generation block reload register (RLD10C). The value transferred can be selected from 1/32, 1/64, 1/128, or 1/256 the original value, according to the setting of bits PIM1 and PIM0 in TIOR10.

16-bit reload counter 10C (TCNT10C) performs down-count operations. When TCNT10C reaches H'0001, the value is read automatically from the reload buffer (RLD10C), internal clock AGCK1 is generated, and the down-count operation is repeated. Internally generated AGCK1 is input as a clock to the multiplied clock correction block 16-bit correction counter (TCNT10E) and 16-bit free-running counter 10G (TCNT10G).

16-bit register free-running counter 10G (TCNT10G) counts on AGCK1 generated by TCNT10C. It is initialized to H'0000 by external input from TI10.

The 16-bit general register (GR10G) can be used in a compare-match with free-running counter 10G (TCNT10G) by setting bits IO10G2 to IO10G0 in the timer I/O control register (TIOR10). An interrupt can be requested when a compare-match occurs. Also, by setting timer interrupt enable register 10 (TIER10), an interrupt can be request in the event of TI10 input after a compare-match.

### (3) Multiplied clock correction block

This block has three 16-bit correction counters (TCNT10D, TCNT10E, TCNT10F) and a 16-bit correction counter clear register (TCCLR10). When 32-bit input capture register 10A (ICR10A) performs a capture operation due to input from external input pin TI10, the value in correction counter 10D (TCNT10D) is transferred to TCNT10E and TCNT10D is incremented. The value transferred to TCNT10E is 32, 64, 128, or 256 times the TCNT10D value, according to the setting of bits PIM1 and PIM0 in the timer I/O control register (TIOR10).

16-bit correction counter 10E (TCNT10E) counts up on AGCK1 generated by reload counter 10C (TCNT10C, RLD10C) in the multiplied clock generation block. However, by setting the CCS bit in the timer I/O control register (TIOR10), it is possible to stop free-running counter 10E (TCNT10E) when the free-running counter 10D (TCNT10D) multiplication value specified by PIM1 and PIM0 and the free-running counter 10E (TCNT10E) value match. The multiplied TCNT10D value is transferred when input capture register 10A (ICR10A) performs a capture operation due to TI10 input.

16-bit correction counter 10F (TCNT10F) has  $P\phi$  as its input and is constantly compared with 16-bit correction counter 10E (TCNT10E). When the 16-bit correction counter 10F (TCNT10F) value is smaller than that in 16-bit correction counter 10E (TCNT10E), it is incremented and generates count-up AGCKM. When the 16-bit correction counter 10F (TCNT10F) value exceeds that in 16-bit correction counter 10E (TCNT10E), no count-up operation is performed. The TI10 multiplied signal (AGCKM) generated when TCNT10F is incremented is output to the channel 1 to 5 free-running counters (TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3, TCNT4, TCNT5), and an up-count can be performed on AGCKM by setting this as the counter clock on each channel. TCNT10F is constantly compared with the 16-bit correction counter clear register (TCCLR10), and when the free-running counter 10F (TCNT10F) and correction counter clear register (TCCLR10) values match, the TCNT10F up-count stops. Setting TRG1AEN, TRG1BEN, TRG2AEN, and TRG2BEN in the timer control register (TCR10) enables the channel 1 and 2 free-running counters (TCNT1A, TCNT1B, TCNT2A, TCNT2B) to be cleared at this time. If TI10 is input when TCNT10D = H'0000, initialization and correction operations are performed. When TCNT10F = TCCLR10, TCNT10F is cleared to H'0001. When TCNT10F  $\neq$  TCCLR10, TCNT10F automatically counts up to the TCCLR10 value, and is cleared to H'0001.

**Channel 11:** Channel 11 has a 16-bit free-running counter (TCNT11) and two 16-bit general registers (GR11A and GR11B). TCNT11 is an up-counter that performs free-running operation. The counter can generate an interrupt request when it overflows. The two general registers (GR11A and GR11B) each have a corresponding external signal I/O pin (TIO11A, TIO11B), and can be used as input capture or output compare registers.

When used for input capture, the free-running counter (TCNT11) value is captured by means of input from the corresponding external signal I/O pin (TIO11A, TIO11B). Rising edge, falling edge, or both edges can be selected for the input capture signal in the timer I/O control register (TIOR11). When used for output compare, compare-match with the free-running counter (TCNT11) is performed. For the output from the external signal I/O pins by compare-match, 0 output, 1 output, or toggle output can be selected in the timer I/O control register (TIOR11). An interrupt can be requested on the occurrence of the respective input capture or compare-match. When the two general registers (GR11A and GR11B) are designated for compare-match use, a compare-match signal can be output to the APC.

**Prescaler:** The ATU-II has a dedicated prescaler with a 2-stage configuration. The first stage comprises 5-bit prescalers (PSCR1 to PSCR4) that generate a  $1/m$  clock (where  $m = 1$  to  $32$ ) with respect to clock  $P\phi$ . The second prescaler stage allows selection of a clock obtained by further scaling the clock from the first stage by  $2^n$  (where  $n = 0$  to  $5$ ) according to the timer control registers for the respective channels (TCR1A, TCR1B, TCR2A, TCR2B, TCR3 to TCR5, TCR6A, TCR6B, TCR7A, TCR7B, TCR8, TCR11).

The prescalers of channels 1 to 8 and 11 have a 2-stage configuration, while the channel 0 and 10 prescalers only have a first stage. The first-stage prescaler is common to channels 0 to 5, 8, and

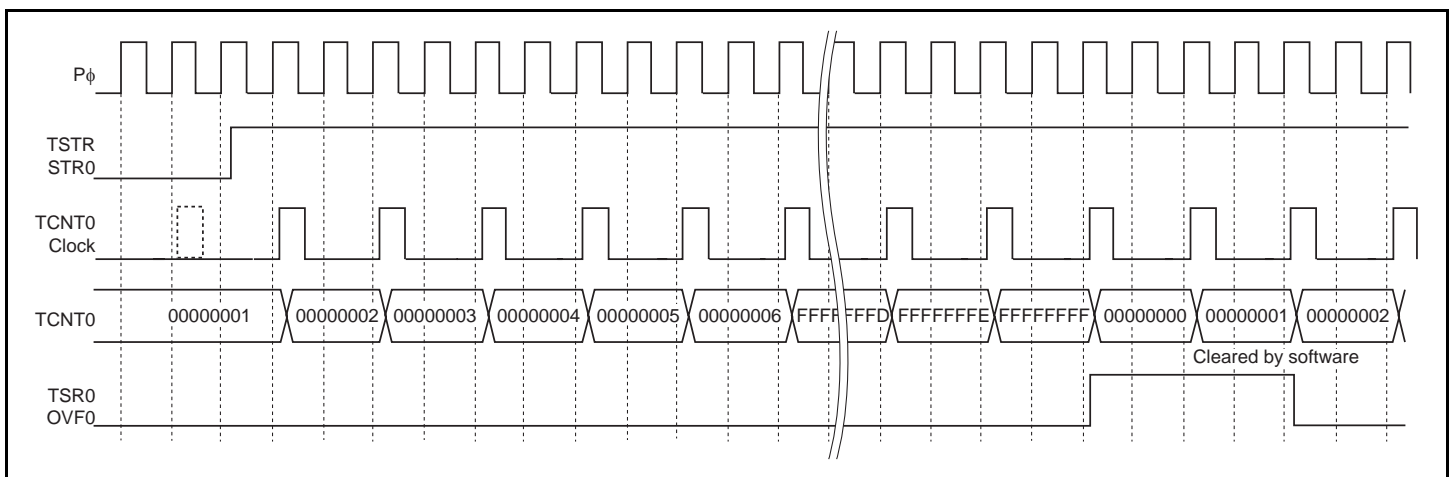
11, and it is not possible to set different first-stage division ratios for each. Channels 6, 7, and 10 each have a first-stage prescaler, and different first-stage division ratios can be set for each.

### 11.3.2 Free-Running Counter Operation and Cyclic Counter Operation

The free-running counters (TCNT) in ATU-II channels 0 to 5 and 11 start counting up as free-running counters when the corresponding timer start register (TSTR) bit is set to 1. When TCNT overflows (channel 0: from H'FFFFFFFF to H'00000000; channels 1 to 5 and 11: from H'FFFF to H'0000), the OVF bit in the timer status register (TSR) is set to 1. If the OVE bit in the corresponding timer interrupt enable register (TIER) is set to 1 at this time, an interrupt request is sent to the CPU. After overflowing, TCNT starts counting up again from H'00000000 or H'0000.

If the TSTR value is cleared to 0 during TCNT operation, the corresponding TCNT halts. In this case, TCNT is not reset. If external output is being performed from the GR for the corresponding TCNT, the output value does not change.

Channel 0 free-running counter operation is shown in figure 11.13.



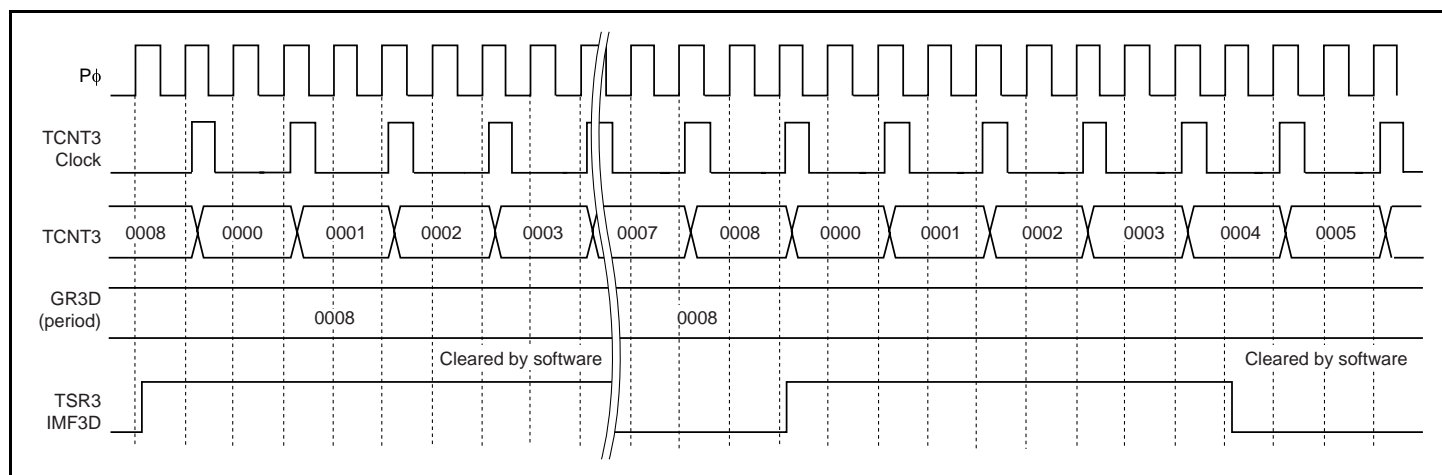
**Figure 11.13 Free-Running Counter Operation and Overflow Timing**

The free-running counters (TCNT) in ATU-II channels 6 and 7 perform cyclic count operations unconditionally. With channel 3 to 5 free-running counters (TCNT), when the corresponding T3PWM to T5PWM bit in the timer mode register (TMDR) is set to 1, or the corresponding CCI bit in the timer I/O control register (TIOR) is set to 1 when bits T3PWM to T5PWM are 0, the counter for the relevant channel performs a cyclic count. The relevant TCNT counter is cleared by a compare-match of TCNT with GR3D, GR4D, or GR5D in channel 3 to 5, or CYLR in channels 6 and 7 (counter clear function). TCNT starts counting up as a cyclic counter when the corresponding STR bit in TSTR is set to 1 after the TMDR setting is made. When the count value matches the GR3D, GR4D, GR5D, or CYLR value, the corresponding IMF3D, IMF4D, or IMF5D bit in the timer status register (TSR) (or the CMF bit in TSR6 or TSR7 for channels 6 and 7) is set to 1, and TCNT is cleared to H'0000 (H'0001 in channels 6 and 7).



If the corresponding TIER bit is set to 1 at this time, an interrupt request is sent to the CPU. After the compare-match, TCNT starts counting up again from H'0000 (H'0001 in channels 6 and 7).

Figure 11.14 shows the operation when channel 3 is used as a cyclic counter (with a cycle setting of H'0008).



**Figure 11.14 Example of Cyclic Counter Operation**

### 11.3.3 Compare-Match Function

Designating general registers in channels 1 to 5 and 11 (GR1A to GR1H, GR2A to GR2H, GR3A to GR3D, GR4A to GR4D, GR5A to GR5D, GR11A, GR11B) for compare-match operation in the timer I/O control registers (TIOR1 to TIOR5, TIOR11) enables compare-match output to be performed at the corresponding external pins (TIO1A to TIO1H, TIO2A to TIO2H, TIO3A to TIO3D, TIO4A to TIO4D, TIO5A to TIO5D, TIO11A, TIO11B).

A free-running counter (TCNT) starts counting up when 1 is set in the timer status register (TSTR). When the desired number is set beforehand in GR, and the TCNT value matches the GR value, the timer status register (TSR) bit corresponding to GR is set and a waveform is output from the corresponding external pin.

1 output, 0 output, or toggle output can be selected by means of a setting in TIOR. If the appropriate interrupt enable register (TIER) setting is made, an interrupt request will be sent to the CPU when a compare-match occurs.

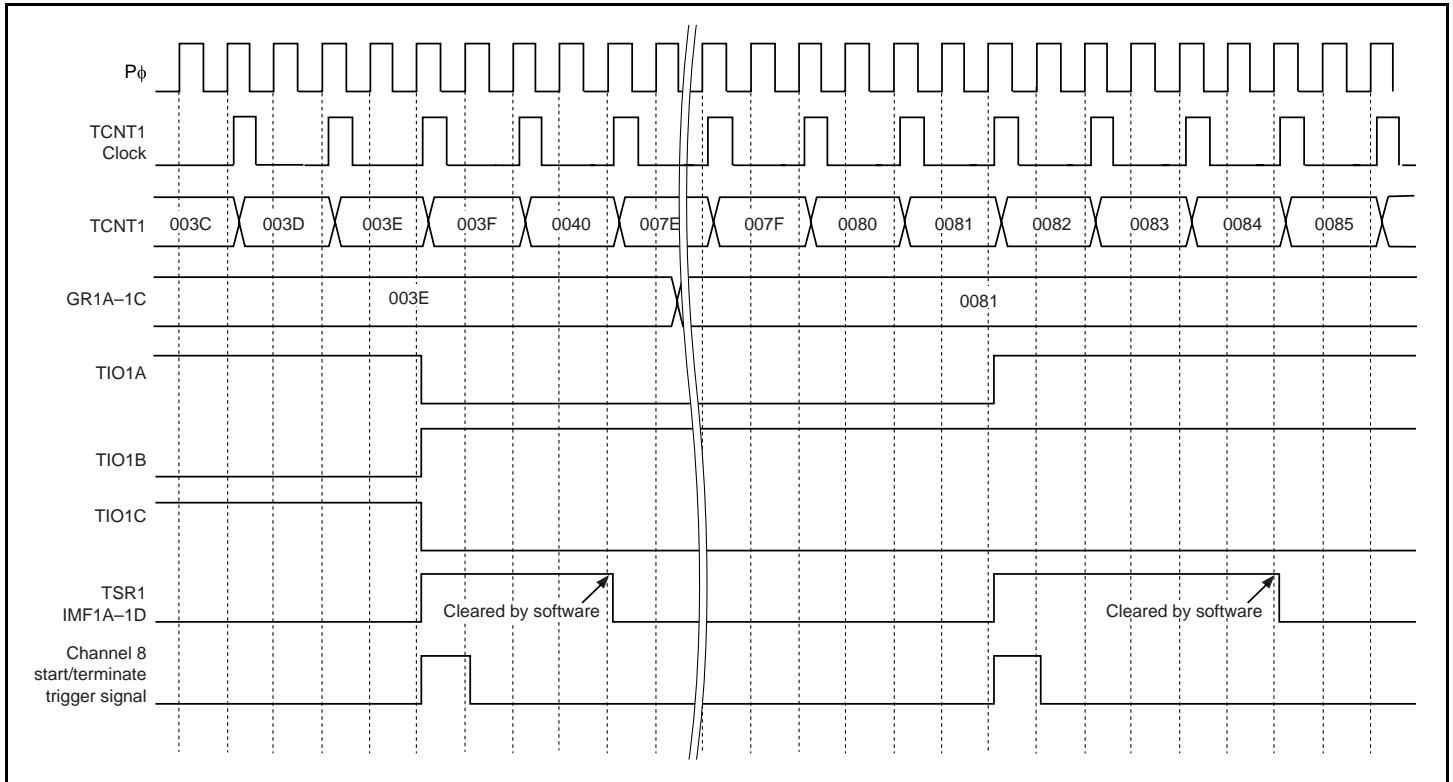
To perform internal interrupts by compare-match or compare-match flag polling processing without performing compare-match output, designate the corresponding compare-match output pin as a general I/O pin and select 1 output, 0 output, or toggle output on compare-match in TIOR.

Channel 1 and 2 compare-match registers (OCR1, OCR2A to OCR2H) perform compare-match operations unconditionally. However, there are no corresponding output pins. If the appropriate TIER setting is made, an interrupt request will be sent to the CPU when a compare-match occurs.

Channel 1 and 2 GR and OCR registers can send a trigger/terminate signal to channel 8 when a compare-match occurs. In this case, settings should be made in the trigger mode register (TRGMDR), timer connection register (TCNR), and one-shot pulse terminate register (OTR).

An example of compare-match operation is shown in figure 11.15.

In the example in figure 11.15, channel 1 is activated, and external output is performed with toggle output specified for GR1A, 1 output for GR1B, and 0 output for GR1C.



**Figure 11.15 Compare-Match Operation**

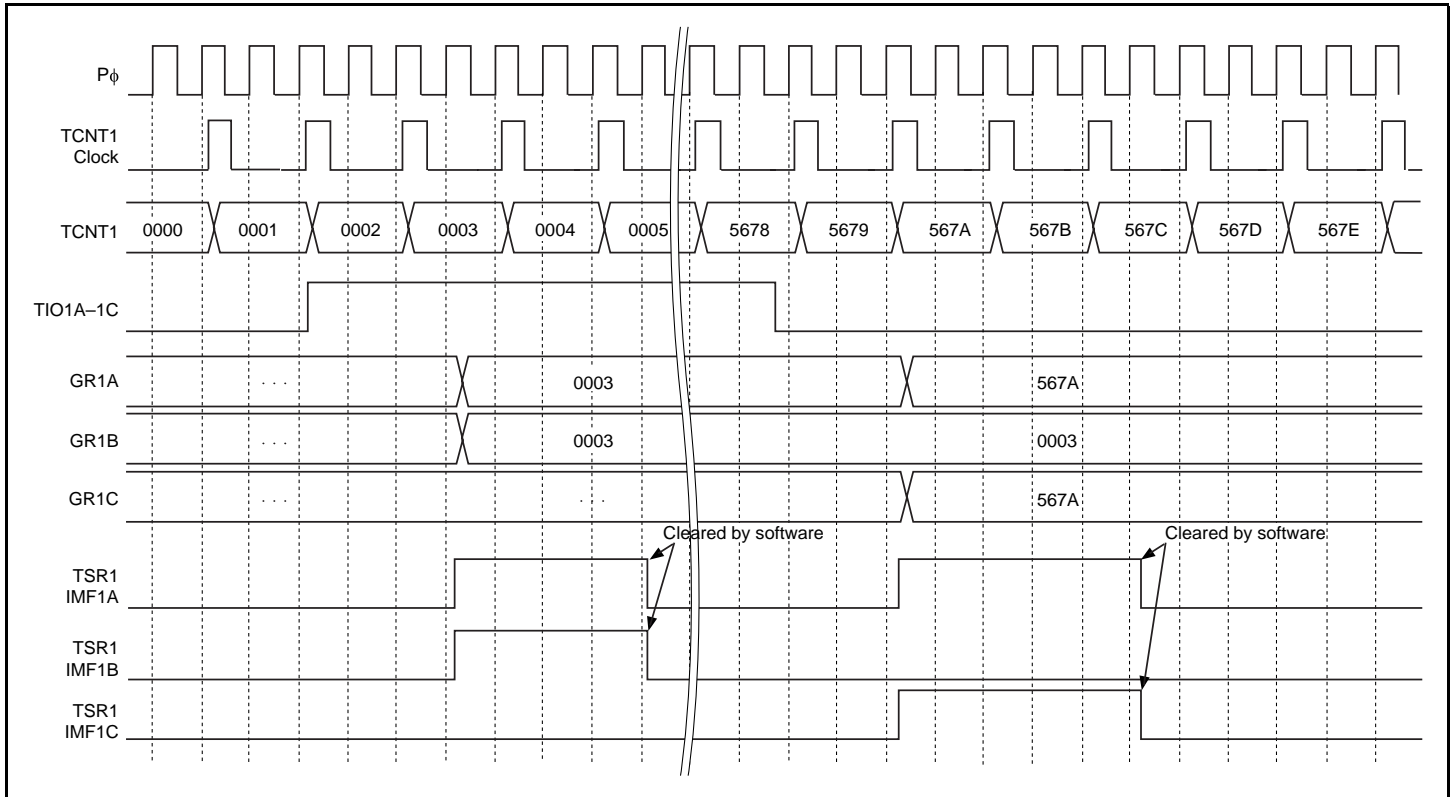
### 11.3.4 Input Capture Function

If input capture registers (ICR0A to ICR0D) and general registers (GR1A to GR1H, GR2A to GR2H, GR3A to GR3D, GR4A to GR4D, GR5A to GR5D, GR11A, GR11B) in channels 1 to 5 and 11 are designated for input capture operation in the timer I/O control registers (TIOR0 to TIOR5, TIOR11), input capture is performed when an edge is input at the corresponding external pins (TIO0A to TIO0D, TIO1A to TIO1H, TIO2A to TIO2H, TIO3A to TIO3D, TIO4A to TIO4D, TIO5A to TIO5D).

A free-running counter (TCNT) starts counting up when a setting is made in the timer start register (TSTR). When an edge is input at an external pin corresponding to ICR or GR, the corresponding timer status register (TSR) bit is set and the TCNT value is transferred to ICR or GR. Rising-edge, falling-edge, or both-edge detection can be selected. By making the appropriate setting in the interrupt enable register (TIER), an interrupt request can be sent to the CPU.

An example of input capture operation is shown in figure 11.16.

In the example in figure 11.16, channel 1 is activated, and input capture operation is performed with both-edge detection specified for TIO1A, rising-edge detection for TIO1B, and falling-edge detection for TIO1C.



**Figure 11.16 Input Capture Operation**

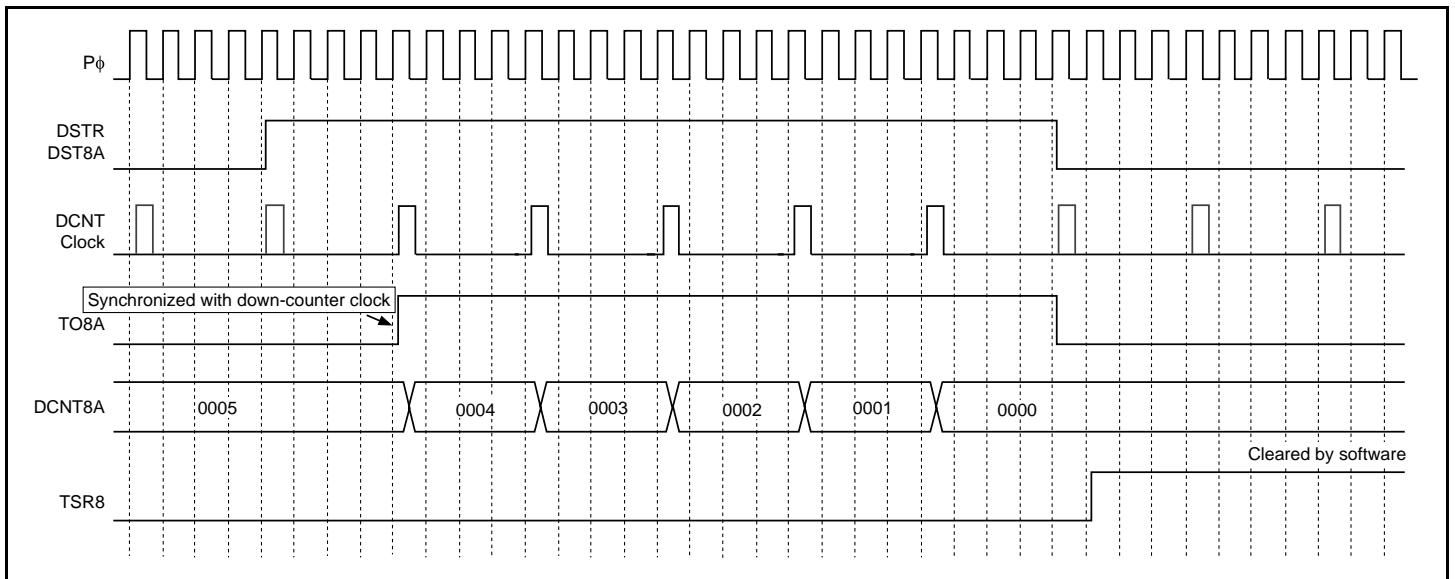
### 11.3.5 One-Shot Pulse Function

Channel 8 has sixteen down-counters (DCNT8A to DCNT8P) and corresponding external pins (TO8A to TO8P) which can be used as one-shot pulse output pins.

When a value is set beforehand in DCNT and the corresponding bit in the down-counter start register (DSTR) is set, DCNT starts counting down, and at the same time 1 is output from the corresponding external pin. When DCNT reaches H'0000 the down-count stops, the corresponding bit in the timer status register (TSR) is set, and 0 is output from the external pin. The corresponding bit in DSTR is cleared automatically. By making the appropriate setting in the interrupt enable register (TIER), an interrupt request can be sent to the CPU.

An example of one-shot pulse operation is shown in figure 11.17.

In the example in figure 11.17, H'0005 is set in DCNT and a down-count is started.



**Figure 11.17 One-Shot Pulse Output Operation**

### 11.3.6 Offset One-Shot Pulse Function and Output Cutoff Function

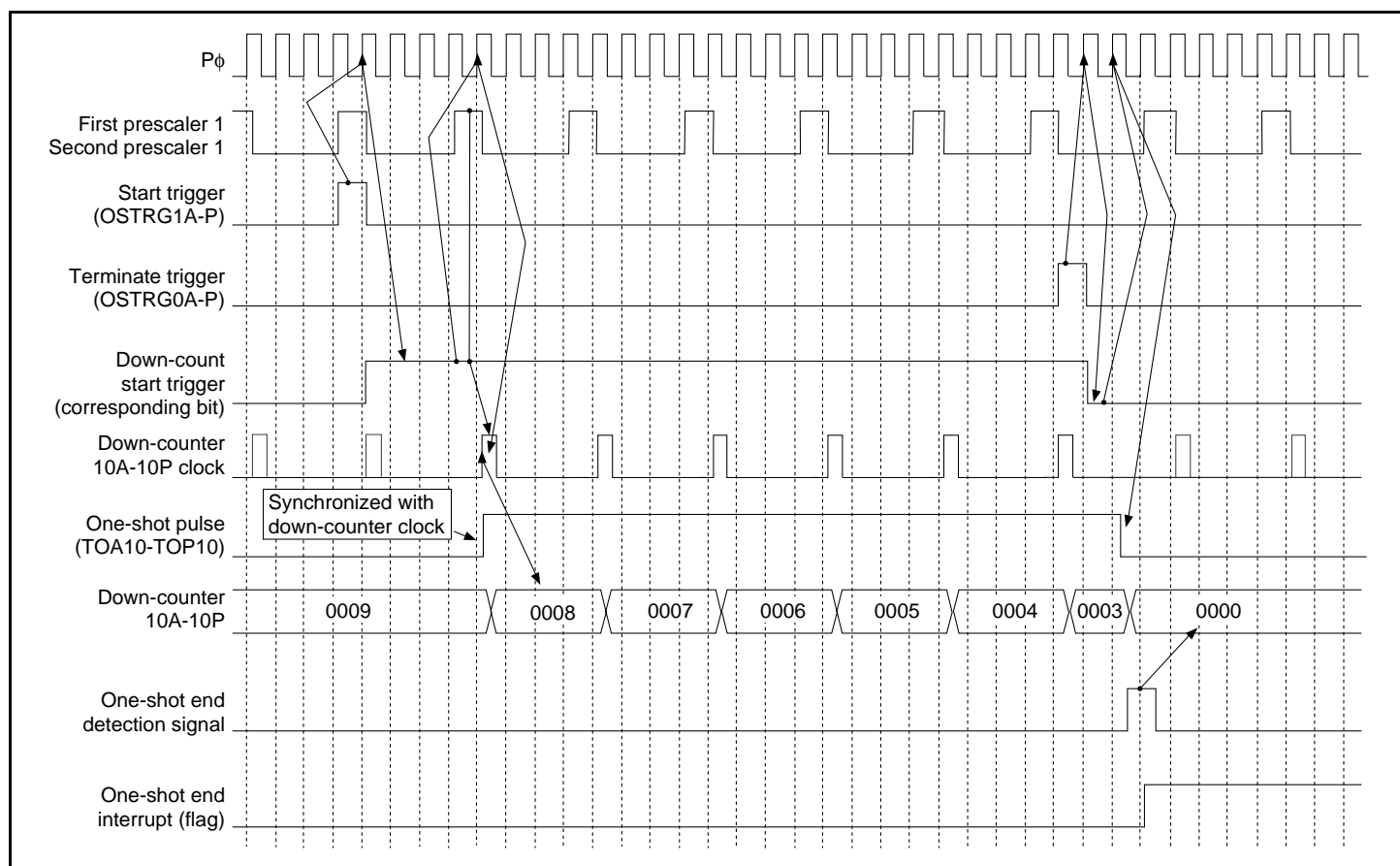
By making an appropriate setting in the timer connection register (TCNR), down-counting by channel 8 down-counters (DCNT8A to DCNT8P) can be started using compare-match signals from channel 1 general registers (GR1A to GR1H) or channel 1 and 2 compare-match registers (OCR1, OCR2A to OCR2H). DCNT8A to DCNT8H are connected to channel 1 OCR1 or GR1A to GR1H, and DCNT8I to DCNT8P are connected to channel 2 OCR2A to OCR2H or GR2A to GR2H. This enables one-shot pulse output from the external pin (TO8A to TO8P) corresponding to DCNT. The down-count can be forcibly stopped by making a setting in the one-shot pulse terminate register (OTR). On channel 1, down-count start or termination by a GR or OCR compare-match can be selected with the trigger mode register (TRGMDR).

Making a setting in the timer start register (TSTR) starts an up-count by a free-running counter (TCNT) in channel 1 or 2. When TCNT matches GR or OCR while connection is enabled by TCNR, the corresponding DSTR is automatically set and DCNT starts counting down. At the same time, 1 is output from the corresponding external pin (TO8A to TO8P). By making the appropriate setting in the interrupt enable register (TIER), an interrupt request can be sent to the CPU.

When TCNT1 matches GR or OCR, or TCNT2 matches GR, while channel 8 one-shot pulse termination by a channel 1 or 2 compare-match signal is enabled by OTR, the corresponding DSTR is automatically cleared and DCNT stops counting down. DCNT is cleared to H'0000 at this time, and must be rewritten before the down-count is restarted.

DCNT8I to DCNT8P are connected to the reload register (RLDR8), and when the DSTR corresponding to DCNT8I to DCNT8P is set, the DCNT8I to DCNT8P counter loads RLDR8 before starting the down-count.

An example of the offset one-shot pulse output function and output cutoff function is shown in figure 11.18.



**Figure 11.18 Offset One-Shot Pulse Output Function and Output Cutoff Operation**

### 11.3.7 Interval Timer Operation

The interval interrupt request registers (ITVRR1, ITVRR2A, ITVRR2B) are connected to bits 6 to 9 and 10 to 13 of the channel 0 free-running counter (TCNT0). The ITVRR registers are 8-bit registers; the upper 4 bits (ITVA) are used for A/D converter activation, and the lower 4 bits (ITVE) are used for interrupt requests. ITVRR1 is connected to A/D converter 2 (AD2), ITVRR2A to A/D converter 0 (AD0), and ITVRR2B to A/D converter 1 (AD1).

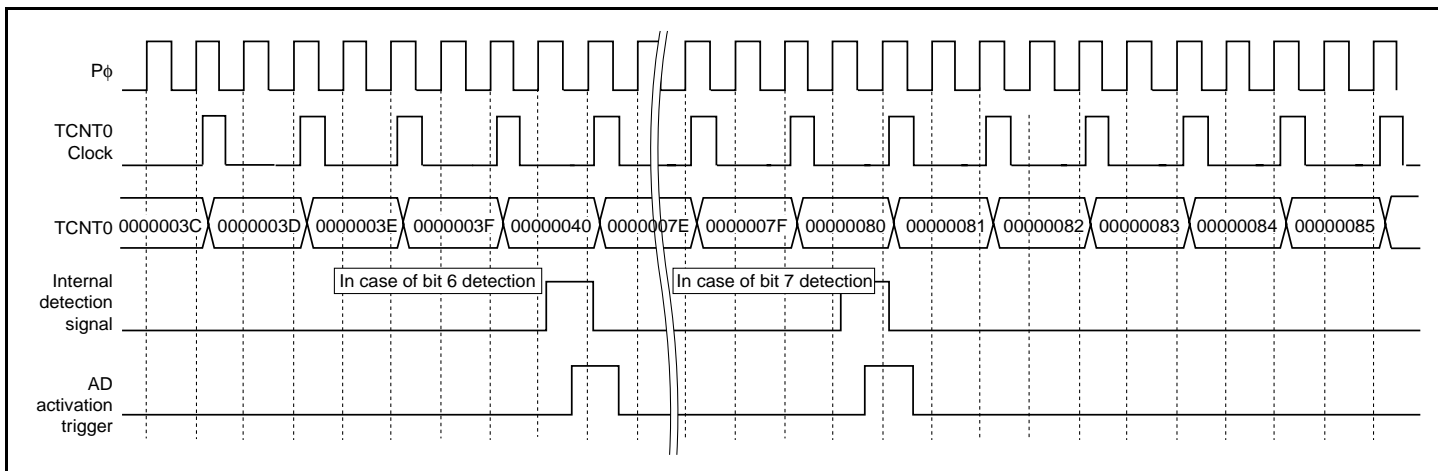
When the ITVA bit for the desired timing is set, the A/D converter is activated when the corresponding bit of TCNT0 changes to 1.

When the ITVE bit for the desired timing is set, an interrupt can be requested when the corresponding bit of TCNT0 changes to 1. At this time, the corresponding bit of the timer status register (TSR0) is set. There are four interrupt sources for the respective ITVRR registers, but there is only one interrupt vector.

To suppress interrupts and A/D converter activation, ITVRR bits should be cleared to 0.

An example of interval timer function operation is shown in figure 11.19.

In the example in figure 11.19, TCNT0 is started by setting ITVE to 1 in ITVRR1.



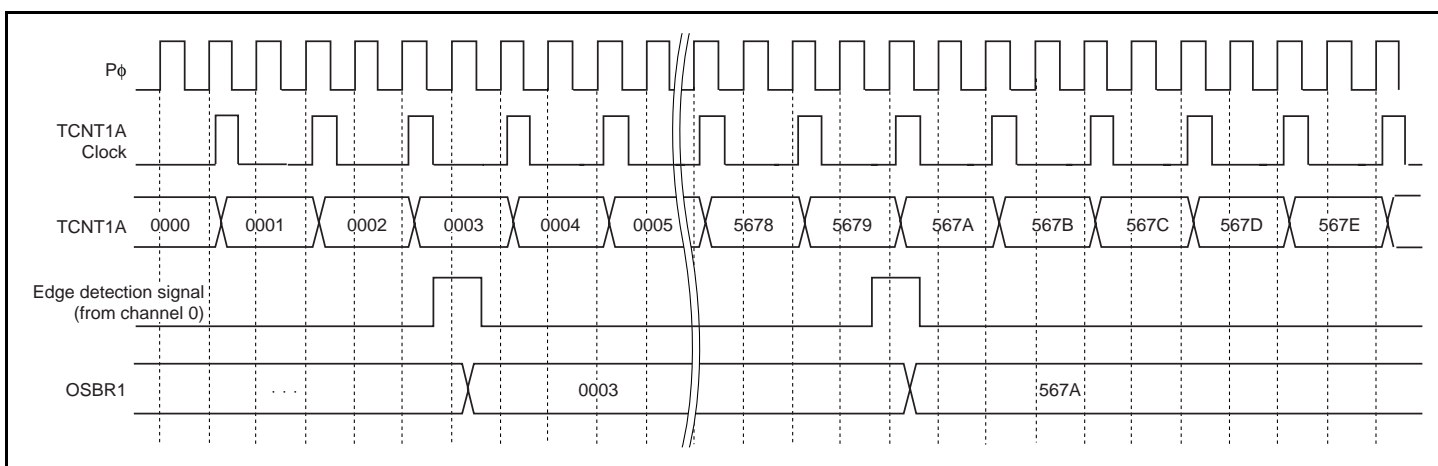
**Figure 11.19 Interval Timer Function**

### 11.3.8 Twin-Capture Function

Channel 0 input capture register ICR0A, channel 1 offset base register 1 (OSBR1), and channel 2 offset base register 2 (OSBR2) can be made to perform input capture in response to the same trigger by means of a setting in timer I/O control register 0 (TIOR0).

When TCNT0, TCNT1A, and TCNT2A in channel 0, channel 1, and channel 2 are started by a setting in the timer start register (TSTR), and an edge of TI0A input (a trigger signal) is detected, the TCNT1A value is transferred to OSBR1, and the TCNT2A value to OSBR2. Edge detection is as described in section 11.3.4, Input Capture Function.

An example of twin-capture operation is shown in figure 11.20.



**Figure 11.20 Twin-Capture Operation**

### 11.3.9 PWM Timer Function

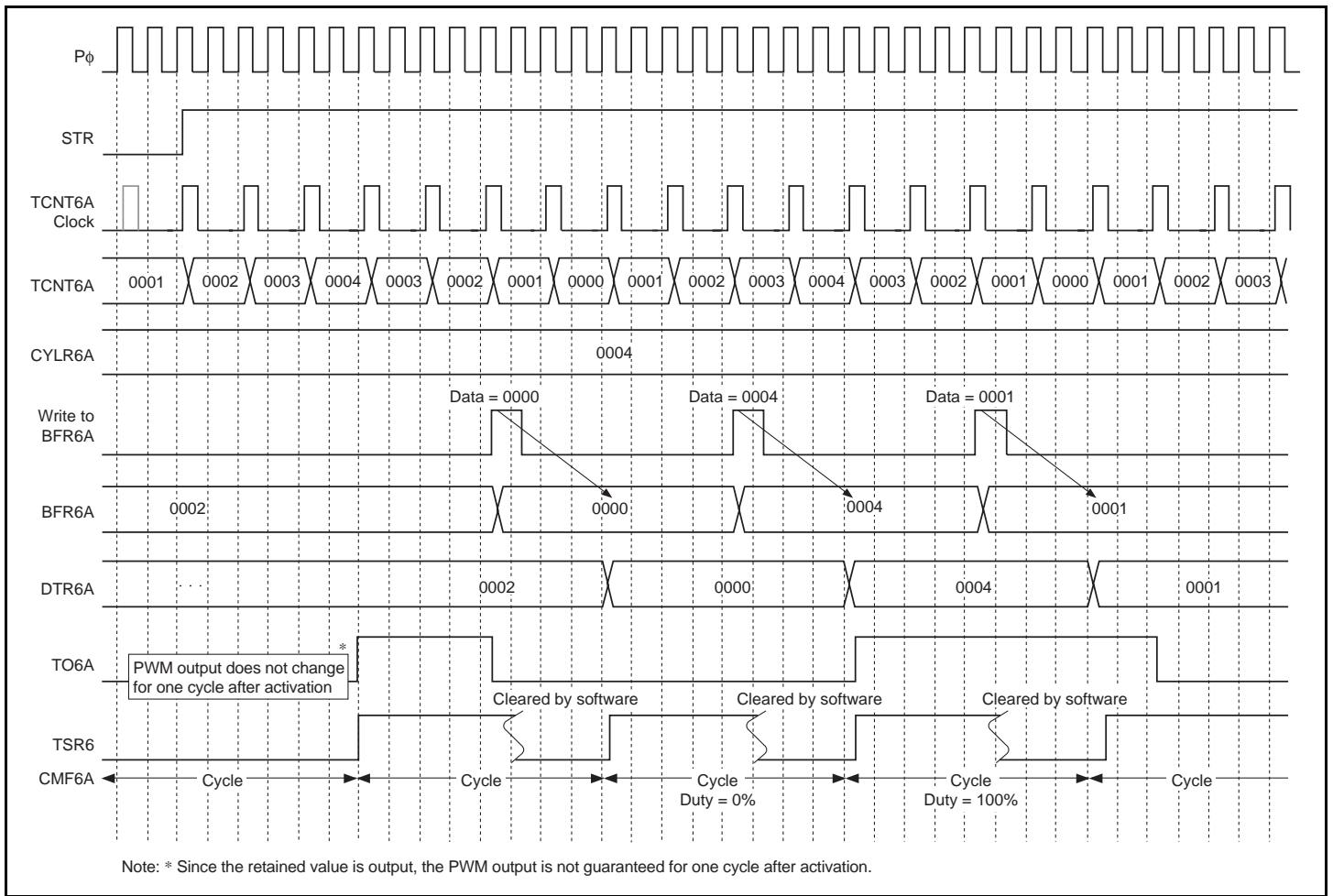
Channels 6 and 7 can be used unconditionally as PWM timers using external pins (TO6A to TO6D, TO7A to TO7D).

In channels 6 and 7, when the corresponding bit is set in the timer start register (TSTR) and the free-running counter (TCNT) is started, the counter counts up until its value matches the corresponding cycle register (CYLR). When TCNT matches CYLR, it is cleared to H'0001 and starts counting up again from that value. At this time, 1 is output from the corresponding external pin. An interrupt request can be sent to the CPU by setting the corresponding bit in the timer interrupt enable register (TIER). If a value has been set in the duty register (DTR), when TCNT matches DTR, 0 is output to the corresponding external pin. If the DTR value is H'0000, the output does not change (0% duty). To set H'0000 to DTR, not write H'0000 directly to DTR but set H'0000 to BFR and then transfer the value to DTR. Writing H'0000 directly to DTR may not give a duty of 0%. A duty of 100% is specified by setting  $DTR = CYLR$ . Do not set a value in DTR that will result in the condition  $DTR > CYLR$ .

Channels 6 and 7 have buffers (BFR); the BFR value is transferred to DTR when TCNT matches CYLR. The duty value written into BFR is reflected in the output value in the cycle following that in which BFR is written to.

An example of PWM timer operation is shown in figure 11.21.

In the example in figure 11.21, H'0004 is set in channel 6 CYLR6A, and H'0002, H'0000 (0%), H'0004 (100%), and H'0001 in BFR6A.



**Figure 11.21 PWM Timer Operation**

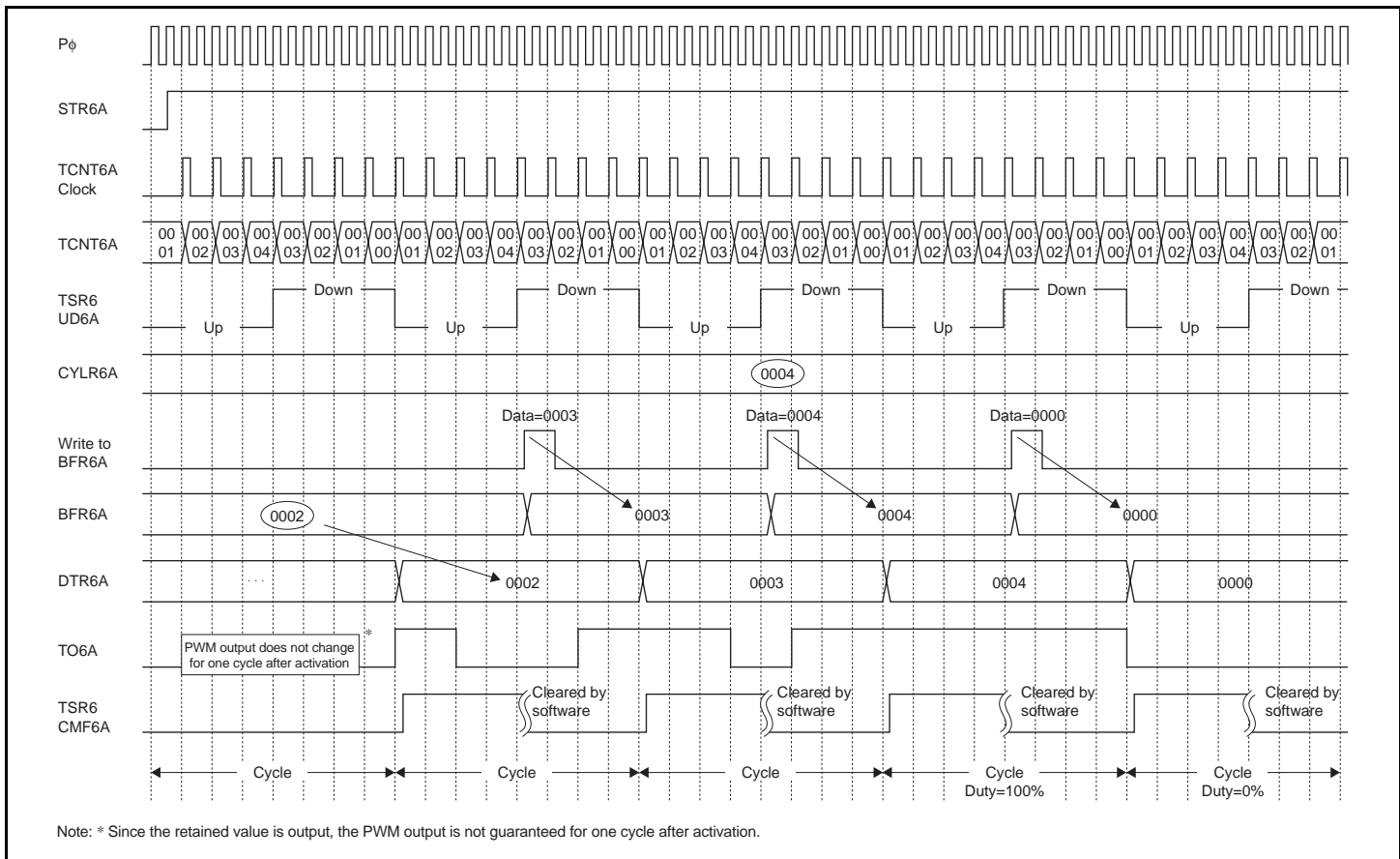
Channel 6 can be used in complementary PWM mode by making a setting in the PWM mode control register (PMDR). On-duty or off-duty can also be selected with a setting in PMDR.

When TCNT6 is started by a setting in TSTR, it starts counting up. When TCNT6 reaches the CYLR6 value, it starts counting down, and on reaching H'000, starts counting up again. The counter status is shown by TSR6. When TCNT6 underflows, an interrupt request can be sent to the CPU by setting the corresponding bit in TIER. When TCNT6 matches the duty register (DTR6) value, the output is inverted. The output prior to the match depends on the PMDR setting. When a value including dead time is set in DTR6, a maximum of 4-phase PWM output is possible. Data transfer from BFR6 to DTR6 is performed when TCNT6 underflows.

An example of channel 6 complementary PWM mode operation is shown in figure 11.22.

In the example in figure 11.22, H'0004 is set in channel 6 CYLR6A, and H'0002, H'0003, H'0004 (100%), and H'0000 (0%) in BFR6A.





**Figure 11.22 Complementary PWM Mode Operation**

### 11.3.10 Channel 3 to 5 PWM Function

PWM mode is selected for channels 3 to 5 by setting the corresponding bits to 1 in the timer mode register (TMDR), enabling the channels to operate as PWM timers with the same cycle.

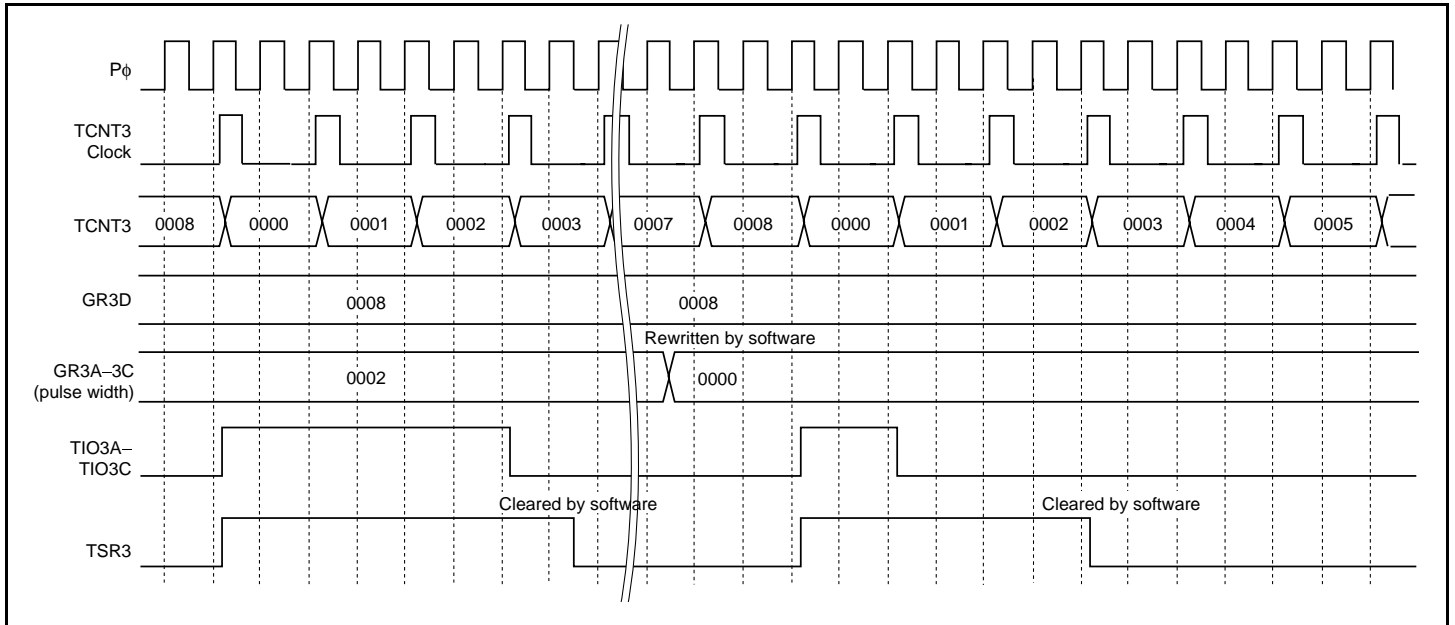
In PWM mode, general registers D (GR3D, GR4D, GR5D) are used as cycle registers, and general registers A to C (GR3A to GR3C, GR4A to GR4C, GR5A to GR5C) as duty registers. The external pins (TIO3A to TIO3C, TIO4A to TIO4C, TIO5A to TIO5C) corresponding to the GRs used as duty registers are used as PWM outputs. External pins TIO3D, TIO4D, and TIO5D should not be used as timer outputs.

The free-running counter (TCNT) is started by making a setting in the timer start register (TSTR), and when TCNT reaches the cycle register (GR3D, GR4D, GR5D) value, a compare-match is generated and TCNT starts counting up again from H'0000. At the same time, the corresponding bit is set in the timer status register (TSR) and 1 is output from the corresponding external pin. When TCNT reaches the duty register (GR3A to GR3C, GR4A to GR4C, GR5A to GR5C) value, 0 is output to the external pin. The corresponding status flag is not set. When PWM operation is performed by starting the free-running counter from its initial value of H'0000, PWM output is not performed for one cycle. To perform immediate PWM output, the value in the cycle register must be set in the free-running counter before the counter is started. If PWM operation is performed after setting H'FFFF in the cycle register, the cycle register's compare-match flag and overflow flag will be set simultaneously.

Note that 0% or 100% duty output is not possible in channel 3 to 5 PWM mode.

An example of channel 3 to 5 PWM mode operation is shown in figure 11.23.

In the example in figure 11.23, H'0008 is set in GR3D, H'0002 is set in GR3A, GR3B, and GR3C, and channel 3 is activated; then, during operation, H'0000 is set in GR3A, GR3B, and GR3C, and output is performed to external pins TIOA3 to TIOC3. Note that 0% duty output is not possible even though H'0000 is set.



**Figure 11.23 Channel 3 to 5 PWM Mode Operation**

### 11.3.11 Event Count Function and Event Cycle Measurement

Channel 9 has six 8-bit event counters (ECNT9A to ECNT9F) and corresponding general registers (GR9A to GR9F). Each event counter has an external pin (TI9A to TI9F).

Each ECNT9 operates unconditionally as an event counter. When an edge is input from the external pin, ECNT9 is incremented. When ECNT9 matches the value set in GR9, it is cleared, and then counts up when an edge is again input at the external pin. By making the appropriate setting in the interrupt enable register (TIER) beforehand, an interrupt request can be sent to the CPU on compare-match.

For ECNT9A to ECNT9D, a trigger can be transmitted to channel 3 when a compare-match occurs. In channel 3, if the channel 9 trigger input is set in the timer I/O control register (TIOR) and the corresponding bit is set to 1 in the timer start register (TSTR), the TCNT3 value is captured in the corresponding general register (GR3A to GR3D) when an ECNT9A to ECNT9D compare-match occurs. This enables the event cycle to be measured.

An example of event count operation is shown in figure 11.24. In this example, ECNT9A counts up on both-edge, falling-edge, and rising-edge detection, H'10 is set in GR9A, and a compare-match is generated.

An example of event cycle measurement operation is shown in figure 11.25. In this example, GR3A in channel 3 captures TCNT3 in response to a trigger from channel 9.

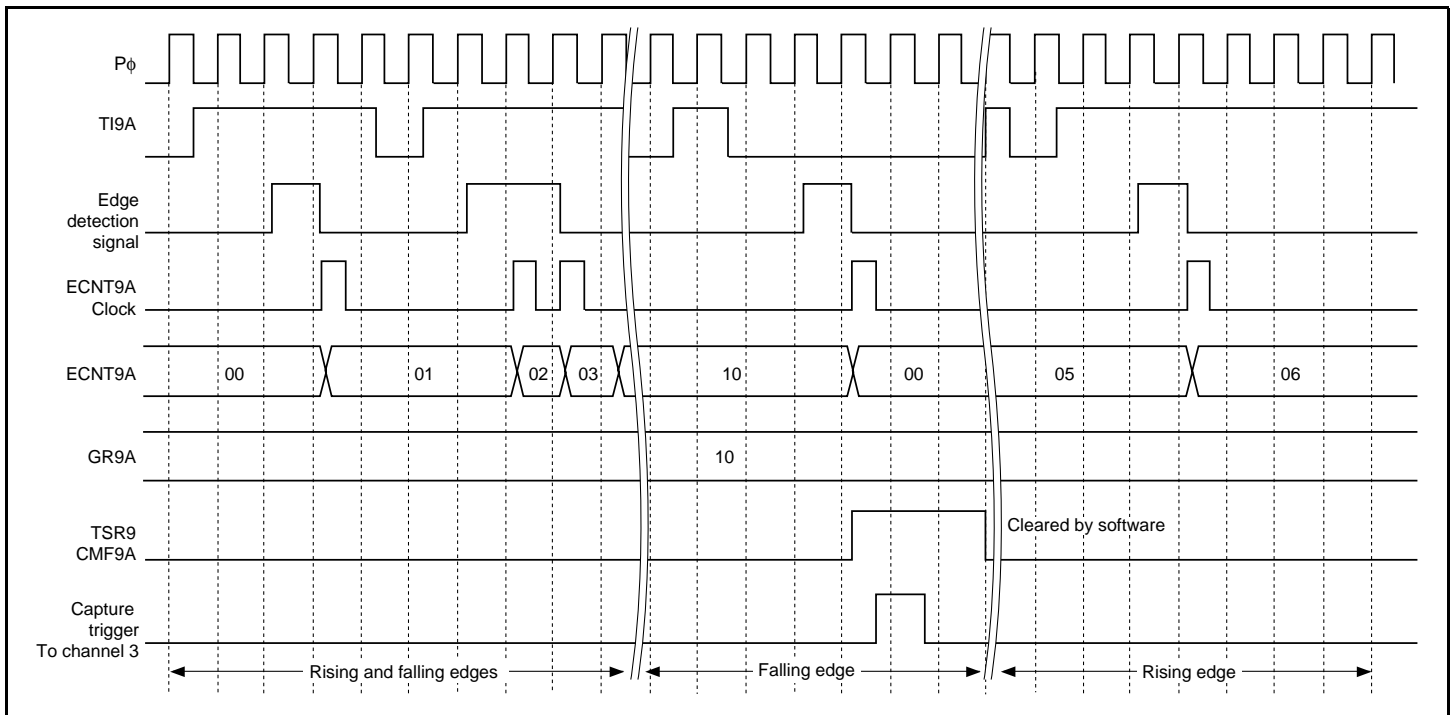
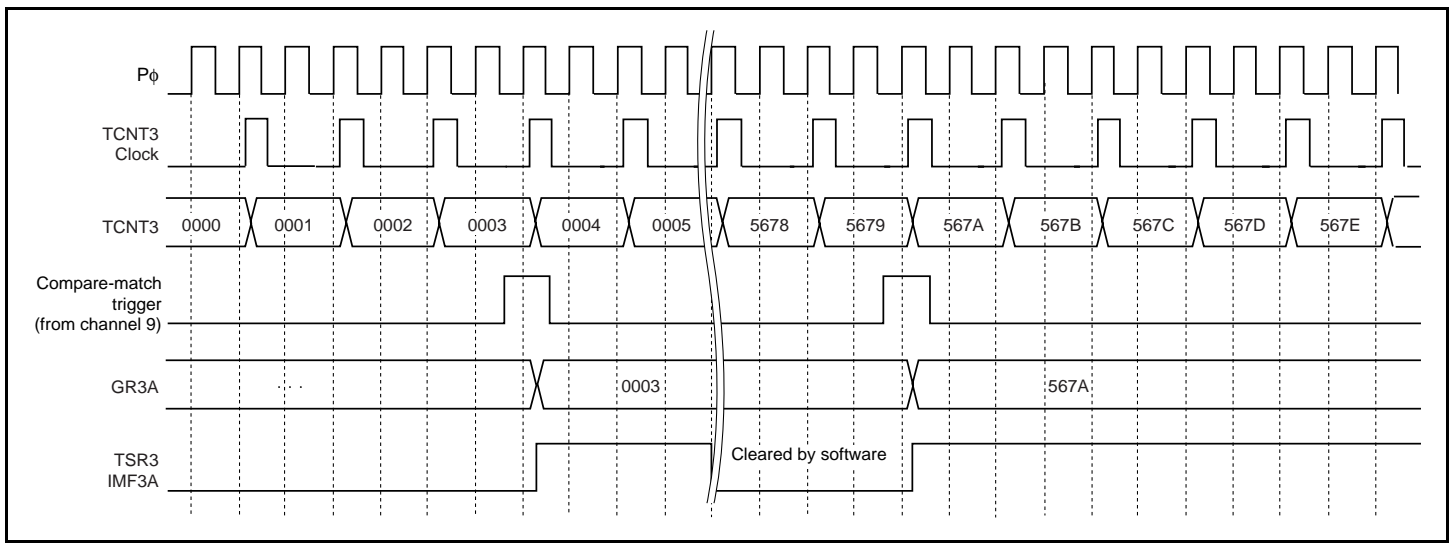


Figure 11.24 Event Count Operation



**Figure 11.25 Event Cycle Measurement Operation**

### 11.3.12 Channel 10 Functions

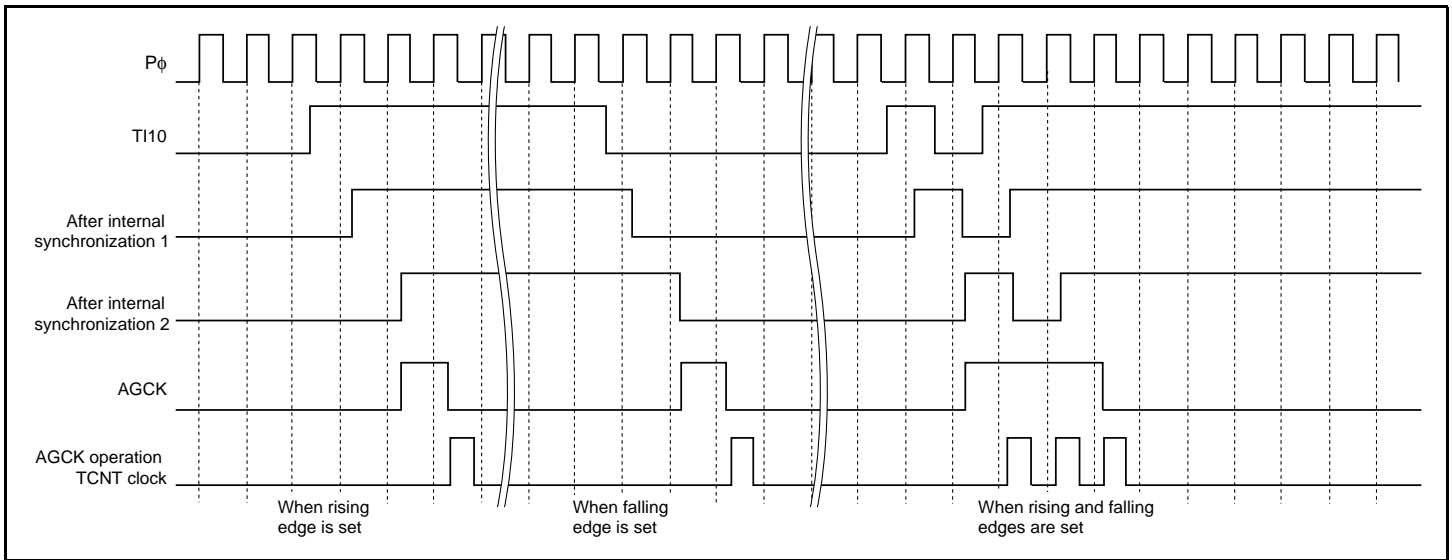
**Inter-Edge Measurement Function and Edge Input Cessation Detection Function:** 32-bit input capture register 10A (ICR10A) and 32-bit output compare register 10A (OCR10A) in channel 10 unconditionally perform input capture and compare-match operations, respectively. These registers are connected to 32-bit free-running counter TCNT10A.

When the corresponding bit is set in the timer start register (TSTR), the entire channel 10 starts operating. ICR10A has an external input pin (TI10), and when an edge is input at this input pin, ICR10A captures the TCNT10A value. At this time, TCNT10A is cleared to H'00000001. The captured value is transferred to the read register (RLD10C) in the multiplied clock generation block. By making the appropriate setting in the interrupt enable register (TIER), an interrupt request can be sent to the CPU. This allows inter-edge measurement to be carried out.

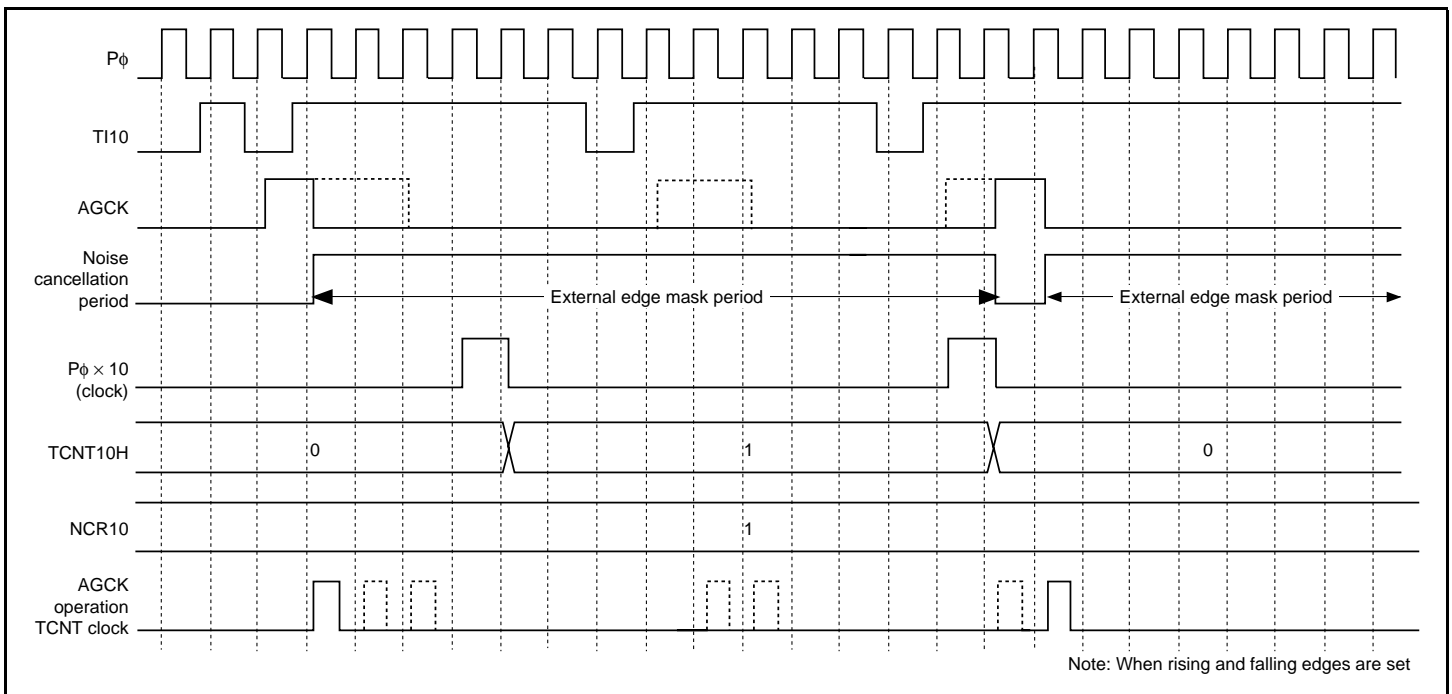
When TCNT10A reaches the value set in OCR10A, a compare-match interrupt can be requested. In this way it is possible to detect the cessation of edge input beyond the time set in OCR10A.

The input edge from TI10 is synchronized internally; the internal signal is AGCK. Noise cancellation is possible for edges input at TI10 using the timer 10H (TCNT10H) input cancellation function by setting the NCE bit in timer control register TCR10. When an edge is input at TI10, TCNT10H starts and input is disabled until it reaches compare-match register NCR10.

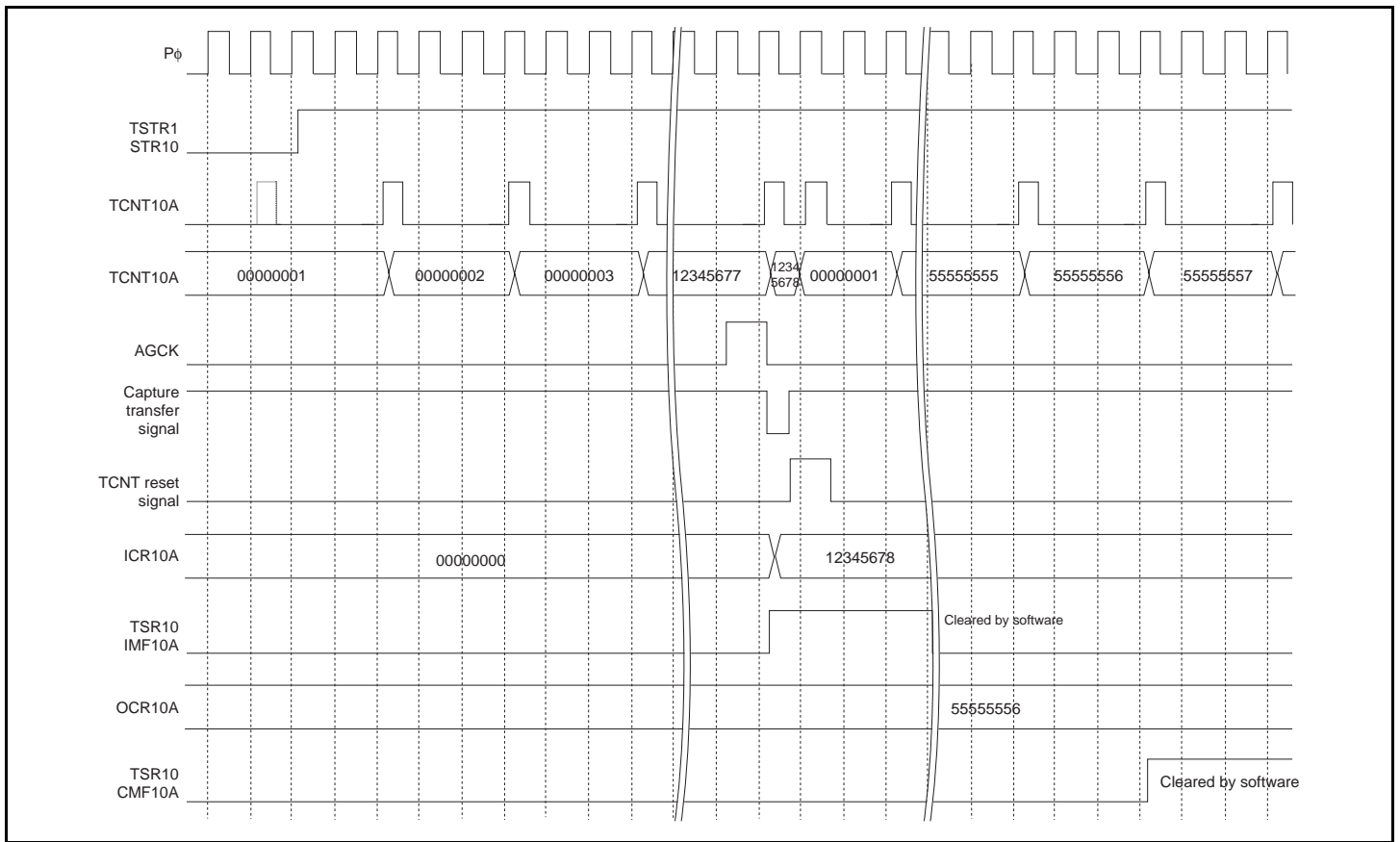
Edge input operation without noise cancellation is shown in figure 11.26, edge input operation with noise cancellation in figure 11.27, and TCNT10A capture operation and compare-match operation in figure 11.28.



**Figure 11.26 Edge Input Operation (Without Noise Cancellation)**



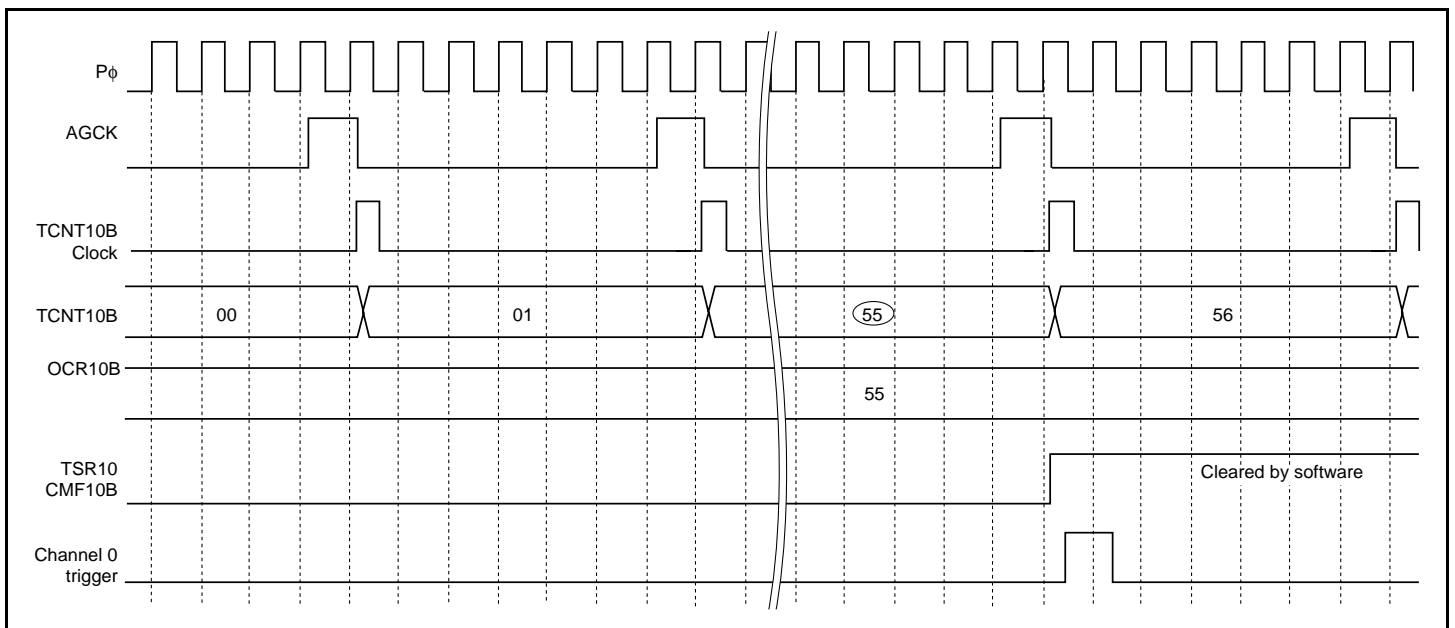
**Figure 11.27 Edge Input Operation (With Noise Cancellation)**



**Figure 11.28 TCNT10A Capture Operation and Compare-Match Operation**

Internally synchronized AGCK is counted by event count 10B (TCNT10B), and when TCNT10B reaches the value set beforehand in compare-match register 10B (OCR10B), a compare-match occurs, and the compare-match trigger signal is transmitted to channel 0. By setting the corresponding bit in TIER, an interrupt request can be sent to the CPU.

Figure 11.29 shows TCNT10B compare-match operation.



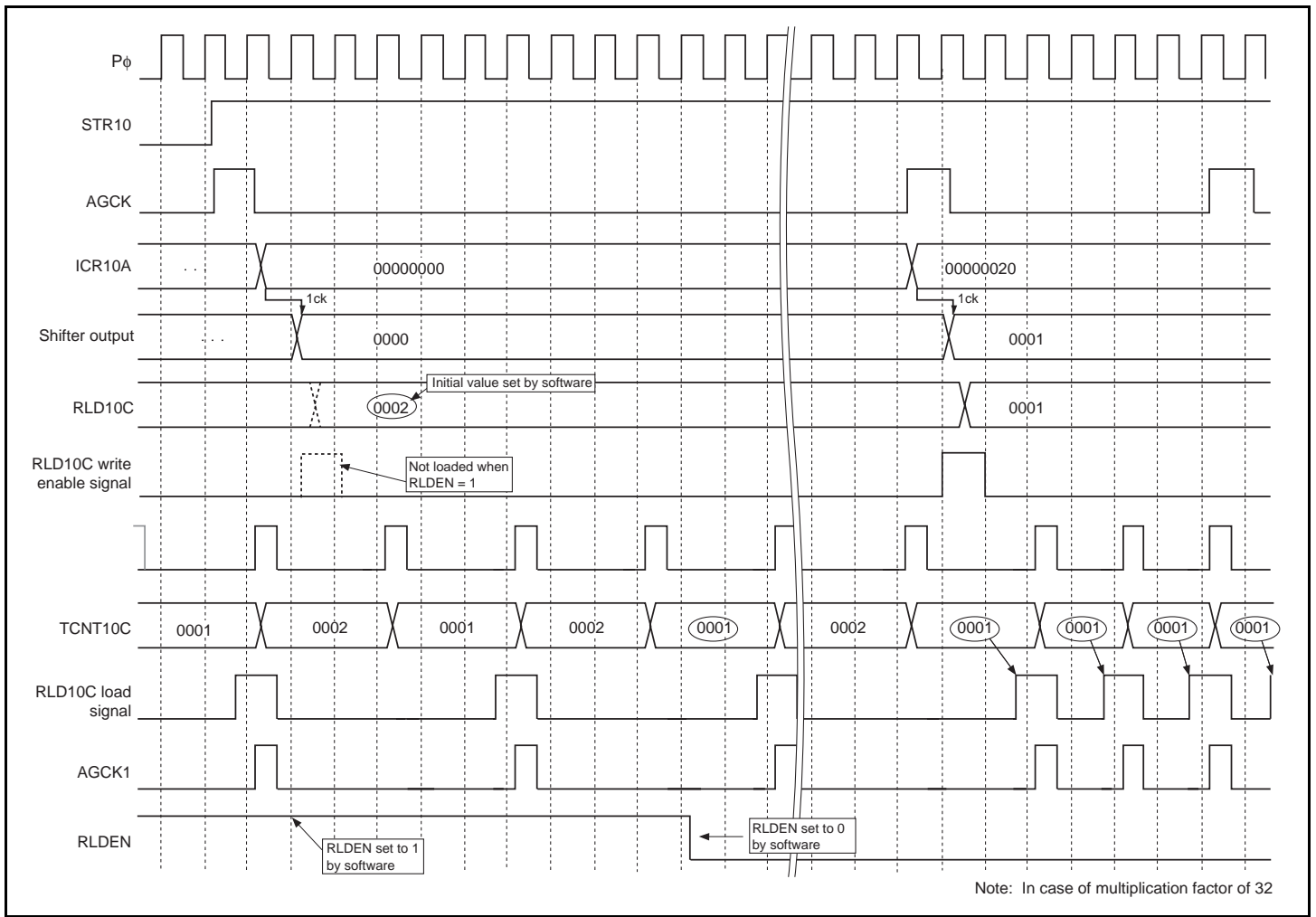
**Figure 11.29 TCNT10B Compare-Match Operation**

**Multiplied Clock Generation Function:** The channel 10 16-bit reload counter (TCNT10C, RLD10C) and 16-bit free-running counter 10G (TCNT10G) can be used to multiply the interval between edges input from external pin TI10 by 32, 64, 128, or 256.

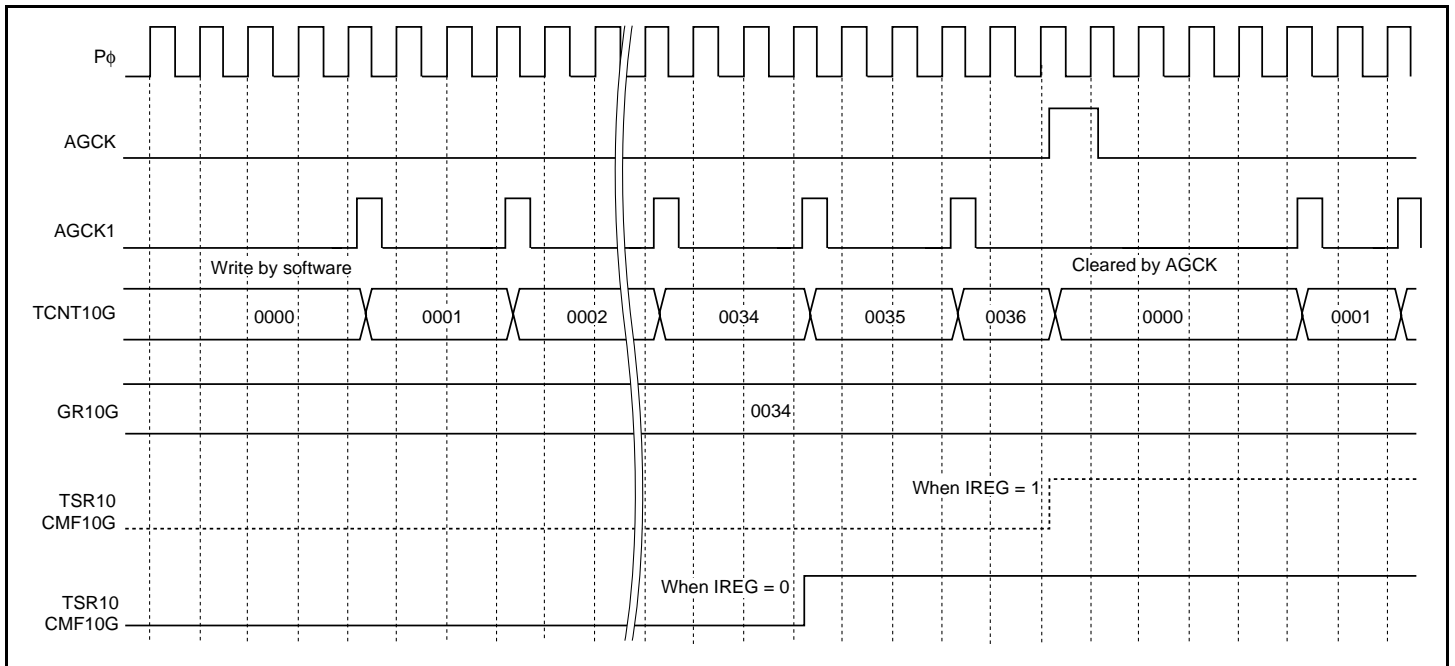
The value captured in ICR10A above is multiplied by 1/32, 1/64, 1/128, or 1/256 according to the value set in the timer I/O control register (TIOR10), and transferred to the reload buffer (RLD10C). At the same time, the same value is transferred to 16-bit reload counter 10C (TCNT10C) and a down-count operation is started. When this counter reaches H'0001, the value is read automatically from RLD10C and the down-count operation is repeated. When this reload occurs, a multiplied clock signal (AGCK1) is generated. AGCK1 is converted to a corrected clock (AGCKM) by the multiplied clock correction function described in the following section.

Channel 10 can also perform compare-match operation by means of the multiplied clock (AGCK1) using general register 10G (GR10G) and 16-bit free-running counter 10G (TCNT10G). TCNT10G is incremented unconditionally by AGCK1. By making the appropriate setting in the interrupt enable register (TIER), an interrupt request can be sent to the CPU when TCNT10G and GR10G match. The timing of this interrupt can be selected with the IREG bit in TIER as either on occurrence of the compare-match or on input of the first TI10 edge after the compare-match.

TCNT10C operation is shown in figure 11.30, and TCNT10G compare-match operation in figure 11.31.



**Figure 11.30 TCNT10C Operation**



**Figure 11.31 TCNT10G Compare-Match Operation**



**Multiplied Clock Correction Function:** Channel 10's three 16-bit correction counters (TCNT10D, TCNT10E, TCNT10F) and correction counter clear register (TCCLR10) have a correction function that makes the interval between edges input from TI10 the frequency multiplication value set in TIOR10.

When AGCK is input, the value in TCNT10D multiplied by the multiplication factor set in TIOR10 is transferred to TCNT10E. At the same time, TCNT10D is incremented.

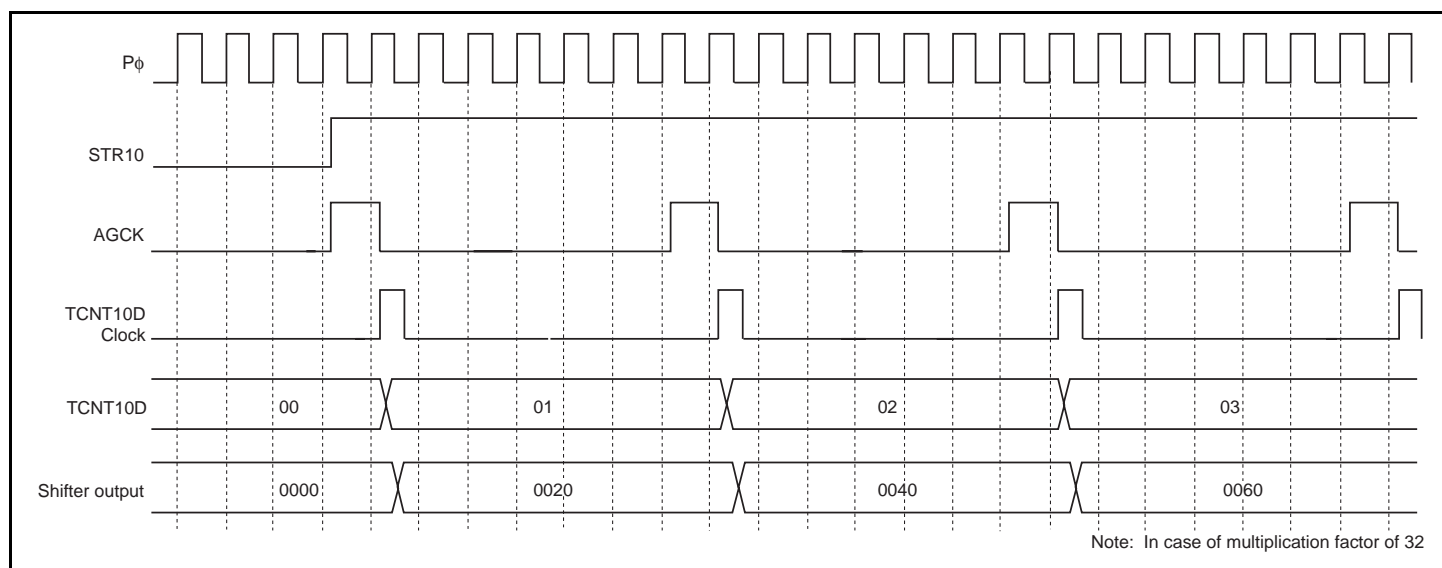
TCNT10E counts up on AGCK1. For example, TCNT10E loads TCNT10D on AGCK, and counts up again on AGCK1. Using the counter correction select bit (CCS) in TIOR10, it is possible to select whether or not TCNT10E is halted when TCNT10D = TCNT10E.

TCNT10F has the peripheral clock (Pφ) as its input and is constantly compared with TCNT10E. When the TCNT10F value is smaller than that in TCNT10E, TCNT10F is incremented and outputs a corrected multiplied clock signal (AGCKM).

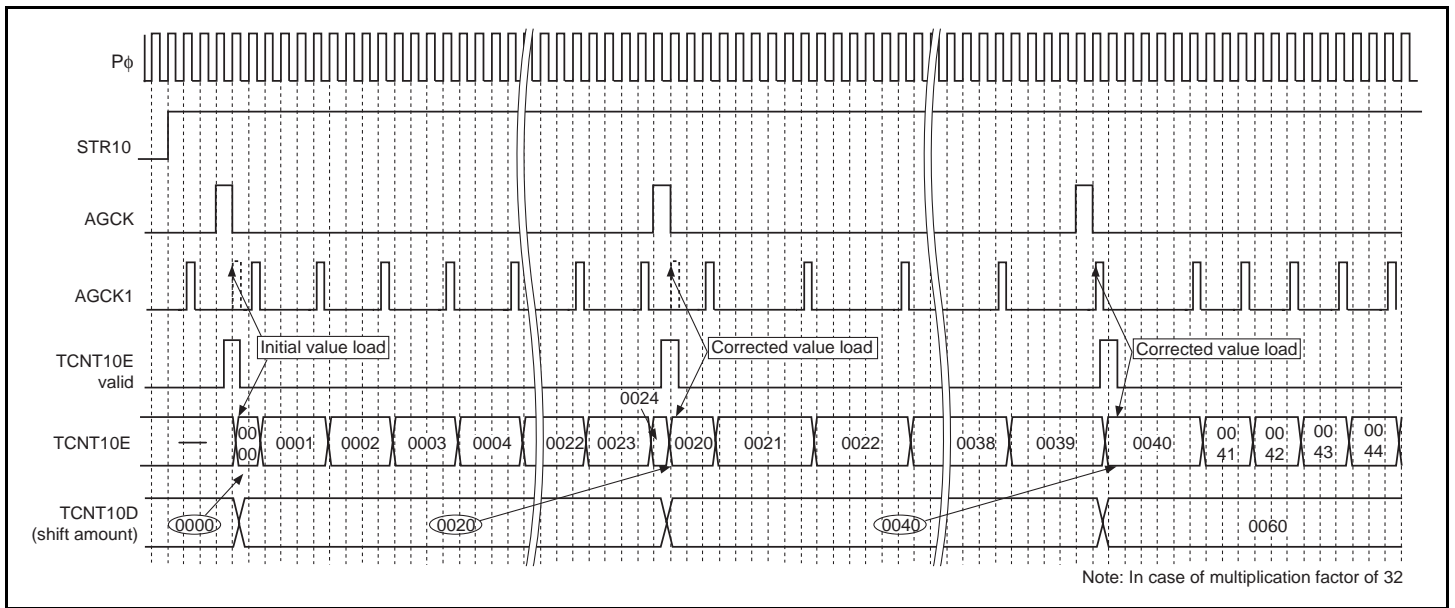
When the TCNT10F value exceeds the TCNT10E value, no count-up operation is performed. AGCKM is output to the channel 1 to 5 free-running counters (TCNT1 to TCNT5).

Channel 10 also has a correction counter clear register (TCCLR10). The correction counters (TCNT10D, TCNT10E, TCNT10F) and channel 1 and 2 free-running counters (TCNT1 and TCNT2) can be cleared when TCNT10F reaches the value set in TCCLR10.

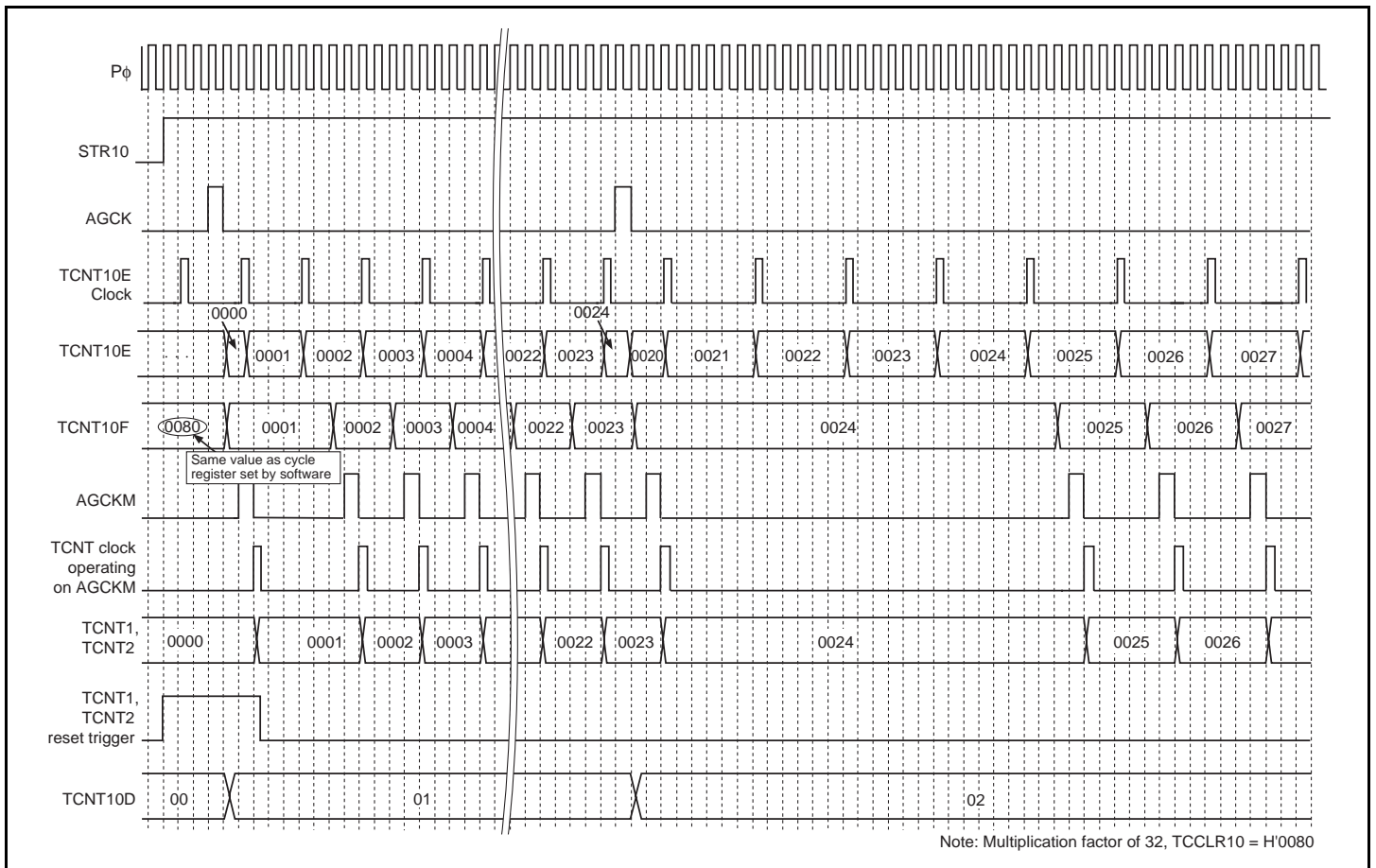
TCNT10D operation is shown in figure 11.32, TCNT10E operation in figure 11.33, TCNT10F operation (at startup) in figure 11.34, TCNT10F operation (end of cycle, acceleration, deceleration) in figure 11.35, and TCNT10F operation (end of cycle, steady-state) in figure 11.36.



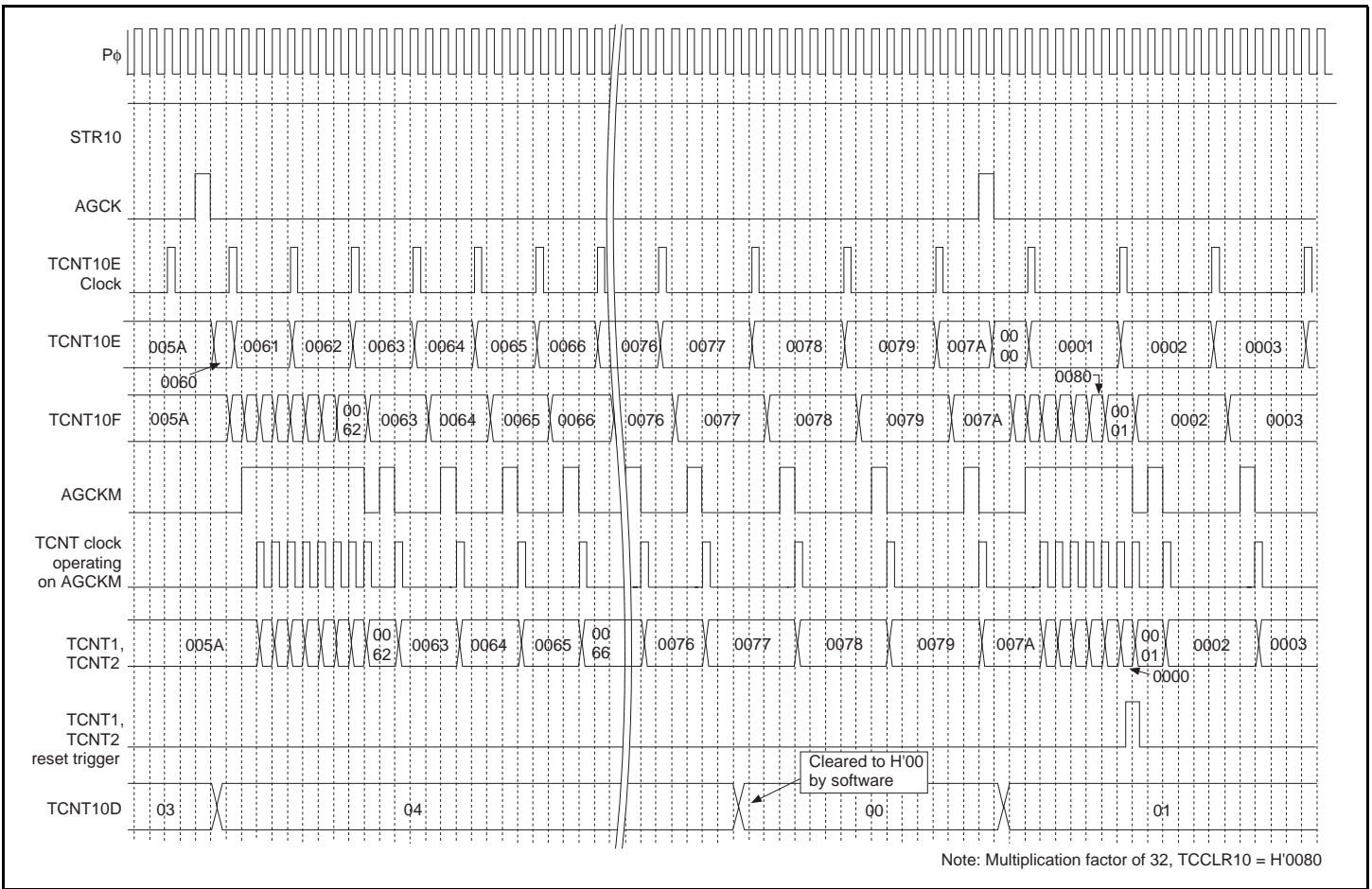
**Figure 11.32 TCNT10D Operation**



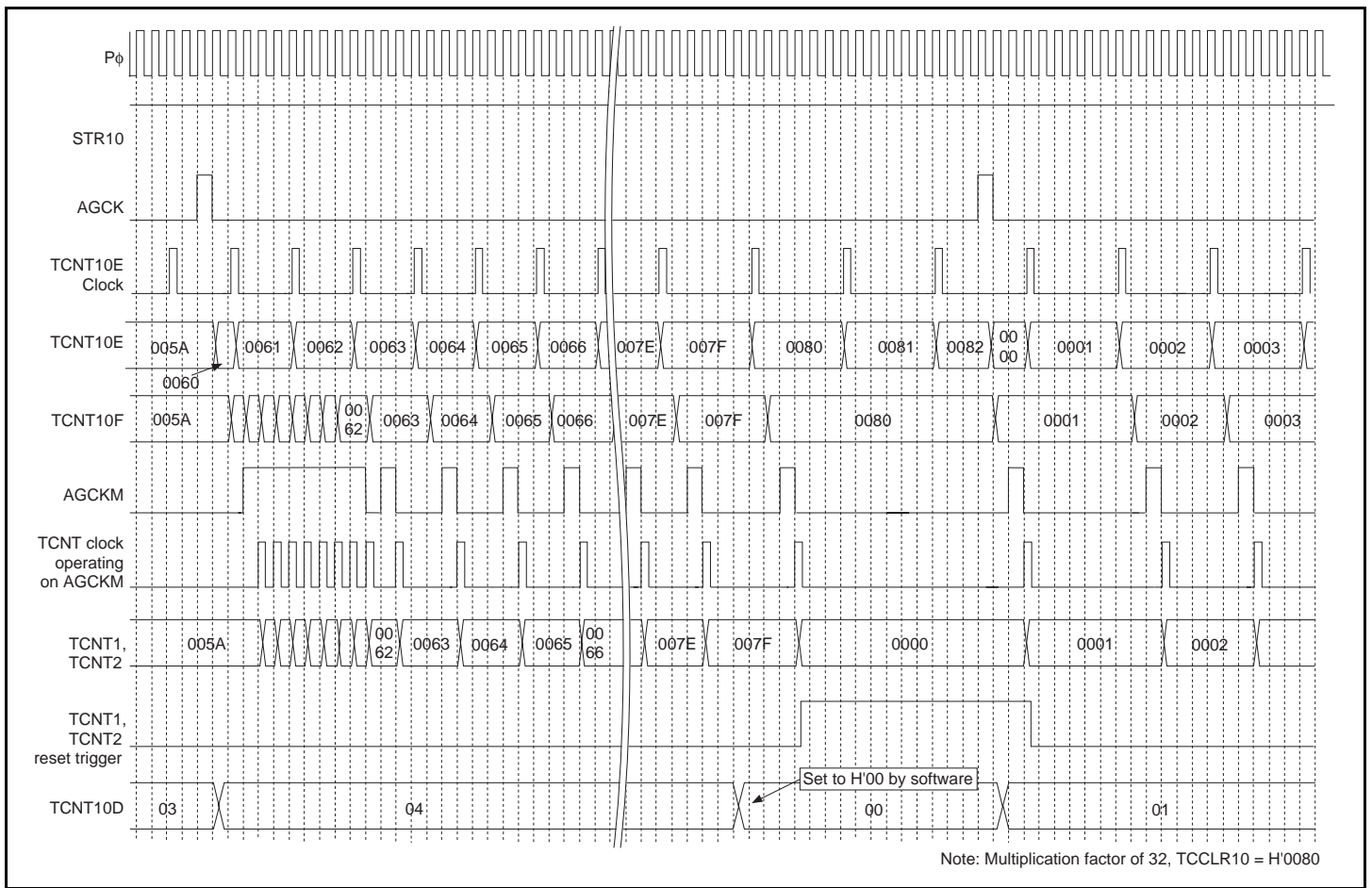
**Figure 11.33 TCNT10E Operation**



**Figure 11.34 TCNT10F Operation (At Startup)**



**Figure 11.35 TCNT10F Operation (End of Cycle, Acceleration, Deceleration)**



**Figure 11.36 TCNT10F Operation (End of Cycle, Steady-State)**

## 11.4 Interrupts

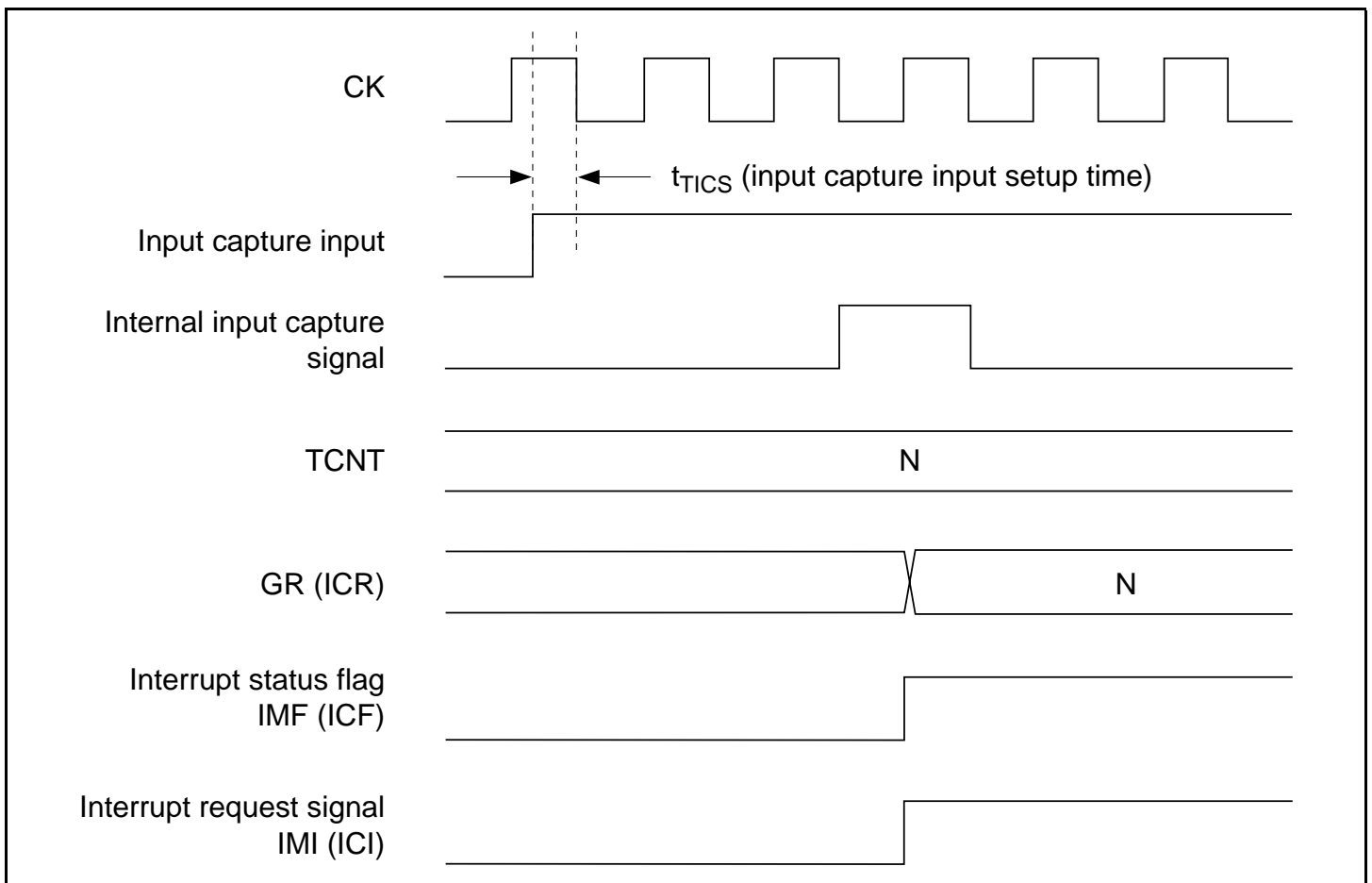
The ATU has 75 interrupt sources of five kinds: input capture interrupts, compare-match interrupts, overflow interrupts, underflow interrupts, and interval interrupts.

### 11.4.1 Status Flag Setting Timing

**IMF (ICF) Setting Timing in Input Capture:** When an input capture signal is generated, the IMF bit and ICF bit are set to 1 in the timer status register (TSR), and the TCNT value is simultaneously transferred to the corresponding GR, ICR, and OSBR.

The timing in this case is shown in figure 11.37.

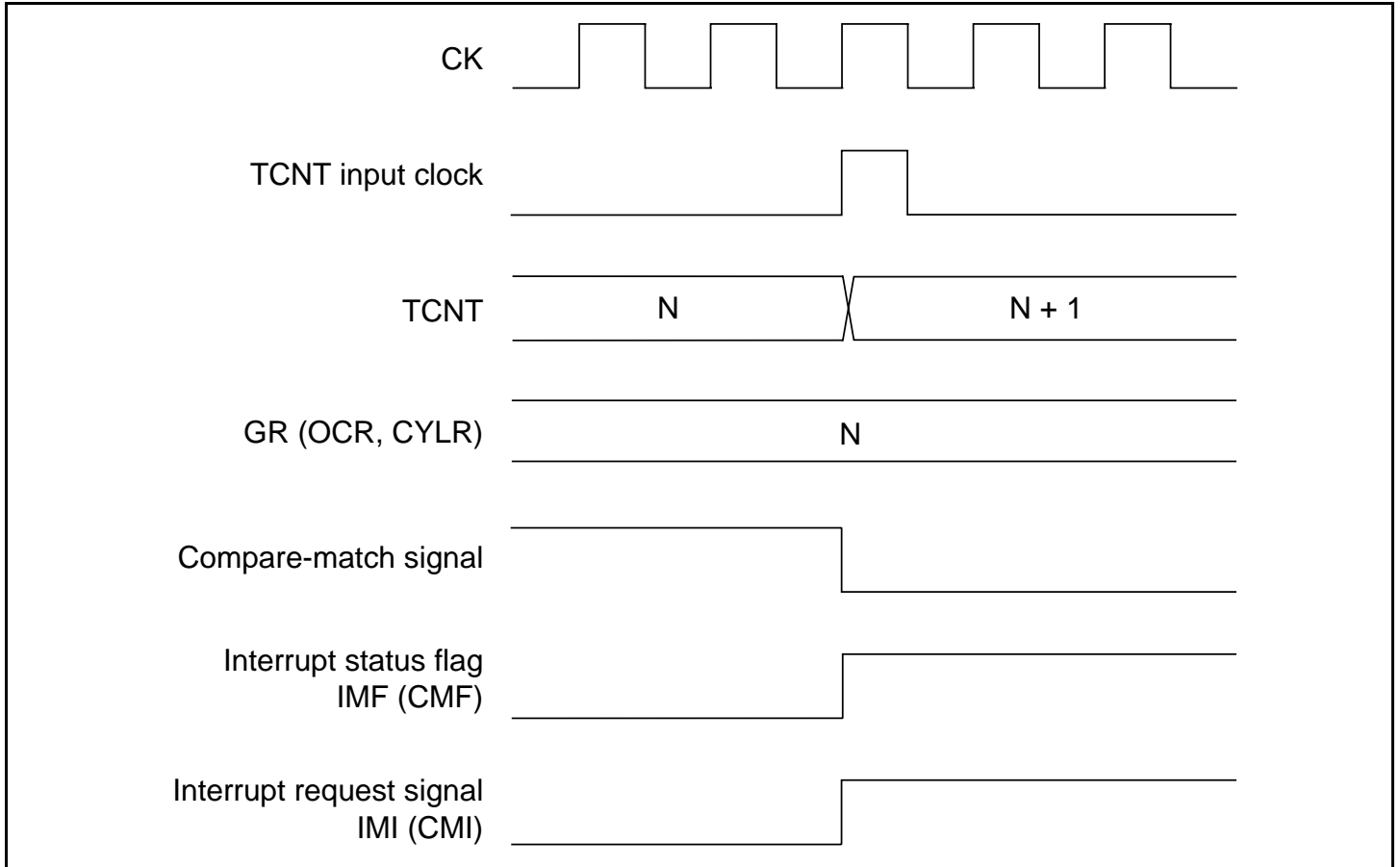
In the example in figure 11.37, a signal is input from an external pin, and input capture is performed on detection of a rising edge.



**Figure 11.37 IMF (ICF) Setting Timing in Input Capture**

**IMF (ICF) Setting Timing in Compare-Match:** The IMF bit and CMF bit are set to 1 in the timer status register (TSR) by the compare-match signal generated when the general register (GR) output compare register (OCR), or cycle register (CYLR) value matches the timer counter (TCNT) value. The compare-match signal is generated in the last state of the match (when the matched TCNT count value is updated).

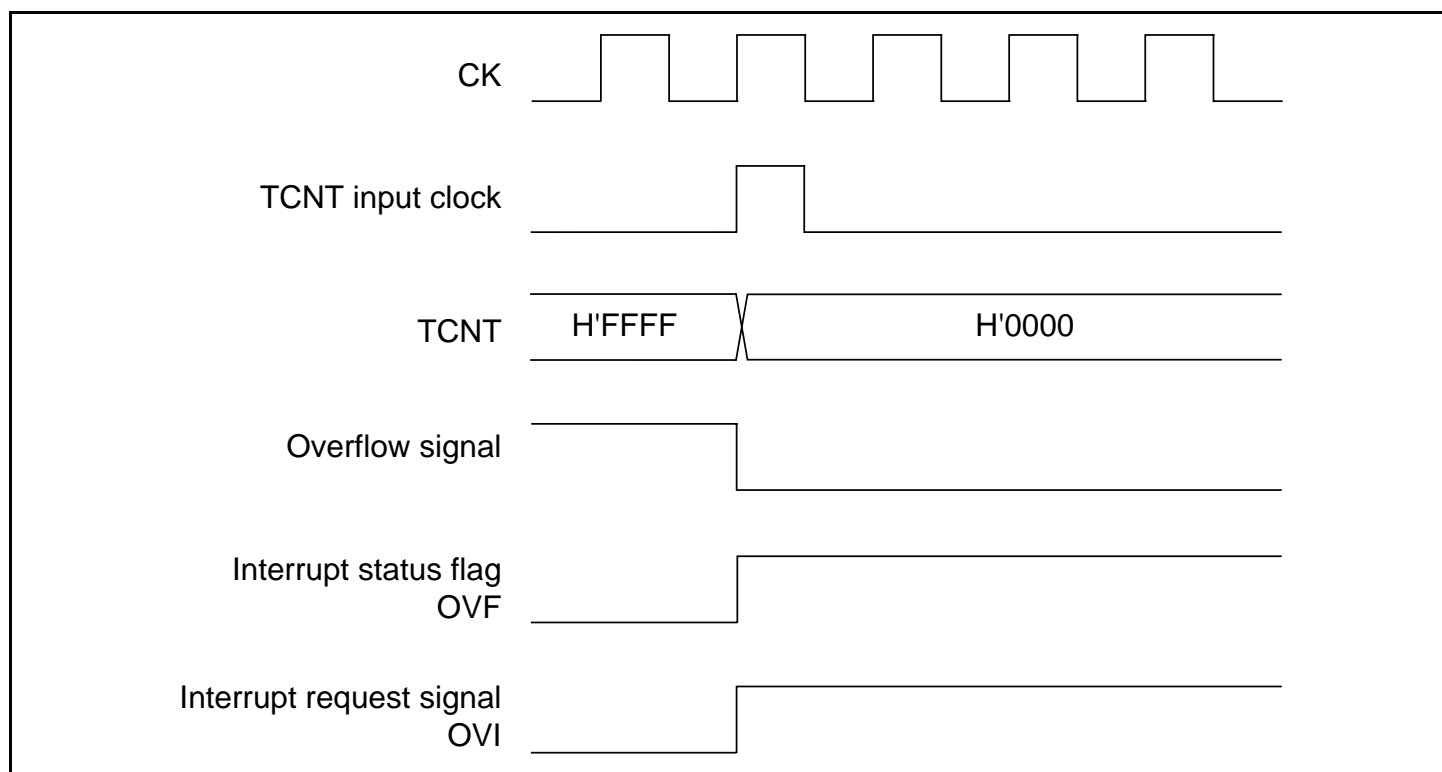
The timing in this case is shown in figure 11.38.



**Figure 11.38 IMF (CMF) Setting Timing in Compare-Match**

**OVF Setting Timing in Overflow:** When TCNT overflows (from H'FFFF to H'0000, or from H'FFFFFFFF to H'00000000), the OVF bit is set to 1 in the timer status register (TSR).

The timing in this case is shown in figure 11.39.

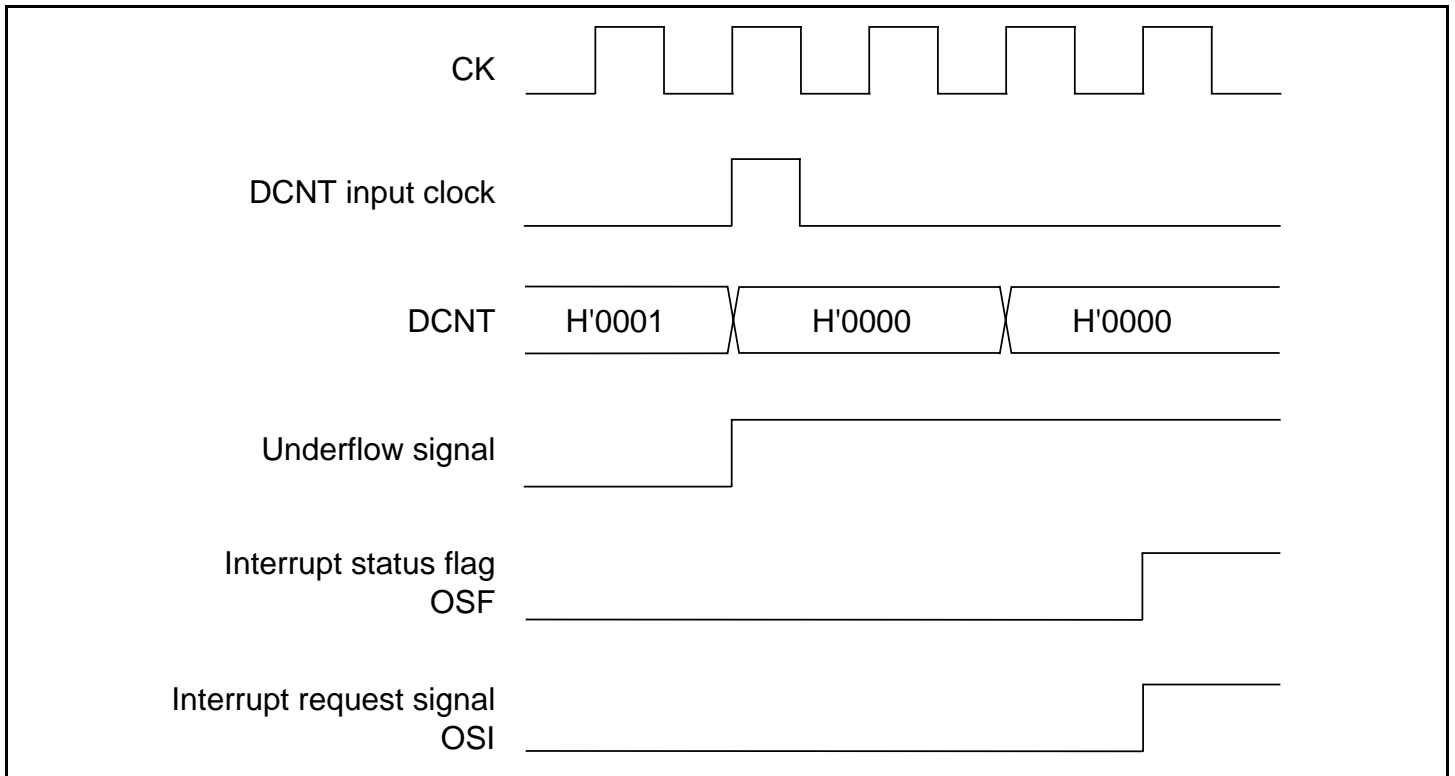


**Figure 11.39 OVF Setting Timing in Overflow**

**OSF Setting Timing in Underflow:** When a down-counter (DCNT) counts down from H'0001 to H'0000 on DCNT input clock input, the OSF bit is set to 1 in the timer status register (TSR) when the next DCNT input clock pulse is input (when underflow occurs). However, when DCNT is H'0000, it remains unchanged at H'0000 no matter how many DCNT input clock pulses are input.

When DCNT is cleared by means of the one-shot pulse function, the OSF bit is cleared when the next DCNT input clock is input.

The timing in this case is shown in figure 11.40.

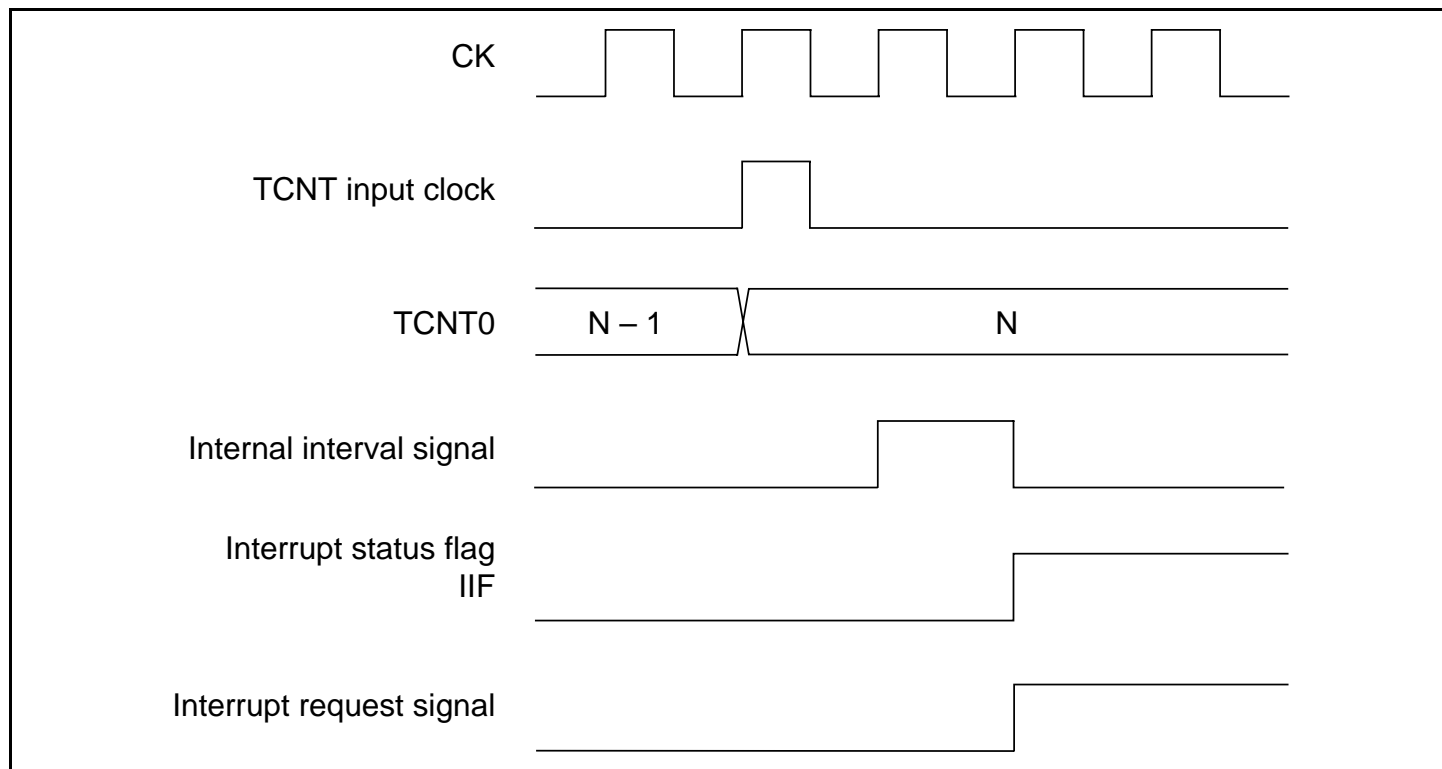


**Figure 11.40 OSF Setting Timing in Underflow**



**Timing of IIF Setting by Interval Timer:** When 1 is generated by ANDing the rise of bit 10–13 in free-running counter TCNT0L with bit ITVE0–ITVE3 in the interval interrupt request register (ITVRR), the IIF bit is set to 1 in the timer status register (TSR).

The timing in this case is shown in figure 11.41. TCNT0 value N in the figure is the counter value when TCNT0L bit 6–13 changes to 1. (For example, N = H'00000400 in the case of bit 10, H'00000800 in the case of bit 11, etc.)

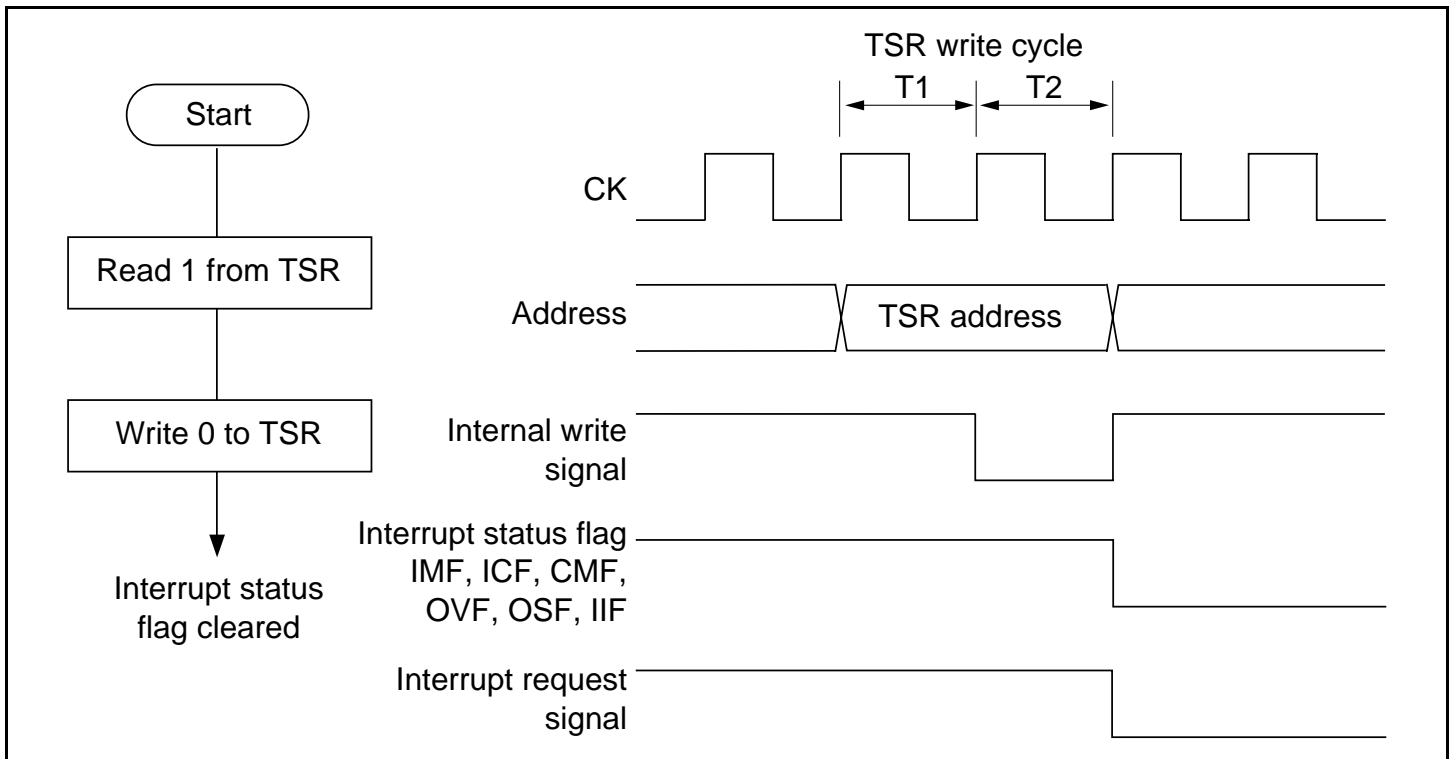


**Figure 11.41 Timing of IIF Setting Timing by Interval Timer**

## 11.4.2 Status Flag Clearing

**Clearing by CPU Program:** The interrupt status flag is cleared when the CPU writes 0 to the flag after reading it while set to 1.

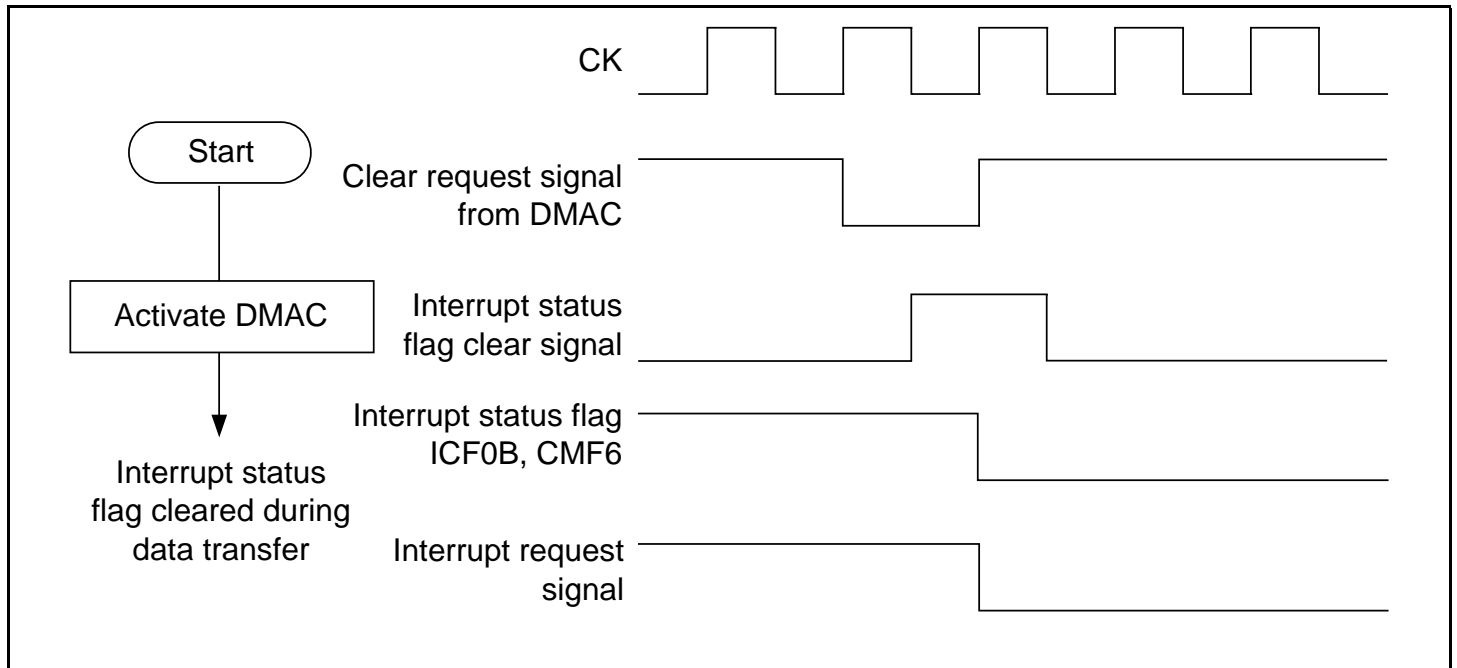
The procedure and timing in this case are shown in figure 11.42.



**Figure 11.42 Procedure and Timing for Clearing by CPU Program**

**Clearing by DMAC:** The interrupt status flag (ICF0A to ICF0D, CMF6A to CMF6D, CMF7A to CMF7D) is cleared automatically during data transfer when the DMAC is activated by input capture or compare-match.

The procedure and timing in this case are shown in figure 11.43.



**Figure 11.43 Procedure and Timing for Clearing by DMAC**

## 11.5 CPU Interface

### 11.5.1 Registers Requiring 32-Bit Access

Free-running counters 0 and 10A (TCNT0, TCNT10A), input capture registers 0A to 0D and 10A (ICR0A to ICR0D, ICR10A), and output compare register 10A (OCR10A) are 32-bit registers. As these registers are connected to the CPU via an internal 16-bit data bus, a read or write (read only, in the case of ICR0A to ICR0D and ICR10A) is automatically divided into two 16-bit accesses.

Figure 11.44 shows a read from TCNT0, and figure 11.45 a write to TCNT0.

When reading TCNT0, in the first read the TCNT0H (upper 16-bit) value is output to the internal data bus, and at the same time, the TCNT0L (lower 16-bit) value is output to an internal buffer register. Then, in the second read, the TCNT0L (lower 16-bit) value held in the internal buffer register is output to the internal data bus.

When writing to TCNT0, in the first write the upper 16 bits are output to an internal buffer register. Then, in the second write, the lower 16 bits are output to TCNT0L, and at the same time, the upper 16 bits held in the internal buffer register are output to TCNT0H to complete the write. The above method performs simultaneous reading and simultaneous writing of 32-bit data, preventing contention with an up-count.

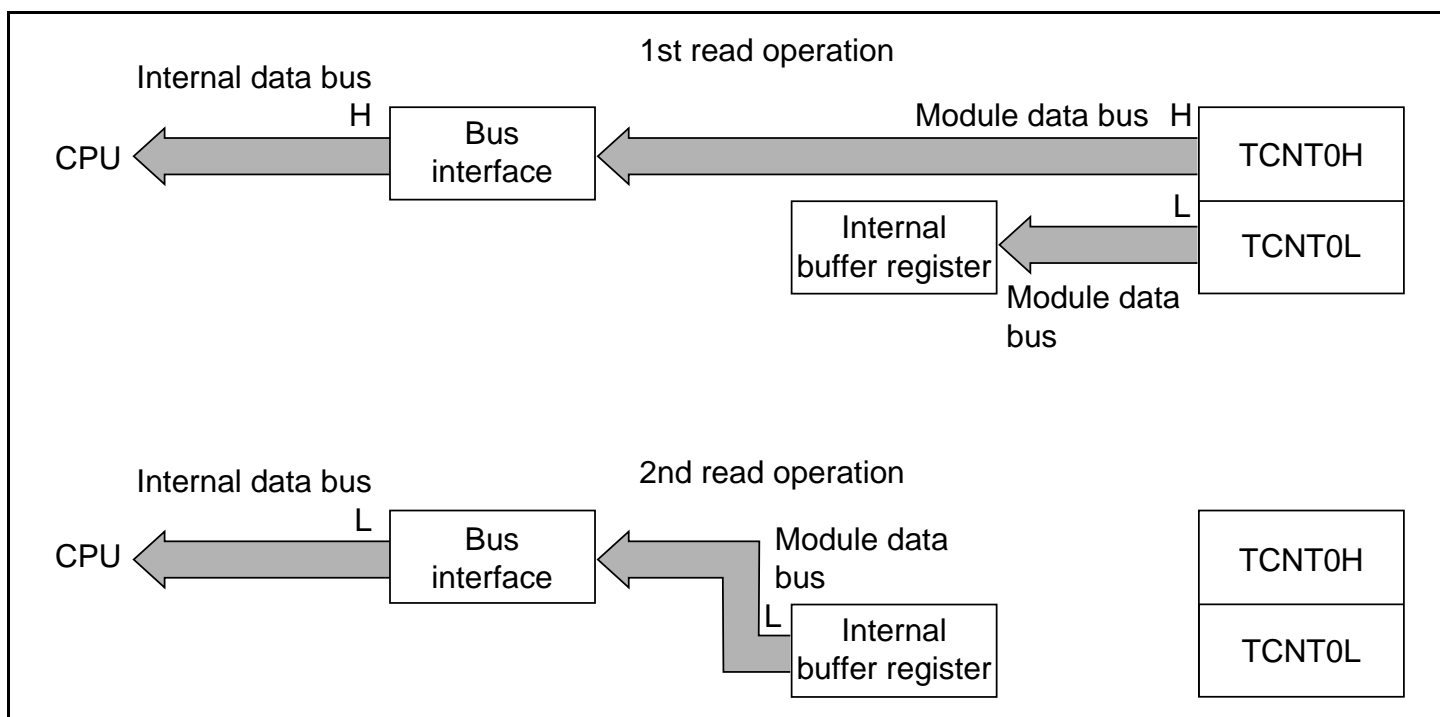
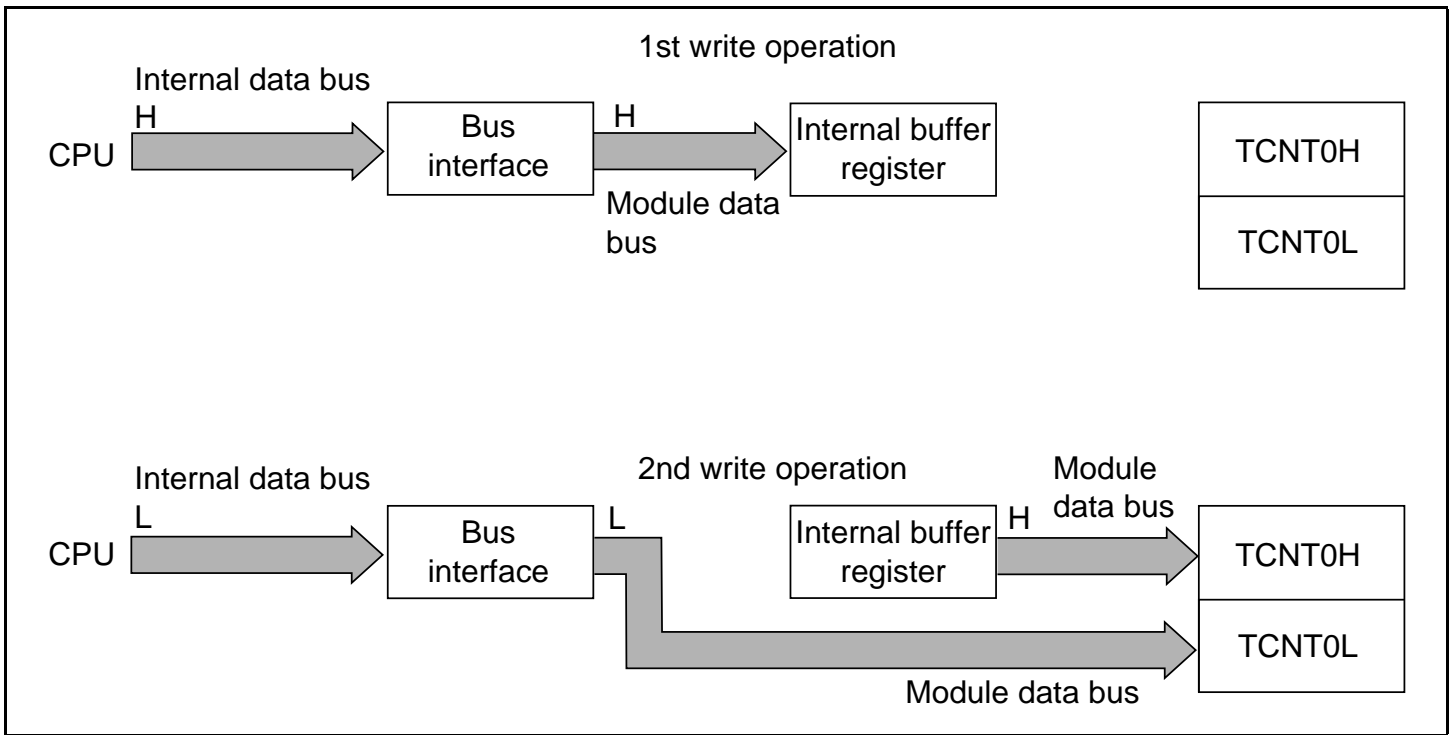


Figure 11.44 Read from TCNT0



**Figure 11.45 Write to TCNT0**

## 11.5.2 Registers Permitting 8-Bit, 16-Bit, or 32-Bit Access

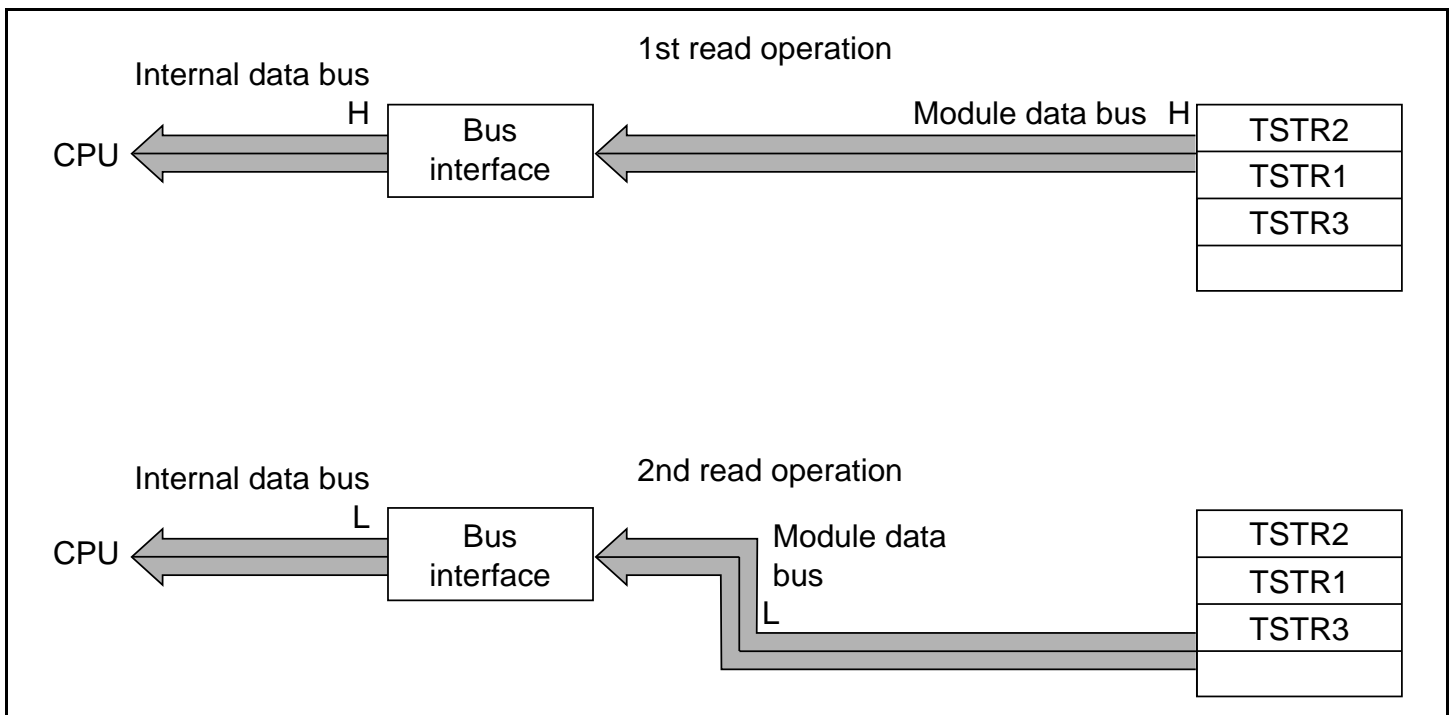
Timer registers 1, 2, and 3 (TSTR1, TSTR2, TSTR3) are 8-bit registers. As these registers are connected to the CPU via an internal 16-bit data bus, a simultaneous 32-bit read or write access to TSTR1, TSTR2, and TSTR3 is automatically divided into two 16-bit accesses.

Figure 11.46 shows a read from TSTR, and figure 11.47 a write to TSTR.

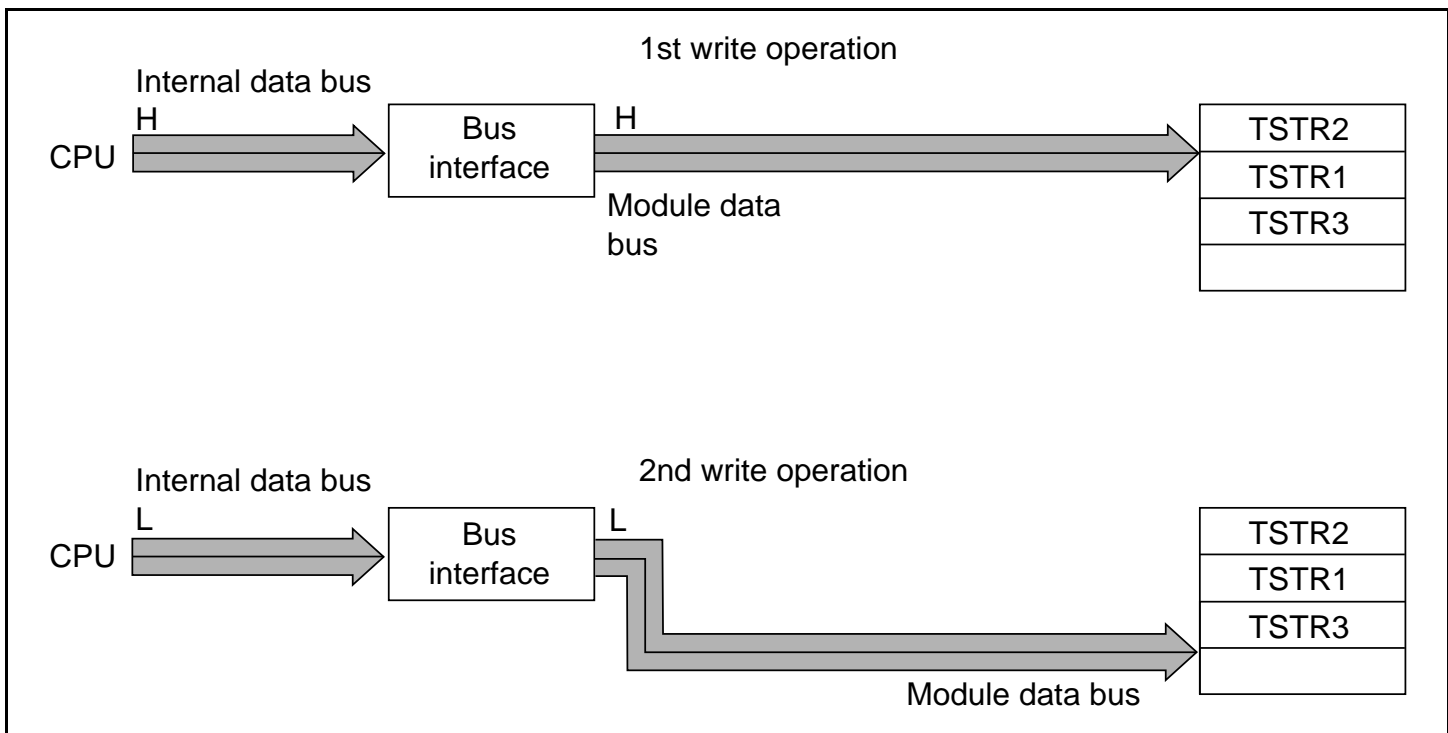
When reading TSTR, in the first read the TSTR1 and TSTR2 (upper 16-bit) value is output to the internal data bus. Then, in the second read, the TSTR3 (lower 16-bit) value is output to the internal data bus.

When writing to TSTR, in the first write the upper 16 bits are written to TSTR1 and TSTR2. Then, in the second write, the lower 16 bits are written to TSTR3. Note that, with the above method, in a 32-bit write the write timing is not the same for TSTR1/TSTR2 and TSTR3.

For information on 8-bit and 16-bit access, see section 11.5.4, 8-Bit or 16-Bit Accessible Registers.



**Figure 11.46 Read from TSTR1, TSTR2, and TSTR3**

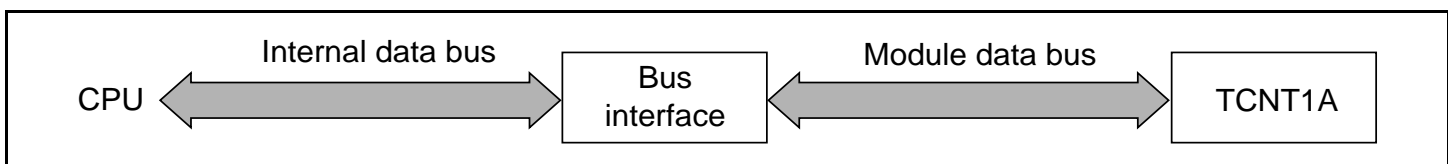


**Figure 11.47 Write to TSTR1, TSTR2 and TSTR3**

### 11.5.3 Registers Requiring 16-Bit Access

The free-running counters (TCNT; but excluding TCNT0, TCNT10A, TCNT10B, TCNT10D, and TCNT10H), the general registers (GR; but excluding GR9A to GR9D), down-counters (DCNT), offset base register (OSBR), cycle registers (CYLR), buffer registers (BFR), duty registers (DTR), timer connection register (TCNR), one-shot pulse terminate register (OTR), down-count start register (DSTR), output compare registers (OCR: but excluding OCR10B), reload registers (RLDR8, RLD10C), correction counter clear register (TCCLR10), timer interrupt enable register (TIER), and timer status register (TSR) are 16-bit registers. These registers are connected to the CPU via an internal 16-bit data bus, and can be read or written (read only, in the case of OSBR) a word at a time.

Figure 11.48 shows the operation when performing a word read or write access to TCNT1A.



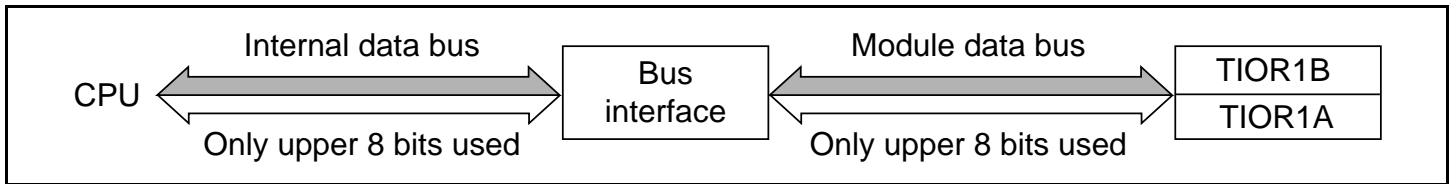
**Figure 11.48 TCNT1A Read/Write Operation**

### 11.5.4 8-Bit or 16-Bit Accessible Registers

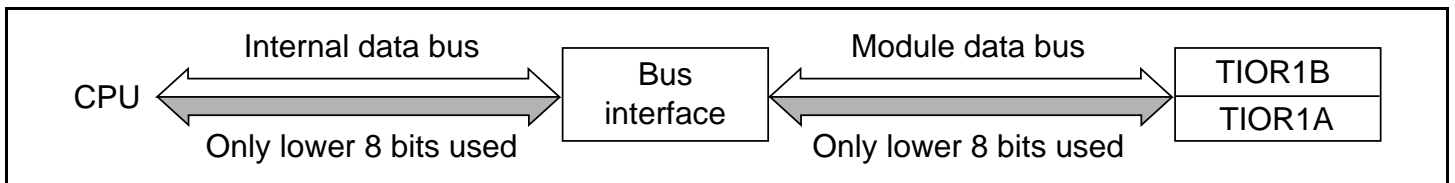
The timer control registers (TCR1A, TCR1B, TCR2A, TCR2B, TCR6A, TCR6B, TCR7A, TCR7B), timer I/O control registers (TIOR1A to TIOR1D, TIOR2A to TIOR2D, TIOR3A, TIOR3B, TIOR4A, TIOR4B, TIOR5A, TIOR5B), and the timer start register (TSTR1, TSTR2, TSTR3) are 8-bit registers. These registers are connected to the CPU with the upper 8 bits or lower 8 bits of the internal 16-bit data bus, and can be read or written a byte at a time.

In addition, a pair of 8-bit registers for which only the least significant bit of the address is different, such as timer I/O control register 1A (TIOR1A) and timer I/O control register 1B (TIOR1B), can be read or written in combination a word at a time.

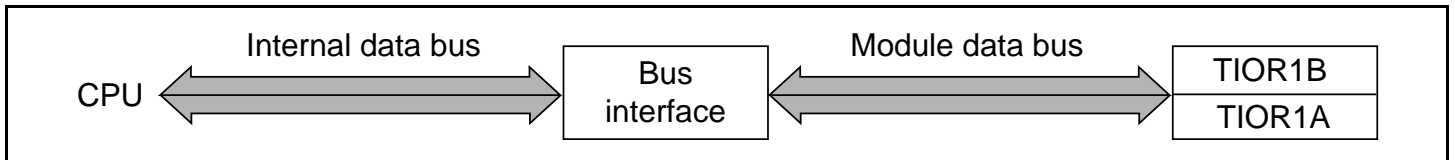
Figures 11.49 and 11.50 show the operation when performing individual byte read or write accesses to TIOR1A and TIOR1B. Figure 11.51 shows the operation when performing a word read or write access to TIOR1A and TIOR1B simultaneously.



**Figure 11.49 Byte Read/Write Access to TIOR1B**



**Figure 11.50 Byte Read/Write Access to TIOR1A**



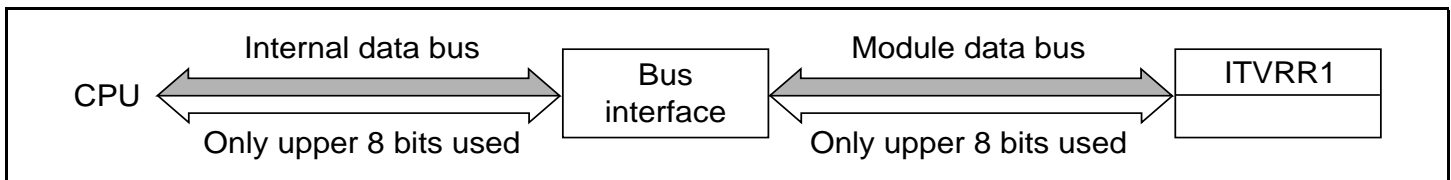
**Figure 11.51 Word Read/Write Access to TIOR1A and TIOR1B**



### 11.5.5 Registers Requiring 8-Bit Access

The timer mode register (TMDR), prescaler register (PSCR), timer I/O control registers (TIOR0, TIOR10, TIOR11), trigger mode register (TRGMDR), interval interrupt request register (ITVRR), timer control registers (TCR3, TCR4, TCR5, TCR8, TCR9A to TCR9C, TCR10, TCR11), PWM mode register (PMDR), reload enable register (RLDENR), free-running counters (TCNT10B, TCNT10D, TCNT10H), event counter (ECNT), general registers (GR9A to GR9F), output compare register (OCR10B), and noise canceler register (NCR) are 8-bit registers. These registers are connected to the CPU with the upper 8 bits of the internal 16-bit data bus, and can be read or written a byte at a time.

Figure 11.52 shows the operation when performing individual byte read or write accesses to ITVRR1.

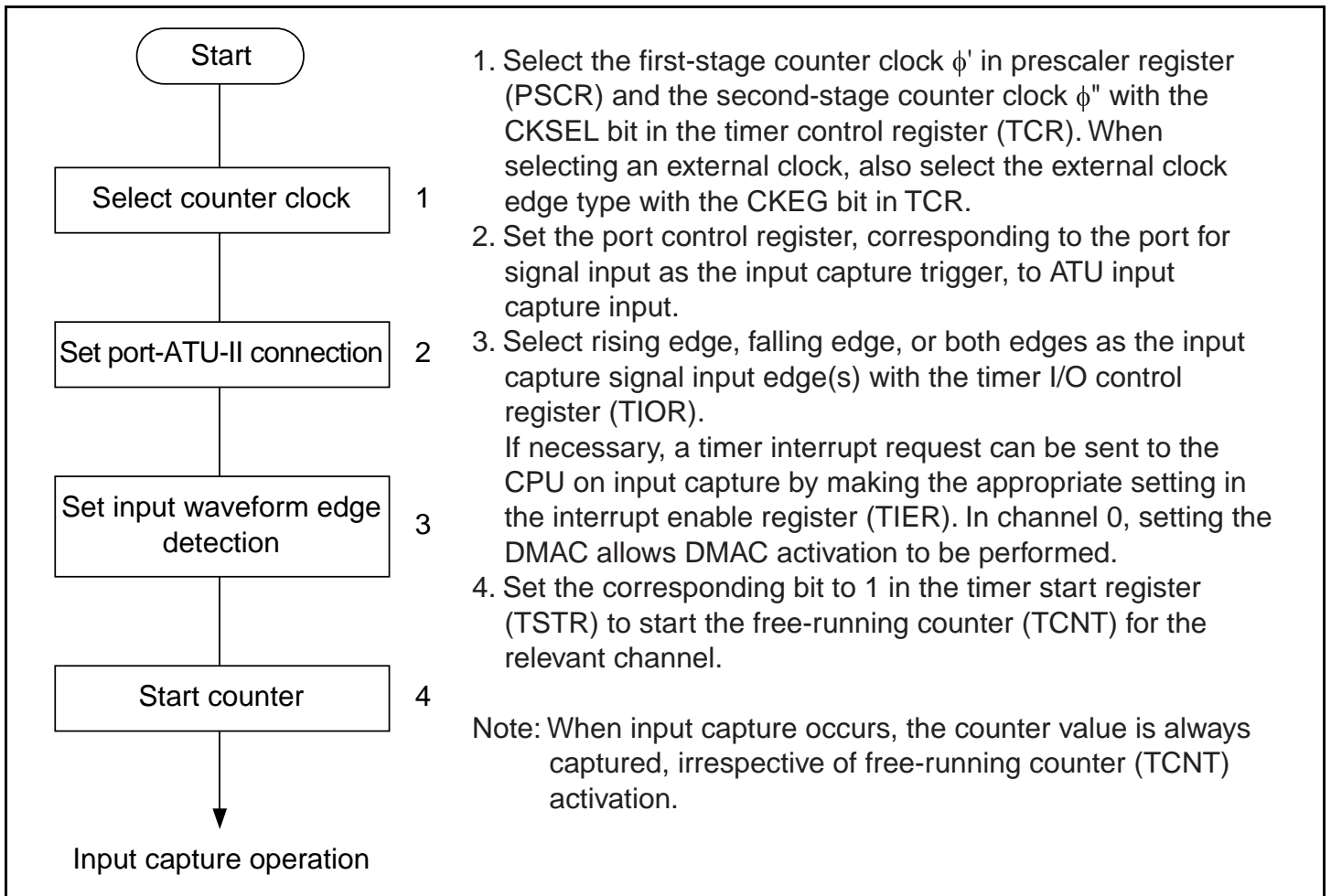


**Figure 11.52 Byte Read/Write Access to ITVRR1**

## 11.6 Sample Setup Procedures

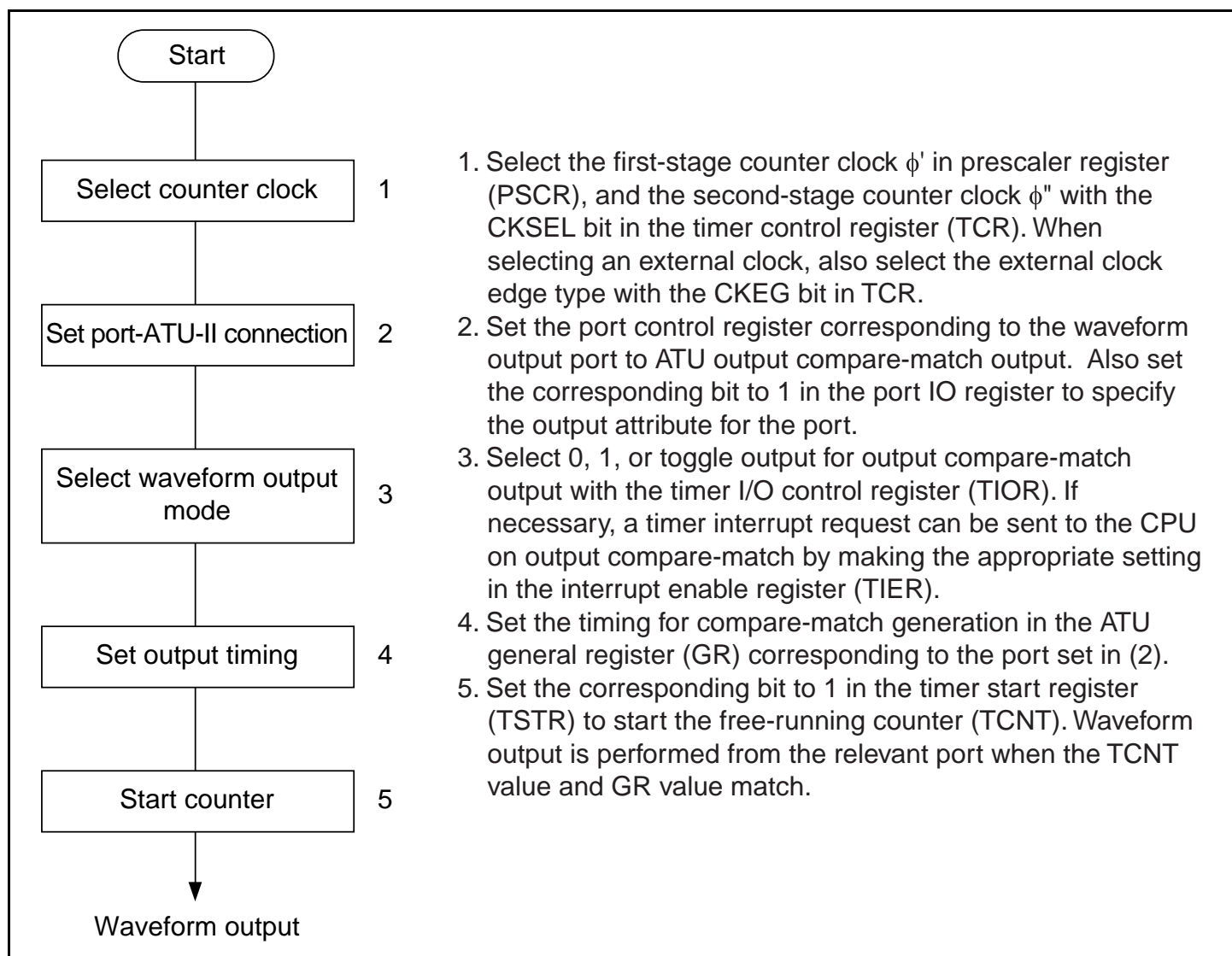
Sample setup procedures for activating the various ATU-II functions are shown below.

**Sample Setup Procedure for Input Capture:** An example of the setup procedure for input capture is shown in figure 11.53.



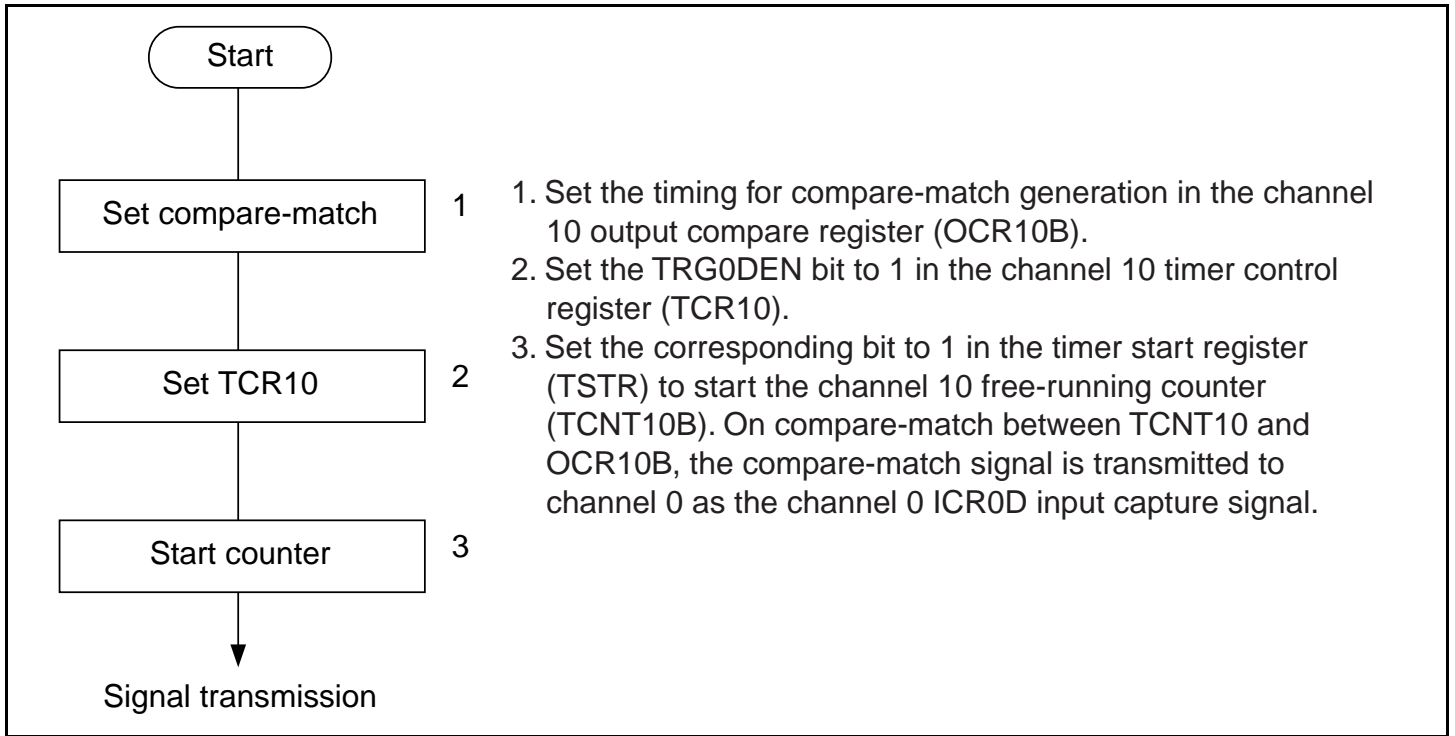
**Figure 11.53 Sample Setup Procedure for Input Capture**

**Sample Setup Procedure for Waveform Output by Output Compare-Match:** An example of the setup procedure for waveform output by output compare-match is shown in figure 11.54.



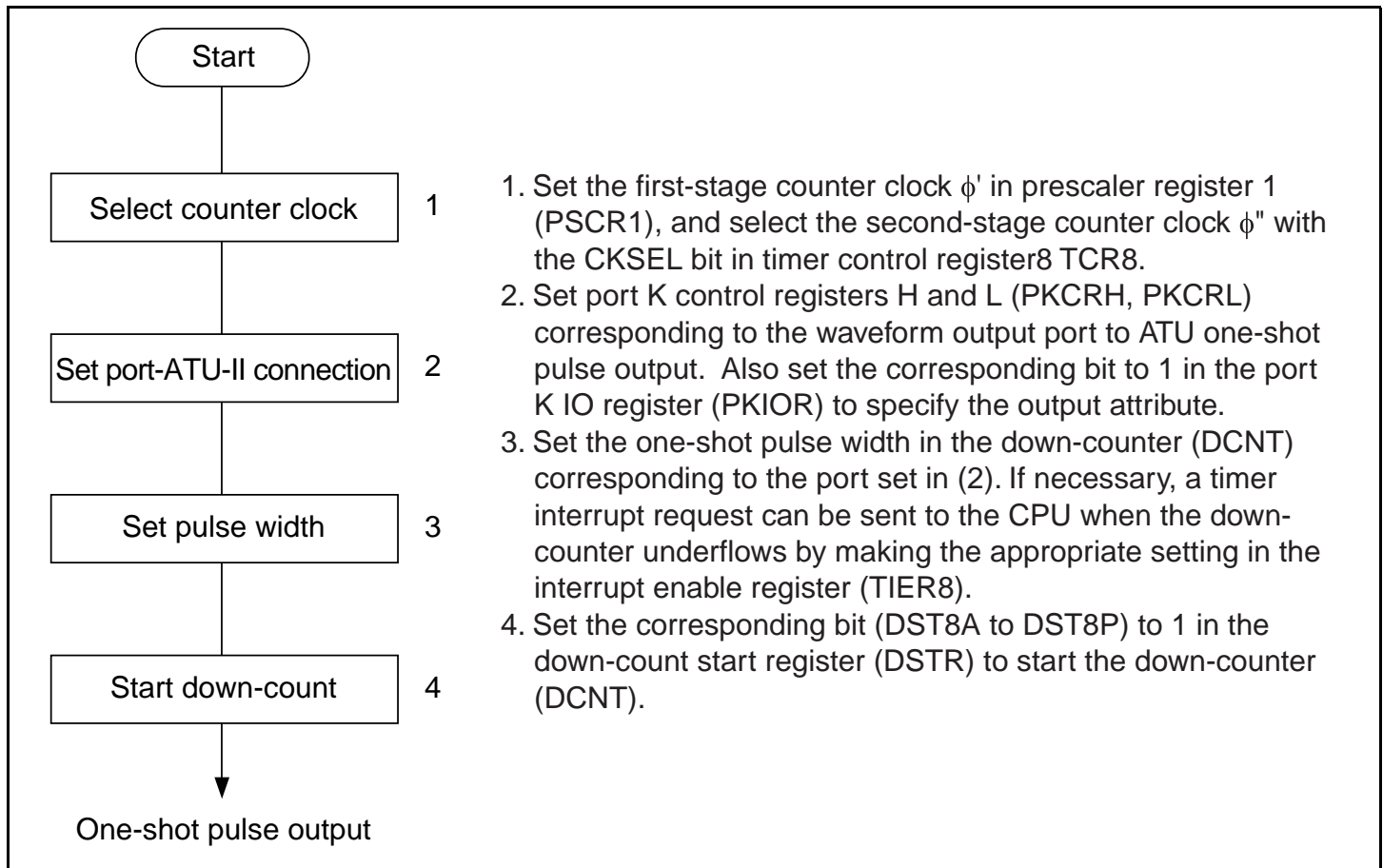
**Figure 11.54 Sample Setup Procedure for Waveform Output by Output Compare-Match**

**Sample Setup Procedure for Channel 0 Input Capture Triggered by Channel 10 Compare-Match:** An example of the setup procedure for compare-match signal transmission is shown in figure 11.55.



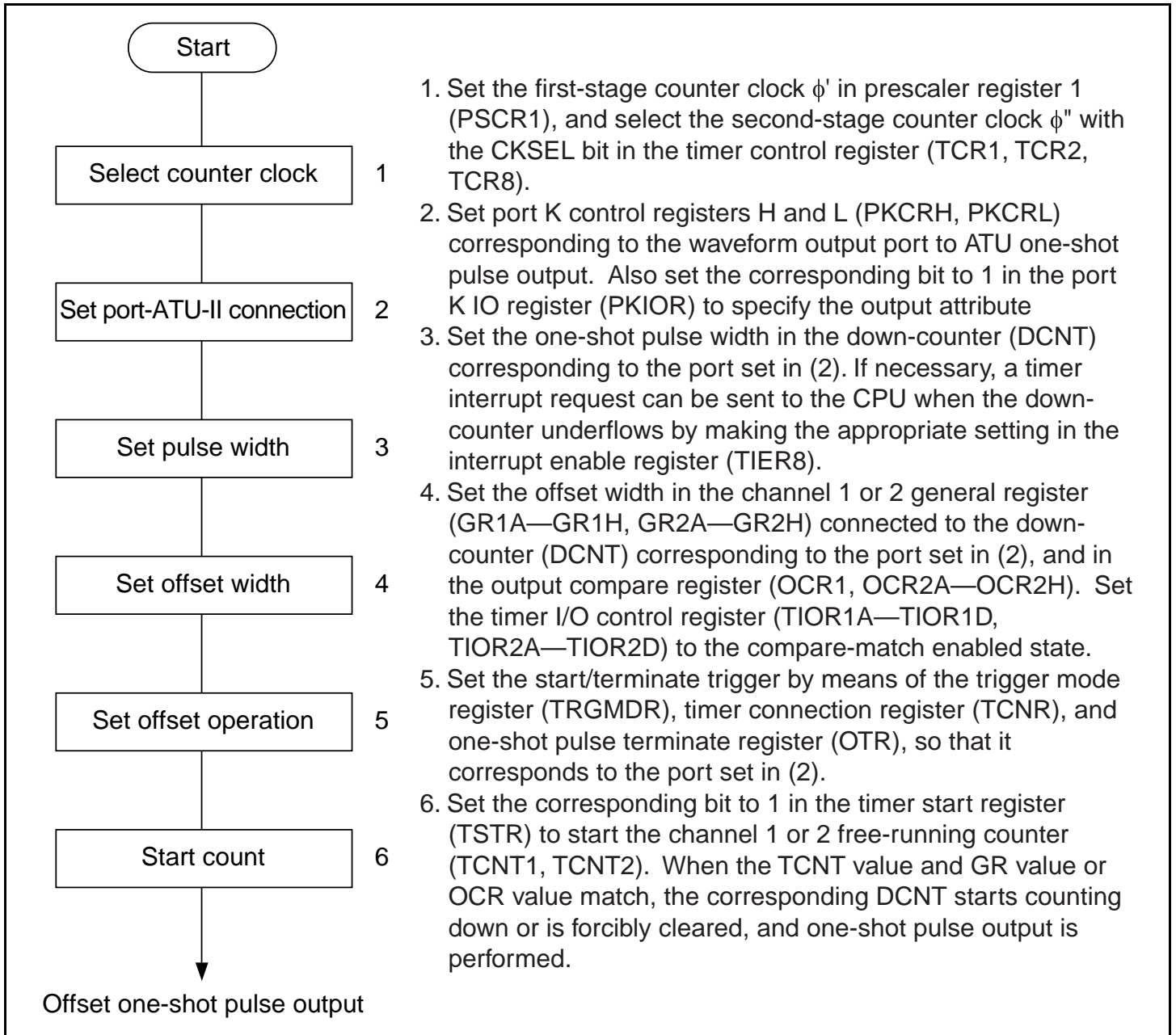
**Figure 11.55 Sample Setup Procedure for Compare-Match Signal Transmission**

**Sample Setup Procedure for One-Shot pulse Output:** An example of the setup procedure for one-shot pulse output is shown in figure 11.56.



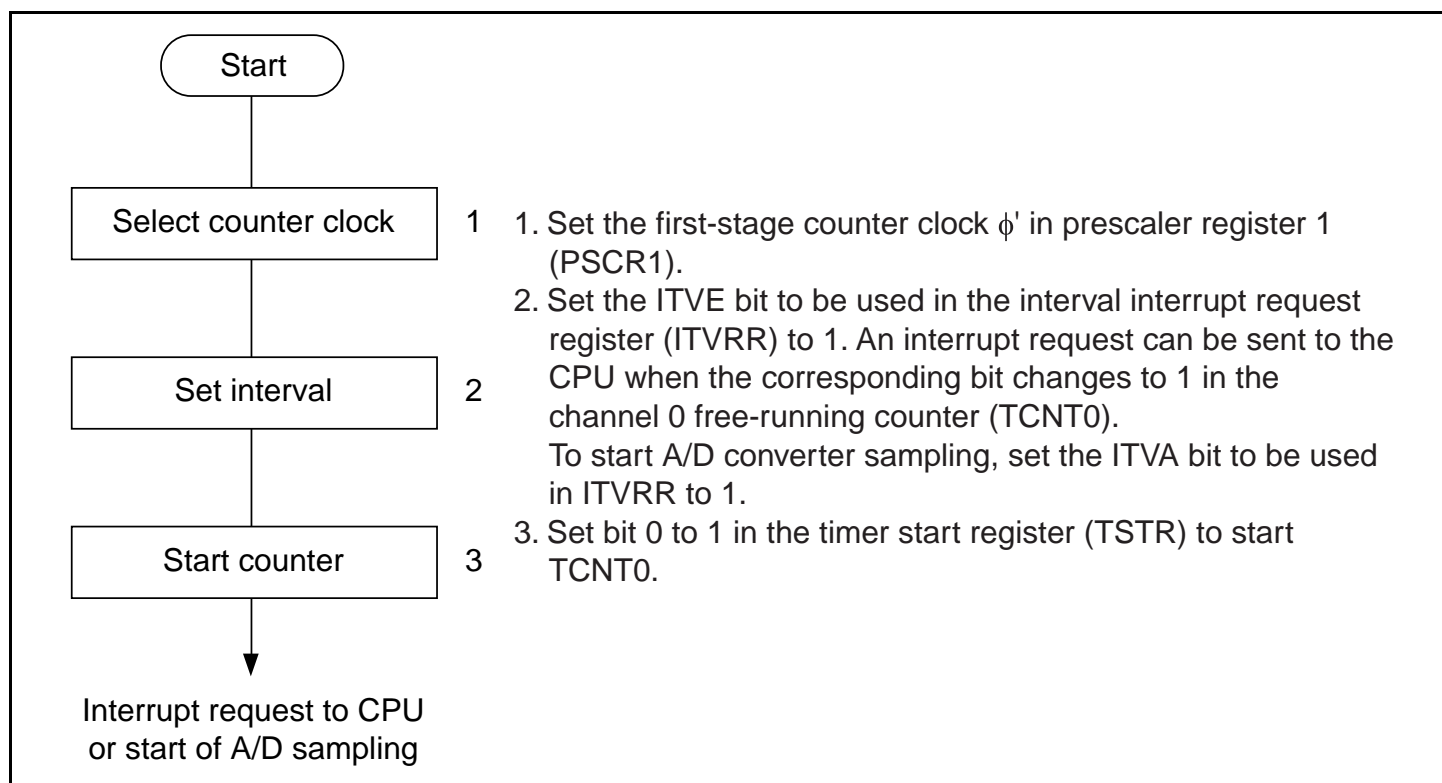
**Figure 11.56 Sample Setup Procedure for One-Shot Pulse Output**

**Sample Setup Procedure for Offset One-Shot Pulse Output/Cutoff Operation:** An example of the setup procedure for offset one-shot pulse output is shown in figure 11.57.



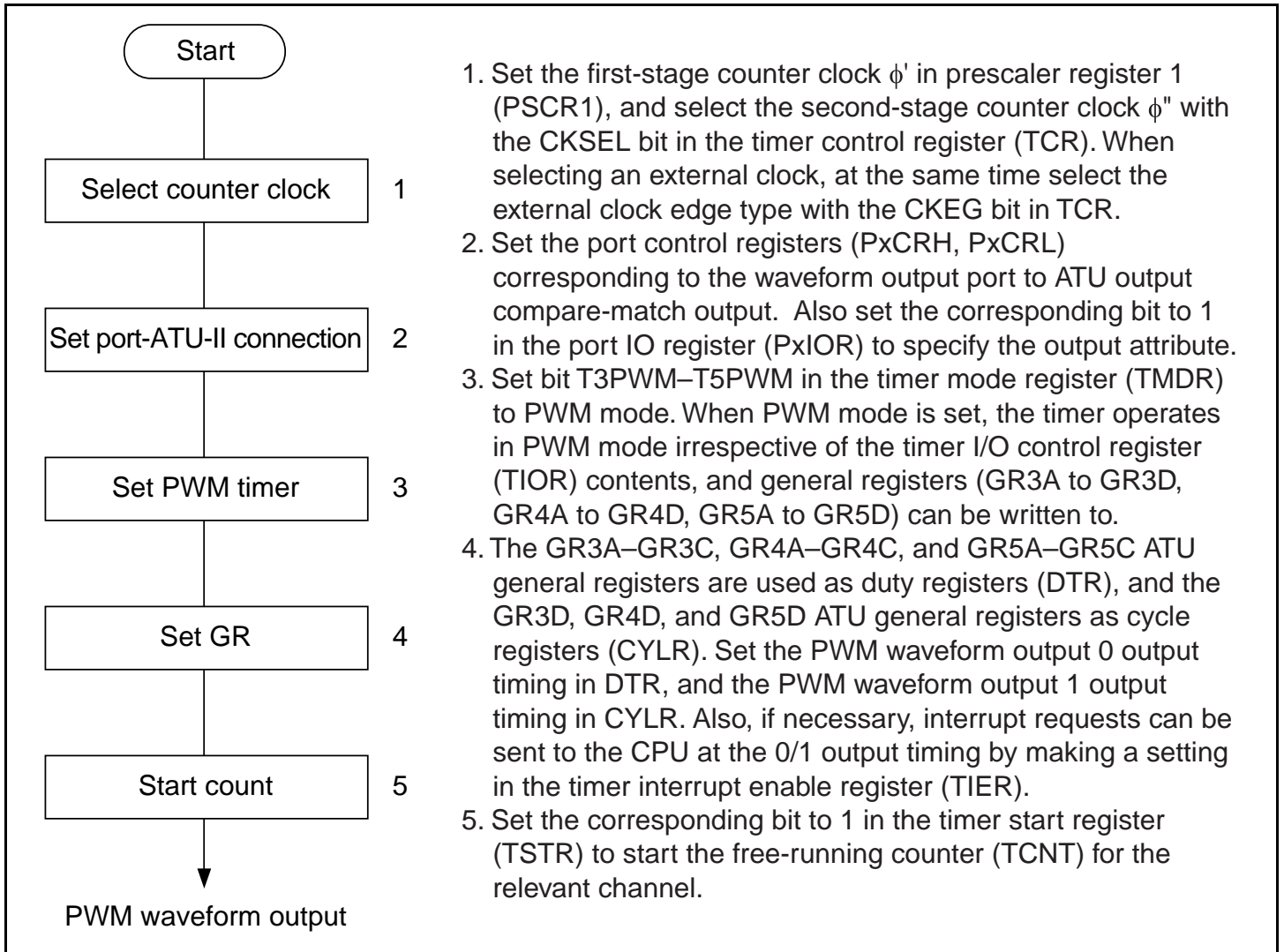
**Figure 11.57 Sample Setup Procedure for Offset One-Shot Pulse Output**

**Sample Setup Procedure for Interval Timer Operation:** An example of the setup procedure for interval timer operation is shown in figure 11.58.



**Figure 11.58 Sample Setup Procedure for Interval Timer Operation**

**Sample Setup Procedure for PWM Timer Operation (Channels 3 to 5):** An example of the setup procedure for PWM timer operation (channels 3 to 5) is shown in figure 11.59.

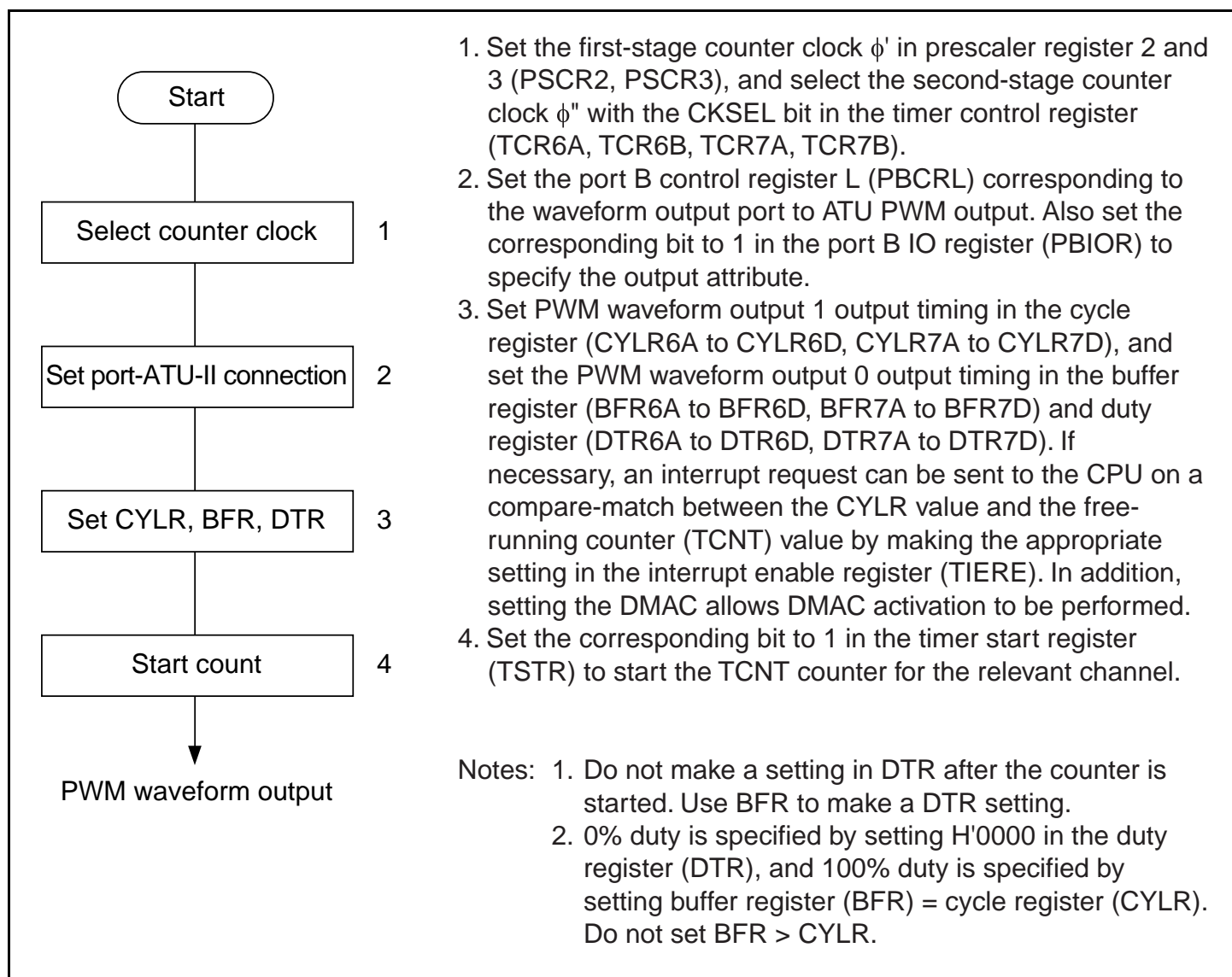


1. Set the first-stage counter clock  $\phi'$  in prescaler register 1 (PSCR1), and select the second-stage counter clock  $\phi''$  with the CKSEL bit in the timer control register (TCR). When selecting an external clock, at the same time select the external clock edge type with the CKEG bit in TCR.
2. Set the port control registers (PxCRH, PxCRH) corresponding to the waveform output port to ATU output compare-match output. Also set the corresponding bit to 1 in the port IO register (PxIOR) to specify the output attribute.
3. Set bit T3PWM–T5PWM in the timer mode register (TMDR) to PWM mode. When PWM mode is set, the timer operates in PWM mode irrespective of the timer I/O control register (TIOR) contents, and general registers (GR3A to GR3D, GR4A to GR4D, GR5A to GR5D) can be written to.
4. The GR3A–GR3C, GR4A–GR4C, and GR5A–GR5C ATU general registers are used as duty registers (DTR), and the GR3D, GR4D, and GR5D ATU general registers as cycle registers (CYLR). Set the PWM waveform output 0 output timing in DTR, and the PWM waveform output 1 output timing in CYLR. Also, if necessary, interrupt requests can be sent to the CPU at the 0/1 output timing by making a setting in the timer interrupt enable register (TIER).
5. Set the corresponding bit to 1 in the timer start register (TSTR) to start the free-running counter (TCNT) for the relevant channel.

**Figure 11.59 Sample Setup Procedure for PWM Timer Operation (Channels 3 to 5)**

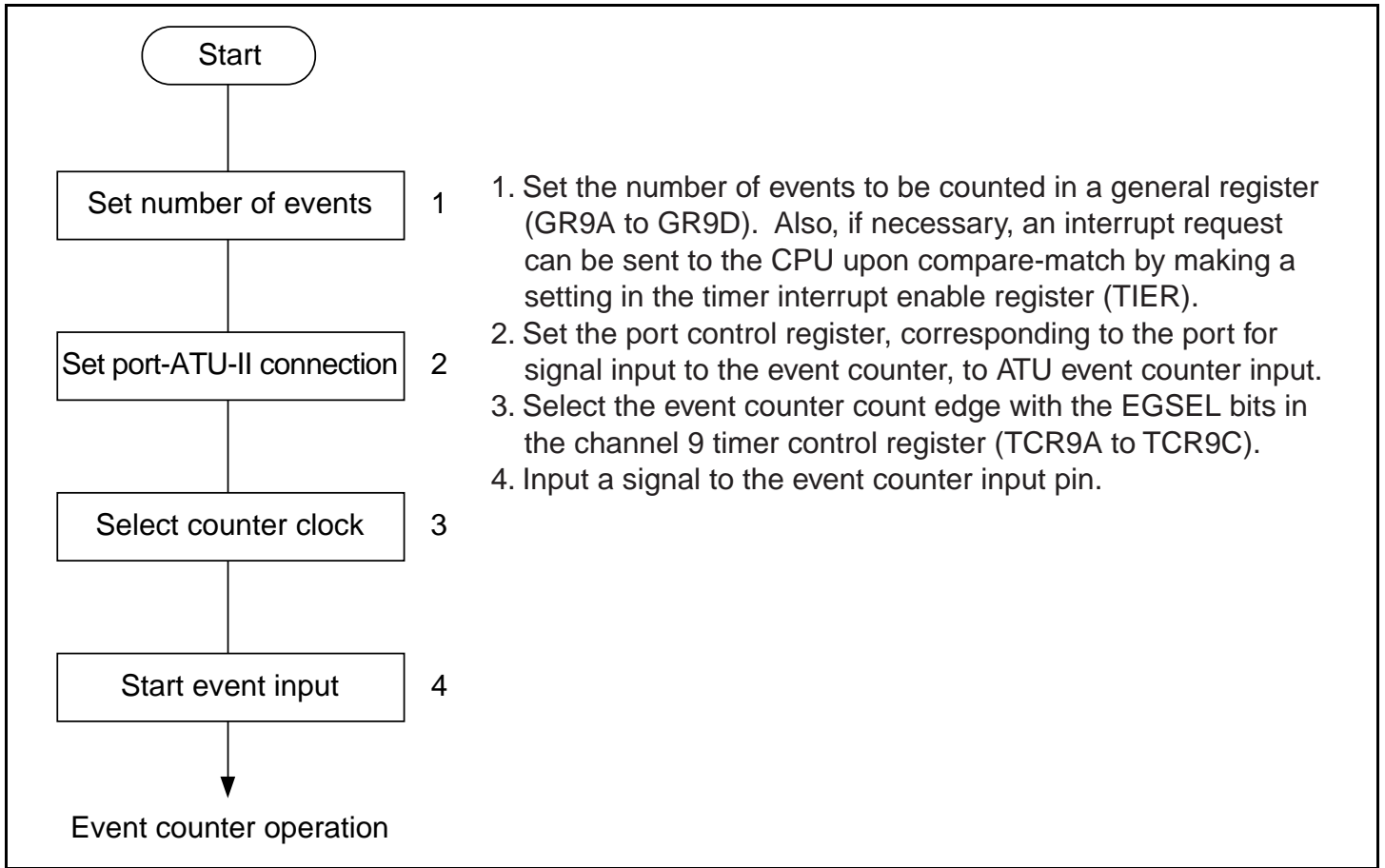


**Sample Setup Procedure for PWM Timer Operation (Channels 6 and 7):** An example of the setup procedure for PWM timer operation (channels 6 and 7) is shown in figure 11.60.



**Figure 11.60 Sample Setup Procedure for PWM Timer Operation (Channels 6 and 7)**

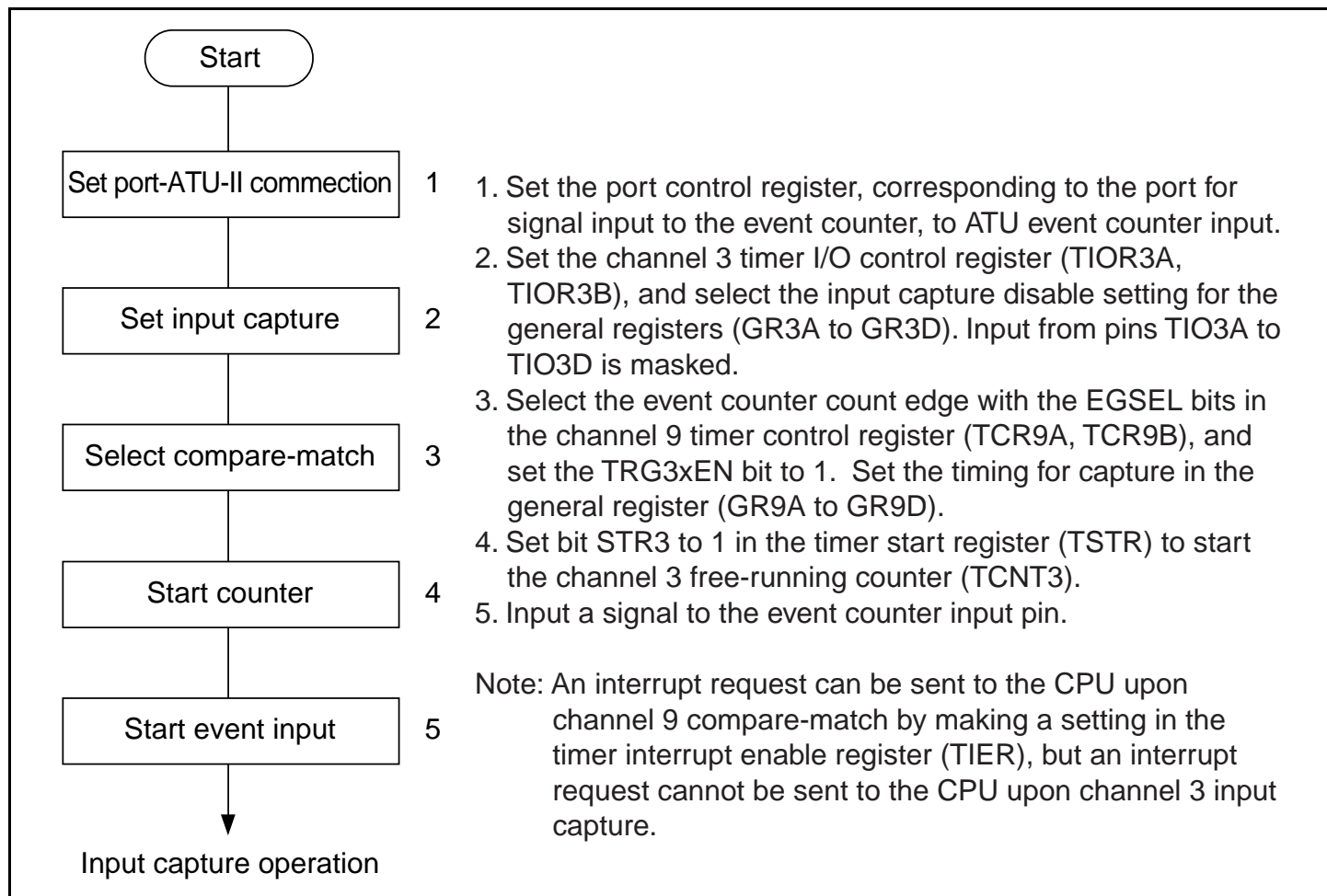
**Sample Setup Procedure for Event Counter Operation:** An example of the setup procedure for event counter operation is shown in figure 11.61.



**Figure 11.61 Sample Setup Procedure for Event Counter Operation**

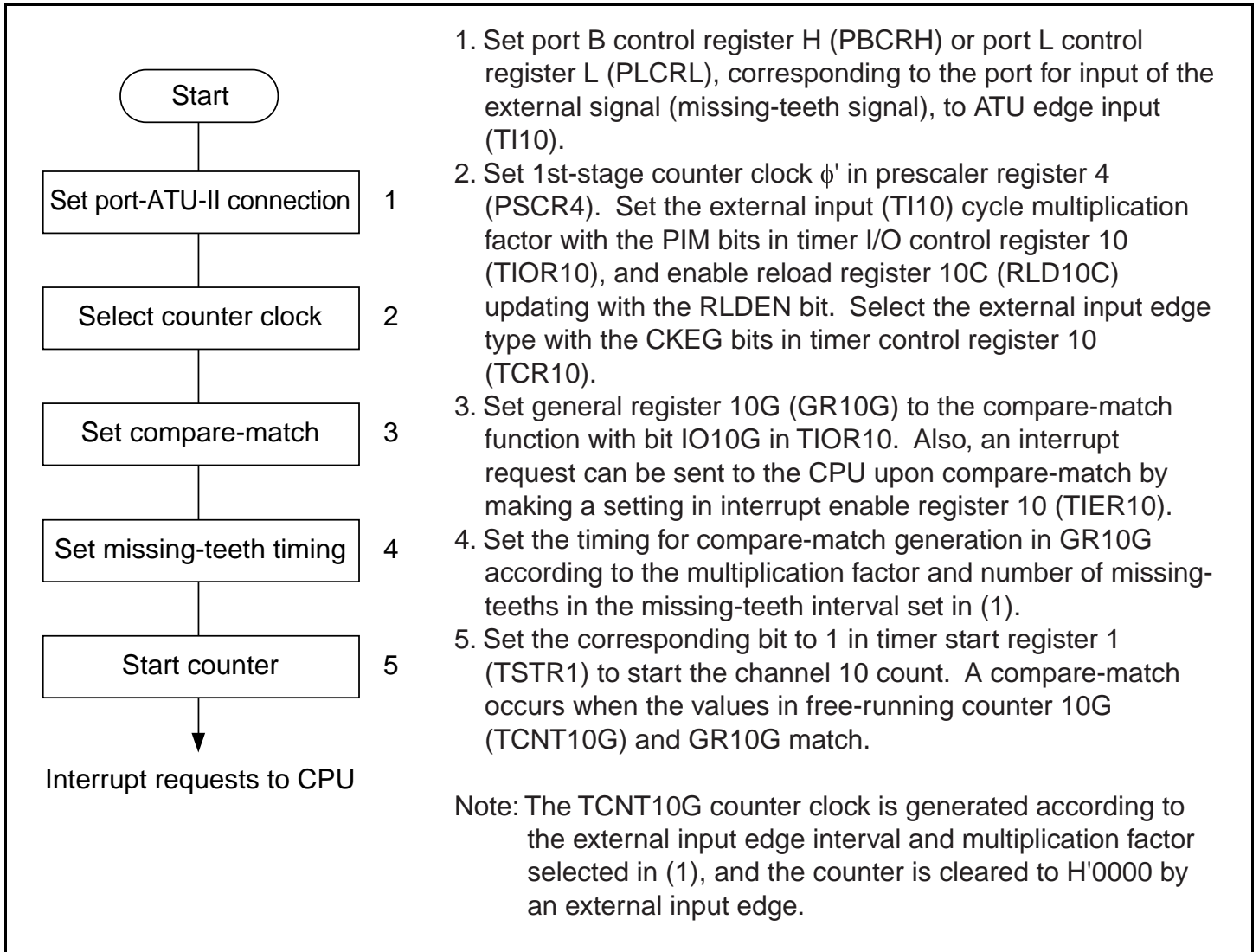
## Sample Setup Procedure for Channel 3 Input Capture Triggered by Channel 9 Compare-Match

**Match:** An example of the setup procedure for compare-match signal transmission is shown in figure 11.62.



**Figure 11.62 Sample Setup Procedure for Compare-Match Signal Transmission**

**Sample Setup Procedure for Channel 10 Missing-Teeth Detection:** An example of the setup procedure for missing-teeth detection is shown in figure 11.63.



**Figure 11.63 Sample Setup Procedure for Missing-Teeth Detection**

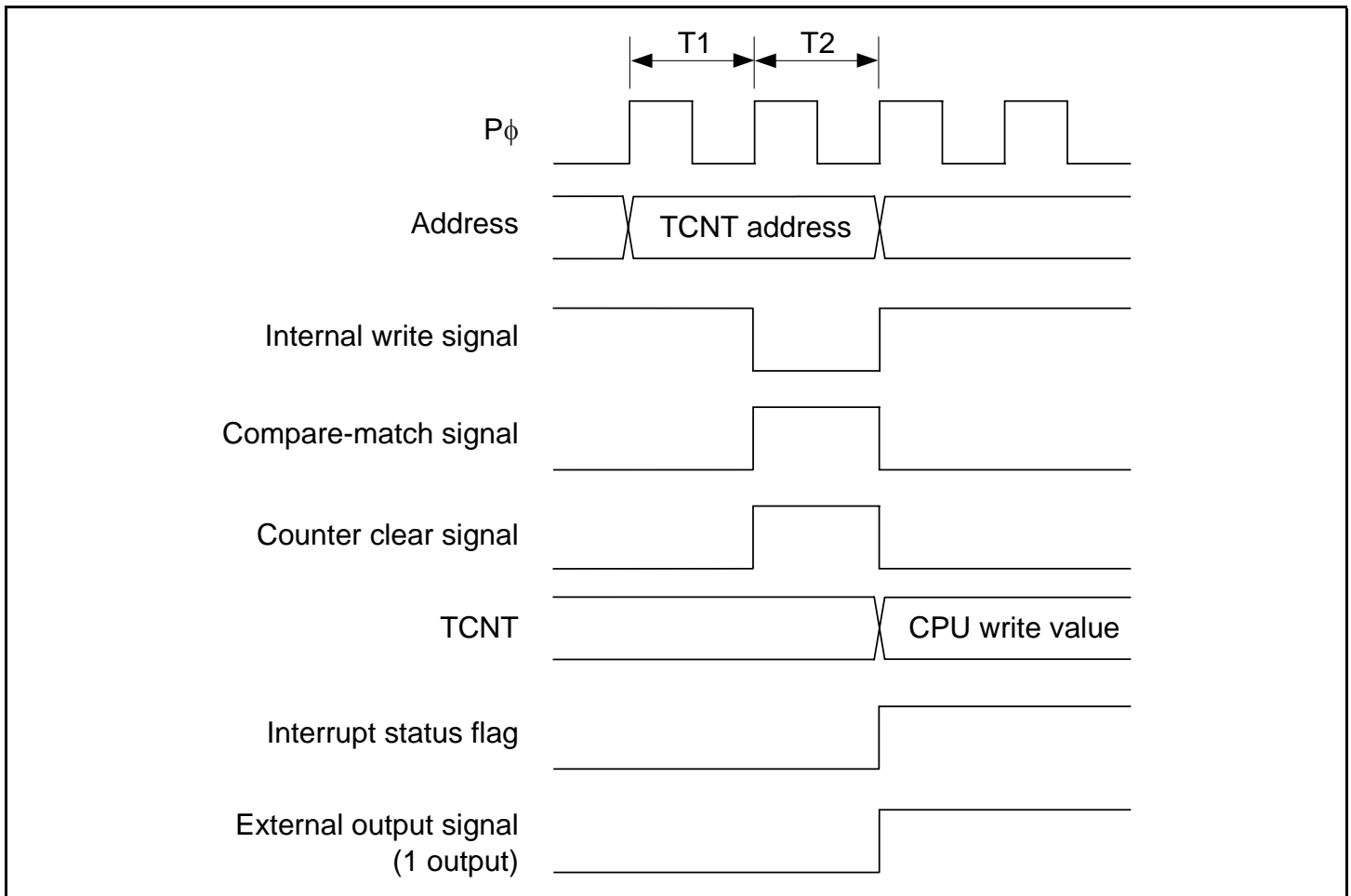
## 11.7 Usage Notes

Note that the kinds of operation and contention described below occur during ATU operation.

**Contention between TCNT Write and Clearing by Compare-Match:** With channel 3 to 7 free-running counters (TCNT3 to TCNT5, TCNT6A to TCNT6D, TCNT7A to TCNT7D), if a compare-match occurs in the T2 state of a CPU write cycle when counter clearing by compare-match has been set, or when PWM mode is used, the write to TCNT has priority and TCNT clearing is not performed.

The compare-match remains valid, and writing of 1 to the interrupt status flag and waveform output to an external destination are performed in the same way as for a normal compare-match.

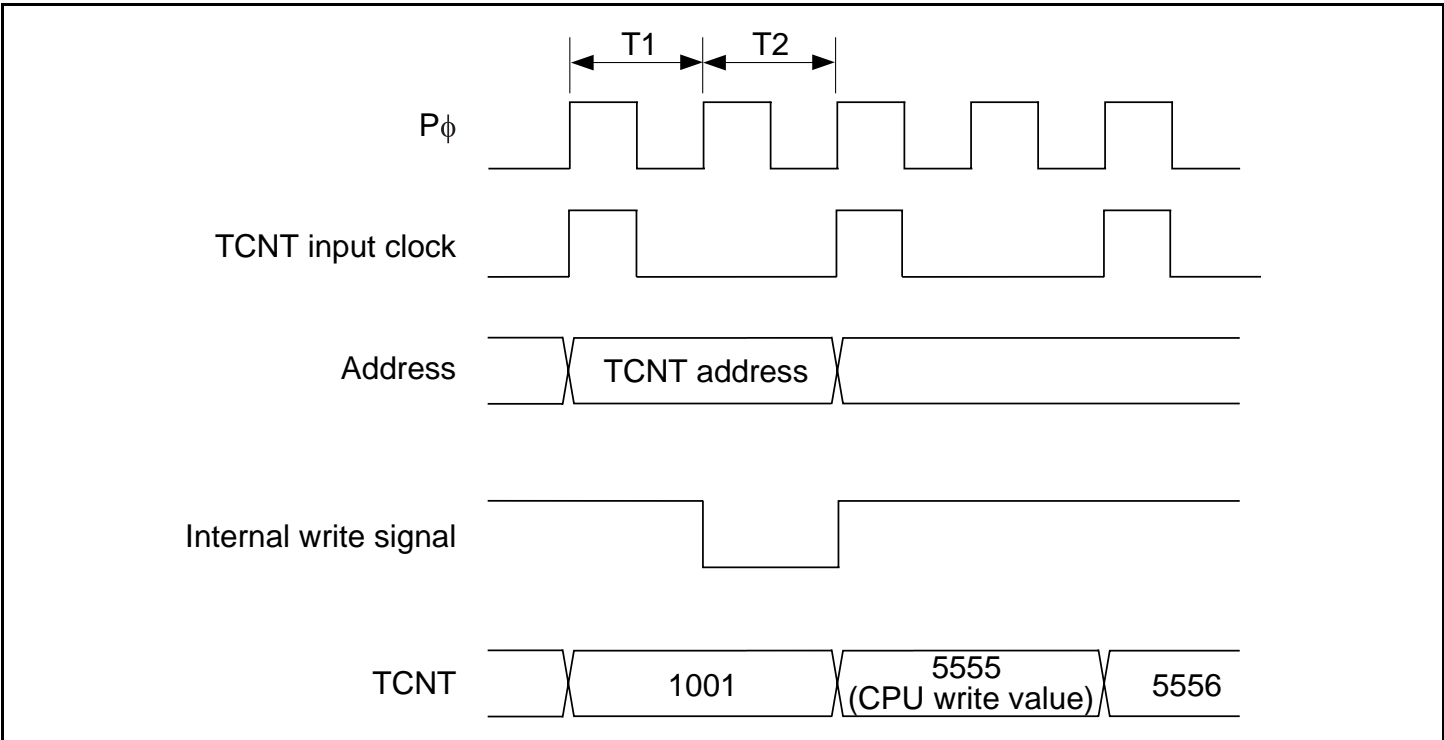
The timing in this case is shown in figure 11.64.



**Figure 11.64** Contention between TCNT Write and Clear

**Contention between TCNT Write and Increment:** If a write to a channel 0 to 11 free-running counter (TCNT0, TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3 to TCNT5, TCNT6A to TCNT6D, TCNT7A to TCNT7D, TCNT10A to TCNT10H, TCNT11), down-counter (DCNT8A to DCNT8P), or event counter 9 (ECNT9A to ECNT9F) is performed while that counter is counting up or down, the write to the counter has priority and the counter is not incremented or decremented.

The timing in this case is shown in figure 11.65. In this example, the CPU writes H'5555 at the point at which TCNT is to be incremented from H'1001 to H'1002.

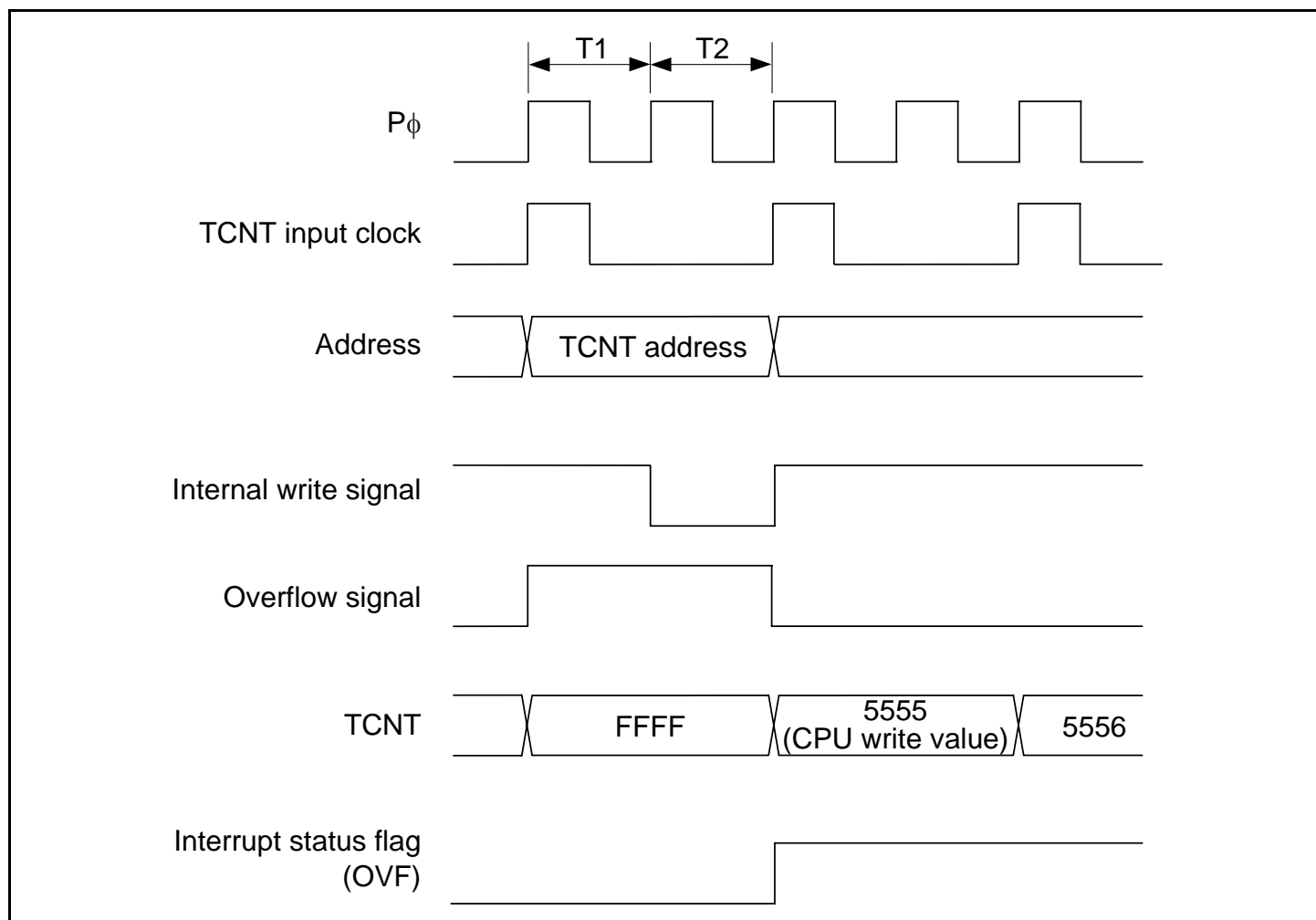


**Figure 11.65 Contention between TCNT Write and Increment**

**Contention between TCNT Write and Counter Clearing by Overflow:** With channel 0 to 5 and 11 free-running counters (TCNT0, TCNT1A, TCNT1B, TCNT2A, TCNT2B, TCNT3 to TCNT5, TCNT11), if overflow occurs in the T2 state of a CPU write cycle, the write to TCNT has priority and TCNT is not cleared.

Writing of 1 to the interrupt status flag (OVF) due to the overflow is performed in the same way as for normal overflow.

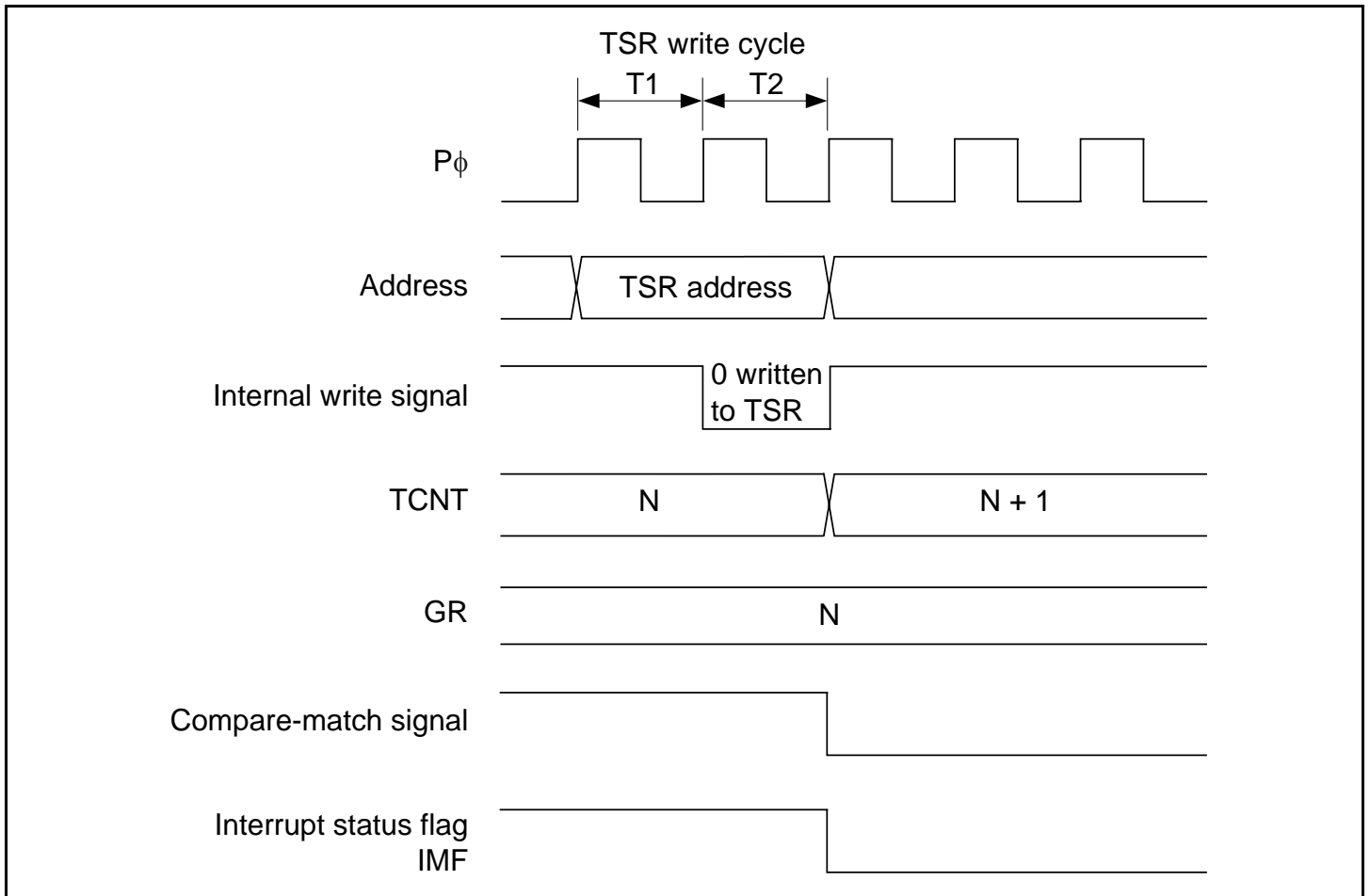
The timing in this case is shown in figure 11.66. In this example, H'5555 is written at the point at which TCNT overflows.



**Figure 11.66 Contention between TCNT Write and Overflow**

**Contention between Interrupt Status Flag Setting by Interrupt Generation and Clearing:** If an event such as input capture/compare-match or overflow/underflow occurs in the T2 state of an interrupt status flag 0 write cycle by the CPU, clearing by the 0 write has priority and the interrupt status flag is cleared.

The timing in this case is shown in figure 11.67.

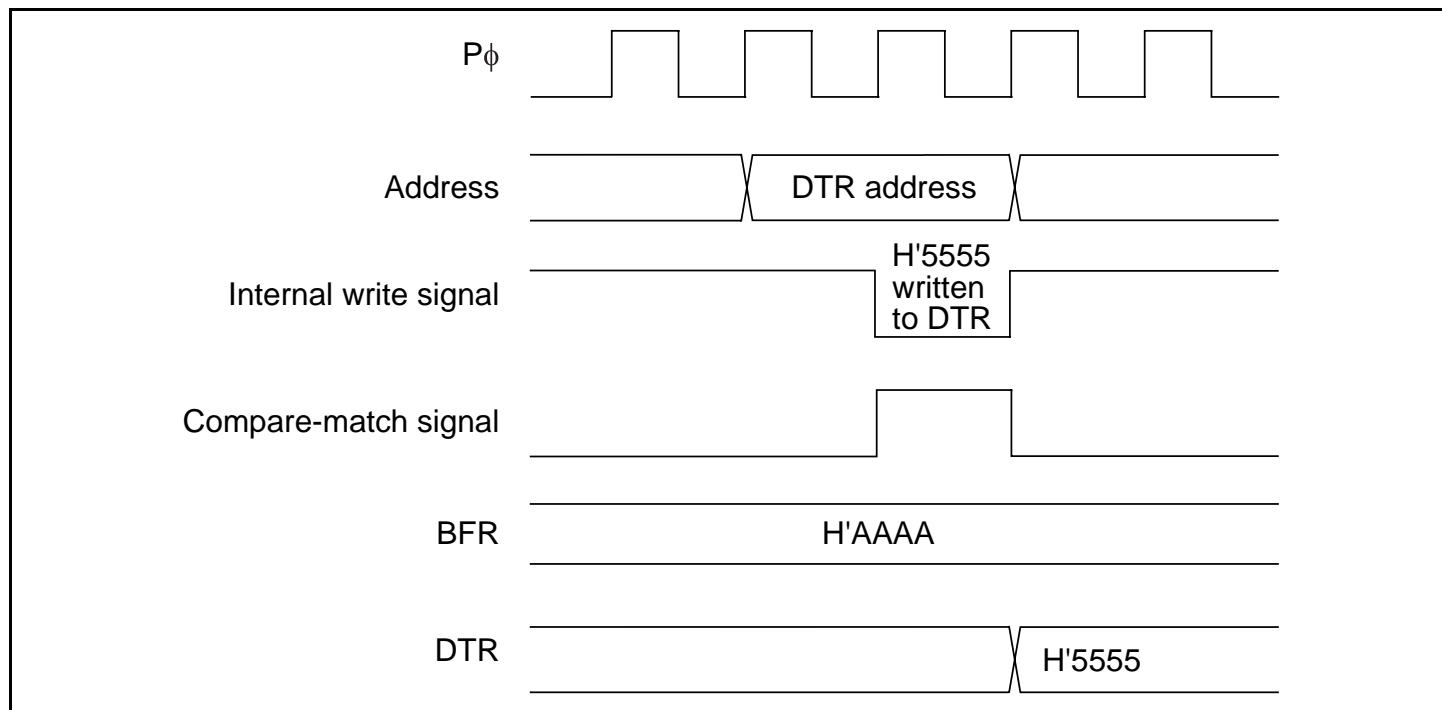


**Figure 11.67 Contention between Interrupt Status Flag Setting by Compare-Match and Clearing**



**Contention between DTR Write and BFR Value transfer by Buffer Function:** In channels 6 and 7, if there is contention between transfer of the buffer register (BFR) value to the corresponding duty register (DTR) due to a cycle register (CYLR) compare-match, and a write to DTR by the CPU, the CPU write value is written to DTR.

Figure 11.68 shows an example in which contention arises when the BFR value is H'AAAA and the value to be written to DTR is H'5555.

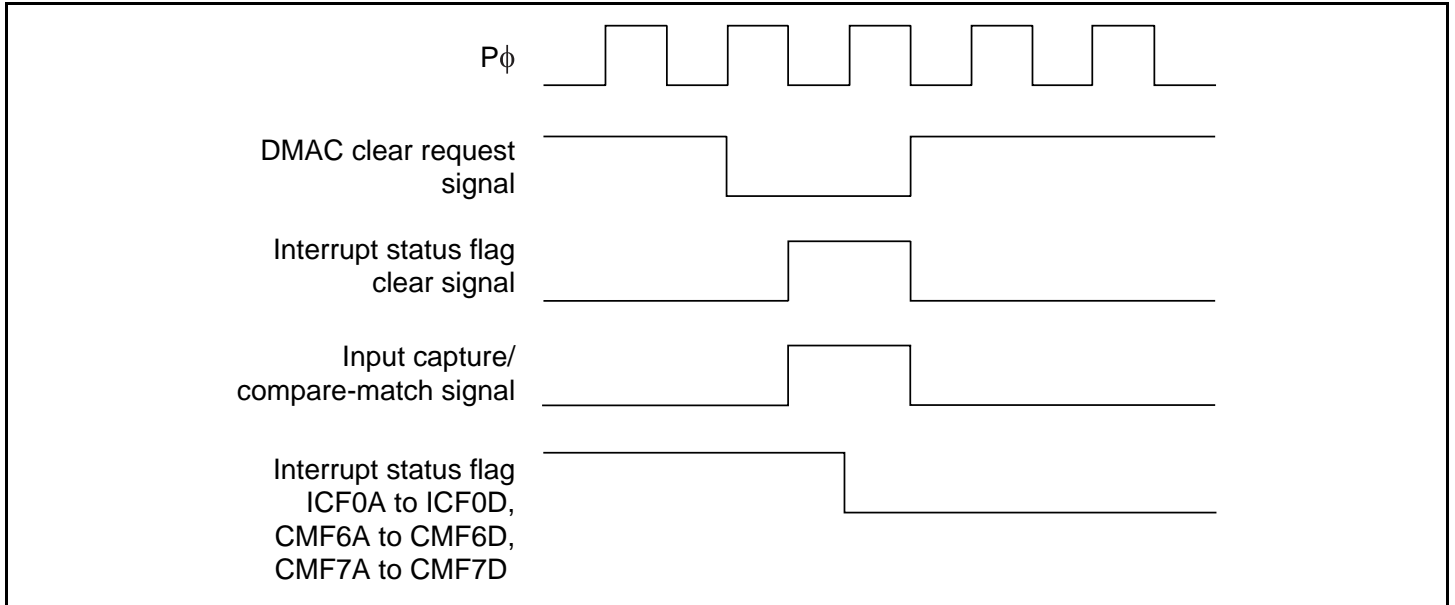


**Figure 11.68 Contention between DTR Write and BFR Value Transfer by Buffer Function**

### Contention between Interrupt Status Flag Clearing by DMAC and Setting by Input

**Capture/Compare-Match:** If a clear request signal is generated by the DMAC when the interrupt status flag (ICF0A to ICF0D, CMF6A to CMF6D, CMF7A to CMF7D) is set by input capture (ICR0A to ICR0D) or compare-match (CYLR6A to CYLR6D, CYLR7A to CYLR7D), clearing by the DMAC has priority and the interrupt status flag is not set.

The timing in this case is shown in figure 11.69.

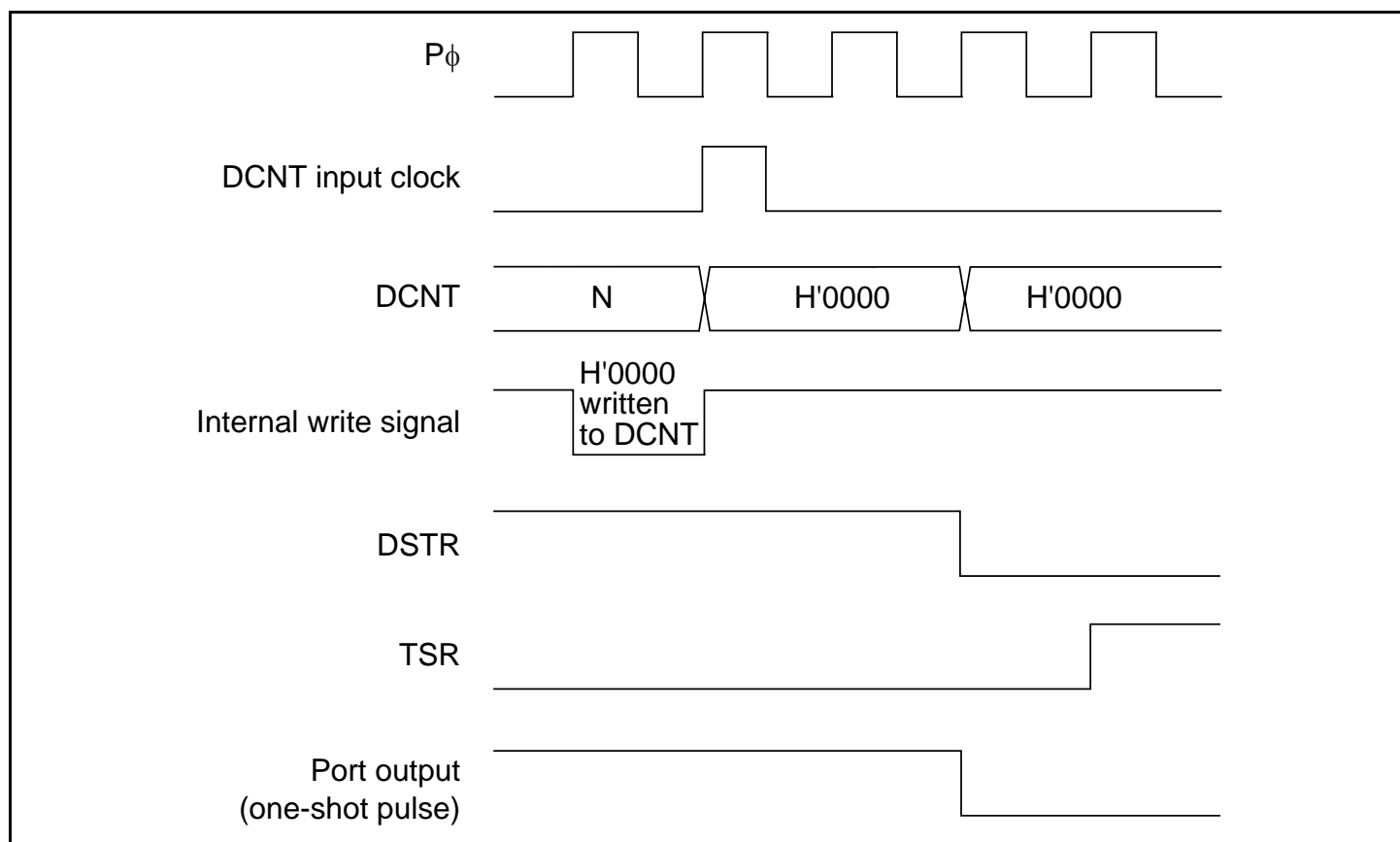


**Figure 11.69 Contention between Interrupt Status Flag Clearing by DMAC and Setting by Input Capture/Compare-Match**

**Halting of a Down-Counter by the CPU:** A down-counter (DCNT) can be halted by writing H'0000 to it. The CPU cannot write 0 directly to the down-count start register (DSTR); instead, by setting DCNT to H'0000, the corresponding DSTR bit is cleared to 0 and the count is stopped. However, the OSF bit in the timer status register (TSR) is set when DCNT underflows.

Note that when H'0000 is written to DCNT, the corresponding DSTR bit is not cleared to 0 immediately; it is cleared to 0, and the down-counter is stopped, when underflow occurs following the H'0000 write.

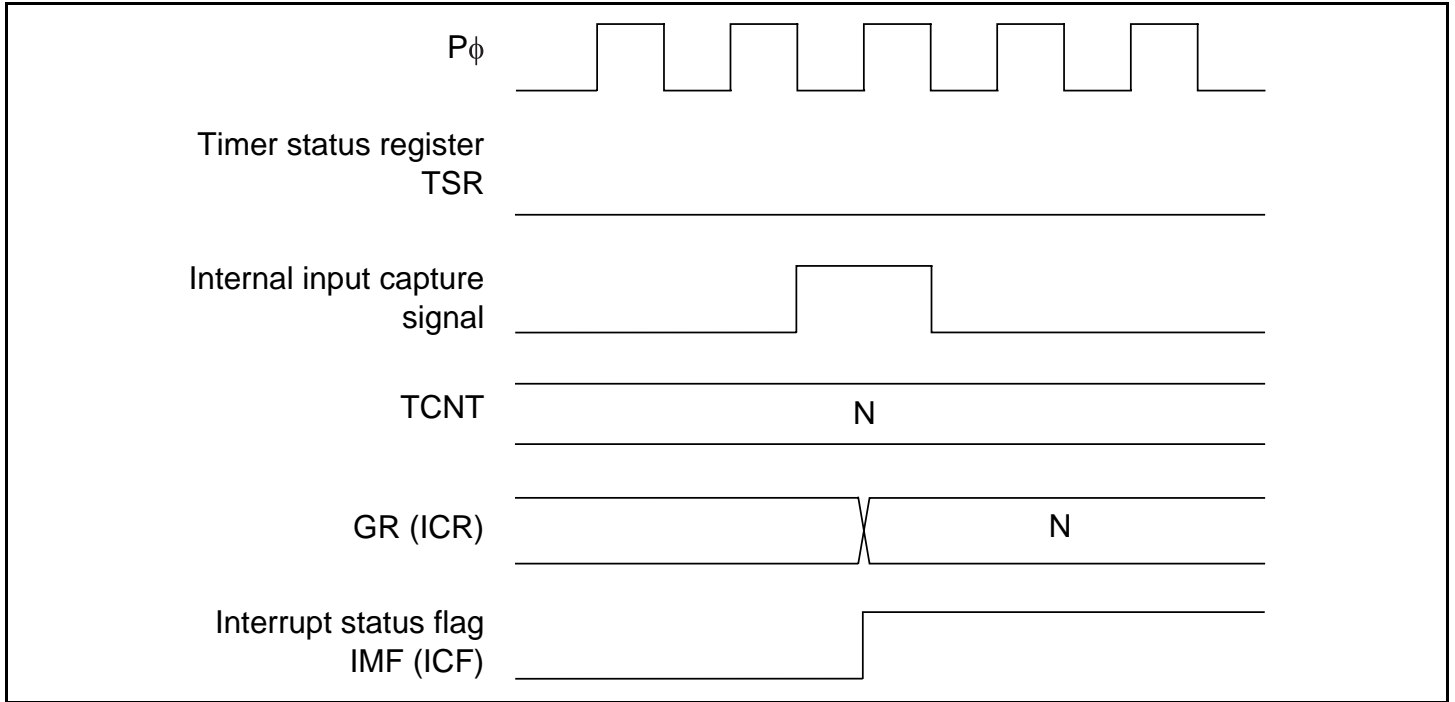
The timing in this case is shown in figure 11.70.



**Figure 11.70 Halting of a Down-Counter by the CPU**

**Input Capture Operation when Free-Running Counter is Halted:** In channels 0 to 5, channel 10, or channel 11, if input capture setting is performed and a trigger signal is input from the input pin, the TCNT value will be transferred to the corresponding general register (GR) or input capture register (ICR) irrespective of whether the free-running counter (TCNT) is running or halted, and the IMF or ICF bit will be set in the timer status register (TSR).

The timing in this case is shown in figure 11.71.

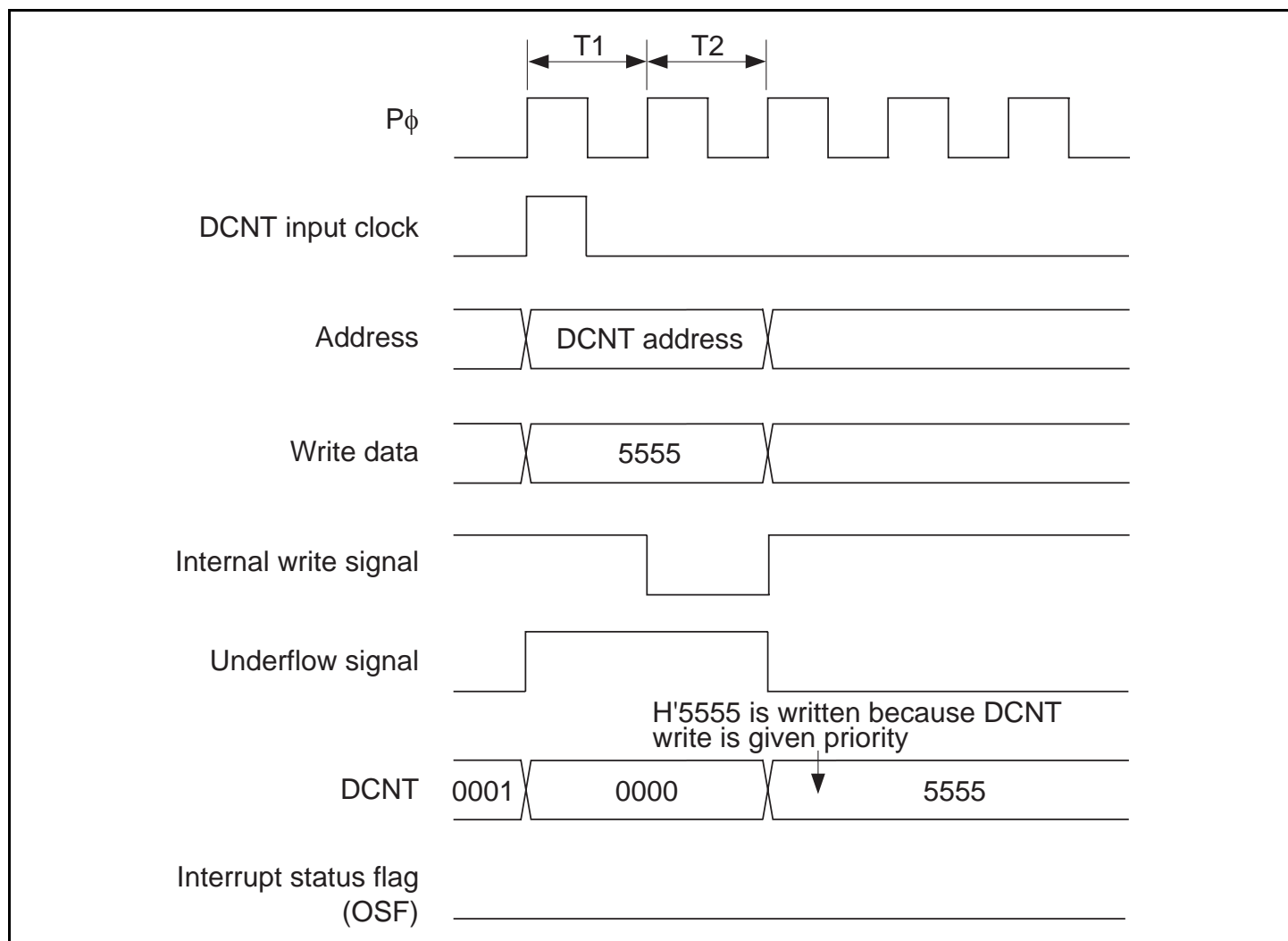


**Figure 11.71 Input Capture Operation before Free-Running Counter is Started**

**Contention between DCNT Write and Counter Clearing by Underflow:** If an underflow occurs in the T2 state of the channel 8 down-counter (DCNT8A to DCNT8P) write cycle by the CPU and the DCNT is stopped, the retention of the H'0000 value has priority and the write to the DCNT by the CPU is not performed. Setting the status flag (OSF) to 1 at the underflow timing is performed in the same way as for a normal underflow.

The timing in this case is shown in figure 11.72. In this example, a write of H'5555 to DCNT is attempted at the same time as DCNT underflows.

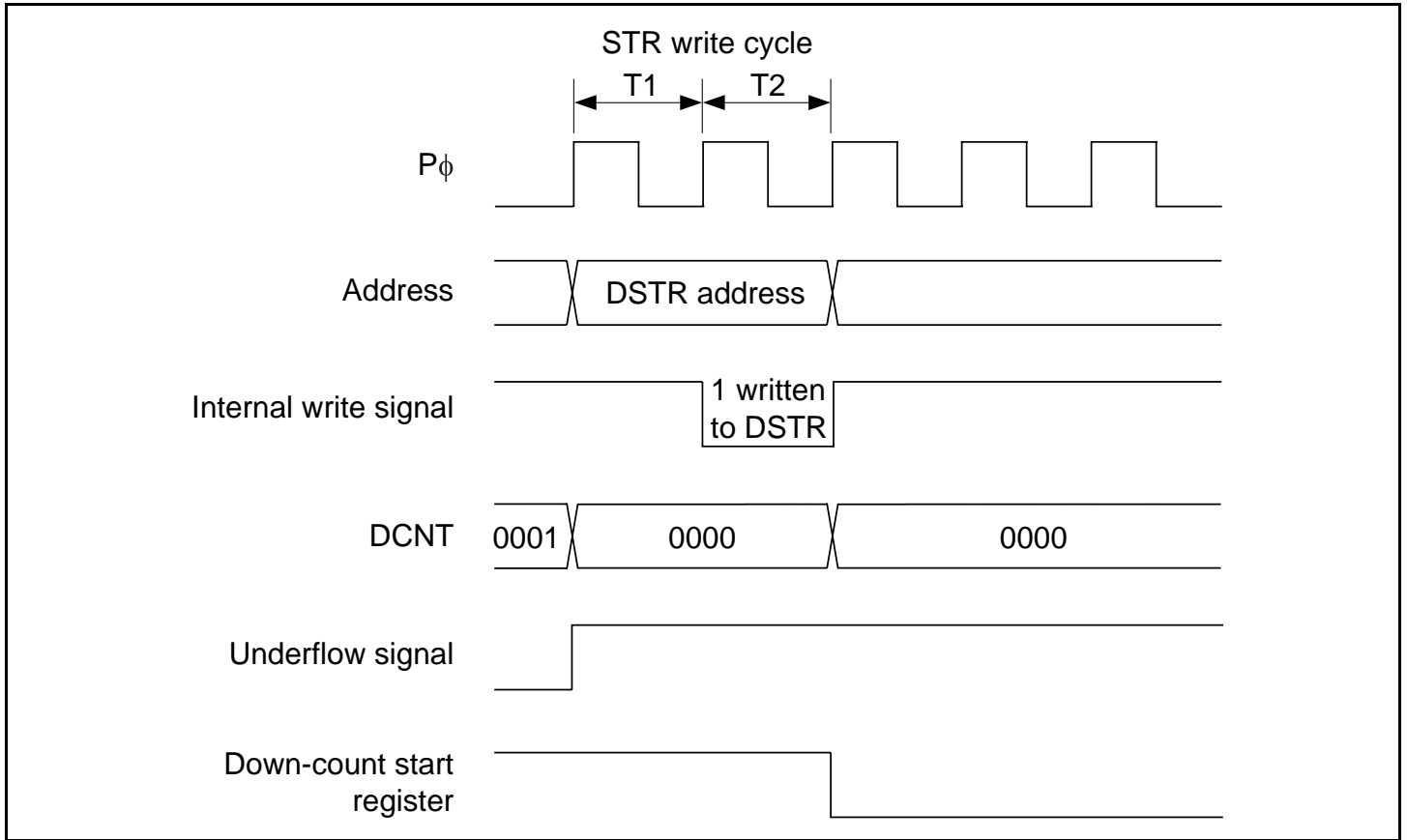
**Note:** In the SH7055, a write to DCNT from the CPU is not attempted, but retention of H'0000 takes precedence. Note that its operation is different.



**Figure 11.72 Contention between DCNT Write and Underflow**

**Contention between DSTR Bit Setting by CPU and Clearing by Underflow:** If underflow occurs in the T2 state of a down-counter start register (DSTR) “1” write cycle by the CPU, clearing to 0 by the underflow has priority, and the corresponding bit of DSTR is not set to 1.

The timing in this case is shown in figure 11.73.



**Figure 11.73 Contention between DSTR Bit Setting by CPU and Clearing by Underflow**

**Timing of Prescaler Register (PSCR), Timer Control Register (TCR), and Timer Mode Register (TMDR) Setting:** Settings in the prescaler register (PSCR), timer control register (TCR), and timer mode register (TMDR) should be made before the counter is started. Operation is not guaranteed if these registers are modified while the counter is running.

Also, the counter must not be started until P $\phi$  has been input 32 times after setting PSCR1 to PSCR4.

**Interrupt Status Flag Clearing Procedure:** When an interrupt status flag is cleared to 0 by the CPU, it must first be read before 0 is written to it. Correct operation cannot be guaranteed if 0 is written without first reading the flag.

**Setting H'0000 in Free-Running Counters 6A to 6D, 7A to 7D (TCNT6A to TCNT6D, TCNT7A to TCNT7D):** If H'0000 is written to a channel 6 and 7 free-running counter (TCNT6A to TCNT6D, TCNT7A to TCNT7D), and the counter is started, the interval up to the first compare-match with the cycle register (CYLR) and duty register (DTR) will be a maximum of one TCNT input clock cycle longer than the set value. With subsequent compare-matches, the correct waveform will be output for the CYLR and DTR values.

**Register Values when a Free-Running Counter (TCNT) Halts:** If the timer start register (TSTR) value is set to 0 during counter operation, only incrementing of the corresponding free-running counter (TCNT) is stopped, and neither the free-running counter (TCNT) nor any other ATU registers are initialized. The external output value at the time TSTR is cleared to 0 will continue to be output.

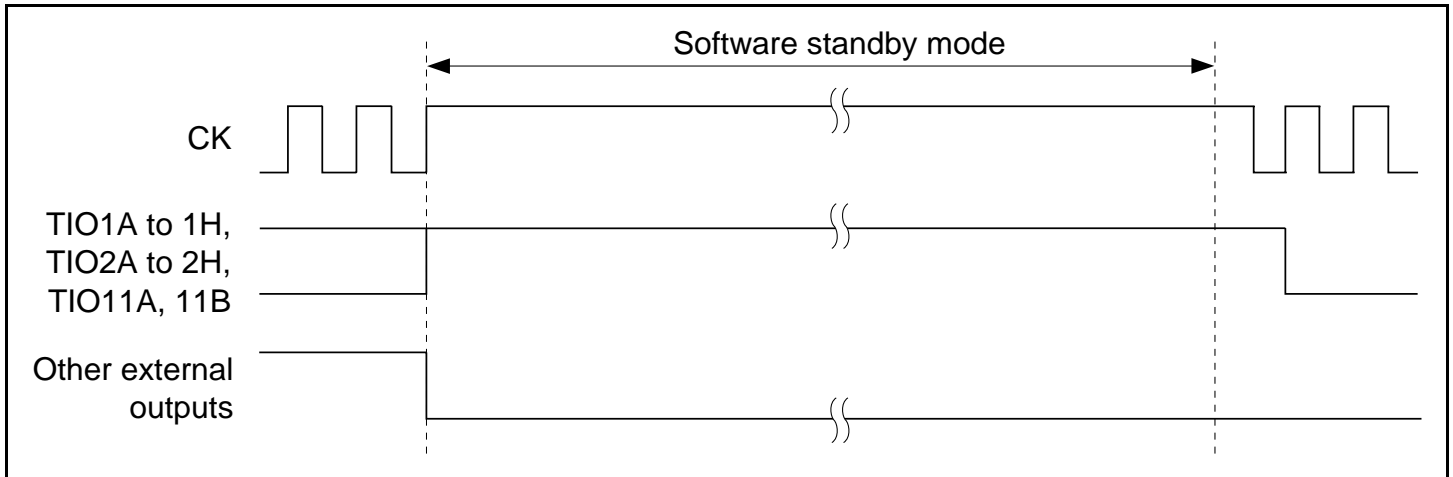
**TCNT0 Writing and Interval Timer Operation:** If the CPU program writes 1 to a bit in free-running counter 0 (TCNT0) corresponding to a bit set to 1 in the interval interrupt request register (ITVRR) when that TCNT0 bit is 0, TCNT0 bit 6, 7, 8, 9, 10, 11, 12, or 13 will be detected as having changed from 0 to 1, and an interrupt request will be sent to INTC and A/D sampling will be started. While the count is halted with the STR0 bit cleared to 0 in timer start register 1 (TSTR1), the bit transition from 0 to 1 will still be detected.

**Automatic TSR Clearing by DMAC Activation by the ATU:** Automatic clearing of TSR is performed after completion of the transfer when the DMAC is in burst mode, and each time the DMAC returns the bus in cycle steal mode.

**Interrupt Status Flag Setting/Resetting:** With TSR, a 0 write to a bit is possible even if overlapping events occur for the same bit before writing 0 after reading 1 to clear that bit. (The duplicate events are not accepted.)

**External Output Value in Software Standby Mode:** In software standby mode, the ATU register and external output values are cleared to 0. However, while the channel 1, 2, and 11 TIO1A to TIO1H, TIO2A to TIO2H, TIO11A, and TIO11B external output values are cleared to 0 immediately after software standby mode is exited, other external output values and all registers are cleared to 0 immediately after a transition to software standby mode.

Also, when pin output is inverted by the pin function controller's port B invert register (PBIR) or port K invert register (PKIR), the corresponding pins are set to 1.



**Figure 11.74 External Output Value Transition Points in Relation to Software Standby Mode**

**Contention between TCNT Clearing from Channel 10 and TCNT Overflow:** When a channel 1 or 2 free-running counter (TCNT1A, TCNT1B, TCNT2A, TCNT2B) overflows, it is cleared to H'0000. If a clear signal from the channel 10 correction counter clear register (TCCLR) is input at the same time, setting 1 to the overflow interrupt status flag (OVF) due to the overflow is still performed in the same way as for a normal overflow.

**Contention between Channel 10 Reload Register Transfer Timing and Write:** If there is contention between a multiplied-output transfer from the input capture register (ICR10A) to the channel 10 reload register (RLDR10C), and the timing of a CPU write to that register, the CPU write has priority and the multiplied output is ignored.

**Contention between Channel 10 Reload Timing and Write to TCNT10C:** If there is contention between a multiplied-output transfer from the input capture register (ICR10A) to the channel 10 reload register (RLDR10C), and a CPU write to the reload counter (TCNT10C), the CPU write has priority and the multiplied output is ignored.



**ATU Pin Setting:** Since input capture or count operation may be occurred when a port is set to the ATU pin function, the following points must be noted.

When using a port for input capture input, the corresponding TIOR register must be in the input capture disabled state when the port is set. Regarding channel 10 TI10 input, TCR10 must be in the TI10 input disabled state when the port is set. When using a port for external clock input, the STR bit for the corresponding channel must be in the count operation disabled state when the port is set. When using a port for event input, the corresponding TCR register must be in the count operation disabled state when the port is set.

Regarding TCLKB and TI10 input, although input is assigned to a number of pins, when using TCLKB and TI10 input, only one pin should be enabled.

**Writing to ROM Area Immediately after ATU Register Write:** If a write cycle for a ROM address for which address bit 11 = 0 and address bit 12 = 1 (H'00001000 to H'000017FF, H'00003000 to H'000037FF, H'00005000 to H'000057FF, ..., H'0007F000 to H'0007F7FF, ..., H'000FF000 to H'000FF7FF) occurs immediately after an ATU register write cycle, the value, or part of the value, written to ROM will be written to the ATU register. The following measures should be taken to prevent this.

- Do not perform a CPU write to a ROM address immediately after an ATU register write cycle. For example, an instruction arrangement in which an MOV instruction that writes to the ATU is located at an even-word address (4n address), and is immediately followed by an MOV instruction that writes to a ROM area, will meet the bug conditions.
- Do not perform an AUD write to any of the above ROM addresses immediately after an ATU register write cycle. For example, in the case of a write to overlap RAM when using the RAM emulation function, the write should be performed to the on-chip RAM area address, not the overlapping ROM area address.
- Do not perform a DMAC write to an ATU register when a ROM address write operation occurs.

# 11.8 ATU-II Registers and Pins

**Table 11.4 ATU-II Registers and Pins**

Register Name* <sup>1</sup>	Channel											
	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11
TSTR (3)	TSTR1	TSTR1	TSTR1	TSTR1	TSTR1	TSTR1	TSTR2	TSTR2	–	–	TSTR1	TSTR3
PSCR (4)	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1	PSCR1	PSCR2	PSCR3	PSCR1	–	PSCR4	PSCR1
TCNT (25)	TCNT0H, TCNT0L	TCNT1A, TCNT1B	TCNT2A, TCNT2B	TCNT3	TCNT4	TCNT5	TCNT6A to TCNT6D	TCNT7A to TCNT7D	–	–	TCNT10AH, TCNT10AL, TCNT10B to TCNT10H	TCNT11
DCNT (16)	–	–	–	–	–	–	–	–	–	DCNT8A to DCNT8P	–	–
ECNT (6)	–	–	–	–	–	–	–	–	–	ECNT9A to ECNT9F	–	–
TCR (17)	–	TCR1A, TCR1B	TCR2A, TCR2B	TCR3	TCR4	TCR5	TCR6A, TCR6B	TCR7A, TCR7B	TCR8	TCR9A to TCR9C	TCR10	TCR11
TIOR (17)	TIOR0	TIOR1A to TIOR1D	TIOR2A to TIOR2D	TIOR3A, TIOR3B	TIOR4A, TIOR4B	TIOR5A, TIOR5B	–	–	–	–	TIOR10	TIOR11
TSR (12)	TSR0	TSR1A, TSR1B	TSR2A, TSR2B	TSR3	TSR3	TSR3	TSR6	TSR7	TSR8	TSR9	TSR10	TSR11
TIER (12)	TIER0	TIER1A, TIER1B	TIER2A, TIER2B	TIER3	TIER3	TIER3	TIER6	TIER7	TIER8	TIER9	TIER10	TIER11
ITVRR (3)	ITVRR1, ITVRR2A, ITVRR2B	–	–	–	–	–	–	–	–	–	–	–
GR (37)	–	GR1A to GR1H	GR2A to GR2H	GR3A to GR3D	GR4A to GR4D	GR5A to GR5D	–	–	–	GR9A to GR9F	GR10G	GR11A, GR11B
ICR (5)	ICR0AH, ICR0AL to ICR0DH, ICR0DL	–	–	–	–	–	–	–	–	–	ICR10AH, ICR10AL	–
OCR (11)	–	OCR1	OCR2A to OCR2H	–	–	–	–	–	–	–	OCR10AH, OCR10AL, OCR10B	–
OSBR (2)	–	OSBR1	OSBR2	–	–	–	–	–	–	–	–	–
TRGMDR (1)	–	TRGMDR	–	–	–	–	–	–	–	–	–	–
TMDR (1)	–	–	–	TMDR	TMDR	TMDR	–	–	–	–	–	–

**Table 11.4 ATU-II Registers and Pins (cont)**

Register Name* <sup>1</sup>	Channel											
	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7	Channel 8	Channel 9	Channel 10	Channel 11
CYLR (8)	–	–	–	–	–	–	CYLR6A to CYLR6D	CYLR7A to CYLR7D	–	–	–	–
BFR (8)	–	–	–	–	–	–	BFR6A to BFR6D	BFR7A to BFR7D	–	–	–	–
DTR (8)	–	–	–	–	–	–	DTR6A to DTR6D	DTR7A to DTR7D	–	–	–	–
PMDR (1)	–	–	–	–	–	–	PMDR	–	–	–	–	–
RLDR (1)	–	–	–	–	–	–	–	–	RLDR	–	–	–
TCNR (1)	–	–	–	–	–	–	–	–	TCNR	–	–	–
OTR (1)	–	–	–	–	–	–	–	–	OTR	–	–	–
DSTR (1)	–	–	–	–	–	–	–	–	DSTR	–	–	–
RLDENR (1)	–	–	–	–	–	–	–	–	RLDENR	–	–	–
RLD (1)	–	–	–	–	–	–	–	–	–	–	RLD10C	–
NCR (1)	–	–	–	–	–	–	–	–	–	–	NCR10	–
TCCLR (1)	–	–	–	–	–	–	–	–	–	–	TCCLR10	–
Pins* <sup>2</sup>	TIOA to D	TIO1A to H, TCLKA, TCLKB	TIO2A to H, TCLKA, TCLKB	TIO3A to D, TCLKA, TCLKB	TIO4A to D, TCLKA, TCLKB	TIO5A to D, TCLKA, TCLKB	TO6A to D	TO7A to D	TO8A to P	TI9A to F	T10	TIO11A, TIO11B, TCLKA, TCLKB

Notes: 1. Figures in parentheses show the number of registers. A 32-bit register is shown as a single register.

2. Pin functions should be set as described in section 21, Pin Function Controller (PFC).

# Section 12 Advanced Pulse Controller (APC)

## 12.1 Overview

The SH7058 has an on-chip advanced pulse controller (APC) that can generate a maximum of eight pulse outputs, using the advanced timer unit II (ATU-II) as the time base.

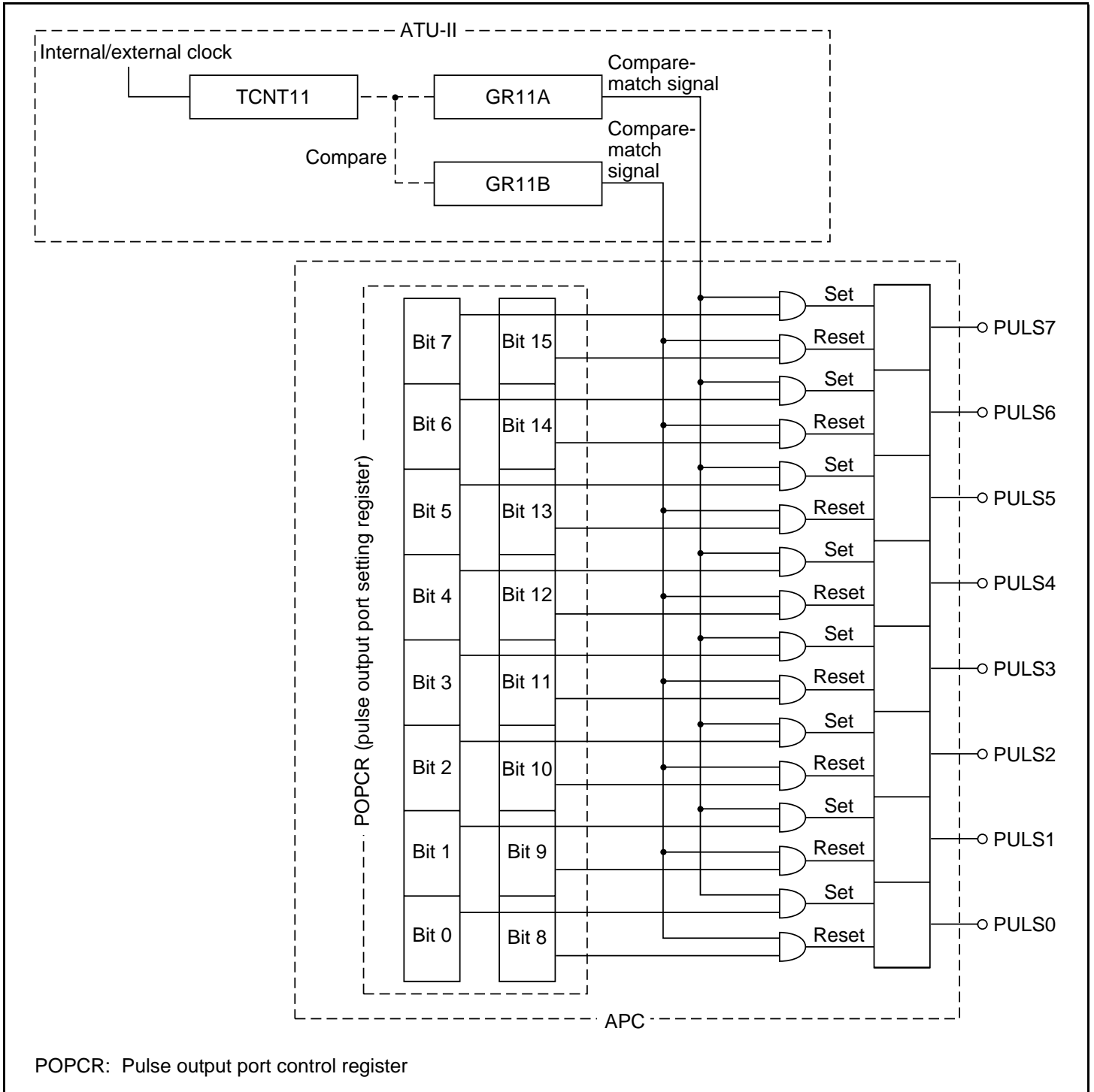
### 12.1.1 Features

The features of the APC are summarized below.

- Maximum eight pulse outputs  
The pulse output pins can be selected from among eight pins. Multiple settings are possible.
- Output trigger provided by advanced timer unit II (ATU-II) channel 11  
Pulse 0 output and 1 output is performed using the compare-match signal generated by the ATU-II channel 11 compare-match register as the trigger.

## 12.1.2 Block Diagram

Figure 12.1 shows a block diagram of the advanced pulse controller.



**Figure 12.1 Advanced Pulse Controller Block Diagram**

### 12.1.3 Pin Configuration

Table 12.1 summarizes the advanced pulse controller's output pins.

**Table 12.1 Advanced Pulse Controller Pins**

Pin Name	I/O	Function
PULS0	Output	APC pulse output 0
PULS1	Output	APC pulse output 1
PULS2	Output	APC pulse output 2
PULS3	Output	APC pulse output 3
PULS4	Output	APC pulse output 4
PULS5	Output	APC pulse output 5
PULS6	Output	APC pulse output 6
PULS7	Output	APC pulse output 7

### 12.1.4 Register Configuration

Table 12.2 summarizes the advanced pulse controller's register.

**Table 12.2 Advanced Pulse Controller Register**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Pulse output port control register	POPCR	R/W	H'0000	H'FFFFFF700	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 12.2 Register Descriptions

### 12.2.1 Pulse Output Port Control Register (POPCR)

The pulse output port control register (POPCR) is a 16-bit readable/writable register.

POPCR is initialized to H'0000 by a power-on reset and in hardware standby mode. It is not initialized in software standby mode.

Bit:	15	14	13	12	11	10	9	8
	PULS7 ROE	PULS6 ROE	PULS5 ROE	PULS4 ROE	PULS3 ROE	PULS2 ROE	PULS1 ROE	PULS0 ROE
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:	7	6	5	4	3	2	1	0
	PULS7 SOE	PULS6 SOE	PULS5 SOE	PULS4 SOE	PULS3 SOE	PULS2 SOE	PULS1 SOE	PULS0 SOE
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

- Bits 15 to 8—PULS7 to PULS0 Reset Output Enable (PULS7ROE to PULS0ROE): These bits enable or disable 0 output to the APC pulse output pins (PULS7 to PULS0) bit by bit.

#### Bits 15 to 8:

#### PULS7ROE to PULS0ROE Description

0	0 output to APC pulse output pin (PULS7—PULS0) is disabled (Initial value)
1	0 output to APC pulse output pin (PULS7—PULS0) is enabled

When one of these bits is set to 1, 0 is output from the corresponding pin on a compare-match between the GR11B and TCNT11 values.

- Bits 7 to 0—PULS7 to PULS0 Set Output Enable (PULS7SOE to PULS0SOE): These bits enable or disable 1 output to the APC pulse output pins (PULS7 to PULS0) bit by bit.

#### Bits 7 to 0:

#### PULS7SOE to PULS0SOE Description

0	1 output to APC pulse output pin (PULS7—PULS0) is disabled (Initial value)
1	1 output to APC pulse output pin (PULS7—PULS0) is enabled

When one of these bits is set to 1, 1 is output from the corresponding pin on a compare-match between the GR11A and TCNT11 values.

## 12.3 Operation

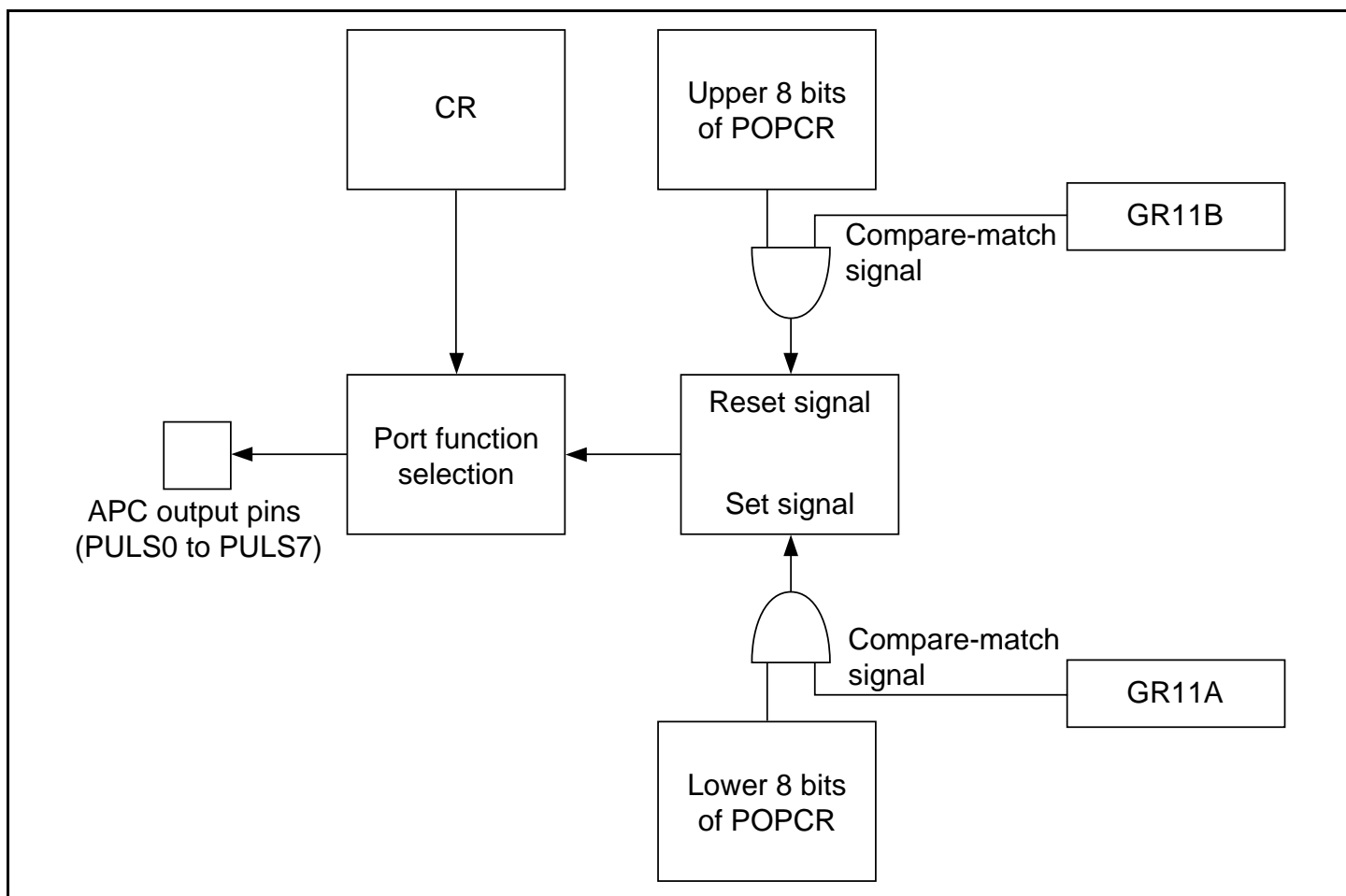
### 12.3.1 Overview

APC pulse output is enabled by designating multiplex pins for APC pulse output with the pin function controller (PFC), and setting the corresponding bits to 1 in the pulse output port control register (POPCR).

When general register 11A (GR11A) in the advanced timer unit II (ATU-II) subsequently generates a compare-match signal, 1 is output from the pins set to 1 by bits 7 to 0 in POPCR. When general register 11B (GR11B) generates a compare-match signal, 0 is output from the pins set to 1 by bits 15 to 8 in POPCR.

0 is output from the output-enabled state until the first compare-match occurs.

The advanced pulse controller output operation is shown in figure 12.2.



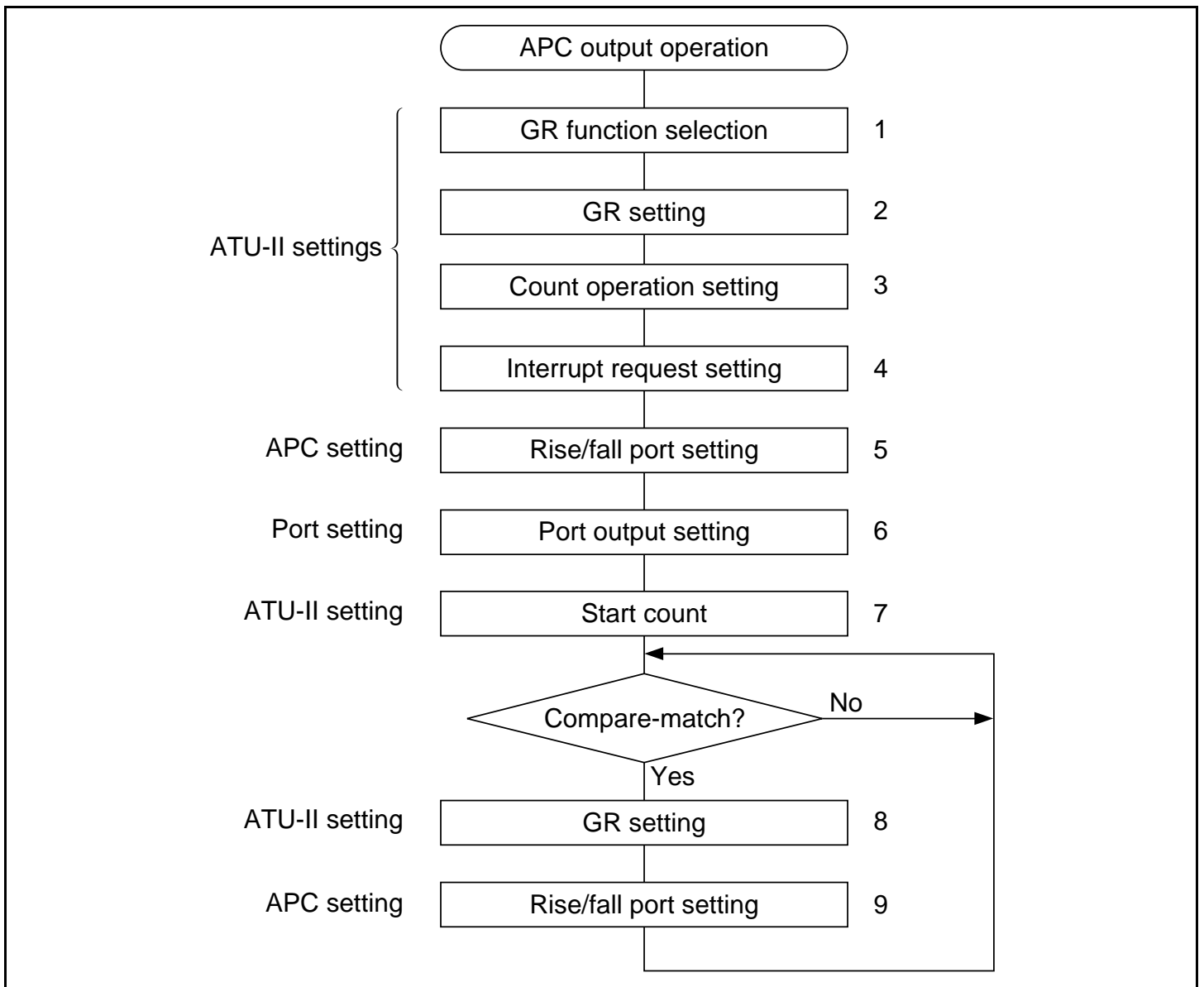
**Figure 12.2 Advanced Pulse Controller Output Operation**



## 12.3.2 Advanced Pulse Controller Output Operation

**Example of Setting Procedure for Advanced Pulse Controller Output Operation:** Figure 12.3 shows an example of the setting procedure for advanced pulse controller output operation.

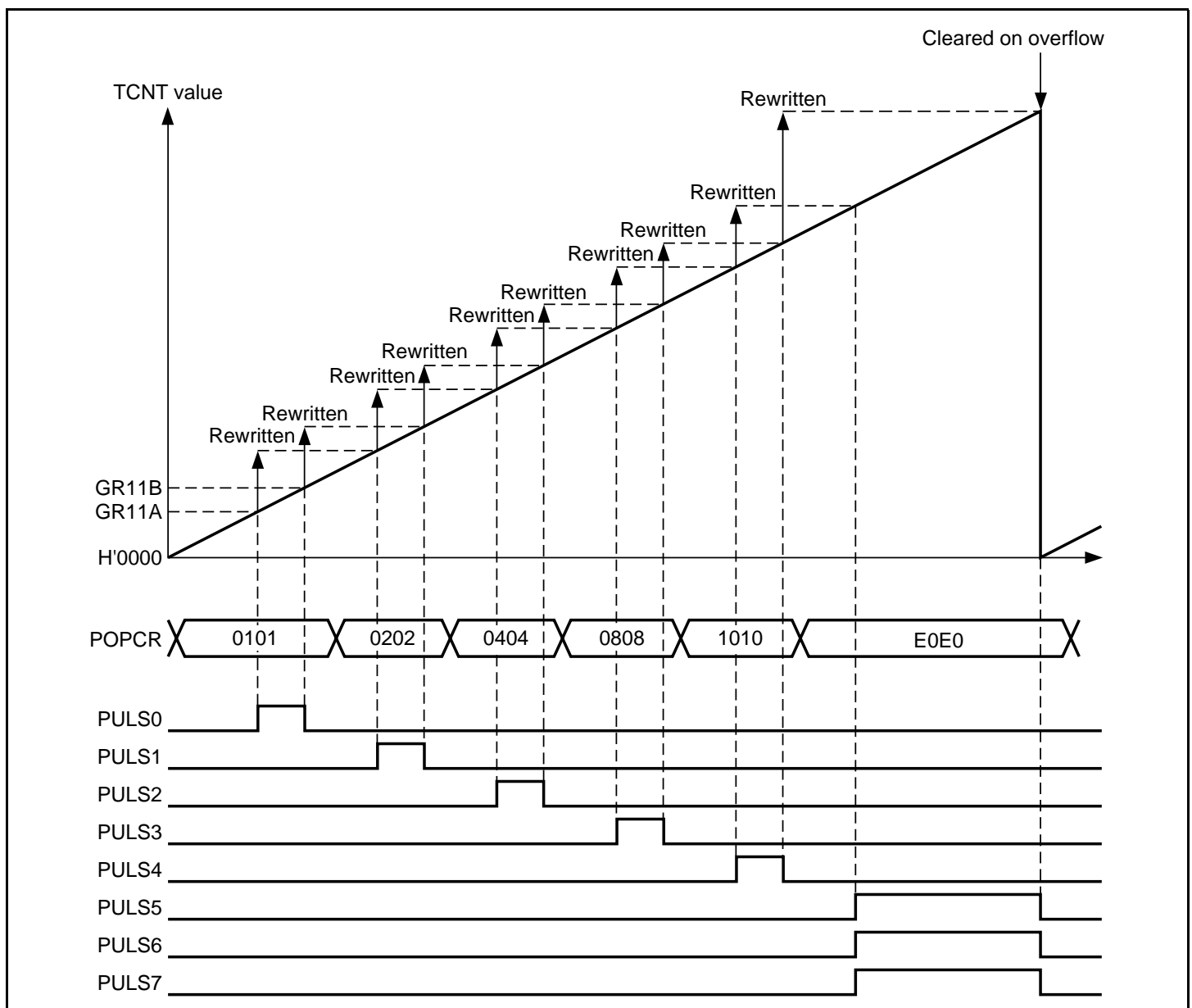
1. Set general registers GR11A and GR11B as output compare registers with the timer I/O control register (TIOR).
2. Set the pulse rise point with GR11A and the pulse fall point with GR11B.
3. Select the timer counter 11 (TCNT11) counter clock with the timer prescale register (PSCR). TCNT11 can only be cleared by an overflow.
4. Enable the respective interrupts with the timer interrupt enable register (TIER).
5. Set the pins for 1 output and 0 output with POPCR.
6. Set the control register for the port to be used by the APC to the APC output pin function.
7. Set the STR bit to 1 in the timer start register (TSTR) to start timer counter 11 (TCNT11).
8. Each time a compare-match interrupt is generated, update the GR value and set the next pulse output time.
9. Each time a compare-match interrupt is generated, update the POPCR value and set the next pin for pulse output.



**Figure 12.3 Example of Setting Procedure for Advanced Pulse Controller Output Operation**

**Example of Advanced Pulse Controller Output Operation:** Figure 12.4 shows an example of advanced pulse controller output operation.

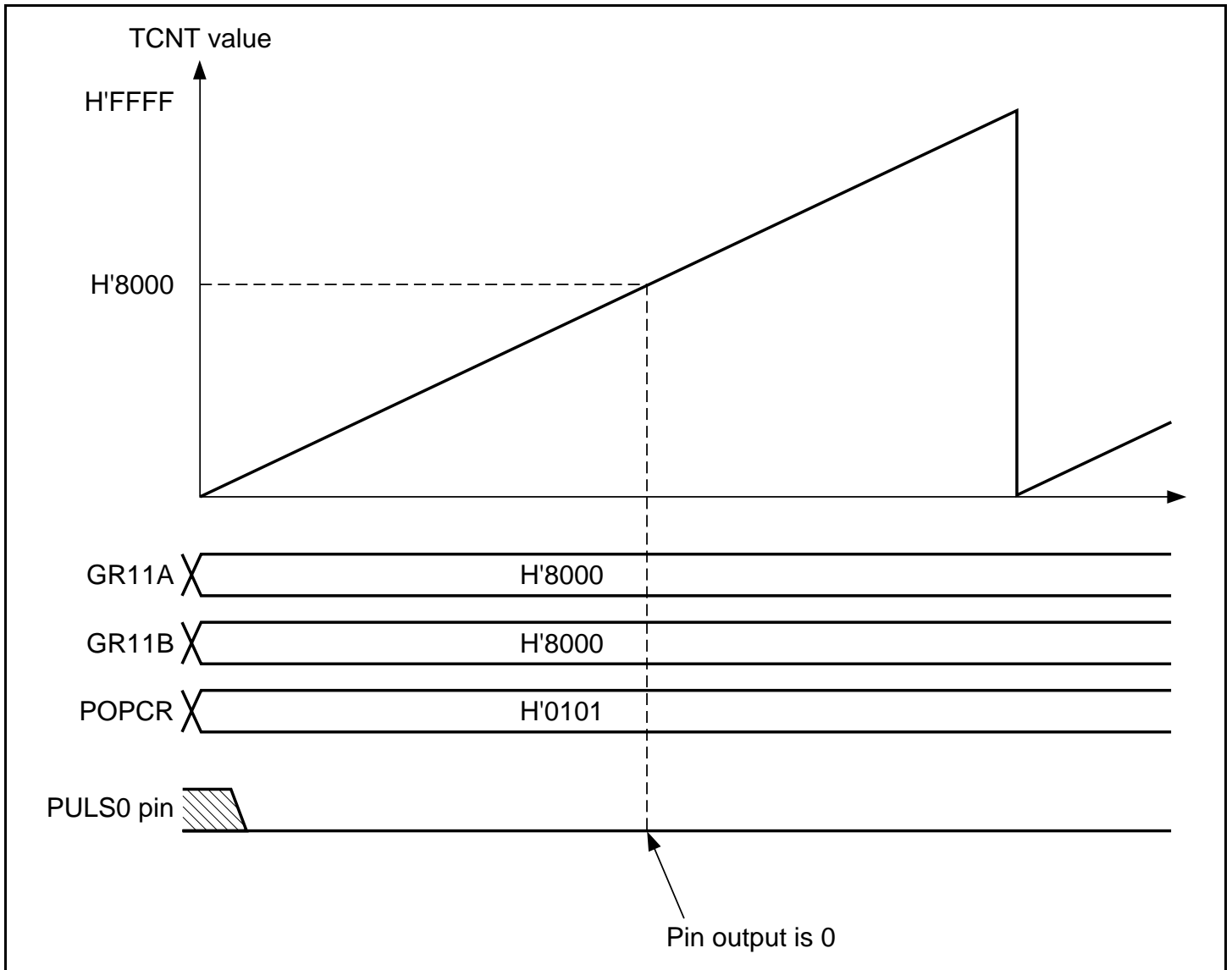
1. Set ATU-II registers GR11A and GR11B (to be used for output trigger generation) as output compare registers. Set the rise point in GR11A and the fall point in GR11B, and enable the respective compare-match interrupts.
2. Write H'0101 to POPCR.
3. Start the TCNT11 count, when a GR11A compare-match occurs, 1 is output from the PULS0 pin. When a GR11B compare-match occurs, 0 is output from the PULS0 pin.
4. Pulse widths and output pins can be continually changed by successively rewriting GR11A, GR11B, and POPCR in response to compare-match interrupts.
5. By setting POPCR to a value such as H'E0E0, pulses can be output from up to eight pins in response to a single compare-match.



**Figure 12.4 Example of Advanced Pulse Controller Output Operation**

## 12.4 Usage Notes

**Contention between Compare-Match Signals:** If the same value is set for both GR11A and GR11B, and 0 output and 1 output are both enabled for the same pin by the POPCR settings, 0 output has priority on pins PULS0 to PULS7 when compare-matches occur.



**Figure 12.5 Example of Compare-Match Contention**



# Section 13 Watchdog Timer (WDT)

## 13.1 Overview

The watchdog timer (WDT) is a 1-channel timer for monitoring system operations. If a system encounters a problem (crashes, for example) and the timer counter overflows without being rewritten correctly by the CPU, an overflow signal ( $\overline{\text{WDTOVF}}$ ) is output externally. The WDT can simultaneously generate an internal reset signal for the entire chip.

When the 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.

### 13.1.1 Features

The WDT has the following features:

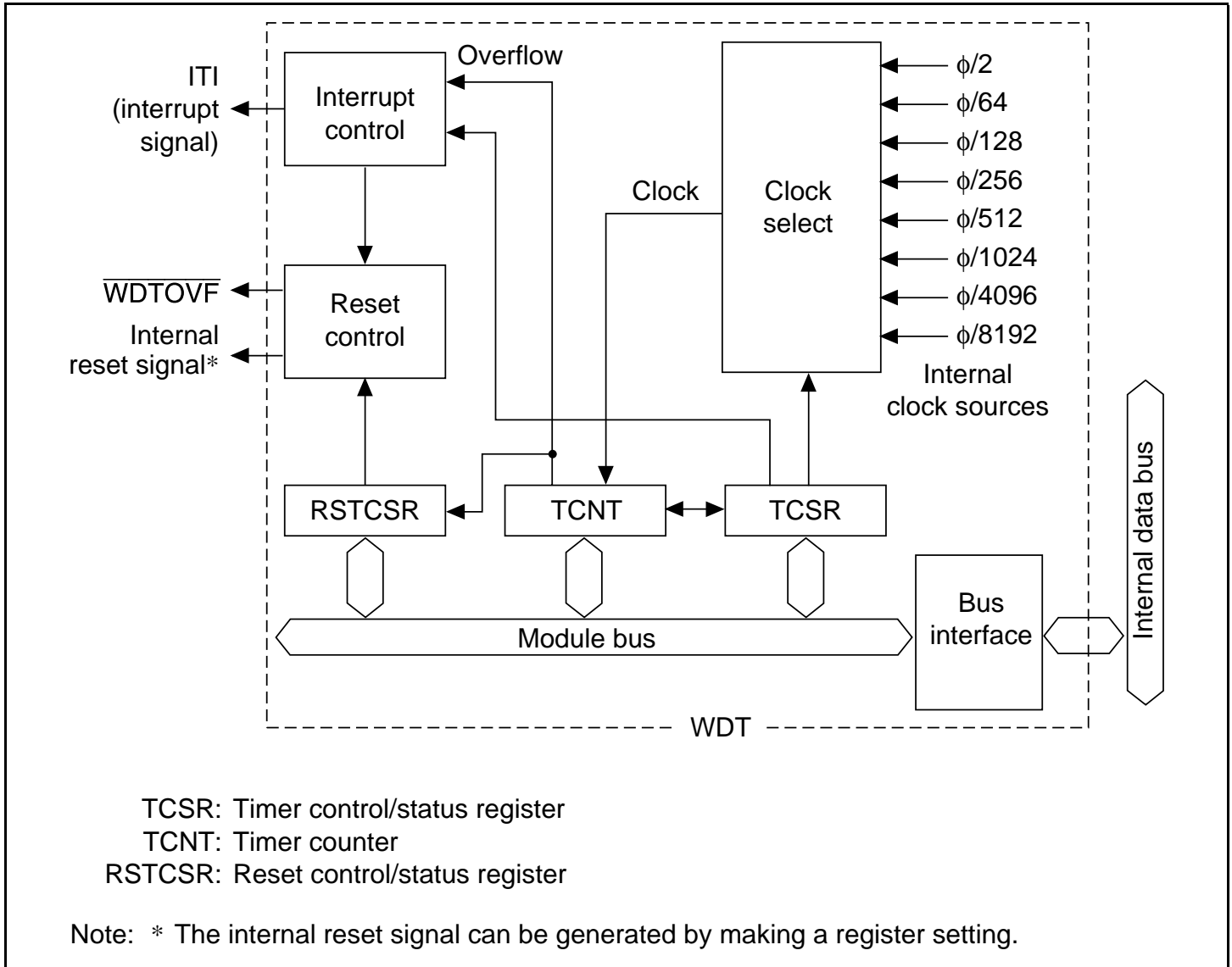
- Works in watchdog timer mode or interval timer mode
- Outputs  $\overline{\text{WDTOVF}}$  in watchdog timer mode

When the counter overflows in watchdog timer mode, overflow signal  $\overline{\text{WDTOVF}}$  is output externally. It is possible to select whether to reset the chip internally when this happens. Either the power-on reset or manual reset signal can be selected as the internal reset signal.

- Generates interrupts in interval timer mode  
When the counter overflows, it generates an interval timer interrupt.
- Works with eight counter input clocks

### 13.1.2 Block Diagram

Figure 13.1 is the block diagram of the WDT.



**Figure 13.1 WDT Block Diagram**

### 13.1.3 Pin Configuration

Table 13.1 shows the pin configuration.

**Table 13.1 Pin Configuration**

Pin	Abbreviation	I/O	Function
Watchdog timer overflow	WDTOVF	O	Outputs the counter overflow signal in watchdog timer mode

## 13.1.4 Register Configuration

Table 13.2 summarizes the three WDT registers. They are used to select the clock, switch the WDT mode, and control the reset signal.

**Table 13.2 WDT Registers**

Name	Abbreviation	R/W	Initial Value	Address	
				Write* <sup>1</sup>	Read* <sup>2</sup>
Timer control/status register	TCSR	R/(W)* <sup>3</sup>	H'18	H'FFFFFFEC10	H'FFFFFFEC10
Timer counter	TCNT	R/W	H'00		H'FFFFFFEC11
Reset control/status register	RSTCSR	R/(W)* <sup>3</sup>	H'1F	H'FFFFFFEC12	H'FFFFFFEC13

Notes: In register access, four cycles are required for both byte access and word access.

1. Write by word transfer. These registers cannot be written in bytes or longwords.
2. Read by byte transfer. These registers cannot be read in words or longwords.
3. Only 0 can be written to bit 7 to clear the flag.

## 13.2 Register Descriptions

### 13.2.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable upcounter. (TCNT differs from other registers in that it is more difficult to write to. See section 13.2.4, Register Access, for details.) When the timer enable bit (TME) in the timer control/status register (TCSR) is set to 1, the watchdog timer counter starts counting pulses of an internal clock selected by clock select bits 2 to 0 (CKS2 to CKS0) in TCSR. When the value of TCNT overflows (changes from H'FF to H'00), a watchdog timer overflow signal ( $\overline{\text{WDTOVF}}$ ) or interval timer interrupt (ITI) is generated, depending on the mode selected in the  $\overline{\text{WT/IT}}$  bit in TCSR.

TCNT is initialized to H'00 by a power-on reset, in hardware and software standby modes, and when the TME bit is cleared to 0.

Bit:	7	6	5	4	3	2	1	0									
	<table border="1" style="width: 100%; height: 20px;"> <tr> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> <td style="width: 12.5%;"></td> </tr> </table>																
Initial value:	0	0	0	0	0	0	0	0									
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W									



## 13.2.2 Timer Control/Status Register (TCSR)

The timer control/status register (TCSR) is an 8-bit readable/writable register. (TCSR differs from other registers in that it is more difficult to write to. See section 13.2.4, Register Access, for details.) TCSR performs selection of the timer counter (TCNT) input clock and mode.

TCSR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	OVF	WT/ $\overline{\text{IT}}$	TME	–	–	CKS2	CKS1	CKS0
Initial value:	0	0	0	1	1	0	0	0
R/W:	R/(W)*	R/W	R/W	R	R	R/W	R/W	R/W

Note: \* The only operation permitted on the OVF bit is a write of 0 after reading 1.

- Bit 7—Overflow Flag (OVF): Indicates that TCNT has overflowed from H'FF to H'00 in interval timer mode. This flag is not set in the watchdog timer mode.

Bit 7: OVF	Description
0	No overflow of TCNT in interval timer mode (Initial value) [Clearing condition] When 0 is written to OVF after reading OVF
1	TCNT overflow in interval timer mode

- Bit 6—Timer Mode Select (WT/ $\overline{\text{IT}}$ ): Selects whether to use the WDT as a watchdog timer or interval timer. When TCNT overflows, the WDT either generates an interval timer interrupt (ITI) or generates a  $\overline{\text{WDTOVF}}$  signal, depending on the mode selected.

Bit 6: WT/ $\overline{\text{IT}}$	Description
0	Interval timer mode: interval timer interrupt (ITI) request to the CPU when TCNT overflows (Initial value)
1	Watchdog timer mode: $\overline{\text{WDTOVF}}$ signal output externally when TCNT overflows. (Section 13.2.3, Reset Control/Status Register (RSTCSR), describes in detail what happens when TCNT overflows in watchdog timer mode.)

- Bit 5—Timer Enable (TME): Enables or disables the timer.

Bit 5: TME	Description
0	Timer disabled: TCNT is initialized to H'00 and count-up stops (Initial value)
1	Timer enabled: TCNT starts counting. A $\overline{\text{WDTOVF}}$ signal or interrupt is generated when TCNT overflows.

- Bits 4 and 3—Reserved: These bits are always read as 1. The write value should always be 1.
- Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight internal clock sources for input to TCNT. The clock signals are obtained by dividing the frequency of the system clock ( $\phi$ ).

			Description	
Bit 2: CKS2	Bit 1: CKS1	Bit 0: CKS0	Clock Source	Overflow Interval* ( $\phi = 40 \text{ MHz}$ )
0	0	0	$\phi/2$ (Initial value)	12.8 $\mu\text{s}$
0	0	1	$\phi/64$	409.6 $\mu\text{s}$
0	1	0	$\phi/128$	0.8 ms
0	1	1	$\phi/256$	1.6 ms
1	0	0	$\phi/512$	3.3 ms
1	0	1	$\phi/1024$	6.6 ms
1	1	0	$\phi/4096$	26.2 ms
1	1	1	$\phi/8192$	52.4 ms

Note: \* The overflow interval listed is the time from when the TCNT begins counting at H'00 until an overflow occurs. Refer to section 13.4.7, Multiplication Factor for Internal Clock Signal ( $\phi$ ) and Overflow Time.

### 13.2.3 Reset Control/Status Register (RSTCSR)

RSTCSR is an 8-bit readable/writable register. (RSTCSR differs from other registers in that it is more difficult to write. See section 13.2.4, Register Access, for details.) It controls output of the internal reset signal generated by timer counter (TCNT) overflow. RSTCSR is initialized to H'1F by input of a reset signal from the  $\overline{\text{RES}}$  pin, but is not initialized by the internal reset signal generated by overflow of the WDT. It is initialized to H'1F in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	WOVF	RSTE	RSTS	–	–	–	–	–
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/(W)*	R/W	R/W	R	R	R	R	R

Note: \* Only 0 can be written to bit 7 to clear the flag.

- Bit 7—Watchdog Timer Overflow Flag (WOVF): Indicates that TCNT has overflowed (H'FF to H'00) in watchdog timer mode. This flag is not set in interval timer mode.

Bit 7: WOVF	Description
0	No TCNT overflow in watchdog timer mode (Initial value) [Clearing condition] When 0 is written to WOVF after reading WOVF
1	Set by TCNT overflow in watchdog timer mode

- Bit 6—Reset Enable (RSTE): Selects whether to reset the chip internally if TCNT overflows in watchdog timer mode.

Bit 6: RSTE	Description
0	Not reset when TCNT overflows (Initial value) LSI not reset internally, but TCNT and TCSR reset within WDT.
1	Reset when TCNT overflows

- Bit 5—Reset Select (RSTS): Selects the kind of internal reset to be generated when TCNT overflows in watchdog timer mode.

Bit 5: RSTS	Description
0	Power-on reset (Initial value)
1	Manual reset

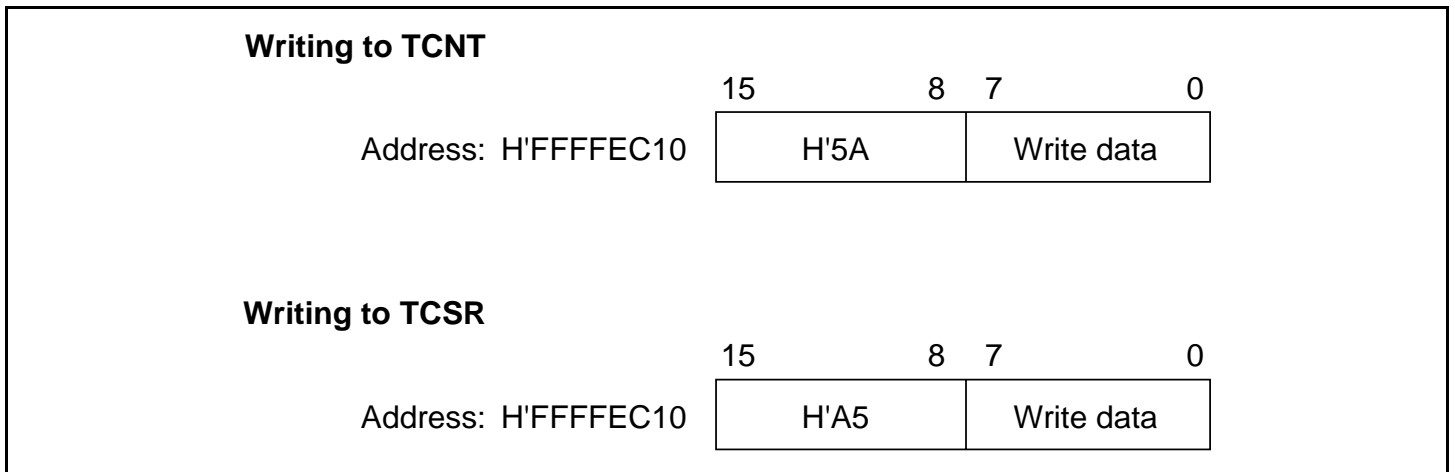
- Bits 4 to 0—Reserved: These bits are always read as 1. The write value should always be 1.

### 13.2.4 Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in that they are more difficult to write to. The procedures for writing and reading these registers are given below.

**Writing to TCNT and TCSR:** These registers must be written by a word transfer instruction. They cannot be written by byte transfer instructions.

TCNT and TCSR both have the same write address. The write data must be contained in the lower byte of the written word. The upper byte must be H'5A (for TCNT) or H'A5 (for TCSR) (figure 13.2). This transfers the write data from the lower byte to TCNT or TCSR.

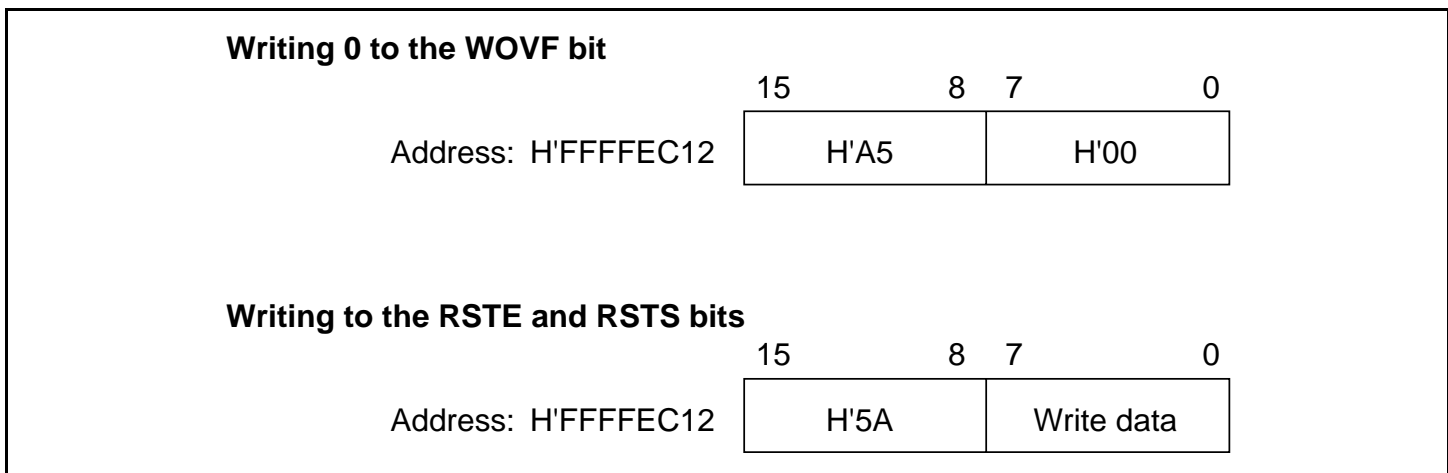


**Figure 13.2 Writing to TCNT and TCSR**

**Writing to RSTCSR:** RSTCSR must be written by a word access to address H'FFFFEC12. It cannot be written by byte transfer instructions.

Procedures for writing 0 to WOVF (bit 7) and for writing to RSTE (bit 6) and RSTS (bit 5) are different, as shown in figure 13.3.

To write 0 to the WOVF bit, the write data must be H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0. The RSTE and RSTS bits are not affected. To write to the RSTE and RSTS bits, the upper byte must be H'5A and the lower byte must be the write data. The values of bits 6 and 5 of the lower byte are transferred to the RSTE and RSTS bits, respectively. The WOVF bit is not affected.



**Figure 13.3 Writing to RSTCSR**

**Reading from TCNT, TCSR, and RSTCSR:** TCNT, TCSR, and RSTCSR are read like other registers. Use byte transfer instructions. The read addresses are H'FFFFEC10 for TCSR, H'FFFFEC11 for TCNT, and H'FFFFEC13 for RSTCSR.

## 13.3 Operation

### 13.3.1 Watchdog Timer Mode

To use the WDT as a watchdog timer, set the  $\overline{WT/IT}$  and TME bits in TCSR to 1. Software must prevent TCNT overflow by rewriting the TCNT value (normally by writing H'00) before overflow occurs. No TCNT overflows will occur while the system is operating normally, but if TCNT fails to be rewritten and overflows occur due to a system crash or the like, a  $\overline{WDTOVF}$  signal is output externally (figure 13.4). The  $\overline{WDTOVF}$  signal can be used to reset the system. The  $\overline{WDTOVF}$  signal is output for 128  $\phi$  clock cycles.

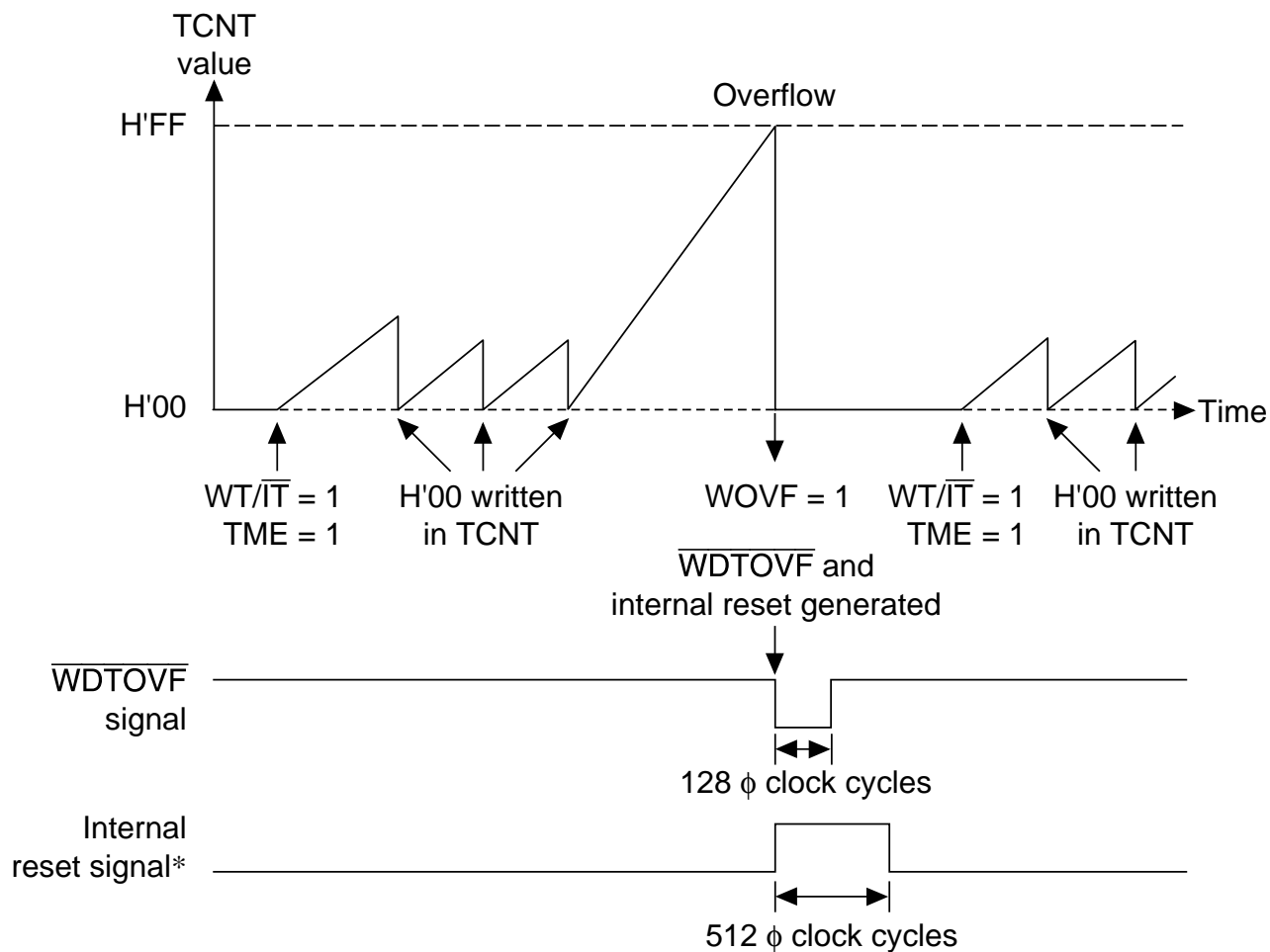
If the RSTE bit in RSTCSR is set to 1, a signal to reset the chip will be generated internally simultaneous with the  $\overline{WDTOVF}$  signal when TCNT overflows. Either a power-on reset or a manual reset can be selected by the RSTS bit in RSTCSR. The internal reset signal is output for 512  $\phi$  clock cycles.

When a WDT overflow reset is generated simultaneously with a reset input at the  $\overline{RES}$  pin, the  $\overline{RES}$  reset takes priority, and the WOVF bit in RSTCSR is cleared to 0.

The following registers are not initialized by a WDT reset signal:

- PFC (pin function controller) registers
- I/O port registers

These registers are initialized only by an external power-on reset.



**Figure 13.4 Operation in Watchdog Timer Mode**

### 13.3.2 Interval Timer Mode

To use the WDT as an interval timer, clear  $\overline{WT/IT}$  to 0 and set TME to 1 in TCSR. An interval timer interrupt (ITI) is generated each time the timer counter overflows. This function can be used to generate interval timer interrupts at regular intervals (figure 13.5).

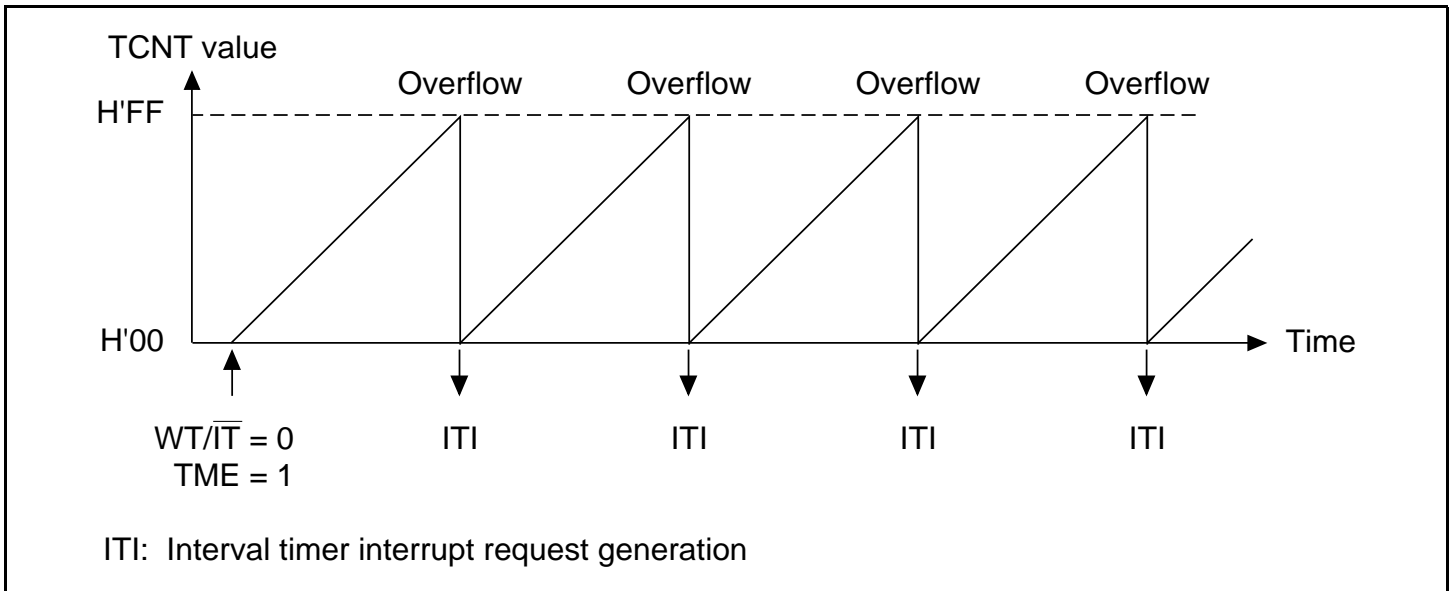


Figure 13.5 Operation in Interval Timer Mode

### 13.3.3 Timing of Setting the Overflow Flag (OVF)

In interval timer mode, when TCNT overflows, the OVF flag in TCSR is set to 1 and an interval timer interrupt (ITI) is simultaneously requested (figure 13.6).

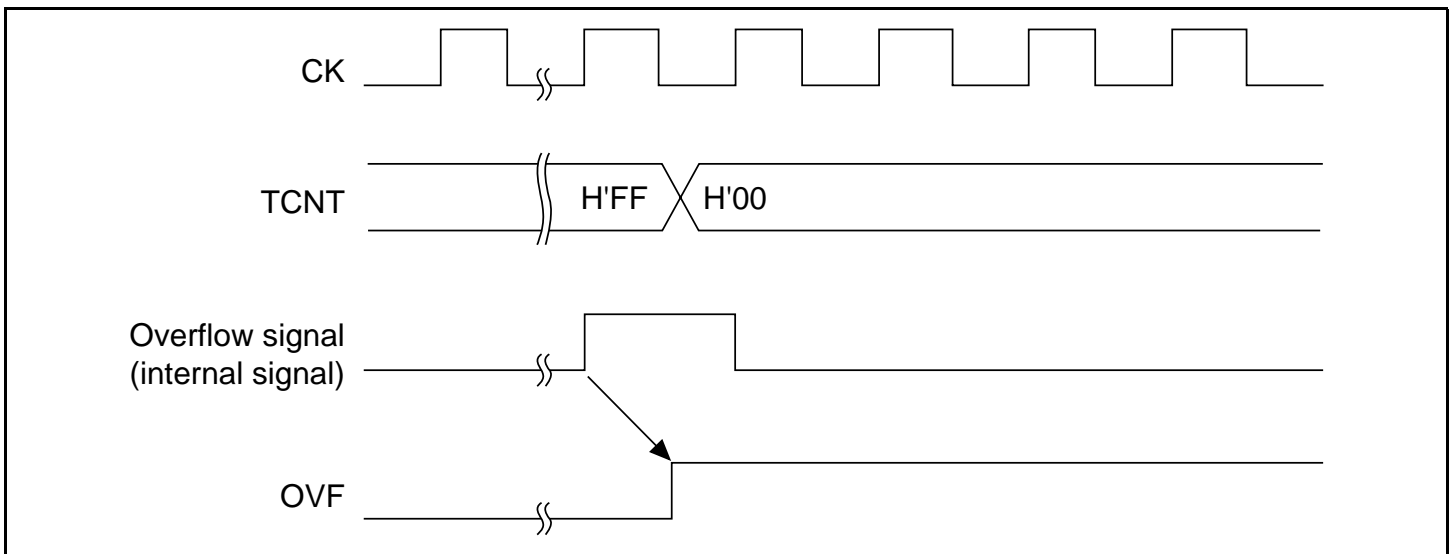


Figure 13.6 Timing of Setting OVF

### 13.3.4 Timing of Setting the Watchdog Timer Overflow Flag (WOVF)

When TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1 and a  $\overline{\text{WDTOVF}}$  signal is output. When the RSTE bit in RSTCSR is set to 1, TCNT overflow enables an internal reset signal to be generated for the entire chip (figure 13.7).

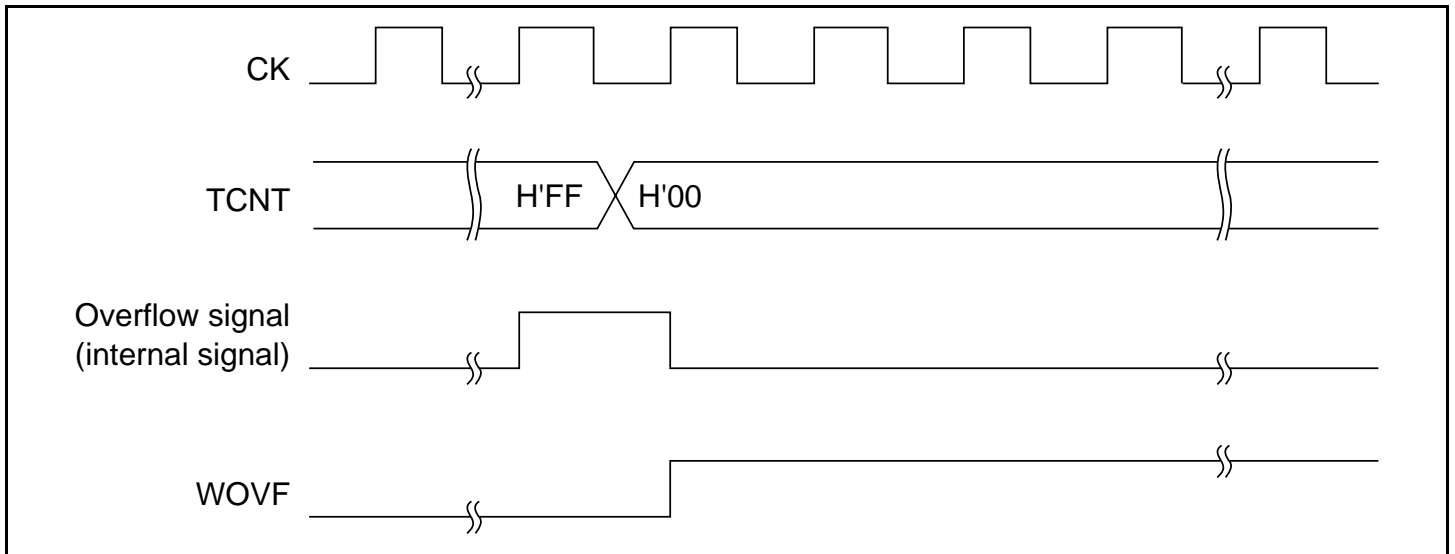


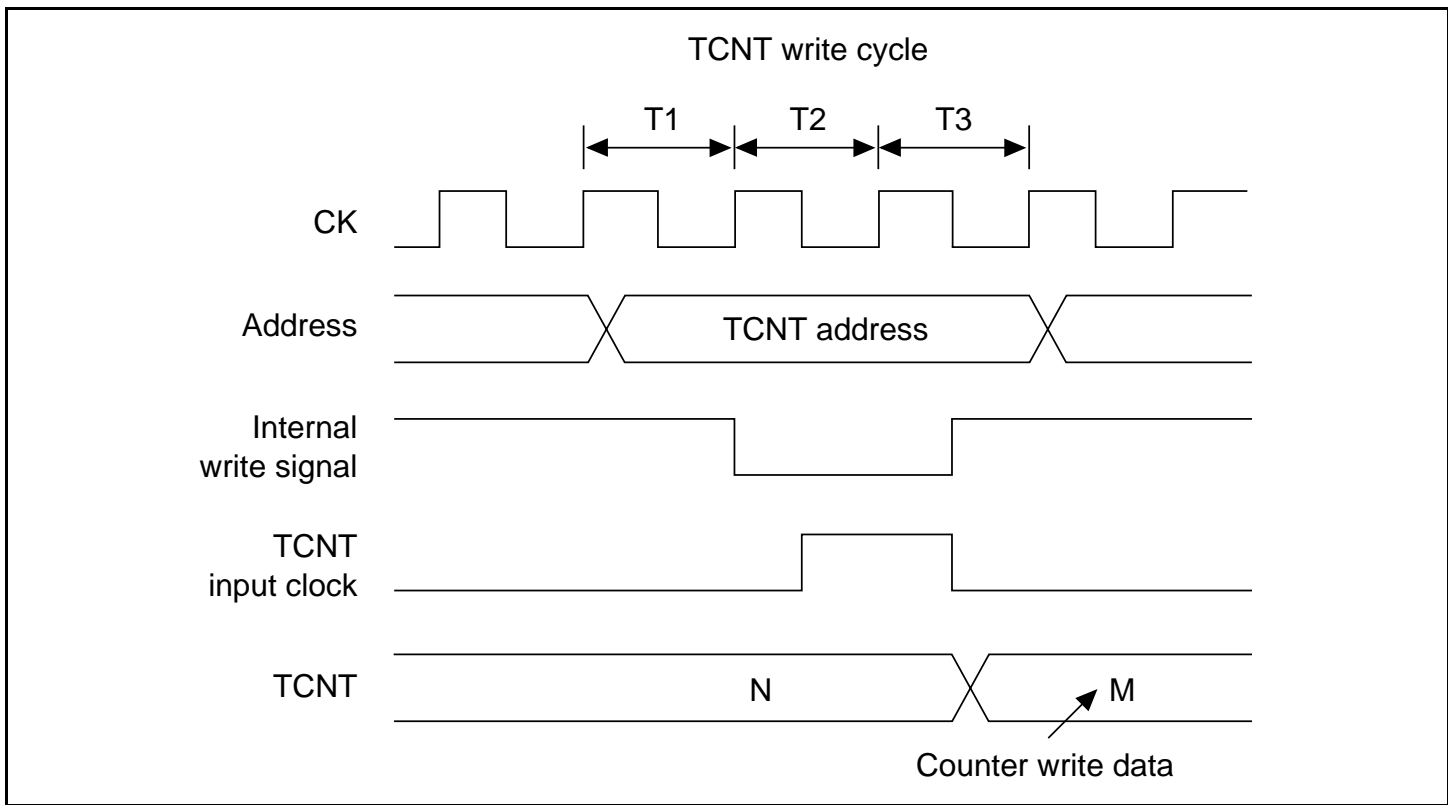
Figure 13.7 Timing of Setting WOVF

## 13.4 Usage Notes

### 13.4.1 TCNT Write and Increment Contention

If a timer counter increment clock pulse is generated during the T3 state of a write cycle to TCNT, the write takes priority and the timer counter is not incremented (figure 13.8).





**Figure 13.8 Contention between TCNT Write and Increment**

### 13.4.2 Changing CKS2 to CKS0 Bit Values

If the values of bits CKS2 to CKS0 in the timer control/status register (TCSR) are rewritten while the WDT is running, the count may not increment correctly. Always stop the watchdog timer (by clearing the TME bit to 0) before changing the values of bits CKS2 to CKS0.

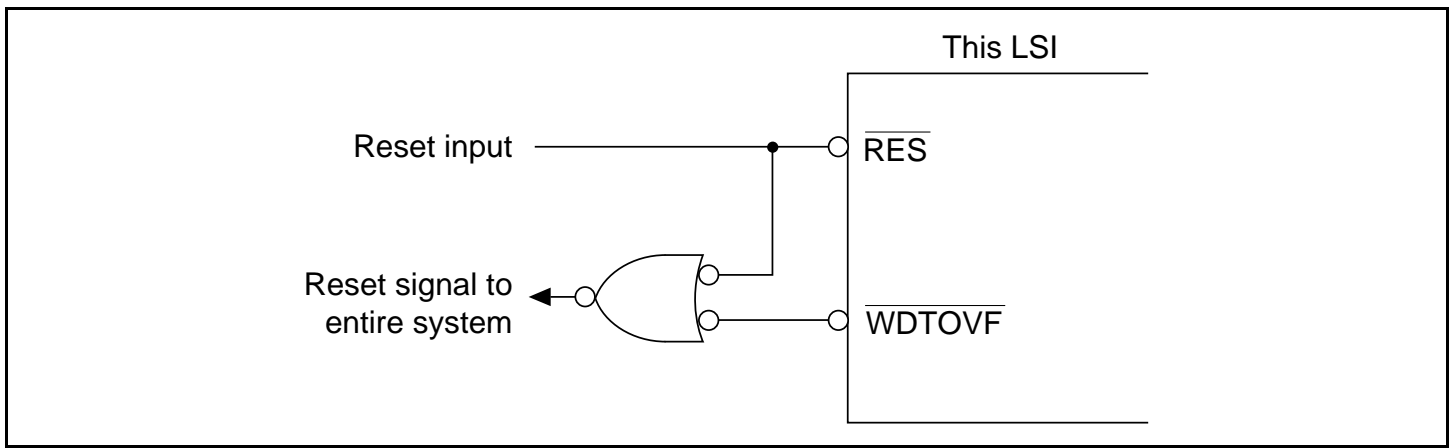
### 13.4.3 Changing between Watchdog Timer/Interval Timer Modes

To prevent incorrect operation, always stop the watchdog timer (by clearing the TME bit to 0) before switching between interval timer mode and watchdog timer mode.

### 13.4.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If a  $\overline{\text{WDTOVF}}$  signal is input to the  $\overline{\text{RES}}$  pin, the chip cannot be initialized correctly. Avoid logical input of the  $\overline{\text{WDTOVF}}$  output signal to the  $\overline{\text{RES}}$  input pin.

To reset the entire system with the  $\overline{\text{WDTOVF}}$  signal, use the circuit shown in figure 13.9.



**Figure 13.9 Example of System Reset Circuit Using  $\overline{\text{WDTOVF}}$  Signal**

### 13.4.5 Internal Reset in Watchdog Timer Mode

If the RSTE bit is cleared to 0 in watchdog timer mode, the chip will not be reset internally when a TCNT overflow occurs, but TCNT and TCSR in the WDT will be reset.

Because the internal clock obtained by dividing the system clock( $\phi$ ) is also reset at this time, the SCI, A/D converter, and CMT that use the internal clock may not operate correctly from hereafter. To continue using these modules, initialize them before use.

### 13.4.6 Manual Reset in Watchdog Timer

When an internal reset is effected by TCNT overflow in watchdog timer mode, the processor waits until the end of the bus cycle at the time of manual reset generation before making the transition to manual reset exception processing. Therefore, the bus cycle is retained in a manual reset, but if a manual reset occurs while the bus is released or during DMAC burst transfer, manual reset exception processing will be deferred until the CPU acquires the bus. However, if the interval from generation of the manual reset until the CPU acquires the bus cycle is equal to or longer than the internal manual reset interval of 512 cycles, the internal manual reset source is ignored instead of being deferred, and manual reset exception processing is not executed.

### 13.4.7 Multiplication Factor for Internal Clock Signal ( $\phi$ ) and Overflow Time

The watchdog timer operates synchronously with the internal clock signal ( $\phi$ ) (at four or eight times the frequency of the input clock signal).

Therefore, even if the same clock signal is selected using the clock select bits (CKS2 to CKS0) in the timer control/status register (TCSR), the overflow timing differs depending on whether the multiplication factor for the internal clock signal ( $\phi$ ) is four or eight.



# Section 14 Compare Match Timer (CMT)

## 14.1 Overview

The SH7058 has an on-chip compare match timer (CMT) comprising two 16-bit timer channels. The CMT has 16-bit counters and can generate interrupts at set intervals.

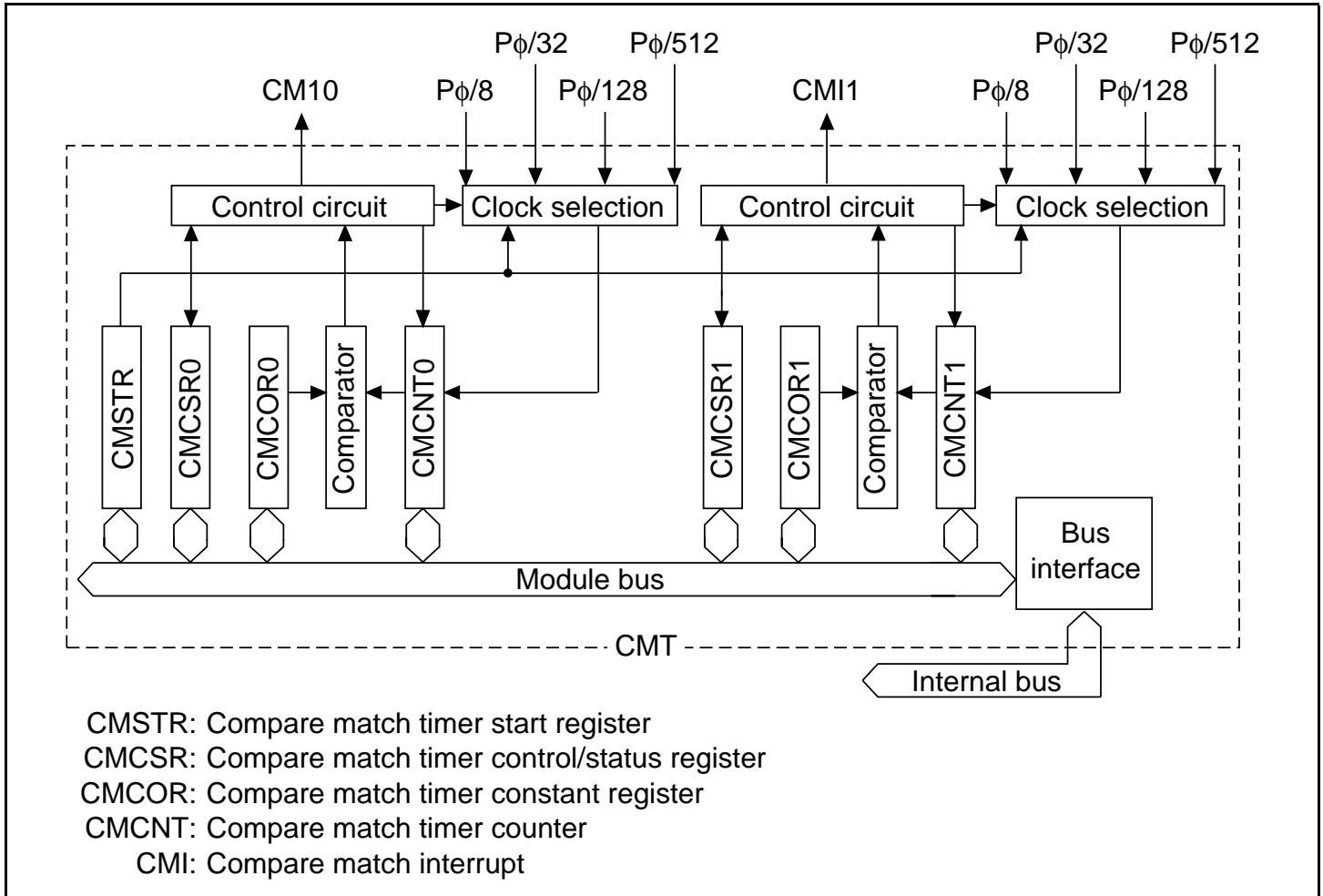
### 14.1.1 Features

The CMT has the following features:

- Four types of counter input clock can be selected
  - One of four internal clocks ( $P\phi/8$ ,  $P\phi/32$ ,  $P\phi/128$ ,  $P\phi/512$ ) can be selected independently for each channel.
- Interrupt sources
  - A compare match interrupt can be requested independently for each channel.

## 14.1.2 Block Diagram

Figure 14.1 shows a block diagram of the CMT.



**Figure 14.1 CMT Block Diagram**

### 14.1.3 Register Configuration

Table 14.1 summarizes the CMT register configuration.

**Table 14.1 Register Configuration**

Channel Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits)
Shared	Compare match timer start register	CMSTR	R/W	H'0000	H'FFFFFF710 8, 16, 32
0	Compare match timer control/status register 0	CMCSR0	R/(W)*	H'0000	H'FFFFFF712 8, 16, 32
	Compare match timer counter 0	CMCNT0	R/W	H'0000	H'FFFFFF714 8, 16, 32
	Compare match timer constant register 0	CMCOR0	R/W	H'FFFF	H'FFFFFF716 8, 16, 32
1	Compare match timer control/status register 1	CMCSR1	R/(W)*	H'0000	H'FFFFFF718 8, 16, 32
	Compare match timer counter 1	CMCNT1	R/W	H'0000	H'FFFFFF71A 8, 16, 32
	Compare match timer constant register 1	CMCOR1	R/W	H'FFFF	H'FFFFFF71C 8, 16, 32

Notes: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles for byte access and word access, and eight or nine internal clock ( $\phi$ ) cycles for longword access.

\* Only 0 can be written to the CMCSR0 and CMCSR1 CMF bits to clear the flags.

## 14.2 Register Descriptions

### 14.2.1 Compare Match Timer Start Register (CMSTR)

The compare match timer start register (CMSTR) is a 16-bit register that selects whether to operate or halt the channel 0 and channel 1 counters (CMCNT). It is initialized to H'0000 by a power-on reset and in the standby modes.

Bit:	15	14	13	12	11	10	9	8
	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	STR1	STR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

- Bits 15–2—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 1—Count Start 1 (STR1): Selects whether to operate or halt compare match timer counter 1.

Bit 1: STR1	Description
0	CMCNT1 count operation halted (Initial value)
1	CMCNT1 count operation

- Bit 0—Count Start 0 (STR0): Selects whether to operate or halt compare match timer counter 0.

Bit 0: STR0	Description
0	CMCNT0 count operation halted (Initial value)
1	CMCNT0 count operation

## 14.2.2 Compare Match Timer Control/Status Register (CMCSR)

The compare match timer control/status register (CMCSR) is a 16-bit register that indicates the occurrence of compare matches, sets the enable/disable status of interrupts, and establishes the clock used for incrementation. It is initialized to H'0000 by a power-on reset and in the standby modes.

Bit:	15	14	13	12	11	10	9	8
	–	–	–	–	–	–	–	–
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	CMF	CMIE	–	–	–	–	CKS1	CKS0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R	R	R	R	R/W	R/W

Note: \* Only 0 can be written to clear the flag.

- Bits 15–8 and 5–2—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 7—Compare Match Flag (CMF): This flag indicates whether or not the CMCNT and CMCOR values have matched.

Bit 7: CMF	Description
0	CMCNT and CMCOR values have not matched (Initial value) [Clearing condition] Write 0 to CMF after reading 1 from it
1	CMCNT and CMCOR values have matched

- Bit 6—Compare Match Interrupt Enable (CMIE): Selects whether to enable or disable a compare match interrupt (CMI) when the CMCNT and CMCOR values have matched (CMF = 1).

Bit 6: CMIE	Description
0	Compare match interrupt (CMI) disabled (Initial value)
1	Compare match interrupt (CMI) enabled



- Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock input to CMCNT from among the four internal clocks obtained by dividing the peripheral clock ( $P\phi$ ). When the STR bit of CMSTR is set to 1, CMCNT begins incrementing with the clock selected by CKS1 and CKS0.

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	$P\phi/8$ (Initial value)
	1	$P\phi/32$
1	0	$P\phi/128$
	1	$P\phi/512$

### 14.2.3 Compare Match Timer Counter (CMCNT)

The compare match timer counter (CMCNT) is a 16-bit register used as an up-counter for generating interrupt requests.

When an internal clock is selected with the CKS1 and CKS0 bits of the CMCSR register and the STR bit of CMSTR is set to 1, CMCNT begins incrementing with that clock. When the CMCNT value matches that of the compare match timer constant register (CMCOR), CMCNT is cleared to H'0000 and the CMF flag of CMCSR is set to 1. If the CMIE bit of CMCSR is set to 1 at this time, a compare match interrupt (CMI) is requested.

CMCNT is initialized to H'0000 by a power-on reset and in the standby modes. It is not initialized by a manual reset.

Bit:	15	14	13	12	11	10	9	8
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
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:	7	6	5	4	3	2	1	0
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

## 14.2.4 Compare Match Timer Constant Register (CMCOR)

The compare match timer constant register (CMCOR) is a 16-bit register that sets the period for compare match with CMCNT.

CMCOR is initialized to H'FFFF by a power-on reset and in the standby modes. It is not initialized by a manual reset.

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

## 14.3 Operation

### 14.3.1 Cyclic Count Operation

When an internal clock is selected with the CKS1, CKS0 bits of the CMCSR register and the STR bit of CMSTR is set to 1, CMCNT begins incrementing with the selected clock. When the CMCNT counter value matches that of the compare match constant register (CMCOR), the CMCNT counter is cleared to H'0000 and the CMF flag of the CMCSR register is set to 1. If the CMIE bit of the CMCSR register is set to 1 at this time, a compare match interrupt (CMI) is requested. The CMCNT counter begins counting up again from H'0000.

Figure 14.2 shows the compare match counter operation.

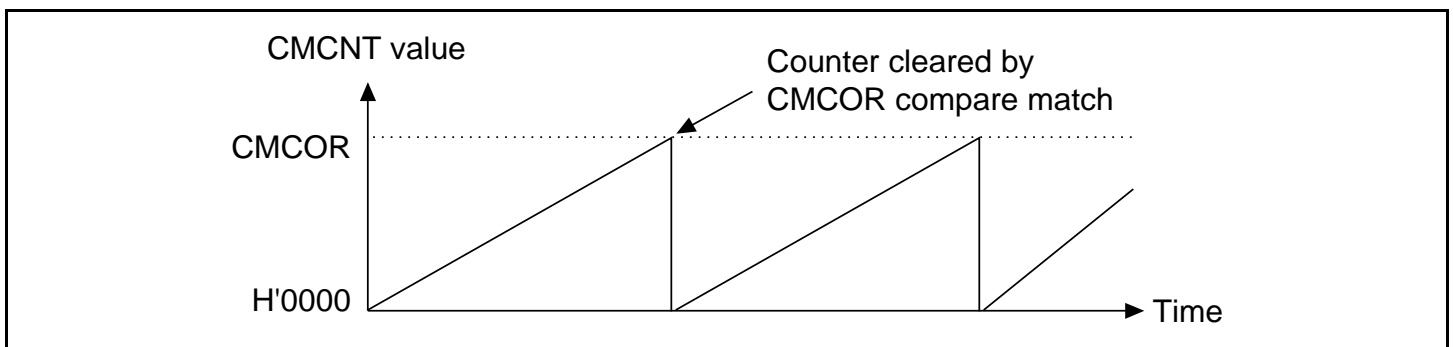


Figure 14.2 Counter Operation

### 14.3.2 CMCNT Count Timing

One of four clocks ( $P\phi/8$ ,  $P\phi/32$ ,  $P\phi/128$ ,  $P\phi/512$ ) obtained by dividing the peripheral clock ( $P\phi$ ) can be selected by the CKS1 and CKS0 bits of CMCSR. Figure 14.3 shows the timing.

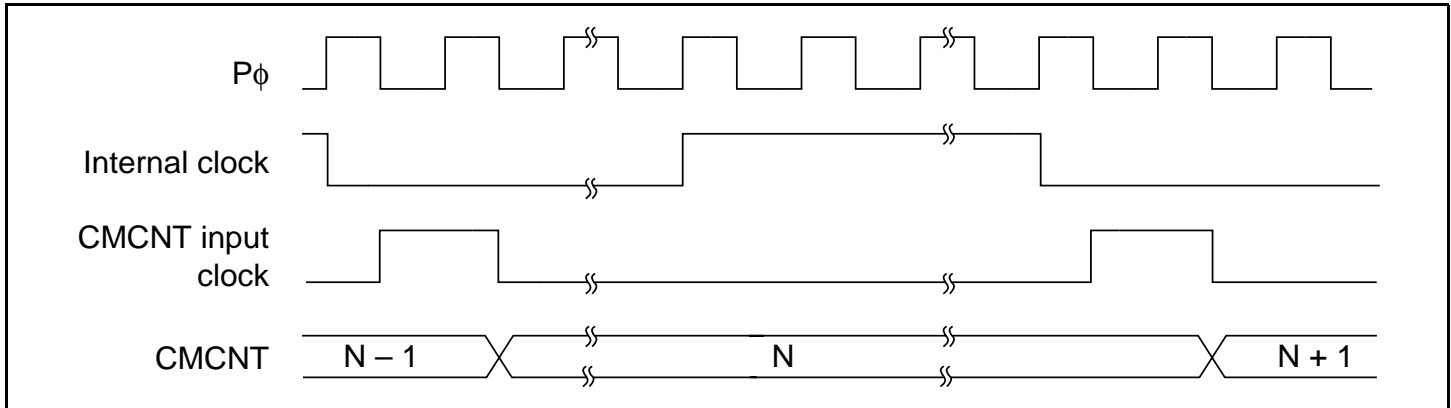


Figure 14.3 Count Timing

## 14.4 Interrupts

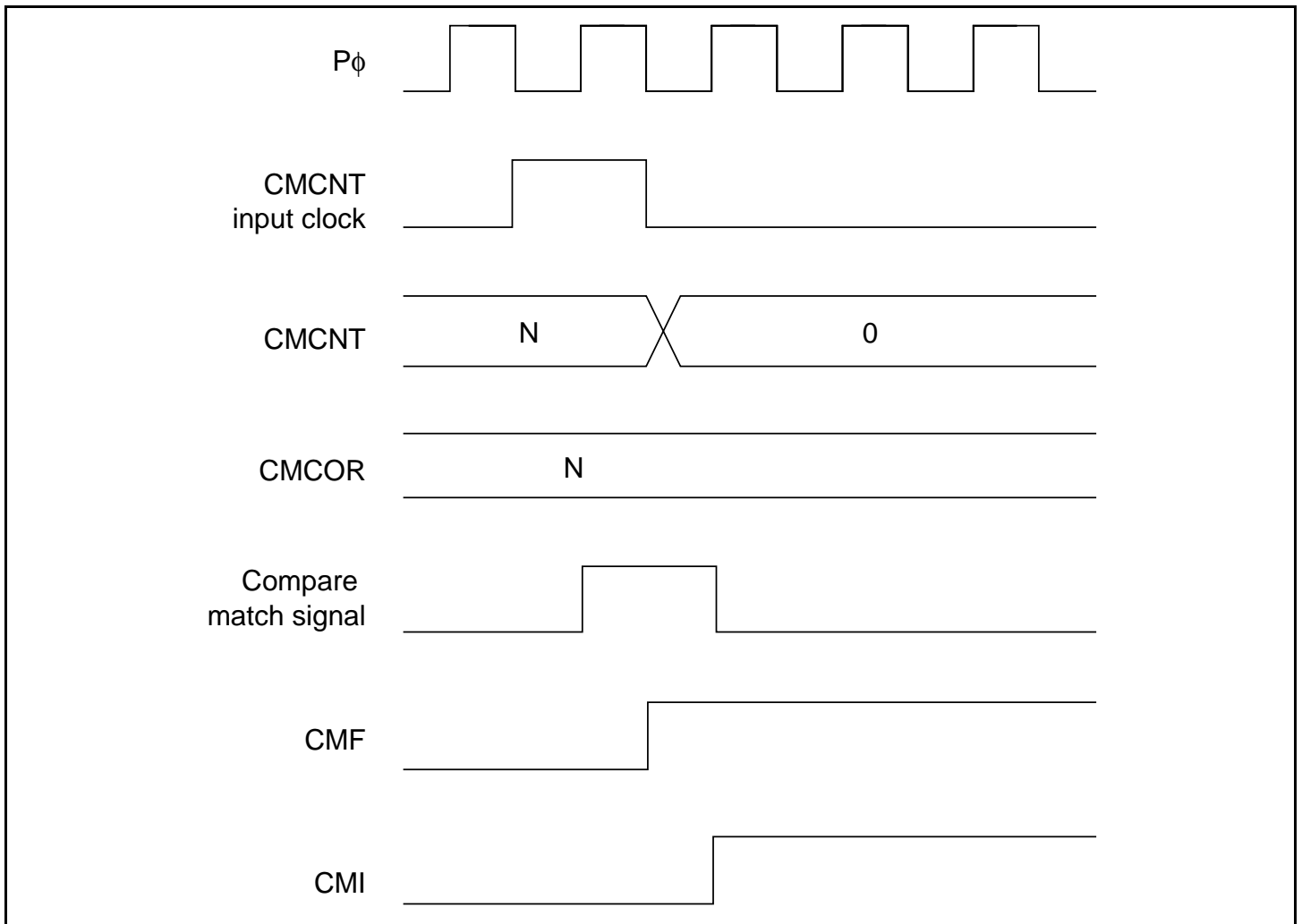
### 14.4.1 Interrupt Sources and DTC Activation

The CMT has a compare match interrupt for each channel, with independent vector addresses allocated to each of them. The corresponding interrupt request is output when interrupt request flag CMF is set to 1 and interrupt enable bit CMIE has also been set to 1.

When activating CPU interrupts by interrupt request, the priority between the channels can be changed by means of interrupt controller settings. See section 7, Interrupt Controller (INTC), for details.

### 14.4.2 Compare Match Flag Set Timing

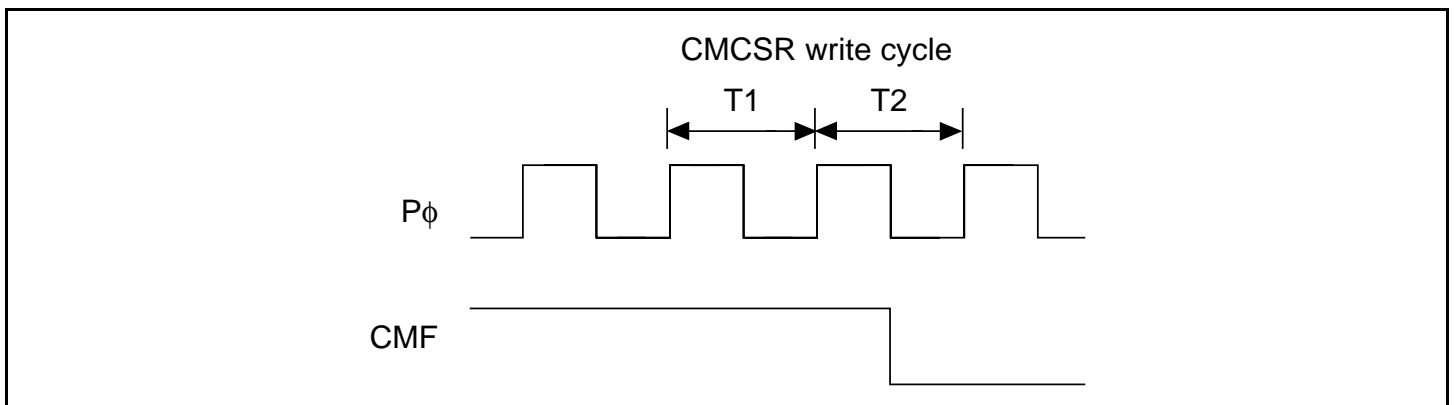
The CMF bit of the CMCSR register is set to 1 by the compare match signal generated when the CMCOR register and the CMCNT counter match. The compare match signal is generated upon the final state of the match (timing at which the CMCNT counter matching count value is updated). Consequently, after the CMCOR register and the CMCNT counter match, a compare match signal will not be generated until a CMCNT counter input clock occurs. Figure 14.4 shows the CMF bit set timing.



**Figure 14.4 CMF Set Timing**

### 14.4.3 Compare Match Flag Clear Timing

The CMF bit of the CMCSR register is cleared by writing a 0 to it after reading a 1. Figure 14.5 shows the timing when the CMF bit is cleared by the CPU.



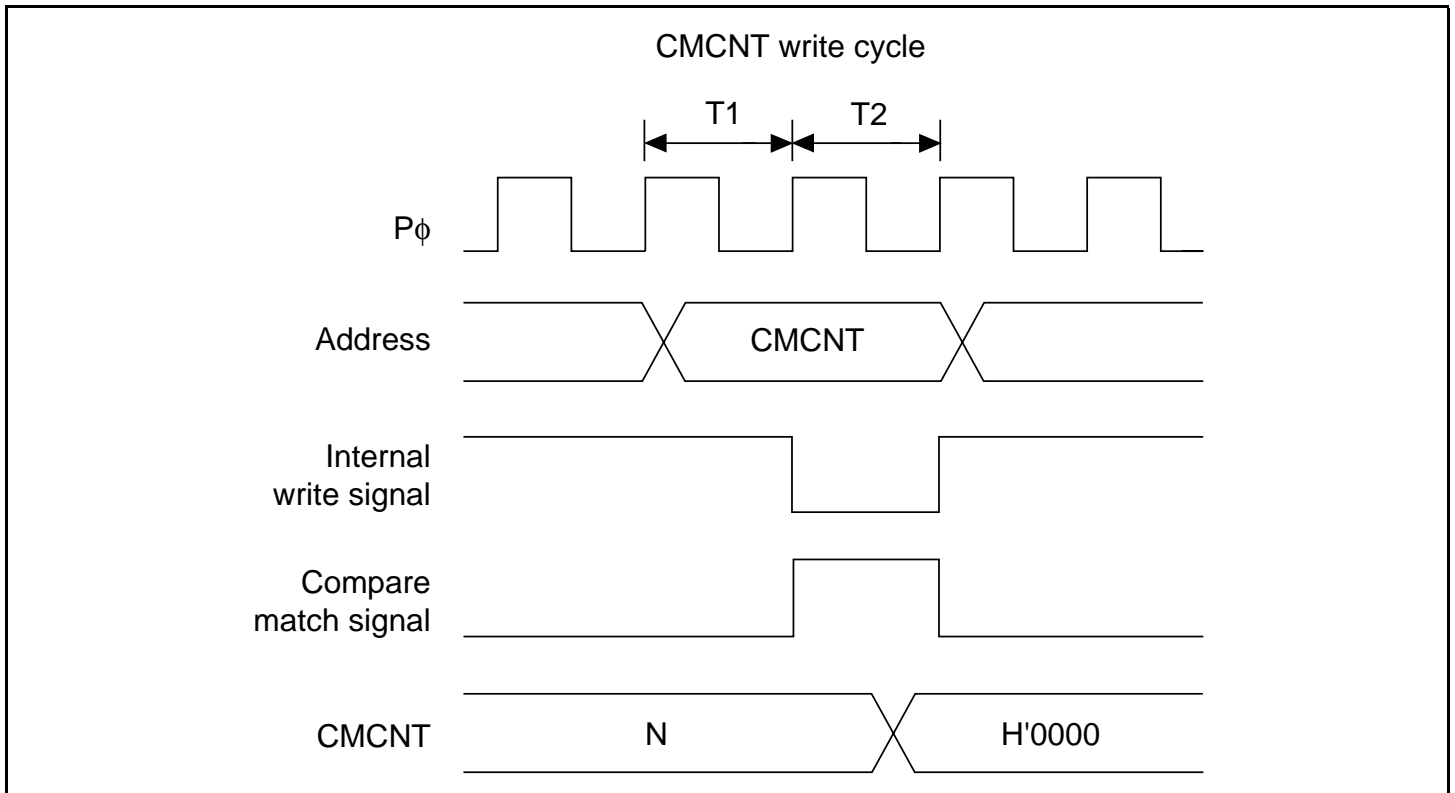
**Figure 14.5 Timing of CMF Clear by the CPU**

## 14.5 Usage Notes

Take care that the contentions described in sections 14.5.1 to 14.5.3 do not arise during CMT operation.

### 14.5.1 Contention between CMCNT Write and Compare Match

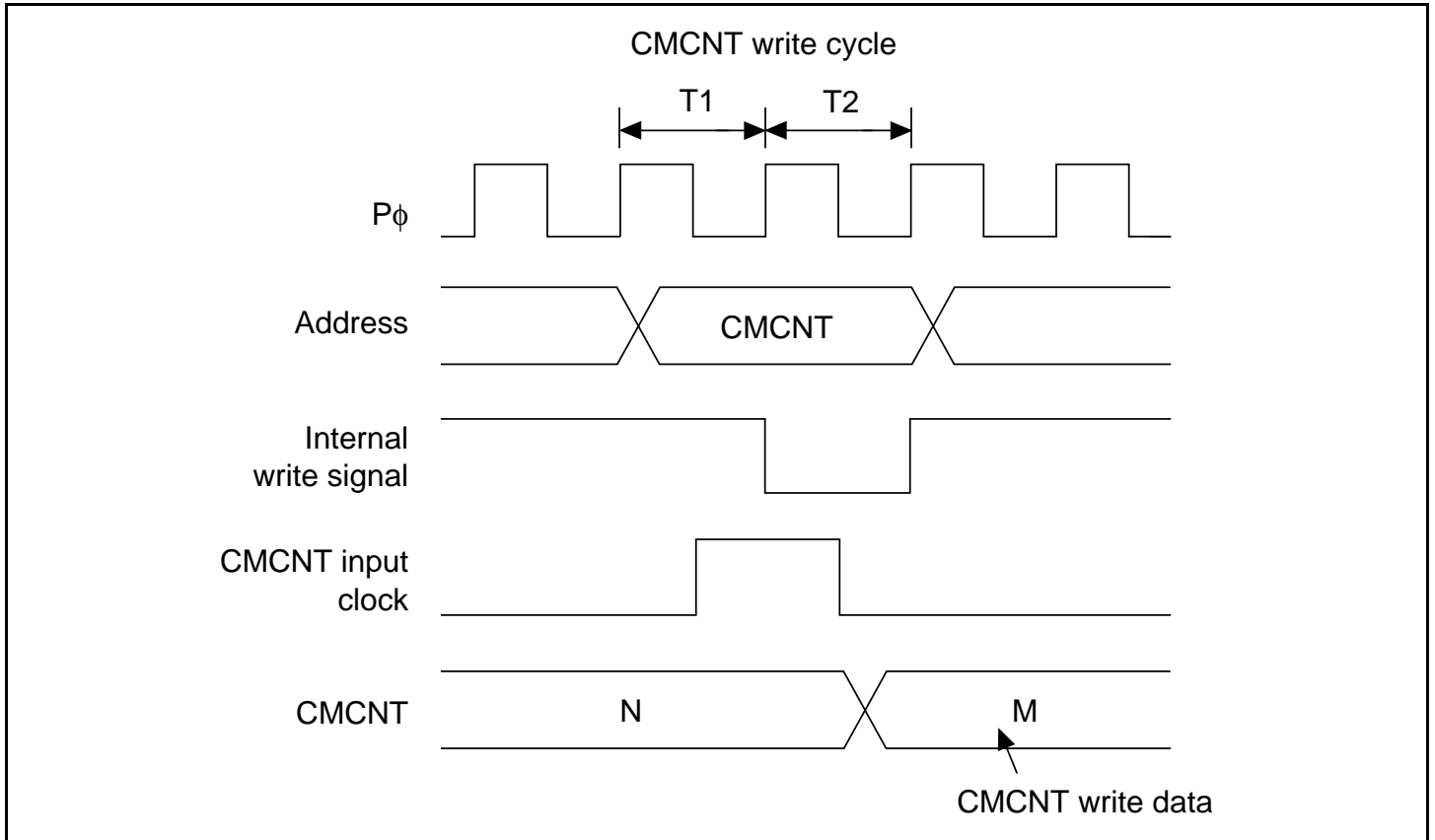
If a compare match signal is generated during the T2 state of the CMCNT counter write cycle, the CMCNT counter clear has priority, so the write to the CMCNT counter is not performed. Figure 14.6 shows the timing.



**Figure 14.6 CMCNT Write and Compare Match Contention**

## 14.5.2 Contention between CMCNT Word Write and Incrementation

If an increment occurs during the T2 state of the CMCNT counter word write cycle, the counter write has priority, so no increment occurs. Figure 14.7 shows the timing.

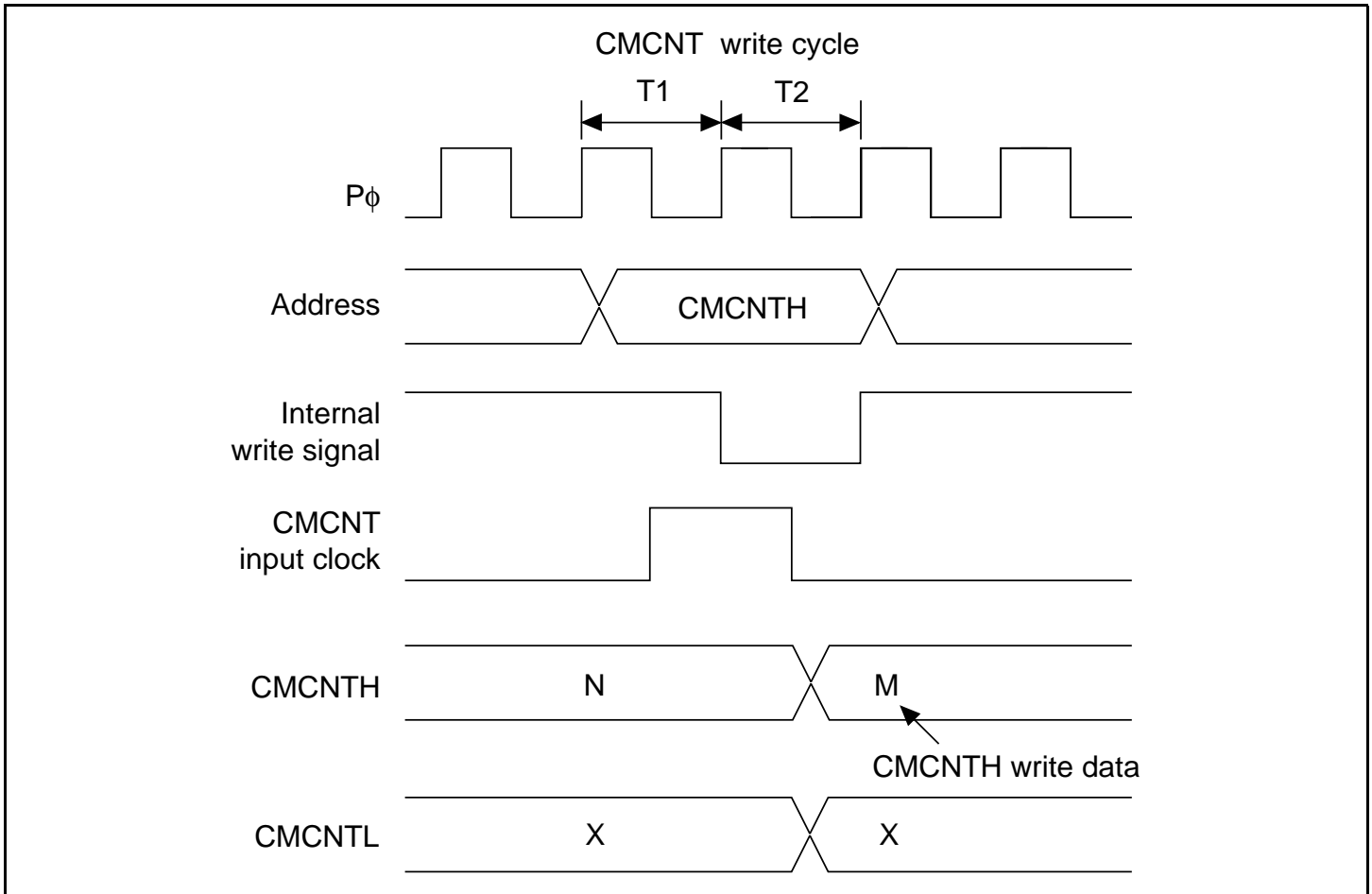


**Figure 14.7 CMCNT Word Write and Increment Contention**

### 14.5.3 Contention between CMCNT Byte Write and Incrementation

If an increment occurs during the T2 state of the CMCNT byte write cycle, the counter write has priority, so no increment of the write data results on the side on which the write was performed. The byte data on the side on which writing was not performed is also not incremented, so the contents are those before the write.

Figure 14.8 shows the timing when an increment occurs during the T2 state of the CMCNTH write cycle.



**Figure 14.8 CMCNT Byte Write and Increment Contention**

# Section 15 Serial Communication Interface (SCI)

## 15.1 Overview

The SH7058 has a serial communication interface (SCI) with five independent channels.

The SCI supports both asynchronous and synchronous serial communication. It also has a multiprocessor communication function for serial communication between two or more processors, and a clock inverted input/output function.

### 15.1.1 Features

The SCI has the following features:

- Selection of asynchronous or synchronous as the serial communication mode
  - Asynchronous mode

Serial data communication is synchronized in character units. The SCI can communicate with a universal asynchronous receiver/transmitter (UART), an asynchronous communication interface adapter (ACIA), or any other chip that employs standard asynchronous serial communication. It can also communicate with two or more other processors using the multiprocessor communication function. There are twelve selectable serial data communication formats.

    - Data length: seven or eight bits
    - Stop bit length: one or two bits
    - Parity: even, odd, or none
    - Multiprocessor bit: one or none
    - Receive error detection: parity, overrun, and framing errors
    - Break detection: by reading the RxD level directly when a framing error occurs
  - Synchronous mode

Serial data communication is synchronized with a clock signal. The SCI can communicate with other chips having a synchronous communication function. There is one serial data communication format.

    - Data length: eight bits
    - Receive error detection: overrun errors
    - Serial clock inverted input/output
- Full duplex communication: The transmitting and receiving sections are independent, so the SCI can transmit and receive simultaneously. Both sections use double buffering, so continuous data transfer is possible in both the transmit and receive directions.
- On-chip baud rate generator with selectable bit rates

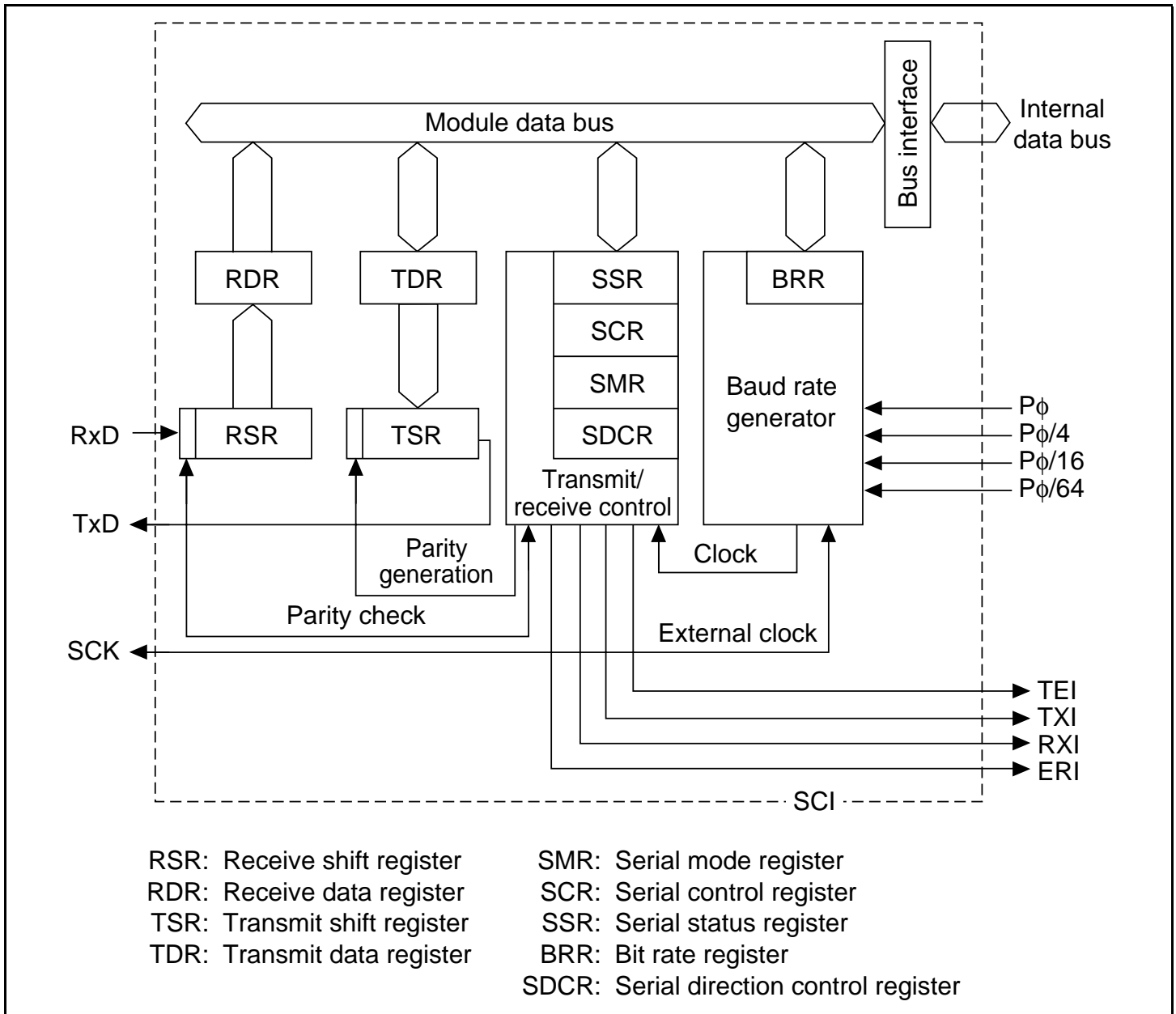


- Internal or external transmit/receive clock source: baud rate generator (internal) or SCK pin (external)
- Four types of interrupts: Transmit-data-empty, transmit-end, receive-data-full, and receive-error interrupts are requested independently. The transmit-data-empty and receive-data-full interrupts can start the direct memory access controller (DMAC) to transfer data.
- Selection of LSB-first or MSB-first transfer (8-bit length)

This selection is available regardless of the communication mode. (The descriptions in this section are based on LSB-first transfer.)

### 15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the SCI.



**Figure 15.1 SCI Block Diagram**

### 15.1.3 Pin Configuration

Table 15.1 summarizes the SCI pins by channel.

**Table 15.1 SCI Pins**

Channel	Pin Name	Abbreviation	Input/Output	Function
0	Serial clock pin	SCK0	Input/output	SCI0 clock input/output
	Receive data pin	RxD0	Input	SCI0 receive data input
	Transmit data pin	TxD0	Output	SCI0 transmit data output
1	Serial clock pin	SCK1	Input/output	SCI1 clock input/output
	Receive data pin	RxD1	Input	SCI1 receive data input
	Transmit data pin	TxD1	Output	SCI1 transmit data output
2	Serial clock pin	SCK2	Input/output	SCI2 clock input/output
	Receive data pin	RxD2	Input	SCI2 receive data input
	Transmit data pin	TxD2	Output	SCI2 transmit data output
3	Serial clock pin	SCK3	Input/output	SCI3 clock input/output
	Receive data pin	RxD3	Input	SCI3 receive data input
	Transmit data pin	TxD3	Output	SCI3 transmit data output
4	Serial clock pin	SCK4	Input/output	SCI4 clock input/output
	Receive data pin	RxD4	Input	SCI4 receive data input
	Transmit data pin	TxD4	Output	SCI4 transmit data output

Note: In the text the pins are referred to as SCK, RxD, and TxD, omitting the channel number.

## 15.1.4 Register Configuration

Table 15.2 summarizes the SCI internal registers. These registers select the communication mode (asynchronous or synchronous), specify the data format and bit rate, and control the transmitter and receiver sections.

**Table 15.2 Registers**

Channel	Name	Abbreviation	R/W	Initial Value	Address* <sup>2</sup>	Access Size
0	Serial mode register 0	SMR0	R/W	H'00	H'FFFFFF000	8, 16
	Bit rate register 0	BRR0	R/W	H'FF	H'FFFFFF001	
	Serial control register 0	SCR0	R/W	H'00	H'FFFFFF002	
	Transmit data register 0	TDR0	R/W	H'FF	H'FFFFFF003	
	Serial status register 0	SSR0	R/(W) * <sup>1</sup>	H'84	H'FFFFFF004	
	Receive data register 0	RDR0	R	H'00	H'FFFFFF005	
	Serial direction control register 0	SDCR0	R/W	H'F2	H'FFFFFF006	8
1	Serial mode register 1	SMR1	R/W	H'00	H'FFFFFF008	8, 16
	Bit rate register 1	BRR1	R/W	H'FF	H'FFFFFF009	
	Serial control register 1	SCR1	R/W	H'00	H'FFFFFF00A	
	Transmit data register 1	TDR1	R/W	H'FF	H'FFFFFF00B	
	Serial status register 1	SSR1	R/(W) * <sup>1</sup>	H'84	H'FFFFFF00C	
	Receive data register 1	RDR1	R	H'00	H'FFFFFF00D	
	Serial direction control register 1	SDCR1	R/W	H'F2	H'FFFFFF00E	8
2	Serial mode register 2	SMR2	R/W	H'00	H'FFFFFF010	8, 16
	Bit rate register 2	BRR2	R/W	H'FF	H'FFFFFF011	
	Serial control register 2	SCR2	R/W	H'00	H'FFFFFF012	
	Transmit data register 2	TDR2	R/W	H'FF	H'FFFFFF013	
	Serial status register 2	SSR2	R/(W) * <sup>1</sup>	H'84	H'FFFFFF014	
	Receive data register 2	RDR2	R	H'00	H'FFFFFF015	
	Serial direction control register 2	SDCR2	R/W	H'F2	H'FFFFFF016	8

**Table 15.2 Registers (cont)**

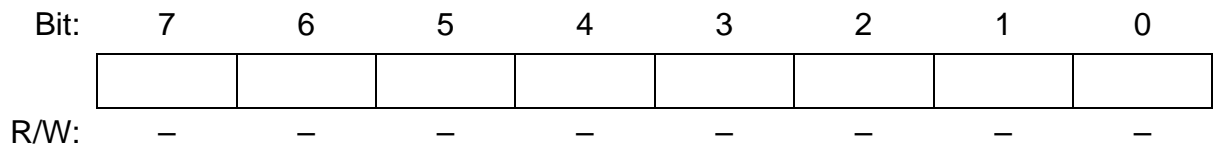
Channel	Name	Abbreviation	R/W	Initial Value	Address* <sup>2</sup>	Access Size
3	Serial mode register 3	SMR3	R/W	H'00	H'FFFFFF018	8, 16
	Bit rate register 3	BRR3	R/W	H'FF	H'FFFFFF019	
	Serial control register 3	SCR3	R/W	H'00	H'FFFFFF01A	
	Transmit data register 3	TDR3	R/W	H'FF	H'FFFFFF01B	
	Serial status register 3	SSR3	R/(W) * <sup>1</sup>	H'84	H'FFFFFF01C	
	Receive data register 3	RDR3	R	H'00	H'FFFFFF01D	
	Serial direction control register 3	SDCR3	R/W	H'F2	H'FFFFFF01E	8
4	Serial mode register 4	SMR4	R/W	H'00	H'FFFFFF020	8, 16
	Bit rate register 4	BRR4	R/W	H'FF	H'FFFFFF021	
	Serial control register 4	SCR4	R/W	H'00	H'FFFFFF022	
	Transmit data register 4	TDR4	R/W	H'FF	H'FFFFFF023	
	Serial status register 4	SSR4	R/(W) * <sup>1</sup>	H'84	H'FFFFFF024	
	Receive data register 4	RDR4	R	H'00	H'FFFFFF025	
	Serial direction control register 4	SDCR4	R/W	H'F2	H'FFFFFF026	8

Notes: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles for byte access and word access, and eight or nine internal clock ( $\phi$ ) cycles for longword access.

1. Only 0 can be written to clear the flags.
2. Do not access empty addresses.

## 15.2 Register Descriptions

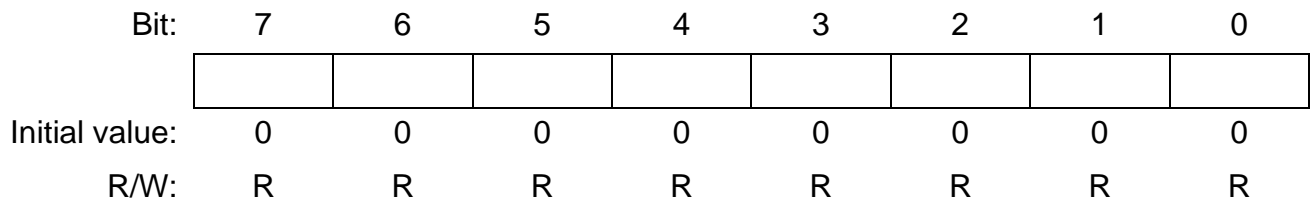
### 15.2.1 Receive Shift Register (RSR)



The receive shift register (RSR) receives serial data. Data input at the RxD pin is loaded into RSR 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 RDR.

The CPU cannot read or write to RSR directly.

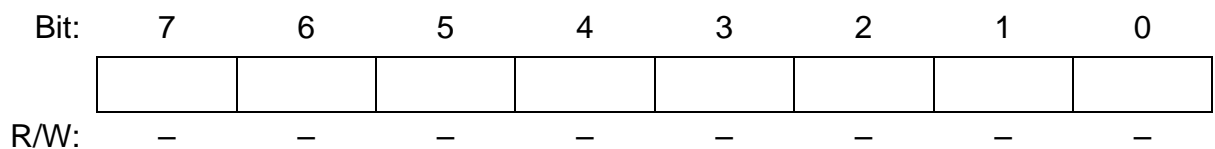
### 15.2.2 Receive Data Register (RDR)



The receive data register (RDR) stores serial receive data. The SCI completes the reception of one byte of serial data by moving the received data from the receive shift register (RSR) into RDR for storage. RSR is then ready to receive the next data. This double buffering allows the SCI to receive data continuously.

The CPU can read but not write to RDR. RDR is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode. It is not initialized by a manual reset.

### 15.2.3 Transmit Shift Register (TSR)



The transmit shift register (TSR) transmits serial data. The SCI loads transmit data from the transmit data register (TDR) into TSR, 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 TDR into TSR and starts transmitting again. If the TDRE bit of SSR is 1, however, the SCI does not load the TDR contents into TSR.

The CPU cannot read or write to TSR directly.

### 15.2.4 Transmit Data Register (TDR)

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The transmit data register (TDR) is an 8-bit register that stores data for serial transmission. When the SCI detects that the transmit shift register (TSR) is empty, it moves transmit data written in TDR into TSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in TDR during serial transmission from TSR.

The CPU can always read and write to TDR. TDR is initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode. It is not initialized by a manual reset.

### 15.2.5 Serial Mode Register (SMR)

Bit:	7	6	5	4	3	2	1	0
	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS1	CKS0
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

The serial mode register (SMR) 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 SMR. SMR is initialized to H'00 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

- Bit 7—Communication Mode (C/ $\bar{A}$ ): Selects whether the SCI operates in asynchronous or synchronous mode.

Bit 7: C/ $\bar{A}$	Description
0	Asynchronous mode (Initial value)
1	Synchronous mode

- Bit 6—Character Length (CHR): Selects 7-bit or 8-bit data in asynchronous mode. In synchronous mode, the data length is always eight bits, regardless of the CHR setting.

Bit 6: CHR	Description
0	Eight-bit data (Initial value)
1	Seven-bit data When 7-bit data is selected, the MSB (bit 7) of the transmit data register is not transmitted. LSB-first/MSB-first selection is not available.

- Bit 5—Parity Enable (PE): Selects whether to add a parity bit to transmit data and to check the parity of receive data, in asynchronous mode. In synchronous mode and when using a multiprocessor format, a parity bit is neither added nor checked, regardless of the PE bit setting.

Bit 5: PE	Description
0	Parity bit not added or checked (Initial value)
1	Parity bit added and checked When PE is set to 1, an even or odd parity bit is added to transmit data, depending on the parity mode ( $O/\bar{E}$ bit) setting. Receive data parity is checked according to the even/odd ( $O/\bar{E}$ bit) setting.

- Bit 4—Parity Mode ( $O/\bar{E}$ ): Selects even or odd parity when parity bits are added and checked. The  $O/\bar{E}$  setting is used only in asynchronous mode and only when the parity enable bit (PE) is set to 1 to enable parity addition and checking. The  $O/\bar{E}$  setting is invalid in synchronous mode, in asynchronous mode when parity bit addition and checking is disabled, and when using a multiprocessor format.

Bit 4: $O/\bar{E}$	Description
0	Even parity (Initial value) 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.
1	Odd parity 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 3—Stop Bit Length (STOP): 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 synchronous mode because no stop bits are added.

In 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.

Bit 3: STOP	Description
0	One stop bit (Initial value) In transmitting, a single bit of 1 is added at the end of each transmitted character.
1	Two stop bits In transmitting, two 1-bits are added at the end of each transmitted character.

- Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, settings of the parity enable (PE) and parity mode ( $O/\bar{E}$ ) bits are ignored. The MP bit setting is used only in asynchronous mode; it is ignored in synchronous mode. For the multiprocessor communication function, see section 15.3.3, Multiprocessor Communication.

Bit 2: MP	Description
0	Multiprocessor function disabled (Initial value)
1	Multiprocessor format selected

- Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the internal clock source of the on-chip baud rate generator. Four clock sources are available:  $P\phi$ ,  $P\phi/4$ ,  $P\phi/16$ , or  $P\phi/64$  ( $P\phi$  is the peripheral clock). For further information on the clock source, bit rate register settings, and baud rate, see section 15.2.8, Bit Rate Register (BRR).

Bit 1: CKS1	Bit 0: CKS0	Description
0	0	$P\phi$ (Initial value)
	1	$P\phi/4$
1	0	$P\phi/16$
	1	$P\phi/64$



## 15.2.6 Serial Control Register (SCR)

Bit:	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
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

The serial control register (SCR) operates the SCI transmitter/receiver, selects the serial clock output in asynchronous mode, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCR. SCR is initialized to H'00 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

- **Bit 7—Transmit Interrupt Enable (TIE):** Enables or disables the transmit-data-empty interrupt (TXI) requested when the transmit data register empty bit (TDRE) in the serial status register (SSR) is set to 1 by transfer of serial transmit data from TDR to TSR.

Bit 7: TIE	Description
0	Transmit-data-empty interrupt request (TXI) is disabled (Initial value) The TXI interrupt request can be cleared by reading TDRE after it has been set to 1, then clearing TDRE to 0, or by clearing TIE to 0.
1	Transmit-data-empty interrupt request (TXI) is enabled

- **Bit 6—Receive Interrupt Enable (RIE):** Enables or disables the receive-data-full interrupt (RXI) requested when the receive data register full bit (RDRF) in the serial status register (SSR) is set to 1 by transfer of serial receive data from RSR to RDR. It also enables or disables receive-error interrupt (ERI) requests.

Bit 6: RIE	Description
0	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are disabled (Initial value) RXI and ERI interrupt requests can be cleared by reading the RDRF flag or error flag (FER, PER, or ORER) after it has been set to 1, then clearing the flag to 0, or by clearing RIE to 0.
1	Receive-data-full interrupt (RXI) and receive-error interrupt (ERI) requests are enabled

- Bit 5—Transmit Enable (TE): Enables or disables the SCI serial transmitter.

Bit 5: TE	Description
0	Transmitter disabled (Initial value) The transmit data register empty bit (TDRE) in the serial status register (SSR) is locked at 1.
1	Transmitter enabled Serial transmission starts when the transmit data register empty (TDRE) bit in the serial status register (SSR) is cleared to 0 after writing of transmit data into TDR. Select the transmit format in SMR before setting TE to 1.

- Bit 4—Receive Enable (RE): Enables or disables the SCI serial receiver.

Bit 4: RE	Description
0	Receiver disabled (Initial value) Clearing RE to 0 does not affect the receive flags (RDRF, FER, PER, ORER). These flags retain their previous values.
1	Receiver enabled Serial reception starts when a start bit is detected in asynchronous mode, or synchronous clock input is detected in synchronous mode. Select the receive format in SMR before setting RE to 1.

- Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE setting is used only in asynchronous mode, and only if the multiprocessor mode bit (MP) in the serial mode register (SMR) is set to 1 during reception. The MPIE setting is ignored in synchronous mode or when the MP bit is cleared to 0.

Bit 3: MPIE	Description
0	<p>Multiprocessor interrupts are disabled (normal receive operation) (Initial value)</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When the MPIE bit is cleared to 0</li> <li>• When data with MPB = 1 is received</li> </ul>
1	<p>Multiprocessor interrupts are enabled. Receive-data-full interrupt requests (RXI), receive-error interrupt requests (ERI), and setting of the RDRF, FER, and ORER status flags in the serial status register (SSR) are disabled until data with the multiprocessor bit set to 1 is received.</p> <p>The SCI does not transfer receive data from RSR to RDR, does not detect receive errors, and does not set the RDRF, FER, and ORER flags in the serial status register (SSR). When it receives data that includes MPB = 1, MPB is set to 1, and the SCI automatically clears MPIE to 0, generates RXI and ERI interrupts (if the TIE and RIE bits in SCR are set to 1), and allows the FER and ORER bits to be set.</p>

- Bit 2—Transmit-End Interrupt Enable (TEIE): Enables or disables the transmit-end interrupt (TEI) requested if TDR does not contain valid transmit data when the MSB is transmitted.

Bit 2: TEIE	Description
0	Transmit-end interrupt (TEI) requests are disabled* (Initial value)
1	Transmit-end interrupt (TEI) requests are enabled*

Note: \* The TEI request can be cleared by reading the TDRE bit in the serial status register (SSR) after it has been set to 1, then clearing TDRE to 0 and clearing the transmit end (TEND) bit to 0; or by clearing the TEIE bit to 0.

- Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits select the SCI clock source and enable or disable clock output from the SCK pin. Depending on the combination of CKE1 and CKE0, the SCK pin can be used for serial clock output, or serial clock input. Select the SCK pin function by using the pin function controller (PFC).

The CKE0 setting is valid only in asynchronous mode, and only when the SCI is internally clocked (CKE1 = 0). The CKE0 setting is ignored in synchronous mode, or when an external clock source is selected (CKE1 = 1). For further details on selection of the SCI clock source, see table 15.9 in section 15.3, Operation.

Bit 1: CKE1	Bit 0: CKE0	Description* <sup>1</sup>	
0	0	Asynchronous mode	Internal clock, SCK pin used for input pin (input signal is ignored) or output pin (output level is undefined)* <sup>2</sup>
		Synchronous mode	Internal clock, SCK pin used for synchronous clock output* <sup>2</sup>
0	1	Asynchronous mode	Internal clock, SCK pin used for clock output* <sup>3</sup>
		Synchronous mode	Internal clock, SCK pin used for synchronous clock output
1	0	Asynchronous mode	External clock, SCK pin used for clock input* <sup>4</sup>
		Synchronous mode	External clock, SCK pin used for synchronous clock input
1	1	Asynchronous mode	External clock, SCK pin used for clock input* <sup>4</sup>
		Synchronous mode	External clock, SCK pin used for synchronous clock input

- Notes:
1. The SCK pin is multiplexed with other functions. Use the pin function controller (PFC) to select the SCK function for this pin, as well as the I/O direction.
  2. Initial value.
  3. The output clock frequency is the same as the bit rate.
  4. The input clock frequency is 16 times the bit rate.

## 15.2.7 Serial Status Register (SSR)

Bit:	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value:	1	0	0	0	0	1	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: \* Only 0 can be written to clear the flag.

The serial status register (SSR) is an 8-bit register containing multiprocessor bit values, and status flags that indicate the SCI operating status.

The CPU can always read and write to SSR, but cannot write 1 in the status flags (TDRE, RDRF, ORER, PER, and FER). These flags can be cleared to 0 only if they have first been read (after being set to 1). Bits 2 (TEND) and 1 (MPB) are read-only bits that cannot be written. SSR is initialized to H'84 by a power-on reset, and in hardware standby mode and software standby mode. It is not initialized by a manual reset.

- Bit 7—Transmit Data Register Empty (TDRE): Indicates that the SCI has loaded transmit data from TDR into TSR and new serial transmit data can be written in TDR.

Bit 7: TDRE	Description
0	TDR contains valid transmit data [Clearing conditions] <ul style="list-style-type: none"> <li>• When 0 is written to TDRE after reading TDRE = 1</li> <li>• When the DMAC writes data in TDR</li> </ul>
1	TDR does not contain valid transmit data (Initial value) [Setting conditions] <ul style="list-style-type: none"> <li>• Power-on reset, hardware standby mode, or software standby mode</li> <li>• When the TE bit in SCR is 0</li> <li>• When data is transferred from TDR to TSR, enabling new data to be written in TDR</li> </ul>

- Bit 6—Receive Data Register Full (RDRF): Indicates that RDR contains received data.

Bit 6: RDRF	Description
0	RDR does not contain valid receive data (Initial value) [Clearing conditions] <ul style="list-style-type: none"> <li>• Power-on reset, hardware standby mode, or software standby mode</li> <li>• When 0 is written to RDRF after reading RDRF = 1</li> <li>• When the DMAC reads data from RDR</li> </ul>
1	RDR contains valid received data [Setting condition] RDRF is set to 1 when serial data is received normally and transferred from RSR to RDR

Note: RDR and RDRF are not affected by detection of receive errors or by clearing of the RE bit to 0 in the serial control register. They retain their previous contents. If RDRF is still set to 1 when reception of the next data ends, an overrun error (ORER) occurs and the receive data is lost.

- Bit 5—Overrun Error (ORER): Indicates that data reception ended abnormally due to an overrun error.

Bit 5: ORER	Description
0	Receiving is in progress or has ended normally (Initial value) Clearing the RE bit to 0 in the serial control register does not affect the ORER bit, which retains its previous value. [Clearing conditions] <ul style="list-style-type: none"> <li>• Power-on reset, hardware standby mode, or software standby mode</li> <li>• When 0 is written to ORER after reading ORER = 1</li> </ul>
1	A receive overrun error occurred RDR continues to hold the data received before the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while ORER is set to 1. In synchronous mode, serial transmitting is disabled. [Setting condition] ORER is set to 1 if reception of the next serial data ends when RDRF is set to 1

- Bit 4—Framing Error (FER): Indicates that data reception ended abnormally due to a framing error in asynchronous mode.

Bit 4: FER	Description
0	<p>Receiving is in progress or has ended normally (Initial value)</p> <p>Clearing the RE bit to 0 in the serial control register does not affect the FER bit, which retains its previous value.</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Power-on reset, hardware standby mode, or software standby mode</li> <li>• When 0 is written to FER after reading FER = 1</li> </ul>
1	<p>A receive framing error occurred</p> <p>When the stop bit length is two bits, only the first bit is checked to see if it is a 1. The second stop bit is not checked. When a framing error occurs, the SCI transfers the receive data into RDR but does not set RDRF. Serial receiving cannot continue while FER is set to 1. In synchronous mode, serial transmitting is also disabled.</p> <p>[Setting condition]</p> <p>FER is set to 1 if the stop bit at the end of receive data is checked and found to be 0</p>

- Bit 3—Parity Error (PER): Indicates that data reception (with parity) ended abnormally due to a parity error in asynchronous mode.

Bit 3: PER	Description
0	<p>Receiving is in progress or has ended normally (Initial value)</p> <p>Clearing the RE bit to 0 in the serial control register does not affect the PER bit, which retains its previous value.</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Power-on reset, hardware standby mode, or software standby mode</li> <li>• When 0 is written to PER after reading PER = 1</li> </ul>
1	<p>A receive parity error occurred</p> <p>When a parity error occurs, the SCI transfers the receive data into RDR but does not set RDRF. Serial receiving cannot continue while PER is set to 1.</p> <p>[Setting condition]</p> <p>PER is set to 1 if the number of 1s in receive data, including the parity bit, does not match the even or odd parity setting of the parity mode bit (<math>O/\bar{E}</math>) in the serial mode register (SMR)</p>

- Bit 2—Transmit End (TEND): Indicates that when the last bit of a serial character was transmitted, TDR did not contain valid data, so transmission has ended. TEND is a read-only bit and cannot be written.

Bit 2: TEND	Description
0	Transmission is in progress [Clearing conditions] <ul style="list-style-type: none"> <li>• When 0 is written to TDRE after reading TDRE = 1</li> <li>• When the DMAC writes data in TDR</li> </ul>
1	End of transmission (Initial value) [Setting conditions] <ul style="list-style-type: none"> <li>• Power-on reset, hardware standby mode, or software standby mode</li> <li>• When the TE bit in SCR is 0</li> <li>• If TDRE = 1 when the last bit of a one-byte serial transmit character is transmitted</li> </ul>

- Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in receive data when a multiprocessor format is selected for receiving in asynchronous mode. MPB is a read-only bit and cannot be written.

Bit 1: MPB	Description
0	Multiprocessor bit value in receive data is 0 (Initial value) If RE is cleared to 0 when a multiprocessor format is selected, the MPB retains its previous†value.
1	Multiprocessor bit value in receive data is 1

- Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit added to transmit data when a multiprocessor format is selected for transmitting in asynchronous mode. The MPBT setting is ignored in synchronous mode, when a multiprocessor format is not selected, or when the SCI is not transmitting.

Bit 0: MPBT	Description
0	Multiprocessor bit value in transmit data is 0 (Initial value)
1	Multiprocessor bit value in transmit data is 1



## 15.2.8 Bit Rate Register (BRR)

Bit:	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The bit rate register (BRR) 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 (SMR), determines the serial transmit/receive bit rate.

The CPU can always read and write to BRR. BRR is initialized to H'FF by a power-on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode. Each channel has independent baud rate generator control, so different values can be set for each channel.

Table 15.3 lists examples of BRR settings in the asynchronous mode; table 15.4 lists examples of BRR settings in the clock synchronous mode.

**Table 15.3 Bit Rates and BRR Settings in Asynchronous Mode**

Bit Rate (Bits/s)	P $\phi$ (MHz)								
	10			11.0592			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	-0.25	2	195	0.19	2	212	0.03
150	2	129	0.16	2	143	0.00	2	155	0.16
300	2	64	0.16	2	71	0.00	2	77	0.16
600	1	129	0.16	1	143	0.00	1	155	0.16
1200	1	64	0.16	1	71	0.00	1	77	0.16
2400	0	129	0.16	0	143	0.00	0	155	0.16
4800	0	64	0.16	0	71	0.00	0	77	0.16
9600	0	32	-1.36	0	35	0.00	0	28	0.16
14400	0	21	-1.36	0	23	0.00	0	25	0.16
19200	0	15	1.73	0	19	0.00	0	19	-2.34
28800	0	10	-1.36	0	11	0.00	0	12	0.16
31250	0	9	0.00	0	10	0.54	0	11	0.00
38400	0	7	1.73	0	8	0.00	0	9	-2.34

**Table 15.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)**

Bit Rate (Bits/s)	P <sub>φ</sub> (MHz)								
	12.288			14			14.7456		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	-0.17	3	64	0.70
150	2	159	0.00	2	181	0.16	2	191	0.00
300	2	79	0.00	2	90	0.16	2	95	0.00
600	1	159	0.00	1	181	0.16	1	191	0.00
1200	1	79	0.00	1	90	0.16	1	95	0.00
2400	0	159	0.00	0	181	0.16	0	191	0.00
4800	0	79	0.00	0	90	0.16	0	95	0.00
9600	0	39	0.00	0	45	-0.93	0	47	0.00
14400	0	26	-1.23	0	29	1.27	0	31	0.00
19200	0	19	0.00	0	22	-0.93	0	23	0.00
28800	0	12	2.56	0	14	1.27	0	15	0.00
31250	0	11	2.40	0	13	0.00	0	14	-1.70
38400	0	9	0.00	0	10	3.57	0	11	0.00

**Table 15.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)**

Bit Rate (Bits/s)	P <sub>φ</sub> (MHz)								
	16			17.2032			18		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	70	0.03	3	75	0.48	3	79	-0.12
150	2	207	0.16	2	223	0.00	2	233	0.16
300	2	103	0.16	2	111	0.00	2	116	0.16
600	1	207	0.16	1	223	0.00	1	233	0.16
1200	1	103	0.16	1	111	0.00	1	116	0.16
2400	0	207	0.16	0	223	0.00	0	233	0.16
4800	0	103	0.16	0	111	0.00	0	116	0.16
9600	0	51	0.16	0	55	0.00	0	58	-0.69
14400	0	34	-0.79	0	36	0.90	0	38	0.16
19200	0	25	0.16	0	27	0.00	0	28	1.02
28800	0	16	2.12	0	18	-1.75	0	19	-2.34
31250	0	15	0.00	0	16	1.20	0	17	0.00
38400	0	12	0.16	0	13	0.00	0	14	-2.34

**Table 15.3 Bit Rates and BRR Settings in Asynchronous Mode (cont)**

Bit Rate (Bits/s)	$\phi$ (MHz)								
	18.432			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	81	-0.22	3	86	0.31	3	88	-0.25
150	2	239	0.00	2	255	0.00	3	64	0.16
300	2	119	0.00	2	127	0.00	2	129	0.16
600	1	239	0.00	1	255	0.00	2	64	0.16
1200	1	119	0.00	1	127	0.00	1	129	0.16
2400	0	239	0.00	0	255	0.00	1	64	0.16
4800	0	119	0.00	0	127	0.00	0	129	0.16
9600	0	59	0.00	0	63	0.00	0	64	0.16
14400	0	39	0.00	0	42	-0.78	0	42	0.94
19200	0	29	0.00	0	31	0.00	0	32	-1.36
28800	0	19	0.00	0	20	1.59	0	21	-1.36
31250	0	17	2.40	0	19	-1.70	0	19	0.00
38400	0	14	0.00	0	15	0.00	0	15	1.73

**Table 15.4 Bit Rates and BRR Settings in Synchronous Mode**

Bit Rate (Bits/s)	P <sub>φ</sub> (MHz)							
	10		12		16		20	
	n	N	n	N	n	N	n	N
250	–	–	3	187	3	249		
500	–	–	3	93	3	124	–	–
1 k	–	–	2	187	2	249	–	–
2.5 k	1	249	2	74	2	99	2	124
5 k	1	124	1	149	1	199	2	249
10 k	0	249	1	74	1	99	1	124
25 k	0	99	0	119	0	159	1	199
50 k	0	49	0	59	0	79	0	99
100 k	0	24	0	29	0	39	0	49
250 k	0	9	0	11	0	15	0	19
500 k	0	4	0	5	0	7	0	9
1 M			0	2	0	3	0	4
2.5 M	0	0*	0	0*	–	–	0	1
5 M							0	0*

Note: Settings with an error of 1% or less are recommended.

Legend

Blank: No setting available

–: Setting possible, but error occurs

\*: Continuous transmission/reception not possible

The BRR setting is calculated as follows:

Asynchronous mode:

$$N = \frac{P_{\phi}}{64 \cdot 2^{2n-1} \cdot B} \cdot 10^6 - 1$$

Synchronous mode:

$$N = \frac{P_{\phi}}{8 \cdot 2^{2n-1} \cdot B} \cdot 10^6 - 1$$

- B: Bit rate (bits/s)  
 N: Baud rate generator BRR setting ( $0 \leq N \leq 255$ )  
 Pf: Peripheral module operating frequency (MHz) (1/2 of system clock)  
 n: Baud rate generator input clock ( $n = 0$  to 3)  
 (See the following table for the clock sources and value of n.)

n	Clock Source	SMR Settings	
		CKS1	CKS2
0	$P\phi$	0	0
1	$P\phi/4$	0	1
2	$P\phi/16$	1	0
3	$P\phi/64$	1	1

The bit rate error in asynchronous mode is calculated as follows:

$$\text{Error (\%)} = \left\{ \frac{P\phi \cdot 10^6}{(N + 1) \cdot B \cdot 64 \cdot 2^{2n-1}} - 1 \right\} \cdot 100$$

Table 15.5 indicates the maximum bit rates in asynchronous mode when the baud rate generator is being used for various frequencies. Tables 15.6 and 15.7 show the maximum rates for external clock input.

**Table 15.5 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (Asynchronous Mode)**

Pφ (MHz)	Maximum Bit Rate (Bits/s)	Settings	
		n	N
10	312500	0	0
11.0592	345600	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
18.432	576000	0	0
19.6608	614400	0	0
20	625000	0	0

**Table 15.6 Maximum Bit Rates during External Clock Input (Asynchronous Mode)**

<b>P<math>\phi</math> (MHz)</b>	<b>External Input Clock (MHz)</b>	<b>Maximum Bit Rate (Bits/s)</b>
10	2.5000	156250
11.0592	2.7648	172800
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
18.432	4.6080	288000
19.6608	4.9152	307200
20	5.0000	312500

**Table 15.7 Maximum Bit Rates during External Clock Input (Clock Synchronous Mode)**

<b>P<math>\phi</math> (MHz)</b>	<b>External Input Clock (MHz)</b>	<b>Maximum Bit Rate (Bits/s)</b>
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3

**15.2.9 Serial Direction Control Register (SDCR)**

Bit:	7	6	5	4	3	2	1	0
	–	–	–	–	DIR	–	–	–
Initial value:	1	1	1	1	0	0	1	0
R/W:	R	R	R	R	R/W	R	R	R

The DIR bit in the serial direction control register (SDCR) selects LSB-first or MSB-first transfer. With an 8-bit data length, LSB-first/MSB-first selection is available regardless of the communication mode. With a 7-bit data length, LSB-first transfer must be selected. The description in this section assumes LSB-first transfer.

SDCR is initialized to H'F2 by a power-on reset and in the hardware standby mode. It is not initialized by a manual reset and in software standby mode.

- Bits 7–4—Reserved: The write value should always be 1. If 0 is written to these bits, correct operation cannot be guaranteed.
- Bit 3—Data Transfer Direction (DIR): Selects the serial/parallel conversion format. Valid for an 8-bit transmit/receive format.

Bit 3: DIR	Description
0	TDR contents are transmitted in LSB-first order (Initial value) Receive data is stored in RDR in LSB-first order
1	TDR contents are transmitted in MSB-first order Receive data is stored in RDR in MSB-first order

- Bit 2—Reserved: The write value should always be 0. If 1 is written to this bit, correct operation cannot be guaranteed.
- Bit 1—Reserved: This bit is always read as 1, and cannot be modified.
- Bit 0—Reserved: The write value should always be 0. If 1 is written to this bit, correct operation cannot be guaranteed.

### 15.2.10 Inversion of SCK Pin Signal

The signal input from the SCK pin and the signal output from the SCK pin can be inverted by means of a port control register setting. See section 21, Pin function Controller (PFC), for details.



## 15.3 Operation

### 15.3.1 Overview

For serial communication, the SCI has an asynchronous mode in which characters are synchronized individually, and a synchronous mode in which communication is synchronized with clock pulses. Asynchronous synchronous mode and the transmission format are selected in the serial mode register (SMR), as shown in table 15.8. The SCI clock source is selected by the  $\overline{C/A}$  bit in the serial mode register (SMR) and the CKE1 and CKE0 bits in the serial control register (SCR), as shown in table 15.9.

#### Asynchronous Mode:

- Data length is selectable: seven or eight bits.
- Parity and multiprocessor bits are selectable, as well as the stop bit length (one or two bits). These selections determine the transmit/receive format and character length.
- In receiving, it is possible to detect framing errors (FER), parity errors (PER), overrun errors (ORER), and the break state.
- 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 clock, and can output a clock with a frequency matching the bit rate.
  - When an external clock is selected, the external clock input must have a frequency 16 times the bit rate. (The on-chip baud rate generator is not used.)

#### Synchronous Mode:

- The communication format has a fixed 8-bit data length.
- In receiving, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCI clock source.
  - When an internal clock is selected, the SCI operates using the on-chip baud rate generator clock, 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 15.8 Serial Mode Register Settings and SCI Communication Formats**

Mode	SMR Settings					SCI Communication Format										
	Bit 7 C/ $\bar{A}$	Bit 6 CHR	Bit 5 PE	Bit 2 MP	Bit 3 STOP	Data Length	Parity Bit	Multipro-cessor Bit	Stop Bit Length							
Asynchronous	0	0	0	0	0	8-bit	Absent	Absent	1 bit							
					1				2 bits							
					0				Present	1 bit						
					1					2 bits						
					1				0	7-bit	Absent	1 bit				
					1				2 bits							
					1				1	Present	1 bit					
					1				2 bits							
					Asynchronous (multiprocessor format)				0	*	*	1	8-bit	Absent	Present	1 bit
												1				2 bits
1	7-bit	1 bit														
1	2 bits															
Synchronous	1	*	*	*	8-bit		Absent	None								

Note: Asterisks (\*) in the table indicate don't-care bits.

**Table 15.9 SMR and SCR Settings and SCI Clock Source Selection**

Mode	SMR	SCR Settings		SCI Transmit/Receive Clock		
	Bit 7 C/ $\bar{A}$	Bit 1 CKE1	Bit 0 CKE0	Clock Source	SCK Pin Function*	
Asynchronous	0	0	0	Internal	SCI does not use the SCK pin	
			1			Outputs a clock with frequency matching the bit rate
			1	0	External	Inputs a clock with frequency 16 times the bit rate
			1	1		
Synchronous	1	0	0	Internal	Outputs the serial clock or the inverted serial clock	
			1			
			1	0	External	Inputs the serial clock or the inverted serial clock
			1	1		

Note: \* Select the function in combination with the pin function controller (PFC).

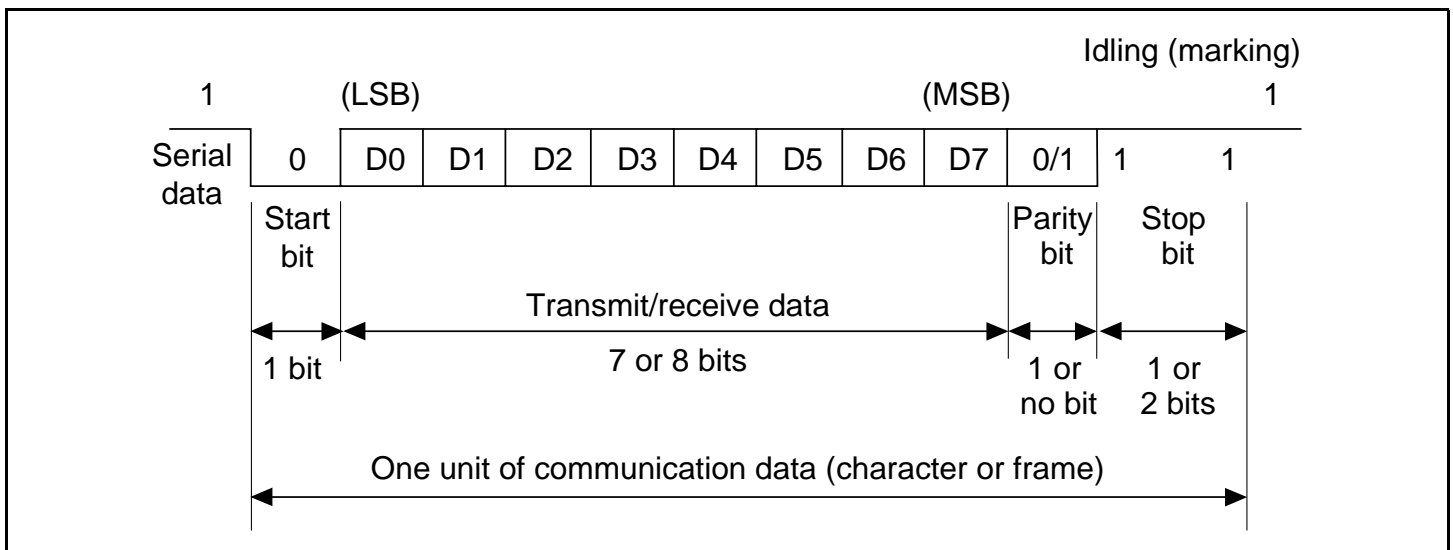
### 15.3.2 Operation in Asynchronous Mode

In asynchronous mode, each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronized one character at a time.

The transmitting and receiving sections of the SCI are independent, so full duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 15.2 shows the general format of asynchronous serial communication. In asynchronous serial communication, the communication line is normally held in the marking (high) state. The SCI monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCI synchronizes on 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 15.2 Data Format in Asynchronous Communication (Example: 8-bit Data with Parity and Two Stop Bits)**

**Transmit/Receive Formats:** Table 15.10 shows the 12 communication formats that can be selected in asynchronous mode. The format is selected by settings in the serial mode register (SMR).

**Table 15.10 Serial Communication Formats (Asynchronous Mode)**

SMR Bits				Serial Transmit/Receive Format and Frame Length												
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12	
0	0	0	0	START	8-bit data							STOP				
0	0	0	1	START	8-bit data							STOP	STOP			
0	1	0	0	START	8-bit data							P	STOP			
0	1	0	1	START	8-bit data							P	STOP	STOP		
1	0	0	0	START	7-bit data						STOP					
1	0	0	1	START	7-bit data						STOP	STOP				
1	1	0	0	START	7-bit data						P	STOP				
1	1	0	1	START	7-bit data						P	STOP	STOP			
0	-	1	0	START	8-bit data							MPB	STOP			
0	-	1	1	START	8-bit data							MPB	STOP	STOP		
1	-	1	0	START	7-bit data						MPB	STOP				
1	-	1	1	START	7-bit data						MPB	STOP	STOP			

**Legend**

- START: Start bit
- STOP: Stop bit
- P: Parity bit
- MPB: Multiprocessor bit

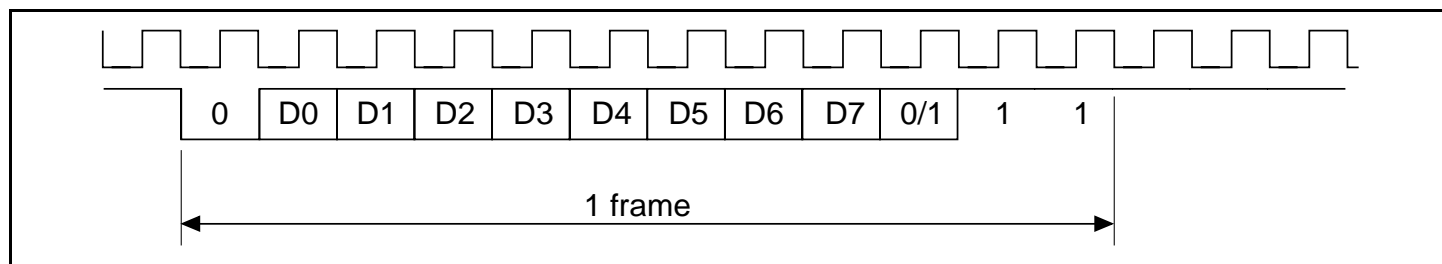
**Note**

-: Don't-care bits.

**Clock:** An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the  $C/\bar{A}$  bit in the serial mode register (SMR) and bits CKE1 and CKE0 in the serial control register (SCR) (table 15.9).

When an external clock is input at the SCK pin, it must have a frequency equal to 16 times the desired bit rate.

When the SCI operates on an internal clock, it can output a clock signal at the SCK pin. The frequency of this output clock is equal to the bit rate. The phase is aligned as in figure 15.3 so that the rising edge of the clock occurs at the center of each transmit data bit.



**Figure 15.3 Output Clock and Communication Data Phase Relationship (Asynchronous Mode)**

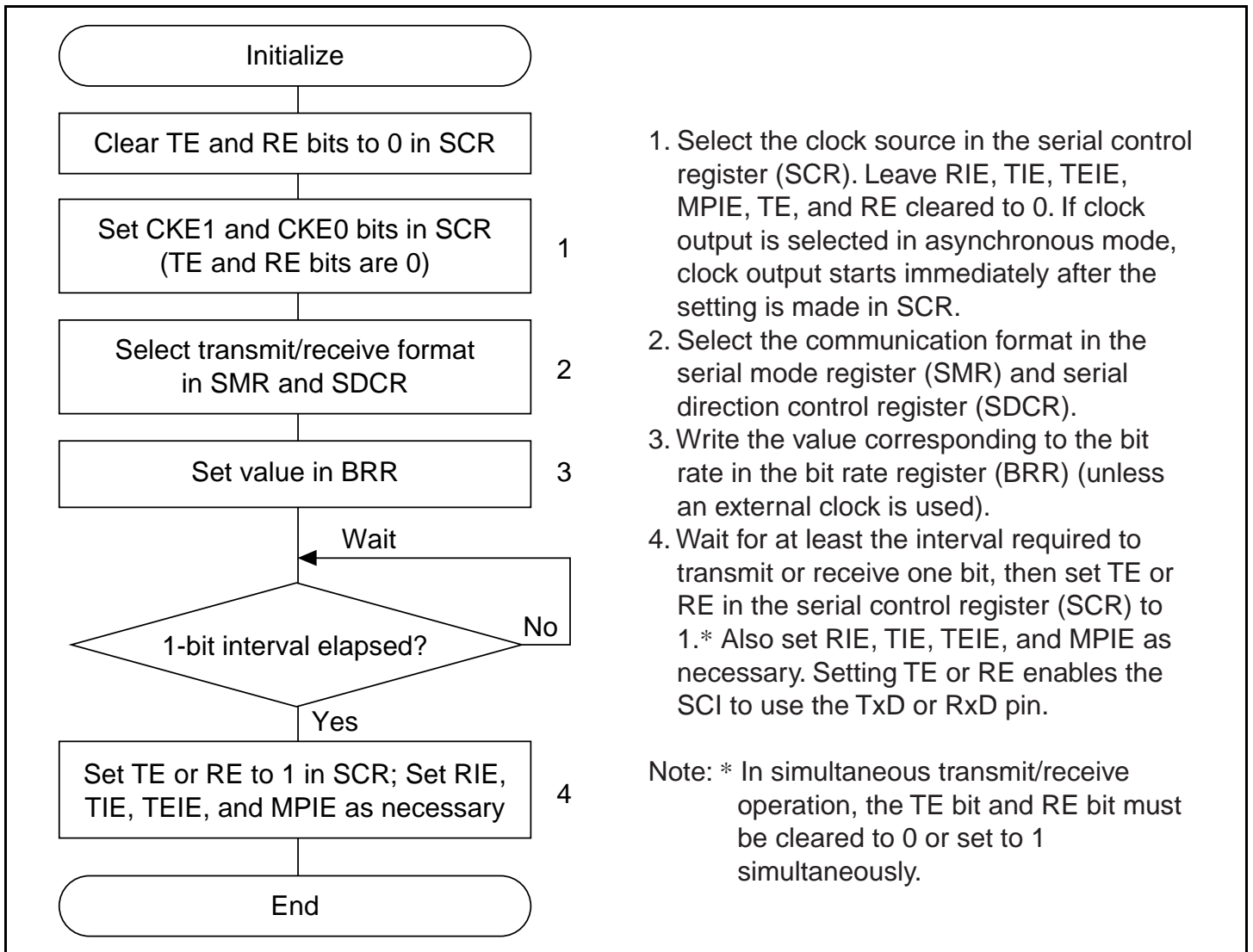
### Data Transmit/Receive Operation

**SCI Initialization (Asynchronous Mode):** Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as follows.

When changing the operation mode or communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

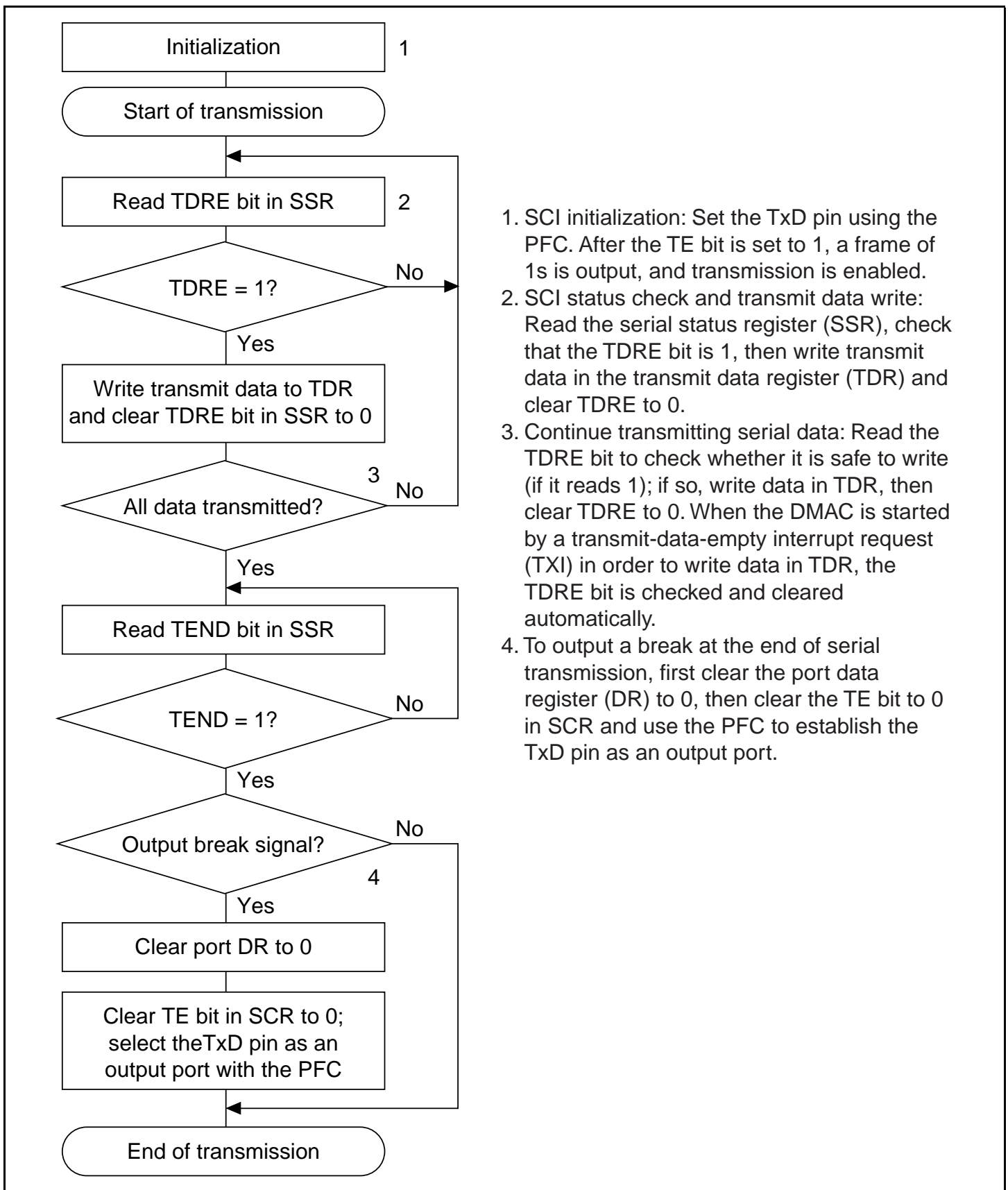
When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

Figure 15.4 is a sample flowchart for initializing the SCI. The procedure is as follows (the steps correspond to the numbers in the flowchart):



**Figure 15.4 Sample Flowchart for SCI Initialization**

**Transmitting Serial Data (Asynchronous Mode):** Figure 15.5 shows a sample flowchart for transmitting serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart):



1. SCI initialization: Set the TxD pin using the PFC. After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0.
3. Continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in TDR, then clear TDRE to 0. When the DMAC is started by a transmit-data-empty interrupt request (TXI) in order to write data in TDR, the TDRE bit is checked and cleared automatically.
4. To output a break at the end of serial transmission, first clear the port data register (DR) to 0, then clear the TE bit to 0 in SCR and use the PFC to establish the TxD pin as an output port.

**Figure 15.5 Sample Flowchart for Transmitting Serial Data**

In transmitting serial data, the SCI operates as follows:

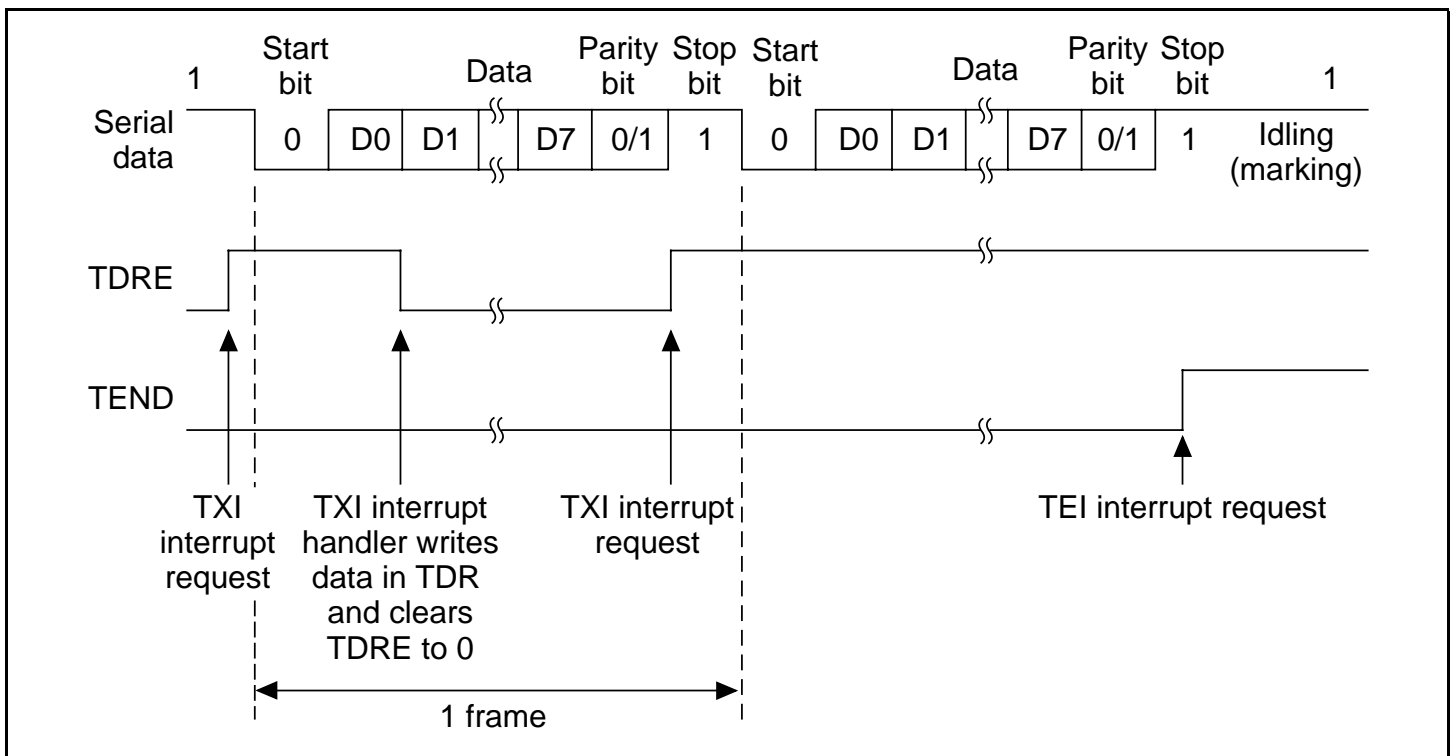
1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0, the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).
2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) is set to 1 in SCR, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

Serial transmit data is transmitted in the following order from the TxD pin:

- a. Start bit: one 0-bit is output.
  - b. Transmit data: seven or eight bits of data are output, LSB first.
  - c. Parity bit or multiprocessor bit: one parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.
  - d. Stop bit: one or two 1-bits (stop bits) are output.
  - e. Marking: output of 1-bits continues until the start bit of the next transmit data.
3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads new data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit to 1 in SSR, outputs the stop bit, then continues output of 1-bits (marking). If the transmit-end interrupt enable bit (TEIE) in SCR is set to 1, a transmit-end interrupt (TEI) is requested.

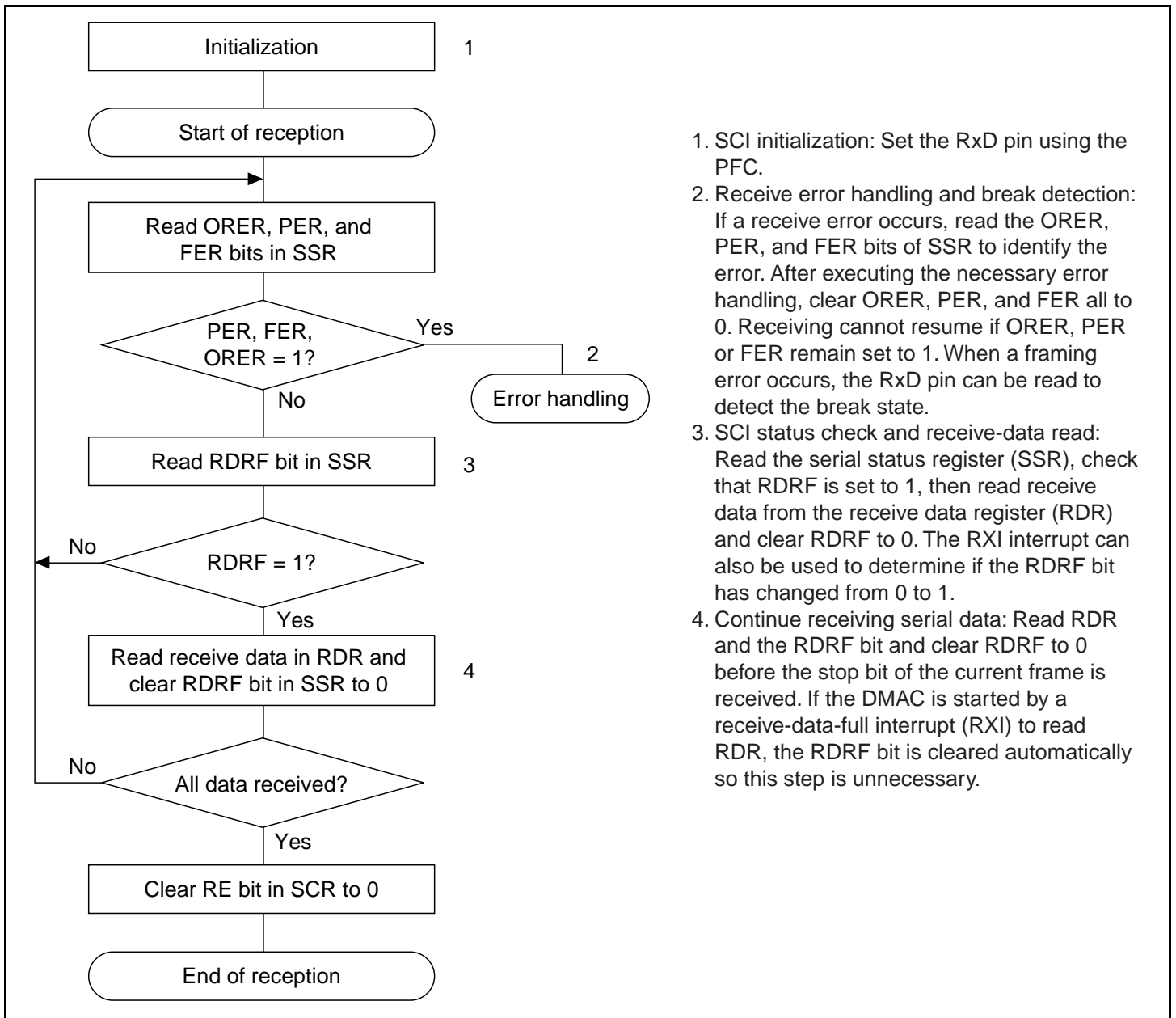
Figure 15.6 shows an example of SCI transmit operation in asynchronous mode.





**Figure 15.6 SCI Transmit Operation in Asynchronous Mode  
(Example: 8-Bit Data with Parity and One Stop Bit)**

**Receiving Serial Data (Asynchronous Mode):** Figures 15.7 and 15.8 show a sample flowchart for receiving serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart).



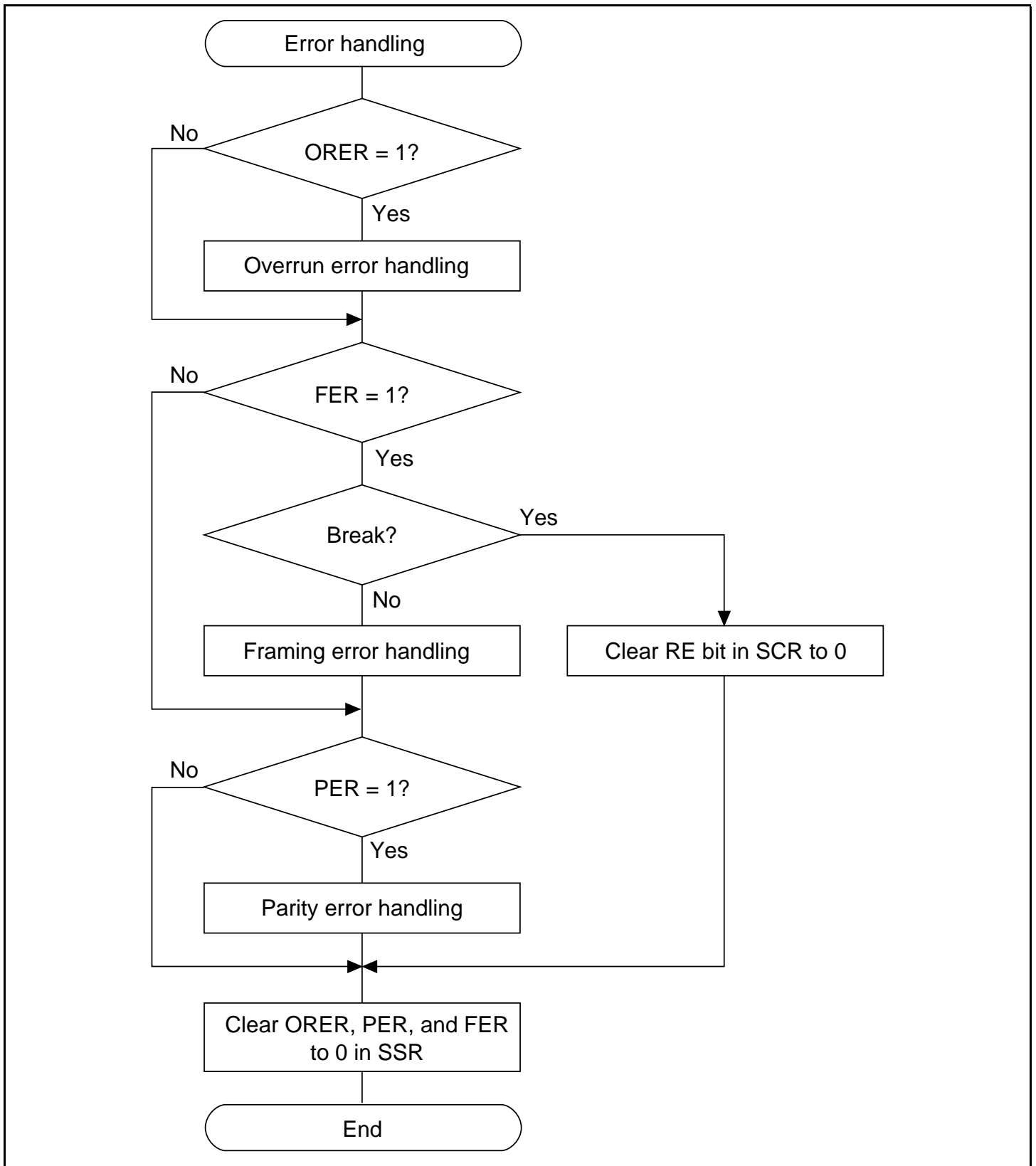
1. SCI initialization: Set the RxD pin using the PFC.

2. Receive error handling and break detection: If a receive error occurs, read the ORER, PER, and FER bits of SSR to identify the error. After executing the necessary error handling, clear ORER, PER, and FER all to 0. Receiving cannot resume if ORER, PER or FER remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.

3. SCI status check and receive-data read: Read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.

4. Continue receiving serial data: Read RDR and the RDRF bit and clear RDRF to 0 before the stop bit of the current frame is received. If the DMAC is started by a receive-data-full interrupt (RXI) to read RDR, the RDRF bit is cleared automatically so this step is unnecessary.

**Figure 15.7 Sample Flowchart for Receiving Serial Data (1)**



**Figure 15.8 Sample Flowchart for Receiving Serial Data (2)**

In receiving, the SCI operates as follows:

1. The SCI monitors the communication line. When it detects a start bit (0), the SCI synchronizes internally and starts receiving.
2. Receive data is shifted into RSR in order from the LSB to the MSB.
3. The parity bit and stop bit are received. After receiving these bits, the SCI makes the following checks:
  - a. Parity check. The number of 1s in the receive data must match the even or odd parity setting of the O/E bit in SMR.
  - b. Stop bit check. The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.
  - c. Status check. RDRF must be 0 so that receive data can be loaded from RSR into RDR.

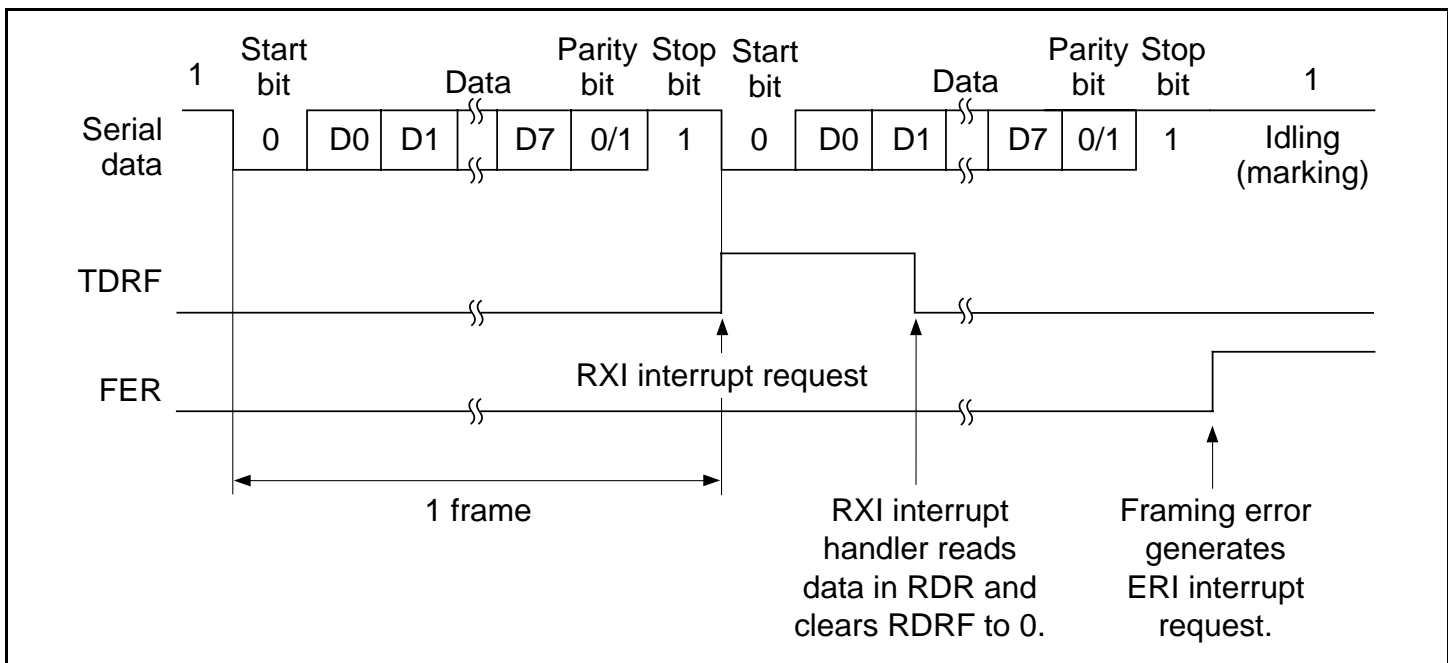
If the data passes these checks, the SCI sets RDRF to 1 and stores the receive data in RDR. If one of the checks fails (receive error), the SCI operates as indicated in table 15.11.

Note: When a receive error occurs, further receiving is disabled. While receiving, the RDRF bit is not set to 1, so be sure to clear the error flags.
4. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCR, the SCI requests a receive-data-full interrupt (RXI). If one of the error flags (ORER, PER, or FER) is set to 1 and the receive-data-full interrupt enable bit (RIE) in SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

**Table 15.11 Receive Error Conditions and SCI Operation**

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SSR	Receive data not loaded from RSR into RDR
Framing error	FER	Stop bit is 0	Receive data loaded from RSR into RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data loaded from RSR into RDR

Figure 15.9 shows an example of SCI receive operation in asynchronous mode.



**Figure 15.9 SCI Receive Operation  
(Example: 8-Bit Data with Parity and One Stop Bit)**

### 15.3.3 Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line for sending and receiving data. The processors communicate in the asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

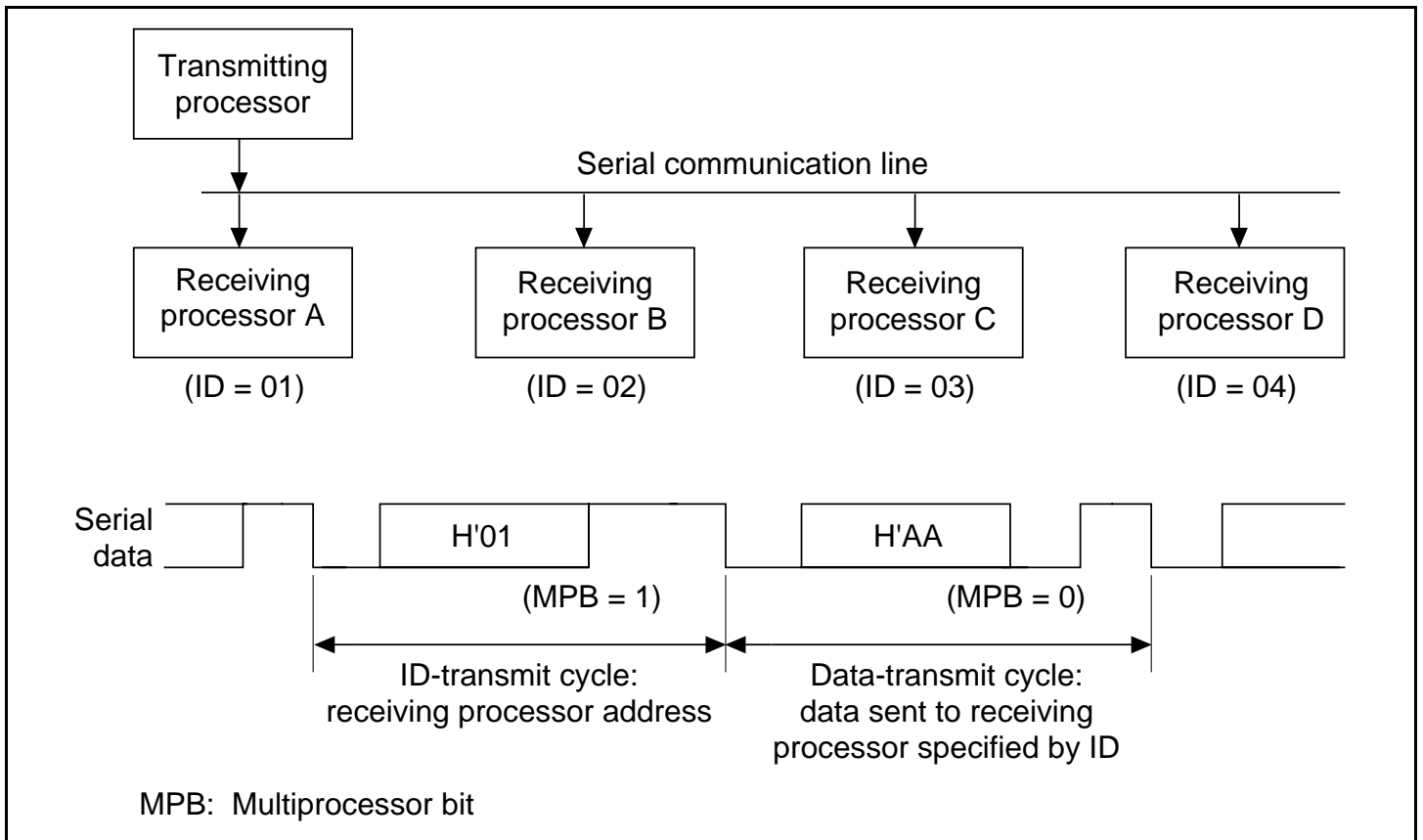
In multiprocessor communication, each receiving processor is addressed by a unique ID. A serial communication cycle consists of an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles. The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1. When they receive data with the multiprocessor bit set to 1, receiving processors compare the data with their IDs. The receiving processor with a matching ID continues to receive further incoming data. Processors with IDs not matching the received data skip further incoming data until they again receive data with the multiprocessor bit set to 1. Multiple processors can send and receive data in this way.

Figure 15.10 shows an example of communication among processors using the multiprocessor format.

**Communication Formats:** Four formats are available. Parity-bit settings are ignored when the multiprocessor format is selected. For details see table 15.8.

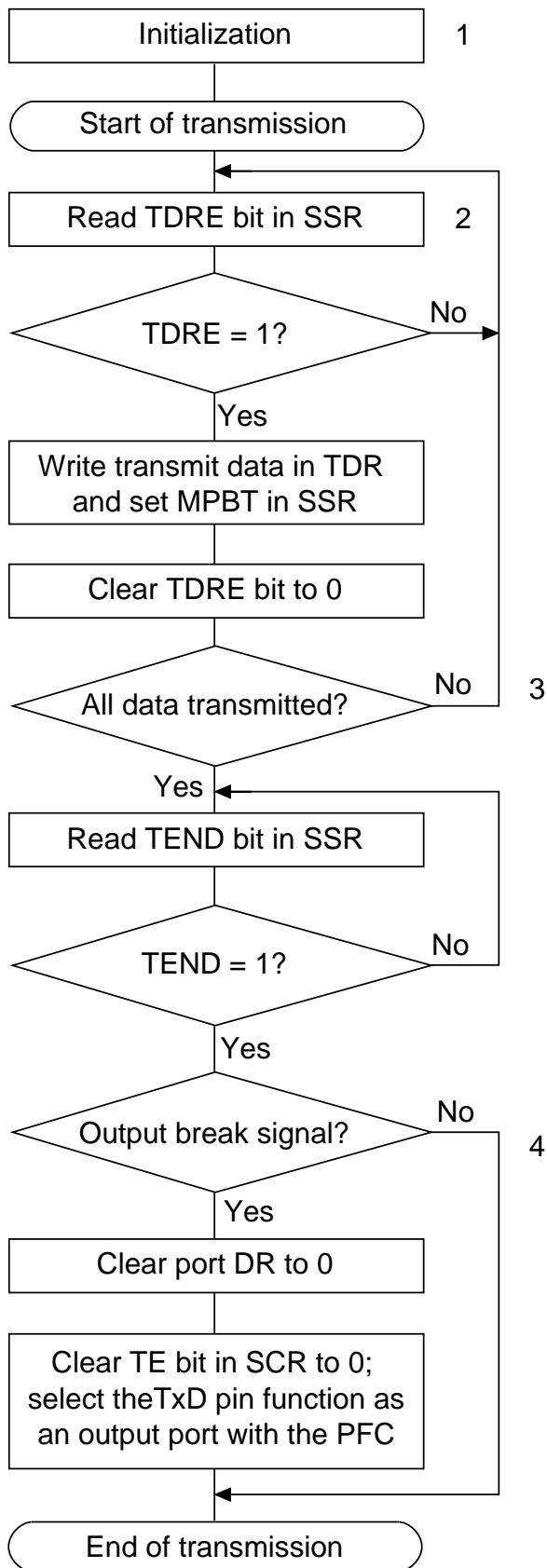
**Clock:** See the description in the asynchronous mode section.



**Figure 15.10 Communication among Processors Using Multiprocessor Format  
(Example: Sending Data H'AA to Receiving Processor A)**

### Data Transmit/Receive Operation

**Transmitting Multiprocessor Serial Data:** Figure 15.11 shows a sample flowchart for transmitting multiprocessor serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart):



1. SCI initialization: Set the TxD pin using the PFC. After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.
2. SCI status check and transmit data write: Read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR). Also set MPBT (multiprocessor bit transfer) to 0 or 1 in SSR. Finally, clear TDRE to 0.
3. Continue transmitting serial data: Read the TDRE bit to check whether it is safe to write (if it reads 1); if so, write data in TDR, then clear TDRE to 0. When the DMAC is started by a transmit-data-empty interrupt request (TXI) to write data in TDR, the TDRE bit is checked and cleared automatically.
4. Output a break at the end of serial transmission: Set the data register (DR) of the port to 0, then clear TE to 0 in SCR and set the TxD pin function as output port with the PFC.

**Figure 15.11 Sample Flowchart for Transmitting Multiprocessor Serial Data**

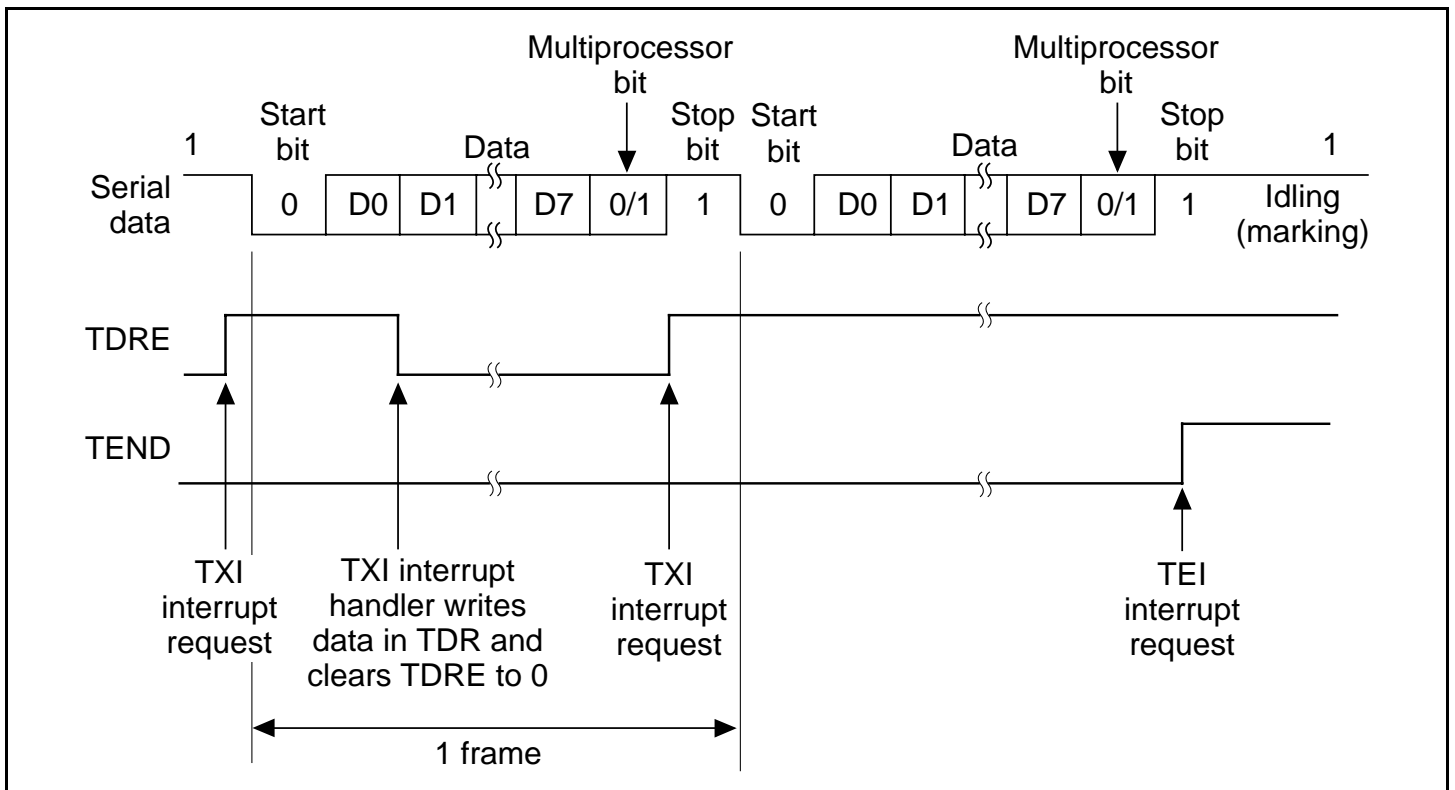
In transmitting serial data, the SCI operates as follows:

1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).
2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

Serial transmit data is transmitted in the following order from the TxD pin:

- a. Start bit: one 0-bit is output.
  - b. Transmit data: seven or eight bits are output, LSB first.
  - c. Multiprocessor bit: one multiprocessor bit (MPBT value) is output.
  - d. Stop bit: one or two 1-bits (stop bits) are output.
  - e. Marking: output of 1-bits continues until the start bit of the next transmit data.
3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, the SCI loads data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in SSR to 1, outputs the stop bit, then continues output of 1-bits in the marking state. If the transmit-end interrupt enable bit (TEIE) in SCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.

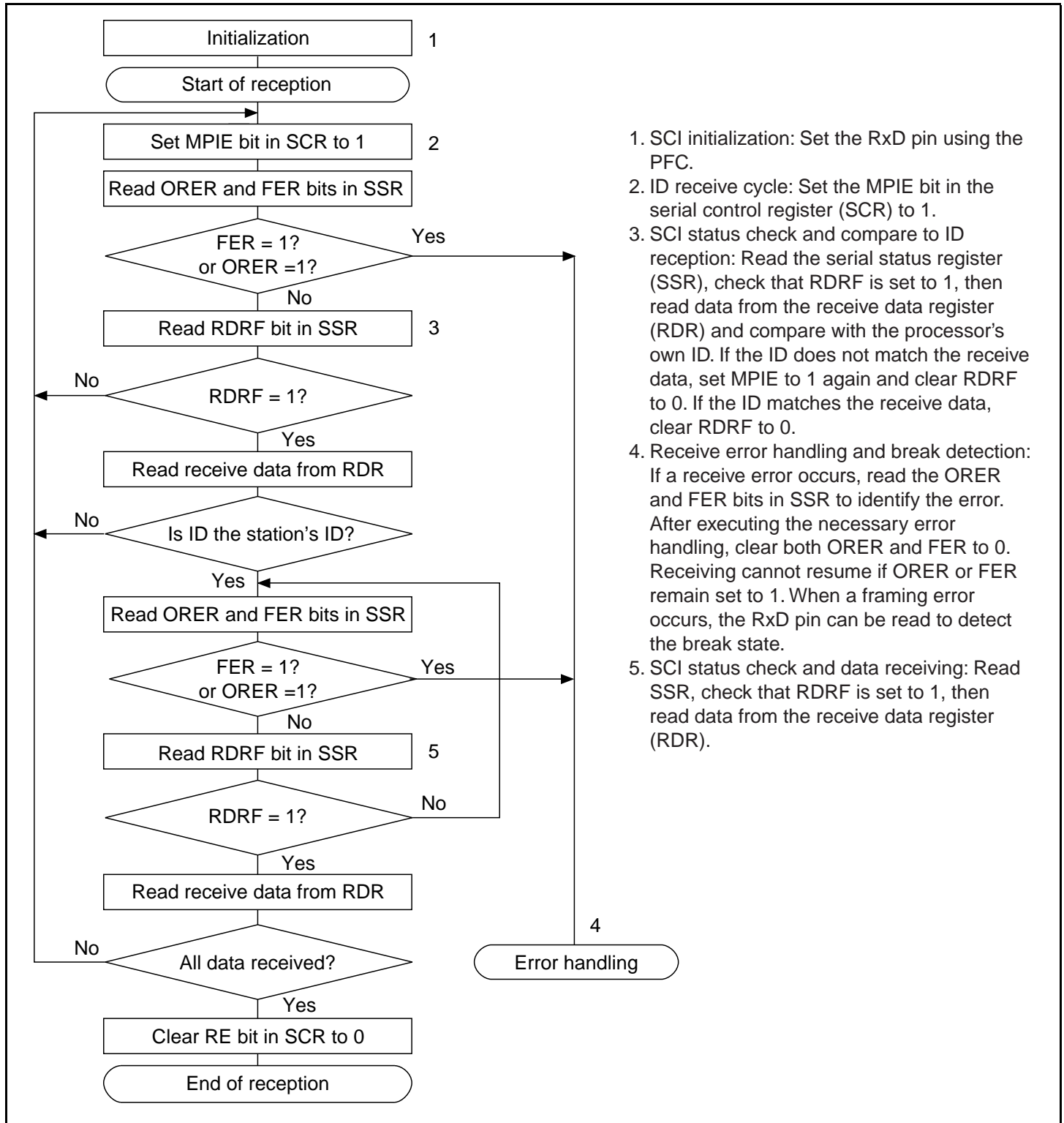
Figure 15.12 shows an example of SCI receive operation in the multiprocessor format.



**Figure 15.12 SCI Multiprocessor Transmit Operation  
(Example: 8-Bit Data with Multiprocessor Bit and One Stop Bit)**

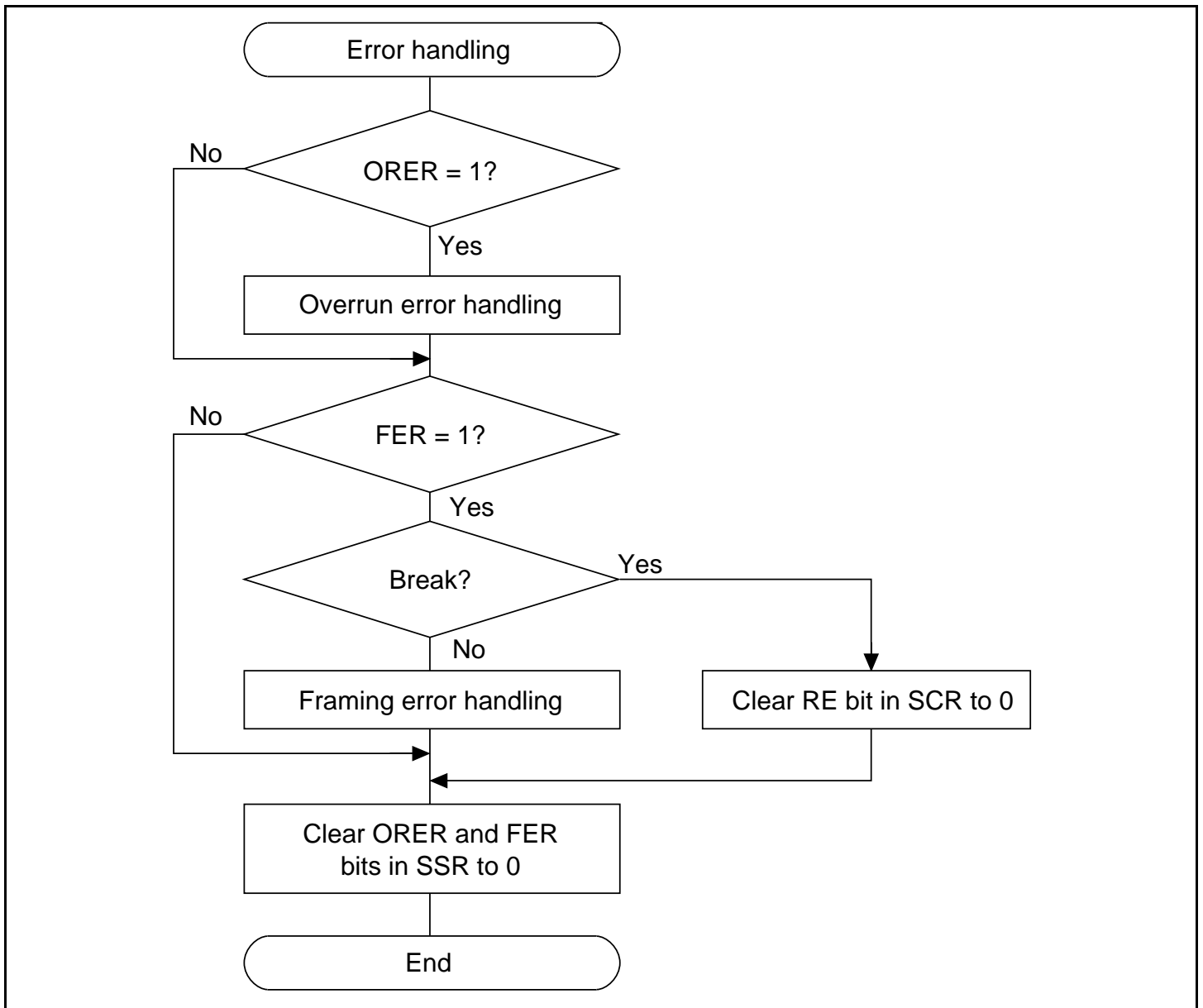


**Receiving Multiprocessor Serial Data:** Figure 15.13 shows a sample flowchart for receiving multiprocessor serial data. The procedure for receiving multiprocessor serial data is as follows (the steps correspond to the numbers in the flowchart):



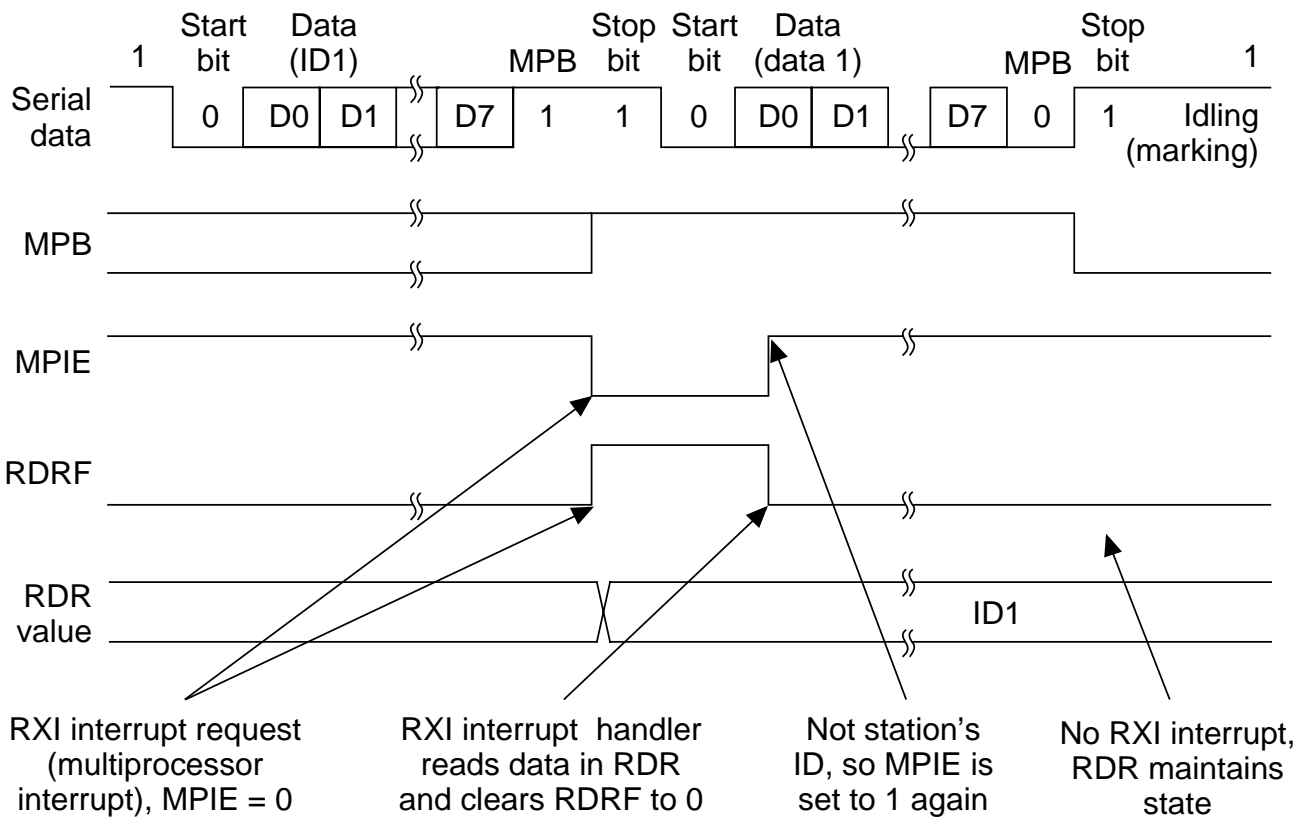
1. SCI initialization: Set the RxD pin using the PFC.
2. ID receive cycle: Set the MPIE bit in the serial control register (SCR) to 1.
3. SCI status check and compare to ID reception: Read the serial status register (SSR), check that RDRF is set to 1, then read data from the receive data register (RDR) and compare with the processor's own ID. If the ID does not match the receive data, set MPIE to 1 again and clear RDRF to 0. If the ID matches the receive data, clear RDRF to 0.
4. Receive error handling and break detection: If a receive error occurs, read the ORER and FER bits in SSR to identify the error. After executing the necessary error handling, clear both ORER and FER to 0. Receiving cannot resume if ORER or FER remain set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.
5. SCI status check and data receiving: Read SSR, check that RDRF is set to 1, then read data from the receive data register (RDR).

**Figure 15.13 Sample Flowchart for Receiving Multiprocessor Serial Data (1)**

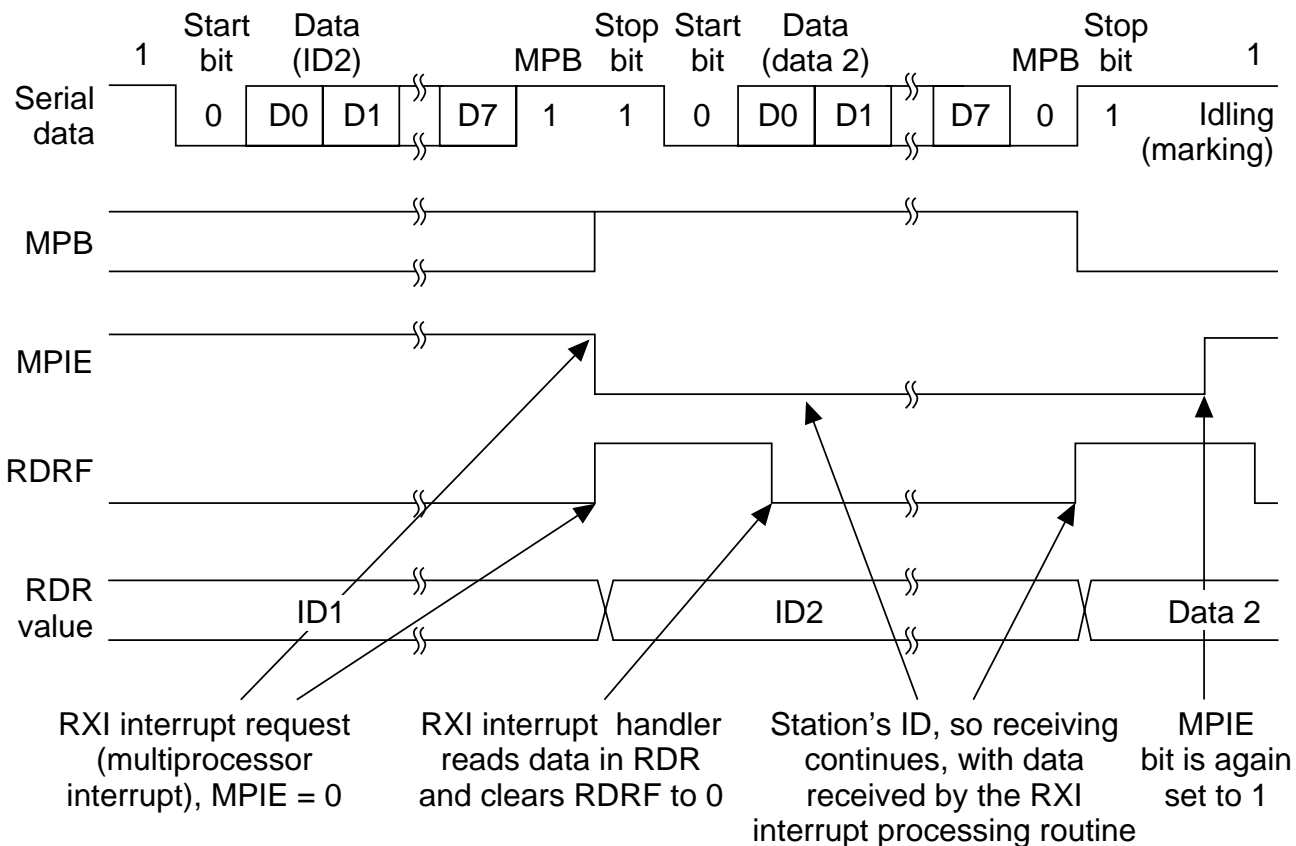


**Figure 15.14 Sample Flowchart for Receiving Multiprocessor Serial Data (2)**

Figure 15.15 shows examples of SCI receive operation using a multiprocessor format.



(A) ID Does Not Match



(B) ID Matches

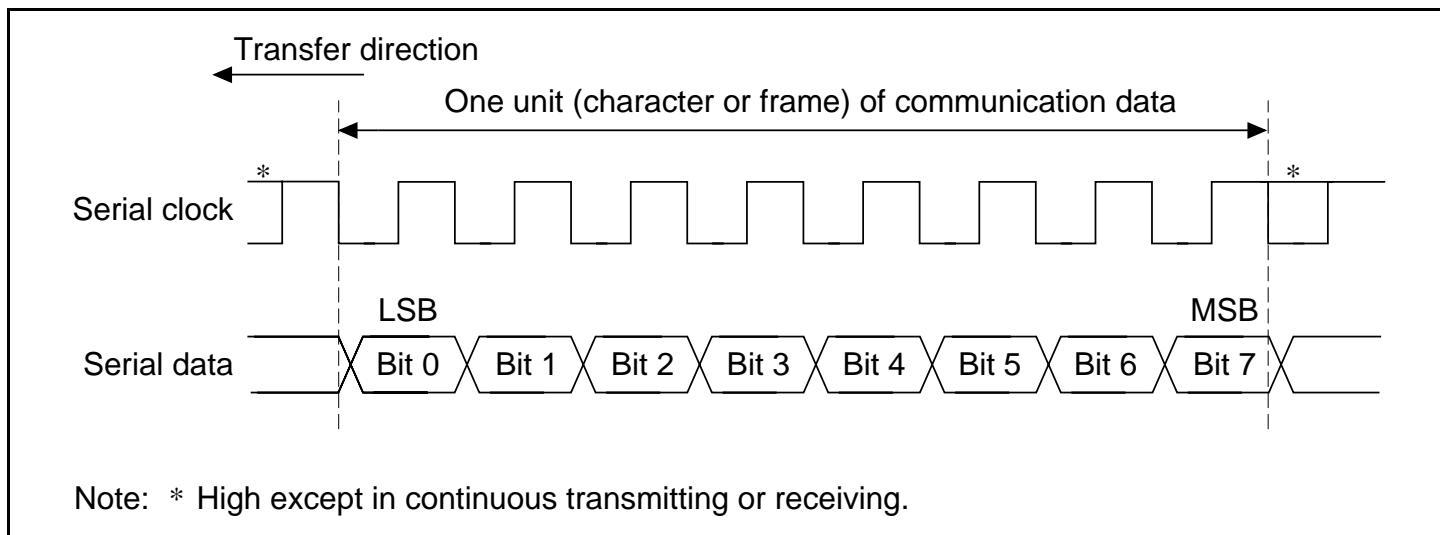
**Figure 15.15 SCI Receive Operation**  
**(Example: 8-Bit Data with Multiprocessor Bit and One Stop Bit)**

### 15.3.4 Synchronous Operation

In synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver are independent, so full duplex communication is possible while sharing the same clock. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 15.16 shows the general format in synchronous serial communication.



**Figure 15.16 Data Format in Synchronous Communication**

In synchronous serial communication, each data bit is output on the communication line from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock. In each character, the serial data bits are transmitted in order from the LSB (first) to the MSB (last). After output of the MSB, the communication line remains in the state of the MSB. In synchronous mode, the SCI transmits or receives data by synchronizing with the rise of the serial clock.

**Communication Format:** The data length is fixed at eight bits. No parity bit or multiprocessor bit can be added.

**Clock:** An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected as the SCI transmit/receive clock. The clock source is selected by the  $C/\bar{A}$  bit in the serial mode register (SMR) and bits CKE1 and CKE0 in the serial control register (SCR). See table 15.9.

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. An overrun error occurs only during the

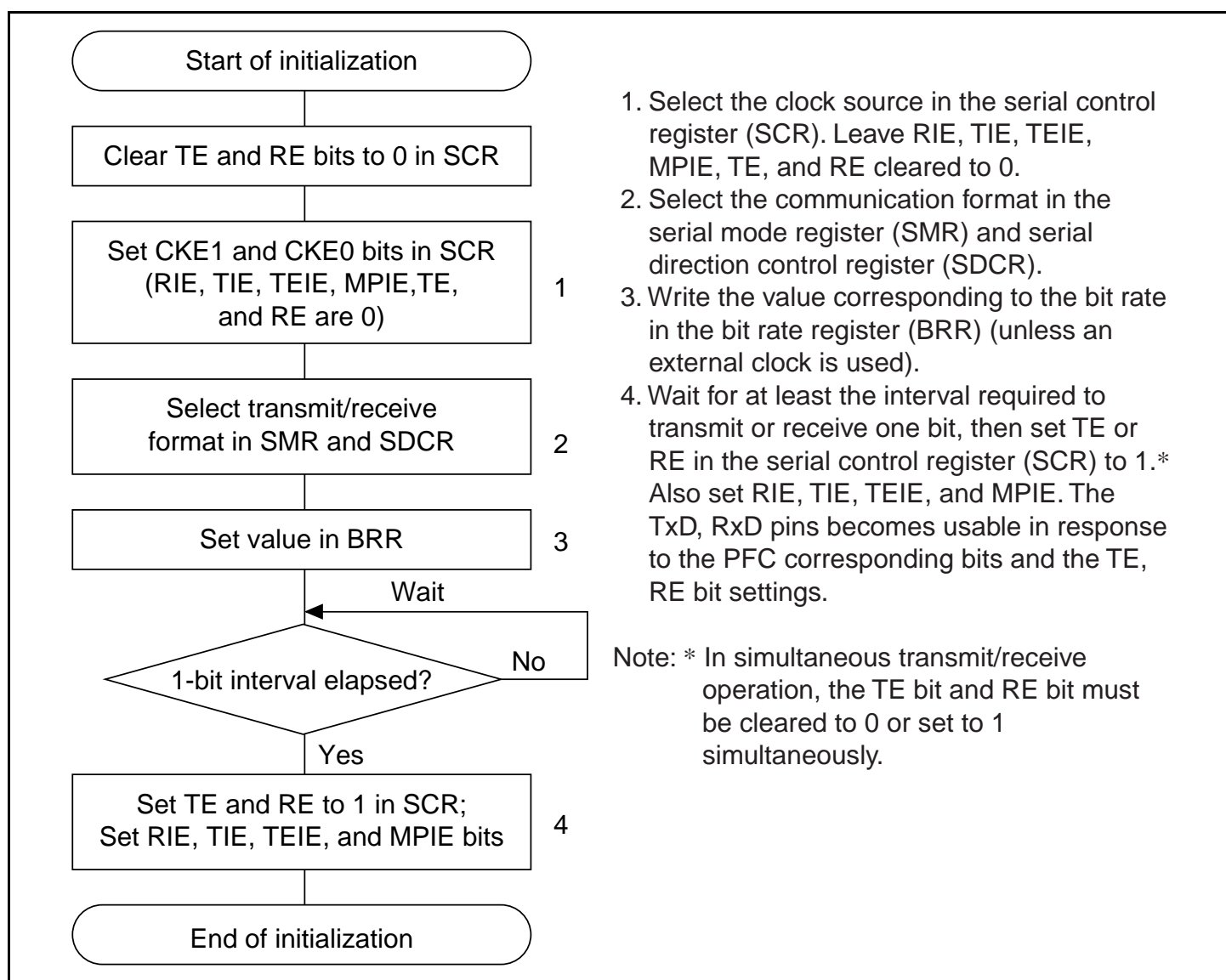
receive operation, and the serial clock is output until the RE bit is cleared to 0. To perform a receive operation in one-character units, select an external clock for the clock source.

## Transmitting and Receiving Data

**SCI Initialization (Synchronous Mode):** Before transmitting or receiving, software must clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as follows.

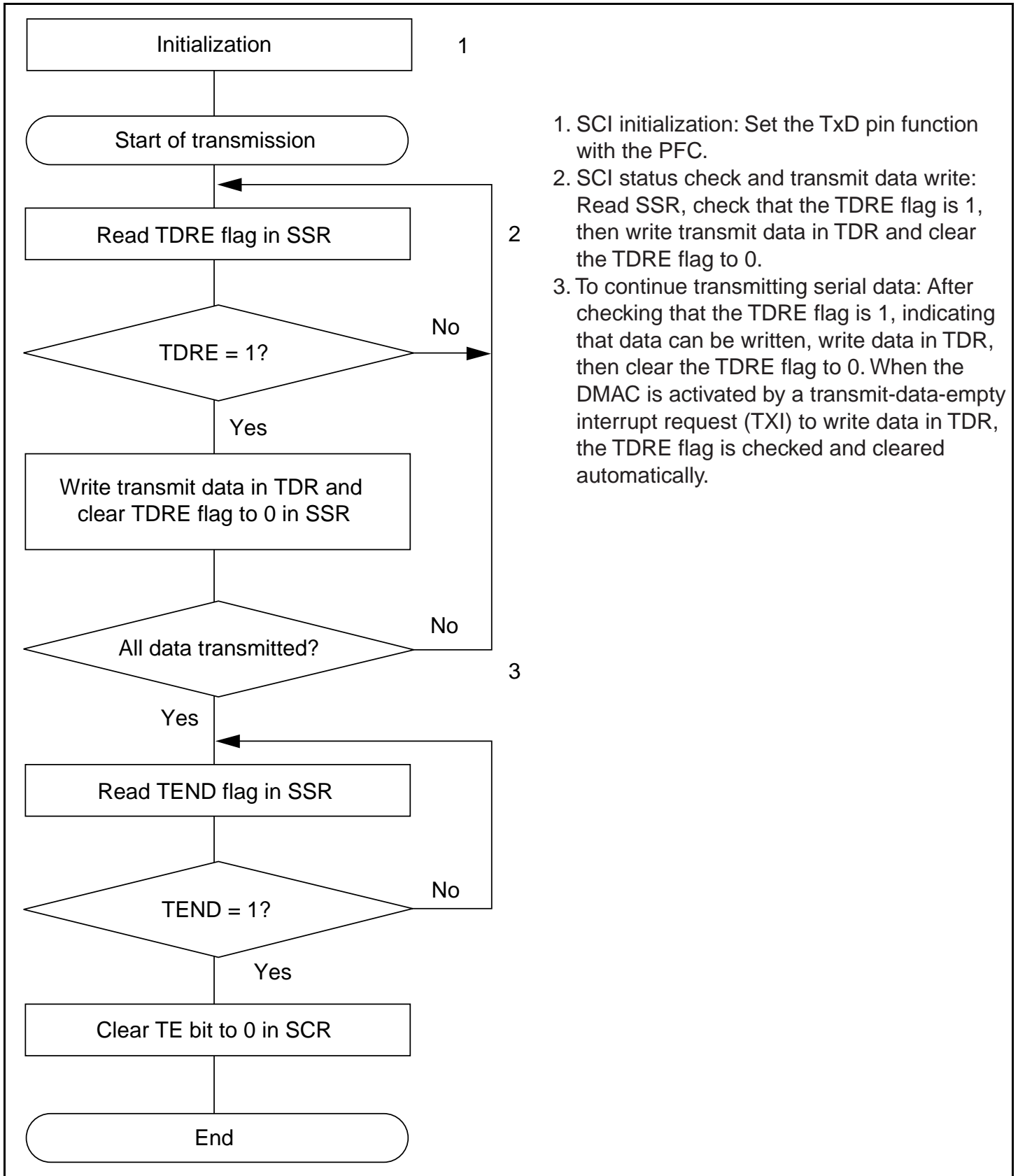
When changing the mode or communication format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

Figure 15.17 is a sample flowchart for initializing the SCI.



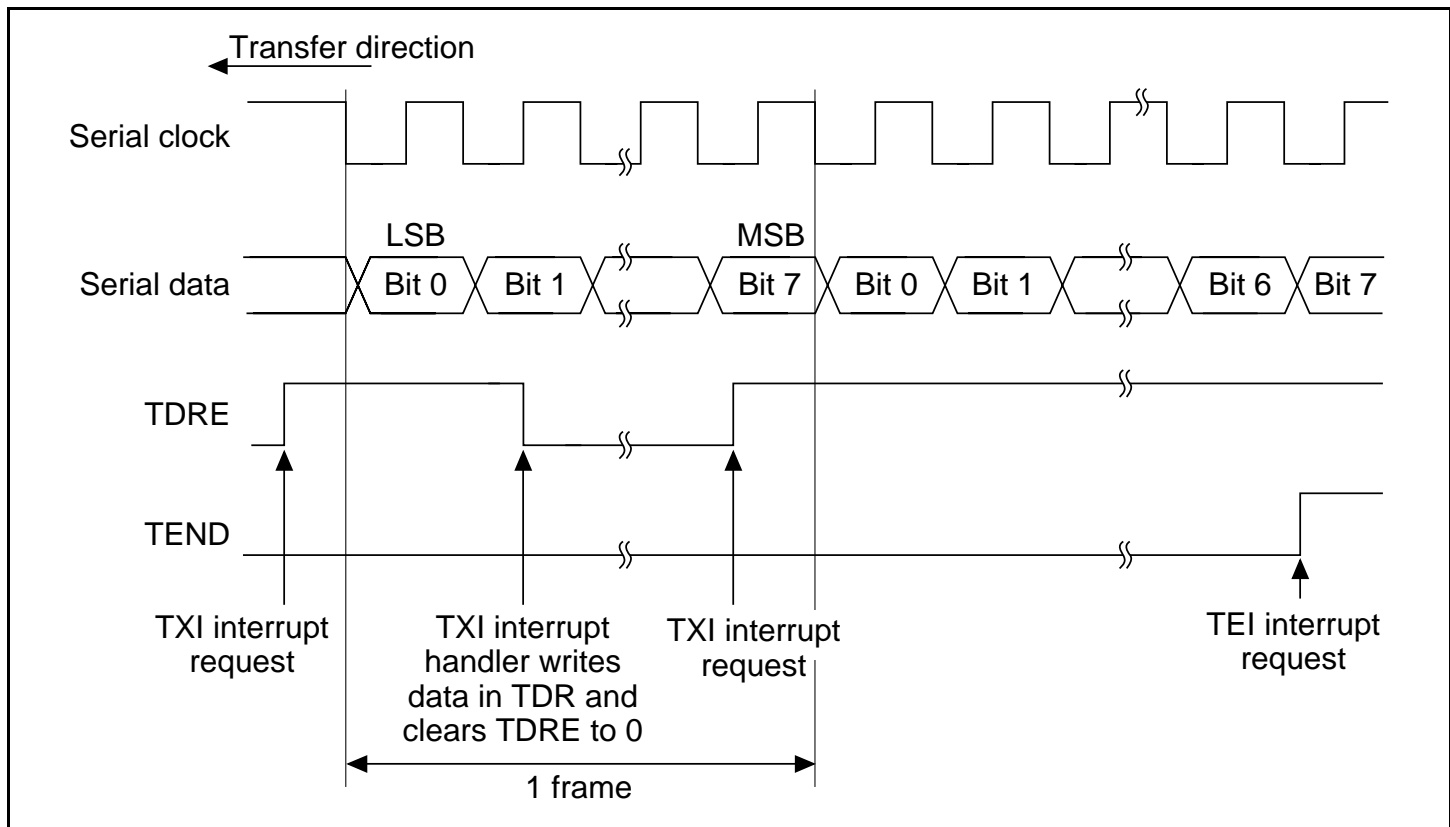
**Figure 15.17 Sample Flowchart for SCI Initialization**

**Transmitting Serial Data (Synchronous Mode):** Figure 15.18 shows a sample flowchart for transmitting serial data. The procedure is as follows (the steps correspond to the numbers in the flowchart):



**Figure 15.18 Sample Flowchart for Serial Transmitting**

Figure 15.19 shows an example of SCI transmit operation.



**Figure 15.19 Example of SCI Transmit Operation**

SCI serial transmission operates as follows.

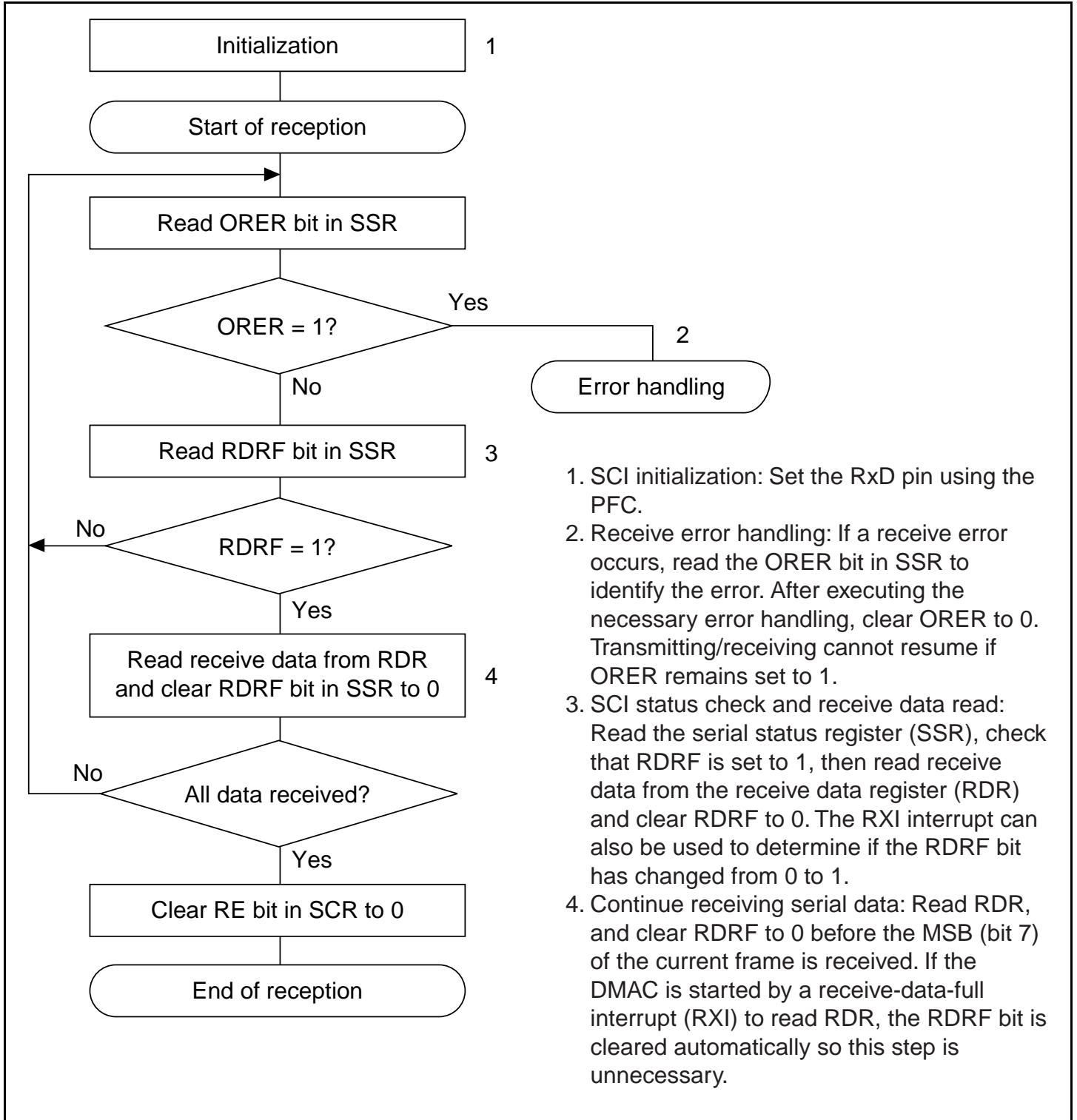
1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data and loads this data from TDR into the transmit shift register (TSR).
2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the transmit-data-empty interrupt enable bit (TIE) in SCR is set to 1, the SCI requests a transmit-data-empty interrupt (TXI) at this time.

If clock output mode is selected, the SCI outputs eight serial 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 bit when it outputs the MSB (bit 7). If TDRE is 0, the SCI loads data from TDR into TSR, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in SSR to 1, transmits the MSB, then holds the transmit data pin (TxD) in the MSB state. If the transmit-end interrupt enable bit (TEIE) in SCR is set to 1, a transmit-end interrupt (TEI) is requested at this time.
4. After the end of serial transmission, the SCK pin is held in the high state.

**Receiving Serial Data (Synchronous Mode):** Figures 15.20 and 15.21 show a sample flowchart for receiving serial data. When switching from asynchronous mode to synchronous mode, make sure that ORER, PER, and FER are cleared to 0. If PER or FER is set to 1, the RDRF bit will not be set and both transmitting and receiving will be disabled.

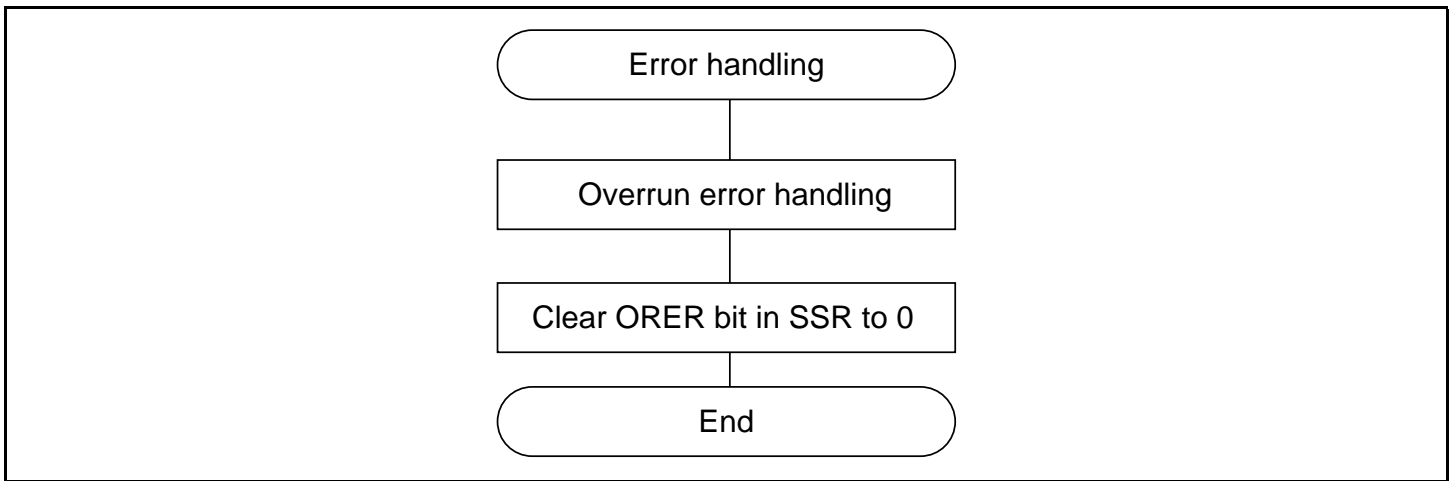
The procedure for receiving serial data is as follows (the steps correspond to the numbers in the flowchart):



1. SCI initialization: Set the RxD pin using the PFC.
2. Receive error handling: If a receive error occurs, read the ORER bit in SSR to identify the error. After executing the necessary error handling, clear ORER to 0. Transmitting/receiving cannot resume if ORER remains set to 1.
3. SCI status check and receive data read: Read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. The RXI interrupt can also be used to determine if the RDRF bit has changed from 0 to 1.
4. Continue receiving serial data: Read RDR, and clear RDRF to 0 before the MSB (bit 7) of the current frame is received. If the DMAC is started by a receive-data-full interrupt (RXI) to read RDR, the RDRF bit is cleared automatically so this step is unnecessary.

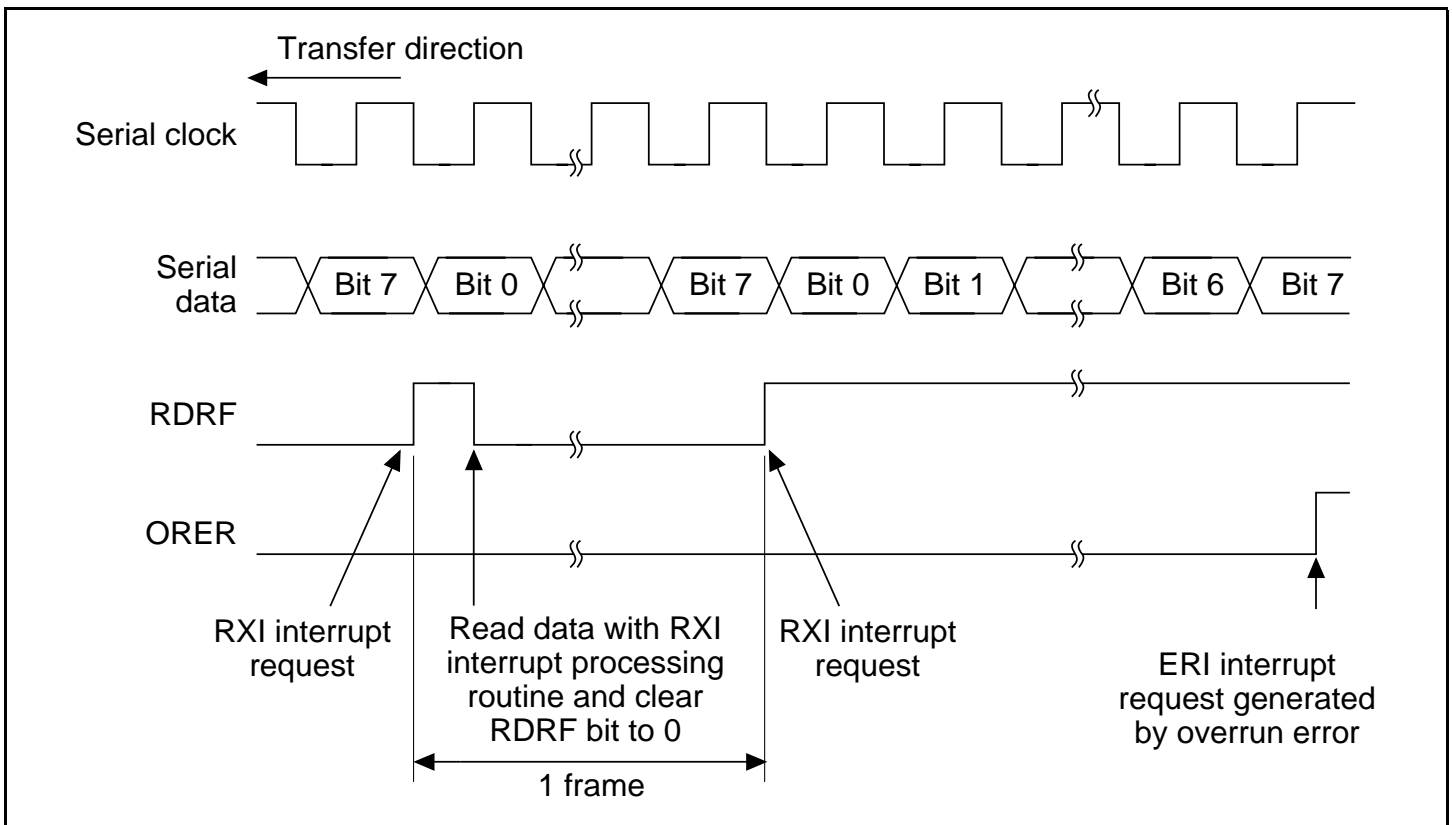
**Figure 15.20 Sample Flowchart for Serial Receiving (1)**





**Figure 15.21 Sample Flowchart for Serial Receiving (2)**

Figure 15.22 shows an example of the SCI receive operation.

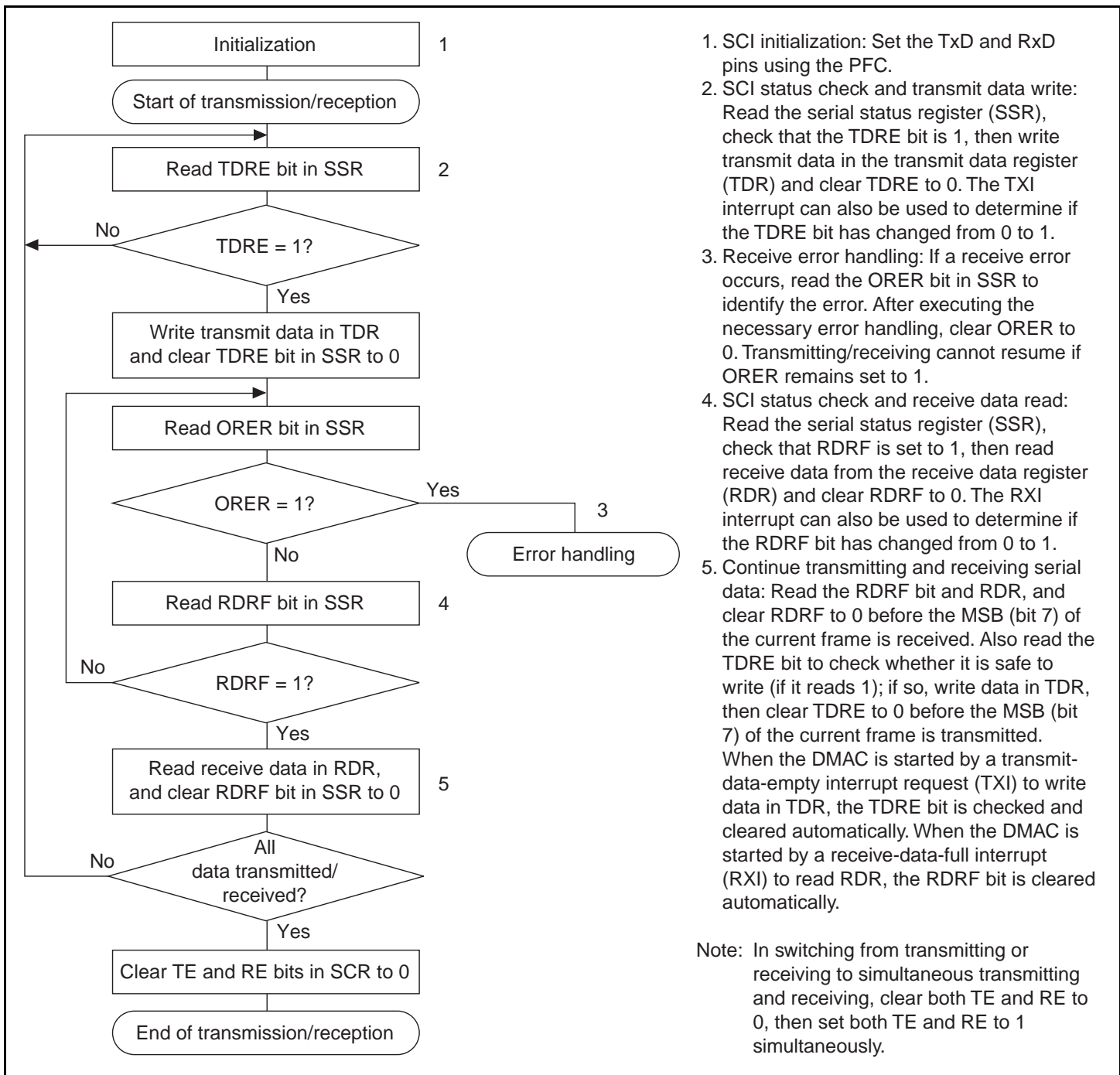


**Figure 15.22 Example of SCI Receive Operation**

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 RSR in order from the LSB to the MSB. After receiving the data, the SCI checks that RDRF is 0 so that receive data can be loaded from RSR into RDR. If this check passes, the SCI sets RDRF to 1 and stores the receive data in RDR. If the check does not pass (receive error), the SCI operates as indicated in table 15.11 and no further transmission or reception is possible. If the error flag is set to 1, the RDRF bit is not set to 1 during reception, even if the RDRF bit is 0 cleared. When restarting reception, be sure to clear the error flag.
3. After setting RDRF to 1, if the receive-data-full interrupt enable bit (RIE) is set to 1 in SCR, the SCI requests a receive-data-full interrupt (RXI). If the ORER bit is set to 1 and the receive-data-full interrupt enable bit (RIE) in SCR is also set to 1, the SCI requests a receive-error interrupt (ERI).

**Transmitting and Receiving Serial Data Simultaneously (Synchronous Mode):** Figure 15.23 shows a sample flowchart for transmitting and receiving serial data simultaneously. The procedure is as follows (the steps correspond to the numbers in the flowchart):



**Figure 15.23 Sample Flowchart for Serial Transmission and Reception**

## 15.4 SCI Interrupt Sources and the DMAC

The SCI has four interrupt sources: transmit-end (TEI), receive-error (ERI), receive-data-full (RXI), and transmit-data-empty (TXI). Table 15.12 lists the interrupt sources and indicates their priority. These interrupts can be enabled and disabled by the TIE, RIE, and TEIE bits in the serial control register (SCR). Each interrupt request is sent separately to the interrupt controller.

TXI is requested when the TDRE bit in SSR is set to 1. TXI can start the direct memory access controller (DMAC) to transfer data. TDRE is automatically cleared to 0 when the DMAC writes data in the transmit data register (TDR).

RXI is requested when the RDRF bit in SSR is set to 1. RXI can start the DMAC to transfer data. RDRF is automatically cleared to 0 when the DMAC reads the receive data register (RDR).

ERI is requested when the ORER, PER, or FER bit in SSR is set to 1. ERI cannot start the DMAC.

TEI is requested when the TEND bit in SSR is set to 1. TEI cannot start the DMAC. Where the TXI interrupt indicates that transmit data writing is enabled, the TEI interrupt indicates that the transmit operation is complete.

**Table 15.12 SCI Interrupt Sources**

Interrupt Source	Description	DMAC Activation	Priority
ERI	Receive error (ORER, PER, or FER)	No	High
RXI	Receive data full (RDRF)	Yes	↑
TXI	Transmit data empty (TDRE)	Yes	↓
TEI	Transmit end (TEND)	No	Low

## 15.5 Usage Notes

Sections 15.5.1 to 15.5.9 provide information concerning use of the SCI.

### 15.5.1 TDR Write and TDRE Flag

The TDRE bit in the serial status register (SSR) is a status flag indicating loading of transmit data from TDR into TSR. The SCI sets TDRE to 1 when it transfers data from TDR to TSR. Data can be written to TDR regardless of the TDRE bit status. If new data is written in TDR when TDRE is 0, however, the old data stored in TDR will be lost because the data has not yet been transferred to TSR. Before writing transmit data to TDR, be sure to check that TDRE is set to 1.

## 15.5.2 Simultaneous Multiple Receive Errors

Table 15.13 indicates the state of the SSR status flags when multiple receive errors occur simultaneously. When an overrun error occurs, the RSR contents cannot be transferred to RDR, so receive data is lost.

**Table 15.13 SSR Status Flags and Transfer of Receive Data**

Receive Error Status	SSR Status Flags				Receive Data Transfer
	RDRF	ORER	FER	PER	RSR → RDR
Overrun error	1	1	0	0	X
Framing error	0	0	1	0	O
Parity error	0	0	0	1	O
Overrun error + framing error	1	1	1	0	X
Overrun error + parity error	1	1	0	1	X
Framing error + parity error	0	0	1	1	O
Overrun error + framing error + parity error	1	1	1	1	X

Notes: O: Receive data is transferred from RSR to RDR.

X: Receive data is not transferred from RSR to RDR.

## 15.5.3 Break Detection and Processing (Asynchronous Mode Only)

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 FER is set and the parity error flag (PER) may also be set. In the break state, the SCI receiver continues to operate, so if the FER bit is cleared to 0, it will be set to 1 again.

## 15.5.4 Sending a Break Signal (Asynchronous Mode Only)

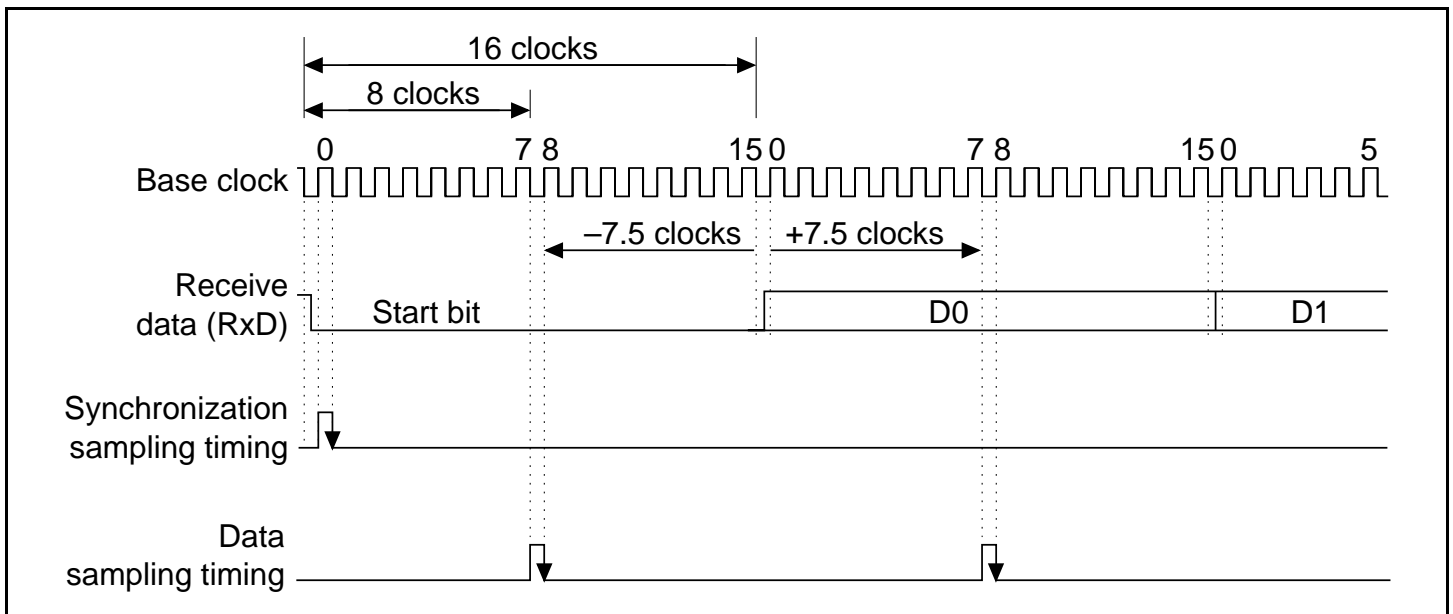
The TxD pin becomes a general I/O pin with the I/O direction and level determined by the I/O port data register (DR) and pin function controller (PFC) control register (CR). These conditions allow break signals to be sent. The DR value is substituted for the marking status until the PFC is set. Consequently, the output port is set to initially output a 1. To send a break in serial transmission, first clear the DR to 0, then establish the TxD pin as an output port using the PFC. When TE is cleared to 0, the transmission section is initialized regardless of the present transmission status.

### 15.5.5 Receive Error Flags and Transmitter Operation (Synchronous Mode Only)

When a receive error flag (ORER, PER, or FER) is set to 1, the SCI will not start transmitting even if TDRE is set to 1. Be sure to clear the receive error flags to 0 before starting to transmit. Note that clearing RE to 0 does not clear the receive error flags.

### 15.5.6 Receive Data Sampling Timing and Receive Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times the transfer rate. In receiving, the SCI synchronizes internally with the falling edge of the start bit, which it samples on the base clock. Receive data is latched on the rising edge of the eighth base clock pulse (figure 15.24).



**Figure 15.24 Receive Data Sampling Timing in Asynchronous Mode**

The receive margin in asynchronous mode can therefore be expressed as:

$$M = \left| \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \cdot 100\%$$

- M : Receive margin (%)
- N : Ratio of clock frequency to bit rate (N = 16)
- D : Clock duty cycle (D = 0 – 1.0)
- L : Frame length (L = 9 – 12)
- F : Absolute deviation of clock frequency

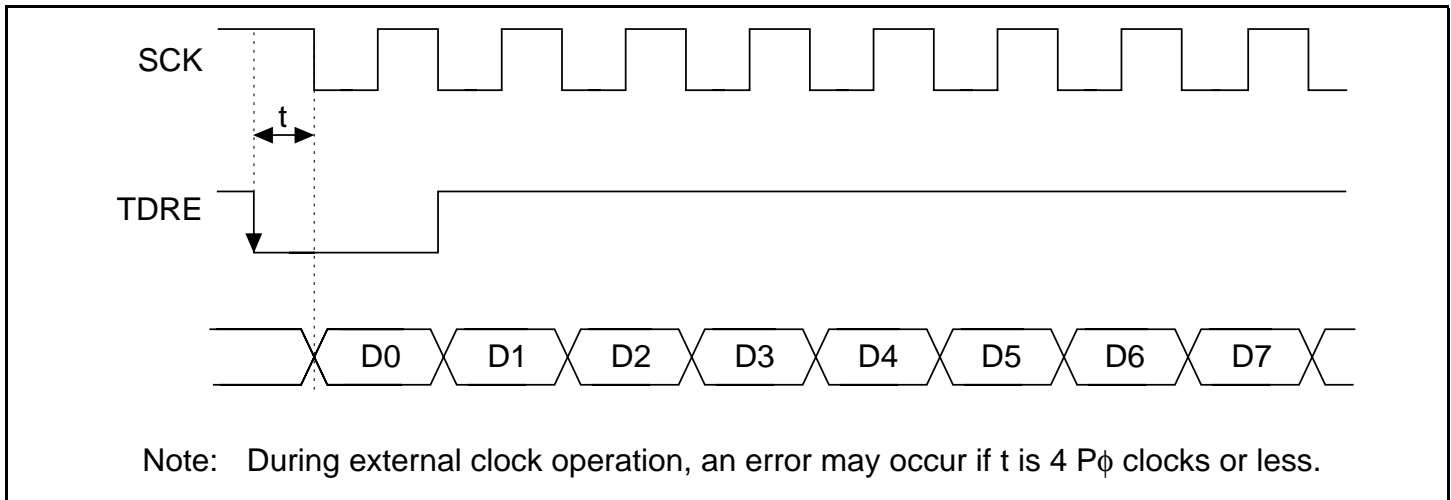
From the equation above, if F = 0 and D = 0.5 the receive margin is 46.875%:

$$\begin{aligned} D &= 0.5, F = 0 \\ M &= (0.5 - 1/(2 \cdot 16)) \cdot 100\% \\ &= 46.875\% \end{aligned}$$

This is a theoretical value. A reasonable margin to allow in system designs is 20–30%.

### 15.5.7 Constraints on DMAC Use

- When using an external clock source for the serial clock, update TDR with the DMAC, and then after the elapse of five peripheral clocks ( $P\phi$ ) or more, input a transmit clock. If a transmit clock is input in the first four  $P\phi$  clocks after TDR is written, an error may occur (figure 15.25).
- Before reading the receive data register (RDR) with the DMAC, select the receive-data-full (RXI) interrupt of the SCI as a start-up source.



**Figure 15.25 Example of Synchronous Transmission with DMAC**

### 15.5.8 Cautions on Synchronous External Clock Mode

- Set  $TE = RE = 1$  only when external clock SCK is 1.
- Do not set  $TE = RE = 1$  until at least four  $P\phi$  clocks after external clock SCK has changed from 0 to 1.
- When receiving, RDRF is 1 when RE is cleared to zero 2.5–3.5  $P\phi$  clocks after the rising edge of the RxD D7 bit SCK input, but copying to RDR is not possible.

### 15.5.9 Caution on Synchronous Internal Clock Mode

When receiving, RDRF is 1 when RE is cleared to zero 1.5  $P\phi$  clocks after the rising edge of the RxD D7 bit SCK output, but copying to RDR is not possible.

# Section 16 Controller Area Network-II (HCAN-II)

## 16.1 Overview

The controller area network-II (HCAN-II) is a module that controls the controller area network (CAN) for realtime communication in the car and industrial device systems, etc. It serves to facilitate the hardware/software interface so that engineers involved in the CAN implementation can ensure the design is successful.

The CAN data link controller function is not described in this document. The following CAN-specification documents should be referred to. The interfaces from the CAN controller are described, in so far as they pertain to the connection with the user interface.

References:

1. CAN License Specification, Robert Bosch GmbH, 1992
2. CAN Specification Version 2.0, Robert Bosch GmbH, 1991
3. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany
4. OSEK Communication Specification, Version 2.1 revision 1, OSEK /VDX, 17<sup>th</sup> June 1998

### 16.1.1 Features

- Supports CAN specification 2.0A/2.0B and ISO-11898-1
- 31 programmable mailboxes for transmission/reception and one receive-only mailbox (there is a limitation for usage only in mailbox 31)
- Sleep mode for low power consumption and automatic recovery from sleep mode by detecting CAN bus activity
- Programmable receive filter mask (standard and extended IDs) supported by all mailboxes
- Programmable CAN data rate up to 500 kbits/s (or 1 Mbit/s with a limitation)
- Transmit message queuing with an on-chip priority sorting mechanism against the problem of priority inversion for realtime applications
- Flexible interrupt structure
- Read section 16.8, Usage Notes carefully.



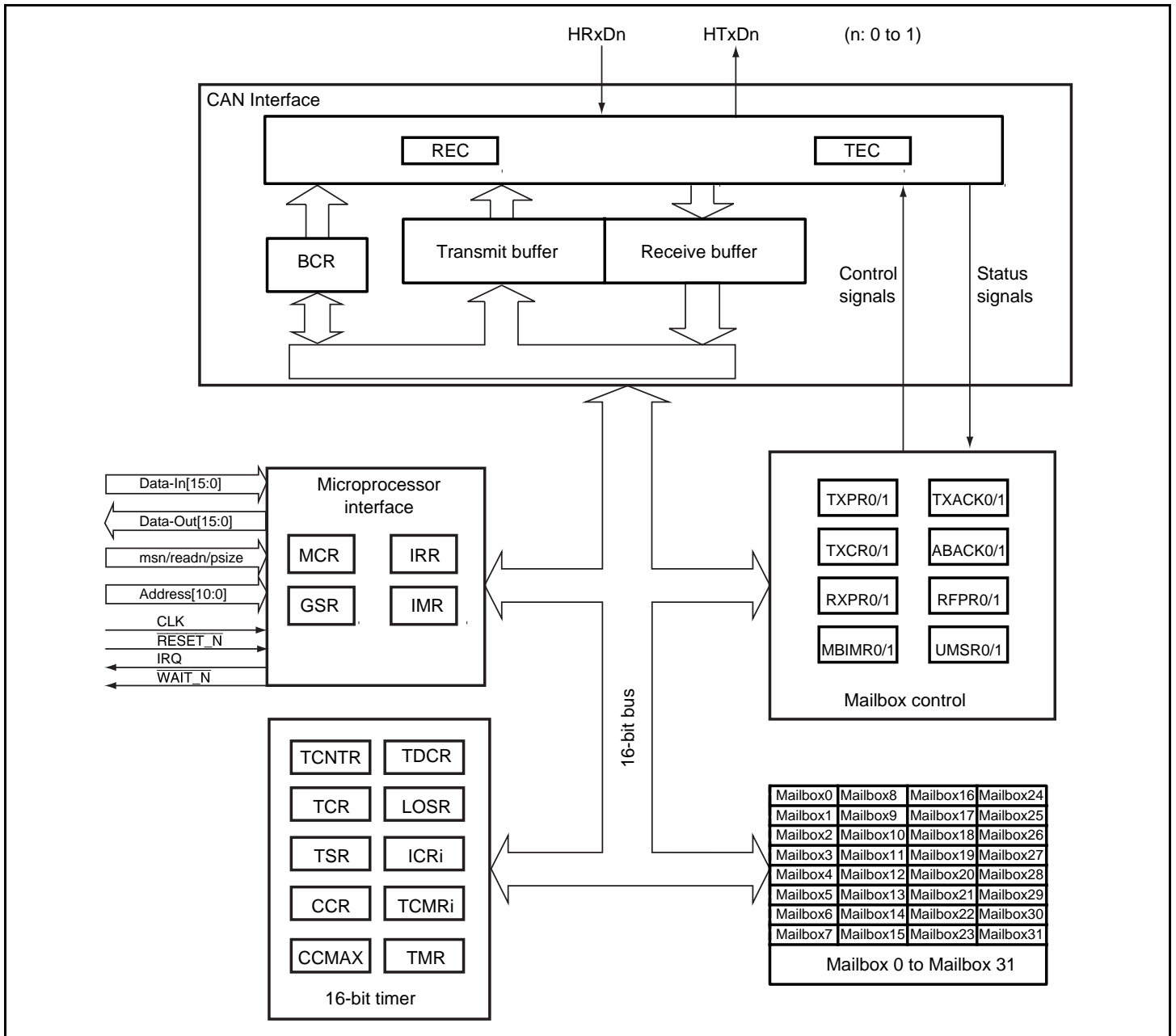
The following features have been added in the HCAN-II.

- IRR0 function to notify a software reset and halt
- Halt mode status bit and error passive status bit added to GSR
- Supports various test modes
- Data frame and remote frame are separated (IRR2 is independent from IRR1 and RXPR from RFRR)
- When transmitting, the highest priority search is scanned from mailbox 31 down to mailbox 1
- When receiving, the matching ID search is scanned from mailbox 31 down to mailbox 0, and one received message is only stored into one mailbox
- More flexible BCR
- Bus off/bus off recover interrupt (IRR6)
  
- Others:
- HCAN-II connection method: Two connections are available
  - 32-buffer HCAN-II  $\times$  2 channels (transmit pin  $\times$  2 and receive pin  $\times$  2)
  - 64-buffer HCAN-II (wire AND)  $\times$  1 channel (transmit pin  $\times$  1 and receive pin  $\times$  1)
- DMAC can be activated by a receive message of a mailbox (only mailbox 0 in HCAN0)

## 16.2 Architecture

### 16.2.1 Block Diagram

The HCAN-II device offers a flexible and sophisticated way to configure and control CAN frames, supporting CAN2.0B Active and ISO-11898. The module is configured of 5 different functional blocks. These are the Microprocessor Interface (MPI), mailbox, mailbox control, timer, and CAN interface. Figure 16.1 shows a block diagram of the HCAN-II module. The bus interface timing is designed based on the SuperH peripheral bus interface (P-Bus).



**Figure 16.1 Block Diagram of HCAN-II (for One Channel)**

Note: Since the HCAN-II is designed based on a 16-bit bus system, longword (32-bit) access is prohibited. Thus, word access must be used for all the registers, and word or byte access must be used for the mailboxes.

## 16.2.2 Each Block Function

### (1) Microprocessor Interface (MPI)

The MPI allows communication between the host CPU and the HCAN's registers/mailboxes to control the memory interface, and the data controller, etc. It also contains the wakeup control logic that detects the CAN bus state and notifies the MPI and the other parts of the HCAN so that the HCAN can automatically exit sleep mode.

Contains registers such as MCR, IRR, GSR, and IMR.

### (2) Mailboxes

The mailboxes are message buffers which are configured of RAM. There are 32 mailboxes, and each mailbox stores the following information.

- CAN message control (StdID, RTR, DLC, IDE, etc.)
- CAN message data (for CAN data frames)
- Local acceptance filter mask (LAFM) during reception
- 3-bit mailbox configuration, automatic transmit bit for remote request, new message control bit

### (3) Mailbox Control

The mailbox control handles the following functions.

For receive messages, compares the IDs, generates appropriate RAM addresses to store messages from the CAN interface into the mailbox, and sets/clears corresponding registers.

To transmit messages, runs the internal arbitration to select the correct priority message which is event-triggered, loads the message from the mailbox into the Tx-buffer of the CAN interface, and sets/clears corresponding registers accordingly.

Arbitrates mailbox accesses between the host CPU and the CAN interface or mailbox control.

Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, and MBIMR.

### (4) Timer

The timer is a block which transmits and receives messages at a specific time frame and records the result. The timer is a 16-bit free-running up counter which is controlled by the host CPU. It provides three 16-bit compare match registers. They can generate interrupt signals, set or clear the counter value in the local offset value, and clear messages in the transmission queue. Two 16-bit input capture registers are included to record timestamps on CAN messages and synchronize the timer value globally within a CAN system.

The clock period of this timer offers a wide selection generated from the peripheral clock.

Contains registers such as TCNTR, TCR, TPSR, TDCR, LOSR, ICR0\_tm, ICR0\_cc, ICR0\_buf, ICR1, TCMR0, TCMR1, TCMR2, TMR, CCR, CCR\_buf, and CMAX.

**Important:** The timer function is not supported by the SH7058.

## (5) CAN Interface

The CAN interface supports the requirements for a CAN bus data link controller which is specified in Reference 2 (section 16.1). It fulfils all the functions of a data link layer (DLC layer) as specified by the 7 layers of the OSI model. This block provides the receive error counter, transmit error counter, and bit timing set registers, and various test modes corresponding to the CAN bus specification. This block also stores transmit/receive data for the CAN data link controller.

### 16.2.3 Pin Configuration

Table 16.1 lists the pin configuration and functions.

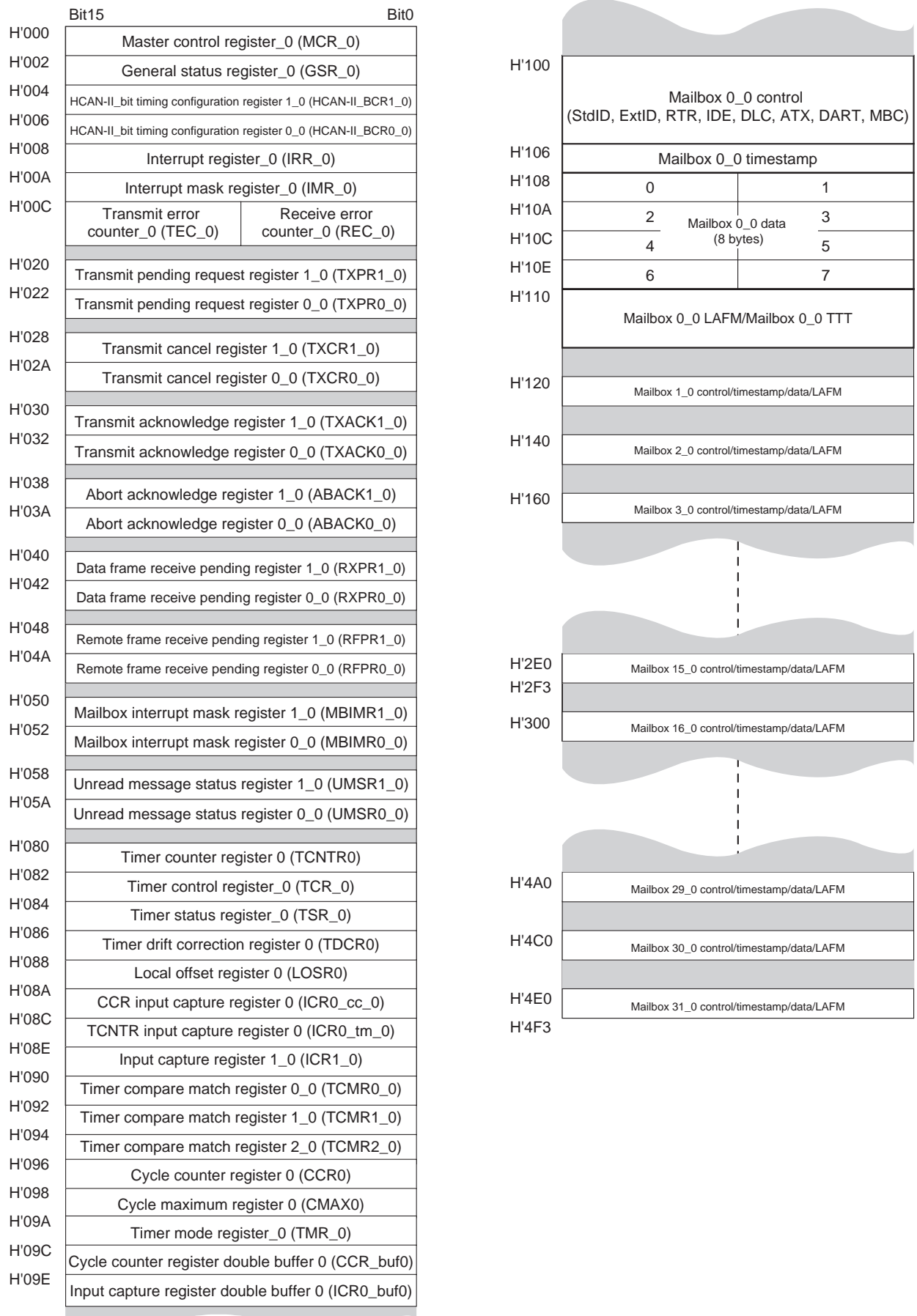
**Table 16.1 Pin Configuration**

<b>Name</b>	<b>Input/Output</b>	<b>Function</b>
HRxD0	Input	CAN bus receive signal of channel 0
HTxD0	Output	CAN bus transmit signal of channel 0
HRxD1	Input	CAN bus receive signal of channel 1
HTxD1	Output	CAN bus transmit signal of channel 1

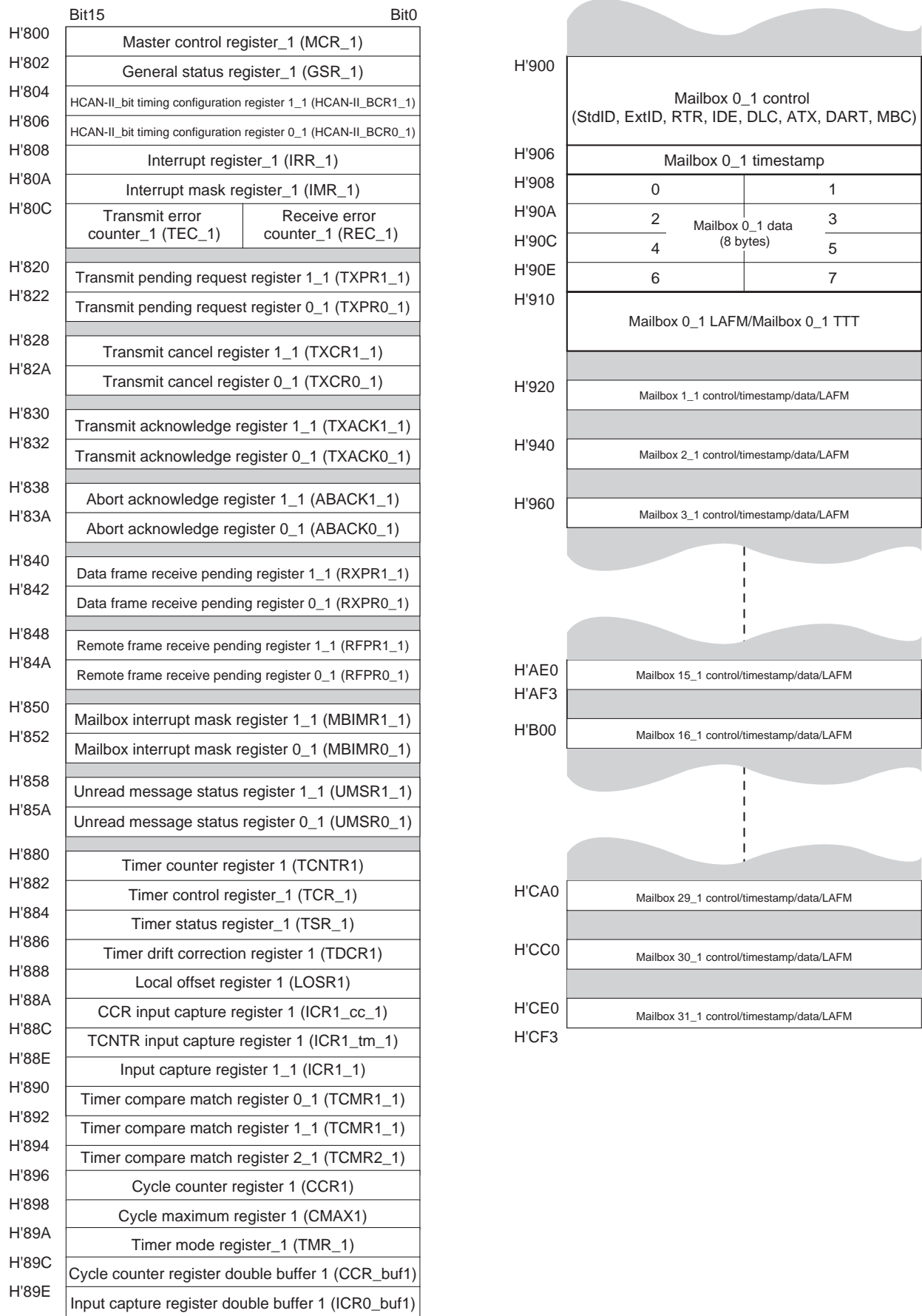
### 16.2.4 Memory Map

Figures 16.2 (1) and 16.2 (2) show the memory maps of registers which can be accessed by software.

Base address: Channel 0 → H'FFFFFFD000, channel 1 → H'FFFFFFD800



**Figure 16.2 (1) HCAN-II Memory Map for Channel 0 (HCAN0)**



**Figure 16.2 (2) HCAN-II Memory Map for Channel 1 (HCAN1)**

## 16.3 Mailboxes

### 16.3.1 Mailbox Configuration

Mailboxes play a role as message buffers to transmit/receive CAN frames. Each mailbox is comprised of 4 identical storage fields that are 1): Message control, 2): Message data, 3): Timestamp, and 4): Local acceptance filter mask (LAFM)/Transmission trigger time. Table 16.2 shows the memory map for each mailbox.

Note: The message control (STDID/EXTID/RTR/ZDE), timestamp, and LAFM/transmission trigger time fields can only be accessed in word size (16 bits), whereas the message control (NMC/ATX/MBC/DLC) and the message data area can be accessed in word (16-bit) or byte (8-bit) size. Also, when the setting of the MBC bits makes the mailbox inactive, all settings other than the MBC bits must be initialized to 0 because an unused mailbox affects the RAM configuration. When the LAFM is not used to receive messages, it must be cleared to 0.

**Table 16.2 Mailbox Configuration**

Mailbox	Address			
	Control	Timestamp	Data	LAFM/Trigger Time
	6 Bytes	2 Bytes	8 Bytes	4 Bytes
0 (Receive only)	100 – 105	106 – 107	108 – 10F	110 – 113
1	120 – 125	126 – 127	128 – 12F	130 – 133
2	140 – 145	146 – 147	148 – 14F	150 – 153
3	160 – 165	166 – 167	168 – 16F	170 – 173
4	180 – 185	186 – 187	188 – 18F	190 – 193
5	1A0 – 1A5	1A6 – 1A7	1A8 – 1AF	1B0 – 1B3
6	1C0 – 1C5	1C6 – 1C7	1C8 – 1CF	1D0 – 1D3
7	1E0 – 1E5	1E6 – 1E7	1E8 – 1EF	1F0 – 1F3
8	200 – 205	206 – 207	208 – 20F	210 – 213
9	220 – 225	226 – 227	228 – 22F	230 – 233
10	240 – 245	246 – 247	248 – 24F	250 – 253
11	260 – 265	266 – 267	268 – 26F	270 – 273
12	280 – 285	286 – 287	288 – 28F	290 – 293
13	2A0 – 2A5	2A6 – 2A7	2A8 – 2AF	2B0 – 2B3
14	2C0 – 2C5	2C6 – 2C7	2C8 – 2CF	2D0 – 2D3
15	2E0 – 2E5	2E6 – 2E7	2E8 – 2EF	2F0 – 2F3
16	300 – 305	306 – 307	308 – 30F	310 – 313
17	320 – 325	326 – 327	328 – 32F	330 – 333
18	340 – 345	346 – 347	348 – 34F	350 – 353
19	360 – 365	366 – 367	368 – 36F	370 – 373
20	380 – 385	386 – 387	388 – 38F	390 – 393
21	3A0 – 3A5	3A6 – 3A7	3A8 – 3AF	3B0 – 3B3
22	3C0 – 3C5	3C6 – 3C7	3C8 – 3CF	3D0 – 3D3
23	3E0 – 3E5	3E6 – 3E7	3E8 – 3EF	3F0 – 3F3
24	400 – 405	406 – 407	408 – 40F	410 – 413
25	420 – 425	426 – 427	428 – 42F	430 – 433
26	440 – 445	446 – 447	448 – 44F	450 – 453
27	460 – 465	466 – 467	468 – 46F	470 – 473
28	480 – 485	486 – 487	488 – 48F	490 – 493
29	4A0 – 4A5	4A6 – 4A7	4A8 – 4AF	4B0 – 4B3
30	4C0 – 4C5	4C6 – 4C7	4C8 – 4CF	4D0 – 4D3
31	4E0 – 4E5	4E6 – 4E7	4E8 – 4EF	4F0 – 4F3



Mailbox 0 is a receive-only mailbox, and all the rest of mailbox 1 to mailbox 31 can operate as both receive and transmit mailboxes according to the MBC (Mailbox Configuration) bits in the message control. Figure 16.3 shows the configuration of a mailbox in detail.

**Important:** If mailbox 31 is used as a transmit buffer, there is a usage limitation. For details, see section 16.8, Usage Notes.

Register Name	Address		Data Bus													Access Size	Field Name		
	HCAN0	HCAN1	15	14	13	12	11	10	9	8	7	6	5	4	3			2	1
MBx[0] to [1]	H'100+N×32	H'900+N×32	0	STDID[10:0]										RTR	IDE	EXIDT[17:16]		16 bits	Control
MBx[2] to [3]	H'102+N×32	H'902+N×32	EXTID[15:0]													16 bits			
MBx[4] to [5]	H'104+N×32	H'904+N×32	CCM	TTE	NMC	ATX	DART	MBC[2:0]		0	TCT	CBE	CLE	DLC[3:0]			8/16 bits	Timestamp	
MBx[6]	H'106+N×32	H'906+N×32	Timestamp[15:0]													16 bits			
MBx[7] to [8]	H'108+N×32	H'908+N×32	MSG_DATA_0 (first Rx/Tx byte)						MSG_DATA_1						8/16 bits	Data			
MBx[9] to [10]	H'10A+N×32	H'90A+N×32	MSG_DATA_2						MSG_DATA_3						8/16 bits				
MBx[11] to [12]	H'10C+N×32	H'90C+N×32	MSG_DATA_4						MSG_DATA_5						8/16 bits				
MBx[13] to [14]	H'10E+N×32	H'90E+N×32	MSG_DATA_6						MSG_DATA_7						8/16 bits				
MBx[15] to [16]	H'110+N×32	H'910+N×32	Local acceptance filter mask 0 (LAFM0)/Tx trigger time 0 (TTT0)														16 bits	LAFM/Tx trigger control	
MBx[17] to [18]	H'112+N×32	H'912+N×32	Local acceptance filter mask 1 (LAFM1)/Tx trigger time 1 (TTT1)													16 bits			

Notes: 1. All bits shadowed in gray are reserved and the write value should be 0. The value read as the initial value is not guaranteed.  
2. ATX, DART, and CLE are not supported by mailbox 0 and the MBC setting of mailbox 0 is limited.  
3. If the CAN bus is configured in little endian (MCR4 = 1), transmission is started from MSG\_DATA\_1 instead of MSG\_DATA\_0 (i.e. the sequence becomes: MSG\_DATA\_1, MSG\_DATA\_0, MSG\_DATA\_3, MSG\_DATA\_2, MSG\_DATA\_5, MSG\_DATA\_4, MSG\_DATA\_7, and MSG\_DATA\_6).  
4. x/N: 0 to 31 (indicates the mailbox number)

**Figure 16.3 Mailbox-N Configuration**

### 16.3.2 Message Control Field

Register Name	Address	Bit	Bit Name	Description
MBx[0], MBx[1]*	H'100 + N×32	15	—	Reserved  The write value should be 0. The read value is not guaranteed.
		14 to 4	STDID [10:0]	Standard ID  Set the ID (standard ID) of data frames and remote frames.
		3	RTR	Remote Transmission Request  Distinguishes between data frames and remote frames. This bit is overwritten by receive CAN frames depending on data frames or remote frames.  <b>Important:</b> Note that, when the ATX bit is set with the setting MBC = 001 the RTR bit cannot be set. When a remote frame is received, the host CPU can be notified by the corresponding RFPR or IRR2 (remote frame request interrupt), however, as the HCAN needs to transmit the current message as a data frame, the RTR bit remains 0.  0: Data frame 1: Remote frame
		2	IDE	ID Extension  Distinguishes between the standard format and extended format of CAN data frames and remote frames.  0: Standard format 1: Extended format
MBx[2], MBx[3]*	H'102 + N×32	1, 0	EXTID [17:16]	Extended ID
		15 to 0	EXTID [15:0]	Set the ID (extended ID) of data frames and remote frames.

Note: \* x/N: 0 to 31 (Indicates the mailbox number)

Register Name	Address	Bit	Bit Name	Description
MBx[4], MBx[5]*	H'104 + N×32	15	CCM	<p>CAN-ID Compare Match</p> <p>When this bit is set, message reception in the corresponding mailbox can generate two triggers.</p> <p>If TCR9 is set to 1, TCR14 is cleared to freeze ICR0. If TCR10 is set to 1, TCNTR (timer counter register) is automatically cleared and the LOSR (local offset register) value is set.</p> <p><b>Important:</b> This function is not supported by the SH7058. Thus the write value should be 0.</p>
		14	TTE	<p>Time Trigger Enable</p> <p>When this bit is set, a mailbox in which TXPR has been already set transmits a message at a time set in the Tx trigger time field.</p> <p><b>Important:</b> If this bit is set, a failure occurs during message transmission. Therefore setting is prohibited. The write value should be 0. The value read as the initial value is not guaranteed.</p>

Register Name	Address	Bit	Bit Name	Description
MBx[4], MBx[5]*	H'104 + N×32	13	NMC	<p>New Message Control</p> <p>When this bit is cleared, a mailbox in which PXPR/PFPR has been already set does not store the new message but retains the previous one and sets the UMSR corresponding bit.</p> <p>When this bit is set, a mailbox in which PXPR/PFPR has been already set stores the new message and sets the UMSR corresponding bit.</p> <p>If a message is received in a mailbox in overwrite mode (NMC = 1), the host CPU must perform an additional check at the end of the data reading from the mailbox in order to guarantee that the mailbox data have not been corrupted during such operation by another receive message. The additional check, to be performed at the end of the mailbox access, consists in verifying that the associated bit of UMSR has not been set and so no overwrite has occurred; in case such bit is set data have been corrupted and so the message must be discarded.</p>

---

Note: \* x/N: 0 to 31 (Indicates the mailbox number)

Register Name	Address	Bit	Bit Name	Description
MBx[4], MBx[5]*	H'104 + N×32	12	ATX	<p>Automatic Transmission of Data Frame</p> <p>When this bit is set to 1 and a remote frame is received in the mailbox, a data frame is automatically transmitted from the same mailbox using the current contents of the message data. The scheduling of transmission is controlled by the CAN ID. In order to use this function, the MBC[2:0] bits should be set to 001. When transmission is performed by this function, the DLC (data length code) to be used is the one that has been received.</p> <p><b>Important:</b> Note that, when this function is used, the RTR bit is not set even if a remote frame is received. When a remote frame is received, the host CPU will be notified by RFPR or IRR2 (remote frame request interrupt), however, as the HCAN needs to transmit the current message as a data frame, the RTR bit remains 0.</p>
		11	DART	<p>Disable Automatic Retransmission</p> <p>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. When this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is cleared, the HCAN tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR.</p> <p><b>Important:</b> This function is not supported by the SH7058. Thus the write value should be 0. The value read as the initial value is not guaranteed.</p>

Note: \* x/N: 0 to 31 (Indicates the mailbox number)

Register Name	Address	Bit	Bit Name	Description
MBx[4], MBx[5]*	H'104 + N×32	10 to 8	MBC[2:0]	<p>Mailbox Configuration</p> <p>Mailbox functions are set as shown in table 16.3.</p> <p>When MBC = 111, the mailbox is inactive, i.e., it does not receive or transmit a message regardless of TXPR or other settings. When MBC = 000 and the TTE bit is set, the Tx-trigger time field becomes available. The MBC = 110 or 011 setting is prohibited. When MBC is set to any other value, the LAFM field becomes available.</p> <p><b>Important:</b> MB0 should be used as receive-only (MBC = 010).</p>
		7	—	<p>Reserved</p> <p>The write value should be 0. The read value is not guaranteed.</p>
		6	TCT	<p>Timer Counter Transfer</p> <p>When this bit is set, a mailbox is set for transmission, and the DLC is set to 4, the TCNTR value, at the SOF, is embedded in the second and third bytes of the message data, instead of MSG_DATA_2 and MSG_DATA_3, and the CYCLE_COUNT in the first byte instead of MSG_DATA_0[3:0] when this mailbox starts transmission.</p> <p>This function will be useful when the HCAN performs a time master role to transmit the time reference message.</p> <p>For example, considering that two HCAN controllers are connected in the same network and that the receiver stores the message in mailbox N, the data format is shown as figure 16.4 depending on the endian setting for the CAN bus (MCR4).</p> <p><b>Important:</b> This function is not supported by the SH7058.</p> <p>Thus the write value should be 0. The value read as the initial value is not guaranteed.</p>

Note: \* x/N: 0 to 31 (Indicates the mailbox number)

Register Name	Address	Bit	Bit Name	Description
MBx[4], MBx[5]*	H'104 + N×32	5	CBE	<p>CAN Bus Error</p> <p>An external fault-tolerant CAN transceiver can be used together with the HCAN module. If the error output pin of the transceiver (normally active low) is connected to the <u>CAN_NERR</u> pin of this LSI, the value of the <u>CAN_NERR</u> pin is stored into this bit at the end of each transmission/reception (if the message is stored). The inverted value of the <u>CAN_NERR</u> pin is set to this bit. If the error output pin is active high, the setting value is not inverted. When this bit is set, it indicates a potential physical error with the CAN bus. As the <u>CAN_NERR</u> value is updated after the transmission or reception in the corresponding mailbox, non-interrupt is dedicated to this function but instead the normal transmit end interrupt (IRR6) and normal receive end interrupt (IRR2) should be considered.</p> <p><b>Important:</b> This function is not supported by the SH7058. Thus the write value should be 0. The value read as the initial value is not guaranteed.</p>
		4	CLE	<p>Transmit Clear Enable</p> <p>When this bit is set, message reception in the corresponding mailbox cancels the wait messages in the transmission queue. This action is notified by IRR8 and ABACK.</p> <p><b>Important:</b> This function is not supported by the SH7058. Thus the write value should be 0. The value read as the initial value is not guaranteed.</p>

Note: \* x/N: 0 to 31 (Indicates the mailbox number)

Register Name	Address	Bit	Bit Name	Description
MBx[4], MBx[5]*	H'104 + N×32	3 to 0	DLC[3:0]	Data Length Code Indicate the number of data bytes to be transmitted in a data frame. DLC[3:0] Data Length 0000 0 bytes 0001 1 byte 0010 2 bytes 0011 3 bytes 0100 4 bytes 0101 5 bytes 0110 6 bytes 0111 7 bytes 1xxx 8 bytes [Legend] x: Don't care

Note: \* x/N: 0 to 31 (Indicates the mailbox number)

**Table 16.3 Mailbox Configuration (Setting of MBC[2:0] Bits)**

MBC[2]	MBC[1]	MBC[0]	Data Frame Transmission	Remote Frame Transmission	Data Frame Reception	Remote Frame Reception	Description	
0	0	0	Yes	Yes	No	No	<ul style="list-style-type: none"> <li>Not allowed for mailbox 0</li> </ul>	
0	0	1	Yes	Yes	No	Yes	<ul style="list-style-type: none"> <li>Can be used with ATX</li> <li>Not allowed for mailbox 0</li> <li>LAFM can be used</li> </ul>	
0	1	0	No	No	Yes	Yes	<ul style="list-style-type: none"> <li>Allowed for mailbox 0</li> <li>LAFM can be used</li> </ul>	
0	1	1	Setting prohibited					
1	0	0	No	Yes	Yes	Yes	<ul style="list-style-type: none"> <li>Not allowed for mailbox 0</li> <li>LAFM can be used</li> </ul>	
1	0	1	No	Yes	Yes	No	<ul style="list-style-type: none"> <li>Not allowed for mailbox 0</li> <li>LAFM can be used</li> </ul>	
1	1	0	Setting prohibited					
1	1	1	Mailbox inactive					



**Important:** If mailbox 31 is used as a transmit buffer, there is a usage limitation. For details, see section 16.8, Usage Notes.

**Message Data Field when TCT = 1:**

Register Name	Address		Data Bus														Access Size	Field Name
	HCAN0	HCAN1	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
MBx[7] to [8]	H'108+N×32	H'908+N×32	Cycle_Counter (first Rx/Tx byte)							MSG_DATA_1							8/16 bits	Data
MBx[9] to [10]	H'10A+N×32	H'90A+N×32	TCNTR[7:0]							TCNTR[15:8]							8/16 bits	
MBx[11] to [12]	H'10C+N×32	H'90C+N×32	MSG_DATA_4							MSG_DATA_5							8/16 bits	
MBx[13] to [14]	H'10E+N×32	H'90E+N×32	MSG_DATA_6							MSG_DATA_7							8/16 bits	
Big endian																		
MBx[7] to [8]	H'108+N×32	H'908+N×32	MSG_DATA_1							Cycle_Counter (first Rx/Tx byte)							8/16 bits	Data
MBx[9] to [10]	H'10A+N×32	H'90A+N×32	TCNTR[15:8]							TCNTR[7:0]							8/16 bits	
MBx[11] to [12]	H'10C+N×32	H'90C+N×32	MSG_DATA_5							MSG_DATA_4							8/16 bits	
MBx[13] to [14]	H'10E+N×32	H'90E+N×32	MSG_DATA_7							MSG_DATA_6							8/16 bits	
Little endian																		

[Legend] x/N: 0 to 31 (Indicates the mailbox number)

**Figure 16.4 Message Data Field**

## Timestamp Fields:

Records the timestamp on messages for transmission/reception. The timestamp will be a useful function to monitor if messages are received/transmitted within expected schedule or if messages for transmission are scheduled in the appropriate order.

Register Name	Address	Bit	Bit Name	Description
MBx[6]*	H'106 + N × 32	15 to 0	TimeStamp [15:0]	<p>Message Reception:</p> <p>During message reception, when the SOF or EOF is detected, ICR1 (input capture register 1) always captures the TCNTR (timer counter register) value or the value of Cycle_Counter + TCNTR[15:4], depending on the value of bit 3 in TMR (Timer mode register), at either SOF or EOF depending on the value in TCR13 (timer control register), and the ICR1 value is stored into the timestamp field of the corresponding mailbox.</p> <p><b>Important:</b> Capturing at the SOF is not supported by the SH7058. Thus TCR13 should be set to EOF detection mode.</p> <p>Message Transmission:</p> <p>During message transmission, the TCNTR (timer counter register) value or the value of Cycle_Counter + TCNTR[15:4], depending on the value of bit 3 in TMR (timer mode register) is captured when either the TXPR bit or TXACK bit is set depending on the value in TCR12, and the captured value is stored into the timestamp field of the corresponding mailbox.</p> <p><b>Important:</b> Capturing when the TXPR bit is set is not supported by the SH7058.</p> <p>Activation of the TCNR (timer) causes a problem in the SH7058 (timer usage is prohibited). Therefore, the timestamp function is not supported.</p> <p>The write value should be 0. The value read as the initial value is not guaranteed.</p>

Note: \* x/N: 0 to 31 (Indicates the mailbox number)

### 16.3.3 Message Data Fields

Register Name	Address	Bit	Bit Name	Description
MBx[7], MBx[8]*	H'108 + N× 32	15 to 8, 7 to 0	MSG_DATA_0, MSG_DATA_1	Store the CAN message data that is transmitted or received.
MBx[9], MBx[10]*	H'10A + N × 32	15 to 8, 7 to 0	MSG_DATA_2, MSG_DATA_3	MSG_DATA_0 corresponds to the first data byte that is transmitted or received.
MBx[11], MBx[12]*	H'10C + N × 32	15 to 8, 7 to 0	MSG_DATA_4, MSG_DATA_5	
MBx[13], MBx[14]*	H'10E + N × 32	15 to 8, 7 to 0	MSG_DATA_6, MSG_DATA_7	

Note: \* x/N: 0 to 31 (Indicates the mailbox number)

### 16.3.4 Local Acceptance Filter Mask (LAFM)/Tx-Trigger Time (TTT)

This area is used as the local acceptance filter mask (LAFM) for receive boxes or as the Tx-trigger time (TTT) for transmit boxes.

**LAFM:** When the MBC bits are set to 001, 010, 011, 100, and 101, this field becomes the LAFM field. The LAFM is comprised of two 16-bit readable/writable areas. It allows a mailbox to accept more than one receive IDs.

Register Name	Address		Data Bus														Access Size	Field Name		
	HCAN0	HCAN1	15	14	13	12	11	10	9	8	7	6	5	4	3	2			1	0
MBx[15], MBx[16]	H'110+N×32	H'910+N×32	0	STDID[10:0]										0	0	EXTID[17:16]		16 bits	LAFM field	
MBx[17], MBx[18]	H'112+N×32	H'912+N×32	EXTID[15:0]																16 bits	

[Legend] x/N: 0 to 31 (Indicates the mailbox number)

**Figure 16.5 Acceptance Filter**

If a bit is set in the LAFM, the corresponding bit of a received CAN ID is ignored when the HCAN searches a mailbox with the matching CAN ID. If the bit is cleared, the corresponding bit of a received CAN ID must match the STD\_ID/EXT\_ID set in the mailbox to be stored. The configuration of the LAFM is same as the message control in a mailbox. If this function is not required, it must be filled with 0.

- Notes:
1. When the LAFM is used, the HCAN starts to find a matching ID from mailbox 31 down to mailbox 0. As soon as the HCAN finds one, it stops the search and stores the message into the mailbox. This means that a received message can only be stored into one mailbox.
  2. 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 STD\_ID, RTR, IDE,

and EXT\_ID differ to the ones originally set as they are updated with the STD\_ID, RTR, IDE, and EXT\_ID of the received message.

3. If the setting of the LAFM register that has already been set is changed, the HCAN should be set to halt mode before changing the setting. Do not access the LAFM during operation.
4. Do not access the undefined addresses. Correct operation cannot be guaranteed.

### LAFM Field:

Register Name	Address	Bit	Bit Name	Description
MBx[15], MBx[16]	H'110 + N×32	15	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.
		14 to 4	STDID_LAFM [10:0]	Filter Mask Bits[10:0] for CAN Base ID[10:0] 0: Corresponding bit to CAN base ID set in mailbox is valid 1: Corresponding bit to CAN base ID set in mailbox is invalid
		3, 2	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.
		1, 0	EXTID_LAFM [17:16]	Filter Mask Bits[17:16] for CAN Extended ID[17:16] 0: Corresponding bit to extended CAN base ID is valid 1: Corresponding bit to extended CAN base ID is invalid
MBx[17], MBx[18]	H'112 + N×32	15 to 0	EXTID_LAFM [15:0]	Filter Mask Bits[15:0] for CAN Extended ID[15:0] 0: Corresponding bit to extended CAN base ID is valid 1: Corresponding bit to extended CAN base ID is invalid

Note: \* x/N: 0 to 31 (Indicates the mailbox number)

### TTT:

When the MBC bits are set to 000, this field becomes a Tx-trigger time (TTT) field. The TTT is comprised of two 16-bit readable/writable areas.

Register Name	Address		Data Bus														Access Size	Field Name	
	HCAN0	HCAN1	15	14	13	12	11	10	9	8	7	6	5	4	3	2			1
MBx[15], MBx[16]	H'110+N×32	H'910+N×32	Tx-trigger time (absolute value)														16 bits	Tx-trigger control field	
MBx[17], MBx[18]	H'112+N×32	H'912+N×32	0	0	0	0	Offset[3:0]				0	0	0	0	Rep_Count[3:0]				16 bits

[Legend] x/N: 0 to 31 (Indicates the mailbox number)

**Figure 16.6 Tx-Trigger Control Field**

**Tx-Trigger Time Field:**

Register Name	Address	Bit	Bit Name	Description
MBx[15], MBx[16]*	H'110 + N × 32	15 to 0	TTT	Tx-Trigger Time Set the time that triggers message transmission using the absolute value.
MBx[17], MBx[18]*	H'112 + N × 32	15 to 12	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.
		11 to 8	Offset	Offset
		7 to 4	—	Reserved The write value should be 0. The value read as the initial value is not guaranteed.
		3 to 0	Rep_Count [3:0]	Repeat Counter Set the transmit cycle.

Note: \* x/N: 0 to 31 (Indicates the mailbox number)

The first 16-bit area sets the time that triggers message transmission using the absolute value. The second 16-bit area sets the basic cycle in the system matrix where the transmission must start (offset) and in the system matrix of the frequency for periodic transmission. When TXPR is set, the corresponding Tx-trigger time (TTT), repeat counter, and offset are downloaded into an internal register. When the internal TTT register matches the TCNTR value and the internal offset matches the CCR (cycle counter register) value, the corresponding mailbox automatically starts transmission. In order to enable this function, the TTE (time trigger enable) bit must be enabled (set to 1) and the timer (TCNTR) must be running (TCR15 = 1). When the TTE is cleared to 0 and the corresponding TXPR bit is set, it joins the queue for transmission immediately. If the repeat counter is not 0, transmission occurs periodically every Rep\_Count's basic cycle from CCR = offset to CCR = MAX\_CYCLE. In such case once TXPR is set by software, the HCAN does not clear the corresponding TXPR bit to carry on performing the periodic transmission. In order to stop the periodic transmission, TXPR must be cleared by TXCR or the Rep\_Count field must be cleared. If the repeat counter is 0, transmission occurs only once at the programmed basic cycle (i.e. CCR = offset and TCNTR = TTT).

The Tx-trigger time must not be set outside the TCNTR cycle if the compare-match timer clear/set function is used (by TCMR0 or CCM). During a time triggered transmission, only another one time triggered transmission can be triggered and a minimum difference of 200 peripheral clock cycles between them is allowed.

## 16.4 HCAN Control Registers

The following sections describe the HCAN control registers. Table 16.4 shows the address map.

Note: These registers can only be accessed in word size (16 bits).

**Table 16.4 HCAN Control Registers**

Channel	Address	Register Name	Abbreviation	Access Size (Bits)
0	H'FFFFD000	Master control register_0	MCR_0	16
	H'FFFFD002	General status register_0	GSR_0	16
	H'FFFFD004	HCAN-II_bit timing configuration register 1_0	HCAN-II_ BCR1_0	16
	H'FFFFD006	HCAN-II_bit timing configuration register 0_0	HCAN-II_ BCR0_0	16
	H'FFFFD008	Interrupt register_0	IRR_0	16
	H'FFFFD00A	Interrupt mask register_0	IMR_0	16
	H'FFFFD00C	Transmit error counter_0/ Receive error counter_0	TEC_0/REC_0	16
	1	H'FFFFD800	Master control register_1	MCR_1
H'FFFFD802		General status register_1	GSR_1	16
H'FFFFD804		HCAN-II_bit timing configuration register 1_1	HCAN-II_ BCR1_1	16
H'FFFFD806		HCAN-II_bit timing configuration register 0_1	HCAN-II_ BCR0_1	16
H'FFFFD808		Interrupt register_1	IRR_1	16
H'FFFFD80A		Interrupt mask register_1	IMR_1	16
H'FFFFD80C		Transmit error counter_1/ Receive error counter_1	TEC_1/REC_1	16

## 16.4.1 Register Descriptions

Legends for register descriptions are as follows:

Initial Value	: Register value after a reset
—	: Undefined value
R/W	: Readable/writable bit. The write value can be read.
R	: Read-only bit. The write value should always be 0.
R/WC0	: Readable/writable bit. If 0 is written to this bit, the bit is initialized; if 1 is written to this bit, it is ignored.
R/WC1	: Readable/writable bit. If 1 is written to this bit, the bit is initialized; if 0 is written to this bit, it is ignored.
W	: Write-only bit. Reading prohibited. If reserved, the write value should always be 0.
—/W	: Write-only bit. The read value is undefined.

## 16.4.2 Master Control Register\_n (MCR\_n) (n = 0, 1)

The master control register (MCR) is a 16-bit readable/writable register that controls the HCAN.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TST	TST	TST	TST	TST	TST	TST	TST	MCR		MCR	MCR		MCR	MCR	MCR
	7	6	5	4	3	2	1	0	7		5	4		2	1	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

Bit	Bit Name	Initial Value	R/W	Description
15	TST7	0	R/W	<p>Test Mode</p> <p>Enables/disables the test modes settable by TST[6:0]. When this bit is set, the following TST[6:0] are enabled.</p> <p>0: HCAN is in normal mode 1: HCAN is in test mode</p>
14	TST6	0	R/W	<p>Write CAN Error Counters</p> <p>Enables the TEC (transmit error counter) and REC (receive error counter) to be writable. The same value is written to TEC and REC at the same time.</p> <p>The maximum value that can be written to TEC and REC is D'255 (H'FF). This means that the HCAN cannot be forced into the bus off state. Before writing to TEC and REC, the HCAN needs to enter halt mode, and when writing to TEC and REC, the TST7 bit (MCR15) should be set to 1. The value written to TEC is used to write REC.</p> <p>0: TEC/REC is not writable but read-only</p>

Bit	Bit Name	Initial Value	R/W	Description
14	TST6	0	R/W	1: TEC/REC is writable with the same value at the same time
13	TST5	0	R/W	<p>Forced Error Passive</p> <p>Forces the HCAN to behave as an error passive node, regardless of the error counters.</p> <p>0: State of HCAN depends on error counters</p> <p>1: HCAN behaves as an error passive node, regardless of error counters</p>
12	TST4	0	R/W	<p>Automatic Acknowledge Mode</p> <p>Allows the HCAN to generate its own acknowledge bit in order to enable the self test. In order to enter self-test mode, the message transmitted needs to be read back, and there are 2 settings for this. One is to set (Enable Internal Loop = 1, Disable Tx Output = 1, and Disable Rx Input = 1), so that the Tx value is internally provided to the Rx. The other way is to set (Enable Internal Loop = 0, Disable Tx Output = 0, and Disable Rx Input = 0) and connect the Tx and Rx onto the CAN bus so that the transmitted data can be received via the CAN bus.</p> <p>0: HCAN does not generate its own acknowledge bit</p> <p>1: HCAN generates its own acknowledge bit</p>
11	TST3	0	R/W	<p>Disable Error Counters</p> <p>Enables/disables the error counters (TEC/REC). When this bit is disabled, the error counters (TEC/REC) remain unchanged and retain the current value. When this bit is enabled, the error counters (TEC/REC) operate according to the CAN specification.</p> <p>0: Error counters (TEC/REC) operate according to the CAN specification</p> <p>1: Error counters (TEC/REC) remain unchanged and retain the current value</p>



Bit	Bit Name	Initial Value	R/W	Description
10	TST2	0	R/W	<p>Disable Rx Input</p> <p>Controls the Rx to be supplied to the CAN Interface block. When this bit is enabled, the Rx pin value is supplied to the CAN interface block. When this bit is disabled, the Rx value for the CAN block is always retained or the Tx value internally connected if Enable Internal Loop = 1.</p> <p>0: Value of external Rx pin is supplied to the CAN interface block</p> <p>1: Enable Internal Loop = 0: Rx value is retained for the CAN interface block</p> <p>Enable Internal Loop = 1: Tx value is internally supplied to the CAN interface block</p>
9	TST1	0	R/W	<p>Disable Tx Output</p> <p>Controls the Tx to output transmit data or retain data. When this bit is enabled, the value of the internal transmit output pin appears on the Tx pin. When this bit is disabled, the Tx pin always retains the value.</p> <p>0: Value of external Tx pin is supplied from the CAN interface block</p> <p>1: Enable Internal Loop = 0: Tx value is retained</p> <p>Enable Internal Loop = 1: Tx is supplied to the internal Rx</p>
8	TST0	0	R/W	<p>Enable Internal Loop</p> <p>Enables/disables the internal Tx looped back to the internal Rx. For details, see section 16.7.1 Test Mode settings.</p> <p>0: Rx is supplied from the Rx Pin</p> <p>1: Rx is supplied from the internal Tx signal</p>
7	MCR7	0	R/W	<p>Auto-wake Mode</p> <p>Enables or disables auto-wake mode. When this bit is set, the HCAN automatically cancels sleep mode (MCR5) by detecting CAN bus activity (dominant bit). When this bit is not set, the HCAN does not automatically cancel sleep mode.</p> <p>0: Auto-wake by CAN bus activity disabled</p> <p>1: Auto-wake by CAN bus activity enabled</p>
6	—	0	R	<p>Reserved</p> <p>The write value should be 0. The read value is not guaranteed.</p>

Bit	Bit Name	Initial Value	R/W	Description
5	MCR5	0	R/W	<p>HCAN-II Sleep Mode</p> <p>Enables or disables sleep mode transition. When this bit is set, sleep mode is enabled. The HCAN waits for the completion of the current bus access before entering sleep mode. Until this mode is terminated the HCAN will ignore CAN bus operation. The two error counters (REC, TEC) will retain the same value during and after sleep mode. This mode will be exited in two ways:</p> <ul style="list-style-type: none"> <li>• Write 0 to this bit</li> <li>• If MCR7 is enabled, after detecting a dominant bit on the CAN bus</li> </ul> <p>When exiting this mode, the HCAN will synchronize with the CAN bus (by checking for 11 recessive bits) before restart. This means that, when the second way is used, the HCAN cannot receive the first message, however, CAN transceivers have the same feature, and software needs to be designed in this manner.</p> <p>Note: This mode is same as setting the module to halt mode and stopping the clock. This means that, the interrupt is generated from IRR0 when entering sleep mode. During sleep mode, only the MPI block is accessible, i.e., MCR/GSR/IRR/IMR are accessible. However, IRR1 cannot be cleared during sleep mode as it is an ORed signal of RXPR that cannot be cleared during sleep mode, therefore, it is recommended to set halt mode first and then make a transition to sleep mode.</p> <p>0: HCAN sleep mode is exited 1: Transition to HCAN sleep mode enabled</p> <p><b>Important:</b> Usage of sleep mode is limited. Be sure to carefully read section 16.8, Usage Notes.</p>

Bit	Bit Name	Initial Value	R/W	Description
4	MCR4	0	R/W	<p>CAN Endian Mode</p> <p>Controls whether the HCAN should transmit the messages in little endian mode or big endian mode. By using this bit, in other words, it is possible to set different endian mode to the HCAN and the external network. Note that this bit is only valid when data field is transmitted/received.</p> <p>0: Data field transmitted/received in big endian mode</p> <p>1: Data field transmitted/received in little endian mode</p>
3	MCR3	0	R/W	<p>Reserved</p> <p>The initial value should be retained.</p>
2	MCR2	0	R/W	<p>Message Transmission Priority</p> <p>Selects the order of transmission for pending transmit data.</p> <p>When this bit is set, pending transmit data are sent in order of the bit position in the transmit wait 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).</p> <p><b>Important:</b> This function cannot be used for timer triggered transmission.</p> <p>When this bit 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 with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit.</p> <p>0: Transmission order determined by message ID priority</p> <p>1: Transmission order determined by mailbox number priority (mailbox 31 → mailbox 1)</p>

Bit	Bit Name	Initial Value	R/W	Description
1	MCR1	0	R/W	<p>Halt Request</p> <p>Setting this bit causes the CAN controller to complete its current operation and then to cut off the CAN bus. The HCAN remains in halt mode until this bit is cleared. During halt mode, the CAN interface does not join the CAN bus activity or does not store messages nor transmit messages. All of the registers and mailbox contents are retained. The HCAN will complete the current operation if it is a transmitter or a receiver, and then enter halt mode. If the CAN bus is in the idle or intermission state, the HCAN will enter halt mode immediately. Entering halt mode is notified by IRR0 and GSR4. If a halt request is made during bus off, the HCAN-II remains bus off even after <math>128 \times 11</math> recessive bits. In order to exit this state, the halt state needs to be canceled by software.</p> <p>In halt mode, the HCAN configuration can be modified as it does not join the bus activity. This bit has to be cleared to 0 to re-join the CAN bus. After this bit is cleared, the CAN interface waits until it detects 11 recessive bits, and then joins the CAN bus.</p> <p>0: Normal operating mode 1: Halt mode transition request</p>

Bit	Bit Name	Initial Value	R/W	Description
0	MCR0	1	R/W	<p>Reset Request</p> <p>Controls resetting of the HCAN module. After detecting a reset request, the HCAN controller enters its reset routine, re-initializes the internal logic, and then set GSR3 and IRR0 to notify reset mode. Then the HCAN enters reset mode. During re-initialization, all the registers are cleared.</p> <p>This bit has to be cleared by writing a 0 to join the CAN bus. After this bit is cleared, the HCAN needs to be re-configured, waits until it detects 11 recessive bits, and then joins the CAN bus.</p> <p>After a power-on reset, this bit and GSR3 are always set. This means that a reset request has been made and the HCAN is in re-configuration mode.</p> <p>0: CAN interface normal operating mode (MCR0 = 0 and GSR3 = 0)</p> <p>Setting condition: When 0 is written after an HCAN reset</p> <p>1: Reset mode transition request of CAN interface</p>

### 16.4.3 General Status Register\_n (GSR\_n) (n = 0, 1)

The general status register (GSR) is a 16-bit read-only register that indicates the status of the HCAN.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											GSR	GSR	GSR	GSR	GSR	GSR
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

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	0	—	Reserved  The write value should be 0. The read value is not guaranteed.
5	GSR5	0	R	<b>Error Passive Status</b>  Indicates whether the CAN interface is error passive or not. This bit is set as soon as the HCAN enters the error passive state and is cleared when the module enters again the error active state. This means that this bit will remain high during error passive and during bus off. Thus to find out the correct state, both GSR5 and GRS0 must be considered.  0: HCAN is not error passive Setting condition: HCAN is in error active state 1: HCAN is error passive (if GSR0 = 0) Setting condition: When $TEC \geq 128$ or $REC \geq 128$
4	GSR4	0	R	<b>Halt/Sleep Status</b>  Indicates whether the CAN interface is in the halt/sleep state or not.  0: HCAN is not in the halt state nor sleep state 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 state

Bit	Bit Name	Initial Value	R/W	Description
3	GSR3	1	R	<p>Reset Status</p> <p>Indicates whether the CAN interface is in the reset state (configuration mode) or not.</p> <p>0: Normal operating state</p> <p>Setting condition: After an HCAN internal reset</p> <p>1: Reset state (configuration mode)</p>
2	GSR2	1	R	<p>Message Transmission In Progress Flag</p> <p>Indicates to the host CPU if the HCAN is processing transmission requests or if a transmission is completed. This bit is an ORed signal of all the TXPR bits. Note that the IRR8 (slot empty) is an ORed signal of all the TXACK/ABACK bits.</p> <p>0: Transmission in progress</p> <p>1: There is no message requested for transmission</p>
1	GSR1	0	R	<p>Transmit/Receive Warning Flag</p> <p>Indicates an error warning.</p> <p>0: Reset condition: When <math>TEC &lt; 96</math>, <math>REC &lt; 96</math>, or <math>TEC \geq 256</math></p> <p>1: When <math>96 \leq TEC &lt; 256</math> or <math>96 \leq REC</math></p>
0	GSR0	0	R	<p>Bus Off Flag</p> <p>Indicates that the HCAN is in the bus off state.</p> <p>0: Reset condition: Recovery from bus off state</p> <p>1: When <math>TEC \geq 256</math> (bus off state)</p>

#### 16.4.4 HCAN-II\_Bit timing Configuration Register n (HCAN-II\_BCR0\_n, HCAN-II\_BCR1\_n) (n = 0, 1)

The bit configuration registers (BCR0 and BCR1) are 16-bit readable/writable registers that set CAN bit timing parameters and the baud rate prescaler for the CAN interface.

For the following description the following definition is used:

$$\text{Timequanta} = \frac{BRP}{f_{clk}}$$

Where: BRP (baud rate predivider) is stored in BCR0 and  $f_{clk}$  is  $P\phi$  (peripheral clock).

- BCR1

For details on TSEG1 and TSEG2 settings, see table 16.4.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSEG1[3:0]					TSEG2[2:0]					SJW[1:0]				EG	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/W	R/W	R/W	—	—	R/W	R/W	—	—	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TSEG1[3]	0	R/W	Time Segment 1 (TSEG1[3:0] = BCR1[15:12])
14	TSEG1[2]	0	R/W	Set the segment for absorbing output buffer, CAN bus, and input buffer delay. A value from 4 to 16 time quanta can be set.
13	TSEG1[1]	0	R/W	0000: Setting prohibited 0001: Setting prohibited 0010: Setting prohibited
12	TSEG1[0]	0	R/W	0011: PRSEG + PHSEG1 = 4 time quanta 0100: PRSEG + PHSEG1 = 5 time quanta : 1111: PRSEG + PHSEG1 = 16 time quanta
11	—	0	—	Reserved  The write value should be 0. The read value is not guaranteed.
10	TSEG2[2]	0	R/W	Time Segment 2 (TSEG2[2:0] = BCR1[10:8])
9	TSEG2[1]	0	R/W	Set the segment for correcting 1-bit time error. A value from 2 to 8 time quanta can be set.
8	TSEG2[0]	0	R/W	000: Setting prohibited 001: PHSEG2 = 2 time quanta (setting prohibited depending on the condition so see table 16.5) 010: PHSEG2 = 3 time quanta 011: PHSEG2 = 4 time quanta 100: PHSEG2 = 5 time quanta 101: PHSEG2 = 6 time quanta 110: PHSEG2 = 7 time quanta 111: PHSEG2 = 8 time quanta
7, 6	—	0	—	Reserved  The write value should be 0. The read value is not guaranteed.



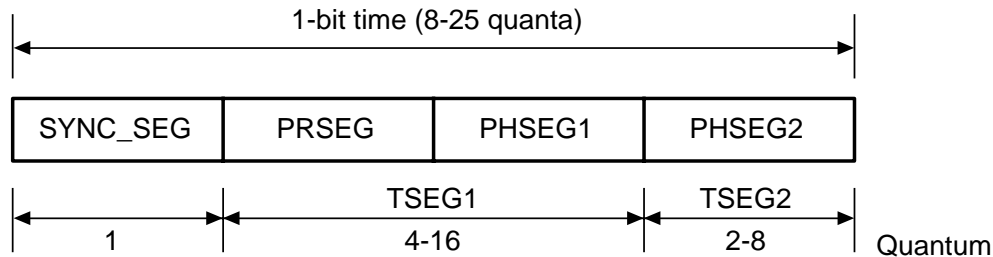
Bit	Bit Name	Initial Value	R/W	Description
5	SJW[1]	0	R/W	Re-Synchronization Jump Width (SJW[1:0] = BCR0[5:4])
4	SJW[0]	0	R/W	Set the synchronization jump width. 00: 1 time quantum 01: 2 time quanta 10: 3 time quanta 11: 4 time quanta
3, 2	—	0	—	Reserved The write value should be 0. The read value is not guaranteed.
1	EG	0	R/W	Edge Select (EG = BCR1[1]) Selects at which edge is to be used for re-synchronization. In order to comply with the standard CAN, 0 should be set. 0: Re-synchronization is performed at falling edge of Rx 1: Re-synchronization is performed at both rising and falling edges of Rx
0	BSP	0	R/W	Bit Sample Point (BSP = BCR1[0]) Sets the point at which data is sampled. <b>Important:</b> Sampling at three points is only available when the BRP[7:0] is programmed to be less than 4. 0: Bit sampling at one point (end of time segment 1) 1: Bit sampling at three points (end of time segment 1, and 1 time quantum before and after)

- BCR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
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 Value	R/W	Description
15 to 8	—	0	—	Reserved The write value should be 0. The read value is not guaranteed.
7	BRP[7]	0	R/W	Baud Rate Prescale (BRP[7:0] = BCR0 [7:0])
6	BRP[6]	0	R/W	Set the clock used for 1 time quantum.
5	BRP[5]	0	R/W	00000000: 1 • P $\phi$ (peripheral clock)
4	BRP[4]	0	R/W	00000001: 2 • P $\phi$ (peripheral clock)
3	BRP[3]	0	R/W	00000010: 3 • P $\phi$ (peripheral clock)
2	BRP[2]	0	R/W	: (BRP + 1) × P $\phi$ (peripheral clock)
1	BRP[1]	0	R/W	11111111: 256 × P $\phi$ (peripheral clock)
0	BRP[0]	0	R/W	

### About Bit Configuration Register:



**SYNC\_SEG:** Segment for establishing synchronization of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

**PRSEG:** Segment for adjusting physical delay between networks.

**PHSEG1:** Buffer segment for correcting phase drift (positive). (This segment is extended when synchronization (re-synchronization) is established.)

**PHSEG2:** Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronization (re-synchronization) is established.)

The CAN-bus bit rate is calculate as follows:

$$\text{Bit rate} = \text{fclk} / \{ (\text{BRP}[7:0] + 1) \times ( (\text{TSEG1}[3:0] + 1) + (\text{TSEG2}[2:0] + 1) + \text{SYNC\_SEG} ) \}$$

The SYNC\_SEG is fixed to 1 time quantum.

$$\text{fclk} = P\phi \text{ (peripheral clock)}$$

## BCR setting constraints

$$\text{TSEG1}[3:0] + 1 > \text{TSEG2}[2:0] + 1 \geq \text{SJW}[1:0] + 1$$

$$\text{TSEG1}[3:0] + \text{TSEG2}[2:0] + 3 = 8 \text{ to } 25 \text{ time quantum}$$

Register set values: TSEG1[3:0], TSEG2[2:0], and SJW[1:0]

These constraints allow the setting range shown in table 16.5 for TSEG1 and TSEG2 in the bit configuration register.

**Table 16.5 TSEG1 and TSEG2 Settings**

		TSEG2 (BCR[10:8])							
		001*	010	011	100	101	110	111	
		2	3	4	5	6	7	8	
TSEG1 (BCR[15:12])	0011	4	No	Yes	No	No	No	No	No
	0100	5	Yes	Yes	Yes	No	No	No	No
	0101	6	Yes	Yes	Yes	Yes	No	No	No
	0110	7	Yes	Yes	Yes	Yes	Yes	No	No
	0111	8	Yes	Yes	Yes	Yes	Yes	Yes	No
	1000	9	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1001	10	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1010	11	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1011	12	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1100	13	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1101	14	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1110	15	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1111	16	Yes	Yes	Yes	Yes	Yes	Yes	Yes	

Note: \* When BRP[7:0] = 0, TSEG2[2:0] ≥ 2

When BRP[7:0] ≥ 1, TSEG2[2:0] ≥ 1

### Examples:

- To have a bit rate of 1 Mbps with a Pφ (peripheral clock) frequency of fclk = 20 MHz, it is possible to set: BRP[7:0] = 1, TSEG1[3:0] = 5, and TSEG2[2:0] = 2. Then BCR1 should be written to H'5200 and BCR0 to H'0001.
- To have a bit rate of 500 kbps with a Pφ (peripheral clock) frequency of fclk = 16 MHz, it is possible to set: BRP[7:0] = 1, TSEG1[3:0] = 9, TSEG2[2:0] = 4. Then BCR1 should be written to H'9400 and BCR0 to H'0001.

**Important:** When BRP[7:0] = H'00, TSEG2[2:0] ≠ B'001

## 16.4.5 Interrupt Register\_n (IRR\_n) (n = 0, 1)

The interrupt register (IRR) is a 16-bit readable/writable register that contains status flags for the various interrupt sources.

- IRR

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRR 15	IRR 14	IRR 13	IRR 12	IRR 11	IRR 10	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	Bit Name	Initial Value	R/W	Description
15	IRR15	0	R/W	<p>Timer Compare Match Interrupt Flag 1</p> <p>Indicates that a compare-match condition occurred to the timer compare match register 1 (TCMR1). When the value set in TCMR1 matches the timer value (TCMR1 = TCNTR), this bit is set. This bit is not set if the TCMR1 value is H'0000.</p> <p>0: Timer compare match has not occurred to TCMR1</p> <p style="padding-left: 20px;">Clearing condition: Writing 1</p> <p>1: Timer compare match has occurred to TCMR1</p> <p style="padding-left: 20px;">Setting condition: TCMR1 matches the timer value (TCMR1 = TCNTR) if TMR1 = 0 or matches Cycle_Count + TCNTR[15:4] if TMR1 = 1</p>
14	IRR14	0	R/W	<p>Timer Compare Match Interrupt Flag 0</p> <p>Indicates that a compare-match condition occurred to the timer compare match register 0 (TCMR0). When the value set in TCMR0 matches the timer value (TCMR0 = TCNTR), this bit is set. This bit is not set if the TCMR0 value is H'0000.</p> <p>0: Timer compare match has not occurred to the TCMR0</p> <p style="padding-left: 20px;">Clearing condition: Writing 1</p> <p>1: Timer compare match has occurred to the TCMR0</p> <p style="padding-left: 20px;">Setting condition: TCMR0 matches the timer value (TCMR0 = TCNTR)</p>

Bit	Bit Name	Initial Value	R/W	Description
13	IRR13	0	R/W	<p>Timer Overrun Interrupt Flag</p> <p>Indicates that the timer has overrun and is reset to the LOSR (local offset register) value. This bit is set even when TCMR0 is enabled to clear/set the timer value and its value is set to H'FFFF.</p> <p>0: Timer has not overrun Clearing condition: Writing 1</p> <p>1: Timer has overrun Setting condition: When the timer (TCNTR) changes from H'FFFF to H'0000</p>
12	IRR12	0	R/W	<p>Wake-up on Bus Activity Interrupt Flag</p> <p>Indicates that a CAN bus activity is present. While the HCAN is in sleep mode and a recessive to dominant bit transition takes place on the CAN bus, this bit is set. The operation of this interrupt is set in the master control register (MCR7: Auto-wake mode). This interrupt is cleared by writing a 1 to this bit. Writing a 0 is ignored.</p> <p>0: Bus idle state Clearing condition: Writing 1</p> <p>1: CAN bus activity detected in HCAN sleep mode Setting condition: Recessive → dominant bit transition detection while in sleep mode</p>
11	IRR11	0	R/W	<p>Timer Compare Match Interrupt Flag 2</p> <p>Indicates that a compare-match condition occurred to the timer compare match register 2 (TCMR2). When the value set in TCMR2 matches the timer value (TCMR2 = TCNTR) or matches Cycle_Count + TCNTR[15:4] depending on the TMR2 (timer mode register) setting, this bit is set. This bit is not set if the TCMR2 value is H'0000.</p> <p>0: Timer compare match has not occurred to TCMR2 Clearing condition: Writing 1</p> <p>1: Timer compare match has occurred to TCMR2 Setting condition: TCMR2 matches the timer value (TCMR2 = TCNTR) if TMR2 = 0 or matches Cycle_Count + TCNTR[15:4] if TMR2 = 1</p>

Bit	Bit Name	Initial Value	R/W	Description
10	IRR10	0	R/W	<p>Cycle Counter Overrun Interrupt Flag</p> <p>Indicates that the Cycle_Counter has reached the maximum value (CMAX). When the CCR counter matches the CMAX value (CCR = CMAX), this bit is set and CCR is cleared. Note that setting CMAX = 0 disables the Cycle_Counter and no interrupt is generated.</p> <p>0: Cycle counter has not reached CMAX or CMAX = 0</p> <p>Clearing condition: Writing 1</p> <p>1: Cycle counter has reached CMAX and CMAX ≠ 0</p> <p>Setting condition: CCR matches the CMAX value (CCR = CMAX)</p>
9	IRR9	0	R	<p>Message Overrun/Overwrite Interrupt Flag</p> <p>Status flag indicating that new message has been received but the existing message in the mailbox has not been read due to the corresponding RXPR or RFPR set to 1. The received message is either abandoned (overrun) or overwritten dependant upon the NMC (new message control) bit. This bit is cleared by writing 1 to the correspondent bit in UMSR (unread message status register). Writing 0 is ignored.</p> <p>0: No message overrun/overwrite</p> <p>Clearing condition: Clearing of all bits in UMSR</p> <p>1: Receive message overrun and its storage has been rejected or message overwrite</p> <p>Setting condition: Message is received while the corresponding RXPR or RFPR = 1 and MBIMR = 0</p>

Bit	Bit Name	Initial Value	R/W	Description
8	IRR8	0	R	<p>Mailbox Empty Interrupt Flag</p> <p>Indicates that message transmission or transmission cancellation has been successfully made and this mailbox is now ready to accept a new message data for the next transmission. This bit is set when at least one TXPR bit is cleared. This bit is also set by an ORed signal of the TXACK and ABACK bits, therefore, this bit is automatically cleared when all the TXACK and ABACK bits are cleared. Writing 0 is ignored. Note that this bit does not represent that all TXPR bits are reset, whereas GSR2 does.</p> <p>0: Messages set for transmission or transmission cancellation not processed</p> <p>Clearing condition: All the TXACK and ABACK bits are cleared</p> <p>1: Message has been transmitted or canceled, and new message can be stored</p> <p>Setting condition: When one of the TXPR bits is cleared by completion of transmission or completion of transmission cancellation, i.e., when a TXACK or ABACK bit is set (if MBIMR = 0)</p>
7	IRR7	0	R/W	<p>Overload Frame Interrupt Flag</p> <p>Indicates that the HCAN has transmitted an overload frame. It remains latched until a reset by writing 1 to this bit. Writing 0 is ignored.</p> <p>0: Clearing condition: Writing 1</p> <p>1: Setting condition: Overload frame transmitted</p>

Bit	Bit Name	Initial Value	R/W	Description
6	IRR6	0	R/W	<p>Bus Off/Bus Off Recover Interrupt Flag</p> <p>This bit is set when the HCAN enters the bus-off state or when the HCAN leaves bus-off and returns to error-active. This is because the existing condition that 11 recessive bits have received 128 times when <math>TEC \geq 256</math> at the node or in the bus-off state. This bit remains latched even when the HCAN node cancels the bus-off state, and needs to be cleared by software. GSR0 should be read to determine whether the HCAN has become bus-off or error active. This bit is cleared by writing 1 even if the HCAN is still in the bus-off state. Writing 0 is ignored.</p> <p>0: Clearing condition: Writing 1</p> <p>1: Bus off state caused by transmit error or error active state returning from bus-off</p> <p>Setting condition: When 11 recessive bits have received 128 times when <math>TEC \geq 256</math> at the node or in the bus-off state</p>
5	IRR5	0	R/W	<p>Error Passive Interrupt Flag</p> <p>Indicates that the error passive state caused by the transmit or receive error counter. This bit is cleared by writing 1. Writing 0 is ignored. If this bit is cleared, the node may still be error passive.</p> <p>0: Clearing condition: Writing 1</p> <p>1: Error passive state caused by transmit/receive error</p> <p>Setting condition: When <math>TEC \geq 128</math> or <math>REC \geq 128</math></p>
4	IRR4	0	R/W	<p>Receive Overload Warning Interrupt Flag</p> <p>This bit is set and latched if the receive error counter (REC) reaches a value greater than 96. This bit is cleared by writing 1. Writing 0 is ignored. When the interrupt is cleared, REC still holds its value greater than 96.</p> <p>0: Clearing condition: Writing 1</p> <p>1: Error warning state caused by receive error</p> <p>Setting condition: When <math>REC \geq 96</math></p>



Bit	Bit Name	Initial Value	R/W	Description
3	IRR3	0	R/W	<p>Transmit Overload Warning Interrupt Flag</p> <p>This bit is set and latched if the transmit error counter (TEC) reaches a value greater than 96. This bit is cleared by writing 1. Writing 0 is ignored. When the interrupt is cleared, TEC still holds its value greater than 96.</p> <p>0: Clearing condition: Writing 1</p> <p>1: Error warning state caused by transmit error</p> <p>Setting condition: When <math>TEC \geq 96</math></p>
2	IRR2	0	R	<p>Remote Frame Request Interrupt Flag</p> <p>Indicates that a remote frame has been received in a mailbox. This bit is set if at least one receive mailbox contains a remote frame transmission request. This bit is cleared by ensuring all bits in the remote request wait register (RFPR) are cleared. Writing to this bit is ignored.</p> <p>0: Clearing condition: Clearing of all bits in RFPR</p> <p>1: At least one remote request is waiting</p> <p>Setting condition: When a remote frame is received and the corresponding MBIMR = 0</p>
1	IRR1	0	R	<p>Data Frame Received Interrupt Flag</p> <p>Indicates that there are waiting data frames received. If at least one receive mailbox contains a waiting message, this bit is set. This bit is cleared when all bits in the receive message waiting register (RXPR) are cleared, i.e. there is no waiting message in any receive mailbox. A logical OR from each set receive mailbox. Writing to this bit is ignored.</p> <p>0: Clearing condition: Clearing of all bits in RXPR</p> <p>1: Data frame received and stored in mailbox</p> <p>Setting condition: When data is received and the corresponding MBIMR = 0</p>

Bit	Bit Name	Initial Value	R/W	Description
0	IRR0	1	R/W	<p>Reset/Halt/Sleep Interrupt Flag</p> <p>Indicates that the CAN interface has been reset or halted and the HCAN is now in configuration mode or in sleep mode.</p> <p>An interrupt signal will be generated through this bit to notify the change of the HCAN's state to the host CPU if an MCR0 (software reset), MCR1 (halt), or MCR5 (sleep) request occurs. GSR can be read after this bit is set to figure out which state the HCAN is in.</p> <p><b>Important:</b> When a sleep mode request needs to be made, halt mode should be used beforehand. For details, see the MCR5 description.</p> <p>0: Clearing condition: Writing 1</p> <p>1: Transition to software reset mode, transition to halt mode, or transition to sleep mode without halt mode</p> <p>Setting condition: When reset/halt processing is completed after an MCR0 (software reset), MCR1 (halt), or MCR5 (sleep) is requested</p>

#### 16.4.6 Interrupt Mask Register\_n (IMR\_n) (n = 0, 1)

The interrupt mask register (IMR) is a 16-bit register that masks output of corresponding interrupt requests in the interrupt register (IRR). An interrupt request is masked if the corresponding bit is set to 1. This register can be read or written to at any time. IMR directly controls the generation of an interrupt request, but does not control the setting of the corresponding bit in IRR.

- IMR

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

Bit	Bit Name	Initial Value	R/W	Description
15	IMR15	1	R/W	Mask the corresponding IRR[15:0] interrupts.
14	IMR14	1	R/W	When this bit is set, the interrupt signal is masked, although the IRR setting is retained.
13	IMR13	1	R/W	0: Corresponding IRR is not masked (an interrupt request is generated for interrupt conditions) 1: Corresponding IRR interrupt is masked
12	IMR12	1	R/W	
11	IMR11	1	R/W	
10	IMR10	1	R/W	
9	IMR9	1	R/W	
8	IMR8	1	R/W	
7	IMR7	1	R/W	
6	IMR6	1	R/W	
5	IMR5	1	R/W	
4	IMR4	1	R/W	
3	IMR3	1	R/W	
2	IMR2	1	R/W	
1	IMR1	1	R/W	
0	IMR0	1	R/W	

#### 16.4.7 Transmit Error Counter<sub>n</sub> (TEC<sub>n</sub>) (n = 0, 1)/ Receive Error Counter<sub>n</sub> (REC<sub>n</sub>) (n = 0, 1)

The transmit error counter (TEC)/receive error counter (REC) is a 16-bit readable/(writable) 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 (References 2 and 3). In normal 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 bus off.

In test mode (i.e. MCR[15] = MCR[14] = 1), it is possible to write to this register. A same value can only be written to TEC and REC, and the value set in TEC is written to TEC and REC. When writing to this register, the HCAN needs to be in halt mode. This function is only intended for test purposes.

[Important] While the HCAN-II is in the bus-off status, the TEC and REC values are undefined.

- TEC/REC

Bit: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

TEC	TEC	TEC	TEC	TEC	TEC	TEC	TEC	REC	REC	REC	REC	REC	REC	REC	REC
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Initial Value: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

R/W: R/W\* R/W\* R/W\* R/W\* R/W\* R/W\* R/W\* R/W\* R/W\* R/W\* R/W\* R/W\* R/W\* R/W\* R/W\* R/W\*

Bit	Bit Name	Initial Value	R/W	Description
15	TEC7	0	R/W*	Transmit Error Counter
14	TEC6	0	R/W*	This register is incremented if an error is detected during transmission as specified on the CAN specification (see CAN specification document).
13	TEC5	0	R/W*	
12	TEC4	0	R/W*	
11	TEC3	0	R/W*	
10	TEC2	0	R/W*	
9	TEC1	0	R/W*	
8	TEC0	0	R/W*	
7	REC7	0	R/W*	Receive Error Counter
6	REC6	0	R/W*	This register is incremented if an error is detected during reception as specified on the CAN specification (see CAN specification document).
5	REC5	0	R/W*	
4	REC4	0	R/W*	
3	REC3	0	R/W*	
2	REC2	0	R/W*	
1	REC1	0	R/W*	
0	REC0	0	R/W*	

Note: \* It is only possible to write the value in test mode when MCR15 = MCR14 = 1.

## 16.5 HCAN Mailbox Registers

The HCAN mailbox registers control individual mailboxes. The address is mapped as follows.

Note: These registers can only be accessed in word size (16 bits).

**Table 16.6 HCAN Mailbox Registers**

Channel	Address (Bytes)	Register Name	Abbreviation	R/W	Access Size (Bits)
0	H'D020	Transmit pending request register 1_0	TXPR1_0	R/W	16
	H'D022	Transmit pending request register 0_0	TXPR0_0	R/W	
	H'D024				
	H'D026				
	H'D028	Transmit cancel register 1_0	TXCR1_0	R/W	16
	H'D02A	Transmit cancel register 0_0	TXCR0_0	R/W	
	H'D02C				
	H'D02E				
	H'D030	Transmit acknowledge register 1_0	TXACK1_0	R/W	16
	H'D032	Transmit acknowledge register 0_0	TXACK0_0	R/W	
	H'D034				
	H'D036				
	H'D038	Abort acknowledge register 1_0	ABACK1_0	R/W	16
	H'D03A	Abort acknowledge register 0_0	ABACK0_0	R/W	
	H'D03C				
	H'D03E				
	H'D040	Data frame receive pending register 1_0	RXPR1_0	R/W	16
	H'D042	Data frame receive pending register 0_0	RXPR0_0	R/W	
	H'D044				
	H'D046				
	H'D048	Remote frame receive pending register 1_0	RFPR1_0	R/W	16
	H'D04A	Remote frame receive pending register 0_0	RFPR0_0	R/W	
	H'D04C				
	H'D04E				
	H'D050	Mailbox interrupt mask register 1_0	MBIMR1_0	R/W	16
	H'D052	Mailbox interrupt mask register 0_0	MBIMR0_0	R/W	
	H'D054				
	H'D056				
	H'D058	Unread message status register 1_0	UMSR1_0	R/W	16
	H'D05A	Unread message status register 0_0	UMSR0_0	R/W	
H'D05C					
H'D05E					

Channel	Address (Bytes)	Register Name	Abbreviation	R/W	Access Size (Bits)
1	H'D820	Transmit pending request register 1_1	TXPR1_1	R/W	16
	H'D822	Transmit pending request register 0_1	TXPR0_1	R/W	
	H'D824				
	H'D826				
	H'D828	Transmit cancel register 1_1	TXCR1_1	R/W	16
	H'D82A	Transmit cancel register 0_1	TXCR0_1	R/W	
	H'D82C				
	H'D82E				
	H'D830	Transmit acknowledge register 1_1	TXACK1_1	R/W	16
	H'D832	Transmit acknowledge register 0_1	TXACK0_1	R/W	
	H'D834				
	H'D836				
	H'D838	Abort acknowledge register 1_1	ABACK1_1	R/W	16
	H'D83A	Abort acknowledge register 0_1	ABACK0_1	R/W	
	H'D83C				
	H'D83E				
	H'D840	Data frame receive pending register 1_1	RXPR1_1	R/W	16
	H'D842	Data frame receive pending register 0_1	RXPR0_1	R/W	
	H'D844				
	H'D846				
	H'D848	Remote frame receive pending register 1_1	RFPR1_1	R/W	16
	H'D84A	Remote frame receive pending register 0_1	RFPR0_1	R/W	
	H'D84C				
	H'D84E				
	H'D850	Mailbox interrupt mask register 1_1	MBIMR1_1	R/W	16
	H'D852	Mailbox interrupt mask register 0_1	MBIMR0_1	R/W	
	H'D854				
	H'D856				
	H'D858	Unread message status register 1_1	UMSR1_1	R/W	16
	H'D85A	Unread message status register 0_1	UMSR0_1	R/W	
	H'D85C				
	H'D85E				

### 16.5.1 Transmit Pending Request Register n (TXPR0n, TXPR1n) (n = 0, 1)

TXPR1 and TXPR0 are 16-bit readable/conditionally-writable registers that contain any transmit wait flags for the CAN module. TXPR1 controls mailbox 31 to mailbox 16, and TXPR0 controls mailbox 15 to mailbox 1. The host CPU makes a transmit message stored in a mailbox be in a transmit wait state by writing 1 to the corresponding bit. Writing 0 is ignored, and TXPR cannot be cleared by writing 0 and must be cleared by setting the corresponding TXCR bits. TXPR may be read by the host CPU to determine which, if any, transmissions are waiting. There is a transmit wait bit for all mailboxes except for mailbox 0. Writing 1 to a bit when the mailbox is set for reception is ignored, and TXPR is automatically cleared when an internal arbitration for transmission runs.

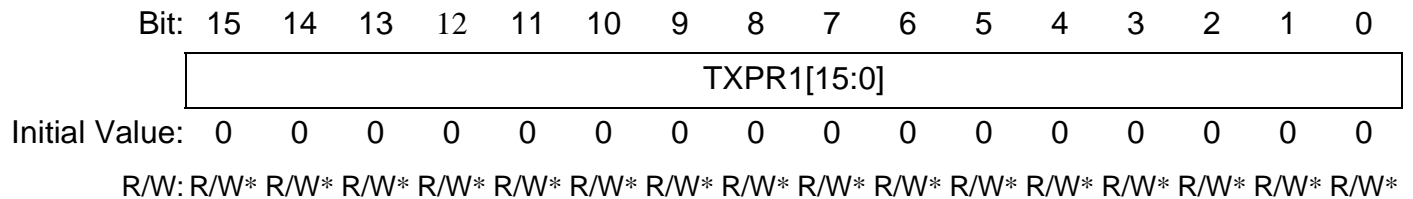
The HCAN will clear a transmit wait flag after successful transmission of its corresponding message or when a transmission wait cancellation is requested successfully from TXCR. TXPR is not cleared if the message is not transmitted due to the CAN node losing the arbitration processing or due to errors on the CAN bus, and the HCAN 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 wait is cleared and notified through mailbox empty interrupt flag (IRR8) and the correspondent bit in the abort acknowledgement register (ABACK).

If the status of TXPR changes, the HCAN shall ensure that in the ID priority scheme (MCR[2] = 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. For details, see section 16.7, Operation.

When the HCAN changes the state of any TXPR bit to 0, a mailbox empty 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 indicated in TXACK, and if a message transmission abortion is successful, it is indicated in ABACK. By checking these registers, the contents of the message data of the corresponding mailbox is modified to prepare for the next transmission.

**Important:** If mailbox 31 is used as a transmit buffer, there is a usage limitation. For details, see section 16.8, Usage Notes.

- TXPR1n (n = 0, 1)

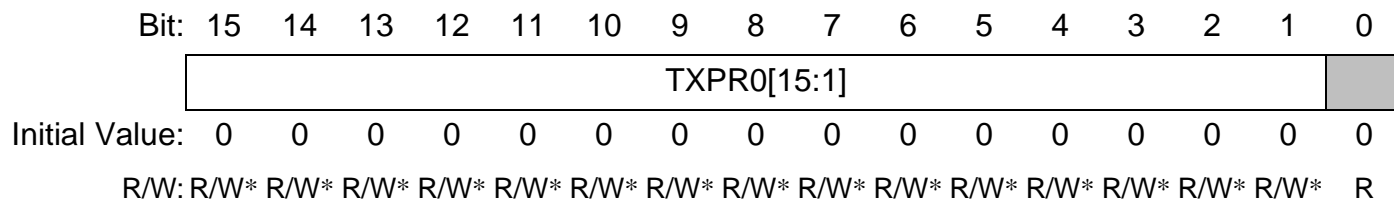


Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TXPR1[15:0]	0	R/W*	<p>Request the corresponding mailbox to transmit a CAN frame. Bits 15 to 0 correspond to mailboxes 31 to 16 respectively. When multiple bits are set, the order of the transmissions is determined by MCR2 (CAN-ID or mailbox number).</p> <p>0: Corresponding mailbox is in transmit message idle state Clearing condition: Completion of message transmission or message transmission wait abortion (automatically cleared)</p> <p>1: Transmission request made for corresponding mailbox</p>

Note: \* Only 1 can be written to set a mailbox for transmission.



- TXPR0n (n = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
15 to 1	TXPR0[15:1]	0	R/W*	<p>Request the corresponding mailbox to transmit a CAN frame. Bits 15 to 1 correspond to mailboxes 15 to 1 respectively. When multiple bits are set, the order of the transmissions is determined by MCR2 (CAN-ID or mailbox number).</p> <p>0: Corresponding mailbox is in transmit message idle state Clearing condition: Completion of message transmission or message transmission wait abortion (automatically cleared)</p> <p>1: Transmission request made for corresponding mailbox</p>
0	—	0	R	<p>Reserved</p> <p>This bit is always 0 as this is a receive-only mailbox. Writing 1 to this bit is ignored. The read value is not guaranteed.</p>

Note: \* Only 1 can be written to set a mailbox for transmission.

## 16.5.2 Transmit Cancel Register n (TXCR1n, TXCR0n) (n = 0, 1)

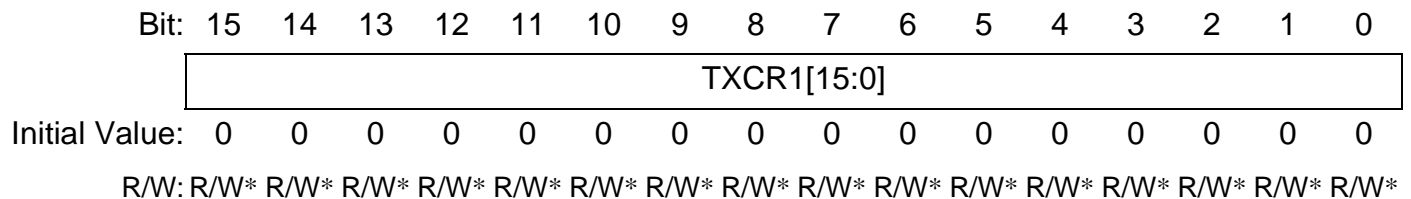
TXCR1 and TXCR0 are 16-bit readable/conditionally-writable registers. TXCR1 controls mailbox 31 to mailbox 16, and TXCR0 controls mailbox 15 to mailbox 1. This register is used by the host CPU to request the transmission wait messages in TXPR to be cancelled. To clear the corresponding bit in TXPR, the host CPU must write 1 to the bit in TXCR. Writing 0 is ignored.

When transmission cancellation has succeeded, the CAN controller clears the corresponding TXPR and 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 and TXCR bits, 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 and TXCR bits, and sets the corresponding ABACK bit. If an attempt is made by the host CPU to cancel a mailbox transmission that is not transmit-waiting, it shall have no effect, and will be automatically cleared when an internal arbitration for transmission runs.

**Important:** For details on the method of canceling a transmit wait, see section 16.7, Operation.

**Important:** If mailbox 31 is used as a transmit buffer, there is a usage limitation. For details, see section 16.8, Usage Notes.

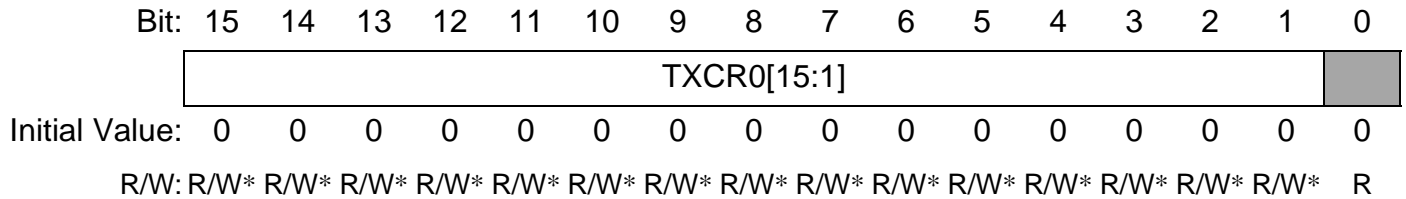
- TXCR1n (n = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TXCR1[15:0]	0	R/W*	<p>Request the corresponding mailbox, that is in the queue for transmission, to cancel its transmission wait. Bits 15 to 0 correspond to mailboxes 31 to 16 and TXPR1[15:0] respectively.</p> <p>0: Corresponding mailbox is in transmit message cancellation idle state Clearing condition: Completion of transmit wait cancellation (automatically cleared)</p> <p>1: Transmit wait cancellation request made for corresponding mailbox</p>

Note: \* 1 can be written only to a mailbox that is requested for transmission or set for transmission.

- TXCR0n (n = 0, 1)



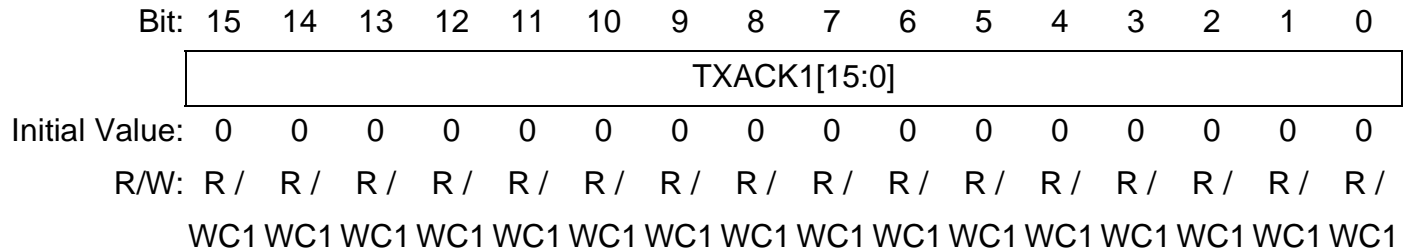
Bit	Bit Name	Initial Value	R/W	Description
15 to 1	TXCR0[15:1]	0	R/W*	Request the corresponding mailbox, that is in the queue for transmission, to cancel its transmission wait. Bits 15 to 1 correspond to mailboxes 15 to 1 and TXPR0[15:1] respectively.  0: Corresponding mailbox is in transmit message cancellation idle state Clearing condition: Completion of transmit wait cancellation (automatically cleared)  1: Transmit wait cancellation request made for corresponding mailbox
0	—	0	R	Reserved  This bit is always 0 as this is a receive-only mailbox. Writing 1 to this bit is ignored. The read value is always 0.

Note: \* 1 can be written only to a mailbox that is requested for transmission or set for transmission.

### 16.5.3 Transmit Acknowledge Register n (TXACK1n, TXACK0n) (n = 0, 1)

TXACK1 and TXACK0 are 16-bit readable/conditionally-writable registers. These registers notify the host CPU that a mailbox transmission has been successfully made. When a transmission has succeeded, the HCAN sets the corresponding bit in TXACK. The host CPU can clear a TXACK bit by writing 1 to the corresponding bit. Writing 0 is ignored.

- TXACK1n (n = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TXACK1[15:0]	0	R/WC1	<p>Notify that the requested transmission of the corresponding mailbox has been finished successfully. Bits 15 to 0 correspond to mailboxes 31 to 16 respectively.</p> <p>0: Clearing condition: Writing 1</p> <p>1: Corresponding mailbox has successfully transmitted message (data or remote frame)</p> <p>Setting condition: Completion of message transmission for corresponding mailbox</p>

- TXACK0n (n = 0, 1)

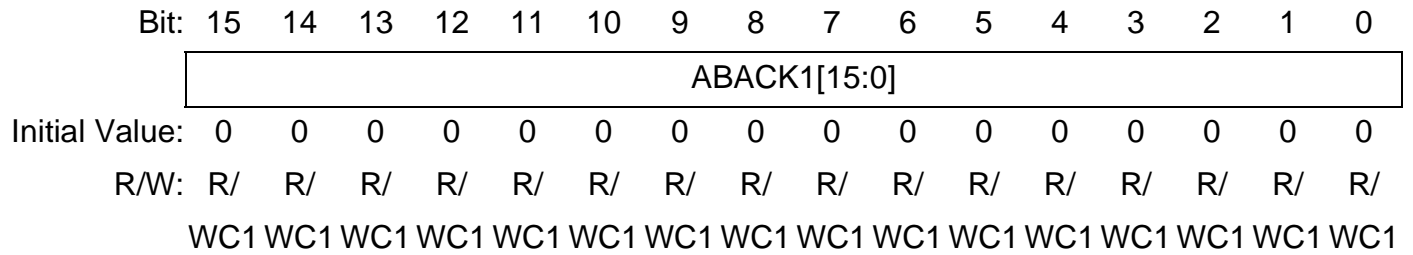
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0[15:1]															
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R
	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	TXACK0[15:1]	0	R/WC1	Notify that the requested transmission of the corresponding mailbox has been finished successfully. Bits 15 to 1 correspond to mailboxes 15 to 1 respectively.  0: Clearing condition: Writing 1 1: Corresponding mailbox has successfully transmitted message (data or remote frame)  Setting condition: Completion of message transmission for corresponding mailbox
0	TXACK0[0]	0	R	Reserved  This bit is always 0 as this is a receive-only mailbox. Writing 1 to this bit is ignored. The read value is always 0.

### 16.5.4 Abort Acknowledge Register n (ABACK1n, ABACK0n) (n = 0, 1)

ABACK1 and ABACK0 are 16-bit readable/conditionally-writable registers. These registers notify the host CPU that a mailbox transmission has been aborted as per its request. When an abort has succeeded, the HCAN sets the corresponding bit in ABACK. The host CPU can clear the ABACK bit by writing 1 to the corresponding bit. Writing 0 is ignored. An ABACK bit is used by the HCAN to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

- ABACK1n (n = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ABACK1[15:0]	0	R/WC1	<p>Notify that the requested transmit wait cancellation of the corresponding mailbox has been finished successfully. Bits 15 to 0 correspond to mailboxes 31 to 16 respectively.</p> <p>0: Clearing condition: Writing 1</p> <p>1: Corresponding mailbox has cancelled transmission of message (data or remote frame)</p> <p>Setting condition: Completion of transmit wait cancellation for corresponding mailbox</p>

- ABACK0n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[15:1]															
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
	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	ABACK0[15:1]	0	R/WC1	Notify that the requested transmit wait cancellation of the corresponding mailbox has been finished successfully. Bits 15 to 1 correspond to mailboxes 15 to 1 respectively.  0: Clearing condition: Writing 1 1: Corresponding mailbox has cancelled transmission of message (data or remote frame)  Setting condition: Completion of transmit wait cancellation for corresponding mailbox
0	0	0	R	Reserved  This bit is always 0 as this is a receive-only mailbox. Writing 1 to this bit is ignored. The read value is always 0.

### 16.5.5 Data Frame Receive Pending Register n (RXPR1n, RXPR0n) (n = 0, 1)

RXPR1 and RXPR0 are 16-bit readable/conditionally-writable registers. RXPR is a register that contains the data frame receive complete flags associated with receive mailboxes. When a CAN data frame is successfully stored in a receive mailbox, the corresponding bit is set in RXPR. The corresponding bit is cleared by writing 1. Writing 0 is ignored. However, the bit may only be set if the mailbox is set by its MBC (mailbox configuration) to receive data frames. When an RXPR bit is set, IRR1 (data frame receive interrupt flag) is also set if its MBIMR (mailbox interrupt mask register) is not set, and the interrupt signal is generated if IMR1 is not set. These bits are only set by receiving data frames and not by receiving remote frames.

If a data frame is overwritten/overrun with a remote frame or vice versa, UMSR, RXPR, and RFPR will be set for the same mailbox. In this case the application needs to check the RTR bit within the mailbox control field to understand the nature of the message on the mailbox. Consequently when UMSR is set, both RXPR and RFPR should be checked and, if necessary, cleared.

- RXPR1n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR1[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/
	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RXPR1[15:0]	0	R/WC1	Set receive mailboxes corresponding to mailboxes 31 to 16 respectively.  0: Clearing condition: Writing 1  1: Corresponding mailbox has received a CAN data frame  Setting condition: Completion of data frame reception in corresponding mailbox

- RXPR0n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR0[15:0]															
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/
	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RXPR0[15:0]	0	R/WC1	Set receive mailboxes corresponding to mailboxes 15 to 0 respectively.  0: Clearing condition: Writing 1  1: Corresponding mailbox has received a CAN data frame  Setting condition: Completion of data frame reception in corresponding mailbox



## 16.5.6 Remote Frame Receive Pending Register n (RFPR1n, RFPR0n) (n = 0, 1)

RFPR1 and RFPR0 are 16-bit readable/conditionally-writable registers. RFPR is a register that contains the remote request flags associated with the receive mailboxes. When a CAN remote frame is successfully stored in a receive mailbox, the corresponding bit is set in RFPR. The corresponding bit is cleared by writing 1. Writing 0 is ignored. There is a bit for all mailboxes. However, the bit is only set if the mailbox is set by its MBC (mailbox configuration) to receive remote frames. When an RFPR bit is set, IRR2 (remote frame request interrupt flag) is also set if its MBIMR (mailbox interrupt mask register) is not set, and the interrupt signal is generated if IMR2 is not set. These bits are only set by receiving remote frames and not by receiving data frames.

If a data frame is overwritten/overrun with a remote frame or vice versa, UMSR, RXPR, and RFPR will be set for the same mailbox. In this case the application needs to check the RTR bit within the mailbox control field to understand the nature of the message on the mailbox. Consequently when UMSR is set, both RXPR and RFPR should be checked and, if necessary, cleared.

- RFPR1n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR1[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/
	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RFPR1[15:0]	0	R/WC1	Remote request wait flags for receive mailboxes 31 to 16.  0: Clearing condition: Writing 1 1: Corresponding mailbox has received a remote frame  Setting condition: Completion of remote frame reception in corresponding mailbox

- RFPR0n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR0[15:0]															
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/
	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1

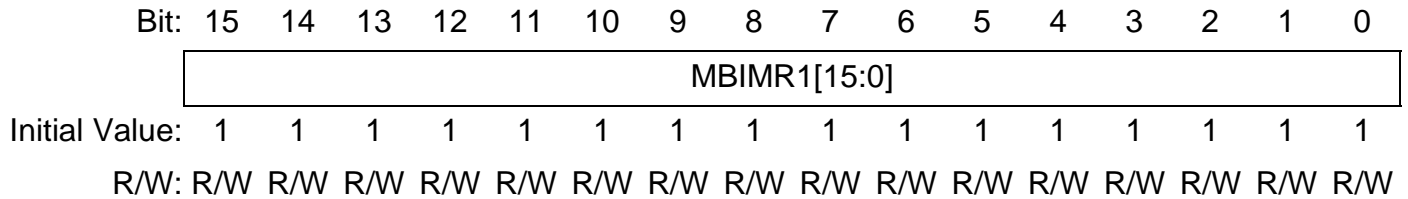
Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RFPR0[15:0]	0	R/WC1	Remote request wait flags for receive mailboxes 15 to 0.  0: Clearing condition: Writing 1 1: Corresponding mailbox has received a remote frame  Setting condition: Completion of remote frame reception in corresponding mailbox

### 16.5.7 Mailbox Interrupt Mask Register n (MBIMR1n, MBIMR0n) (n = 0, 1)

MBIMR1 and MBIMR0 are 16-bit readable/writable registers. MBIMR only masks IRR (IRR1: data frame receive interrupt, IRR2: remote frame request interrupt, IRR8: mailbox empty interrupt, and IRR9: message overflow interrupt) related to the mailbox activities. If a mailbox is set for reception, the generation of a receive interrupt (IRR1, IRR2, and IRR9) is masked but the setting of the corresponding bit in RXPR, RFPR, or UMSR is not modified. Similarly when a mailbox is set for transmission, the generation of an interrupt signal and setting of an mailbox empty interrupt due to successful transmission or abortion of transmission (IRR8) are masked, however, clearing the corresponding TXPR/TXCR bit and setting the TXACK bit for successful transmission are not masked, or clearing the corresponding TXPR/TXCR bit and setting the ABACK bit for abortion of the transmission are not masked.

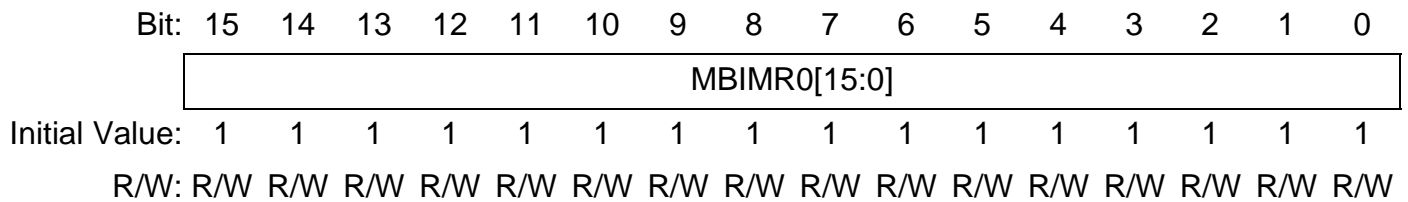
A mask is set by writing 1 to the corresponding bit for the mailbox activity to be masked. At a reset all mailbox interrupts are masked.

- MBIMR1n (n = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	MBIMR1[15:0]	1	R/W	Enable or disable interrupts requests from individual mailbox 31 to mailbox 16 respectively.  0: Interrupt request from IRR1/IRR2/IRR8/IRR9 enabled  1: Interrupt request from IRR1/IRR2/IRR8/IRR9 disabled

- MBIMR0n (n = 0, 1)



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	MBIMR0[15:0]	1	R/W	Enable or disable interrupt requests from individual mailbox 15 to mailbox 0 respectively.  0: Interrupt request from IRR1/IRR2/IRR8/IRR9 enabled  1: Interrupt request from IRR1/IRR2/IRR8/IRR9 disabled

### 16.5.8 Unread Message Status Register n (UMSR1n, UMSR0n) (n = 0, 1)

UMSR1 and UMSR0 are 16-bit readable/writable registers that record the receive mailboxes whose contents have not been accessed by the host CPU prior to a new message being received. If the host CPU has not cleared the corresponding bit in RXPR/RFPR when a new message for a mailbox is received, the corresponding UMSR bit is set. This bit is cleared by writing 1. Writing 0 is ignored.

If a mailbox is set for transmission, the corresponding UMSR bit cannot be set.

- UMSR1n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR1[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/
	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	UMSR1[15:0]	0	R/WC1	Indicate that an unread message has been overwritten/overrun for mailboxes 31 to 16. 0: Clearing condition: Writing 1 1: Unread message is overwritten by a new message or overrun Setting Condition: When a new message is received before RXPR/RFPR is cleared.

- UMSR0n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR0[15:0]															
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/
	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	UMSR0[15:0]	0	R/WC1	<p>Indicate that an unread message has been overwritten for mailboxes 15 to 0.</p> <p>0: Clearing condition: Writing 1</p> <p>1: Unread message is overwritten by a new message</p> <p>Setting Condition: When a new message is received before RXPR/RFPR is cleared.</p>

## 16.6 Timer Registers

The timer is a new function for the HCAN-II. The timer is 16 bits and supports several clock sources. It is divided by a prescale counter to reduce the clock speed. It also supports two input capture registers (ICR1 and ICR0) and three compare match registers (TCMR2, TCMR1, and TCMR0). The address map is as follows.

Note: These registers can only be accessed in word size (16 bits).

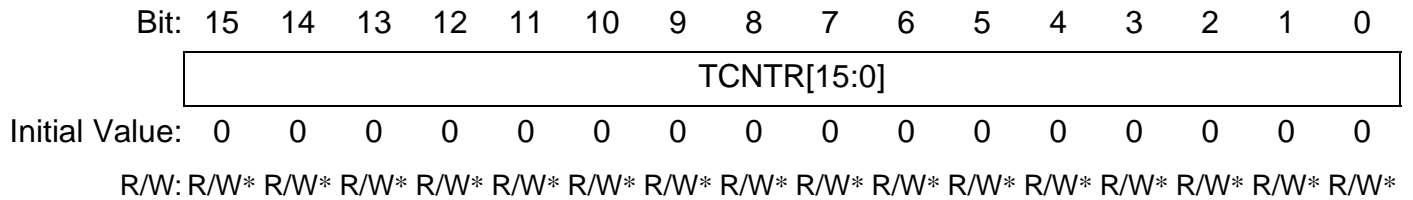
**Table 16.7 HCAN Timer Registers**

<b>Channel</b>	<b>Address (Bytes)</b>	<b>Register Name</b>	<b>Abbreviation</b>	<b>Access Size (Bits)</b>
0	H'D080	Timer counter register 0	TCNTR0	16
	H'D082	Timer control register_0	TCR_0	16
	H'D084	Timer status register_0	TSR_0	16
	H'D086	Timer drift correction register 0	TDCR0	16
	H'D088	Local offset register 0	LOSR0	16
	H'D08A	Input capture register for cycle counter 0	ICR0-cc0	16
	H'D08C	Input capture register for timer counter 0	ICR0-tm0	16
	H'D08E	Input capture register 1_0	ICR1_0	16
	H'D090	Timer compare match register 0_0	TCMR0_0	16
	H'D092	Timer compare match register 1_0	TCMR1_0	16
	H'D094	Timer compare match register 2_0	TCMR2_0	16
	H'D096	Cycle counter register 0	CCR0	16
	H'D098	Cycle maximum register 0	CMAX0	16
	H'D09A	Timer mode register_0	TMR_0	16
	H'D09C	Cycle counter double buffer 0	CCR_buf0	16
	H'D09E	Input capture double buffer 0	ICR0_buf0	16
	1	H'D880	Timer counter register 1	TCNTR1
H'D882		Timer control register_1	TCR_1	16
H'D884		Timer status register_1	TSR_1	16
H'D886		Timer drift correction register 1	TDCR1	16
H'D8D8		Local offset register 1	LOSR1	16
H'D88A		Input capture register for cycle counter 1	ICR0-cc1	16
H'D88C		Input capture register for timer counter 1	ICR0-tm1	16
H'D88E		Input capture register 1_1	ICR1_1	16
H'D890		Timer compare match register 0_1	TCMR0_1	16
H'D892		Timer compare match register 1_1	TCMR1_1	16
H'D894		Timer compare match register 2_1	TCMR2_1	16
H'D896		Cycle counter register 1	CCR1	16
H'D898		Cycle maximum register 1	CMAX1	16
H'D89A		Timer mode register_1	TMR_1	16
H'D89C		Cycle counter double buffer 1	CCR_buf1	16
H'D89E		Input capture double buffer 1	ICR0_buf1	16

Note: It is recommended that the timer should be disabled (TCR15 = 0) to change the setting of the registers related to the timer.

### 16.6.1 Timer Counter Register n (TCNTRn) (n = 0, 1)

The timer counter register (TCNTR) is a 16-bit readable/writable register that allows the CPU to monitor and modify the value of the free-running timer counter. When the timer matches TCMR0 (timer compare match register 0) and TCR11 is set to 1, TCNTR is set to LOSR (local offset register) and counting starts again.



Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TCNTR[15:0]	0	R/W*	Indicate the value of the free-running timer.

Note: \* This register is cleared by the compare match condition.

## 16.6.2 Timer Control Register\_n (TCR\_n) (n = 0, 1)

The timer control register (TCR) is a 16-bit readable/writable register that controls the operation of the timer. This register should be set before each periodical transmission or the deadline monitor register is set and the timer operation starts.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCR 15	TCR 14	TCR 13	TCR 12	TCR 11	TCR 10	TCR9		TCR7		TPSC 5	TPSC 4	TPSC 3	TPSC 2	TPSC 1	TPSC 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	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
15	TCR15	0	R/W	<p>Enable Timer</p> <p>When this bit is set, the timer runs. When this bit is cleared, the timer completes the current cycle (notified by timer overrun or a compare match condition on TCMR0) and is cleared to 0.</p> <p>0: Timer stops running and is cleared at the end of current cycle</p> <p>1: Timer is running</p> <p><b>Important:</b> There is a failure on the timer function in the SH7058. This bit must be written to 0 not to activate the timer.</p>
14	TCR14	0	R/W	<p>Disable ICR0</p> <p>Enables or disables the input capture register 0 (ICR0). When this bit is enabled, the timer value is always captured every time a start of frame (SOF) is output to the CAN bus, whether the HCAN is a transmitter or receiver. When this bit is disabled, the value of ICR0 remains latched.</p> <p>0: ICR0 is disabled and holds the current value</p> <p style="padding-left: 20px;">Clearing condition: TCR9 = 1 when CAN-ID of receive message is equal to the ID of a mailbox with CCM set</p> <p>1: ICR0 is enabled and captures the timer value at every SOF</p>



Bit	Bit Name	Initial Value	R/W	Description
13	TCR13	0	R/W	<p>Timestamp Control for Reception</p> <p>Specifies whether the timestamp in the message control of each mailbox is recorded at the start of frame (SOF) or end of frame (EOF) when a message is received. This bit selects the trigger for the input capture register 1 (ICR1) that is used to timestamp for transmit mailboxes.</p> <p>0: Timestamp is recorded at the SOF of every message received</p> <p>1: Timestamp is recorded at the EOF of every message received</p> <p><b>Important:</b> The timestamp recorded at the SOF of every message received is not supported by the SH7058. When a receive timestamp is used, this bit should be set to 1.</p>
12	TCR12	0	R/W	<p>Timestamp Control for Transmission</p> <p>Specifies whether the timestamp of each transmit mailbox is recorded at the point that the corresponding TXPR bit is set or the corresponding TXACK bit is set when a transmit request is made. This bit selects the trigger for the input capture register 1 (ICR1) that is used for timestamp of receive mailboxes. The input capture register 1 (ICR1) is used for timestamp, regardless of whether ICR0 is enabled or disabled.</p> <p>0: Timestamp is recorded at the point that the TXPR bit is set for message transmission</p> <p>1: Timestamp is recorded at the point that the TXACK bit is set for message transmission</p>
11	TCR11	0	R/W	<p>Timer Clear/Set Control by TCMR0</p> <p>Specifies whether the timer is to be cleared and set to LOSR when TCMR0 matches TCNTR. TCMR0 is also capable of generating an interrupt signal to the host CPU via IRR15.</p> <p>0: Timer is not cleared by TCMR0</p> <p>1: Timer is cleared by TCMR0</p>

Bit	Bit Name	Initial Value	R/W	Description
10	TCR10	0	R/W	<p>Timer Clear/Set Control by CCM</p> <p>Specifies whether the timer is to be cleared and set to LOSR by the CAN-ID compare match for receive mailboxes. When a mailbox stores a receive message, the timer counter (TCNTR) is automatically cleared and set to LOSR, if the CCM bit of the corresponding mailbox and this bit are set. CCM is not capable of generating an interrupt signal since this is performed by the message receive interrupt (IRR1) or remote frame request interrupt (IRR2).</p> <p>0: Timer is not cleared/set by CCM 1: Timer is cleared and set to LOSR by CCM</p>
9	TCR9	0	R/W	<p>ICR0 Automatic Disable by CCM</p> <p>Specifies whether ICR0 is to be disabled by the CAN-ID compare match (CCM) for receive mailboxes. When a mailbox stores a receive message, bit 14 of this register (TCR14) is automatically cleared and the value of ICR0 is retained, if the CCM bit of the corresponding mailbox and this bit are set.</p> <p>0: TCR14 is not cleared by CCM 1: TCR14 is automatically cleared by CCM</p>
8	—	0	—	<p>Reserved</p> <p>Writing 0 to this bit is ignored. The read value is not guaranteed.</p>
7	TCR7	0	R/W	<p>Drift Correction Control</p> <p>Specifies whether TCNTR is to be incremented by 2 or 0 every time TCNTR reaches the cycle specified by TDCR. If this function is not required, TDCR must be set to H'0000.</p> <p>0: Timer is incremented by 0 (i.e. retains the same value for one clock cycle) every cycle specified by TDCR. 1: Timer is incremented by 2 every cycle specified by TDCR (see TDCR description).</p>
6	—	0	—	<p>Reserved</p> <p>Writing 0 to this bit is ignored. The read value is not guaranteed.</p>

Bit	Bit Name	Initial Value	R/W	Description
5	TCR5	0	R/W	HCAN-II Timer Prescaler
4	TCR4	0	R/W	Divide the source clock (2 • HCAN peripheral clock) before it is used for the timer. The following relationship exists between source clocks and the timer
3	TCR3	0	—	
2	TCR2	0	—	
1	TCR1	0	—	
0	TCR0	0	—	
				000000: 1 • source clock
				000001: 2 • source clock
				000010: 4 • source clock
				000011: 6 • source clock
				000100: 8 • source clock
				:
				111111: 126 • source clock

### 16.6.3 Timer Status Register\_n (TSR\_n) (n = 0, 1)

The timer status register (TSR) is a 16-bit read-only register that allows the host CPU to monitor the timer compare match status and the timer overrun status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												TSR	TSR	TSR	TSR	TSR
												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

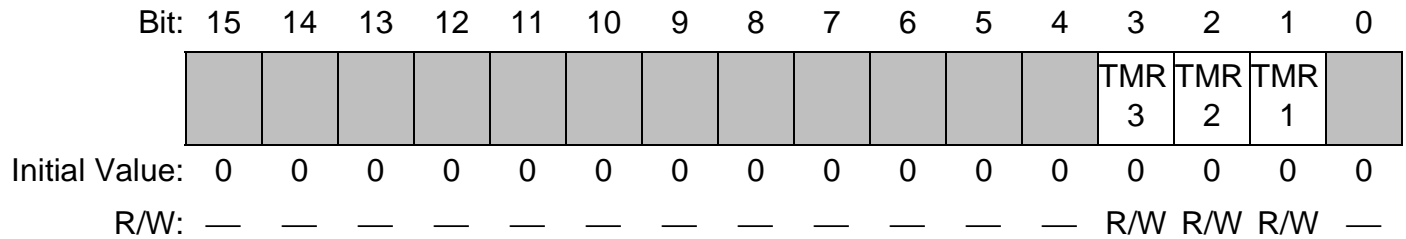
Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	0	—	Reserved Writing 0 to this bit is ignored. The read value is not guaranteed.
4 to 0	TSR[4:0]	0	R	These bits are read-only that allow the CPU to monitor the status of the cycle counter, the timer, and the compare match registers. Writing to these bits is ignored.

Bit	Bit Name	Initial Value	R/W	Description
4	TSR4	0	R	<p>Cycle Counter Overflow Flag</p> <p>Indicates that the cycle counter has reached its maximum value and is reset to H'0. Setting CMAX = 0 makes the cycle counter be disabled and TSR4 be always cleared to 0.</p> <p>0: Cycle counter has not overflow Clearing condition: Writing 1 to IRR10 (cycle counter overflow interrupt)</p> <p>1: Cycle counter has overflow Setting condition: When the cycle counter value changes from the maximum value (CMAX) to H'0</p>
3	TSR3	0	R	<p>Timer Compare Match Flag 2</p> <p>Indicates that a compare-match condition occurred to the timer compare match register 2 (TCMR2). When the value set in TCMR2 matches the timer value (TCMR2 = TCNTR), this bit is set. This bit is not set if the TCMR2 value is H'0000. Also, this bit is read-only and is cleared when IRR11 (timer compare match interrupt 2) is cleared.</p> <p>0: Timer compare match has not occurred to TCMR2 Clearing condition: Writing 1 to IRR11 (timer compare match interrupt 2)</p> <p>1: Timer compare match has occurred to TCMR2 Setting condition: TCMR2 matches the timer value (TCMR2 = TCNTR)</p>

Bit	Bit Name	Initial Value	R/W	Description
2	TSR2	0	R	<p>Timer Compare Match Flag 1</p> <p>Indicates that a compare-match condition occurred to the timer compare match register 1 (TCMR1). When the value set in TCMR1 matches the timer value (TCMR1 = TCNTR), this bit is set. This bit is not set if the TCMR1 value is H'0000. Also, this bit is read-only and is cleared when IRR15 (timer compare match interrupt 1) is cleared.</p> <p>0: Timer compare match has not occurred to TCMR1 Clearing condition: Writing 1 to IRR15 (timer compare match interrupt 1)</p> <p>1: Timer compare match has occurred to TCMR1 Setting condition: TCMR1 matches the timer value (TCMR1 = TCNTR)</p>
1	TSR1	0	R	<p>Timer Compare Match Flag 0</p> <p>Indicates that a compare-match condition occurred to the timer compare match register 0 (TCMR0). When the value set in TCMR0 matches the timer value (TCMR0 = TCNTR), this bit is set. This bit is not set if the TCMR0 value is H'0000. Also, this bit is read-only and is cleared when IRR14 (timer compare match interrupt 0) is cleared.</p> <p>0: Timer compare match has not occurred to TCMR0 Clearing condition: Writing 1 to IRR14 (timer compare match interrupt 0)</p> <p>1: Timer compare match has occurred to TCMR0 Setting condition: TCMR0 matches the timer value (TCMR0 = TCNTR)</p>
0	TSR0	0	R	<p>Timer Overrun Flag</p> <p>Indicates that the timer has overrun and is reset to H'0000. This bit is set even when TCMR0 is set to H'FFFF and is enabled to clear the timer value.</p> <p>0: Timer has not overrun Clearing condition: Writing 1 to IRR13 (timer overrun interrupt)</p> <p>1: Timer has overrun Setting condition: When the timer value changes the value from H'FFFF to H'0000</p>

## 16.6.4 Timer Mode Register\_n (TMR\_n) (n = 0, 1)

The timer mode register (TMR) is a 16-bit readable/writable register that specifies the value to be used for the timer functions.



Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	0	—	Reserved Writing 0 to this bit is ignored. The read value is not guaranteed.
3	TMR3	0	R/W	Timestamp Value Specifies whether the timestamp for transmission and reception contains the timer value (TCNTR) or the value of Cycle_Counter + TCNTR[15:4]. This function is very useful for time triggered transmission. 0: TCNTR[15:0] is used for the timestamp 1: Cycle_Counter + TCNTR[15:4] is used for the timestamp
2	TMR2	0	R/W	TCMR2 Control Specifies whether the timer compare match 2 is compared with the timer value (TCNTR) or with Cycle_Counter + TCNTR[15:4]. 0: TCNTR[15:0] is used for a compare match 1: Cycle_Counter + TCNTR[15:4] is used for a compare match
1	TMR1	0	R/W	TCMR1 Control Specifies whether the timer compare match 1 is compared with the timer value (TCNTR) or with Cycle_Counter + TCNTR[15:4]. 0: TCNTR[15:0] is used for a compare match 1: Cycle_Counter + TCNTR[15:4] is used for a compare match
0	—	0	—	Reserved Writing 0 to this bit is ignored. The read value is not guaranteed.

## 16.6.5 Timer Drift Correction Register n (TDCRn) (n = 0, 1)

The timer drift correction register (TDCR) is a 16-bit readable/writable register. The purpose of this register is to adjust the drift of the timer caused by a different clock running at other CAN nodes on the same system. When TCNTR reaches to the cycle specified by this register, the timer value is incremented by 2 or 0 (i.e. retains the same value). This register does not point at a specific time nor a specific cycle. This means, if  $TCNTR/2 > TDCR$ , the drift correction will be performed more than twice (unless TCMR0 is used to clear TCNTR before it reaches the second cycle). When TDCR is set to H'0000, the drift correction will not be performed at all.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDCR[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 Value	R/W	Description
15 to 0	TDCR[15:0]	0	R/W	Timer Drift Correction Register  Set the value of the cycle to adjust the drift of the timer.  <b>Important:</b> For a proper operation of the timer, the maximum value must be $TDCR \leq 8000$ (hexadecimal).

## 16.6.6 Local Offset Register n (LOSRn) (n = 0, 1)

The local offset register (LOSR) is a 16-bit readable/writable register that sets a local offset value to TCNTR. When TCNTR is cleared by an overflow, timer compare match, or CAN-ID compare match, TCNTR starts running at the value set in this register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOSR[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 Value	R/W	Description
15 to 0	LOSR[15:0]	0	R/W	Local Offset Register  Indicate the value of the local offset for TCNTR to start with.

### 16.6.7 Cycle Counter Register n (CCRn) (n = 0, 1)

The cycle counter register (CCR) is a 4-bit readable/writable register that stores the number of the basic cycles for time triggered transmission. Its value is incremented by one every time the free-running counter (TCNTR) is cleared to 0 by a compare match condition on TCMR0.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	0	R	Reserved
3 to 0	CCR[3:0]	0	R/W	Cycle Counter Indicate the number of the current basic cycles of the matrix cycle for timer triggered transmission.

### 16.6.8 Cycle Counter Double-Buffer Register n (CCR\_buf n) (n = 0, 1)

The cycle counter double-buffer register (CCR\_buf) is a 4-bit readable/writable register that is used when the cycle counter (CCR) and timer counter (TCNTR) are read from or written to simultaneously to refer the same basic cycle constantly. (This register is used as a temporary retain register to prevent the 20-bit counter value from being updated in CPU access.)

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	0	R	Reserved
3 to 0	CCR_buf [3:0]	0	R/W	Cycle Counter Double-Buffer A temporary retain buffer when accessing the basic cycle of the matrix cycle for timer triggered transmission (CCR) and timer counter (TCNTR) simultaneously. The CCR_buf value indicates the same value as write/read data to/from CCR.

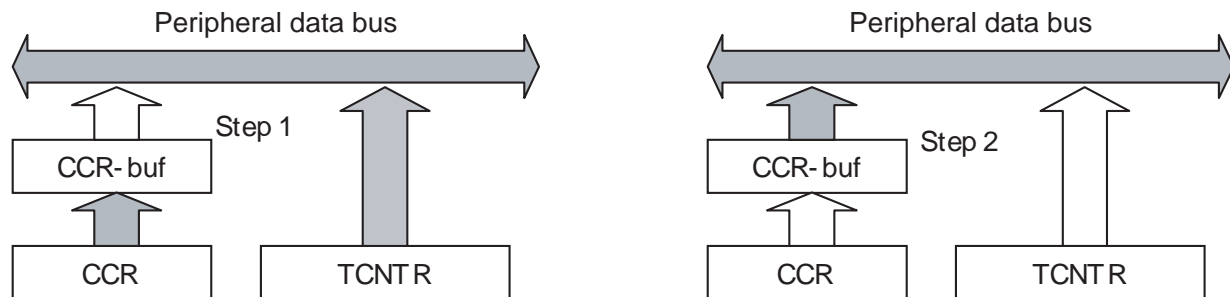


The procedure for accessing the cycle counter (CCR) and timer counter (TCNTR) using the cycle counter double-buffer (CCR\_buf) is described below.

- Read operation

Read the timer counter (TCNTR). (The value of the cycle counter (CCR) is written to the cycle counter double-buffer (CCR\_buf) simultaneously.)

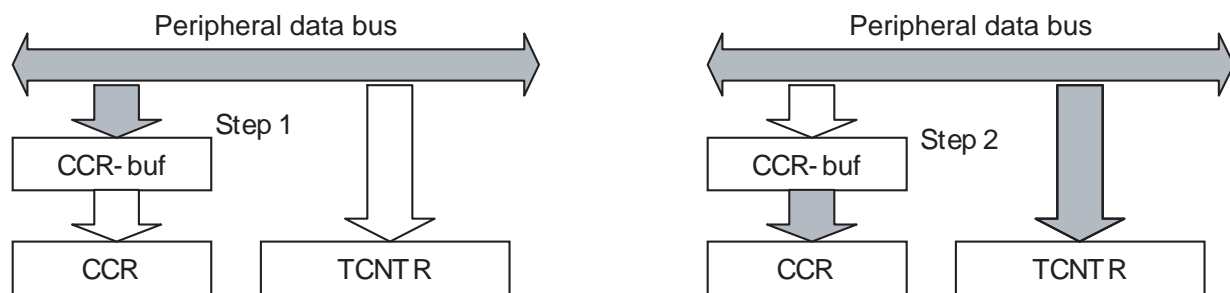
Then read the cycle counter double-buffer (CCR\_buf).



- Write operation

Write data to the cycle counter double-buffer (CCR\_buf).

Then write data to the timer counter (TCNTR). (The value of the cycle counter double-buffer (CCR\_buf) is written to the cycle counter (CCR) simultaneously.)



## 16.6.9 Cycle Maximum Register n (CMAXn) (n = 0, 1)

The cycle maximum register (CMAX) is a 4-bit readable/writable register that stores the maximum value for the cycle counter (CCR) for timer triggered transmission to set the number of basic cycles in the matrix system. When the cycle counter reaches the maximum value (CCR = CMAX), the cycle counter is cleared to 0 and an interrupt is generated on IRR10.

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

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	0	R	Reserved
3 to 0	CMAX[3:0]	0	R/W	Cycle Maximum Value Store the maximum value of CCR. The initial value of CMAX is 0 making the cycle counter be disabled. During the time trigger setting, the requested value must be programmed.

## 16.6.10 Input Capture Registers n (ICR0\_cc n, ICR0\_buf, ICR0\_tm n, ICR1 n) (n = 0, 1)

The input capture registers are composed of one 4-bit readable/writable register (ICR0\_cc) and two 16-bit readable/writable registers (ICR0\_tm and ICR1).

- ICR0\_cc n (n = 0, 1)

ICR0\_cc can be used for global synchronization, when used with ICR0\_tm. The current basic cycle value (Cycle\_Counter) is captured at the SOF if ICR0\_cc is enabled by bit 14 in TCR, regardless whether the receive message matches the ID set in the receive mailboxes or not. If ICR0\_cc is disabled by bit 14 in TCR, it retains the current value.

- ICR0\_buf n (n = 0, 1): Input Capture Double-Buffer Register

A temporary retain buffer that accesses ICR0\_cc and ICR0\_tm simultaneously. The ICR0\_buf value is same as the ICR0\_cc value.

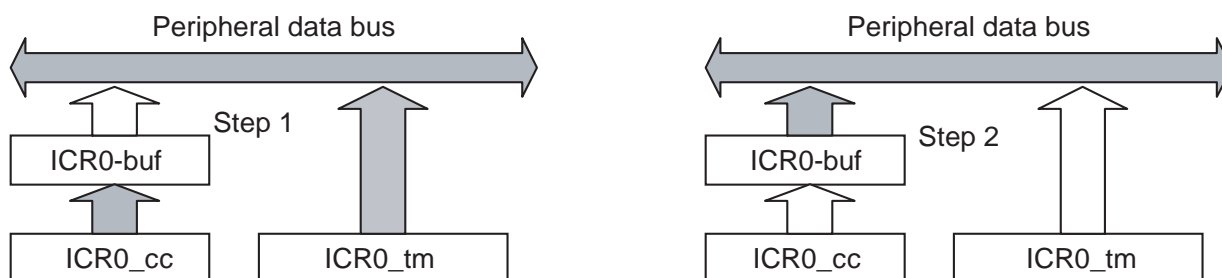
- ICR0\_tm n (n = 0, 1)

ICR0\_tm can be used for global synchronization, when used with ICR0\_cc. The timer value is captured at the SOF if ICR0\_tm is enabled by bit 14 in TCR, regardless whether the receive message matches the ID set in the receive mailboxes or not. If ICR0\_tm is disabled by bit 14 in TCR, it retains the current value.

- Read operation for ICR0\_cc, ICR0\_buf, and ICR0\_tm

Read the input capture register (ICR0\_tm). (The value of ICR0\_cc is written to the input capture double-buffer register (ICR0\_buf) simultaneously.)

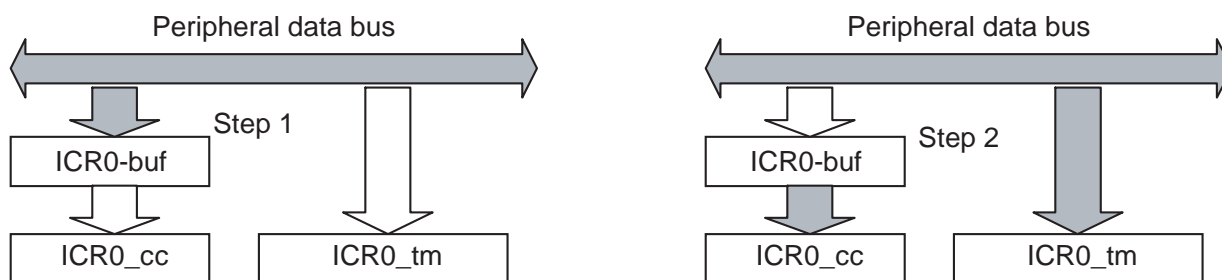
Then read the input capture double-buffer (ICR0\_buf).



- Write operation for ICR0\_cc, ICR0\_buf, and ICR0\_tm

Write data to the input capture double-buffer (ICR0\_buf).

Then write data to the input capture register (ICR0\_tm). (The value of the input capture double-buffer (ICR0\_buf) is written to ICR0\_cc simultaneously.)



- ICR1 n (n = 0, 1)

ICR1 records the timestamp for messages to be transmitted and received. Bit 13 (for reception) and bit 12 (for transmission) in TCR control at which point the timestamp should be recorded. The difference to ICR0 is that ICR1 cannot be disabled so that the timestamps recorded on messages are always correct.

- ICR0\_cc/ICR0\_buf

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ICR0_cc[3:0]/ ICR0_buf[3:0]			
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	0	R	Reserved The write value should be 0. The read value is not guaranteed.
3	ICR0_cc	0	R/W*	This register samples the value of the cycle counter register (CCR) at every SOF on the CAN bus when enabled by TCR[14].
2	[3:0]/	0	R/W*	
1	ICR0_buf	0	R/W*	
0	[3:0]	0	R/W*	

Note: \* This register can be written to, however, the written value is ignored.

- ICR0\_tm/ICR1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICR0_tm[15:0], ICR1[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 Value	R/W	Description
15 to 0	ICR0_tm[15:0]	0	R/W*	This register samples the value of the timer (TCNTR) at every SOF on the CAN bus when enabled by TCR[14].

Note: \* This register can be written to, however, the written value is ignored.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ICR1[15:0]	0	R/W*	This register samples the value of the timer (TCNTR) at the condition specified by bit 13 (for reception) and bit 12 (for transmission) in TCR.

Note: \* This register can be written to, however, the written value is ignored.

### 16.6.11 Timer Compare Match Registers n (TCMR0n, TCMR1n, TCMR2n) (n = 0, 1)

- TCMR0, TCMR1, and TCMR2

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR0[15:0], TCMR1[15:0], TCMR2[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 Value	R/W	Description
15 to 0	TCMR0[15:0] , TCMR1[15:0], TCMR2[15:0]	0	R/W	<p>The timer compare match registers (TCMR0, TCMR1, and TCMR2) are 16-bit readable/writable registers that generate interrupt signals, clear/set the timer value (only supported by TCMR0), or clear the transmit messages in the queue (only supported by TCMR2). (These registers offer exactly the same function except for the clear of the timer and the clear of the transmission.) The value used for the compare can be set independently for each register, using bits 1, 2, and 3 in TMR (timer mode register), to be the timer value (TCNTR[15:0]) or the value of Cycle_Count + TCNTR[15:4].</p> <p>Interrupts are flagged by bits 15, 14, and 11 in IRR when a compare match occurs, and these bits cannot be prevented from being set in IRR except when the TCMR value is H'0000. The generation of interrupt signals can be masked by bits 15, 14, and 11 in IMR. When a compare match occurs and IRR15 (or IRR14 or IRR11) is set, bits 2, 1, and 3 in TSR (HCAN timer status register) are also set. Clearing the IRR bit also clears the corresponding bit in TSR.</p> <p>The timer value is cleared and LOSR is set when a compare match occurs to TCMR0 if bit 11 in TCR is enabled (timer clear/set function). TCMR1 and TCMR2 do not have this function.</p> <p>The messages in the transmit queue are cleared only when a compare match occurs to TCMR2 (cancellation of the messages in the transmit queue). TCMR1 and TCMR0 do not have this function.</p> <p><b>Important:</b> TCMR0 and TCMR2 are not supported by the SH7058. The setting must be H'0000.</p>

## 16.7 Operation

### 16.7.1 Test Mode Settings

The HCAN has various test modes. Bits TST[7:0] (bits 15 to 8 in MCR) are used to select the HCAN-II test mode. The initial settings allow the HCAN to operate in normal mode. The following table is examples for test modes.

**Table 16.8 Test Modes**

Bit15: TST7	Bit14: TST6	Bit13: TST5	Bit12: TST4	Bit11: TST3	Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	0	0	0	0	0	Normal mode (initial value)
1	0	0	0	1	0	1	0	Listen-only mode (receive-only)
1	0	0	1	0	0	0	0	Self test mode 1 (external)
1	0	0	1	0	1	1	1	Self test mode 2 (internal)
1	1	0	0	0	0	0	0	Error passive mode 1
1	0	1	0	0	0	0	0	Error passive mode 2

**Normal Mode:** The HCAN operates normally.

**Listen-Only Mode:** The ISO-11898 requires this mode for baud rate detection etc. The error counters are disabled so that TEC/REC does not increment the values, and the Tx output is disabled so that the HCAN does not generate error frames.

**Self Test Mode 1:** The HCAN generates its own acknowledge bit. The Rx and Tx pins must be connected to the CAN bus.

**Self Test Mode 2:** The HCAN generates its own acknowledge bit. The Rx and Tx pins do not need to be connected to the CAN bus or any external devices, as the internal Tx is looped back to be connected to the internal Rx.

**Important:** In self test modes 1 and 2, the transmitted data is not received in the internal mailbox.

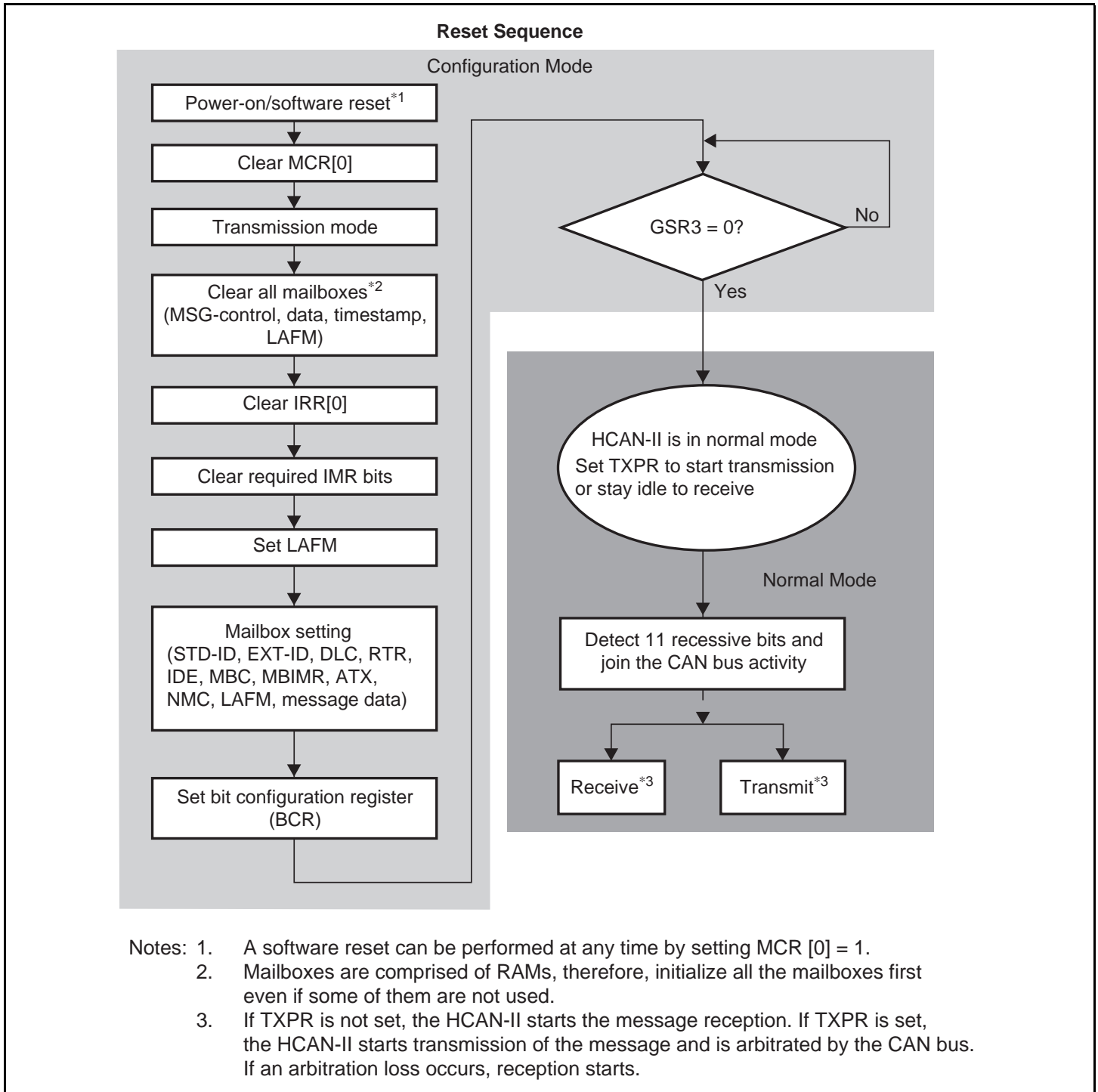
**Error Passive Mode 1:** The HCAN can be forced to become an error passive node by writing a value (greater than 127) to the error counter. (MCR1 must be 1 when writing to the error counter). The value written to TEC is used to write to REC, so only the same value can be set to these registers. Also, the HCAN needs to be in halt mode when writing to TEC/REC.

**Error Passive Mode 2:** The HCAN can be forced to become an error passive node by setting TST5.

## 16.7.2 HCAN Settings

- Reset Sequence

The following sequence is an example to set the HCAN after a software or hardware reset. After a reset, all the registers are initialized, therefore, the HCAN needs to be set before joining the CAN bus activity. Please read the notes carefully.



**Figure 16.7 Reset Sequence**

### 16.7.3 Message Transmission Sequence

#### (1) Event Triggered Transmission

- Message Transmission Request

Figure 16.8 is an example to transmit a CAN frame onto the bus. As described in Register Description, note that IRR8 is set when the TXACK or ABACK bit is set. This means that one of the mailboxes has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas, GSR2 means that there is currently no transmission request made (TXPR = H'0000).

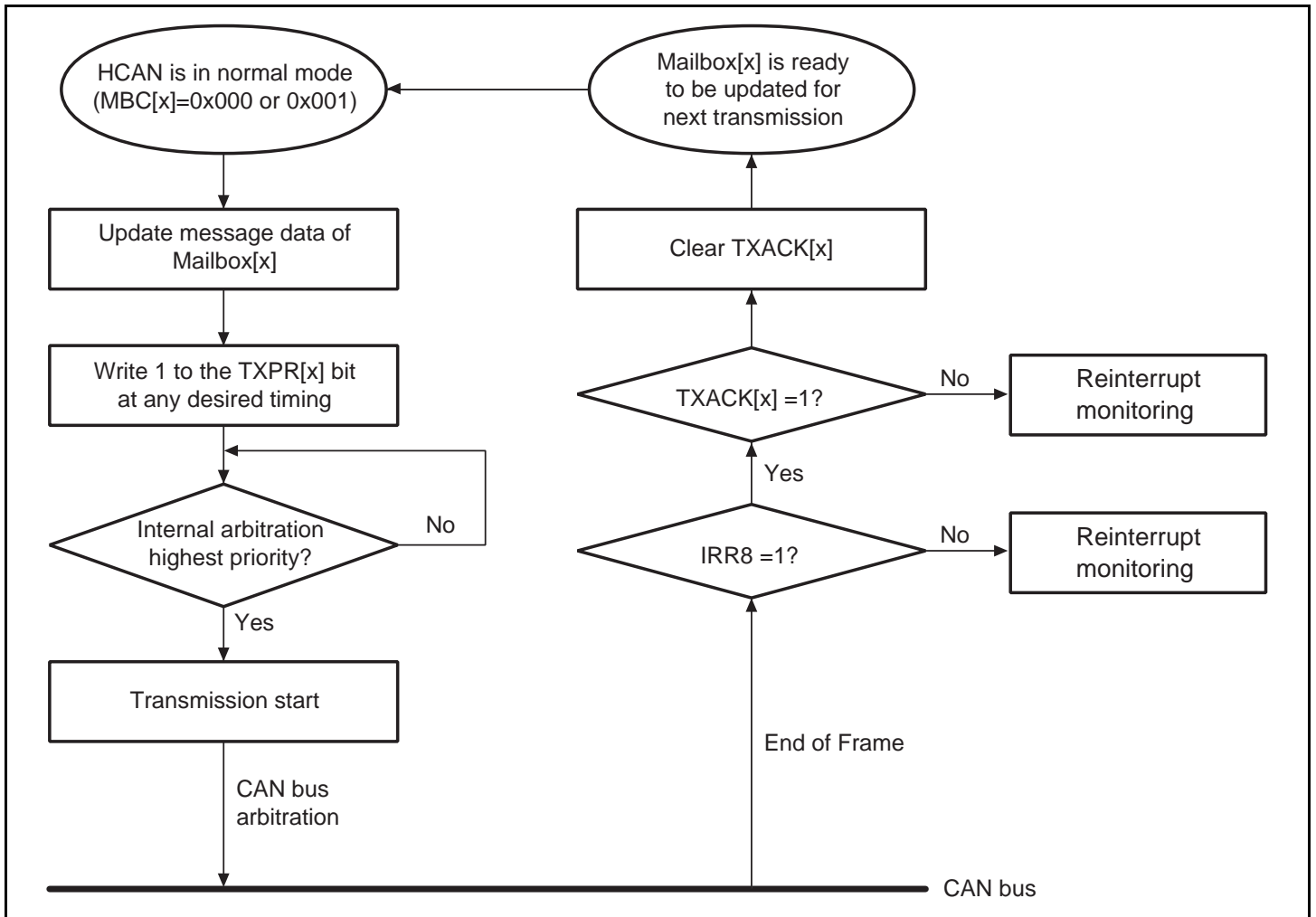
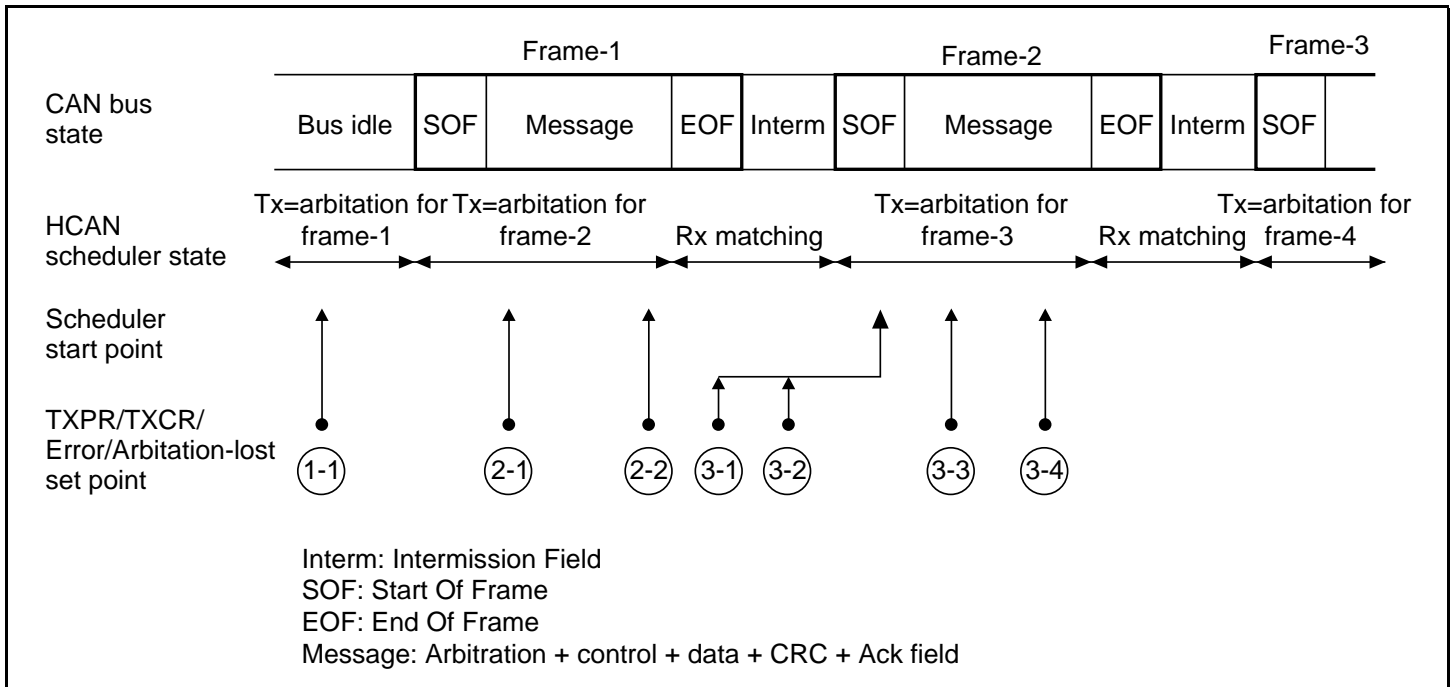


Figure 16.8 Transmission Request



- Internal Arbitration for Transmission

Figure 16.9 explains how the HCAN manages to schedule transmit-requested messages in the correct order based on the CAN ID. “Internal arbitration” picks up the highest priority message among transmit-requested messages.



**Figure 16.9 Internal Arbitration for Transmission**

The HCAN scheduler, which runs internal arbitration, has 2 states – Tx arbitration state and Rx matching state. The HCAN scheduler is in the Rx matching state if the CAN bus is in the EOF or intermission cycles, or otherwise is in the Tx arbitration state. When a transmit request or transmit abort request is made in the Tx arbitration state, the internal arbitration starts running immediately. When a transmit request or transmit abort request is made in the Rx matching state, the internal arbitration waits until the Rx matching state (i.e. intermission field) is finished, and then starts running as soon as the HCAN scheduler state becomes the Tx arbitration.

There are 5 sources that can run internal arbitration, which are:

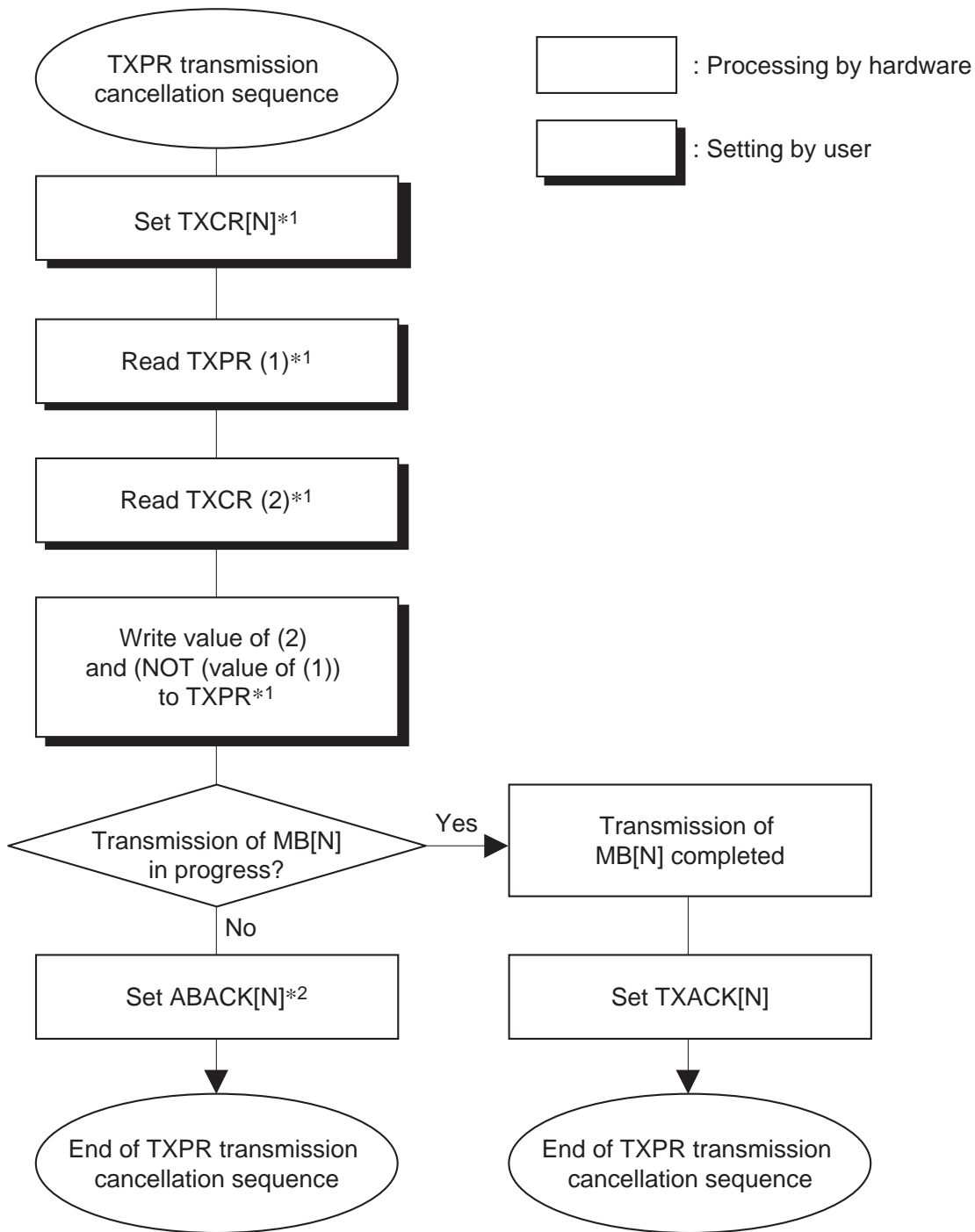
- TXPR is set
- TXCR is set (if TXCR is set for the message currently under transmission, the HCAN does not stop the transmission but completes. If the message loses the bus arbitration or causes an error on the bus, the HCAN will cancel the transmit request.)
- Error occurs on the CAN bus
- Message under transmission loses the arbitration on the CAN bus
- Mailbox with the setting MBC = 001 receives a remote frame

When these sources occur, the internal arbitration starts running to ensure that the highest priority message is always transmitted first. The followings are examples set in figure 16.9.

- 1-1: When a TXPR bit is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.
- 2-1, 2-2: During this period (Tx-arbitration for frame-2), when any of the above 5 sources occurs, the internal arbitration starts running and the next frame (Frame-2) to be transmitted is scheduled.
- 3-1, 3-2: During this period (Rx matching), any internal arbitration is not allowed to run, but scheduling is performed at the SOF of the next frame (Frame-2). If the transmit-requested message has the highest priority, the transmission will be set for the Frame-3.
- 3-3, 3-4: This is the same case as 2-1, 2-2.

#### **16.7.4 Message Transmission Cancellation Sequence**

Figure 16.10 shows the sequence for canceling a message transmit request set by TXPR.

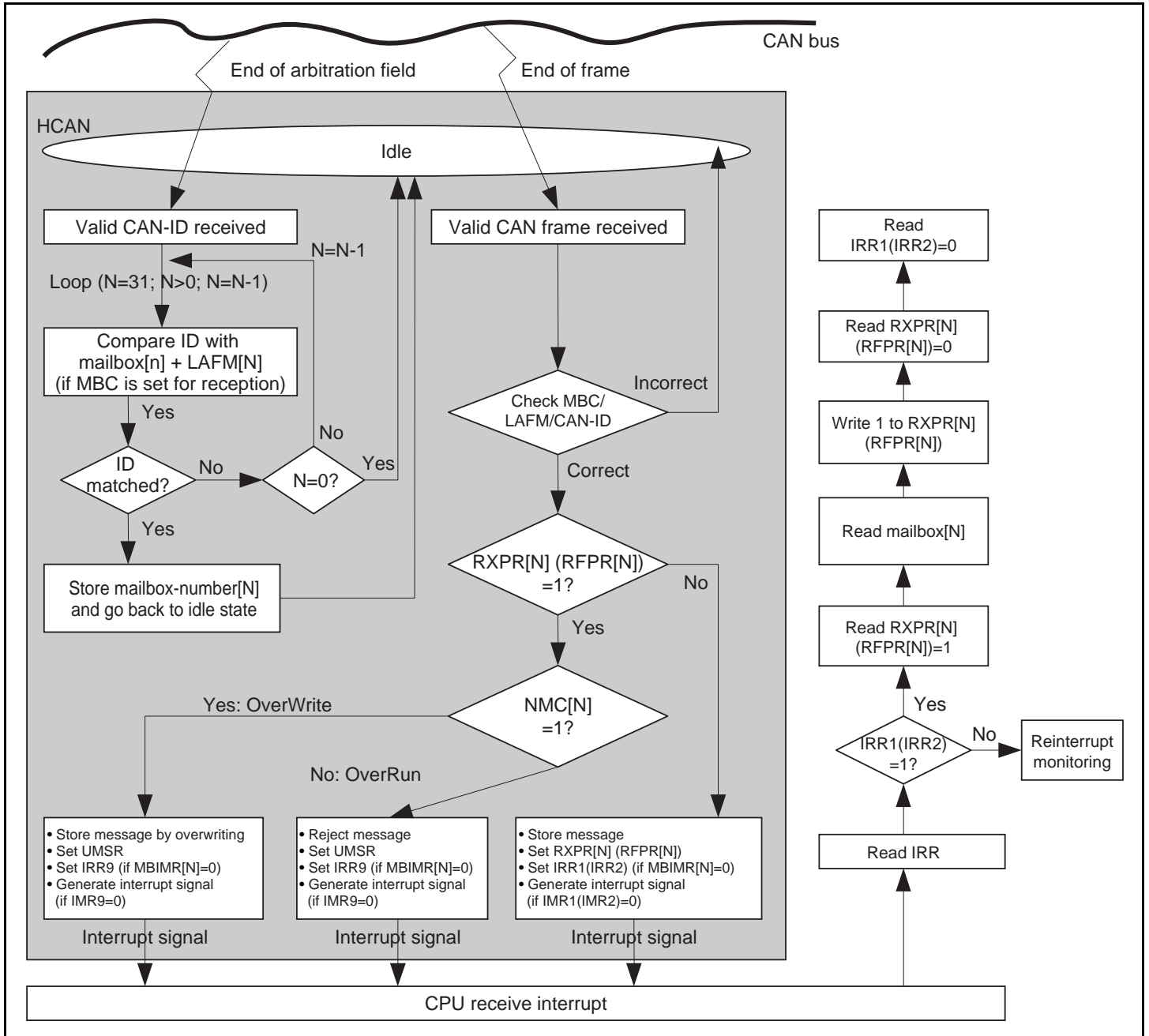


- Notes: 1. This setting must be made when transmission is canceled regardless of whether the mailbox is transmitting the message or no message is being transmitted.
2. For a message being transmitted, canceling operation of this transmission near EOF may also set ABACK in some case, though TXACK is normally set. (Flag invalid)  
In this case, clear ABACK.

**Figure 16.10 Transmission Cancellation Sequence**

## 16.7.5 Message Receive Sequence

Figure 16.11 shows the message receive sequence.



**Figure 16.11 Message Receive Sequence**

When the HCAN recognizes the end of the arbitration field during receiving of a message, it starts comparing the received ID to the IDs 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 to finally compare them to the received ID. If it does not match, the same check takes place at mailbox 30 (if configured as a receive box). Once the HCAN finds a matching ID, it stores the number of mailbox  $n$  into an internal buffer, stops the search, and goes back to the idle state, waiting for the end of frame (EOF) to come. When an EOF is notified by the CAN interface logic, the HCAN reads the MBC, LAFM, and CAN-ID of mailbox  $n$  to confirm the

matching condition again (i.e., there has been no modification to the configuration of mailbox n). This re-confirmation guarantees the data consistency even when a mailbox is reconfigured during receiving a message. If it still matches, then the message is written to or abandoned, depending on the setting of the NMC bit. If it is written to 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 ID of a received message matches the 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 IDs and LAFMs need to be carefully made.

### 16.7.6 Reconfiguration of Mailboxes

When reconfiguration of mailboxes is required, the following procedures should be taken.

**Change ID of Transmit Box or Change Transmit Box to Receive Box:** Confirm that the corresponding TXPR is not set. The ID or corresponding MBC bit can be changed at any time. When both need to be changed, change the ID first and then change the corresponding MBC bit.

#### **Change ID of Receive Box or Change Receive Box to Transmit Box:**

Method-1: Using Halt Mode

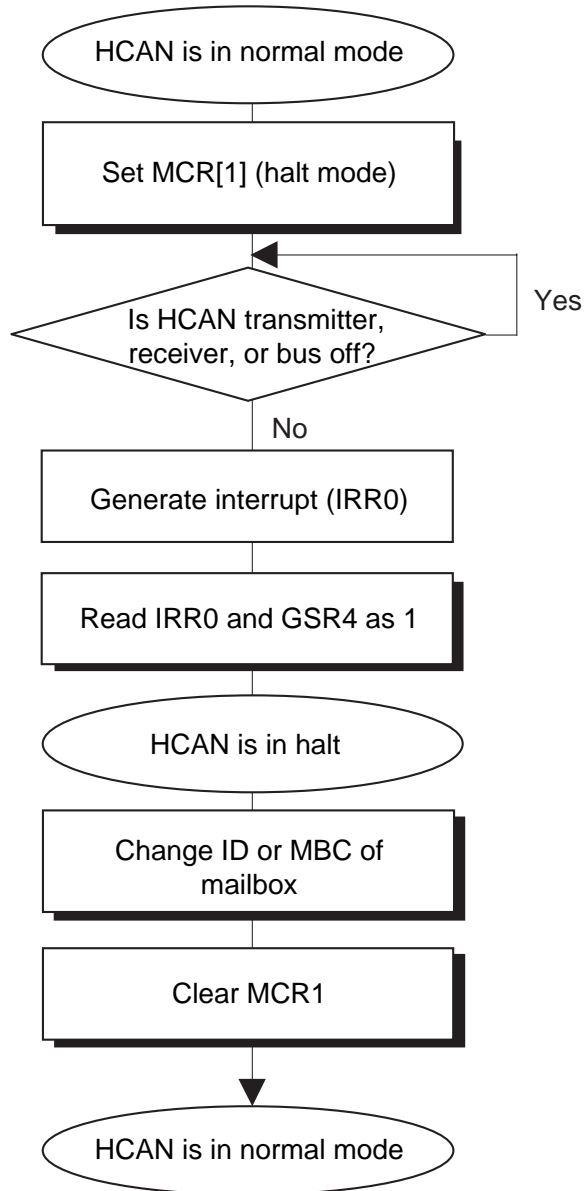
The advantage of this method is that the HCAN will not lose a message even if the message is on the CAN bus and the HCAN is a receiver. The HCAN-II will be in halt mode after completing the reception. The disadvantage is that it might take long if the HCAN is receiving a message (as the transition to halt mode is delayed until the end of the reception), and also the HCAN will not be able to receive/transmit messages during halt mode.

Method-2: Without Using Halt Mode

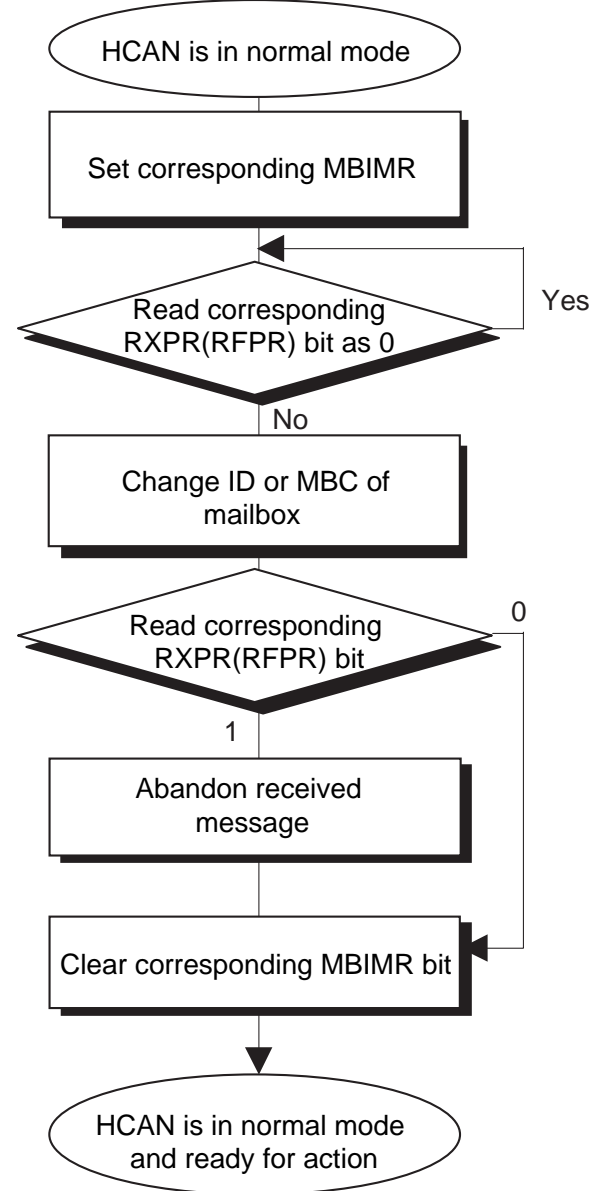
The advantage of this method is that the reconfiguration is done immediately, and the software overhead will be less as there is no interrupt. RXPR needs to be read before and after the reconfiguration. This is because to check if a message is received or not during this period. Note that MBIMR does not prevent the IRR1 from being set but simply prevents the interrupt signal from being generated. If a message is received, it is unknown if the received message is for the previous ID or for the new ID. Therefore, if a message is received during this period, it is better to abandon this message, and this is the disadvantage of this method.

## Reconfiguration of Mailboxes

### Method-1 (Using Halt Mode)



### Method-2 (Without Using Halt Mode)



: Processing by hardware

: Setting by user

**Figure 16.12 Change ID of Receive Box or Change Receive Box to Transmit Box**

## 16.7.7 List of Registers

**Table 16.9 List of Registers**

<b>Symbol</b>	<b>Register Name</b>	<b>Description</b>
MCR	Master control register	General configurations for HCAN and test mode setting
GSR	General status register	Status register for HCAN
HCAN_BCR0/1	Bit configuration register	Timing configurations for baud rate setting
IRR	Interrupt register	Interrupt request status
IMR	Interrupt mask register	Mask for interrupt request
TXPR0/1	Transmit pending request register	Transmission request
TXCR0/1	Transmit cancel register	Abort transmission request
TXACK0/1	Transmission acknowledge register	Transmission successful flag
ABACK0/1	Abort acknowledge register	Transmission abort flag
RXPR0/1	Data frame receive pending register	Data frame receive flag
RFPR0/1	Remote frame receive pending register	Remote frame receive flag
MBIMR0/1	Mailbox interrupt mask register	Mask for mailbox related interrupt
UMSR0/1	Unread message status register	Overwrite message flag
TCNTR	Timer counter register	Current timer value
TCR	Timer control register	General timer setting
TSR	Timer status register	Status flag for timer
TMR	Timer mode register	Value to be used for timestamp and TCMR
TDCR	Timer drift correction register	Timer adjustment for synchronization with network
LOSR	Local offset register	Offset for timer
CCR	Cycle counter register	Current cycle counter value for time triggered transmission
CMAX	Cycle maximum register	Number of basic cycles
ICR0/1	Input capture register	Input capture value
TCMR0-2	Timer compare match register	Compare value for timer
MB	Mailbox	Mailbox setting

## 16.7.8 Interrupt Sources

Table 16.10 lists the HCAN-II interrupt sources. These sources can be masked using the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR). For details on the interrupt vector of each interrupt source, see section 7, Interrupt Controller (INTC).

**Table 16.10 Interrupt Sources**

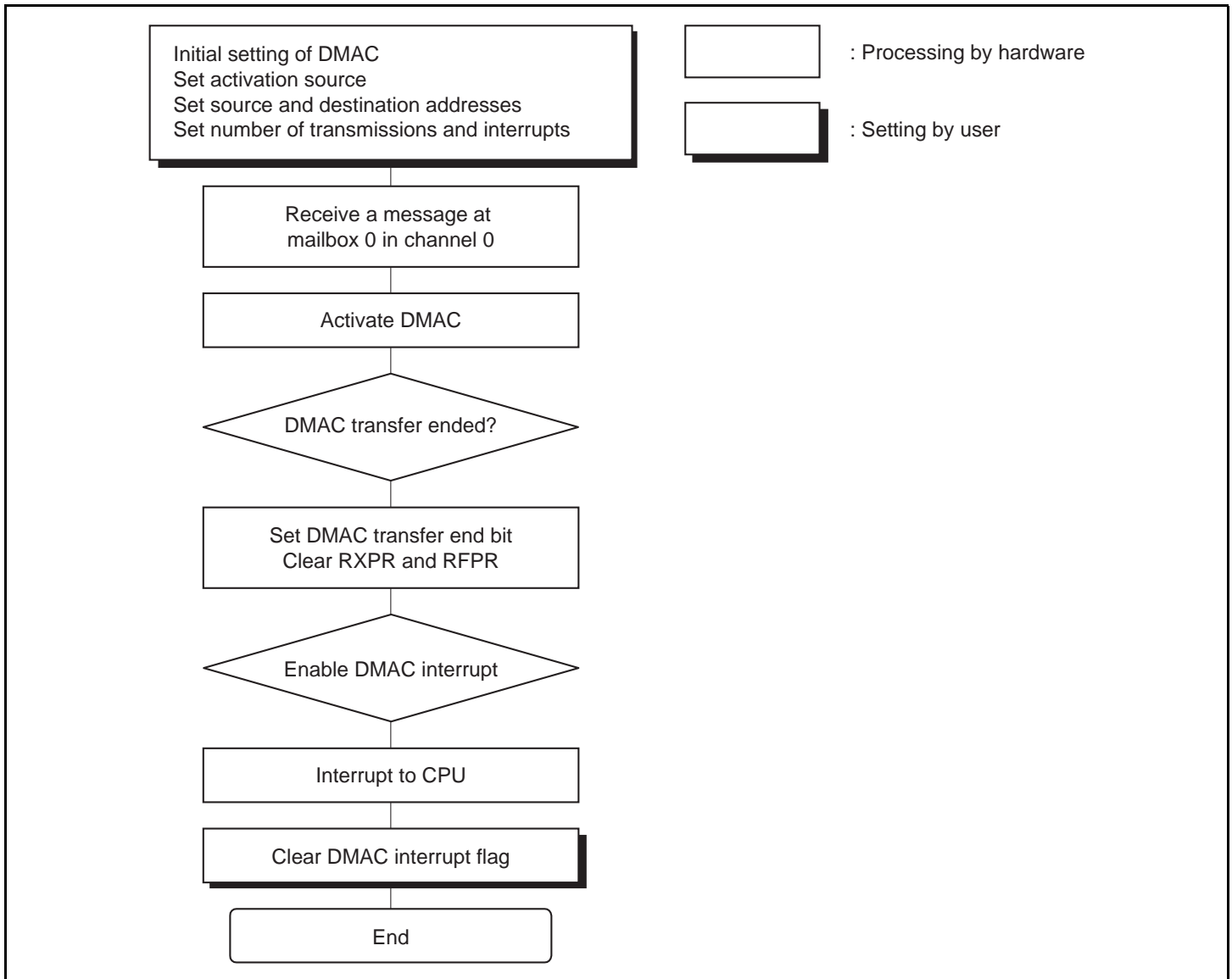
Interrupt Vector		Description	Interrupt Flag (IRR Bit)	DMAC Activation	
HCAN0	HCAN1			HCAN0	HCAN1
ERS0	ERS1	Error passive interrupt (TEC $\geq$ 128 or REC $\geq$ 128)	IRR5	Not possible	Not possible
		Bus off interrupt (TEC $\geq$ 256)/bus off recovery (receives 11 recessive bits 128 times)	IRR6		
		Error warning interrupt (TEC $\geq$ 96)	IRR3		
		Error warning interrupt (REC $\geq$ 96)	IRR4		
OVR0	OVR1	Reset processing interrupt by power-on reset	IRR0		
		Overload frame transmission	IRR7		
		Unread message overwrite/overrun	IRR9		
		Cycle counter overflow	IRR10		
		TCMR2 compare match	IRR11		
		Detection of CAN bus operation in HCAN-II sleep mode	IRR12		
		Timer overrun	IRR13		
		TCMR0 compare match	IRR14		
TCMR1 compare match	IRR15				
RM0	RM1	Data frame reception	IRR1	Possible*	
		Remote frame reception	IRR2		
SLE0	SLE1	Mailbox empty	IRR8	Not possible	

Note: \* Mailbox 0 only



## 16.7.9 DMAC Interface

The HCAN-II can activate the DMAC when a message is received at mailbox 0 in channel 0. When an interrupt occurs by mailbox 0 and the DMAC transfer ends after settings of the DMAC activation has been made, the RXPR0 and RFPR0 flags are cleared automatically. An interrupt request due to a receive interrupt from the HCAN-II cannot be sent to the CPU in this case. Figure 16.13 shows a DMAC transfer flowchart. For details on the settings of the DMAC activation, see section 10, Direct Memory Access Controller(DMAC).



**Figure 16.13 DMAC Transfer Flowchart**

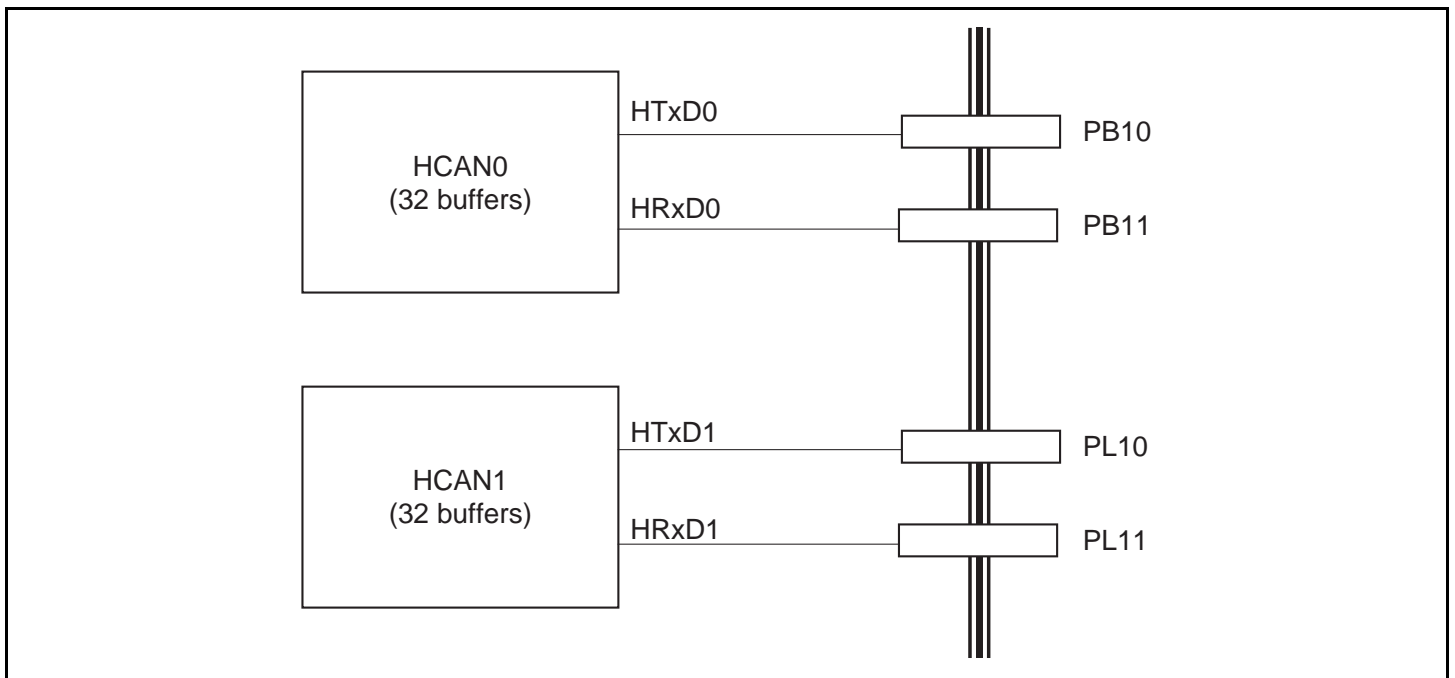
### 16.7.10 HCAN-II Port Settings

The HCAN-II port settings must be made in configuration mode or before entering the mode. For details on port settings, see section 21, Pin Function Controller(PFC). The SH7058 has the HCAN-II with two channels and there are two methods of using the HCAN-II.

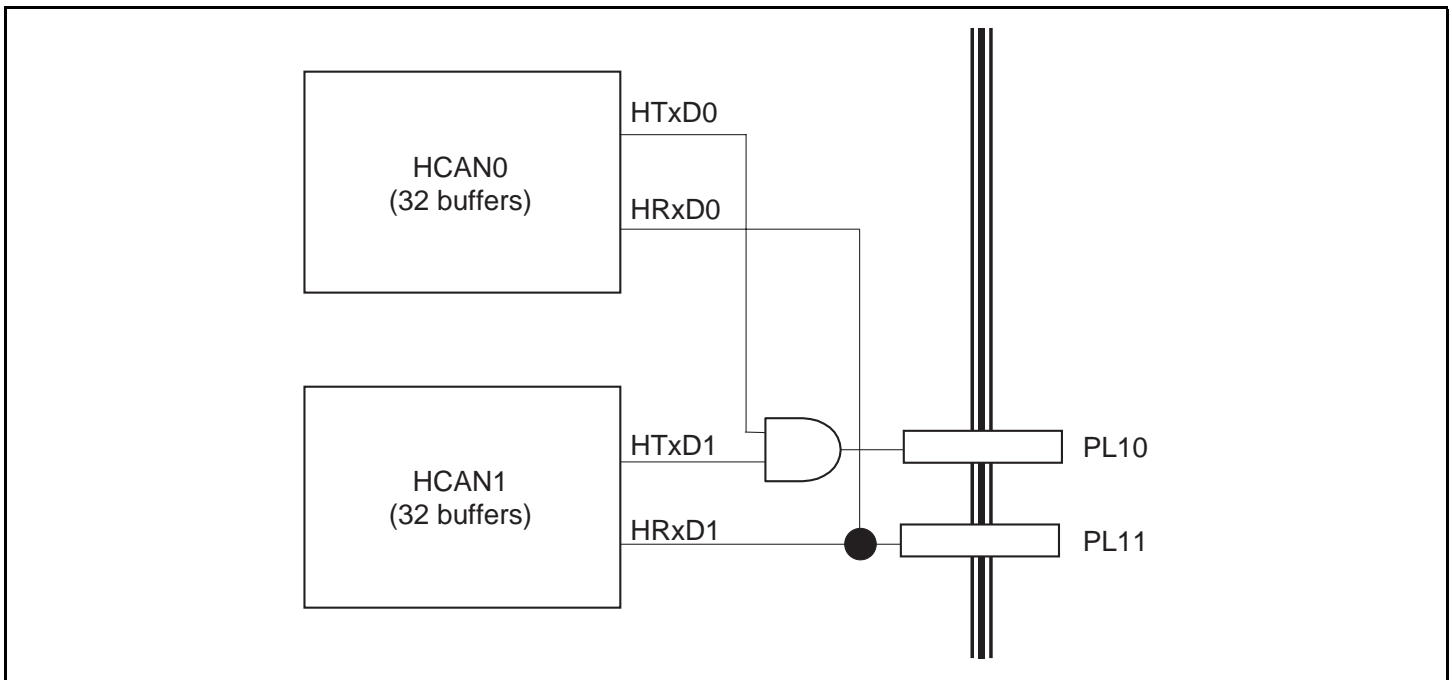
- 32-buffer HCAN-II with two channels
- 64-buffer HCAN-II with one channel\*

Note: \* When the HCAN-II is used as a 64-buffer with one channel, care is required. Be sure to carefully read section 16.8, Usage Notes.

Following figures show examples of the 32-buffer HCAN-II with two channels and 64-buffer HCAN-II with one channel.



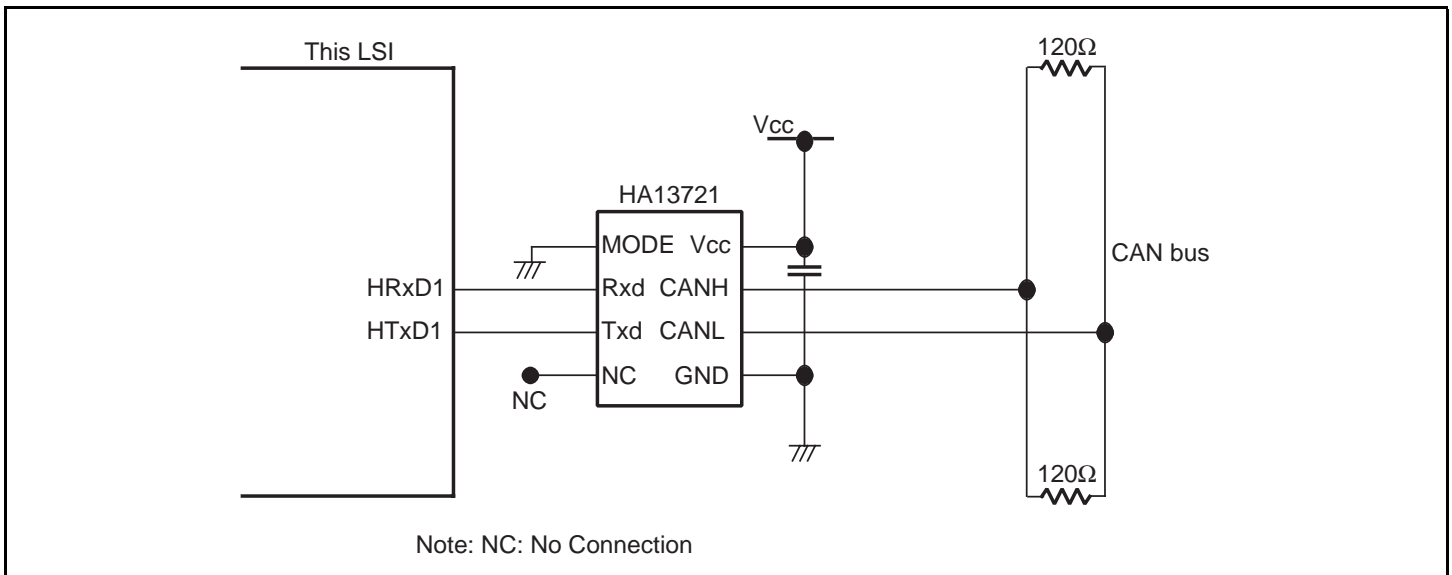
**Figure 16.14 32-Buffer HCAN-II with Two Channels**



**Figure 16.15 64-Buffer HCAN-II with One Channel**

### 16.7.11 CAN Bus Interface

A bus transceiver IC is necessary to connect this LSI to a CAN bus. A Renesas HA13721 transceiver IC and its compatible products are recommended. Figure 16.16 shows a sample connection diagram.



**Figure 16.16 High-Speed Interface Using HA13721**

## 16.8 Usage Notes

### 16.8.1 TXPR Setting during Reception

When the HCAN-II is used with the baud rate set to 1 Mbps and the transmission setting is made during message reception, there are following limitations on the number of transmit mailboxes (MB) and the number of accesses to mailboxes. Note that there is no limitation when 500 kbps of baud rate is used.

#### **Important: Limitations on setting TXPR during reception**

There are limitations on the number of mailboxes set by TXPR and the number of accesses to mailboxes.

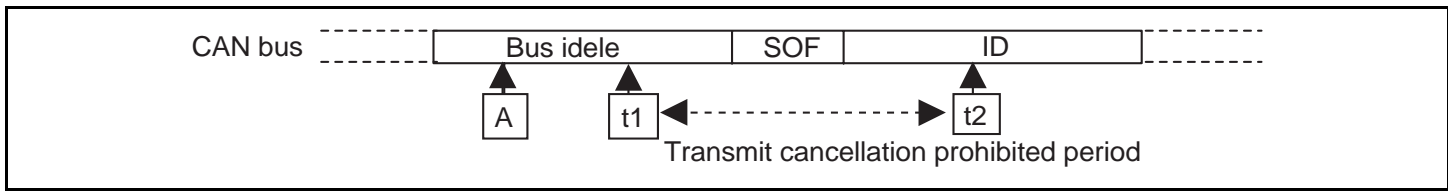
**Table 16.11 Limitations on Setting TXPR during Reception**

<b>P<math>\phi</math></b>	<b>Baud Rate</b>	<b>Number of Transmit MB to be Set Simultaneously</b>	<b>Upper-Limit Number of Accesses to MB in Words</b>
20 MHz	1.0 Mbps	25	36
		30	30
		31	29
	0.5 Mbps	31	No limitation
16 MHz	1.0 Mbps	10	34
		20	24
		25	18
		30	12
		31	11
	0.5 Mbps	31	No limitation

### 16.8.2 Transmit Cancellation Setting immediately after Transmission Setting in Bus Idle

When the transmission setting is made and then the transmit cancellation (TXCR) setting is made while the HCAN-II is in the bus idle state, there are following limitations.

#### **Important: Limitation on transmit cancellation setting immediately after transmission setting in bus idle**



When the transmission setting (TXPR) is made to a mailbox at the point A shown in the above figure and then transmit cancellation setting (TXCR) is made at the timing between t1 and t2, transmission may be performed to the CAN bus regardless of the fact that a flag is set in the abort acknowledge register. (The transmit acknowledge (TXACK) of the transmitted mailbox is set.)

The t1 and t2 timings are as follows after the transmission setting (TXPR) has been made.

**Table 16.12 Transmit Cancellation Prohibited Period**

Pφ	Baud Rate		t1	t2
20 MHz	1 Mbps	MB order	1.90 μs	6.30 μs
		ID order	5.05 μs	13.55 μs
20 MHz	0.5 Mbps	MB order	2.55 μs	7.65 μs
		ID order	5.45 μs	13.55 μs

### 16.8.3 Failure on Transmit Cancellation at Mailbox 31

When mailbox 31 is used as a transmit buffer and the transmit cancellation setting is made by TXCR, the following failures may occur. Note that these failures do not occur in the bus-off state.

- When the transmit cancellation setting is made by TXCR for mailbox 31 during message transmission (except for mailbox 31), a message may be transmitted and the transmit acknowledge register (TXACK) may be set regardless of the fact that the abort acknowledge register (ABACK) is set.
- When the transmit cancellation setting is made by TXCR for mailbox 31 during message transmission of mailbox 31, TXPR may not be cleared even if transmission is completed at mailbox 31 and retransmission may be performed according to the internal arbitration sequence.

### 16.8.4 TXPR Setting during Transmission

When the HCAN-II is used with the baud rate set to 1 Mbps and the TXPR setting is made during transmission, there are the following limitations on the number of transmit mailboxes (MB) and the number of accesses to mailboxes until transmission is completed. Note that there is no limitation when 500 kbps of baud rate is used.

## Important: Limitations on transmission setting during transmission

**Table 16.13 Limitations on Accesses during Transmission Setting**

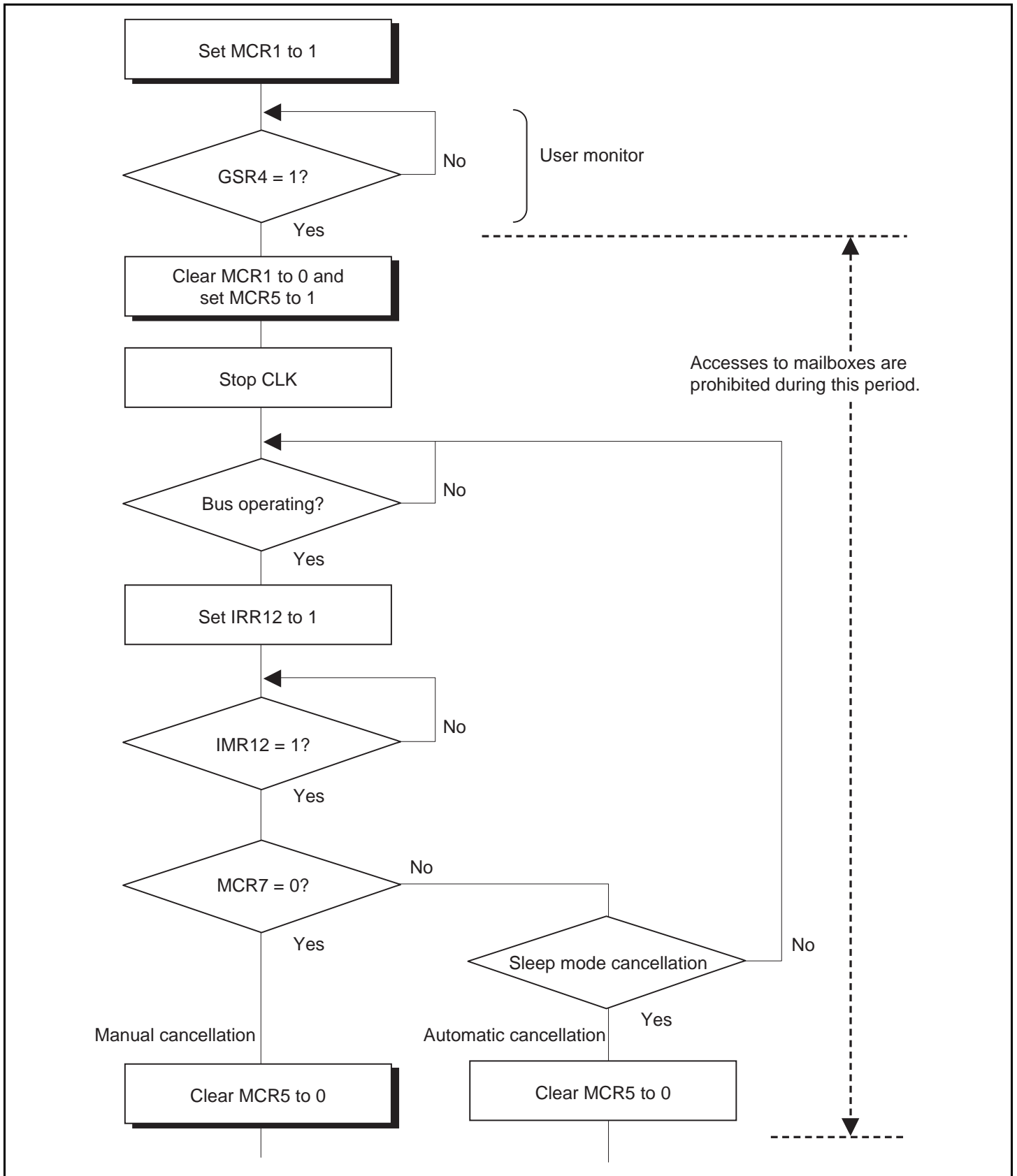
<b>Number of Transmit MB to be Set Simultaneously</b>	<b>Upper-Limit Number of Accesses to MB in Words</b>
1	36
2	34
3	34
4	32
5	32
6	30
7	30
8	28
9	28
10	26
11	26
12	24
13	24
14	22
15	22
16	22
17	22
18	20
19	20
20	20
21	18
22	18
23	16
24	16
25	14
26	12
27	12
28	10
29	8
30	8

### **16.8.5 Time Triggered Transmission Setting/Timer Operation Disabled**

- The TTE (time trigger enable) bit for setting mailboxes must be written to 0. A failure may occur during event triggered transmission.
- The timer must not be operated during event triggered transmission (TCR15 bit = 0). A failure may occur during event triggered transmission.

### **16.8.6 Mailbox Access in HCAN Sleep Mode**

Do not access a mailbox in HCAN sleep mode. When a mailbox is accessed in HCAN sleep mode, CPU operation may be halted. CPU operation is not halted when a register is accessed in HCAN sleep mode. Accessing a mailbox does not halt CPU operation except for in HCAN sleep mode.

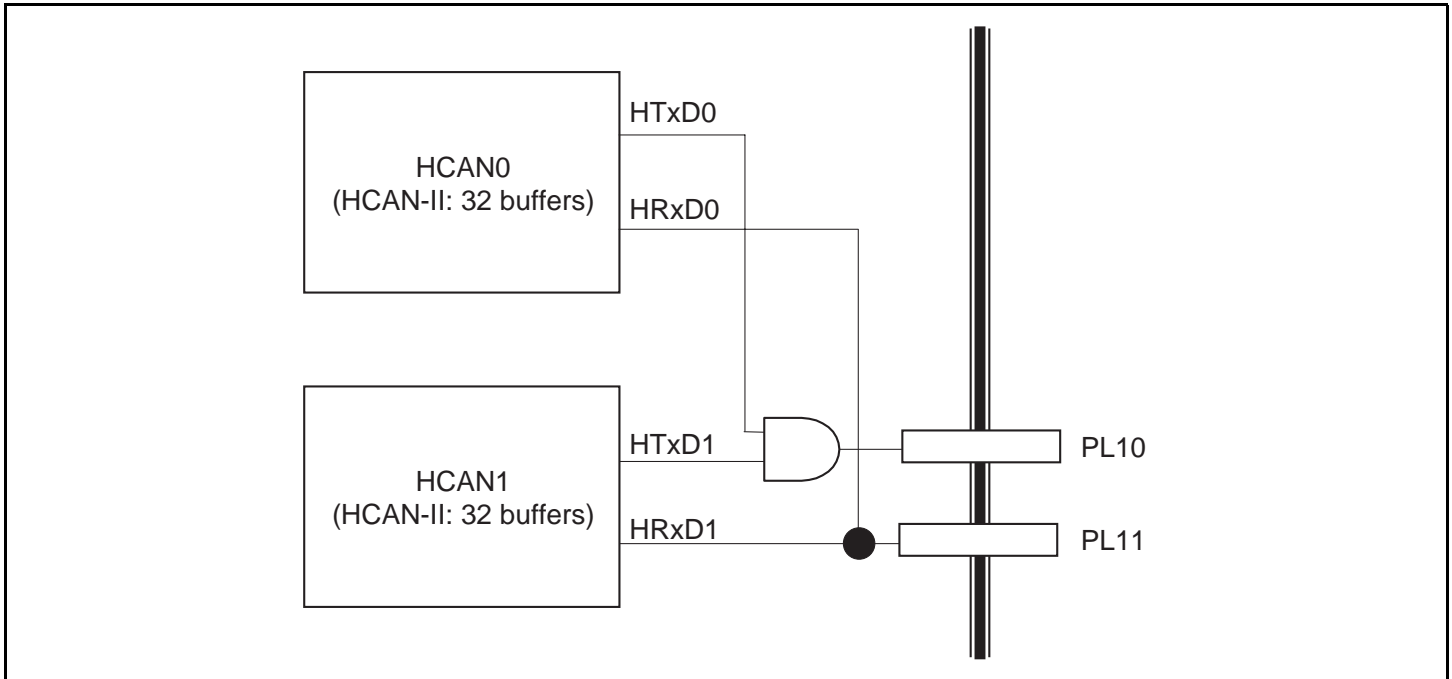


**Figure 16.17 HCAN Sleep Mode Flowchart**



## 16.8.7 Notes on Port Settings for 64-Buffer HCAN-II with One Channel

The SH7058 has the HCAN-II with two channels. When using the HCAN-II as a 64-buffer with one channel, the following notice should be taken at port settings.



1. When a message is transmitted to the CAN bus without connecting to other nodes, an ACK error will not occur. For example, when a message is transmitted from HCAN0 in the above figure, HCAN1 transmits ACK in the ACK field.  
HCAN1 which already received the message on the CAN bus transmits ACK in the ACK field according to the CAN protocol and HCAN0 receives the ACK.  
For a countermeasure, please set the channel that will not transmit the message to the reset state ( $MCR0 = 1$ ). Accordingly, a channel that will not transmit the message does not transmit ACK.
2. Internal arbitration which determines the transmission order is independently carried out by HCAN0 and HCAN1, respectively. The HCAN-II has 31 transmission buffers per channel. However, internal arbitration cannot be carried out in the range of the 62 transmission buffers.
3. Please do not set the same transmit message ID to HCAN0 and HCAN1.  
Otherwise, the same message will be transmitted from the two channels after arbitration on the CAN bus.

# Section 17 A/D Converter

## 17.1 Overview

The SH7058 includes a 10-bit successive-approximation A/D converter, with software selection of up to 32 analog input channels.

The A/D converter is composed of three independent modules, A/D0, A/D1, and A/D2. A/D0 and A/D1 each comprise three groups, while A/D2 comprises two groups.

Module	Analog Groups	Channels
A/D0	Analog group 0	AN0–AN3
	Analog group 1	AN4–AN7
	Analog group 2	AN8–AN11
A/D1	Analog group 3	AN12–AN15
	Analog group 4	AN16–AN19
	Analog group 5	AN20–AN23
A/D2	Analog group 6	AN24–AN27
	Analog group 7	AN28–AN31

### 17.1.1 Features

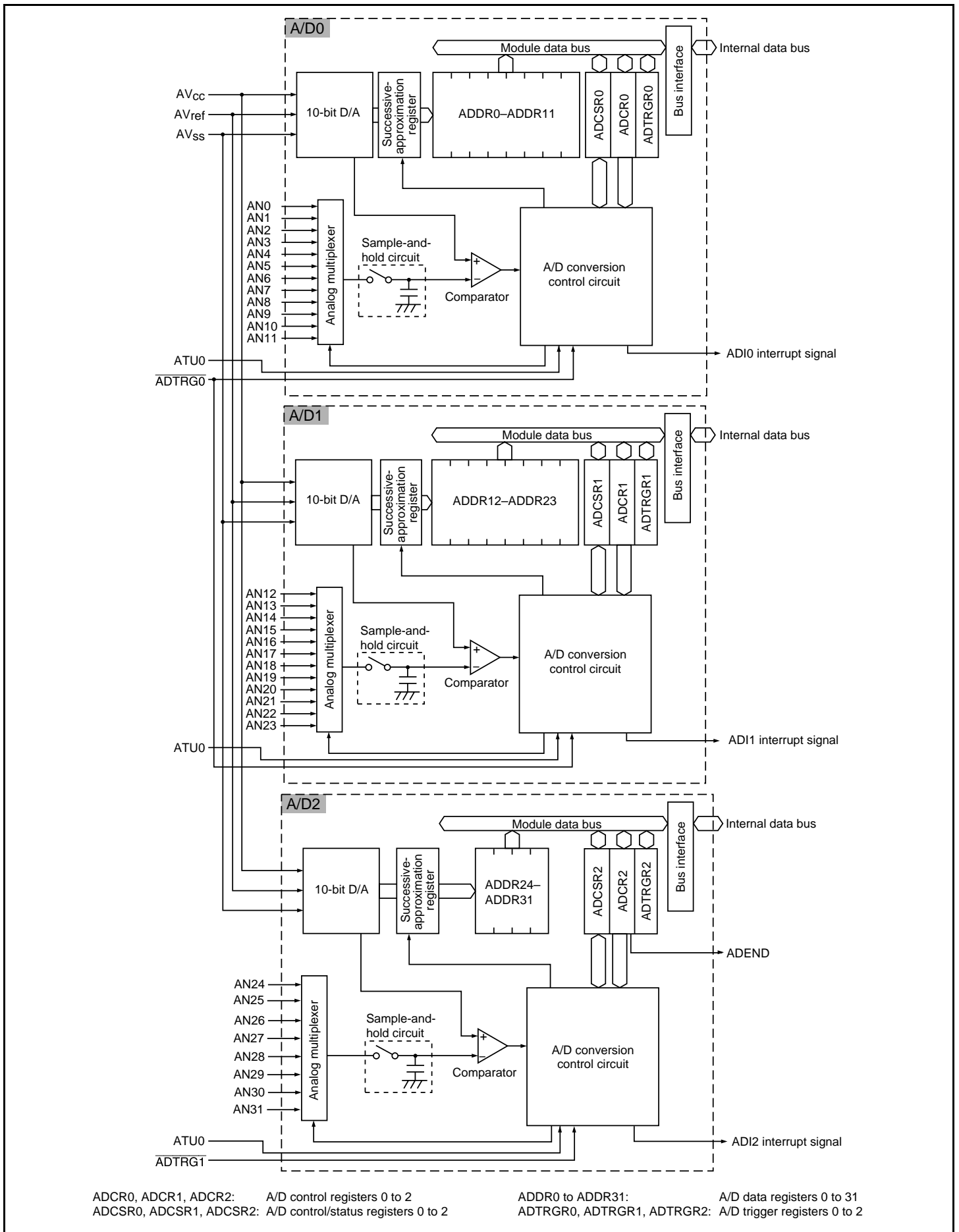
The features of the A/D converter are summarized below.

- 10-bit resolution  
32 input channels (A/D0: 12 channels, A/D1: 12 channels, A/D2: 8 channels)
- High-speed conversion  
Conversion time: minimum 13.3  $\mu$ s per channel (when peripheral clock (P $\phi$ ) = 20 MHz)
- Two conversion modes
  - Single mode: A/D conversion on one channel
  - Scan mode: continuous scan mode, single-cycle scan mode (AN0–AN3, AN4–AN7, AN8–AN11, AN12–AN15, AN16–AN19, AN20–AN23, AN24–AN27, AN28–AN31)
    - Continuous conversion on 1 to 12 channels (A/D0)
    - Continuous conversion on 1 to 12 channels (A/D1)
    - Continuous conversion on 1 to 8 channels (A/D2)
- Thirty-two 10-bit A/D data registers  
A/D conversion results are transferred for storage into data registers corresponding to the channels.

- Three sample-and-hold circuits  
A sample-and-hold circuit is built into each A/D converter module (AD/0, AD/1, and AD/2), simplifying the configuration of external analog input circuitry.
- A/D conversion interrupts and DMA function supported  
An A/D conversion interrupt request (ADI) can be sent to the CPU at the end of A/D conversion (ADI0: A/D0 interrupt request; ADI1: A/D1 interrupt request; ADI2: A/D2 interrupt request). Also, the DMAC can be activated by an ADI interrupt request.
- Two kinds of conversion activation
  - Software or external trigger ( $\overline{\text{ADTER0}}$ , ATU-II (ITVRR2A)) can be selected (A/D0)
  - Software or external trigger ( $\overline{\text{ADTGR0}}$ , ATU-II (ITVRR2B)) can be selected (A/D1)
  - Software or external trigger ( $\overline{\text{ADTGR1}}$ , ATU-II (ITVRR1)) can be selected (A/D2)
- ADEND output  
Conversion timing can be monitored with the ADEND output pin when using channel 31 in scan mode.

### 17.1.2 Block Diagram

Figure 17.1 shows a block diagram of the A/D converter.



**Figure 17.1 A/D Converter Block Diagram**

### 17.1.3 Pin Configuration

Table 17.1 summarizes the A/D converter's input pins. There are 32 analog input pins, AN0 to AN31. The 12 pins AN0 to AN11 are A/D0 analog inputs, divided into three groups: AN0 to AN3 (group 0), AN4 to AN7 (group 1), and AN8 to AN11 (group 2). The 12 pins AN12 to AN23 are A/D1 analog inputs, divided into three groups: AN12 to AN15 (group 3), AN16 to AN19 (group 4), and AN20 to AN23 (group 5). The 8 pins AN24 to AN31 are A/D2 analog inputs, divided into two groups: AN24 to AN27 (group 6), and AN28 to AN31 (group 7).

The  $\overline{\text{ADTRG0}}$  and  $\overline{\text{ADTRG1}}$  pins are used to provide A/D conversion start timing from off-chip. When the low level of a pulse is applied to one of these pins, A/D0, A/D1, or A/D2 starts conversion.

The ADEND pin is an output used to monitor conversion timing when channel 31 is used in scan mode.

The  $\text{AV}_{\text{CC}}$  and  $\text{AV}_{\text{SS}}$  pins are power supply voltage pins for the analog section in A/D converter modules A/D0 to A/D2. The  $\text{AV}_{\text{ref}}$  pin is the A/D converter module A/D0 to A/D2 reference voltage pin.

To maintain chip reliability, ensure that  $\text{AV}_{\text{CC}} = 5 \text{ V} \pm 0.5 \text{ V}$  and  $\text{AV}_{\text{SS}} = \text{V}_{\text{SS}}$  during normal operation, and never leave the  $\text{AV}_{\text{CC}}$  and  $\text{AV}_{\text{SS}}$  pins open, even when the A/D converter is not being used.

The voltage applied to the analog input pins should be in the range  $\text{AV}_{\text{SS}} \leq \text{ANn} \leq \text{AV}_{\text{ref}}$ .

**Table 17.1 A/D Converter Pins**

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	$AV_{CC}$	Input	A/D0–A/D2 analog section power supply
Analog ground pin	$AV_{SS}$	Input	A/D0–A/D2 analog section ground and reference voltage
Analog reference power supply pin	$AV_{ref}$	Input	A/D0–A/D2 analog section reference voltage
Analog input pin 0	AN0	Input	A/D0 analog inputs 0 to 3 (analog group 0)
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	A/D0 analog inputs 4 to 7 (analog group 1)
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
Analog input pin 12	AN12	Input	A/D1 analog inputs 12 to 15 (analog group 3)
Analog input pin 13	AN13	Input	
Analog input pin 14	AN14	Input	
Analog input pin 15	AN15	Input	
Analog input pin 16	AN16	Input	
Analog input pin 17	AN17	Input	
Analog input pin 18	AN18	Input	A/D1 analog inputs 16 to 19 (analog group 4)
Analog input pin 19	AN19	Input	
Analog input pin 20	AN20	Input	
Analog input pin 21	AN21	Input	
Analog input pin 22	AN22	Input	
Analog input pin 23	AN23	Input	

**Table 17.1 A/D Converter Pins (cont)**

<b>Pin Name</b>	<b>Abbreviation</b>	<b>I/O</b>	<b>Function</b>
Analog input pin 24	AN24	Input	A/D2 analog inputs 24 to 27 (analog group 6)
Analog input pin 25	AN25	Input	
Analog input pin 26	AN26	Input	
Analog input pin 27	AN27	Input	
Analog input pin 28	AN28	Input	A/D2 analog inputs 28 to 31 (analog group 7)
Analog input pin 29	AN29	Input	
Analog input pin 30	AN30	Input	
Analog input pin 31	AN31	Input	
A/D conversion trigger input pin 0	$\overline{\text{ADTRG0}}$	Input	A/D0 and A/D1 A/D conversion trigger input
A/D conversion trigger input pin 1	$\overline{\text{ADTRG1}}$	Input	A/D2 A/D conversion trigger input
ADEND output pin	ADEND	Output	A/D2 channel 31 conversion timing monitor output

## 17.1.4 Register Configuration

Table 17.2 summarizes the A/D converter's registers.

**Table 17.2 A/D Converter Registers**

Name	Abbreviation	R/W	Initial Value	Address	Access Size* <sup>1</sup>
A/D data register 0 (H/L)	ADDR0 (H/L)	R	H'0000	H'FFFFFF800	8, 16
A/D data register 1 (H/L)	ADDR1 (H/L)	R	H'0000	H'FFFFFF802	8, 16
A/D data register 2 (H/L)	ADDR2 (H/L)	R	H'0000	H'FFFFFF804	8, 16
A/D data register 3 (H/L)	ADDR3 (H/L)	R	H'0000	H'FFFFFF806	8, 16
A/D data register 4 (H/L)	ADDR4 (H/L)	R	H'0000	H'FFFFFF808	8, 16
A/D data register 5 (H/L)	ADDR5 (H/L)	R	H'0000	H'FFFFFF80A	8, 16
A/D data register 6 (H/L)	ADDR6 (H/L)	R	H'0000	H'FFFFFF80C	8, 16
A/D data register 7 (H/L)	ADDR7 (H/L)	R	H'0000	H'FFFFFF80E	8, 16
A/D data register 8 (H/L)	ADDR8 (H/L)	R	H'0000	H'FFFFFF810	8, 16
A/D data register 9 (H/L)	ADDR9 (H/L)	R	H'0000	H'FFFFFF812	8, 16
A/D data register 10 (H/L)	ADDR10 (H/L)	R	H'0000	H'FFFFFF814	8, 16
A/D data register 11 (H/L)	ADDR11 (H/L)	R	H'0000	H'FFFFFF816	8, 16
A/D data register 12 (H/L)	ADDR12 (H/L)	R	H'0000	H'FFFFFF820	8, 16
A/D data register 13 (H/L)	ADDR13 (H/L)	R	H'0000	H'FFFFFF822	8, 16
A/D data register 14 (H/L)	ADDR14 (H/L)	R	H'0000	H'FFFFFF824	8, 16
A/D data register 15 (H/L)	ADDR15 (H/L)	R	H'0000	H'FFFFFF826	8, 16
A/D data register 16 (H/L)	ADDR16 (H/L)	R	H'0000	H'FFFFFF828	8, 16
A/D data register 17 (H/L)	ADDR17 (H/L)	R	H'0000	H'FFFFFF82A	8, 16
A/D data register 18 (H/L)	ADDR18 (H/L)	R	H'0000	H'FFFFFF82C	8, 16
A/D data register 19 (H/L)	ADDR19 (H/L)	R	H'0000	H'FFFFFF82E	8, 16
A/D data register 20 (H/L)	ADDR20 (H/L)	R	H'0000	H'FFFFFF830	8, 16
A/D data register 21 (H/L)	ADDR21 (H/L)	R	H'0000	H'FFFFFF832	8, 16
A/D data register 22 (H/L)	ADDR22 (H/L)	R	H'0000	H'FFFFFF834	8, 16
A/D data register 23 (H/L)	ADDR23 (H/L)	R	H'0000	H'FFFFFF836	8, 16
A/D data register 24 (H/L)	ADDR24 (H/L)	R	H'0000	H'FFFFFF840	8, 16
A/D data register 25 (H/L)	ADDR25 (H/L)	R	H'0000	H'FFFFFF842	8, 16
A/D data register 26 (H/L)	ADDR26 (H/L)	R	H'0000	H'FFFFFF844	8, 16



**Table 17.2 A/D Converter Registers (cont)**

<b>Name</b>	<b>Abbreviation</b>	<b>R/W</b>	<b>Initial Value</b>	<b>Address</b>	<b>Access Size*<sup>1</sup></b>
A/D data register 27 (H/L)	ADDR27 (H/L)	R	H'0000	H'FFFFFF846	8, 16
A/D data register 28 (H/L)	ADDR28 (H/L)	R	H'0000	H'FFFFFF848	8, 16
A/D data register 29 (H/L)	ADDR29 (H/L)	R	H'0000	H'FFFFFF84A	8, 16
A/D data register 30 (H/L)	ADDR30 (H/L)	R	H'0000	H'FFFFFF84C	8, 16
A/D data register 31 (H/L)	ADDR31 (H/L)	R	H'0000	H'FFFFFF84E	8, 16
A/D control/status register 0	ADCSR0	R/(W)* <sup>2</sup>	H'00	H'FFFFFF818	8, 16
A/D control register 0	ADCR0	R/W	H'0F	H'FFFFFF819	8, 16
A/D trigger register 0	ADTRGR0	R/W	H'FF	H'FFFFFF76E	8
A/D control/status register 1	ADCSR1	R/(W)* <sup>2</sup>	H'00	H'FFFFFF838	8, 16
A/D control register 1	ADCR1	R/W	H'0F	H'FFFFFF839	8, 16
A/D trigger register 1	ADTRGR1	R/W	H'FF	H'FFFFFF72E	8
A/D control/status register 2	ADCSR2	R/(W)* <sup>2</sup>	H'08	H'FFFFFF858	8, 16
A/D control register 2	ADCR2	R/W	H'0F	H'FFFFFF859	8, 16
A/D trigger register 2	ADTRGR2	R/W	H'FF	H'FFFFFF72F	8

Notes: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

1. A 16-bit access must be made on a word boundary.
2. Only 0 can be written to bit 7 to clear the flag.

## 17.2 Register Descriptions

### 17.2.1 A/D Data Registers 0 to 31 (ADDR0 to ADDR31)

A/D data registers 0 to 31 (ADDR0 to ADDR31) are 16-bit read-only registers that store the results of A/D conversion. There are 32 registers, corresponding to analog inputs 0 to 31 (AN0 to AN31).

The ADDR registers are initialized to H'0000 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
ADDRnH (upper byte)	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
ADDRnL (lower byte)	AD1	AD0	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

(n = 0 to 31)

The A/D converter converts analog input to a 10-bit digital value. The upper 8 bits of this data are stored in the upper byte of the ADDR corresponding to the selected channel, and the lower 2 bits in the lower byte of that ADDR. Only the most significant 2 bits of the ADDR lower byte data are valid.

Table 17.3 shows correspondence between the analog input channels and A/D data registers.

**Table 17.3 Analog Input Channels and A/D Data Registers**

Analog Input Channel	A/D Data Register	Analog Input Channel	A/D Data Register	Analog Input Channel	A/D Data Register	Analog Input Channel	A/D Data Register
AN0	ADDR0	AN8	ADDR8	AN16	ADDR16	AN24	ADDR24
AN1	ADDR1	AN9	ADDR9	AN17	ADDR17	AN25	ADDR25
AN2	ADDR2	AN10	ADDR10	AN18	ADDR18	AN26	ADDR26
AN3	ADDR3	AN11	ADDR11	AN19	ADDR19	AN27	ADDR27
AN4	ADDR4	AN12	ADDR12	AN20	ADDR20	AN28	ADDR28
AN5	ADDR5	AN13	ADDR13	AN21	ADDR21	AN29	ADDR29
AN6	ADDR6	AN14	ADDR14	AN22	ADDR22	AN30	ADDR30
AN7	ADDR7	AN15	ADDR15	AN23	ADDR23	AN31	ADDR31

**17.2.2 A/D Control/Status Registers 0 and 1 (ADCSR0, ADCSR1)**

A/D control/status registers 0 and 1 (ADCSR0, ADCSR1) are 8-bit readable/writable registers whose functions include selection of the A/D conversion mode for A/D0 and A/D1.

ADCSR0 and ADCSR1 are initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	ADF	ADIE	ADM1	ADM0	CH3	CH2	CH1	CH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: \* Only 0 can be written to clear the flag.

- Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

#### Bit 7:

ADF	Description
0	Indicates that A/D0 or A/D1 is performing A/D conversion, or is in the idle state (Initial value)  [Clearing conditions] <ul style="list-style-type: none"> <li>• When ADF is read while set to 1, then 0 is written to ADF</li> <li>• When the DMAC is activated by ADI0 or ADI1</li> </ul>
1	Indicates that A/D0 or A/D1 has finished A/D conversion, and the digital value has been transferred to ADDR  [Setting conditions] <ul style="list-style-type: none"> <li>• Single mode: When A/D conversion ends</li> <li>• Scan mode: When all set A/D conversions end</li> </ul>

The operation of the A/D converter after ADF is set to 1 differs between single mode and scan mode.

In single mode, after the A/D converter transfers the digit value to ADDR, ADF is set to 1 and the A/D converter enters the idle state. In scan mode, ADF is set to 1 after all the set conversions end. For example, in the case of 12-channel scanning, ADF is set to 1 immediately after the end of conversion for AN8 to AN11 (group 2) or AN20 to AN23 (group 5). After ADF is set to 1, conversion continues in the case of continuous scanning, and ends in the case of single-cycle scanning.

Note that 1 cannot be written to ADF.

- Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the A/D interrupt (ADI).

To prevent incorrect operation, ensure that the ADST bit in A/D control registers 0 and 1 (ADCR0, ADCR1) is cleared to 0 before switching the operating mode.

#### Bit 6:

ADIE	Description
0	A/D interrupt (ADI0, ADI1) is disabled (Initial value)
1	A/D interrupt (ADI0, ADI1) is enabled

When A/D conversion ends and the ADF bit is set to 1, an A/D0 or A/D1 A/D interrupt (ADI0, ADI1) will be generated if the ADIE bit is 1. ADI0 and ADI1 are cleared by clearing ADF or ADIE to 0.

- Bits 5 and 4: A/D Mode 1 and 0 (ADM1, ADM0): These bits select the A/D conversion mode from single mode, 4-channel scan mode, 8-channel scan mode, and 12-channel scan mode.

To prevent incorrect operation, ensure that the ADST bit in A/D control registers 1 and 0 (ADCR1, ADCR0) is cleared to 0 before switching the operating mode.

Bit 5: ADM1	Bit 4: ADM0	Description	
0	0	Single mode	(Initial value)
	1	4-channel scan mode (analog groups 0, 1, 2, 3, 4, 5)	
1	0	8-channel scan mode (analog groups 0, 1, 3, 4)	
	1	12-channel scan mode (analog groups 0, 1, 2, 3, 4, 5)	

When ADM1 and ADM0 are set to 00, single mode is set. In single mode, operation ends after A/D conversion has been performed once on the analog channels selected with bits CH3 to CH0 in ADCSR.

When ADM1 and ADM0 are set to 01, 4-channel scan mode is set. In scan mode, A/D conversion is performed continuously on a number of channels. The channels on which A/D conversion is to be performed in scan mode are set with bits CH3 to CH0 in ADCSR1 and ADCSR0. In 4-channel scan mode, conversion is performed continuously on the channels in one of analog groups 0 (AN0 to AN3), 1 (AN4 to AN7), 2 (AN8 to AN11), 3 (AN12 to AN15), 4 (AN16 to AN19), or 5 (AN20 to AN23).

When the ADCS bit is cleared to 0, selecting scanning of all channels within the group (AN0 to AN3, AN4 to AN7, AN8 to AN11, or AN12 to AN15, AN16 to AN19, AN20 to AN23), conversion is performed continuously, once only for each channel within the group, and operation stops on completion of conversion for the last (highest-numbered) channel.

When ADM1 and ADM0 are set to 10, 8-channel scan mode is set. In 8-channel scan mode, conversion is performed continuously on the 8 channels in analog groups 0 (AN0 to AN3) and 1 (AN4 to AN7) or analog groups 3 (AN12 to AN15) and 4 (AN16 to AN19). When the ADCS bit is cleared to 0, selecting scanning of all channels within the groups (AN0 to AN7 or AN12 to AN19), conversion is performed continuously, once only for each channel within the groups, and operation stops on completion of conversion for the last (highest-numbered) channel.

When ADM1 and ADM0 are set to 11, 12-channel scan mode is set. In 12-channel scan mode, conversion is performed continuously on the 12 channels in analog groups 0 (AN0 to AN3), 1 (AN4 to AN7), and 2 (AN8 to AN11) or analog groups 3 (AN12 to AN15), 4 (AN16 to AN19), and 5 (AN20 to AN23). When the ADCS bit is cleared to 0, selecting scanning of all channels within the groups (AN0 to AN11 or AN12 to AN19), conversion is performed continuously, once only for each channel within the groups, and operation stops on completion of conversion for the last (highest-numbered) channel.

For details of the operation in single mode and scan mode, see section 17.4, Operation.

- Bits 3 to 0—Channel Select 3 to 0 (CH3 to CH0): These bits, together with the ADM1 and ADM0 bits, select the analog input channels.

To prevent incorrect operation, ensure that the ADST bit in A/D control registers 1 and 0 (ADCR1, ADCR0) is cleared to 0 before changing the analog input channel selection.

### Analog Input Channels

Bit 3: CH3	Bit 2: CH2	Bit 1: CH1	Bit 0: CH0	Single Mode		4-Channel Scan Mode	
				A/D0	A/D1	A/D0	A/D1
0	0	0	0	AN0 (Initial value)	AN12 (Initial value)	AN0	AN12
			1	AN1	AN13	AN0, AN1	AN12, AN13
		1	0	AN2	AN14	AN0–AN2	AN12–AN14
			1	AN3	AN15	AN0–AN3	AN12–AN15
	1	0	0	AN4	AN16	AN4	AN16
			1	AN5	AN17	AN4, AN5	AN16, AN17
		1	0	AN6	AN18	AN4–AN6	AN16–AN18
			1	AN7	AN19	AN4–AN7	AN16–AN19
1	0*	0	0	AN8	AN20	AN8	AN20
			1	AN9	AN21	AN8, AN9	AN20, AN21
		1	0	AN10	AN22	AN8–AN10	AN20–AN22
			1	AN11	AN23	AN8–AN11	AN20–AN23

Note: \* Should be cleared to 0.

## Analog Input Channels

Bit 3: CH3	Bit 2: CH2	Bit 1: CH1	Bit 0: CH0	8-Channel Scan Mode		12-Channel Scan Mode			
				A/D0	A/D1	A/D0	A/D1		
0	0	0	0	AN0, AN4	AN12, AN16	AN0, AN4, AN8	AN12, AN16, AN20		
			1	AN0, AN1, AN4, AN5	AN12, AN13, AN16, AN17	AN0, AN1, AN4, AN5, AN8, AN9	AN12, AN13, AN16, AN17, AN20, AN21		
			1	0	AN0–AN2, AN4–AN6	AN12–AN14, AN16–AN18	AN0–AN2, AN4–AN6, AN8–AN10	AN12–AN14, AN16–AN18, AN20–AN22	
			1	AN0–AN7	AN12–AN19	AN0–AN11	AN12–AN23		
	1	0	0	0	AN0, AN4	AN12, AN16	AN0, AN4, AN8	AN12, AN16, AN20	
				1	AN0, AN1, AN4, AN5	AN12, AN13, AN16, AN17	AN0, AN1, AN4, AN5, AN8, AN9	AN12, AN13, AN16, AN17, AN20, AN21	
				1	0	AN0–AN2, AN4–AN6	AN12–AN14, AN16–AN18	AN0–AN2, AN4–AN6, AN8–AN10	AN12–AN14, AN16–AN18, AN20–AN22
				1	AN0–AN7	AN12–AN19	AN0–AN11	AN12–AN23	
	1	0* <sup>1</sup>	0	0	Reserved* <sup>2</sup>	Reserved* <sup>2</sup>	AN0, AN4, AN8	AN12, AN16, AN20	
				1			AN0, AN1, AN4, AN5, AN8, AN9	AN12, AN13, AN16, AN17, AN20, AN21	
				1	0			AN0–AN2, AN4–AN6, AN8–AN10	AN12–AN14, AN16–AN18, AN20–AN22
				1			AN0–AN11	AN12–AN23	

Notes: 1. Should be cleared to 0.

2. These modes are provided for future expansion, and cannot be used at present.

### 17.2.3 A/D Control Registers 0 to 2 (ADCR0 to ADCR2)

A/D control registers 0 to 2 (ADCR0 to ADCR2) are 8-bit readable/writable registers that control the start of A/D conversion and selects the operating clock for A/D0 to A/D2.

ADCR0 to ADCR2 are initialized to H'0F by a power-on reset, and in hardware standby mode and software standby mode.

Bits 3 to 0 of ADCR0 to ADCR2 are reserved. These bits cannot be modified. These bits are always read as 1.

Bit:	7	6	5	4	3	2	1	0
	TRGE	CKS	ADST	ADCS	—	—	—	—
Initial value:	0	0	0	0	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R	R	R	R

- Bit 7—Trigger Enable (TRGE): Enables or disables triggering of A/D conversion by external input or the ATU-II.

#### Bit 7:

TRGE	Description
0	A/D conversion triggering by external input or ATU-II is disabled (Initial value)
1	A/D conversion triggering by external input or ATU-II is enabled

For details of external or ATU-II trigger selection, see section 17.2.5, A/D Trigger Registers 0 to 2 (ADTRGR0 to ADTRGR2).

When ATU triggering is selected, clear bit 7 of registers ADTRGR0 to ADTRGR2 to 0.

When external triggering is selected, upon input of the low level of a pulse to the  $\overline{\text{ADTRG0}}$  or  $\overline{\text{ADTRG1}}$  pin after TRGE has been set to 1, the A/D converter detects the falling edge of a pulse and sets the ADST bit to 1 in ADCR. The same operation is subsequently performed when 1 is written in the ADST bit by software. External triggering of A/D conversion is only enabled when the ADST bit is cleared to 0.

When external triggering is used, the low level input to the  $\overline{\text{ADTRG0}}$  or  $\overline{\text{ADTRG1}}$  pin must be at least 1.5 P $\phi$  clock cycles in width. For details, see section 17.4.4, External Triggering of A/D Conversion.



- Bit 6—Clock Select (CKS): Selects the A/D conversion time. A/D conversion is executed in a maximum of 266 states when CKS is 0, and a maximum of 134 states when 1. To prevent incorrect operation, ensure that the ADST bit A/D control registers 0 to 2 (ADCR0 to ADCR2) is cleared to 0 before changing the A/D conversion time. For details, see section 17.4.3, Analog Input Sampling and A/D Conversion Time.

**Bit 6:**

CKS	Description
0	Conversion time = 266 states (maximum) (Initial value)
1	Conversion time = 134 states (maximum)

- Bit 5—A/D Start (ADST): Starts or stops A/D conversion. A/D conversion is started when ADST is set to 1, and stopped when ADST is cleared to 0.

**Bit 5:**

ADST	Description
0	A/D conversion is stopped (Initial value)
1	A/D conversion is being executed [Clearing conditions] <ul style="list-style-type: none"> <li>• Single mode: Automatically cleared to 0 when A/D conversion ends</li> <li>• Scan mode: Automatically cleared to 0 on completion of one round of conversion on all set channels (single-cycle scan)</li> </ul>

Note that the operation of the ADST bit differs between single mode and scan mode.

In single mode, ADST is automatically cleared to 0 when A/D conversion ends on one channel. In scan mode (continuous scan), when all conversions have ended for the selected analog inputs, ADST remains set to 1 in order to start A/D conversion again for all the channels. Therefore, in scan mode (continuous scan), the ADST bit must be cleared to 0, stopping A/D conversion, before changing the conversion time or the analog input channel selection. However, in scan mode (single-cycle scan), the ADST bit is automatically cleared to 0, stopping A/D conversion, when one round of conversion ends on all the set channels.

Ensure that the ADST bit in ADCR0 to ADCR2 is cleared to 0 before switching the operating mode.

Also, make sure that A/D conversion is stopped (ADST is cleared to 0) before changing A/D interrupt enabling (bit ADIE in ADCSR0 to ADCSR2), the A/D conversion time (bit CKS in ADCR0 to ADCR2), the operating mode (bits ADM1 and ADM0 in ADSCR0 to ADCSR2), or the analog input channel selection (bits CH3 to CH0 in ADCSR0 to ADCSR2). The A/D data register contents will not be guaranteed if these changes are made while the A/D converter is operating (ADST is set to 1).

- Bit 4—A/D Continuous Scan (ADCS): Selects either single-cycle scan or continuous scan in scan mode. This bit is valid only when scan mode is selected. See section 17.4.2, Scan Mode, for details.

**Bit 4:**

ADCS	Description
0	Single-cycle scan (Initial value)
1	Continuous scan

- Bits 3 to 0—Reserved: These bits are always read as 1. The write value should always be 1.

### 17.2.4 A/D Control/Status Register 2 (ADCSR2)

A/D control/status register 2 (ADCSR2) is an 8-bit readable/writable register whose functions include selection of the A/D conversion mode for A/D2.

ADCSR2 is initialized to H'08 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	ADF	ADIE	ADM1	ADM0	—	CH2	CH1	CH0
Initial value:	0	0	0	0	1	0	0	0
R/W:	R/(W)*	R/W	R/W	R/W	R	R/W	R/W	R/W

Note: \* Only 0 can be written to clear the flag.

- Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

**Bit 7:**

ADF	Description
0	Indicates that A/D2 is performing A/D conversion, or is in the idle state (Initial value) [Clearing conditions] <ul style="list-style-type: none"> <li>• When ADF is read while set to 1, then 0 is written to ADF</li> <li>• When the DMAC is activated by ADI2</li> </ul>
1	Indicates that A/D2 has finished A/D conversion, and the digital value has been transferred to ADDR [Setting conditions] <ul style="list-style-type: none"> <li>• Single mode: When A/D conversion ends</li> <li>• Scan mode: When all set A/D conversions end</li> </ul>

The operation of the A/D converter after ADF is set to 1 differs between single mode and scan mode.

In single mode, after the A/D converter transfers the digit value to ADDR, ADF is set to 1 and the A/D converter enters the idle state. In scan mode, ADF is set to 1 after all the set conversions end. For example, in the case of 8-channel scanning, ADF is set to 1 immediately after the end of conversion for AN28 to AN31 (group 7). After ADF is set to 1, conversion continues in the case of continuous scanning, and ends in the case of single-cycle scanning.

Note that 1 cannot be written to ADF.

- Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the A/D interrupt (ADI).

To prevent incorrect operation, ensure that the ADST bit in A/D control register 2 (ADCR2) is cleared to 0 before switching the operating mode.

**Bit 6:**

<b>ADIE</b>	<b>Description</b>	
0	A/D interrupt (ADI2) is disabled	(Initial value)
1	A/D interrupt (ADI2) is enabled	

When A/D conversion ends and the ADF bit in ADCSR2 is set to 1, an A/D2 A/D interrupt (ADI2) will be generated if the ADIE bit is 1. ADI2 is cleared by clearing ADF or ADIE to 0.

- Bits 5 and 4—A/D Mode 1 and 0 (ADM1, ADM0): These bits select the A/D conversion mode from single mode, 4-channel scan mode, and 8-channel scan mode.

To prevent incorrect operation, ensure that the ADST bit in A/D control register 2 (ADCR2) is cleared to 0 before switching the operating mode.

<b>Bit 5: ADM1</b>	<b>Bit 4: ADM0</b>	<b>Description</b>	
0	0	Single mode	(Initial value)
	1	4-channel scan mode (analog groups 6 and 7)	
1	0	8-channel scan mode (analog groups 6 and 7)	
	1	Reserved	

When ADM1 and ADM0 are set to 00, single mode is set. In single mode, operation ends after A/D conversion has been performed once on the analog channels selected with bits CH2 to CH0 in ADCSR.

When ADM1 and ADM0 are set to 01, 4-channel scan mode is set. In scan mode, A/D conversion is performed continuously on a number of channels. The channels on which A/D conversion is to be performed in scan mode are set with bits CH2 to CH0 in ADCSR2. In 4-channel scan mode, conversion is performed continuously on the channels in one of analog groups 6 (AN24 to AN27) or 7 (AN28 to AN31).

When the ADCS bit is cleared to 0, selecting scanning of all channels within the group (AN24 to AN27, AN28 to AN31), conversion is performed continuously, once only for each channel within the group, and operation stops on completion of conversion for the last (highest-numbered) channel.

When ADM1 and ADM0 are set to 10, 8-channel scan mode is set. In 8-channel scan mode, conversion is performed continuously on the 8 channels in analog groups 6 (AN24 to AN27) and 7 (AN28 to AN31). When the ADCS bit is cleared to 0, selecting scanning of all channels within the groups (AN24 to AN31), conversion is performed continuously, once only for each channel within the groups, and operation stops on completion of conversion for the last (highest-numbered) channel.

For details of the operation in single mode and scan mode, see section 17.4, Operation.

- Bit 3—Reserved: This bit is always read as 1. The write value should always be 0.
- Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits, together with the ADM1 and ADM0 bits, select the analog input channels.

To prevent incorrect operation, ensure that the ADST bit in A/D control register 2 (ADCR2) is cleared to 0 before changing the analog input channel selection.

			<b>Analog Input Channels</b>		
<b>Bit: CH2</b>	<b>Bit: CH1</b>	<b>Bit: CH0</b>	<b>Single Mode</b>	<b>4-Channel Scan Mode</b>	<b>8-Channel Scan Mode</b>
0	0	0	AN24 (Initial value)	AN24	AN24, AN28
		1	AN25	AN24, AN25	AN24, AN25, AN28, AN29
	1	0	AN26	AN24–AN26	AN24–AN26, AN28–AN30
		1	AN27	AN24–AN27	AN24–AN31
1	0	0	AN28	AN28	AN24, AN28
		1	AN29	AN28, AN29	AN24, AN25, AN28, AN29
	1	0	AN30	AN28–AN30	AN24–AN26, AN28–AN30
		1	AN31	AN28–AN31	AN24–AN31

## 17.2.5 A/D Trigger Registers 0 to 2 (ADTRGR0 to ADTRGR2)

The A/D trigger registers (ADTRGR0 to ADTRGR2) are 8-bit readable/writable registers that select the A/D0, A/D1, and A/D2 triggers. Either external pin ( $\overline{\text{ADTRG0}}$ ,  $\overline{\text{ADTRG1}}$ ) or ATU-II (ATU-II interval timer A/D conversion request) triggering can be selected.

ADTRGR0 to ADTRGR2 are initialized to H'FF by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	EXTRG	—	—	—	—	—	—	—
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R	R	R	R	R	R	R

- Bit 7—Trigger Enable (EXTRG): Selects external pin input ( $\overline{\text{ADTRG0}}$ ,  $\overline{\text{ADTRG1}}$ ) or the ATU-II interval timer A/D conversion request.

### Bit 7:

EXTRG	Description
0	A/D conversion is triggered by the ATU-II channel 0 interval timer A/D conversion request
1	A/D conversion is triggered by external pin input ( $\overline{\text{ADTRG}}$ ) (Initial value)

In order to select external triggering or ATU-II triggering, the TGRE bit in ADCR0 to ADCR2 must be set to 1. For details, see section 17.2.3, A/D Control Registers 0 to 2 (ADCR0 to ADCR2).

- Bits 6 to 0—Reserved: These bits are always read as 1. The write value should always be 1.

## 17.3 CPU Interface

A/D data registers 0 to 31 (ADDR0 to ADDR31) are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, the upper and lower bytes must be read separately.

To prevent the data being changed between the reads of the upper and lower bytes of an A/D data register, the lower byte is read via a temporary register (TEMP). The upper byte can be read directly.

Data is read from an A/D data register as follows. When the upper byte is read, the upper-byte value is transferred directly to the CPU and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When performing byte-size reads on an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained. If a word-size read is performed on an A/D data register, reading is performed in upper byte, lower byte order automatically.

Figure 17.2 shows the data flow for access to an A/D data register.

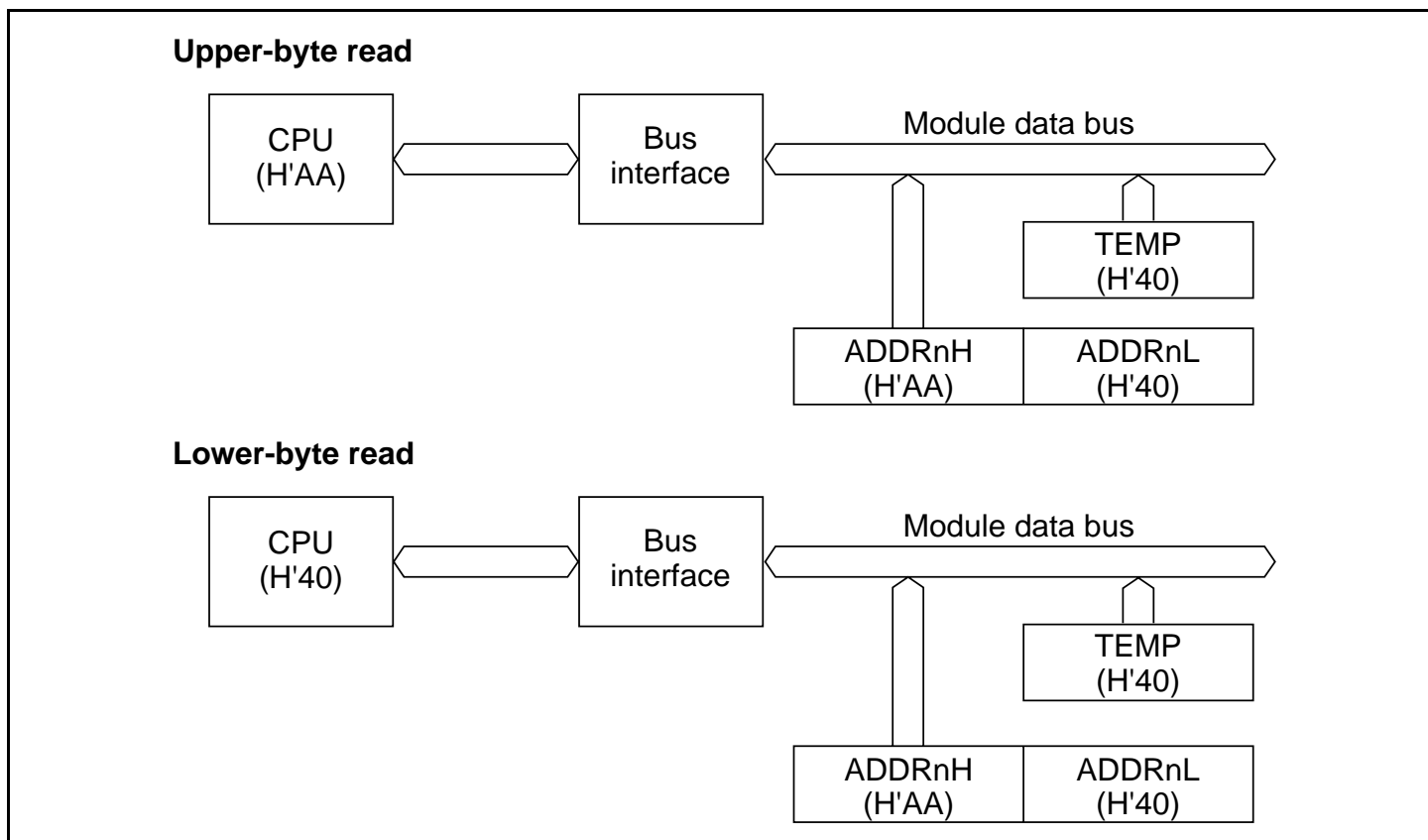


Figure 17.2 A/D Data Register Access Operation (Reading H'AA40)

## 17.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode. There are two kinds of scan mode: continuous and single-cycle. In single mode, conversion is performed once on one specified channel, then ends. In continuous scan mode, A/D conversion continues on one or more specified channels until the ADST bit is cleared to 0. In single-cycle scan mode, A/D conversion ends after being performed once on one or more channels.

### 17.4.1 Single Mode

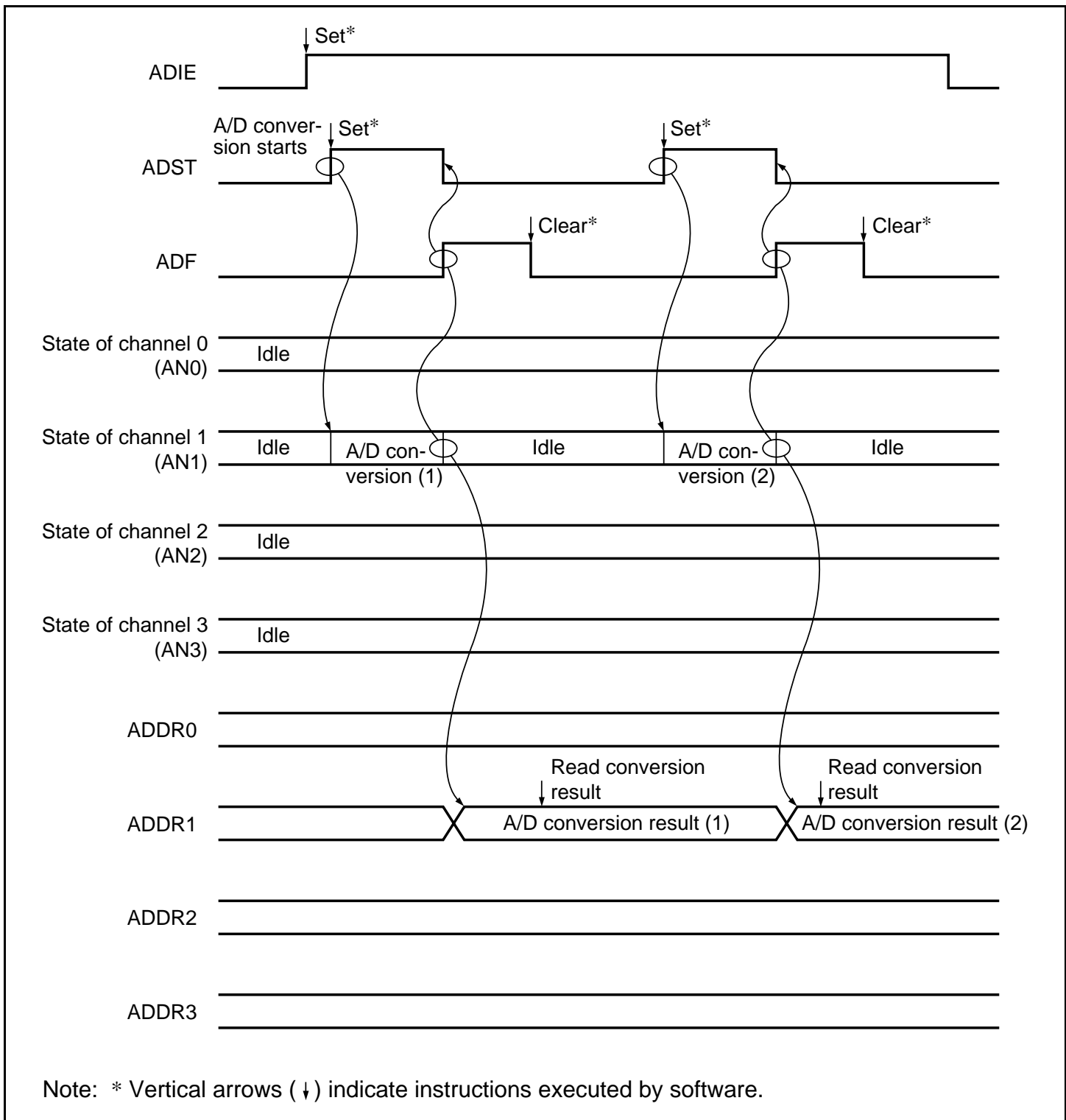
Single mode, should be selected when only one A/D conversion on one channel is required. Single mode is selected by setting the ADM1 and ADM0 bits in the A/D control/status register (ADCSR) to 00. When the ADST bit in the A/D control register (ADCR) is set to 1, A/D conversion is started in single mode.

The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends.

When conversion ends, the ADF flag in ADCSR is set to 1. If the ADIE bit in ADCSR is also 1, an ADI interrupt is requested. To clear the ADF flag, first read ADF when set to 1, then write 0 to ADF. If the DMAC is activated by the ADI interrupt, ADF is cleared automatically.

An example of the operation when analog input channel 1 (AN1) is selected and A/D conversion is performed in single mode is described next. Figure 17.3 shows a timing diagram for this example.

1. Single mode is selected ( $ADM1 = ADM0 = 0$ ), input channel AN1 is selected ( $CH3 = CH2 = CH1 = 0, CH0 = 1$ ), the A/D interrupt is enabled ( $ADIE = 1$ ), and A/D conversion is started ( $ADST = 1$ ).
2. When A/D conversion is completed, the result is transferred to ADDR1. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since  $ADF = 1$  and  $ADIE = 1$ , an ADI interrupt is requested.
4. The A/D interrupt handling routine is started.
5. The routine reads ADF set to 1, then writes 0 to ADF.
6. The routine reads and processes the conversion result (ADDR1).
7. Execution of the A/D interrupt handling routine ends. After this, if the ADST bit is set to 1, A/D conversion starts again and steps 2 to 7 are repeated.



**Figure 17.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)**



## 17.4.2 Scan Mode

Scan mode is useful for monitoring analog inputs in a group of one or more channels. Scan mode is selected for A/D0 or A/D1 by setting the ADM1 and ADM0 bits in A/D control/status register 0 or 1 (ADSCR0 or ADSCR1) to 01 (4-channel scan mode), 10 (8-channel scan mode), or 11 (12-channel scan mode).

For A/D2, scan mode is selected by setting the ADM1 and ADM0 bits in A/D control/status register 2 (ADCSR2) to 01 (4-channel scan mode) or 10 (8-channel scan mode). When the ADCS bit is cleared to 0 and the ADST bit is set to 1 in the A/D control register (ADCR), single-cycle scanning is performed. When the ADCS bit is set to 1 and the ADST bit is set to 1, continuous scanning is performed.

In scan mode, A/D conversion is performed in low-to-high analog input channel number order (AN0, AN1 ... AN11, AN12, AN13 ... AN23, AN24, AN25 ... AN31).

In single-cycle scanning, the ADF bit in ADCSR is set to 1 when conversion has been performed once on all the set channels, and the ADST bit is automatically cleared to 0.

In continuous scanning, the ADF bit in ADCSR is set to 1 when conversion ends on all the set channels. To stop A/D conversion, write 0 to the ADST bit.

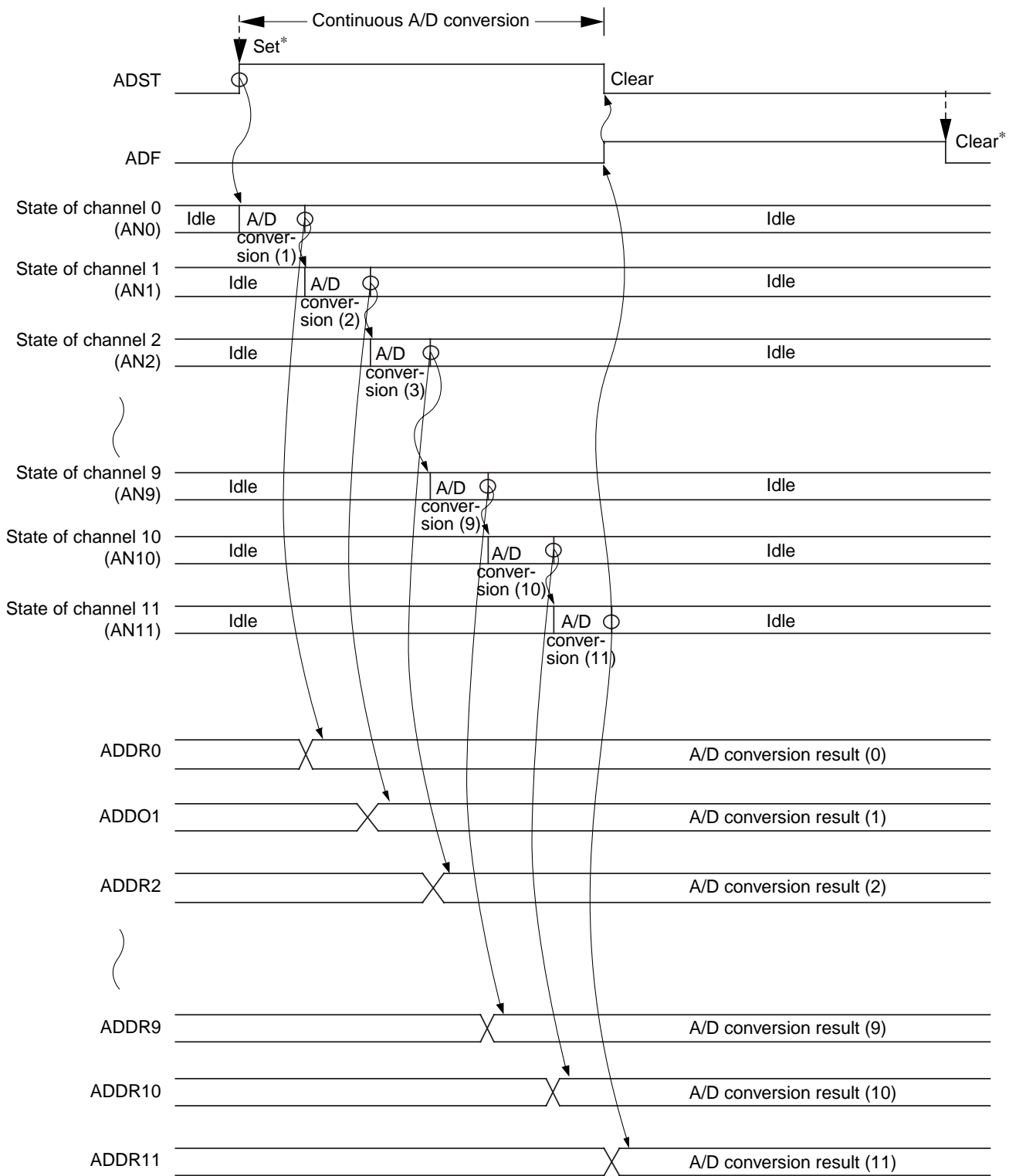
If the ADIE bit in ADCSR is set to 1 when ADF is set to 1, an ADI interrupt (ADI0, ADI1, or ADI2) is requested. To clear the ADF flag, first read ADF when set to 1, then write 0 to ADF. If the DMAC is activated by the ADI interrupt, ADF is cleared to 0 automatically.

An example of the operation when analog inputs 0 to 11 (AN0 to AN11) are selected and A/D conversion is performed in single-cycle scan mode is described below. Figure 17.4 shows the operation timing for this example.

1. 12-channel scan mode is selected (ADM1 = 1, ADM0 = 1), single-cycle scan mode is selected (ADCS = 0), analog input channels AN0 to AN11 are selected (CH3 = 0, CH2 = 0, CH1 = 1, CH0 = 1), and A/D conversion is started.
2. When conversion of the first channel (AN0) is completed, the result is transferred to ADDR0. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the 12th channel (AN11).
4. When conversion is completed for all the selected channels (AN0 to AN11), the ADF flag is set to 1, the ADST bit is cleared to 0 automatically, and A/D conversion stops. If the ADIE bit is 1, an ADI interrupt is requested after A/D conversion ends.

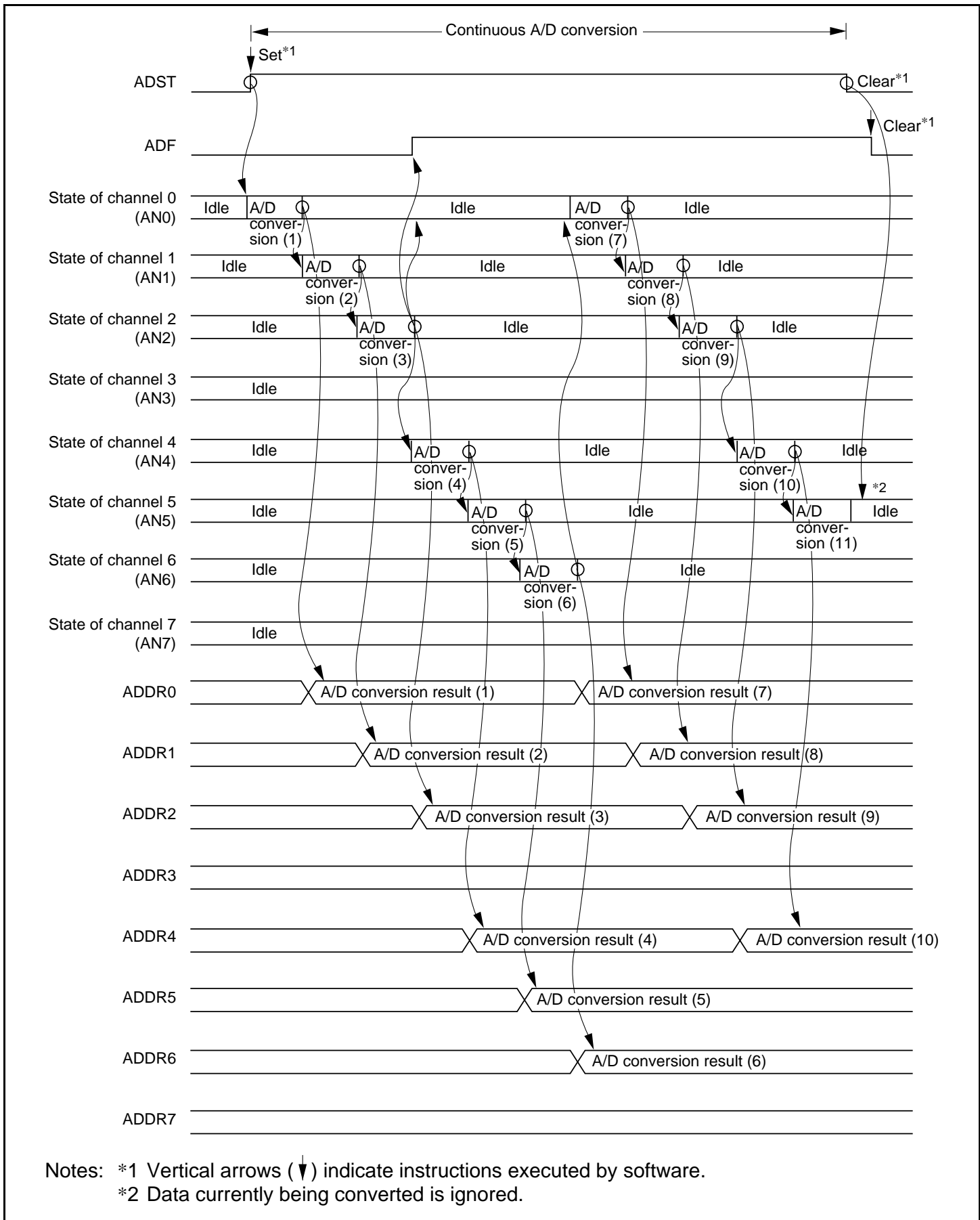
An example of the operation when analog inputs 0 to 2 and 4 to 6 (AN0 to AN2 and AN4 to AN6) are selected and A/D conversion is performed in 8-channel scan mode is described below. Figure 17.5 shows the operation timing.

1. 8-channel scan mode is selected ( $ADM1 = 1$ ,  $ADM0 = 0$ ) continuous scan mode is selected ( $ADCS = 1$ ), analog input channels AN0 to AN2 and AN4 to AN6 are selected ( $CH3 = 0$ ,  $CH2 = 0$ ,  $CH1 = 1$ ,  $CH0 = 0$ ), and A/D conversion is started.
2. When conversion of the first channel (AN0) is completed, the result is transferred to ADDR0. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. Conversion of the fifth channel (AN4) starts automatically.
5. Conversion proceeds in the same way through the seventh channel (AN6)
6. When conversion is completed for all the selected channels (AN0 to AN2 and AN4 to AN6), the ADF flag is set to 1. If the ADIE bit is also 1, an ADI interrupt is requested.
7. Steps 2 to 6 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After this, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).



Note: \* Vertical arrows (↓) indicate instructions executed by software.

**Figure 17.4 Example of A/D Converter Operation (Scan Mode (Single-Cycle Scan), Channels AN0 to AN11 Selected)**



**Figure 17.5 Example of A/D Converter Operation (Scan Mode (Continuous Scan), Channels AN0 to AN2 and AN4 to AN6 Selected)**

### 17.4.3 Analog Input Sampling and A/D Conversion Time

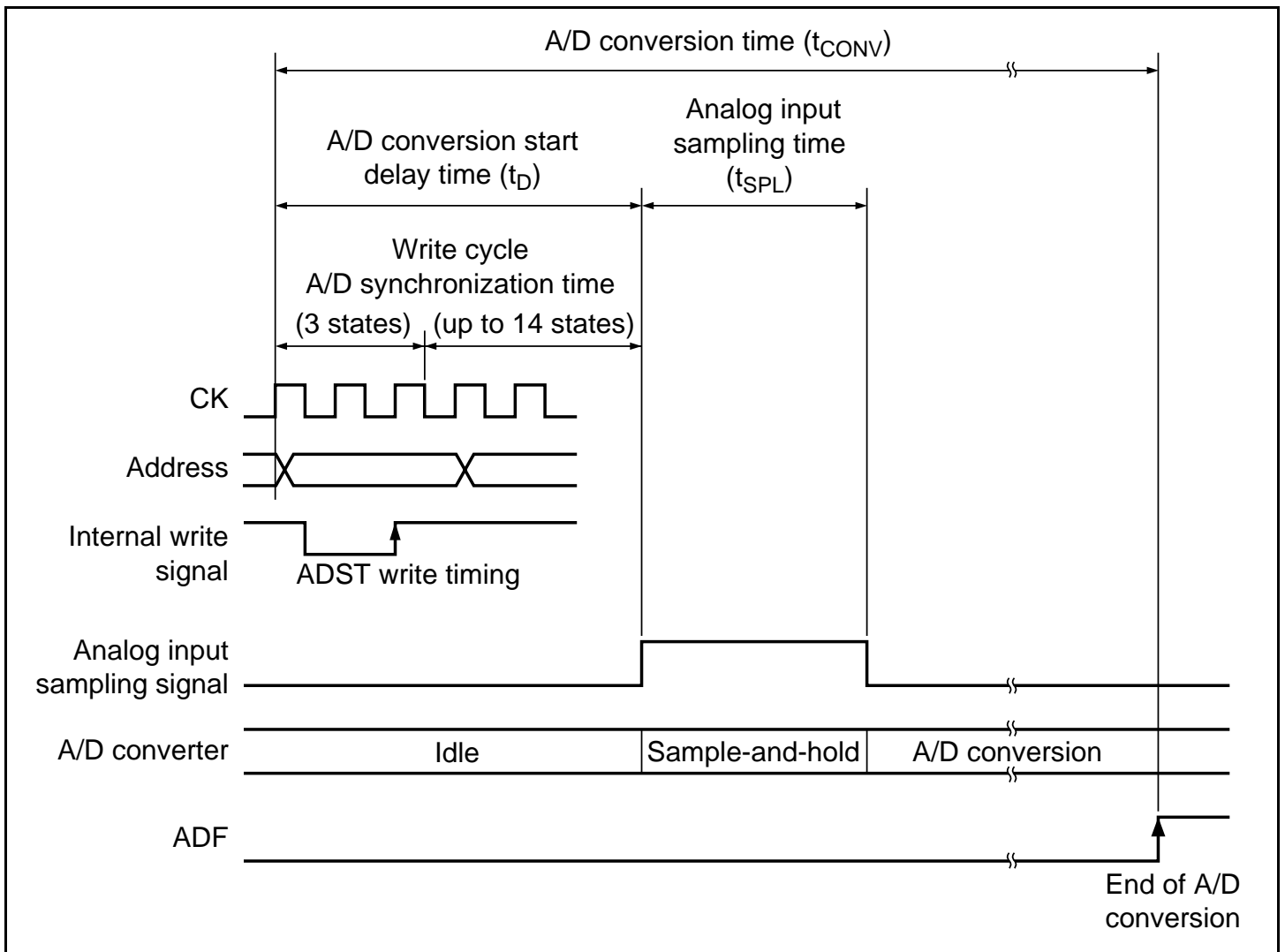
The A/D converter has a built-in sample-and-hold circuit in A/D0, A/D1, and A/D2. The A/D converter samples the analog input at time  $t_d$  (A/D conversion start delay time) after the ADST bit is set to 1, then starts conversion. Figure 17.6 shows the A/D conversion timing.

The A/D conversion time ( $t_{CONV}$ ) includes  $t_d$  and the analog input sampling time ( $t_{SPL}$ ). The length of  $t_d$  is not fixed, since it includes the time required for synchronization of the A/D conversion operation. The total conversion time therefore varies within the ranges shown in table 17.4.

In scan mode, the  $t_{CONV}$  values given in table 17.4 apply to the first conversion. In the second and subsequent conversions,  $t_{CONV}$  is fixed at 256 states when CKS = 0 or 128 states when CKS = 1.

**Table 17.4 A/D Conversion Time (Single Mode)**

Item	Symbol	CKS = 0: Peripheral Clock (P $\phi$ ) = 10 to 20 MHz			CKS = 1: Peripheral Clock (P $\phi$ ) = 10 MHz			Unit
		Min	Typ	Max	Min	Typ	Max	
A/D conversion start delay time	$t_d$	10	—	17	6	—	9	States (peripheral clock (P $\phi$ ))
Input sampling time	$t_{SPL}$	—	64	—	—	32	—	
A/D conversion time	$t_{CONV}$	259	—	266	131	—	134	



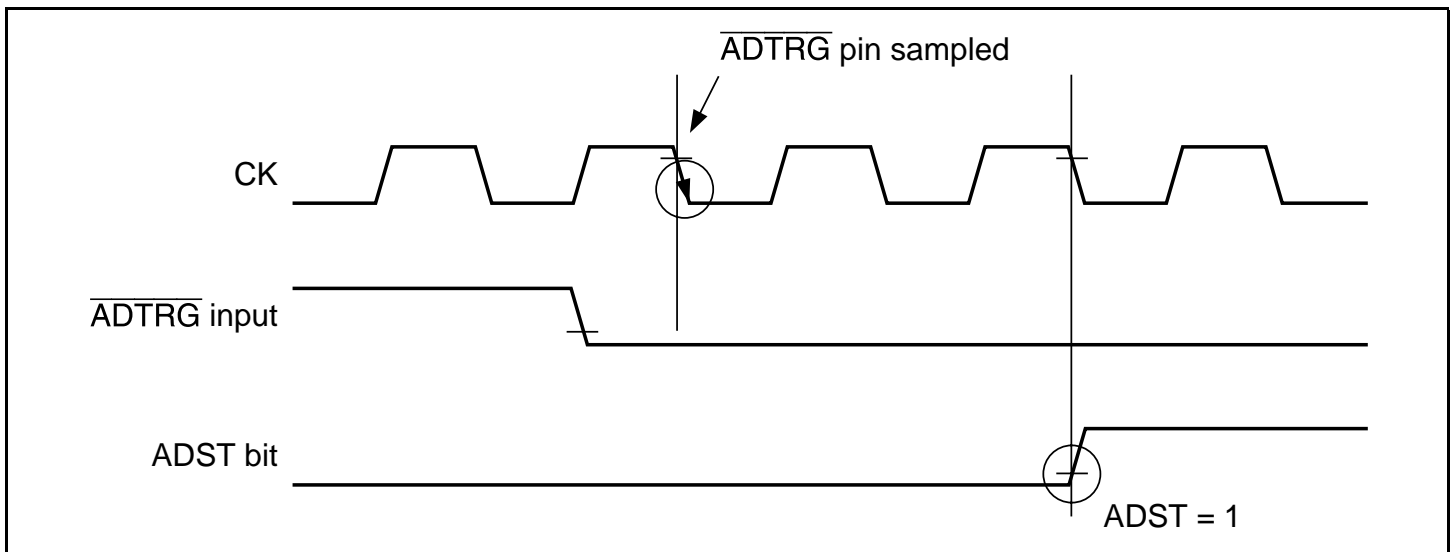
**Figure 17.6 A/D Conversion Timing**

#### 17.4.4 External Triggering of A/D Conversion

The A/D converter can be activated by input of an external A/D conversion start trigger.

To activate the A/D converter with an external trigger, first set the pin functions with the PFC (pin function controller), then set the TRGE bit to 1 in the A/D control register (ADCR), and set the EXTRG bit to 1 in the A/D trigger register (ADTRGR). When a low level is input to the  $\overline{\text{ADTRG}}$  pin after these settings have been made, the A/D converter detects the falling edge of a pulse and sets the ADST bit to 1. Figure 17.7 shows the timing for external trigger input.

The ADST bit is set to 1 two states after the A/D converter samples the falling edge on the  $\overline{\text{ADTRG}}$  pin. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written into the ADST bit by software.



**Figure 17.7 External Trigger Input Timing**

### 17.4.5 A/D Converter Activation by ATU-II

The A/D0, A/D1, and A/D2 converter modules can be activated by an A/D conversion request from the ATU-II's channel 0 interval timer.

To activate the A/D converter by means of the ATU-II, set the TRGE bit to 1 in the A/D control register (ADCR) and clear the EXTRG bit to 0 in the A/D trigger register (ADTRGR). When an ATU-II channel 0 interval timer A/D conversion request is generated after these settings have been made, the ADST bit set to 1. The timing from setting of the ADST bit until the start of A/D conversion is the same as when 1 is written into the ADST bit by software.

### 17.4.6 ADEND Output Pin

When channel 31 is used in scan mode, the conversion timing can be monitored with the ADEND output pin.

After the channel 31 analog voltage has been latched in scan mode, and conversion has started, the ADEND pin goes high. The ADEND pin subsequently goes low when channel 31 conversion ends.

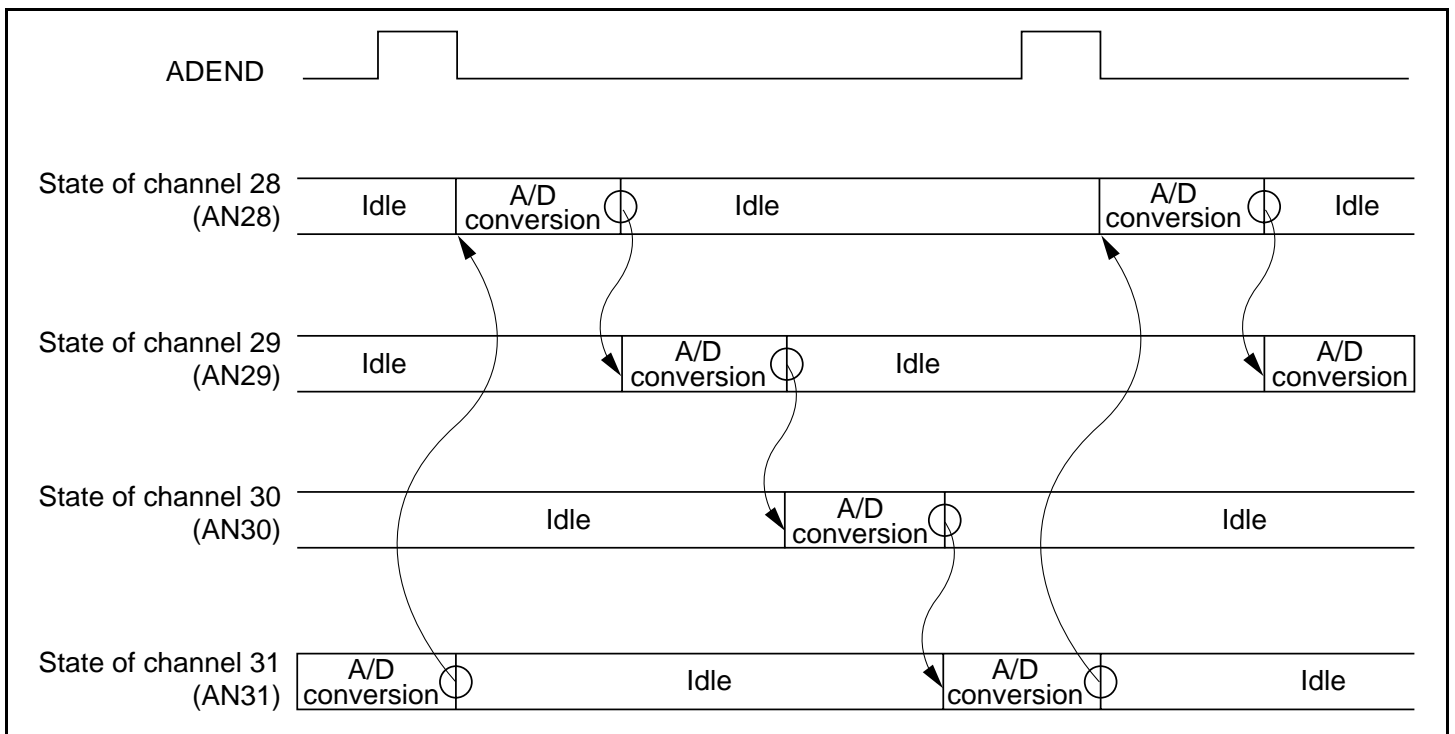


Figure 17.8 ADEND Output Timing



## 17.5 Interrupt Sources and DMA Transfer Requests

The A/D converter can generate an A/D conversion end interrupt request (ADI0, ADI1, or ADI2) upon completion of A/D conversions. The ADI interrupt can be enabled by setting the ADIE bit in the A/D control/status register (ADCSR) to 1, or disabled by clearing the ADIE bit to 0.

The DMAC can be activated by an ADI interrupt. In this case an interrupt request is not sent to the CPU.

When the DMAC is activated by an ADI interrupt, the ADF bit in ADCSR is automatically cleared when data is transferred by the DMAC.

See section 10.4.2, Example of DMA Transfer between A/D Converter and On-Chip Memory (Address Reload On), for an example of this operation.

## 17.6 Usage Notes

The following points should be noted when using the A/D converter.

### 1. Analog input voltage range

The voltage applied to analog input pins during A/D conversion should be in the range  $AV_{SS} \leq AN_n \leq AV_{ref}$ .

### 2. Relation between, $AV_{SS}$ , $AV_{CC}$ and $V_{SS}$ , $V_{CC}$

When using the A/D converter, set  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ , and  $AV_{SS} = V_{SS}$ . When the A/D converter is not used, set  $AV_{SS} = V_{SS}$ , and do not leave the  $AV_{CC}$  pin open.

### 3. $AV_{ref}$ input range

Set  $AV_{ref} = 4.5 \text{ V}$  to  $AV_{CC}$  when the A/D converter is used, and  $AV_{ref} \leq AV_{CC}$  when not used.

If conditions above are not met, the reliability of the device may be adversely affected.

### 4. Notes on board design

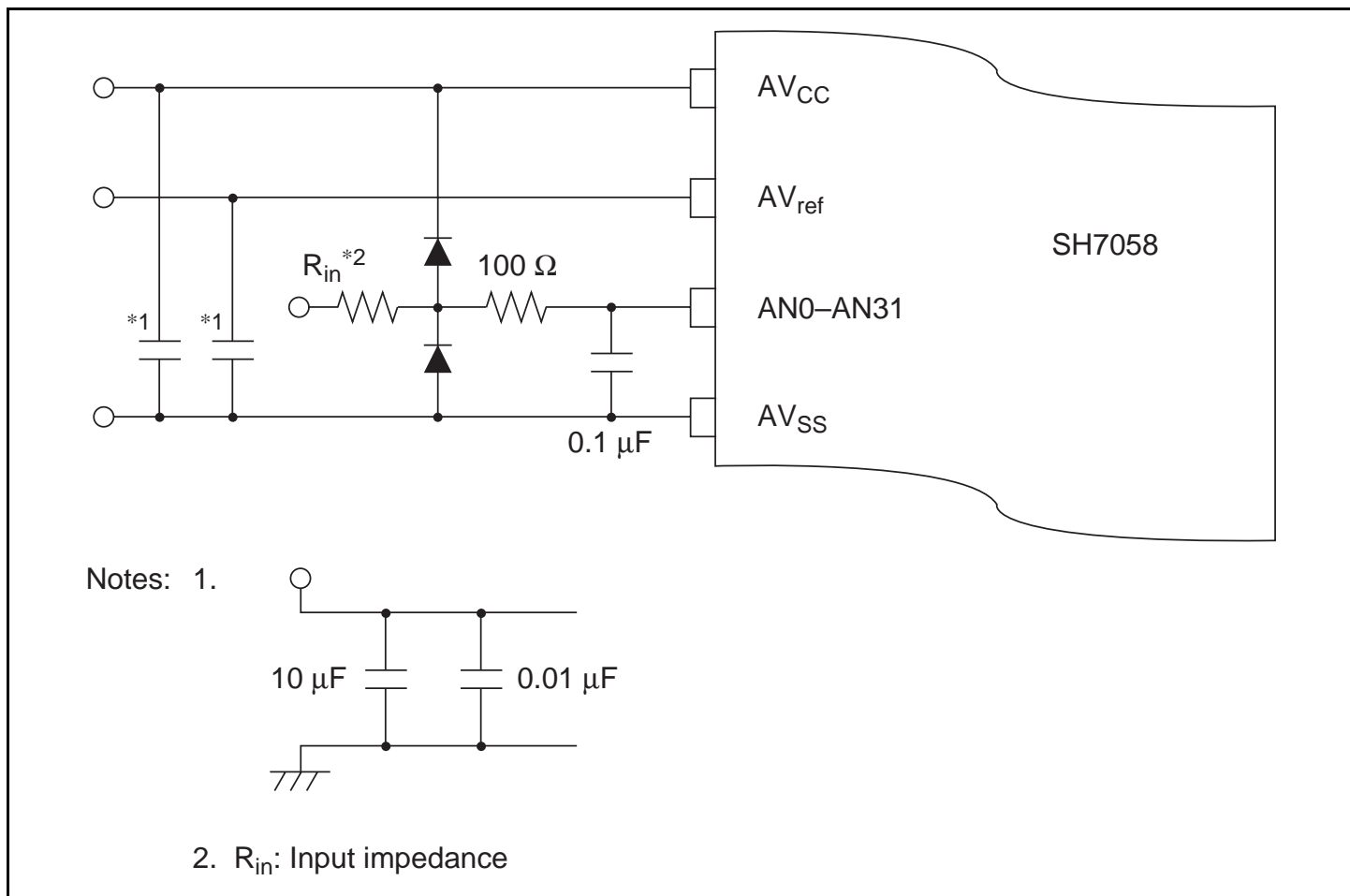
In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Also, digital circuitry must be isolated from the analog input signals ( $AN_n$ ), analog reference voltage ( $AV_{ref}$ ), and analog power supply ( $AV_{CC}$ ) by the analog ground ( $AV_{SS}$ ).  $AV_{SS}$  should be connected at one point to a stable digital ground ( $V_{SS}$ ) on the board.

### 5. Notes on noise countermeasures

A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins ( $AN_n$ ) and analog reference voltage ( $AV_{ref}$ ) should be connected between  $AV_{CC}$  and  $AV_{SS}$  as shown in figure 17.9.

Also, the bypass capacitors connected to AVCC and AVref and the filter capacitor connected to ANn must be connected to AVSS. If a filter capacitor is connected as shown in figure 17.9, the input currents at the analog input pins (ANn) are averaged, and so an error may arise. Careful consideration is therefore required when deciding the circuit constants.



**Figure 17.9 Example of Analog Input Pin Protection Circuit**

**Table 17.5 Analog Pin Specifications**

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	3	k $\Omega$

### 17.6.1 A/D conversion accuracy definitions

A/D conversion accuracy definitions are given below.

#### 1. Resolution

The number of A/D converter digital conversion output codes

2. Offset error

The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 0000000000 to 0000000001 (does not include quantization error) (see figure 17.10).

3. Full-scale error

The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 1111111110 to 1111111111 (does not include quantization error) (see figure 17.10).

4. Quantization error

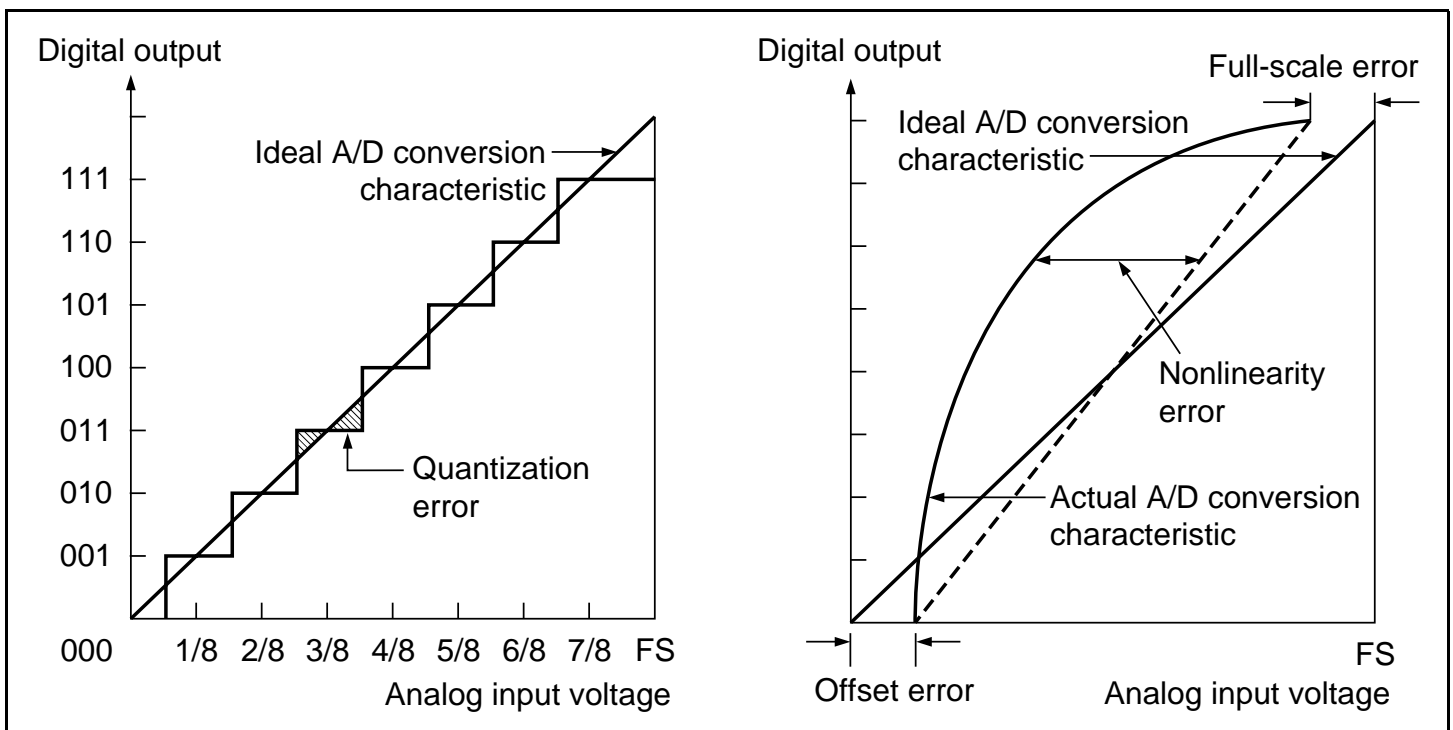
The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 17.10).

5. Nonlinearity error

The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error.

6. Absolute accuracy

The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.



**Figure 17.10 A/D Conversion Accuracy Definitions**

# Section 18 Multi-Trigger A/D Converter (MTAD)

## 18.1 Overview

The multi-trigger A/D converter (MTAD) is composed of two independent modules A/D0 and A/D1, as listed below.

Module	Analog Group	Channels
A/D0	Analog group 2	AN8 to AN11
A/D1	Analog group 5	AN20 to AN23

### 18.1.1 Feature

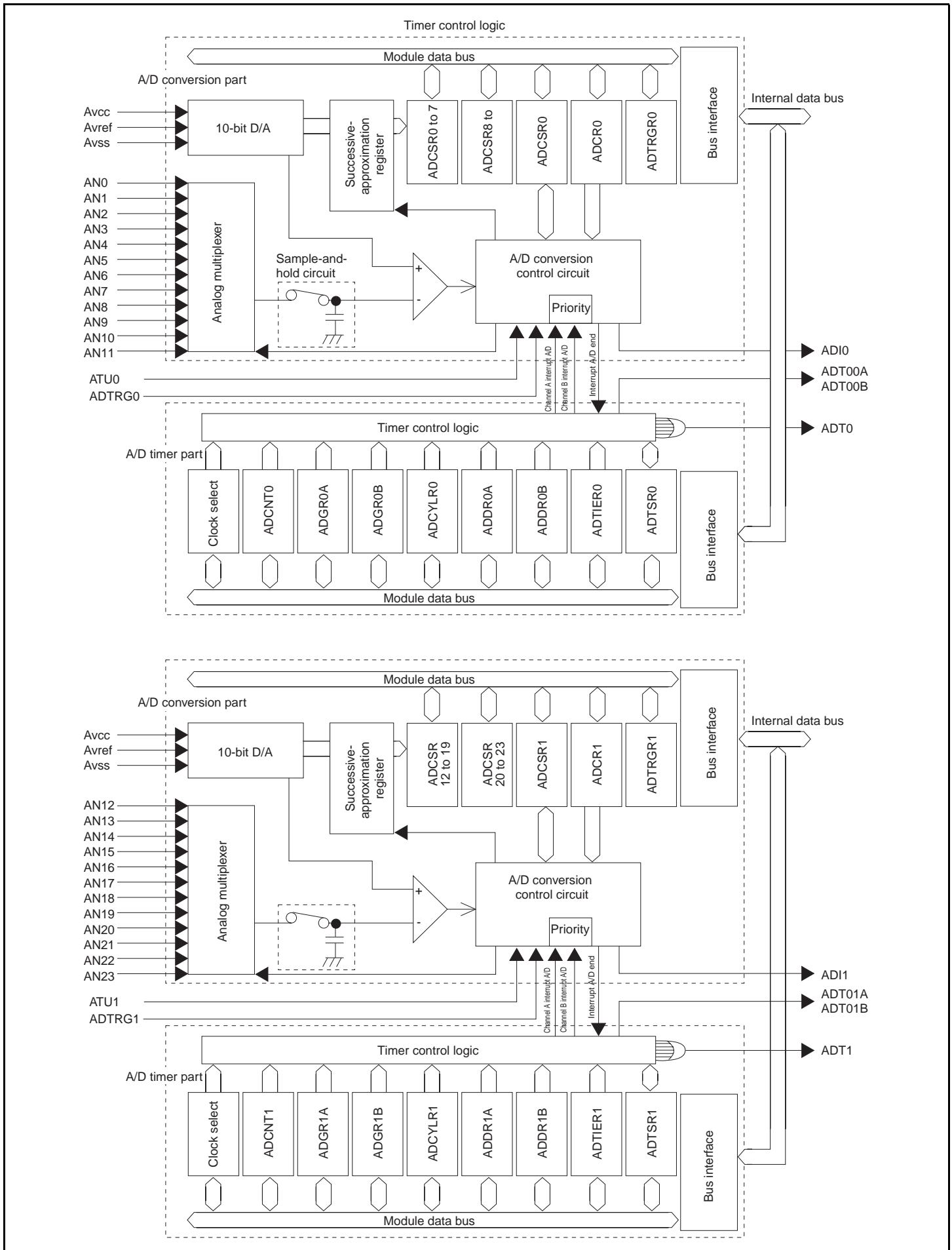
The feature of the multi-trigger A/D conversion is shown below.

- Multi-trigger A/D conversion mode

While performing conversion on the specified channels in scan mode, A/D conversion on the channels for which conversion has been requested can be performed prior to the other channels when a compare match occurs with respect to the timer in the A/D converter.

### 18.1.2 Block Diagram

Figure 18.1 shows a block diagram of the multi-trigger A/D converter.



**Figure 18.1 Simplified Block Diagram of Multi-Trigger A/D Converter**

### 18.1.3 Input/Output Pins

Table 18.1 summarizes the multi-trigger A/D converter output pins. When using these external pins, the pin function controller (PFC) should also be set in accordance with the A/D conversion settings.

**Table 18.1 Pin Configuration**

Channel	Pin Name	Abbreviation	I/O	Function
0	A/D timer output 0A	ADTO0A	Output	PWM output
0	A/D timer output 0B	ADTO0B	Output	PWM output
1	A/D timer output 1A	ADTO1A	Output	PWM output
1	A/D timer output 1B	ADTO1B	Output	PWM output

### 18.1.4 Register Configuration

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	A/D free-running counter	ADCNT0	R/W	H'0001	H'FFFFFF860 (upper byte) H'FFFFFF861 (lower byte)	16
0	A/D cycle register 0	ADCYLR0	R/W	H'FFFF	H'FFFFFF862 (upper byte) H'FFFFFF863 (lower byte)	16
0	A/D duty register 0A	ADDR0A	R/W	H'FFFF	H'FFFFFF864 (upper byte) H'FFFFFF865 (lower byte)	16
0	A/D duty register 0B	ADDR0B	R/W	H'FFFF	H'FFFFFF866 (upper byte) H'FFFFFF867 (lower byte)	16
0	A/D general register 0A	ADGR0A	R/W	H'FFFF	H'FFFFFF868 (upper byte) H'FFFFFF869 (lower byte)	16
0	A/D general register 0B	ADGR0B	R/W	H'FFFF	H'FFFFFF86A (upper byte) H'FFFFFF86B (lower byte)	16

Channel	Register Name	Abbreviation	R/W	Initial Value	Address	Access Size
0	A/D trigger control register 0	ADTCR0	R/W	H'00	H'FFFFFF86C	8
0	A/D trigger status register 0	ADTSR0	R/(W)*	H'00	H'FFFFFF86D	8
0	A/D trigger interrupt enable register 0	ADTIER0	R/W	H'00	H'FFFFFF86E	8
1	A/D free-running counter 1	ADCNT1	R/W	H'0001	H'FFFFFF870 (upper byte) H'FFFFFF871 (lower byte)	16
1	A/D cycle register 1	ADCYLR1	R/W	H'FFFF	H'FFFFFF872 (upper byte) H'FFFFFF873 (lower byte)	16
1	A/D duty register 1A	ADDR1A	R/W	H'FFFF	H'FFFFFF874 (upper byte) H'FFFFFF875 (lower byte)	16
1	A/D duty register 1B	ADDR1B	R/W	H'FFFF	H'FFFFFF876 (upper byte) H'FFFFFF877 (lower byte)	16
1	A/D general register 1A	ADGR1A	R/W	H'FFFF	H'FFFFFF878 (upper byte) H'FFFFFF879 (lower byte)	16
1	A/D general register 1B	ADGR1B	R/W	H'FFFF	H'FFFFFF87A (upper byte) H'FFFFFF87B (lower byte)	16
1	A/D trigger control register	ADTCR1	R/W	H'00	H'FFFFFF87C	8
1	A/D trigger status register 1	ADTSR1	R/(W)*	H'00	H'FFFFFF87D	8
1	A/D trigger interrupt enable register 1	ADTIER1	R/W	H'00	H'FFFFFF87E	8

Note: \* Only 0 can be written.

## 18.2 Register Descriptions

### 18.2.1 A/D Trigger Control Registers 0 and 1 (ADTCR0 and ADTCR1)

A/D trigger control registers 0 and 1 (ADTCR0 and ADTCR1) are 8-bit readable/writable registers whose functions include selection of the prescaler.

ADTCR0 and ADTCR1 are initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	CKSEL1x	CKSEL0x	—	—	DTSELxB	DTSELxA	ADSELxB	ADSELxA
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R	R/W	R/W	R/W	R/W

- Bits 7 and 6—Clock Select 1 and 0 (CKSEL1x and CKSEL0x): Halt the counter or select internal clock  $\phi$  from among  $\phi/2$ ,  $\phi/5$ , and  $\phi/10$ , which are obtained by dividing clock  $\phi$ .

Bit 7: CKSEL1x	Bit 6: CKSEL0x	Description
0	0	Counter is halted
0	1	Counter is incremented with internal clock $\phi = \phi/2$
1	0	Counter is incremented with internal clock $\phi = \phi/5$
1	1	Counter is incremented with internal clock $\phi = \phi/10$

- Bits 5 and 4—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 3—Duty Select 1B or 0B (DTSEL1B or DTSEL0B): Selects either on-duty or off-duty for the PWM output from ADTOxB of channel xB.

Bit 3: DTSELxB	Description
0	On-duty for the PWM output from ADTOxB (Initial value)
1	Off-duty for the PWM output from ADTOxB

Note: x = 0 or 1.



- Bit 2—Duty Select 1A or 0A (DTSEL1A or DTSEL0A): Selects either on-duty or off-duty for the PWM output from ADTOxA of channel xA.

**Bit 2:**

DTSELxA	Description	
0	On-duty for the PWM output from ADTOxA	(Initial value)
1	Off-duty for the PWM output from ADTOxA	

Note: x = 0 or 1.

- Bit 1—A/D Data Select 1B (ADSEL1B): Selects the register to which the result of multi-trigger A/D conversion is transferred.  
This bit is inverted when the ADDR register is updated by the multi-trigger A/D conversion. Switching settings during the multi-trigger A/D conversion operation should be carried out when TADF1B (ADTSR1 register) is 1.

**Bit 1:**

ADSEL1B	Description	
0	Conversion result is transferred to ADDR22	(Initial value)
1	Conversion result is transferred to ADDR23	

- Bit 1—A/D Data Select 0B (ADSEL0B): Selects the register to which the result of multi-trigger A/D conversion is transferred.  
This bit is inverted when the ADDR register is updated by the multi-trigger A/D conversion. Switching settings during the multi-trigger A/D conversion operation should be carried out when TADF0B (ADTSR0 register) is set to 1.

**Bit 1:**

ADSEL0B	Description	
0	Conversion result is transferred to ADDR10	(Initial value)
1	Conversion result is transferred to ADDR11	

- Bit 0—A/D Data Select 1A (ADSEL1A): Selects the register to which the result of multi-trigger A/D conversion is transferred.

This bit is inverted when the ADDR register is updated by the multi-trigger A/D conversion. Switching settings during the multi-trigger A/D conversion operation should be carried out when TADF1A (ADTSR1 register) is set to 1.

**Bit 0:**

ADSEL1A	Description
0	Conversion result is transferred into ADDR20 (Initial value)
1	Conversion result is transferred into ADDR21

- Bit 0—A/D Data Select 0A (ADSEL0A): Selects the register to which the result of multi-trigger A/D conversion is transferred.

This bit is inverted when the ADDR register is updated by the multi-trigger A/D conversion. Switching settings during the multi-trigger A/D conversion operation should be carried out when TADF0A (ADTSR0 register) is set to 1.

**Bit 0:**

ADSEL0A	Description
0	Conversion result is transferred to ADDR8 (Initial value)
1	Conversion result is transferred to ADDR9

### 18.2.2 A/D Trigger Status Registers 0 and 1 (ADTSR0 and ADTSR1)

A/D trigger status registers 0 and 1 (ADTSR0 and ADTSR1) indicate the compare match generation and the multi-trigger A/D conversion status in channels 0 and 1.

ADTSR0 and ADTSR1 are initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	—	TADFxB	TADFxA	ADDFxB	ADDFxA	ADCYLFx	ADCMFxB	ADCMFxA
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)*

Note: x = 0 or 1.

\* Only 0 can be written, to clear the flag.

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—Trigger A/D Flag B (TADFxB): Indicates the end of multi-trigger A/D conversion B.

#### Bit 6: TADFxB

	Description
0	Indicates that the multi-trigger A/D converter is performing A/D conversion B, or the converter is in the idle state (Initial value) [Clearing condition] When TADFxB is read while set to 1, then 0 is written to TADFxB
1	Indicates that the multi-trigger A/D converter has finished A/D conversion B, and the digital value has been transferred to ADDR [Setting condition] When multi-trigger A/D conversion B ends

Note: x = 0 or 1.

- Bit 5—Trigger A/D Flag A (TADFxA): Indicates the end of multi-trigger A/D conversion A.

#### Bit 5: TADFxA

	Description
0	Indicates that the multi-trigger A/D converter is performing A/D conversion A, or the converter is in the idle state (Initial value) [Clearing condition] When TADFxA is read while set to 1, then 0 is written to TADFxA
1	Indicates that the multi-trigger A/D converter has finished A/D conversion A, and the digital value has been transferred to ADDR [Setting condition] When multi-trigger A/D conversion A ends

Note: x = 0 or 1.

- Bit 4—A/D Duty Flag B (ADDFxB): Indicates whether or not the ADDRxB and ADCNT values have matched.

#### Bit 4:

ADDFxB	Description
0	[Clearing condition] (Initial value) When ADDFxB is read while set to 1, then 0 is written to ADDFxB
1	[Setting condition] When ADCNTx and ADDRxB values have matched

Note: x = 0 or 1.

- Bit 3—A/D Duty Flag A (ADDFxA): Indicates whether or not the ADDRxA and ADCNT values have matched.

**Bit 3:**

ADDFxA	Description
0	[Clearing condition] (Initial value) When ADDFxA is read while set to 1, then 0 is written to ADDFxA
1	[Setting condition] When ADCNTx and ADDRxA values have matched

Note: x = 0 or 1.

- Bit 2—A/D Cycle Compare Match Flow Flag (ADCYLFx): Indicates whether or not the ADCYLRx and ADCNT values have matched.

**Bit 2:**

ADCYLFx	Description
0	[Clearing condition] (Initial value) When ADCYLFx is read while set to 1, then 0 is written to ADCYLFx
1	[Setting condition] When ADCNTx and ADCYLRx values have matched

Note: x = 0 or 1.

- Bit 1—A/D Compare Match Flag (ADCMFxB): Indicates whether or not the ADGRxB and ADCNT values have matched.

**Bit 1:**

ADCMFxB	Description
0	[Clearing condition] (Initial value) When ADCMFxB is read while set to 1, then 0 is written to ADCMFxB
1	[Setting condition] When ADCNTx and ADGRxB values have matched

Note: x = 0 or 1.

- Bit 0—A/D Compare Match Flag (ADCMFxA): Indicates whether or not the ADGRxA and ADCNT values have matched.

**Bit 0:**

ADCMFxA	Description
0	[Clearing condition] (Initial value) When ADCMFxA is read while set to 1, then 0 is written to ADCMFxA
1	[Setting condition] When ADCNTx and ADGRxA values have matched

Note: x = 0 or 1.

### 18.2.3 A/D Trigger Interrupt Enable Registers 0 and 1 (ADTIER0 and ADTIER1)

A/D trigger interrupt enable registers 0 and 1 (ADTIER0 and ADTIER1) enable or disable interrupt request triggered by the compare match generation and multi-trigger A/D conversion end in channels 0 and 1.

ADTIER0 and ADTIER1 are initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode.

Bit:	7	6	5	4	3	2	1	0
	ADTRGx	TADExB	TADExA	ADDExB	ADDExA	ADCYLEx	ADCMExB	ADCNExA
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: x = 0 or 1.

- Bit 7—ADT Trigger (ADTRGx): Enables or disables triggering of multi-trigger A/D conversion by a compare match between ADCNTx and ADGRxA or ADGRxB.  
To prevent incorrect operation, ensure that the ADST bit in A/D control register (ADCR) is 0 before switching this setting.

**Bit 1:**

ADTRGx	Description
0	Triggering of multi-trigger A/D conversion by a compare match between ADCNTx and ADGRxA or ADGRxB is disabled (Initial value)
1	Triggering of multi-trigger A/D conversion by a compare match between ADCNTx and ADGRxA or ADGRxB is enabled

Notes: 1. x = 0 or 1.

2. Value 1 can be set to ADTRGx only for the cases below; 0 should always be set for the other cases.

## Conversion mode (ADCR): continuous scan

### Channels for conversion (ADCSR<sub>x</sub>):

Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Analog Input Channels	
ADM1	ADM0	CH3	CH2	CH1	CH0	A/D0	A/D1
0	1	0	0	1	1	AN0 to AN3	AN12 to AN15
1	0	0	0	1	1	AN0 to AN7	AN12 to AN19
0	1	0	1	1	1	AN4 to AN7	AN16 to AN19

Notes: 1. x = 0 or 1.

2. For the ADCR and ADCSR<sub>x</sub> settings, refer to section 17, A/D Converter.

- Bit 6—Trigger A/D Interrupt Enable B (TADE<sub>x</sub>B): Enables or disables the interrupt request by TADF<sub>x</sub>B when the trigger A/D flag xB (TADF<sub>x</sub>B) in ADTSR is set to 1.  
To prevent incorrect operation, ensure that the ADTRG bit in A/D trigger interrupt enable register (ADTIER0 or ADTIER1) is 0 before switching this setting.

#### Bit 6: TADE<sub>x</sub>B Description

0	The interrupt request (TAD <sub>I</sub> xB) by TADF <sub>x</sub> B is disabled	(Initial value)
1	The interrupt request (TAD <sub>I</sub> xB) by TADF <sub>x</sub> B is enabled	

When multi-trigger A/D conversion B ends, setting TADF<sub>x</sub>B to 1, a trigger A/D interrupt for A/D0 or A/D1 (TAD<sub>I</sub>xB) is requested if TADE<sub>x</sub>B is 1. TAD<sub>I</sub>xB can be cleared to 0 by clearing TADF<sub>x</sub>B or TADE<sub>x</sub>B to 0.

- Bit 5—Trigger A/D Interrupt Enable A (TADE<sub>x</sub>A): Enables or disables the interrupt request by TADF<sub>x</sub>A when the trigger A/D flag xA (TADF<sub>x</sub>A) in ADTSR is set to 1.  
To prevent incorrect operation, ensure that the ADTRG bit in A/D trigger interrupt enable register (ADTIER0 or ADTIER1) is 0 before switching this setting.

#### Bit 5: TADE<sub>x</sub>A Description

0	The interrupt request (TAD <sub>I</sub> xA) by TADF <sub>x</sub> A is disabled	(Initial value)
1	The interrupt request (TAD <sub>I</sub> xA) by TADF <sub>x</sub> A is enabled	

When multi-trigger A/D conversion A ends setting TADF<sub>x</sub>A to 1, a trigger A/D interrupt for A/D0 or A/D1 (TAD<sub>I</sub>xA) is requested if TADE<sub>x</sub>A is 1. TAD<sub>I</sub>xA can be cleared to 0 by clearing TADF<sub>x</sub>A or TADE<sub>x</sub>A to 0.

- Bit 4—A/D Duty Interrupt Enable B (ADDExB): Enables or disables the interrupt request by ADDFxB when the ADDRxB compare match flag (ADDFxB) in ADTSR is set to 1.

**Bit 4:**

ADDExB	Description	
0	The interrupt request (ADDIxB) by ADDFxB is disabled	(Initial value)
1	The interrupt request (ADDIxB) by ADDFxB is enabled	

Note: x = 0 or 1.

- Bit 3—A/D Duty Interrupt Enable A (ADDExA): Enables or disables the interrupt request by ADDFxA when the ADDRxA compare match flag (ADDFxA) in ADTSR is set to 1.

**Bit 3:**

ADDExA	Description	
0	The interrupt request (ADDIxA) by ADDFxA is disabled	(Initial value)
1	The interrupt request (ADDIxA) by ADDFxA is enabled	

Note: x = 0 or 1.

- Bit 2—A/D Cycle Interrupt Enable (ADCYLEx): Enables or disables the interrupt request by ADCYLFx when the A/D cycle compare match flow flag (ADCYLFx) in ADTSRx is set to 1.

**Bit 2:**

ADCYLEx	Description	
0	The interrupt request (ADCYIx) by ADCYLFx is disabled	(Initial value)
1	The interrupt request (ADCYIx) by ADCYLFx is enabled	

Note: x = 0 or 1.

- Bit 1—A/D Compare Match Interrupt Enable B (ADCMExB): Enables or disables the interrupt request by ADCMFxB when the ADDRxB compare match flag (ADCMFxB) in ADTSR is set to 1.

**Bit 1:**

ADCMExB	Description	
0	The interrupt request (ADDIxB) by ADCMFxB is disabled	(Initial value)
1	The interrupt request (ADDIxB) by ADCMFxB is enabled	

Note: x = 0 or 1.

- Bit 0—A/D Compare Match Interrupt Enable A (ADCMExA): Enables or disables the interrupt request by ADCMFxA when the ADDRxA compare match flag (ADCMFxA) in ADTSR is set to 1.

**Bit 0:                      Description**  
**ADCMExA**

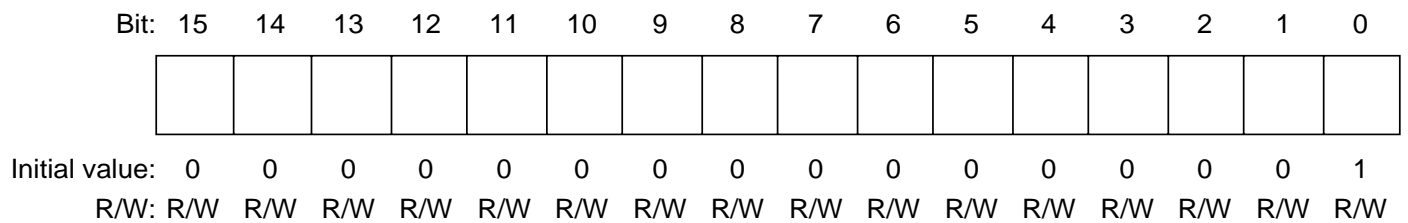
0	The interrupt request (ADDIxA) by ADCMFxA is disabled	(Initial value)
1	The interrupt request (ADDIxA) by ADCMFxA is enabled	

Note: x = 0 or 1.

### 18.2.4 A/D Free-Running Counters (ADCNT0 and ADCNT1)

A/D free-running counters 0 and 1 (ADCNT0 and ADCNT1) are 16-bit readable/writable registers that start incrementing according to the setting of the A/D trigger control registers (ADTCR0 and ADTCR1).

The clock selected by the prescaler (ADTCR0 and ADTCR1) is input to the corresponding counters. ADCNT0 and ADCNT1 are initialized to H'0001 by a power-on reset, and in hardware standby mode and software standby mode. ADCNT0 and ADCNT1 can only be read from or written to in words.





### 18.2.5 A/D General Registers A and B (ADGR0A, ADGR0B, ADGR1A, and ADGR1B)

A/D general registers (ADGR0A, ADGR0B, ADGR1A, and ADGR1B) are 16-bit readable/writable registers. Two registers are provided for each of channels 0 and 1.

The ADGR value is constantly compared with the corresponding free-running counter (ADCNT0 or ADCNT1) value. When the two values match, the ADCMFxA and ADCMFxB bits in the corresponding A/D trigger status register (ADTSR) are set to 1, which requests initiation of the multi-trigger A/D conversion. ADGR0A, ADGR0B, ADGR1A, and ADGR1B can only be read from or written to in words.

ADGR0A, ADGR0B, ADGR1A, and ADGR1B are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

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

### 18.2.6 A/D Cycle Registers 0 and 1 (ADCYLR0 and ADCYLR1)

A/D cycle registers (ADCYLR0 and ADCYLR1) are 16-bit readable/writable registers.

One register is provided for each of channels 0 and 1.

The ADCYLR value is constantly compared with the corresponding free-running counter (ADCNT0 or ADCNT1) value. When the two values match, the ADCYLFx bit in the corresponding A/D trigger status register (ADTSR) is set to 1, which clears ADCNT0 and ADCNT1 to H'0001. ADCYLR0 and ADCYLR1 can only be read from or written to in words.

ADCYLR0 and ADCYLR1 are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.

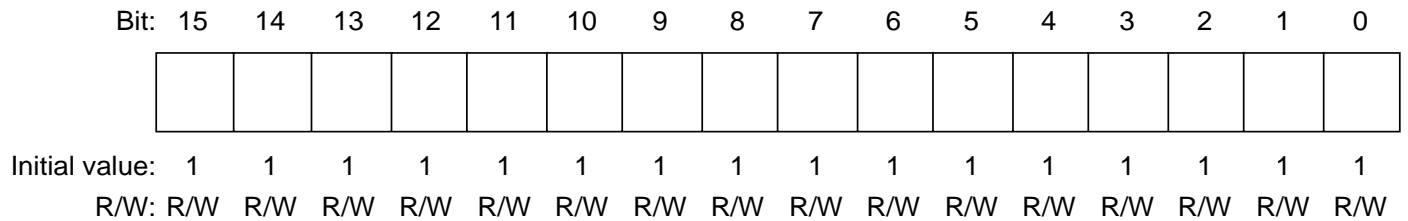
Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

### 18.2.7 A/D Duty Registers A and B (ADDR0A, ADDR0B, ADDR1A, and ADDR1B)

A/D duty registers (ADDR0A, ADDR0B, ADDR1A, and ADDR1B) are 16-bit readable/writable registers. Two registers are provided for each of channels 0 and 1.

The ADDR value is constantly compared with the corresponding free-running counter (ADCNT0 or ADCNT1) value. When the two values match, the ADDFxA and ADDFxB bits in the corresponding A/D trigger status register (ADTSR) are set to 1. ADDR0A, ADDR0B, ADDR1A, and ADDR1B can only be read from or written to in words.

ADDR0A, ADDR0B, ADDR1A, and ADDR1B are initialized to H'FFFF by a power-on reset, and in hardware standby mode and software standby mode.



## 18.3 Interrupt Interface

### 18.3.1 On-Chip Peripheral Module Interrupts

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

- Direct memory access controller (DMAC)
- Advanced timer unit (ATU-II)
- Compare match timer (CMT)
- A/D converter (A/D)
- Multi-trigger A/D (MTAD)
- Serial communication interface (SCI)
- Watchdog timer (WDT)
- Controller area network (HCAN)

A different interrupt vector is assigned to each interrupt source, so the exception service routine does not have to decide which interrupt has occurred. Priority levels between 0 and 15 can be assigned to individual on-chip peripheral modules in interrupt priority registers C–L (IPRC–IPRL).

On-chip peripheral module interrupt exception processing sets the interrupt mask level bits (I3–I0) in the status register (SR) to the priority level value of the on-chip peripheral module interrupt that was accepted.

### 18.3.2 Interrupt Exception Vectors and Priority Rankings

Table 18.2 lists interrupt sources and their vector numbers, vector table address offsets and interrupt priorities.

Each interrupt source is allocated a different vector number and vector table address offset. Vector table addresses are calculated from vector numbers and address offsets. In interrupt exception processing, the exception service routine start address is fetched from the vector table indicated by the vector table address. See table 6.4, Calculating Exception Processing Vector Table Addresses, in section 6, Exception Processing.

IRQ interrupts and on-chip peripheral module interrupt priorities can be set freely between 0 and 15 for each pin or module by setting interrupt priority registers A–L (IPRA–IPRL). The ranking of interrupt sources for IPRC–IPRL, however, must be the order listed under Priority within IPR Setting Range in table 18.2 and cannot be changed. A power-on reset assigns priority level 0 to IRQ interrupts and on-chip peripheral module interrupts. If the same priority level is assigned to

two or more interrupt sources and interrupts from those sources occur simultaneously, their priority order is the default priority order indicated at the right in table 18.2.

**Table 18.2 Interrupt Exception Processing Vectors and Priorities**

Interrupt Source	Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range		Default Priority	
	Vector No.	Vector Table Address Offset			↑	↓		
NMI	11	H'0000002C to H'0000002F	16	—	—	—	High	
UBC	12	H'00000030 to H'00000033	15	—	—	—		
H-UDI	14	H'00000038 to H'0000003B	15	—	—	—		
IRQ0	64	H'00000100 to H'00000103	0 to 15 (0)	IPRA (15–12)	—	—		
IRQ1	65	H'00000104 to H'00000107	0 to 15 (0)	IPRA (11–8)	—	—		
IRQ2	66	H'00000108 to H'0000010B	0 to 15 (0)	IPRA (7–4)	—	—		
IRQ3	67	H'0000010C to H'0000010F	0 to 15 (0)	IPRA (3–0)	—	—		
IRQ4	68	H'00000110 to H'00000113	0 to 15 (0)	IPRB (15–12)	—	—		
IRQ5	69	H'00000114 to H'00000117	0 to 15 (0)	IPRB (11–8)	—	—		
IRQ6	70	H'00000118 to H'0000011B	0 to 15 (0)	IPRB (7–4)	—	—		
IRQ7	71	H'0000011C to H'0000011F	0 to 15 (0)	IPRB (3–0)	—	—		
DMAC0	DEI0	72	H'00000120 to H'00000123	0 to 15 (0)	IPRC (15–12)	↑ 1		
DMAC1	DEI1	74	H'00000128 to H'0000012B	0 to 15 (0)		↓ 2		
DMAC2	DEI2	76	H'00000130 to H'00000133	0 to 15 (0)	IPRC (11–8)	↑ 1		
DMAC3	DEI3	78	H'00000138 to H'0000013B	0 to 15 (0)		↓ 2		Low



**Table 18.2 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR		Default Priority
		Vector No.	Vector Table Address Offset			Setting Range		
ATU2	ATU21	IMI2A/ CMI2A	108	H'000001B0 to H'000001B3	0 to 15 (0)	IPRE (11–8)	↑ 1	High
		IMI2B/ CMI2B	109	H'000001B4 to H'000001B7			2	
		IMI2C/ CMI2C	110	H'000001B8 to H'000001BB			3	
		IMI2D/ CMI2D	111	H'000001BC to H'000001BF			↓ 4	
	ATU22	IMI2E/ CMI2E	112	H'000001C0 to H'000001C3	0 to 15 (0)	IPRE (7–4)	↑ 1	
		IMI2F/ CMI2F	113	H'000001C4 to H'000001C7			2	
		IMI2G/ CMI2G	114	H'000001C8 to H'000001CB			3	
		IMI2H/ CMI2H	115	H'000001CC to H'000001CF			↓ 4	
	ATU23	OVI2A/ OVI2B	116	H'000001D0 to H'000001D3	0 to 15 (0)	IPRE (3–0)		
ATU3	ATU31	IMI3A	120	H'000001E0 to H'000001E3	0 to 15 (0)	IPRF (15–12)	↑ 1	
		IMI3B	121	H'000001E4 to H'000001E7			2	
		IMI3C	122	H'000001E8 to H'000001EB			3	
		IMI3D	123	H'000001EC to H'000001EF			↓ 4	
	ATU32	OVI3	124	H'000001F0 to H'000001F3	0 to 15 (0)	IPRF (11–8)		Low

**Table 18.2 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range		Default Priority
			Vector No.	Vector Address Offset					
ATU4	ATU41	IMI4A	128	H'00000200 to H'00000203	0 to 15 (0)	IPRF (7-4)	↑	1	High
		IMI4B	129	H'00000204 to H'00000207			2		
		IMI4C	130	H'00000208 to H'0000020B			3		
		IMI4D	131	H'0000020C to H'0000020F			↓	4	
	ATU42	OVI4	132	H'00000210 to H'00000213	0 to 15 (0)	IPRF (3-0)			
ATU5	ATU51	IMI5A	136	H'00000220 to H'00000223	0 to 15 (0)	IPRG (15-12)	↑	1	
		IMI5B	137	H'00000224 to H'00000227			2		
		IMI5C	138	H'00000228 to H'0000022B			3		
		IMI5D	139	H'0000022C to H'0000022F			↓	4	
	ATU52	OVI5	140	H'00000230 to H'00000233	0 to 15 (0)	IPRG (11-8)			
ATU6		CMI6A	144	H'00000240 to H'00000243	0 to 15 (0)	IPRG (7-4)	↑	1	
		CMI6B	145	H'00000244 to H'00000247			2		
		CMI6C	146	H'00000248 to H'0000024B			3		
		CMI6D	147	H'0000024C to H'0000024F			↓	4	

**Table 18.2 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source	Interrupt Vector			Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR		Default Priority
	Vector No.	Vector Address Offset	Vector Table Address Offset			Setting Range	Priority	
ATU7	CMI7A	148	H'00000250 to H'00000253	0 to 15 (0)	IPRG (3-0)	↑	1	High
	CMI7B	149	H'00000254 to H'00000257			2		
	CMI7C	150	H'00000258 to H'0000025B			3		
	CMI7D	151	H'0000025C to H'0000025F			↓	4	
ATU8 ATU81	OSI8A	152	H'00000260 to H'00000263	0 to 15 (0)	IPRH (15-12)	↑	1	↑
	OSI8B	153	H'00000264 to H'00000267			2		
	OSI8C	154	H'00000268 to H'0000026B			3		
	OSI8D	155	H'0000026C to H'0000026F			↓	4	
ATU82	OSI8E	156	H'00000270 to H'00000273	0 to 15 (0)	IPRH (11-8)	↑	1	↓
	OSI8F	157	H'00000274 to H'00000277			2		
	OSI8G	158	H'00000278 to H'0000027B			3		
	OSI8H	159	H'0000027C to H'0000027F			↓	4	
ATU83	OSI8I	160	H'00000280 to H'00000283	0 to 15 (0)	IPRH (7-4)	↑	1	↓
	OSI8J	161	H'00000284 to H'00000287			2		
	OSI8K	162	H'00000288 to H'0000028B			3		
	OSI8L	163	H'0000028C to H'0000028F			↓	4	



**Table 18.2 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source			Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range		Default Priority	
			Vector No.	Vector Address Offset			↑	↓		
ATU8	ATU84	OSI8M	164	H'00000290 to H'00000293	0 to 15 (0)	IPRH (3–0)	↑	1	High	
		OSI8N	165	H'00000294 to H'00000297			2			
		OSI8O	166	H'00000298 to H'0000029B			3			
		OSI8P	167	H'0000029C to H'0000029F			↓	4		
ATU9	ATU91	CMI9A	168	H'000002A0 to H'000002A3	0 to 15 (0)	IPRI (15–12)	↑	1	↑	
		CMI9B	169	H'000002A4 to H'000002A7			2			
		CMI9C	170	H'000002A8 to H'000002AB			3			
		CMI9D	171	H'000002AC to H'000002AF			↓	4		
	ATU92	CMI9E	172	H'000002B0 to H'000002B3	0 to 15 (0)	IPRI (11–8)	↑	1		
		CMI9F	174	H'000002B8 to H'000002BB			↓	2		
	ATU10	ATU101	CMI10A	176	H'000002C0 to H'000002C3	0 to 15 (0)	IPRI (7–4)	↑		1
			CMI10B	178	H'000002C8 to H'000002CB			↓		2
ATU102		ICI10A/ CMI10G	180	H'000002D0 to H'000002D3	0 to 15(0)			IPRI (3–0)		
ATU11	IMI11A	IMI11A	184	H'000002E0 to H'000002E3	0 to 15 (0)	IPRJ (15–12)	↑	1		
		IMI11B	186	H'000002E8 to H'000002EB			2			
	OVI11	187	H'000002EC to H'000002EF	↓			3	Low		

**Table 18.2 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source	Interrupt Vector			Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR Setting Range		Default Priority	
	Vector No.	Vector Address Offset	Vector Table Address Offset			Setting	Range		
CMT0	CMTI0	188	H'000002F0 to H'000002F3	0 to 15 (0)	I PRJ (11–8)	↑	1	High ↑          ↓ Low	
MTAD0	ADT0	189	H'000002F4 to H'000002F7				2		
A/D0	ADI0	190	H'000002F8 to H'000002FB			↓	3		
CMT1	CMTI1	192	H'00000300 to H'00000303	0 to 15 (0)	IPRJ (7–4)	↑	1		
MTAD1	ADT1	193	H'00000304 to H'00000307				2		
A/D1	ADI1	194	H'00000308 to H'0000030B			↓	3		
A/D2	ADI2	196	H'00000310 to H'00000313	0 to 15 (0)	IPRJ (3–0)				
SCI0	ERI0	200	H'00000320 to H'00000323	0 to 15 (0)	IPRK (15–12)	↑	1		
	RXI0	201	H'00000324 to H'00000327						2
	TXI0	202	H'00000328 to H'0000032B						3
	TEI0	203	H'0000032C to H'0000032F				↓	4	
SCI1	ERI1	204	H'00000330 to H'00000333	0 to 15 (0)	IPRK (11–8)	↑	1		
	RXI1	205	H'00000334 to H'00000337					2	
	TXI1	206	H'00000338 to H'0000033B					3	
	TEI1	207	H'0000033C to H'0000033F				↓	4	

**Table 18.2 Interrupt Exception Processing Vectors and Priorities (cont)**

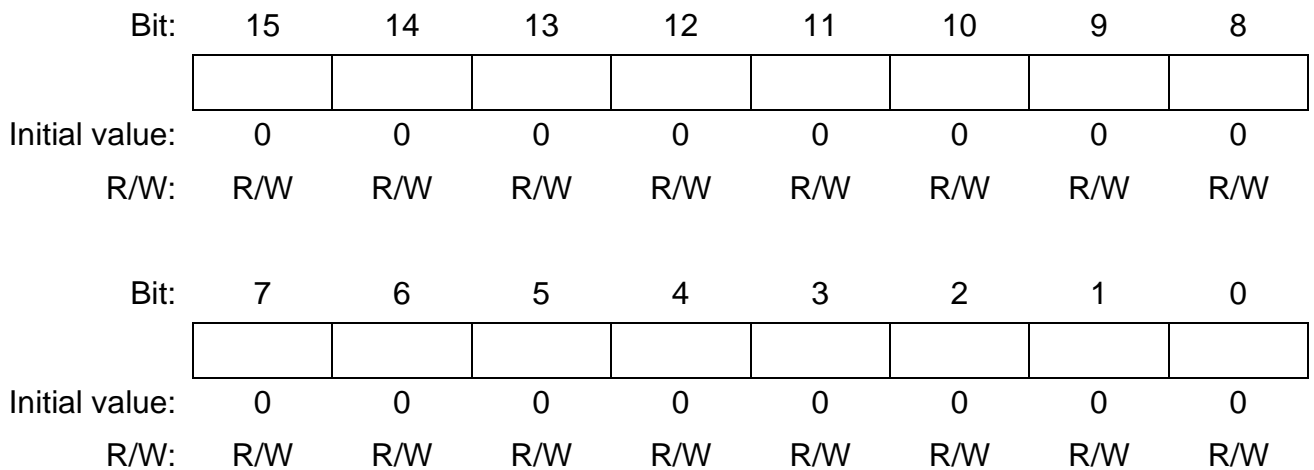
Interrupt Source	Interrupt Vector			Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR		Default Priority
	Vector No.	Vector Address Offset	Vector Table Address Offset			Setting Range		
SCI2	ERI2	208	H'00000340 to H'00000343	0 to 15 (0)	IPRK (7-4)	↑	1	High
	RXI2	209	H'00000344 to H'00000347			2		
	TXI2	210	H'00000348 to H'0000034B			3		
	TEI2	211	H'0000034C to H'0000034F			↓	4	
SCI3	ERI3	212	H'00000350 to H'00000353	0 to 15 (0)	IPRK (3-0)	↑	1	
	RXI3	213	H'00000354 to H'00000357			2		
	TXI3	214	H'00000358 to H'0000035B			3		
	TEI3	215	H'0000035C to H'0000035F			↓	4	
SCI4	ERI4	216	H'00000360 to H'00000363	0 to 15 (0)	IPRL (15-12)	↑	1	
	RXI4	217	H'00000364 to H'00000367			2		
	TXI4	218	H'00000368 to H'0000036B			3		
	TEI4	219	H'0000036C to H'0000036F			↓	4	
HCAN0	ERS0	220	H'00000370 to H'00000373	0 to 15 (0)	IPRL (11-8)	↑	1	
	OVR0	221	H'00000374 to H'00000377			2		
	RM0	222	H'00000378 to H'0000037B			3		
	SLE0	223	H'0000037C to H'0000037F			↓	4	

**Table 18.2 Interrupt Exception Processing Vectors and Priorities (cont)**

Interrupt Source		Interrupt Vector		Interrupt Priority (Initial Value)	Corresponding IPR (Bits)	Priority within IPR		Default Priority
		Vector No.	Vector Table Address Offset			Setting Range		
WDT	ITI	224	H'00000380 to H'00000383	0 to 15 (0)	IPRL (7-4)			High
HCAN1	ERS1	228	H'00000390 to H'00000393	0 to 15 (0)	IPRL (3-0)	↑	1	↑ ↓
	OVR1	229	H'00000394 to H'00000397			2		
	RM1	230	H'00000398 to H'0000039B			3		
	SLE1	231	H'0000039C to H'0000039F			↓	4	

### 18.3.3 Interrupt Priority Registers A-L (IPRA-IPRL)

Interrupt priority registers A-L (IPRA-IPRL) are 16-bit readable/writable registers that set priority levels from 0 to 15 for IRQ interrupts and on-chip peripheral module interrupts. Correspondence between interrupt request sources and each of the IPRA-IPRL bits is shown in table 18.3.



**Table 18.3 Interrupt Request Sources and IPRA–IPRL**

Register	Bits			
	15–12	11–8	7–4	3–0
Interrupt priority register A	IRQ0	IRQ1	IRQ2	IRQ3
Interrupt priority register B	IRQ4	IRQ5	IRQ6	IRQ7
Interrupt priority register C	DMAC0, 1	DMAC2, 3	ATU01	ATU02
Interrupt priority register D	ATU03	ATU04	ATU11	ATU12
Interrupt priority register E	ATU13	ATU21	ATU22	ATU23
Interrupt priority register F	ATU31	ATU32	ATU41	ATU42
Interrupt priority register G	ATU51	ATU52	ATU6	ATU7
Interrupt priority register H	ATU81	ATU82	ATU83	ATU84
Interrupt priority register I	ATU91	ATU92	ATU101	ATU102
<u>Interrupt priority register J</u>	ATU11	CMT0, A/D0, <u>MTAD0</u>	CMT1, A/D1, <u>MTAD1</u>	A/D2
Interrupt priority register K	SCI0	SCI1	SCI2	SCI3
Interrupt priority register L	SCI4	HCAN0	WDT	HCAN1

As indicated in table 18.3, four  $\overline{\text{IRQ}}$  pins or groups of 4 on-chip peripheral modules are allocated to each register. Each of the corresponding interrupt priority ranks are established by setting a value from H'0 (0000) to H'F (1111) in each of the four-bit groups 15–12, 11–8, 7–4 and 3–0. Interrupt priority rank becomes level 0 (lowest) by setting H'0, and level 15 (highest) by setting H'F. If multiple on-chip peripheral modules are assigned to the same bit (DMAC0 and DMAC1, DMAC2 and DMAC3, CMT0, A/D0, and MTAD0, and CMT1, A/D1, and MTAD1), those multiple modules are set to the same priority rank.

IPRA–IPRL are initialized to H'0000 by a reset and in hardware standby mode. They are not initialized in software standby mode.

## 18.4 PFC and I/O Port Interfaces

### 18.4.1 PFC Interface

#### 18.4.2 Port A Control Registers H and L (PACRH, PACRL)

Port A control registers H and L (PACRH, PACRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port A. PACRH selects the functions of the pins for the upper eight bits of port A, and PACRL selects the functions of the pins for the lower eight bits.

PACRH and PACRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

#### Port A Control Register H (PACRH)

Bit:	15	14	13	12	11	10	9	8
	—	PA15MD	—	PA14MD	—	PA13MD	—	PA12MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	PA11MD1	PA11MD0	PA10MD1	PA10MD0	PA9MD1	PA9MD0	PA8MD1	PA8MD0
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 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PA15 Mode Bit (PA15MD): Selects the function of pin PA15/RxD0.

Bit 14: PA15MD	Description
0	General input/output (PA15) (Initial value)
1	Receive data input (RxD0)

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PA14 Mode Bit (PA14MD): Selects the function of pin PA14/TxD0.

Bit 12: PA14MD	Description	
0	General input/output (PA14)	(Initial value)
1	Transmit data output (TxD0)	

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PA13 Mode Bit (PA13MD): Selects the function of pin PA13/TIO5B.

Bit 10: PA13MD	Description	
0	General input/output (PA13)	(Initial value)
1	ATU-II input capture input/output compare output (TIO5B)	

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PA12 Mode Bit (PA12MD): Selects the function of pin PA12/TIO5A.

Bit 8: PA12MD	Description	
0	General input/output (PA12)	(Initial value)
1	ATU-II input capture input/output compare output (TIO5A)	

- Bits 7 and 6—PA11 Mode Bits 1 and 0 (PA11MD1 and 0): Selects the function of pin PA11/TIO4D/ADTO1B.

Bit 7: PA11MD1	Bit 6: PA11MD0	Description	
0	0	General input/output (PA11)	(Initial value)
0	1	ATU-II input capture input/output compare output (TIO4D)	
1	0	Setting prohibited	
1	1	Output compare 1B output (MTAD)	

- Bits 5 and 4—PA10 Mode Bits 1 and 0 (PA10MD1 and 0): Selects the function of pin PA10/TIO4C/ADTO1A.

Bit 5: PA10MD1	Bit 4: PA10MD0	Description	
0	0	General input/output (PA10)	(Initial value)
0	1	ATU-II input capture input/output compare output (TIO4C)	
1	0	Setting prohibited	
1	1	Output compare 1A output (MTAD)	

- Bits 3 and 2—PA9 Mode Bits 1 and 0 (PA9MD1 and 0): Selects the function of pin PA9/TIO4B/ADTO0B.

Bit 3: PA9MD1	Bit 2: PA9MD0	Description
0	0	General input/output (PA9) (Initial value)
0	1	ATU-II input capture input/output compare output (TIO4B)
1	0	Setting prohibited
1	1	Output compare 0B output (MTAD)

- Bits 1 and 0—PA8 Mode Bits 1 and 0 (PA8MD1 and 0): Selects the function of pin PA8/TIO4A/ADTO0A.

Bit 1: PA8MD1	Bit 0: PA8MD0	Description
0	0	General input/output (PA8) (Initial value)
0	1	ATU-II input capture input/output compare output (TIO4A)
1	0	Setting prohibited
1	1	Output compare 0A output (MTAD)

### Port A Control Register L (PACRL)

Bit:	15	14	13	12	11	10	9	8
	—	PA7MD	—	PA6MD	—	PA5MD	—	PA4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PA3MD	—	PA2MD	—	PA1MD	—	PA0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PA7 Mode Bit (PA7MD): Selects the function of pin PA7/TIO3D.

Bit 14: PA7MD	Description
0	General input/output (PA7) (Initial value)
1	ATU-II input capture input/output compare output (TIO3D)



- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PA6 Mode Bit (PA6MD): Selects the function of pin PA6/TIO3C.

Bit 12: PA6MD	Description
0	General input/output (PA6) (Initial value)
1	ATU-II input capture input/output compare output (TIO3C)

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PA5 Mode Bit (PA5MD): Selects the function of pin PA5/TIO3B.

Bit 10: PA5MD	Description
0	General input/output (PA5) (Initial value)
1	ATU-II input capture input/output compare output (TIO3B)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PA4 Mode Bit (PA4MD): Selects the function of pin PA4/TIO3A.

Bit 8: PA4MD	Description
0	General input/output (PA4) (Initial value)
1	ATU-II input capture input/output compare output (TIO3A)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0. If 1 is written to this bit, correct operation cannot be guaranteed.
- Bit 6—PA3 Mode Bit (PA3MD): Selects the function of pin PA3/TIO3D.

Bit 6: PA3MD	Description
0	General input/output (PA3) (Initial value)
1	ATU-II input capture input (TIO3D)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0. If 1 is written to this bit, correct operation cannot be guaranteed.
- Bit 4—PA2 Mode Bit (PA2MD): Selects the function of pin PA2/TIO3C.

Bit 4: PA2MD	Description
0	General input/output (PA2) (Initial value)
1	ATU-II input capture input (TIO3C)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0. If 1 is written to this bit, correct operation cannot be guaranteed.
- Bit 2—PA1 Mode Bit (PA1MD): Selects the function of pin PA1/TI0B.

Bit 2: PA1MD	Description	
0	General input/output (PA1)	(Initial value)
1	ATU-II input capture input (TI0B)	

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0. If 1 is written to this bit, correct operation cannot be guaranteed.
- Bit 0—PA0 Mode Bit (PA0MD): Selects the function of pin PA0/TI0A.

Bit 0: PA0MD	Description	
0	General input/output (PA0)	(Initial value)
1	ATU-II input capture input (TI0A)	

### 18.4.3 I/O Port A

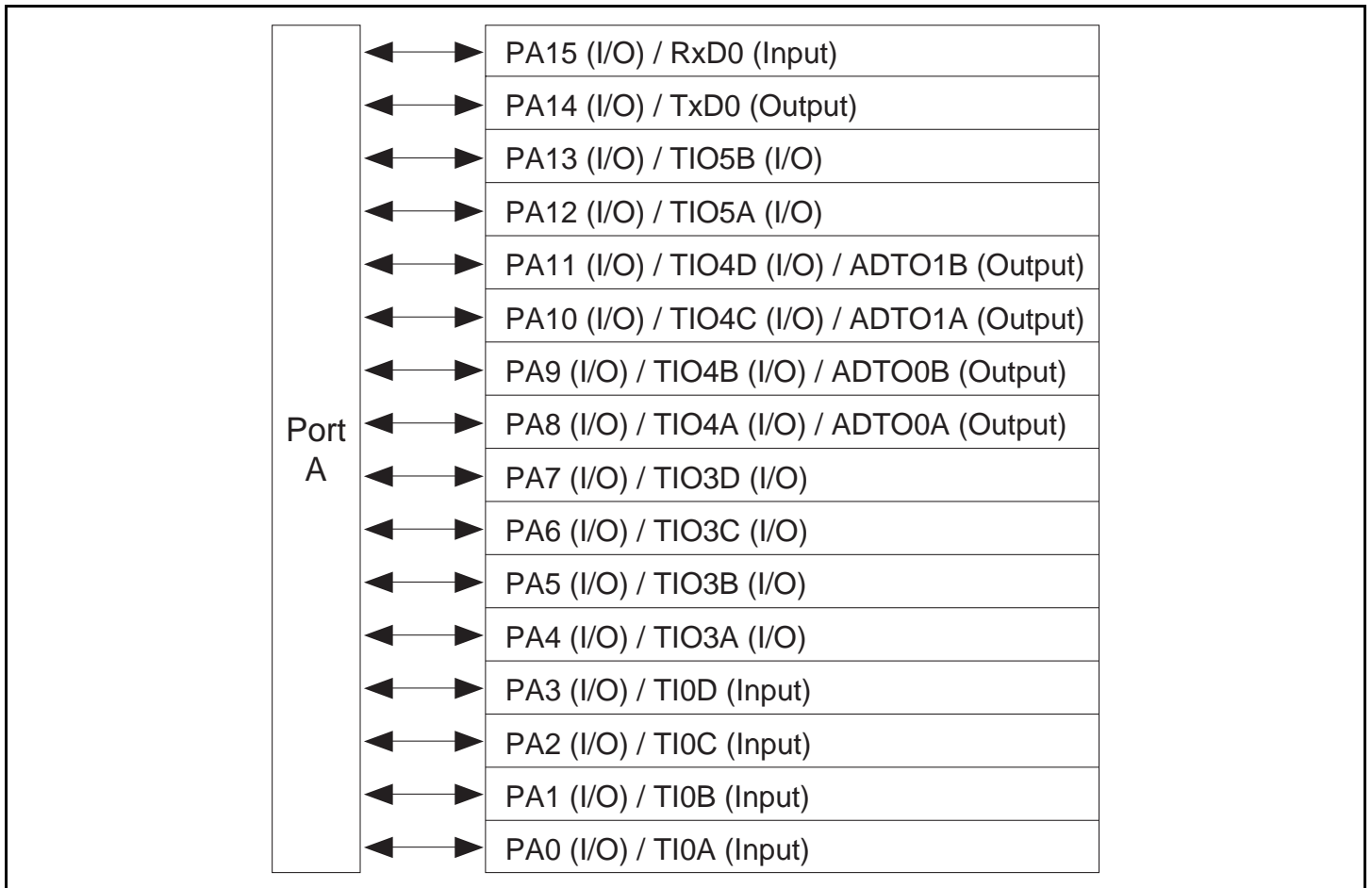


Figure 18.2 Port A

Table 18.4 lists the I/O port pins used for the multi-trigger A/D converter.

**Table 18.4 Pin Function (MTAD)**

Type	Symbol	Pin No.	I/O	Name	Function
Multi-trigger A/D	ADTO0A, ADTO0B, ADTO1A, ADTO1B	135-138	Output	PWM output	PWM output pins.

## 18.5 Operation

### 18.5.1 Overview

The multi-trigger A/D converter is divided into the timer parts and A/D conversion parts. The timer parts include two channels 0 and 1, each of which includes the prescaler that can generate or provide the selection of an input clock having the desired frequency. The following are general descriptions of the operations of the channels and prescalers.

#### (1) Channels 0 and 1

Channels 0 and 1 include 16-bit free running counters (ADCNT0 and ADCNT1), 16-bit cycle registers (ADCYLR0 and ADCYLR1), 16-bit duty registers (ADDR0A, ADDR0B, ADDR1A, and ADDR1B), and 16-bit general registers (ADGR0A, ADGR0B, ADGR1A, and ADGR1B), respectively. They also have external output pins of their own (ADTO0A, ADTO0B, ADTO1A, and ADTO1B), thus allowing the channels to be used as PWM timers.

ADCNT0 and ADCNT1, which are the incrementing counters, output 0 (1)\* to the external output pins when the counter value matches the ADDR value (when ADDR ≠ ADCYLR).

When the counter value matches the ADCYLR value (when ADDR ≠ H'0000), ADCNT0 and ADCNT1 output 1 (0)\* to the external output pins, simultaneously clearing the ADCNT value to H'0001. Due to these operations, channels 0 and 1 can output a waveform having the cycle specified by the ADCYLR value and the duty specified by the ADDR value.

When ADDR = ADCYLR, ADCNT0 and ADCNT1 output 1 (0)\* continuously to the external output pins, thus providing a 100%-duty waveform, and when ADDR = H'0000, these counters output 0 (1)\* continuously to the external output pins, thus providing a 0%-duty waveform.

Note that the ADDR value should never be greater than the ADCYLR value.

Channels 0 and 1 also perform the compare match operation when the ADCNT value matches the ADGR0A, ADGR0B, ADGR1A, or ADGR1B value that has been set in ADGR previously. However, no output pins are provided. The channels can also trigger multi-trigger A/D conversion using the compare matches. Neither ADCNT0 nor ADCNT1 is cleared when the value matches the ADGR0A, ADGR0B, ADGR1A, or ADGR1B value.

Note: \* Selected by the A/D trigger control register (ADTCR).

## (2) Prescalers

The channels incorporate dedicated prescalers, which can halt the clock signal that is input from the first stage or divide the frequency of the clock signal by 2, 5, or 10 according to the setting of the A/D trigger control register in the corresponding channels.

### 18.5.2 PWM Operation

Channels 0 and 1 can be unconditionally used as PWM timers using external output pins (ADTO0A, ADTO0B, ADTO1A, and ADTO1B).

When the prescaler is set using the A/D trigger control register (ADTCR) thus starting the free-running counter (ADCNT) in channels 0 and 1, the counters increment the count value until the value matches the value in the corresponding cycle register (ADCYLR). When the ADCNT value matches the ADCYLR value, the ADCNT value is cleared to H'0001, thus incrementing again from H'0001. Here, the corresponding pins output 1 (0)\*. When the appropriate value is set in the duty register (ADDR) and the ADCNT matches the ADDR value, the corresponding pins output 0 (1)\*. When the ADDR value is H'0000, the output does not change (0% duty). To obtain the 100% duty output, set the same values to the ADDR and ADCYLR. Note that the ADDR value should not be greater than the ADCYLR value.

Note: \* Selected by the DTSEL0A, DTSEL0B, DTSEL1A, and DTSEL1B bits in the A/D trigger control register (ADTCR).

### 18.5.3 Compare Match Operation

The A/D general registers (ADGR0A, ADGR0B, ADGR1A, and ADGR1B) in channels 0 and 1 can trigger the corresponding multi-trigger A/D converters.

When the A/D trigger control register (ADTCR) is set appropriately, the free-running counter (ADCNT) starts incrementing the count value. When the ADCNT value matches the ADGR value that has been set previously, the compare match is generated, requesting the corresponding multi-trigger A/D converter to start. However, no output pins are provided.

### 18.5.4 Multi-Trigger A/D Conversion Operation

The multi-trigger A/D conversion is the special conversion mode, in which A/D conversion on the special channels is performed prior to the other channels during continuous scan mode. When using the multi-trigger A/D conversion operation, only the settings shown below are possible for continuous scan mode; other settings are prohibited.

## Channels for Conversion (ADCSRx)

Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Analog Input Channels	
ADM1	ADM0	CH3	CH2	CH1	CH0	A/D0	A/D1
0	1	0	0	1	1	AN0 to AN3	AN12 to AN15
1	0	0	0	1	1	AN0 to AN7	AN12 to AN19
0	1	0	1	1	1	AN4 to AN7	AN16 to AN19

Note: x = 0 or 1.

Be sure to start multi-trigger A/D conversion only while the ADCMFxB and ADCMFxA bits in the A/D trigger status registers 0 and 1 (ADTSR0 and ADTSR1) are 0. When the multi-trigger A/D conversion is complete, clear these bits.

Multi-trigger A/D conversion can be enabled by setting the ADTRG in the A/D trigger interrupt enable registers 0 and 1 (ADTIER0 and ADTIER1) to 1.

Multi-trigger A/D conversion starts when the A/D counter (ADCNT) value matches the A/D general register (ADGR) value during scan mode on the specified channels while the ADTRG bit in the A/D trigger interrupt enable register (ADTIER) is 1. When the A/D conversion on the current channel in continuous scan mode is complete, the multi-trigger A/D conversion on the channels for which the conversion has been requested is performed prior to the other channels. When the multi-trigger A/D conversion on the channels for which the conversion has been requested is complete, the A/D conversion starts again on the channel that has been halted.

When the multi-trigger A/D conversion A (AN8, AN9, AN20, and AN21) and B (AN10, AN11, AN22, and AN23) on the channel for which the conversion has been requested is complete, the results are transferred to the appropriate ADDR in accordance with the setting of the A/D select bits (ADSEL) in the A/D trigger control register (ADTCR) at the start of the multi-trigger A/D conversion, thus setting the TADFxA and TADFxB bits in ADTSR to 1. Here, if the TADExA and TADExB bits in ADTIER are 1, the TADIxA and TADIxB interrupts are requested. To clear the TADFxA and TADFxB bits to 0, read these bits while they are 1, and write 0 to them.

An example of the operation when analog inputs 0 to 7 (AN0 to AN7) are selected; A/D conversion is performed in 8-channel scan mode; and A/D interrupt conversion is performed is described below. Figure 18.4 shows the operation flowchart for the example.

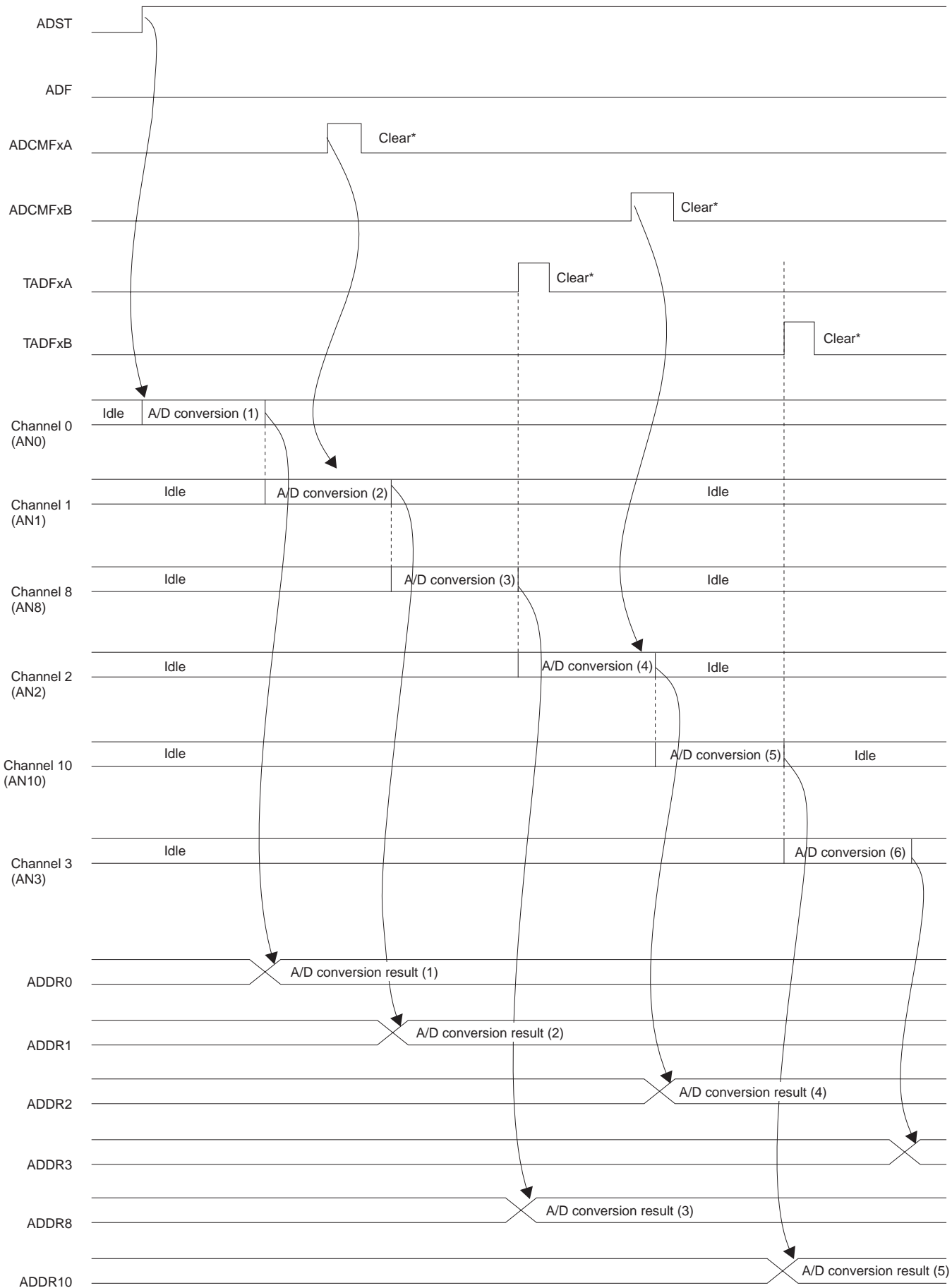
1. 8-channel scan mode is selected (ADM1 = 1 and ADM0 = 0), continuous scan mode is selected (ADCS = 1), analog input channels AN0 to AN7 are selected (CH3 = 0, CH2 = 0, CH1 = 1, and CH0 = 1), the A/D0 module is enabled for triggering of multi-trigger A/D

conversion (ADTRG = 1), multi-trigger A/D conversion end interrupt is enabled (TADExA and TADExB = 1), and A/D conversion is started.

2. When conversion of the first channel (AN0) is completed, the result is transferred to ADDR0.
3. Conversion proceeds in the same way through the eighth channel (AN7).
4. When conversion is completed for all the selected channels (AN0 to AN7), the ADF flag is set to 1. If the ADIE bit is 1 at the completion of conversion, an ADI interrupt is requested after A/D conversion ends.
5. If the A/D counter (ADCNT) and A/D general register (ADGR) values match during conversion of AN0 to AN7, the multi-trigger A/D conversion on the channels for which the conversion has been requested is started after A/D conversion of the current channel ends.
6. When the multi-trigger A/D conversion on the channels for which the conversion has been requested is completed, the result is transferred to ADDR<sub>x</sub> and the A/D data select (ADSEL<sub>x</sub>) is inverted. If the TADIExA or TADIExB is 1 at the completion of multi-trigger A/D conversion, a TADIA or TADIB interrupt of the completed channel is requested.
7. After step 6, the A/D conversion starts again on the channel that has been halted. While ADST is 1, steps 2 to 7 are repeated.

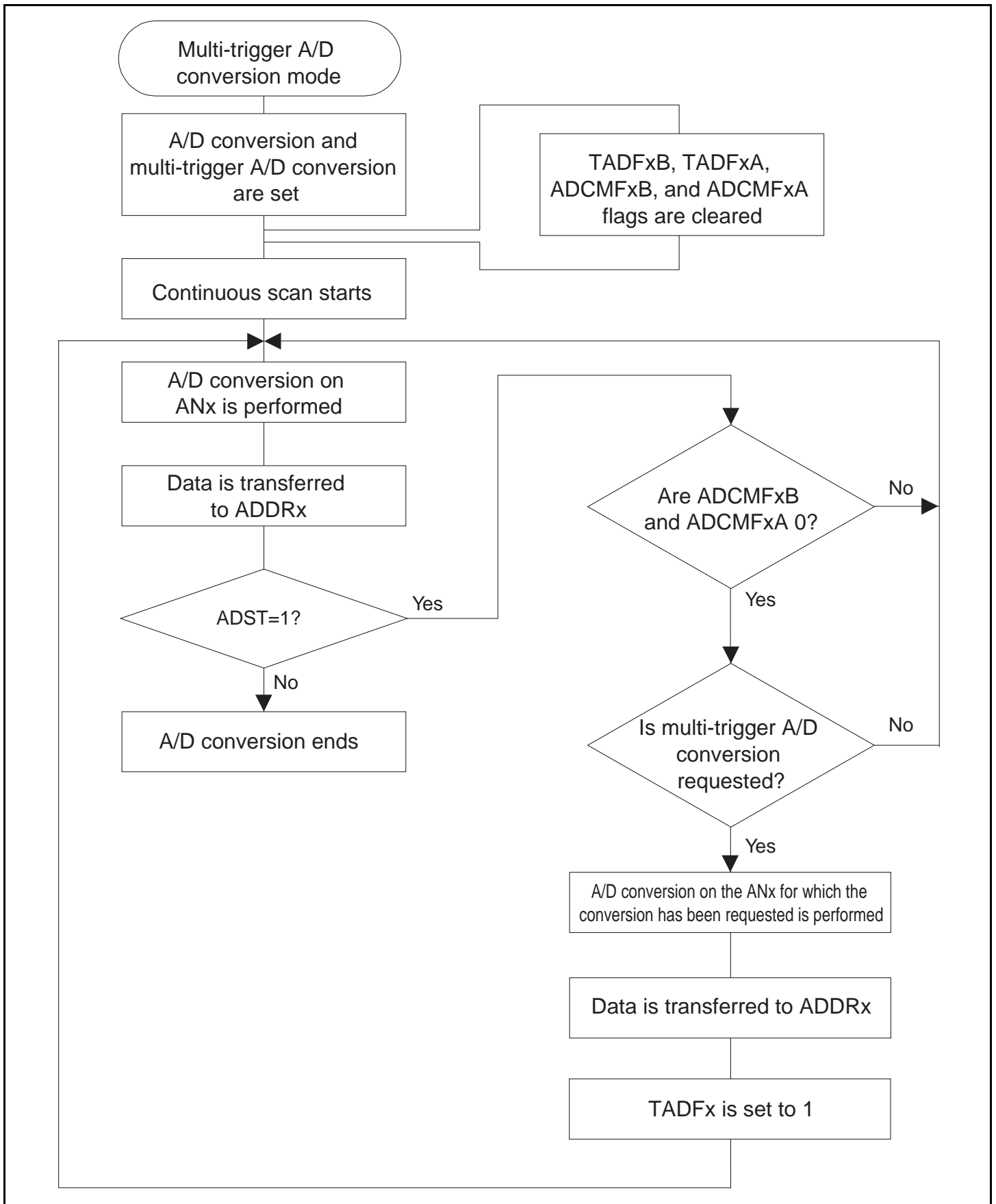
Note: When multi-trigger A/D conversion is requested simultaneously from two sources, conversion is performed according to the priority.

<b>Priority high</b>		<b>Priority low</b>
CMFxA	>	CMFxB



Note: \* Instructions executed by software are indicated.

**Figure 18.3 Example of Multi-Trigger A/D Converter Operation**



**Figure 18.4 Flowchart of Multi-Trigger A/D Converter Operation**



## 18.5.5 Interrupts

Each of channels 0 and 1 generate interrupts from seven sources, that is, a total of 14 sources listed below.

Module	IPR Bit	Vector	Vector Number	Conditions of Interrupt Generation
ADT0	IPRJ (11 to 8)	ADI0	189	Multi-trigger A/D conversion ends when the interrupt is enabled by TADE0A
				Multi-trigger A/D conversion ends when the interrupt is enabled by TADE0B
				ADCNT0 matches ADCYLR0 when the interrupt is enabled by CYE0
				ADCNT0 matches ADDR0A when the interrupt is enabled by ADDE0A
				ADCNT0 matches ADDR0B when the interrupt is enabled by ADDE0B
				ADCNT0 matches ADGR0A when the interrupt is enabled by ADCME0A
				ADCNT0 matches ADGR0B when the interrupt is enabled by ADCME0B

Module	IPR Bit	Vector	Vector Number	Conditions of Interrupt Generation
ADT1	IPRJ (7 to 4)	ADI1	193	Multi-trigger A/D conversion ends when the interrupt is enabled by TADE1A
				Multi-trigger A/D conversion ends when the interrupt is enabled by TADE1B
				ADCNT1 matches ADCYLR1 when the interrupt is enabled by CYE1
				ADCNT1 matches ADDR1A when the interrupt is enabled by ADDE1A
				ADCNT1 matches ADDR1B when the interrupt is enabled by ADDE1B
				ADCNT1 matches ADGR1A when the interrupt is enabled by ADCME1A
				ADCNT1 matches ADGR1B when the interrupt is enabled by ADCME1B

## 18.5.6 Usage Notes

1. When a conflict occurs between a write to ADCNT and clearing of the counter by a compare match  
When a compare match occurs during T2 state of a CPU cycle for writing to ADCNT, ADCNT is not cleared but is written to.  
However, a compare match remains effective, thus allowing a write of 1 to the interrupt status flag and external waveform output, similar to regular compare matches.
2. When a conflict occurs between a write to ADCNT and incrementing of the counter The counter is not incremented but is written to.
3. When a conflict occurs between clearing of the interrupt status flag and setting of the flag by interrupt generation  
When any event, such as a compare match and overflow, occurs during T2 state of a CPU cycle for writing 0 to the interrupt status flag, the compare match takes priority thus allowing the interrupt status flag to be set.
4. When reading the continuous scan A/D conversion data during the multi-trigger A/D conversion is performed  
Reading is performed by the DMA. Following errors are generated according to the interrupt timing.  
When reading ADDR of the first channel by the continuous scan interrupt, if MTAD is executed on the last channel in the previous scan, the data may be overwritten again in this scan because the first channel is converted.

## 18.5.7 Operation Waveform Examples

(A)

### Hardware Operation

1. A compare match occurs, setting the status flag to the corresponding source.
2. Multi-trigger A/D conversion that is enabled by A/D trigger (ADTRG) in the A/D trigger interrupt enable register (ADTIER) starts.

### After Multi-trigger A/D conversion is Over

3. Multi-trigger A/D conversion result is transferred to the register that is specified by A/D select (ADSEL) in the A/D trigger control register (ADTCR) at the start of the conversion.
4. An interrupt is generated if the multi-trigger A/D conversion end interrupt is enabled.

## Software Operation

1. A compare match flag is cleared.
2. The value in the A/D general register (ADGR) is changed.
3. A/D select (ADSEL) in the A/D trigger control register (ADTCR) is changed.

## After Multi-trigger A/D conversion is Over

4. The multi-trigger A/D conversion end flag is cleared.
5. The conversion result is read out.

(B)

## Hardware Operation

1. A compare match occurs, setting the status flag to the corresponding source.
2. An interrupt is generated if the A/D duty enable bit (ADDE) in the A/D trigger interrupt enable register (ADTIER) is set.
3. The level of the external output pin is changed.

## Software Operation

1. The duty compare match flag is cleared.

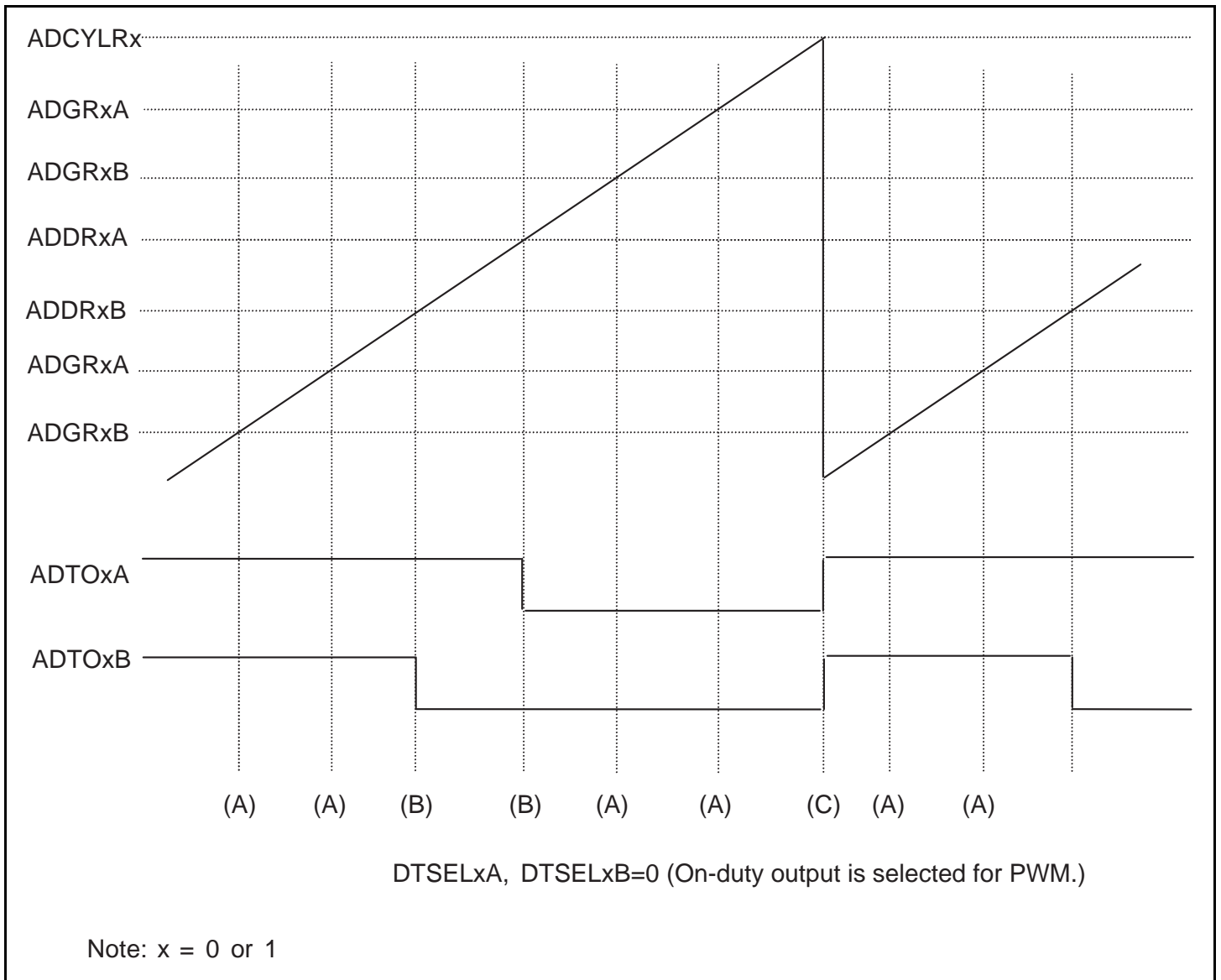
(C)

## Hardware Operation

1. A compare match occurs, setting the status flag to the corresponding source.
2. An interrupt is generated if the A/D cycle enable bit (ADCYLR) in the A/D trigger interrupt enable register (ADTIER) is set.
3. The level of the external output pin is changed.

## Software Operation

1. The cycle compare match flag is cleared.
2. The values in the A/D duty register (ADDR) and the A/D cycle register (ADCYLR) are changed.



**Figure 18.5 Example of Output Waveform from MTAD PWM**

## 18.6 Appendices

### 18.6.1 On-Chip Peripheral Module Registers

#### (1) Address

On-chip peripheral module register addresses and bit names related to the multi-trigger A/D are shown in table 18.5. 16-bit and 32-bit registers are shown in two and four rows of 8 bits, respectively.

**Table 18.5 Address**

Address	Register Abbr.	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF720	PAIOR	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR	Port A
H'FFFFFF721		PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR	
H'FFFFFF722	PACRH	—	PA15MD	—	PA14MD	—	PA13MD	—	PA12MD	
H'FFFFFF723		<u>PM11MD1</u>	PA11MD0	<u>PM10MD1</u>	PA10MD0	<u>PM9MD1</u>	PA9MD0	<u>PM8MD1</u>	PA8MD0	
H'FFFFFF724	PACRL	—	PA7MD	—	PA6MD	—	PA5MD	—	PA4MD	
H'FFFFFF725		—	PA3MD	—	PA2MD	—	PA1MD	—	PA0MD	
H'FFFFFF726	PADR	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	
H'FFFFFF727		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	

#### (2) Register States in Reset and Power-Down States

**Table 18.6 Register States in Reset and Power-Down States**

Type	Name	Reset State	Power-Down State		
		Power-On	Hardware Standby	Software Standby	Sleep
Multi-trigger A/D (MTAD)	ADTCR0, ADTCR1	Initialized	Initialized	Initialized	Held
	ADTSR0, ADTSR1				
	ADTIER0, ADTIER1				
	ADCNT0, ADCNT1				
	ADGR0A, ADGR0B				
	ADGR1A, ADGR1B				
	ADCYLR0, ADCYLR1				
	ADDR0A, ADDR0B				
	ADDR1A, ADDR1B				

## 18.6.2 Pin States

**Table 18.7 Pin States**

Type	Pin Name	Pin State							
		Reset State			Power-Down State				
		Power-On			Hardware Standby	Software Standby	H-UDI Module Standby	AUD Module Standby	Bus-Released State
		ROMless Expanded Mode		Expanded Mode with ROM					
8 Bits	16 Bits								
MTAD	ADTO0A	—			Z	O*	O	O	O
	ADTO0B	—			Z	O*	O	O	O
	ADTO1A	—			Z	O*	O	O	O
	ADTO1B	—			Z	O*	O	O	O

— : Not initial value

I : Input

O : Output

H : High-level output

L : Low-level output

Z : High impedance

K : Input pins become high-impedance, output pins retain their state.

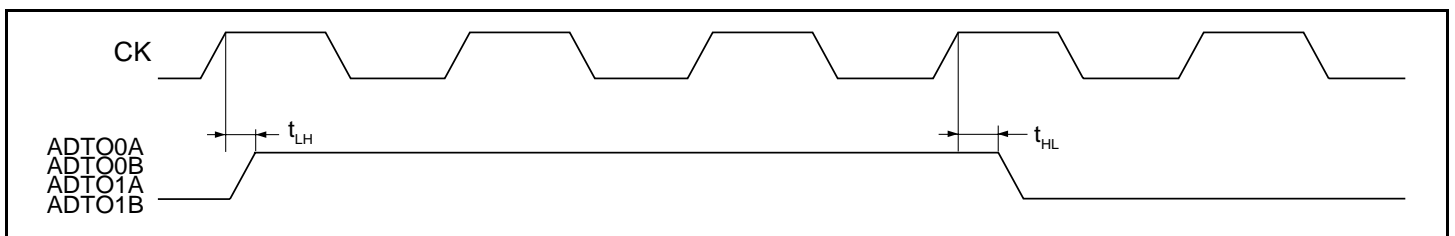
Notes: When the CKHIZ bit in PFCRH is set to 1, becomes high-impedance unconditionally.

\* When the port impedance bit (HIZ) in the standby control register (SBYCR) is set to 1, output pins become high-impedance.

## 18.6.3 AC Characteristics

**Table 18.8 Output Timing of ADTO0A, ADTO0B, ADTO1A, and ADTO1B**

Item	Symbol	min.	max.	Unit	Figures
Delay time	$t_{LH}$	—	100	ns	Figure 18.6
	$t_{HL}$	—	100	ns	Figure 18.6



**Figure 18.6 Output Timing of ADTO0A, ADTO0B, ADTO1A, and ADTO1B**



# Section 19 High-performance User Debug Interface (H-UDI)

## 19.1 Overview

The High-performance user debug interface (H-UDI) provides data transfer, interrupt request, and boundary scan functions. The H-UDI performs serial transfer by means of external signal control.

### 19.1.1 Features

The H-UDI has the following features conforming to the IEEE 1149.1 standard.

- Five test signals (TCK, TDI, TDO, TMS, and  $\overline{\text{TRST}}$ )
- TAP controller
- Instruction register
- Data register
- Bypass register
- Boundary scan register

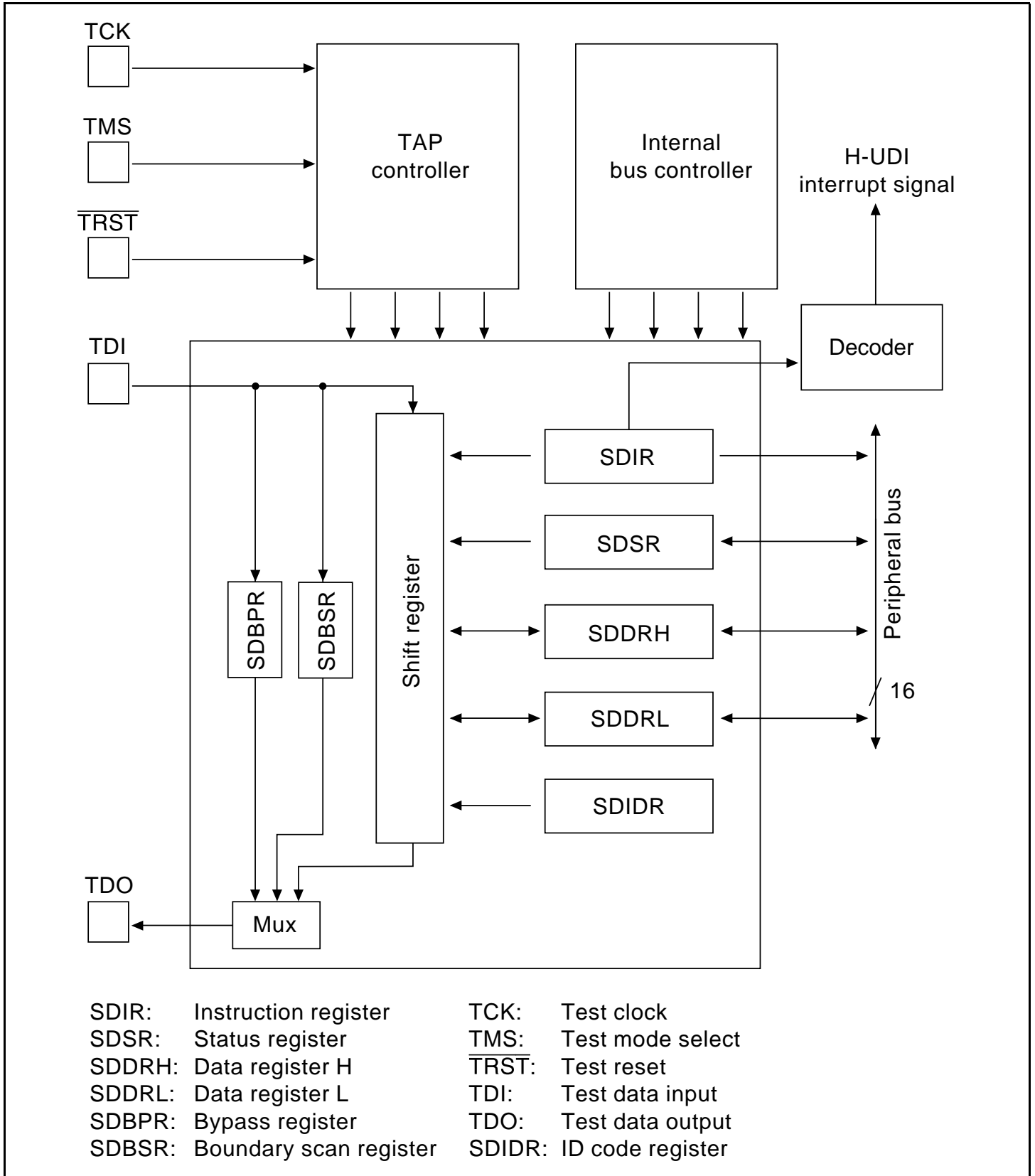
The H-UDI has seven instructions.

- BYPASS mode  
Test mode conforming to IEEE 1149.1
- EXTEST mode  
Test mode conforming to IEEE1149.1.
- SAMPLE/PRELOAD mode  
Test mode conforming to IEEE1149.1.
- CLAMP mode  
Test mode conforming to IEEE1149.1.
- HIGHZ mode  
Test mode conforming to IEEE1149.1.
- IDCODE mode  
Test mode conforming to IEEE1149.1.
- H-UDI interrupt  
H-UDI interrupt request to INTC



## 19.1.2 H-UDI Block Diagram

Figure 19.1 shows a block diagram of the H-UDI.



**Figure 19.1 H-UDI Block Diagram**

### 19.1.3 Pin Configuration

Table 19.1 shows the H-UDI pin configuration.

**Table 19.1 Pin Configuration**

Pin Name	Abbreviation	I/O	Function
Test clock	TCK	Input	Test clock input
Test mode select	TMS	Input	Test mode select input signal
Test data input	TDI	Input	Serial data input
Test data output	TDO	Output	Serial data output
Test reset	$\overline{\text{TRST}}$	Input	Test reset input signal

### 19.1.4 Register Configuration

Table 19.2 shows the H-UDI registers.

**Table 19.2 Register Configuration**

Register	Abbreviation	R/W* <sup>1</sup>	Initial Value* <sup>2</sup>	Address	Access Size (Bits)
Instruction register	SDIR	R	H'E000	H'FFFFFF7C0	8/16/32
Status register	SDSR	R/W	H'0B01	H'FFFFFF7C2	8/16/32
Data register H	SDDRH	R/W	Undefined	H'FFFFFF7C4	8/16/32
Data register L	SDDRL	R/W	Undefined	H'FFFFFF7C6	8/16/32
Bypass register	SDBPR	—	—	—	—
Boundary scan register	SDBSR	—	—	—	—
ID code register	SDIDR	—	H'001D200F	—	—

Notes: 1. Indicates whether the register can be read from/written to by the CPU.

2. Initial value when the  $\overline{\text{TRST}}$  signal is input. Registers are not initialized by a reset (power-on or manual) or in standby mode.

Instructions and data can be input to the instruction register (SDIR) and data register (SDDR) by serial transfer from the test data input pin (TDI). Data from SDIR, the status register (SDSR), and SDDR can be output via the test data output pin (TDO). The bypass register (SDBPR) is a 1-bit register to which TDI and TDO are connected in BYPASS, CLAMP, or HIGHZ mode. The boundary scan register (SDBSR) is a 474-bit register, and is connected to TDI and TDO in the SAMPLE/PRELOAD or EXTEST mode. The ID code register (SDIDR) is a 32-bit register; a fixed code can be output via TDO in the IDCODE mode. All registers, except SDBPR, SDBSR, and SDIDR, can be accessed from the CPU.

Table 19.3 shows the kinds of serial transfer possible with each register.

**Table 19.3 H-UDI Register Serial Transfer**

<b>Register</b>	<b>Serial Input</b>	<b>Serial Output</b>
SDIR	Possible	Possible
SDSR	Impossible	Possible
SDDRH	Possible	Possible
SDDRL	Possible	Possible
SDBPR	Possible	Possible
SDBSR	Possible	Possible
SDIDR	Impossible	Possible

## 19.2 External Signals

### 19.2.1 Test Clock (TCK)

The test clock pin (TCK) provides an independent clock supply to the H-UDI. As the clock input to TCK is supplied directly to the H-UDI, a clock waveform with a duty cycle close to 50% should be input (for details, see section 27, Electrical Characteristics). If no signal is input, TCK is fixed at 1 by internal pull-up.

### 19.2.2 Test Mode Select (TMS)

The test mode select pin (TMS) is sampled at the rise of TCK. TMS controls the internal state of the TAP controller. If no signal is input, TMS is fixed at 1 by internal pull-up.

### 19.2.3 Test Data Input (TDI)

The test data input pin (TDI) performs serial input of instructions and data for H-UDI registers. TDI is sampled at the rise of TCK. If no signal is input, TDI is fixed at 1 by internal pull-up.

### 19.2.4 Test Data Output (TDO)

The test data output pin (TDO) performs serial output of instructions and data from H-UDI registers. Transfer is performed in synchronization with TCK. If there is no output, TDO goes to the high-impedance state.

## 19.2.5 Test Reset ( $\overline{\text{TRST}}$ )

The test reset pin ( $\overline{\text{TRST}}$ ) initializes the H-UDI asynchronously. If no signal is input,  $\overline{\text{TRST}}$  is fixed at 1 by internal pull-up.

## 19.3 Register Descriptions

### 19.3.1 Instruction Register (SDIR)

Bit:	15	14	13	12	11	10	9	8
	TS3	TS2	TS1	TS0	—	—	—	—
Initial value:	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

The instruction register (SDIR) is a 16-bit register that can only be read by the CPU. H-UDI instructions can be transferred to SDIR by serial input from TDI. SDIR can be initialized by the  $\overline{\text{TRST}}$  signal, but is not initialized by a reset or in software standby mode.

SDIR defines four valid bits for instruction. If an instruction exceeding four bits is input, the last four bits of the serial data will be stored in SDIR.

Operation is not guaranteed if a reserved instruction is set in this register.

Bits 15 to 12—Test Set Bits (TS3–TS0): Table 19.4 shows the instruction configuration.

**Table 19.4 Instruction Configuration**

<b>Bit 15: TS3</b>	<b>Bit 14: TS2</b>	<b>Bit 13: TS1</b>	<b>Bit 12: TS0</b>	<b>Description</b>	
0	0	0	0	EXTEST mode	
			1	Reserved	
		1	0	CLAMP mode	
			1	HIGHZ mode	
	1	0	0	SAMPLE/PRELOAD mode	
			1	Reserved	
			1	Reserved	
		1	1	0	Reserved
				1	Reserved
				1	Reserved
1	0	0	0	Reserved	
			1	Reserved	
		1	0	H-UDI interrupt	
			1	Reserved	
		1	0	0	Reserved
				1	Reserved
	1		0	IDCODE mode (Initial value)	
			1	BYPASS mode	

Bits 11 to 0—Reserved: These bits are always read as 0. The write value should always be 0.

### 19.3.2 Status Register (SDSR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	1	0	1	1
R/W:	R	R	R	R	R	R	R	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SDTRF
Initial value:	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W

The status register (SDSR) is a 16-bit register that can be read from and written to by the CPU. SDCR output from TDO is possible, but serial data cannot be written to SDCR via TDI. The SDTRF bit is output by means of a 1-bit shift. In the case of a 2-bit shift, the SDTRF bit is first output, followed by a reserved bit.

SDSR is initialized by  $\overline{\text{TRST}}$  signal input, but is not initialized by a reset or in software standby mode.

Bits 15 to 1—Reserved: Bits 15 to 12 and 7 to 1 are always read as 0, and the write value should always be 0. Bit 11, 9, and 8 are always read as 1, and the write value should always be 1.

Bit 0—Serial Data Transfer Control Flag (SDTRF): Indicates whether H-UDI registers can be accessed by the CPU. The SDTRF bit is reset by the  $\overline{\text{TRST}}$  signal, but is not initialized by a reset or in software standby mode.

Bit 0: SDTRF	Description
0	Serial transfer to SDDR has ended, and SDDR can be accessed
1	Serial transfer to SDDR in progress (Initial value)

### 19.3.3 Data Register (SDDR)

The data register (SDDR) comprises data register H (SDDRH) and data register L (SDDRL), each of which has the following configuration.

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

SDDRH and SDDRL are 16-bit registers that can be read from and written to by the CPU. SDDR is connected to TDO and TDI for serial data transfer to and from an external device.

32-bit data is input and output in serial data transfer. If data exceeding 32 bits is input, only the last 32 bits will be stored in SDDR. Serial data is input starting from the MSB of SDDR (bit 15 of SDDRH), and output starting from the LSB (bit 0 of SDDRL).

This register is not initialized by a reset, in hardware or software standby mode, or by the  $\overline{\text{TRST}}$  signal.

### 19.3.4 Bypass Register (SDBPR)

The bypass register (SDBPR) is a 1-bit shift register. In BYPASS, CLAMP, or HIGHZ mode, SDBPR is connected between TDI and TDO. SDBPR cannot be read or written to by the CPU.

### 19.3.5 Boundary scan register (SDBSR)

The boundary scan register (SDBSR), a shift register that controls the I/O pins of this LSI, is provided on the PAD.

Using the EXTEST mode or the SAMPLE/PRELOAD mode, a boundary scan test conforming to the IEEE1149.1 standard can be performed.

For SDBSR, read/write by the CPU cannot be performed.

Table 19.5 shows the relationship between the pins of the LSI and the boundary scan register.

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits**

Pin No.	Pin Name	Input/Output	Bit No.
from TDI			
238	AUDRST	Input	473
240	AUDMD	Input	472
241	AUDATA0	Input	471
		Output	470
		Output enable	469
242	AUDATA1	Input	468
		Output	467
		Output enable	466
243	AUDATA2	Input	465
		Output	464
		Output enable	463
244	AUDATA3	Input	462
		Output	461
		Output enable	460
245	AUDCK	Input	459
		Output	458
		Output enable	457
246	AUDSYNC	Input	456
		Output	455
		Output enable	454
248	PD0/TOP1A	Input	453
		Output	452
		Output enable	451
250	PD1/TIO1B	Input	450
		Output	449
		Output enable	448
251	PD2/TIO1C	Input	447
		Output	446
		Output enable	445



**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
252	PD3/TIO1D	Input	444
		Output	443
		Output enable	442
253	PD4/TIO1E	Input	441
		Output	440
		Output enable	439
254	PD5/TIO1F	Input	438
		Output	437
		Output enable	436
255	PD6/TIO1G	Input	435
		Output	434
		Output enable	433
256	PD4/TIO1H	Input	432
		Output	431
		Output enable	430
1	PD8/PULS0	Input	429
		Output	428
		Output enable	427
2	PD9/PULS1	Input	426
		Output	425
		Output enable	424
3	PD10/PULS2	Input	423
		Output	422
		Output enable	421
4	PD11/PULS3	Input	420
		Output	419
		Output enable	418
5	PD12/PULS4	Input	417
		Output	416
		Output enable	415

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
6	PD13/PULS6/ HTxD0/HTxD1	Input	414
		Output	413
		Output enable	412
7	PE0/A0	Input	411
		Output	410
		Output enable	409
8	PE1/A1	Input	408
		Output	407
		Output enable	406
9	PE2/A2	Input	405
		Output	404
		Output enable	403
10	PE3/A3	Input	402
		Output	401
		Output enable	400
12	PE4/A4	Input	399
		Output	398
		Output enable	397
14	PE5/A5	Input	396
		Output	395
		Output enable	394
15	PE6/A6	Input	393
		Output	392
		Output enable	391
16	PE4/A7	Input	390
		Output	389
		Output enable	388
17	PE8/A8	Input	387
		Output	386
		Output enable	385

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
18	PE9/A9	Input	384
		Output	383
		Output enable	382
19	PE10/A10	Input	381
		Output	380
		Output enable	379
21	PE11/A11	Input	378
		Output	377
		Output enable	376
23	PE12/A12	Input	375
		Output	374
		Output enable	373
24	PE13/A13	Input	372
		Output	371
		Output enable	370
25	PE14/A14	Input	369
		Output	368
		Output enable	367
26	PE15/A15	Input	366
		Output	365
		Output enable	364
27	PF0/A16	Input	363
		Output	362
		Output enable	361
28	PF1/A17	Input	360
		Output	359
		Output enable	358
29	PF2/A18	Input	357
		Output	356
		Output enable	355

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
31	PF3/A19	Input	354
		Output	353
		Output enable	352
33	PF4/A20	Input	351
		Output	350
		Output enable	349
34	PF5/A21/ $\overline{\text{POD}}$	Input	348
		Output	347
		Output enable	346
35	PF6/ $\overline{\text{WRL}}$	Input	345
		Output	344
		Output enable	343
36	PF7/ $\overline{\text{WRH}}$	Input	342
		Output	341
		Output enable	340
37	PF8/ $\overline{\text{WAIT}}$	Input	339
		Output	338
		Output enable	337
38	PF9/ $\overline{\text{RD}}$	Input	336
		Output	335
		Output enable	334
40	PF10/ $\overline{\text{CS0}}$	Input	333
		Output	332
		Output enable	331
42	PF1/ $\overline{\text{CS1}}$	Input	330
		Output	329
		Output enable	328
43	PF12/ $\overline{\text{CS2}}$	Input	327
		Output	326
		Output enable	325

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
44	PF13/ $\overline{\text{CS3}}$	Input	324
		Output	323
		Output enable	322
45	PF14/ $\overline{\text{BACK}}$	Input	321
		Output	320
		Output enable	319
46	PF15/ $\overline{\text{BREQ}}$	Input	318
		Output	317
		Output enable	316
50	MD2	Input	315
55	MD1	Input	314
56	FWE	Input	313
59	MD0	Input	312
63	PH0/D0	Input	311
		Output	310
		Output enable	309
64	PH1/D1	Input	308
		Output	307
		Output enable	306
65	PH2/D2	Input	305
		Output	304
		Output enable	303
66	PH3/D3	Input	302
		Output	301
		Output enable	300
67	PH4/D4	Input	299
		Output	298
		Output enable	297
68	PH5/D5	Input	296
		Output	295
		Output enable	294

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
69	PH6/D6	Input	293
		Output	292
		Output enable	291
71	PH7/D7	Input	290
		Output	289
		Output enable	288
73	PH8/D8	Input	287
		Output2	286
		Output enable	285
74	PH9/D9	Input	284
		Output	283
		Output enable	282
76	PH10/D10	Input	281
		Output	280
		Output enable	279
78	PH11/D11	Input	278
		Output	277
		Output enable	276
79	PH12/D12	Input	275
		Output	274
		Output enable	273
80	PH13/D13	Input	272
		Output	271
		Output enable	270
81	PH14/D14	Input	269
		Output	268
		Output enable	267
82	PH15/D15	Input	266
		Output	265
		Output enable	264

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
84	NMI	Input	263
124	$\overline{\text{WDTOVF}}$	Output	262
		Output enable	261
125	PA0/TI0A	Input	260
		Output	259
		Output enable	258
127	PA1/TI0B	Input	257
		Output	256
		Output enable	255
129	PA2/TI0C	Input	254
		Output	253
		Output enable	252
130	PA3/TI0D	Input	251
		Output	250
		Output enable	249
131	PA4/TIO3A	Input	248
		Output	247
		Output enable	246
132	PA5/TIO3B	Input	245
		Output	244
		Output enable	243
133	PA6/TIO3C	Input	242
		Output	241
		Output enable	240
134	PA7/TIO3D	Input	239
		Output	238
		Output enable	237
135	PA8/TIIO4A	Input	236
		Output	235
		Output enable	234

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
136	PA9/TIO4B	Input	233
		Output	232
		Output enable	231
137	PA10/TIO4C	Input	230
		Output	229
		Output enable	228
138	PA11/TIO4D	Input	227
		Output	226
		Output enable	225
140	PA12/TIO5A	Input	224
		Output	223
		Output enable	222
142	PA13/TIO5B	Input	221
		Output	220
		Output enable	219
143	PA14/TxD0	Input	218
		Output	217
		Output enable	216
144	PA15/RxD0	Input	215
		Output	214
		Output enable	213
145	PB0/TO6A	Input	212
		Output	211
		Output enable	210
146	PB1/TO6B	Input	209
		Output	208
		Output enable	207
147	PB2/TO6C	Input	206
		Output	205
		Output enable	204



**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
149	PB3/TO6D	Input	203
		Output	202
		Output enable	201
151	PB4/TO7A/TO8A	Input	200
		Output	199
		Output enable	198
152	PB5/TO7B/TO8B	Input	197
		Output	196
		Output enable	195
153	PB6/TO7C/TO8C	Input	194
		Output	193
		Output enable	192
154	PB7/TO7D/TO8D	Input	191
		Output	190
		Output enable	189
155	PD8/TxD3/TO8E	Input	188
		Output	187
		Output enable	186
156	PB9/RxD3/TO8F	Input	185
		Output	184
		Output enable	183
157	PB10/TxD4/HTxD0 /TO8G	Input	182
		Output	181
		Output enable	180
158	PB11/RxD4/HRxD0 /TO8H	Input	179
		Output	178
		Output enable	177
159	PB12/TCLKA /UBCTRG	Input	176
		Output	175
		Output enable	174

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
160	PB13/SCK0	Input	173
		Output	172
		Output enable	171
162	PB14/SCK1/ TCLKB/TI10	Input	170
		Output	169
		Output enable	168
164	PB15/PULS5/ SCK2	Input	167
		Output	166
		Output enable	165
165	PC0/TxD1	Input	164
		Output	163
		Output enable	162
166	PC1/RxD1	Input	161
		Output	160
		Output enable	159
167	PC2/TxD2	Input	158
		Output	157
		Output enable	156
168	PC3/RxD2	Input	155
		Output	154
		Output enable	153
169	PC4/ $\overline{\text{IRQ0}}$	Input	152
		Output	151
		Output enable	150
170	PG0/PULS7/ HRxD0/HRxD1	Input	149
		Output	148
		Output enable	147
171	PG1/ $\overline{\text{IRQ1}}$	Input	146
		Output	145
		Output enable	144

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
173	PG2/ $\overline{\text{IRQ2}}$ /ADEND	Input	143
		Output	142
		Output enable	141
175	PG3/ $\overline{\text{IRQ3}}$ / ADTRG0	Input	140
		Output	139
		Output enable	138
176	PJ0/TIO2A	Input	137
		Output	136
		Output enable	135
177	PJ1/TIO2B	Input	134
		Output	133
		Output enable	132
178	PJ2/TIO2C	Input	131
		Output	130
		Output enable	129
179	PJ3/TIO2D	Input	128
		Output	127
		Output enable	126
180	PJ4/TIO2E	Input	125
		Output	124
		Output enable	123
181	PJ5/TIO2F	Input	122
		Output	121
		Output enable	120
182	PJ6/TIO2G	Input	119
		Output	118
		Output enable	117
183	PJ7/TIO2H	Input	116
		Output	115
		Output enable	114

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
184	PJ8/TIO5C	Input	113
		Output	112
		Output enable	111
186	PJ9/TIO5D	Input	110
		Output	109
		Output enable	108
188	PJ10/TI9A	Input	107
		Output	106
		Output enable	105
189	PJ11/TI9B	Input	104
		Output	103
		Output enable	102
190	PJ12/TI9C	Input	101
		Output	100
		Output enable	99
191	PJ13/TI9D	Input	98
		Output	97
		Output enable	96
192	PJ14/TI9E	Input	95
		Output	94
		Output enable	93
193	PJ15/TI9F	Input	92
		Output	91
		Output enable	90
195	PK0/TO8A	Input	89
		Output	88
		Output enable	87
197	PK1/TO8B	Input	86
		Output	85
		Output enable	84

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
198	PK2/TO8C	Input	83
		Output	82
		Output enable	81
199	PK3/TO8D	Input	80
		Output	79
		Output enable	78
200	PK4/TO8E	Input	77
		Output	76
		Output enable	75
201	PK5/TO8F	Input	74
		Output	73
		Output enable	72
202	PK6/TO8G	Input	71
		Output	70
		Output enable	69
204	PK7/TO8H	Input	68
		Output	67
		Output enable	66
206	PK8/TO8I	Input	65
		Output	64
		Output enable	63
207	PK9/TO8J	Input	62
		Output	61
		Output enable	60
208	PK10/TO8K	Input	59
		Output	58
		Output enable	57
209	PK11/TO8L	Input	56
		Output	55
		Output enable	54

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
210	PK12/TO8M	Input	53
		Output	52
		Output enable	51
211	PK13/TO8N	Input	50
		Output	49
		Output enable	48
213	PK14/TO8O	Input	47
		Output	46
		Output enable	45
215	PK15/TO8P	Input	44
		Output	43
		Output enable	42
216	PL0/TI10	Input	41
		Output	40
		Output enable	39
217	PL1/TIO11A/ $\overline{\text{IRQ6}}$	Input	38
		Output	37
		Output enable	36
218	PL2/TIO11B/ $\overline{\text{IRQ7}}$	Input	35
		Output	34
		Output enable	33
219	PL3/TCLKB	Input	32
		Output	31
		Output enable	30
220	PL4/ADTRG0	Input	29
		Output	28
		Output enable	27
221	PL5/ADTRG1	Input	26
		Output	25
		Output enable	24

**Table 19.5 Correspondence between Pins and Boundary Scan Register Bits (cont)**

<b>Pin No.</b>	<b>Pin Name</b>	<b>Input/Output</b>	<b>Bit No.</b>
222	PL6/ADEND	Input	23
		Output	22
		Output enable	21
223	PL7/SCK2	Input	20
		Output	19
		Output enable	18
224	PL8/SCK3	Input	17
		Output	16
		Output enable	15
226	PL9/SCL4/ $\overline{\text{IRQ5}}$	Input	14
		Output	13
		Output enable	12
228	PL10/HTxD0/ HTxD1/HTxD0& HTxD1	Input	11
		Output	10
		Output enable	9
229	PL11/HRxD0/ HRxD1/HRxD0& HRxD1	Input	8
		Output	7
		Output enable	6
230	PL12/ $\overline{\text{IRQ4}}$	Input	5
		Output	4
		Output enable	3
231	PL13/ $\overline{\text{IRQOUT}}$	Input	2
		Output	1
		Output enable	0

to TDO

### 19.3.6 ID code register (SDIDR)

The ID code register (SDIDR) is a 32-bit register. In the IDCODE mode, SDIDR can output H'001D200F, which is a fixed code, from TDO. However, no serial data can be written to SDIDR via TDI. For SDIDR, read/write by the CPU cannot be performed.

31	28	27		12	11		1	0	
0000		0001	0001	1101	0010	0000	0000	111	1
Version (4 bits)		Part Number (16 bits)				Manufacture Identify (11 bits)			Fixed Code (1 bit)



## 19.4 Operation

### 19.4.1 TAP Controller

Figure 18.2 shows the internal states of the TAP controller. State transitions basically conform with the IEEE1149.1 standard.

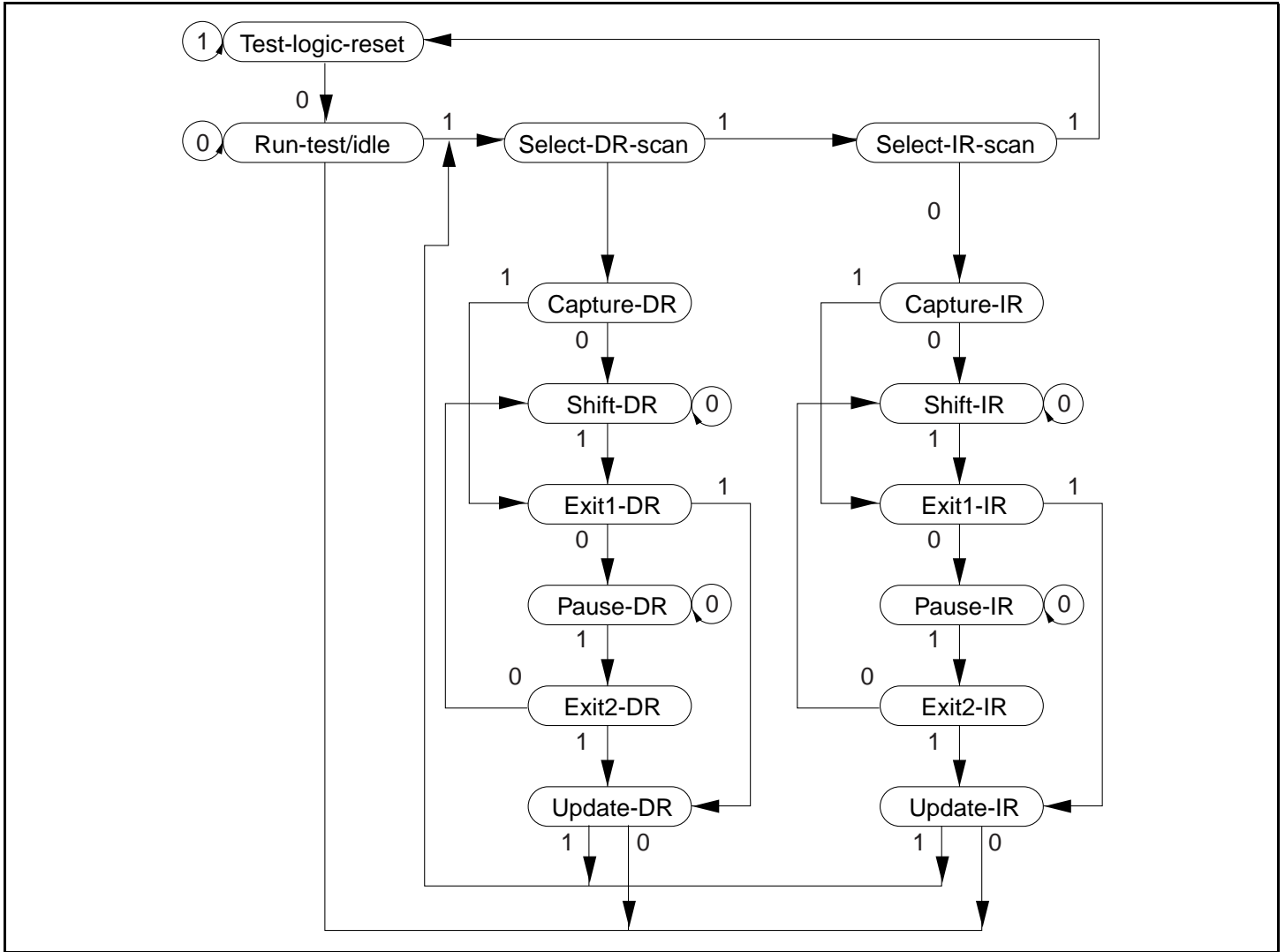


Figure 19.2 TAP Controller State Transitions

## 19.4.2 H-UDI Interrupt and Serial Transfer

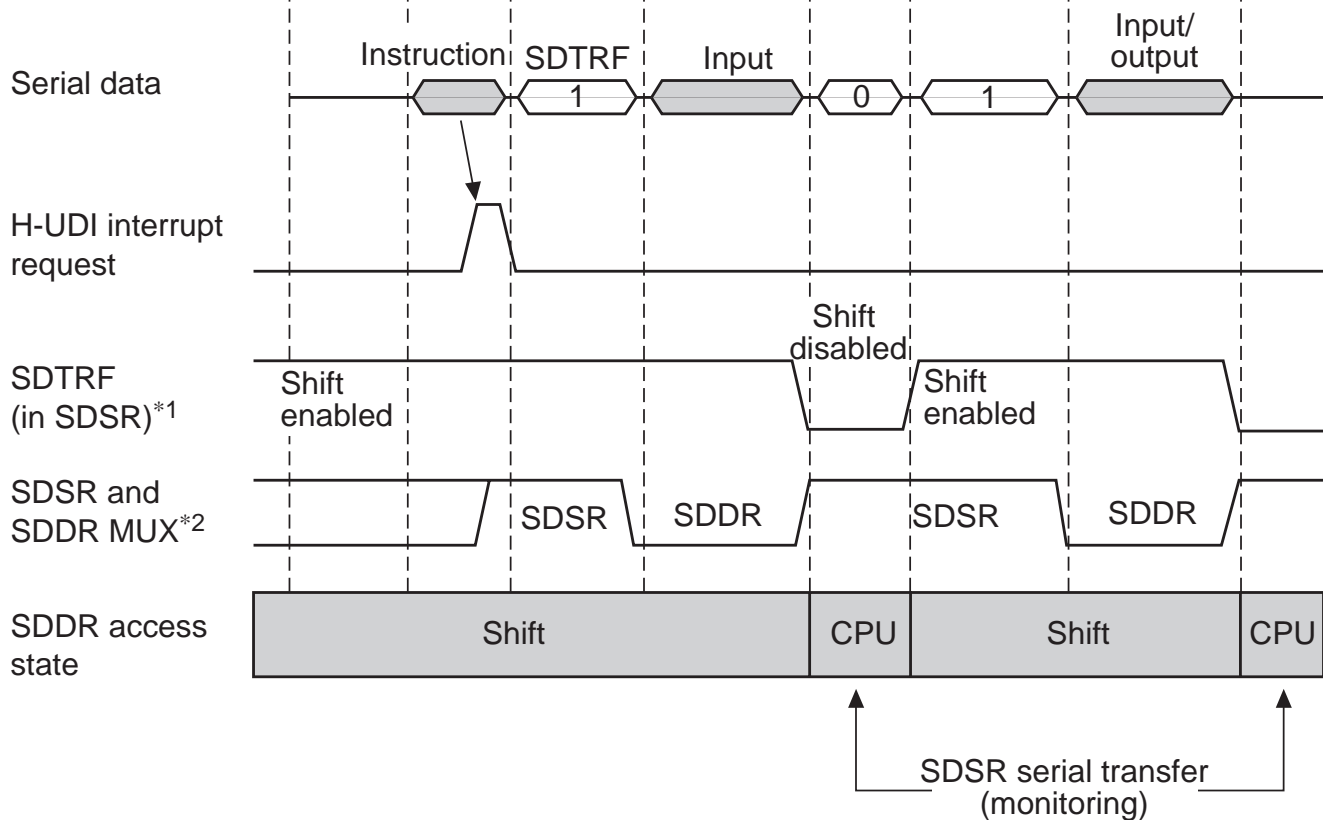
When an H-UDI interrupt instruction is transferred to SDIR via TDI, an interrupt is generated. Data transfer can be controlled by means of the H-UDI interrupt service routine. Transfer can be performed by means of SDDR.

Control of data input/output between an external device and the H-UDI is performed by monitoring the SDTRF bit in SDSR externally and internally. Internal SDTRF bit monitoring is carried out by having SDSR read by the CPU.

The H-UDI interrupt and serial transfer procedure is as follows.

1. An instruction is input to SDIR by serial transfer, and an H-UDI interrupt request is generated.
2. After the H-UDI interrupt request is issued, the SDTRF bit in SDSR is monitored externally. After output of SDTRF = 1 from TDO is observed, serial data is transferred to SDDR.
3. On completion of the serial transfer to SDDR, the SDTRF bit is cleared to 0, and SDDR can be accessed by the CPU. After SDDR has been accessed, SDDR serial transfer is enabled by setting the SDTRF bit to 1 in SDSR.
4. Serial data transfer between an external device and the H-UDI can be carried out by constantly monitoring the SDTRF bit in SDSR externally and internally.

Figures 19.3, 19.4, and 19.5 show the timing of data transfer between an external device and the H-UDI.



Notes: 1. SDTRF flag (in SDSR): Indicates whether SDDR access by the CPU or serial transfer data input/output to SDDR is possible.

1	SDDR is shift-disabled. SDDR access by the CPU is enabled.
2	SDDR is shift-enabled. Do not access SDDR until SDTRF = 0.

Conditions:

- SDTRF = 1
  - When  $\overline{\text{TRST}} = 0$
  - When the CPU writes 1
  - In BYPASS mode
- SDTRF = 0
  - End of SDDR shift access in serial transfer

2. SDSR/SDDR (Update-DR state) internal MUX switchover timing

- Switchover from SDSR to SDDR: On completion of serial transfer in which SDTRF = 1 is output from TDO
- Switchover from SDDR to SDSR: On completion of serial transfer to SDDR

**Figure 19.3 Data Input/Output Timing Chart (1)**

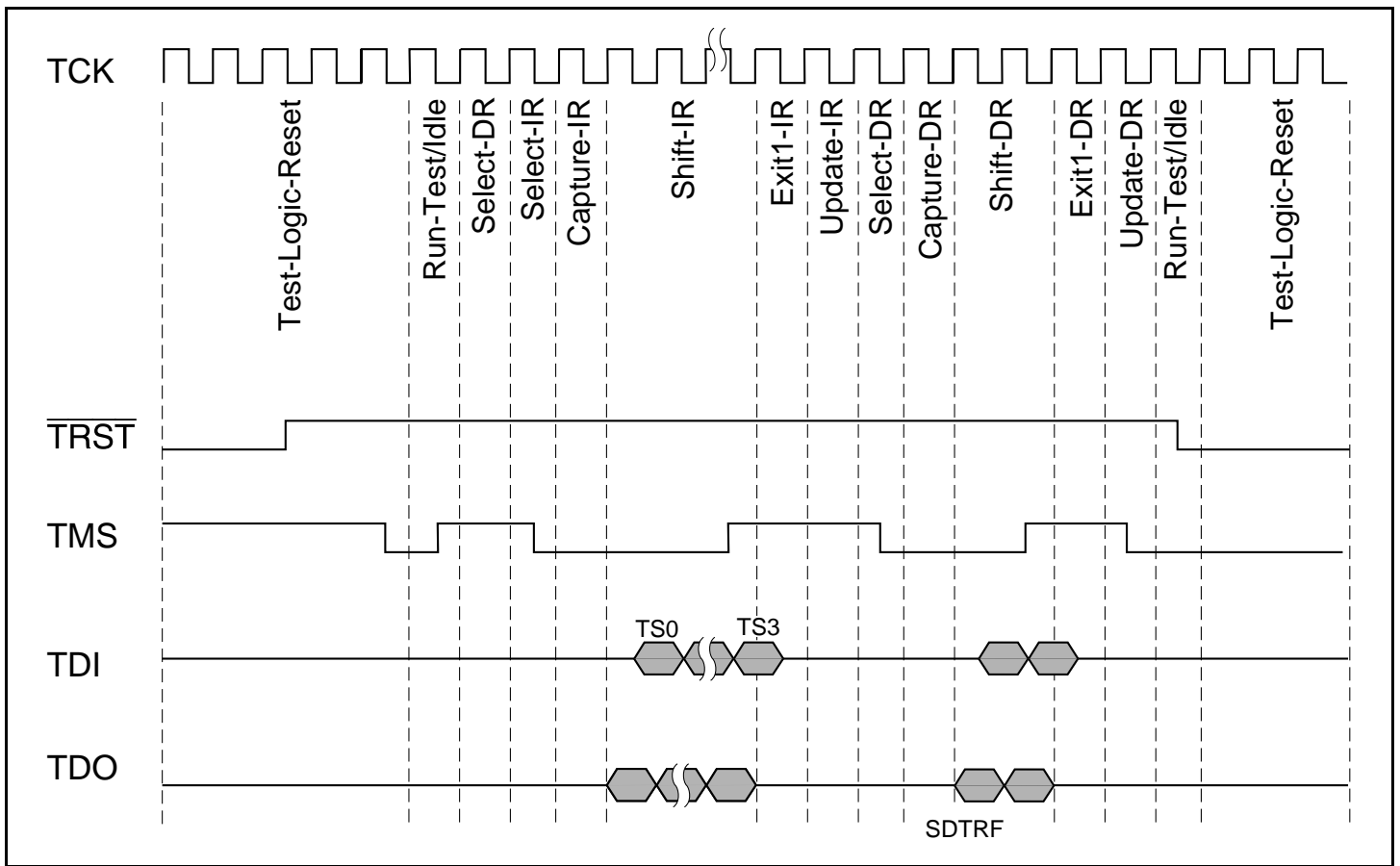


Figure 19.4 Data Input/Output Timing Chart (2)

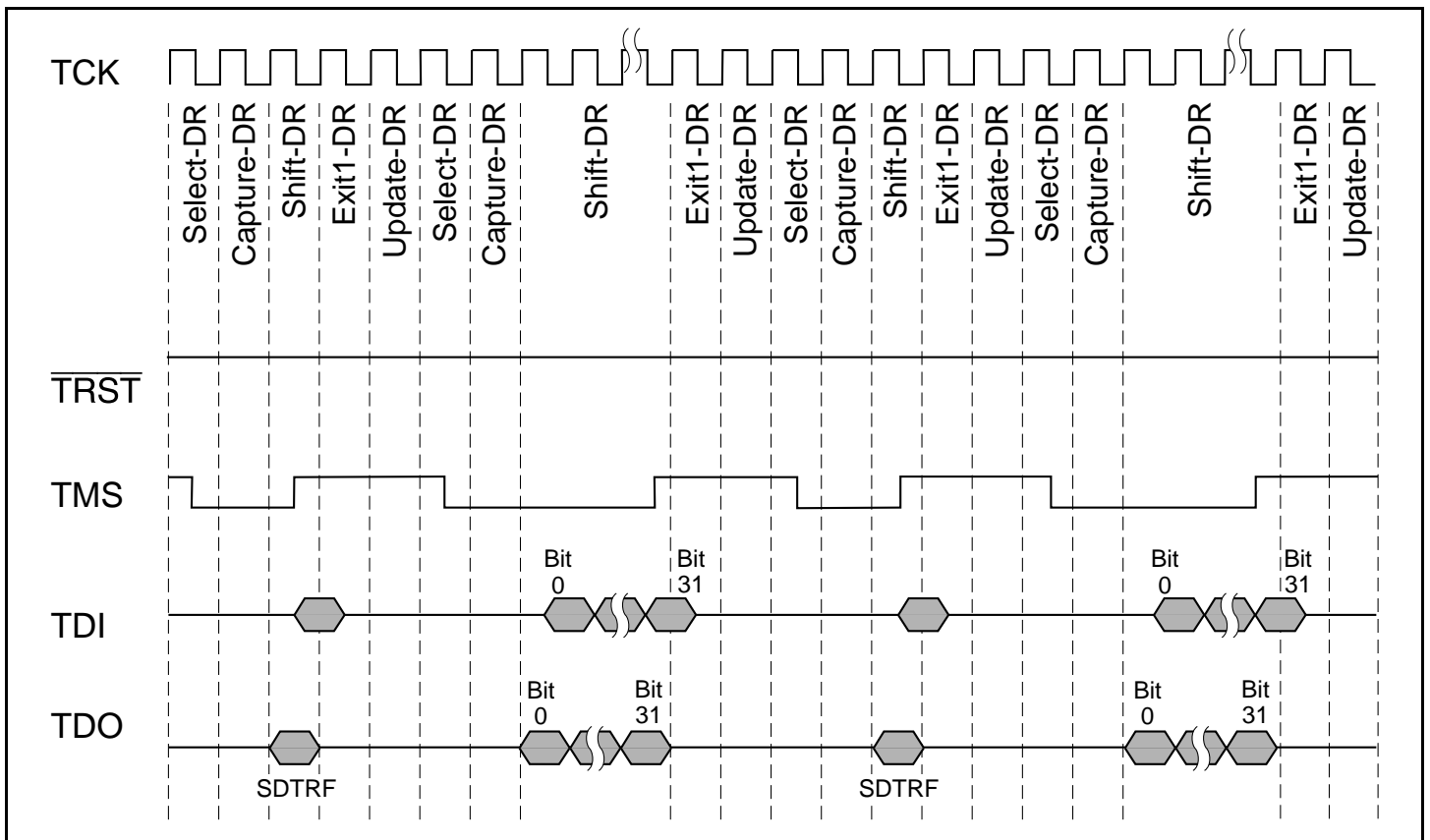


Figure 19.5 Data Input/Output Timing Chart (3)

### 19.4.3 H-UDI Reset

The H-UDI can be reset in the following cases.

- When the  $\overline{\text{TRST}}$  signal is held at 0.
- When  $\overline{\text{TRST}} = 1$  and at least five TCK clock cycles are input while  $\text{TMS} = 1$ .
- When the  $\text{MSTOP2}$  bit in  $\text{SYSCR2}$  is set to 1 (see section 25.2.3).
- In hardware standby mode.

## 19.5 Boundary Scan

The H-UDI pins can be placed in the boundary scan mode stipulated by IEEE1149.1 by setting a command in  $\text{SDIR}$ .

### 19.5.1 Supported Instructions

The SH7058 supports the three essential instructions defined in IEEE1149.1 (BYPASS, SAMPLE/PRELOAD, and EXTEST) and optional instructions (CLAMP, HIGHZ, and IDCODE).

**BYPASS:** The BYPASS instruction is an essential standard instruction that operates the bypass register. This instruction shortens the shift path to speed up serial data transfer involving other chips on the printed circuit board. While this instruction is executing, the test circuit has no effect on the system circuits. The instruction code is 1111.

**SAMPLE/PRELOAD:** The SAMPLE/PRELOAD instruction inputs values from the SH7058's internal circuitry to the boundary scan register, outputs values from the scan path, and loads data onto the scan path. When this instruction is executing, the SH7058's input pin signals are transmitted directly to the internal circuitry, and internal circuit values are directly output externally from the output pins. The SH7058's system circuits are not affected by execution of this instruction. The instruction code is 0100.

In a SAMPLE operation, a snapshot of a value to be transferred from an input pin to the internal circuitry, or a value to be transferred from the internal circuitry to an output pin, is latched into the boundary scan register and read from the scan path. Snapshot latching does not affect normal operation of the SH7058.

In a PRELOAD operation, an initial value is set in the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. Without a PRELOAD operation, when the EXTEST instruction was executed an undefined value would be output from the output pin until completion of the initial scan sequence (transfer to the output latch) (with the EXTEST instruction, the parallel output latch value is constantly output to the output pin).

**EXTEST:** This instruction is provided to test external circuitry when the SH7058 is mounted on a printed circuit board. When this instruction is executed, output pins are used to output test data

(previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed circuit board, and input pins are used to latch test results into the boundary scan register from the printed circuit board. If testing is carried out by using the EXTEST instruction N times, the Nth test data is scanned-in when test data (N-1) is scanned out.

Data loaded into the output pin boundary scan register in the Capture-DR state is not used for external circuit testing (it is replaced by a shift operation).

The instruction code is 0000.

**CLAMP:** When the CLAMP instruction is enabled, the output pin outputs the value of the boundary scan register that has been set by the SAMPLE/PRELOAD instruction. While the CLAMP instruction is enabled, the state of the boundary scan register maintains the previous state regardless of the state of the TAP controller.

A bypass register is connected between TDI and TDO. The related circuit operates in the same way when the BYPASS instruction is enabled.

The instruction code is 0010.

**HIGHZ:** When the HIGHZ instruction is enabled, all output pins enter a high-impedance state. While the HIGHZ instruction is enabled, the state of the boundary scan register maintains the previous state regardless of the state of the TAP controller.

A bypass register is connected between TDI and TDO. The related circuit operates in the same way when the BYPASS instruction is enabled.

The instruction code is 0011.

**IDCODE:** When the IDCODE instruction is enabled, the value of the ID code register is output from TDO with LSB first when the TAP controller is in the Shift-DR state. While this instruction is being executed, the test circuit does not affect the system circuit.

When the TAP controller is in the Test-Logic-Reset state, the instruction register is initialized to the IDCODE instruction.

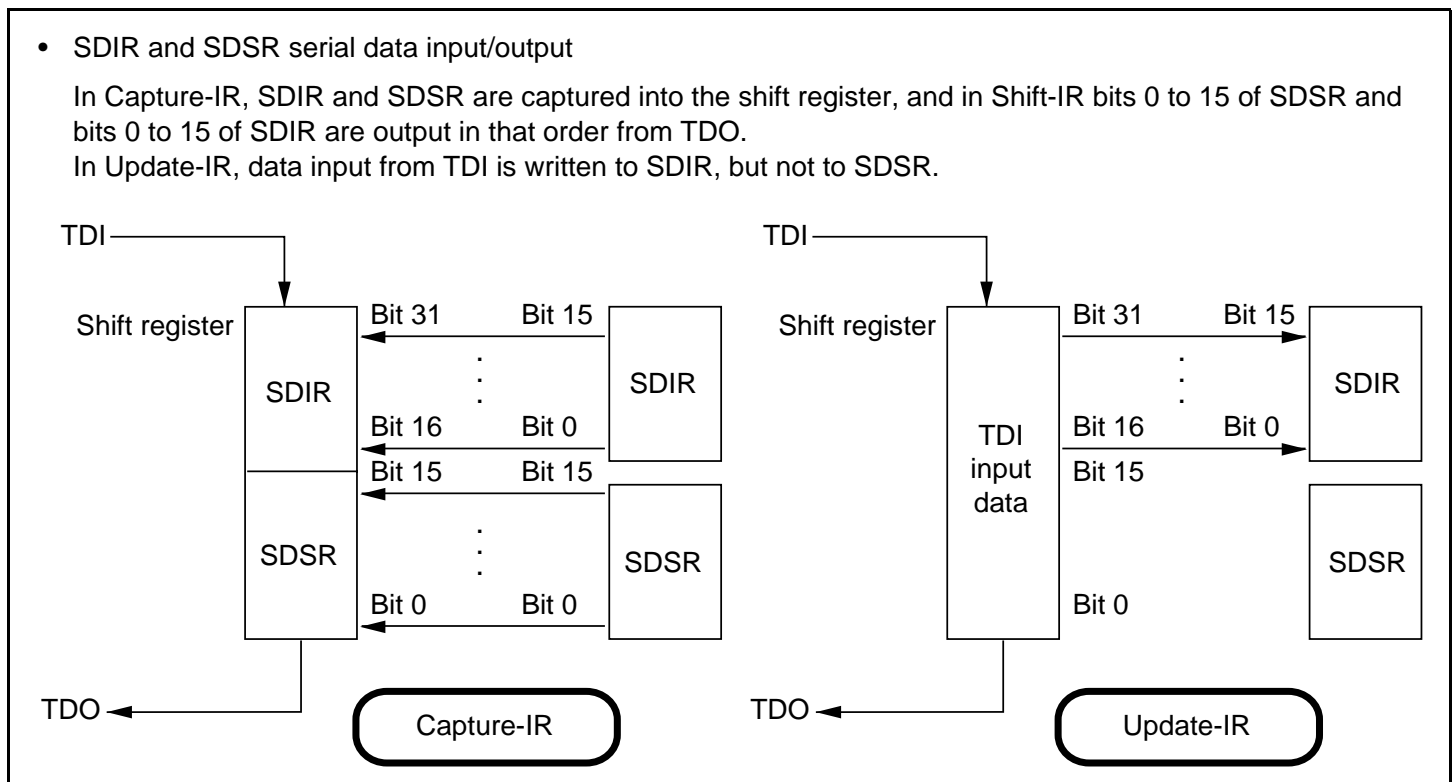
The instruction code is 1110.

## 19.5.2 Notes on Use

1. Boundary scan mode does not cover clock-related signals (EXTAL, XTAL, CK, PLLCAP).
2. Boundary scan mode does not cover reset-related signals ( $\overline{\text{RES}}$ ,  $\overline{\text{HSTBY}}$ ).
3. Boundary scan mode does not cover H-UDI-related signals (TCK, TDI, TDO, TMS,  $\overline{\text{TRST}}$ ).
4. Boundary scan mode does not cover A/D-converter-related signals (AD0 to AN31).

## 19.6 Usage Notes

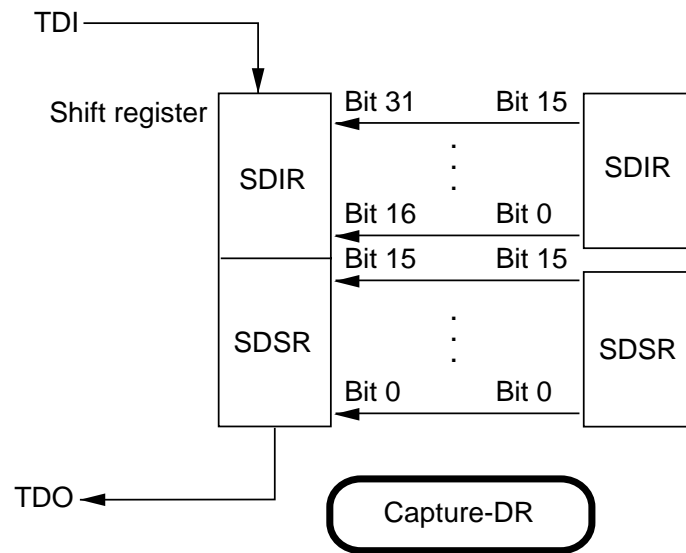
- A reset must always be executed by driving the  $\overline{\text{TRST}}$  signal to 0, regardless of whether or not the H-UDI is to be activated.  $\overline{\text{TRST}}$  must be held low for 20 TCK clock cycles. For details, see section 27, Electrical Characteristics.
- The registers are not initialized in software standby mode. If  $\overline{\text{TRST}}$  is set to 0 in software standby mode, IDCODE mode will be entered.
- The frequency of TCK must be lower than that of the peripheral module clock (P $\phi$ ). For details, see section 27, Electrical Characteristics.
- In serial data transfer, data input/output starts with the LSB. Figure 18.6 shows serial data input/output.
- When data that exceeds the number of bits of the register connected between TDI and TDO is serially transferred, the serial data that exceeds the number of register bits and output from TDO is the same as that input from TDI.
- If the H-UDI serial transfer sequence is disrupted, a  $\overline{\text{TRST}}$  reset must be executed. Transfer should then be retried, regardless of the transfer operation.
- TDO is output at the falling edge of TCK when one of six instructions defined in IEEE1149.1 is selected. Otherwise, it is output at the rising edge of TCK.



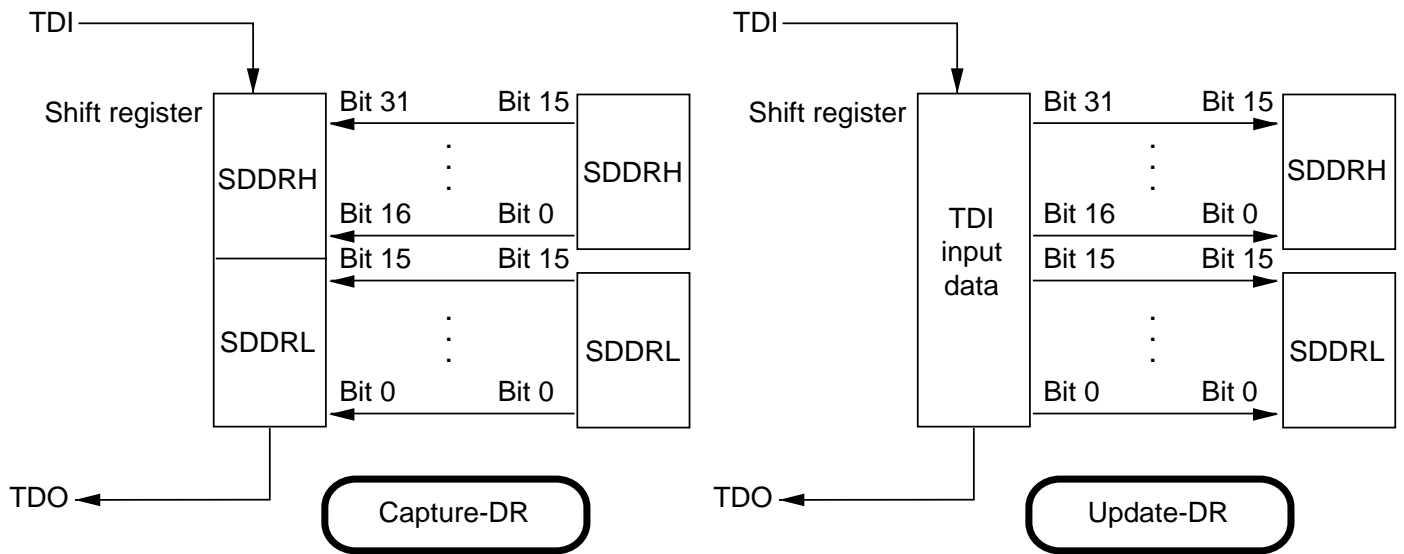
**Figure 19.6 Serial Data Input/Output (1)**

- SDDRH and SDDRL serial data input/output

- (1) In H-UDI interrupt mode, before SDTRF = 1 is read from TDO when an H-UDI interrupt is generated, SDSR and SDIR are captured into the shift register in Capture-DR, and in Shift-DR bits 0 to 15 of SDSR and bits 0 to 15 of SDIR are output in that order from TDO. In Update-DR, TDI input data is not written to any register.



- (2) In H-UDI interrupt mode, after SDTRF = 1 is read from TDO when an H-UDI interrupt is generated, SDDRH and SDDRL are captured into the shift register in Capture-DR, and in Shift-DR bits 0 to 15 of SDDRL and bits 0 to 15 of SDDRH are output in that order from TDO. Data input from TDI is written to SDDRH and SDDRL in Update-DR.



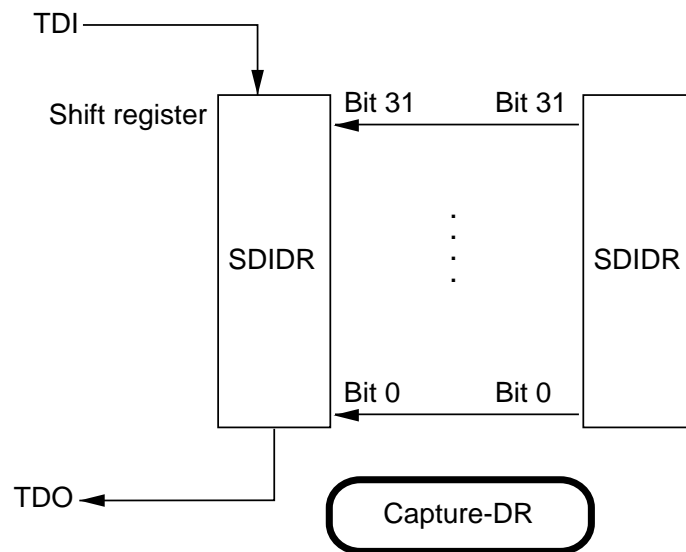
**Figure 19.6 Serial Data Input/Output (2)**



- SDIDR serial data input/output

In IDCODE mode, SDIDR is captured into the shift register in Capture-DR, and in Shift-DR bits 0 to 31 of SDIDR are output in that order from TDO.

In Update-DR, data input from TDI is not written to any register.



**Figure 19.6 Serial Data Input/Output (3)**

# Section 20 Advanced User Debugger (AUD)

## 20.1 Overview

The SH7058 has an on-chip advanced user debugger (AUD). Use of the AUD simplifies the construction of a simple emulator, with functions such as acquisition of branch trace data and monitoring/tuning of on-chip RAM data.

### 20.1.1 Features

The AUD has the following features:

- Eight input/output pins
  - Data bus (AUDATA3–AUDATA0)
  - AUD reset ( $\overline{\text{AUDRST}}$ )
  - AUD sync signal ( $\overline{\text{AUDSYNC}}$ )
  - AUD clock (AUDCK)
  - AUD mode (AUDMD)

- Two modes

Branch trace mode or RAM monitor mode can be selected by switching AUDMD.

- Branch trace mode

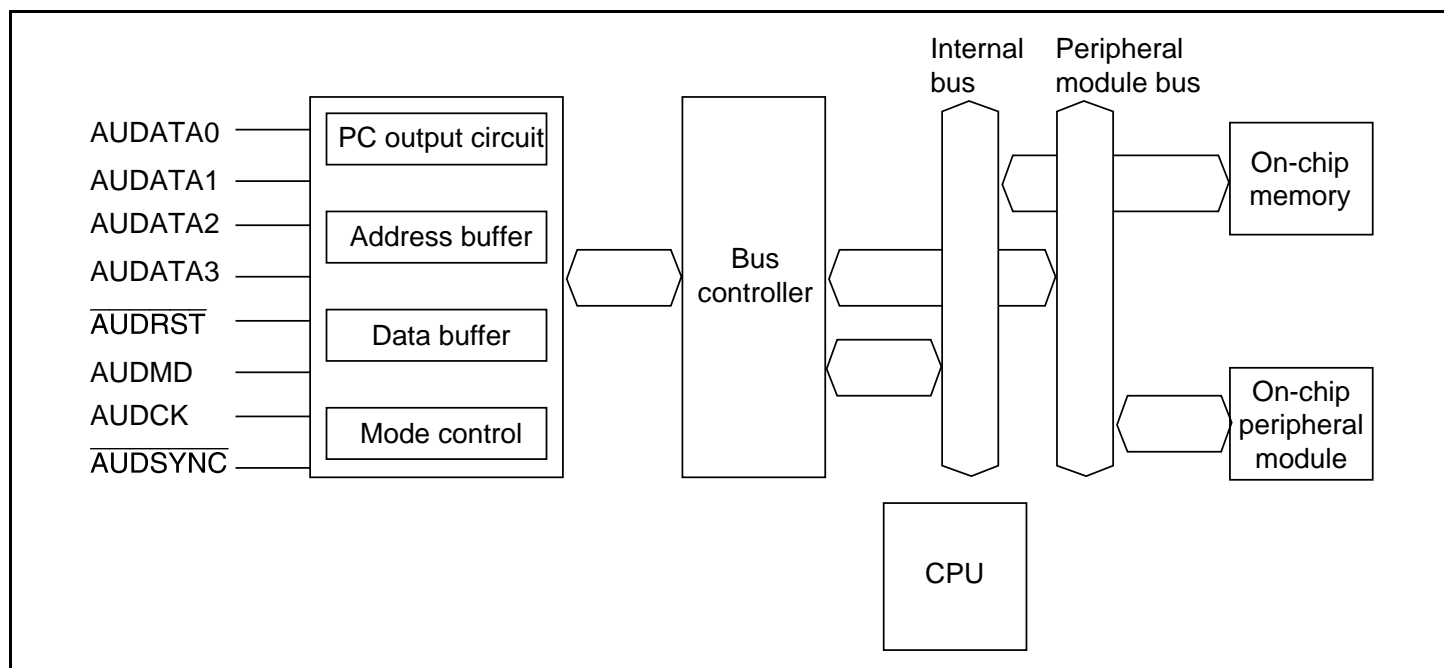
When the PC branches on execution of a branch instruction or generation of an interrupt in the user program, the branch is detected by the AUD and the branch destination address is output from AUDATA. The address is compared with the previously output address, and 4-, 8-, 16-, or 32-bit output is selected automatically according to the upper address matching status.

- RAM monitor mode

When an address is written to AUDATA from off-chip, the data corresponding to that address is output. If an address and data are written to AUDATA, the data is transferred to that address.

## 20.1.2 Block Diagram

Figure 20.1 shows a block diagram of the AUD.



**Figure 20.1 AUD Block Diagram**

## 20.2 Pin Configuration

Table 20.1 shows the AUD's input/output pins.

**Table 20.1 AUD Pins**

Name	Abbreviation	Function	
		Branch Trace Mode	RAM Monitor Mode
AUD data	AUDATA3– AUDATA0	Branch destination address output	Monitor address/data input/output
AUD reset	$\overline{\text{AUDRST}}$	AUD reset input	AUD reset input
AUD mode	AUDMD	Mode select input (L)	Mode select input (H)
AUD clock	AUDCK	Serial clock ( $P\phi$ ) output	Serial clock input
AUD sync signal	$\overline{\text{AUDSYNC}}$	Data start position identification signal output	Data start position identification signal input

## 20.2.1 Pin Descriptions

### Pins Used in Both Modes

Pin	Description
AUDMD	<p>The mode is selected by changing the input level at this pin.</p> <p>Low: Branch trace mode</p> <p>High: RAM monitor mode</p> <p>The input at this pin should be changed when <math>\overline{\text{AUDRST}}</math> is low. When no connection is made, this pin is pulled up internally.</p>
$\overline{\text{AUDRST}}$	<p>The AUD's internal buffers and logic are initialized by inputting a low level to this pin. When this signal goes low, the AUD enters the reset state and the AUD's internal buffers and logic are reset. When <math>\overline{\text{AUDRST}}</math> goes high again after the AUDMD level settles, the AUD starts operating in the selected mode. When no connection is made, this pin is pulled down internally.</p>

## Pin Functions in Branch Trace Mode

Pin	Description
AUDCK	This pin outputs the peripheral module operating frequency ( $P\phi$ ). This is the clock for AUDATA synchronization.
$\overline{\text{AUDSYNC}}$	This pin indicates whether output from AUDATA is valid. High: Valid data is not being output Low: An address is being output
AUDATA3 to AUDATA0	<ol style="list-style-type: none"> <li>When <math>\overline{\text{AUDSYNC}}</math> is low When a program branch or interrupt branch occurs, the AUD asserts <math>\overline{\text{AUDSYNC}}</math> and outputs the branch destination address. The output order is A3–A0, A7–A4, A11–A8, A15–A12, A19–A16, A23–A20, A27–A24, A31–A28.</li> <li>When <math>\overline{\text{AUDSYNC}}</math> is high When waiting for branch destination address output, these pins constantly output 0011. When an branch occurs, AUDATA3–AUDATA2 output 10, and AUDATA1–AUDATA0 indicate whether a 4-, 8-, 16-, or 32-bit address is to be output by comparing the previous fully output address with the address output this time (see table below).</li> </ol>

AUDATA1, AUDATA0	
00	Address bits A31–A4 match; 4 address bits A3–A0 are to be output (i.e. output is performed once).
01	Address bits A31–A8 match; 8 address bits A3–A0 and A7–A4 are to be output (i.e. output is performed twice).
10	Address bits A31–A16 match; 16 address bits A3–A0, A7–A4, A11–A8, and A15–A12 are to be output (i.e. output is performed four times).
11	None of the above cases applies; 32 address bits A3–A0, A7–A4, A11–A8, and A15–A12, A19–A16, A23–A20, A27–A24, and A31–A28 are to be output (i.e. output is performed eight times).

## Pin Functions in RAM Monitor Mode

Pin	Description
AUDCK	The external clock input pin. Input the clock to be used for debugging to this pin. The input frequency must not exceed 10 MHz. When no connection is made, this pin is pulled up internally.
$\overline{\text{AUDSYNC}}$	Do not assert this pin until a command is input to AUDATA from off-chip and the necessary data can be prepared. See the protocol description for details. When no connection is made, this pin is pulled up internally.
AUDATA3 to AUDATA0	When a command is input from off-chip, data is output after Ready reception. Output starts when $\overline{\text{AUDSYNC}}$ is negated. See the protocol description for details. When no connections are made, these pins are pulled up internally.

## 20.3 Branch Trace Mode

### 20.3.1 Overview

In this mode, the branch destination address is output when a branch occurs in the user program. Branches may be caused by branch instruction execution or interrupt/exception processing, but no distinction is made between the two in this mode.

### 20.3.2 Operation

Operation starts in branch trace mode when  $\overline{\text{AUDRST}}$  is asserted, AUDMD is driven low, then  $\overline{\text{AUDRST}}$  is negated.

Figure 20.2 shows an example of data output.

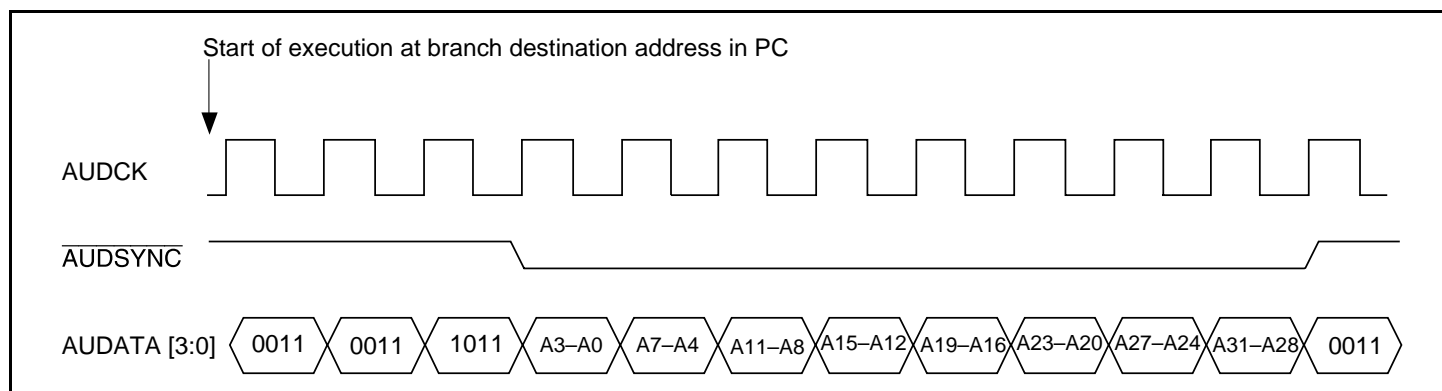
While the user program is being executed without branches, the AUDATA pins constantly output 0011 in synchronization with AUDCK.

When a branch occurs, after execution starts at the branch destination address in the PC, the previous fully output address (i.e. for which output was not interrupted by the occurrence of another branch) is compared with the current branch address, and depending on the result,  $\overline{\text{AUDSYNC}}$  is asserted and the branch destination address output after 1-clock output of 1000 (in the case of 4-bit output), 1001 (8-bit output), 1010 (16-bit output), or 1011 (32-bit output). The initial value of the compared address is H'00000000.

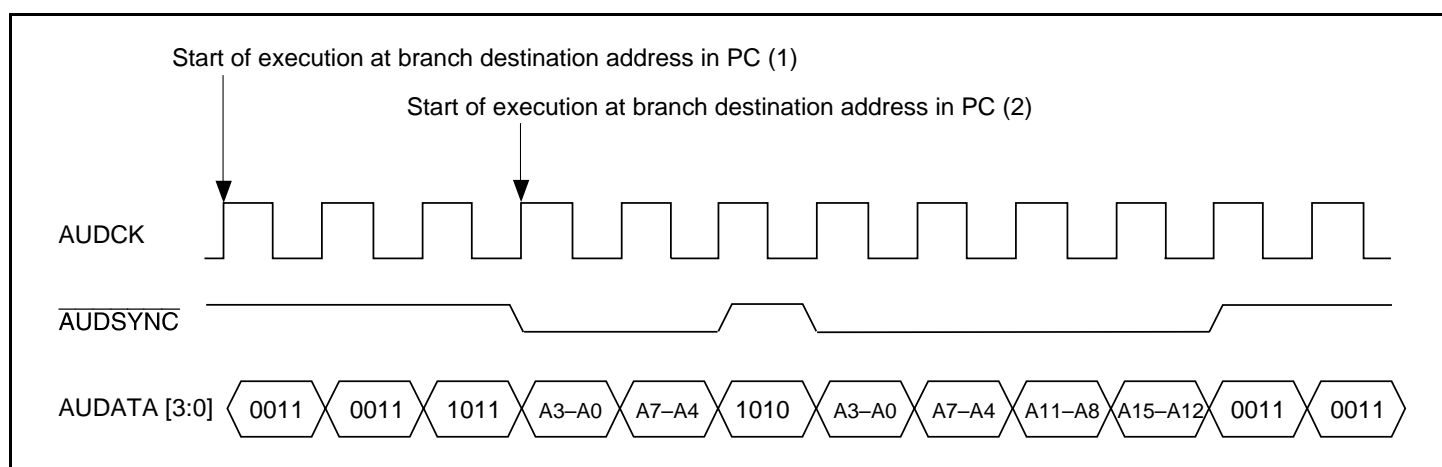
On completion of the cycle in which the address is output,  $\overline{\text{AUDSYNC}}$  is negated and 0011 is output from the AUDATA pins.

If another branch occurs during branch destination address output, the later branch has priority for output. In this case,  $\overline{\text{AUDSYNC}}$  is negated and the AUDATA pins output the address after outputting 10xx again (figure 20.3 shows an example of the output when consecutive branches occur). Note that the compared address is the previous fully output address, and not an interrupted address (since the upper address of an interrupted address will be unknown).

The interval from the start of execution at the branch destination address in the PC until the AUDATA pins output 10xx is 1.5 or 2 AUDCK cycles.



**Figure 20.2 Example of Data Output (32-Bit Output)**



**Figure 20.3 Example of Output in Case of Successive Branches**

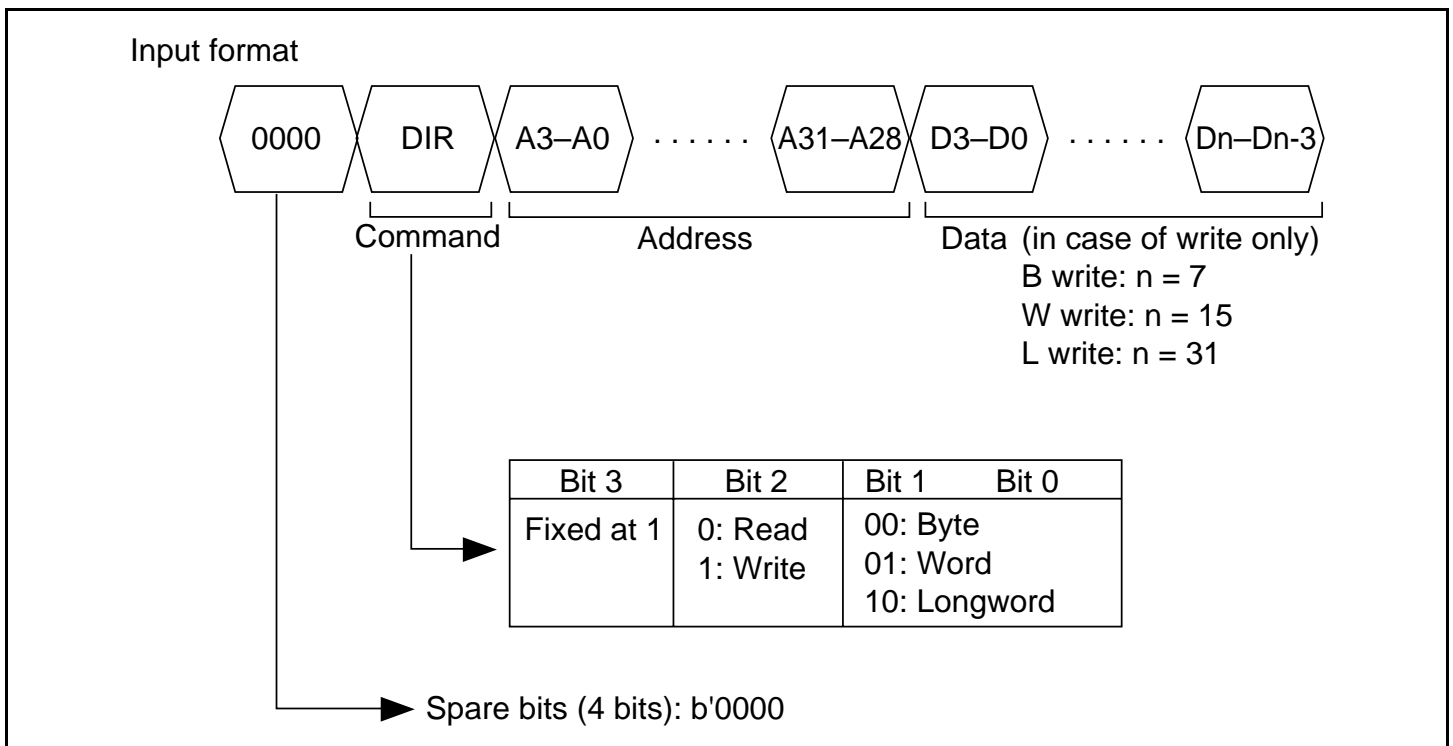
## 20.4 RAM Monitor Mode

### 20.4.1 Overview

In this mode, all the modules connected to the SH7058's internal or external bus can be read and written to, allowing RAM monitoring and tuning to be carried out.

### 20.4.2 Communication Protocol

The AUD latches the AUDATA input when  $\overline{\text{AUDSYNC}}$  is asserted. The following AUDATA input format should be used.



**Figure 20.4 AUDATA Input Format**



### 20.4.3 Operation

Operation starts in RAM monitor mode when AUDMD is driven high after  $\overline{\text{AUDRST}}$  has been asserted, then  $\overline{\text{AUDRST}}$  is negated.

Figure 20.5 shows an example of a read operation, and figure 20.6 an example of a write operation.

When  $\overline{\text{AUDSYNC}}$  is asserted, input from the AUDATA pins begins. When a command, address, or data (writing only) is input in the format shown in figure 20.2, execution of read/write access to the specified address is started. During internal execution, the AUD returns Not Ready (0000). When execution is completed, the Ready flag (0001) is returned (figures 20.5 and 20.6). Table 20.2 shows the Ready flag format.

In a read, data of the specified size is output when  $\overline{\text{AUDSYNC}}$  is negated following detection of this flag (figure 20.7).

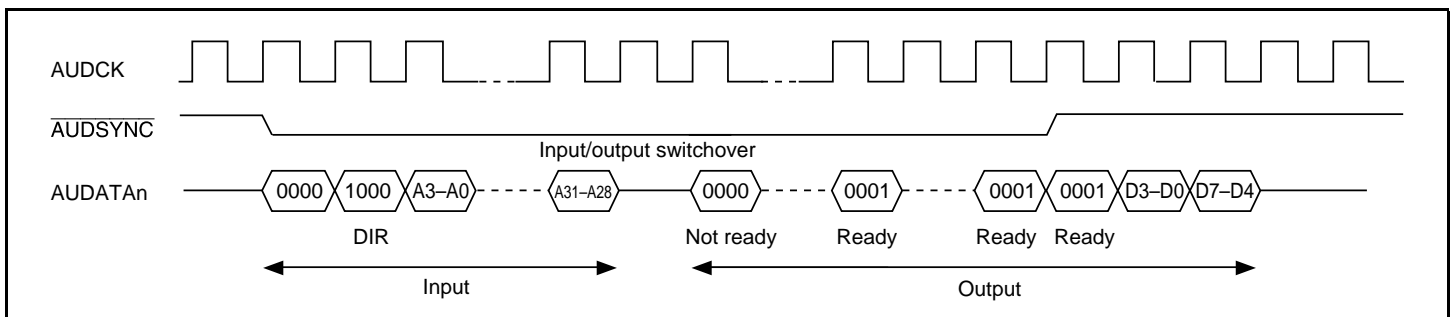
If a command other than the above is input in DIR, the AUD treats this as a command error, disables processing, and sets bit 1 in the Ready flag to 1. If a read/write operation initiated by the command specified in DIR causes a bus error, the AUD disables processing and sets bit 2 in the Ready flag to 1 (figure 20.7).

**Table 20.2 Ready Flag Format**

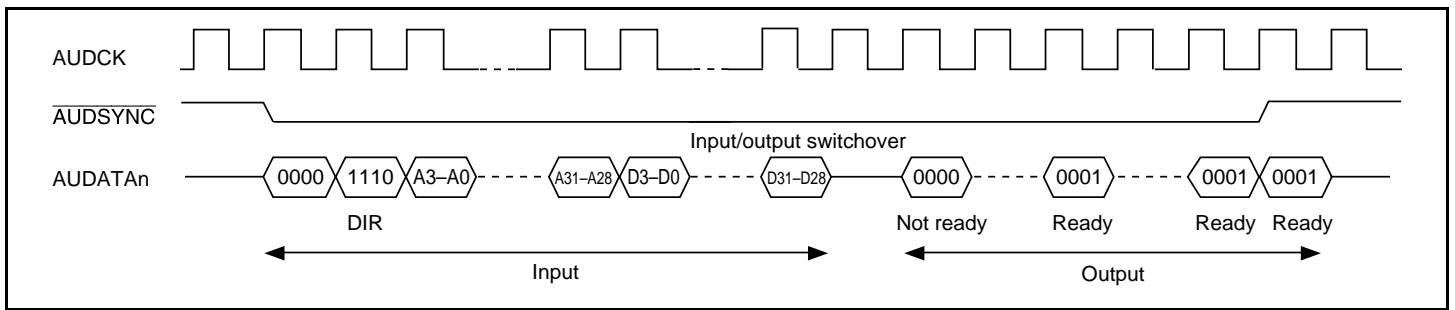
Bit 3	Bit 2	Bit 1	Bit 0
Fixed at 0	0: Normal status 1: Bus error	0: Normal status 1: Bus error	0: Not ready 1: Ready

Bus error conditions are shown below.

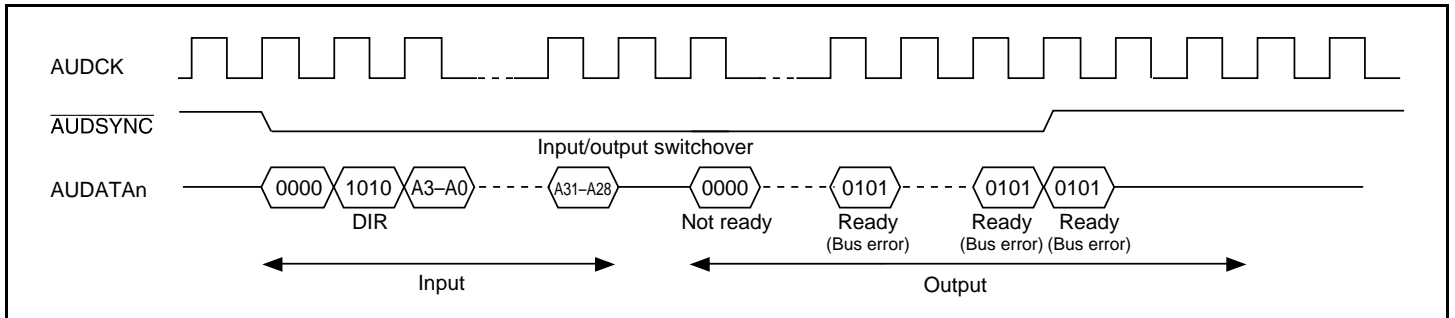
1. Word access to address  $4n+1$  or  $4n+3$
2. Longword access to address  $4n+1$ ,  $4n+2$ , or  $4n+3$
3. Longword access to on-chip I/O 8-bit space
4. Access to external space in single-chip mode



**Figure 20.5 Example of Read Operation (Byte Read)**



**Figure 20.6 Example of Write Operation (Longword Write)**



**Figure 20.7 Example of Error Occurrence (Longword Read)**

## 20.5 Usage Notes

### 20.5.1 Initialization

The debugger's internal buffers and processing states are initialized in the following cases:

1. In a power-on reset
2. In hardware standby mode
3. When  $\overline{\text{AUDRST}}$  is driven low
4. When the AUDSRST bit is set to 1 in the SYSCR1 register (see section 25.2.2)
5. When the MSTOP3 bit is set to 1 in the SYSCR2 register (see section 25.2.3)

### 20.5.2 Operation in Software Standby Mode

The debugger is not initialized in software standby mode. However, since the SH7058's internal operation halts in software standby mode:

1. When AUDMD is high (RAM monitor mode): Operation stops. This setting should not be used in standby mode.
2. When AUDMD is low (PC trace): Operation stops. However, operation continues when software standby is released.



# Section 21 Pin Function Controller (PFC)

## 21.1 Overview

The pin function controller (PFC) consists of registers for selecting multiplex pin functions and their input/output direction. Table 21.1 shows the SH7058's multiplex pins.

**Table 21.1 SH7058 Multiplex Pins**

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
A	PA0 input/output (port)	TIOA input (ATU-II)		
A	PA1 input/output (port)	TIOB input (ATU-II)		
A	PA2 input/output (port)	TIOC input (ATU-II)		
A	PA3 input/output (port)	TIOD input (ATU-II)		
A	PA4 input/output (port)	TIO3A input/output (ATU-II)		
A	PA5 input/output (port)	TIO3B input/output (ATU-II)		
A	PA6 input/output (port)	TIO3C input/output (ATU-II)		
A	PA7 input/output (port)	TIO3D input/output (ATU-II)		
A	PA8 input/output (port)	TIO4A input/output (ATU-II)		ADTO0A output (MTAD)
A	PA9 input/output (port)	TIO4B input/output (ATU-II)		ADTO0B output (MTAD)
A	PA10 input/output (port)	TIO4C input/output (ATU-II)		ADTO1A output (MTAD)
A	PA11 input/output (port)	TIO4D input/output (ATU-II)		ADTO1B output (MTAD)
A	PA12 input/output (port)	TIO5A input/output (ATU-II)		
A	PA13 input/output (port)	TIO5B input/output (ATU-II)		
A	PA14 input/output (port)	TxD0 output (SCI)		
A	PA15 input/output (port)	RxD0 input (SCI)		
B	PB0 input/output (port)	TO6A output (ATU-II)		
B	PB1 input/output (port)	TO6B output (ATU-II)		
B	PB2 input/output (port)	TO6C output (ATU-II)		
B	PB3 input/output (port)	TO6D output (ATU-II)		
B	PB4 input/output (port)	TO7A output (ATU-II)	TO8A output (ATU-II)	
B	PB5 input/output (port)	TO7B output (ATU-II)	TO8B output (ATU-II)	
B	PB6 input/output (port)	TO7C output (ATU-II)	TO8C output (ATU-II)	
B	PB7 input/output (port)	TO7D output (ATU-II)	TO8D output (ATU-II)	
B	PB8 input/output (port)	TxD3 output (SCI)	TO8E output (ATU-II)	

**Table 21.1 SH7058 Multiplex Pins (cont)**

<b>Port</b>	<b>Function 1 (Related Module)</b>	<b>Function 2 (Related Module)</b>	<b>Function 3 (Related Module)</b>	<b>Function 4 (Related Module)</b>
B	PB9 input/output (port)	RxD3 input (SCI)	TO8F output (ATU-II)	
B	PB10 input/output (port)	TxD4 output (SCI)	HTxD0 output (HCAN-II)	TO8G output (ATU-II)
B	PB11 input/output (port)	RxD4 input (SCI)	HRxD0 input (HCAN-II)	TO8H output (ATU-II)
B	PB12 input/output (port)	TCLKA input (ATU-II)	$\overline{\text{UBCTRG}}$ output (UBC)	
B	PB13 input/output (port)	SCK0 input/output (SCI)		
B	PB14 input/output (port)	SCK1 input/output (SCI)	TCLKB input (ATU-II)	TI10 input (ATU-II)
B	PB15 input/output (port)	PULS5 output (APC)	SCK2 input/output (SCI)	
C	PC0 input/output (port)	TxD1 output (SCI)		
C	PC1 input/output (port)	RxD1 input (SCI)		
C	PC2 input/output (port)	TxD2 output (SCI)		
C	PC3 input/output (port)	RxD2 input (SCI)		
C	PC4 input/output (port)	$\overline{\text{IRQ0}}$ input (INTC)		
D	PD0 input/output (port)	TIO1A input/output (ATU-II)		
D	PD1 input/output (port)	TIO1B input/output (ATU-II)		
D	PD2 input/output (port)	TIO1C input/output (ATU-II)		
D	PD3 input/output (port)	TIO1D input/output (ATU-II)		
D	PD4 input/output (port)	TIO1E input/output (ATU-II)		
D	PD5 input/output (port)	TIO1F input/output (ATU-II)		
D	PD6 input/output (port)	TIO1G input/output (ATU-II)		
D	PD7 input/output (port)	TIO1H input/output (ATU-II)		
D	PD8 input/output (port)	PULS0 output (APC)		
D	PD9 input/output (port)	PULS1 output (APC)		
D	PD10 input/output (port)	PULS2 output (APC)		
D	PD11 input/output (port)	PULS3 output (APC)		
D	PD12 input/output (port)	PULS4 output (APC)		
D	PD13 input/output (port)	PULS6 output (APC)	HTxD0 output (HCAN-II)	HTxD1 output (HCAN-II)
E	PE0 input/output (port)	A0 output (BSC)		
E	PE1 input/output (port)	A1 output (BSC)		
E	PE2 input/output (port)	A2 output (BSC)		
E	PE3 input/output (port)	A3 output (BSC)		
E	PE4 input/output (port)	A4 output (BSC)		
E	PE5 input/output (port)	A5 output (BSC)		
E	PE6 input/output (port)	A6 output (BSC)		
E	PE7 input/output (port)	A7 output (BSC)		

**Table 21.1 SH7058 Multiplex Pins (cont)**

<b>Port</b>	<b>Function 1 (Related Module)</b>	<b>Function 2 (Related Module)</b>	<b>Function 3 (Related Module)</b>	<b>Function 4 (Related Module)</b>
E	PE8 input/output (port)	A8 output (BSC)		
E	PE9 input/output (port)	A9 output (BSC)		
E	PE10 input/output (port)	A10 output (BSC)		
E	PE11 input/output (port)	A11 output (BSC)		
E	PE12 input/output (port)	A12 output (BSC)		
E	PE13 input/output (port)	A13 output (BSC)		
E	PE14 input/output (port)	A14 output (BSC)		
E	PE15 input/output (port)	A15 output (BSC)		
F	PF0 input/output (port)	A16 output (BSC)		
F	PF1 input/output (port)	A17 output (BSC)		
F	PF2 input/output (port)	A18 output (BSC)		
F	PF3 input/output (port)	A19 output (BSC)		
F	PF4 input/output (port)	A20 output (BSC)		
F	PF5 input/output (port)	A21 output (BSC)	$\overline{\text{POD}}$ input (port)	
F	PF6 input/output (port)	$\overline{\text{WRL}}$ output (BSC)		
F	PF7 input/output (port)	$\overline{\text{WRH}}$ output (BSC)		
F	PF8 input/output (port)	$\overline{\text{WAIT}}$ input (BSC)		
F	PF9 input/output (port)	$\overline{\text{RD}}$ output (BSC)		
F	PF10 input/output (port)	$\overline{\text{CS0}}$ output (BSC)		
F	PF11 input/output (port)	$\overline{\text{CS1}}$ output (BSC)		
F	PF12 input/output (port)	$\overline{\text{CS2}}$ output (BSC)		
F	PF13 input/output (port)	$\overline{\text{CS3}}$ output (BSC)		
F	PF14 input/output (port)	$\overline{\text{BACK}}$ output (BSC)		
F	PF15 input/output (port)	$\overline{\text{BREQ}}$ input (BSC)		
G	PG0 input/output (port)	PULS7 output (APC)	HRxD0 input (HCAN-II)	HRxD1 input (HCAN-II)
G	PG1 input/output (port)	$\overline{\text{IRQ1}}$ input (INTC)		
G	PG2 input/output (port)	$\overline{\text{IRQ2}}$ input (INTC)	ADEND output (A/D)	
G	PG3 input/output (port)	$\overline{\text{IRQ3}}$ input (INTC)	$\overline{\text{ADTRG0}}$ input (A/D)	
H	PH0 input/output (port)	D0 input/output (BSC)		
H	PH1 input/output (port)	D1 input/output (BSC)		
H	PH2 input/output (port)	D2 input/output (BSC)		
H	PH3 input/output (port)	D3 input/output (BSC)		
H	PH4 input/output (port)	D4 input/output (BSC)		

**Table 21.1 SH7058 Multiplex Pins (cont)**

<b>Port</b>	<b>Function 1 (Related Module)</b>	<b>Function 2 (Related Module)</b>	<b>Function 3 (Related Module)</b>	<b>Function 4 (Related Module)</b>
H	PH5 input/output (port)	D5 input/output (BSC)		
H	PH6 input/output (port)	D6 input/output (BSC)		
H	PH7 input/output (port)	D7 input/output (BSC)		
H	PH8 input/output (port)	D8 input/output (BSC)		
H	PH9 input/output (port)	D9 input/output (BSC)		
H	PH10 input/output (port)	D10 input/output (BSC)		
H	PH11 input/output (port)	D11 input/output (BSC)		
H	PH12 input/output (port)	D12 input/output (BSC)		
H	PH13 input/output (port)	D13 input/output (BSC)		
H	PH14 input/output (port)	D14 input/output (BSC)		
H	PH15 input/output (port)	D15 input/output (BSC)		
J	PJ0 input/output (port)	TIO2A input/output (ATU-II)		
J	PJ1 input/output (port)	TIO2B input/output (ATU-II)		
J	PJ2 input/output (port)	TIO2C input/output (ATU-II)		
J	PJ3 input/output (port)	TIO2D input/output (ATU-II)		
J	PJ4 input/output (port)	TIO2E input/output (ATU-II)		
J	PJ5 input/output (port)	TIO2F input/output (ATU-II)		
J	PJ6 input/output (port)	TIO2G input/output (ATU-II)		
J	PJ7 input/output (port)	TIO2H input/output (ATU-II)		
J	PJ8 input/output (port)	TIO5C input/output (ATU-II)		
J	PJ9 input/output (port)	TIO5D input/output (ATU-II)		
J	PJ10 input/output (port)	TI9A input (ATU-II)		
J	PJ11 input/output (port)	TI9B input (ATU-II)		
J	PJ12 input/output (port)	TI9C input (ATU-II)		
J	PJ13 input/output (port)	TI9D input (ATU-II)		
J	PJ14 input/output (port)	TI9E input (ATU-II)		
J	PJ15 input/output (port)	TI9F input (ATU-II)		
K	PK0 input/output (port)	TO8A output (ATU-II)		
K	PK1 input/output (port)	TO8B output (ATU-II)		
K	PK2 input/output (port)	TO8C output (ATU-II)		
K	PK3 input/output (port)	TO8D output (ATU-II)		
K	PK4 input/output (port)	TO8E output (ATU-II)		
K	PK5 input/output (port)	TO8F output (ATU-II)		

**Table 21.1 SH7058 Multiplex Pins (cont)**

Port	Function 1 (Related Module)	Function 2 (Related Module)	Function 3 (Related Module)	Function 4 (Related Module)
K	PK6 input/output (port)	TO8G output (ATU-II)		
K	PK7 input/output (port)	TO8H output (ATU-II)		
K	PK8 input/output (port)	TO8I output (ATU-II)		
K	PK9 input/output (port)	TO8J output (ATU-II)		
K	PK10 input/output (port)	TO8K output (ATU-II)		
K	PK11 input/output (port)	TO8L output (ATU-II)		
K	PK12 input/output (port)	TO8M output (ATU-II)		
K	PK13 input/output (port)	TO8N output (ATU-II)		
K	PK14 input/output (port)	TO8O output (ATU-II)		
K	PK15 input/output (port)	TO8P output (ATU-II)		
L	PL0 input/output (port)	TI10 input (ATU-II)		
L	PL1 input/output (port)	TIO11A input/output (ATU-II)	$\overline{\text{IRQ6}}$ input (INTC)	
L	PL2 input/output (port)	TIO11B input/output (ATU-II)	$\overline{\text{IRQ7}}$ input (INTC)	
L	PL3 input/output (port)	TCLKB input (ATU-II)		
L	PL4 input/output (port)	$\overline{\text{ADTRG0}}$ input (A/D)		
L	PL5 input/output (port)	$\overline{\text{ADTRG1}}$ input (A/D)		
L	PL6 input/output (port)	ADEND output (A/D)		
L	PL7 input/output (port)	SCK2 input/output (SCI)		
L	PL8 input/output (port)	SCK3 input/output (SCI)		
L	PL9 input/output (port)	SCK4 input/output (SCI)	$\overline{\text{IRQ5}}$ input (INTC)	
L	PL10 input/output (port)	HTxD0 output (HCAN-II)	HTxD1 output (HCAN-II)	HTxD0 & HTxD1 (HCAN-II)
L	PL11 input/output (port)	HRxD0 input (HCAN-II)	HRxD1 input (HCAN-II)	HRxD0 & HRxD1 (HCAN-II)
L	PL12 input/output (port)	$\overline{\text{IRQ4}}$ input (INTC)		
L	PL13 input/output (port)	$\overline{\text{IRQOUT}}$ output (INTC)	$\overline{\text{IRQOUT}}$ output (INTC)	



## 21.2 Register Configuration

PFC registers are listed in table 21.2.

**Table 21.2 PFC Registers**

<b>Name</b>	<b>Abbreviation</b>	<b>R/W</b>	<b>Initial Value</b>	<b>Address</b>	<b>Access Size</b>
Port A IO register	PAIOR	R/W	H'0000	H'FFFFFF720	8, 16
Port A control register H	PACRH	R/W	H'0000	H'FFFFFF722	8, 16
Port A control register L	PACRL	R/W	H'0000	H'FFFFFF724	8, 16
Port B IO register	PBIOR	R/W	H'0000	H'FFFFFF730	8, 16
Port B control register H	PBCRH	R/W	H'0000	H'FFFFFF732	8, 16
Port B control register L	PBCRL	R/W	H'0000	H'FFFFFF734	8, 16
Port B invert register	PBIR	R/W	H'0000	H'FFFFFF736	8, 16
Port C IO register	PCIOR	R/W	H'0000	H'FFFFFF73A	8, 16
Port C control register	PCCR	R/W	H'0000	H'FFFFFF73C	8, 16
Port D IO register	PDIOR	R/W	H'0000	H'FFFFFF740	8, 16
Port D control register H	PDCRH	R/W	H'0000	H'FFFFFF742	8, 16
Port D control register L	PDCRL	R/W	H'0000	H'FFFFFF744	8, 16
Port E IO register	PEIOR	R/W	H'0000	H'FFFFFF750	8, 16
Port E control register	PECR	R/W	H'0000	H'FFFFFF752	8, 16
Port F IO register	PFIOR	R/W	H'0000	H'FFFFFF748	8, 16
Port F control register H	PFCRH	R/W	H'0015	H'FFFFFF74A	8, 16
Port F control register L	PFCRL	R/W	H'5000	H'FFFFFF74C	8, 16
Port G IO register	PGIOR	R/W	H'0000	H'FFFFFF760	8, 16
Port G control register	PGCR	R/W	H'0000	H'FFFFFF762	8, 16
Port H IO register	PHIOR	R/W	H'0000	H'FFFFFF728	8, 16
Port H control register	PHCR	R/W	H'0000	H'FFFFFF72A	8, 16
Port J IO register	PJIOR	R/W	H'0000	H'FFFFFF766	8, 16
Port J control register H	PJCRH	R/W	H'0000	H'FFFFFF768	8, 16
Port J control register L	PJCRL	R/W	H'0000	H'FFFFFF76A	8, 16
Port K IO register	PKIOR	R/W	H'0000	H'FFFFFF770	8, 16
Port K control register H	PKCRH	R/W	H'0000	H'FFFFFF772	8, 16
Port K control register L	PKCRL	R/W	H'0000	H'FFFFFF774	8, 16
Port K invert register	PKIR	R/W	H'0000	H'FFFFFF776	8, 16

**Table 21.2 PFC Registers (cont)**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port L IO register	PLIOR	R/W	H'0000	H'FFFFFF756	8, 16
Port L control register H	PLCRH	R/W	H'0000	H'FFFFFF758	8, 16
Port L control register L	PLCRL	R/W	H'0000	H'FFFFFF75A	8, 16
Port L invert register	PLIR	R/W	H'0000	H'FFFFFF75C	8, 16

## 21.3 Register Descriptions

### 21.3.1 Port A IO Register (PAIOR)

Bit:	15	14	13	12	11	10	9	8
	PA15 IOR	PA14 IOR	PA13 IOR	PA12 IOR	PA11 IOR	PA10 IOR	PA9 IOR	PA8 IOR
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:	7	6	5	4	3	2	1	0
	PA7 IOR	PA6 IOR	PA5 IOR	PA4 IOR	PA3 IOR	PA2 IOR	PA1 IOR	PA0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port A IO register (PAIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port A. Bits PA15IOR to PA0IOR correspond to pins PA15/RxD0 to PA0/TI0A. PAIOR is enabled when port A pins function as general input/output pins (PA15 to PA0) or ATU-II input/output pins, and disabled otherwise. For bits 3 to 0, when ATU-II input capture input is selected, the PAIOR bits should be cleared to 0.

When port A pins function as PA15 to PA0 or ATU-II input/output pins, a pin becomes an output when the corresponding bit in PAIOR is set to 1, and an input when the bit is cleared to 0.

PAIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

### 21.3.2 Port A Control Registers H and L (PACRH, PACRL)

Port A control registers H and L (PACRH, PACRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port A. PACRH selects the functions of the pins for the upper 8 bits of port A, and PACRL selects the functions of the pins for the lower 8 bits.

PACRH and PACRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

#### Port A Control Register H (PACRH)

Bit:	15	14	13	12	11	10	9	8
	—	PA15MD	—	PA14MD	—	PA13MD	—	PA12MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	PA11MD1	PA11MD0	PA10MD1	PA10MD0	PA9MD1	PA9MD0	PA8MD1	PA8MD0
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 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PA15 Mode Bit (PA15MD): Selects the function of pin PA15/RxD0.

#### Bit 14: PA15MD Description

0	General input/output (PA15)	(Initial value)
1	Receive data input (RxD0)	

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PA14 Mode Bit (PA14MD): Selects the function of pin PA14/TxD0.

#### Bit 12: PA14MD Description

0	General input/output (PA14)	(Initial value)
1	Transmit data output (TxD0)	

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 10—PA13 Mode Bit (PA13MD): Selects the function of pin PA13/TIO5B.

Bit 10: PA13MD	Description	
0	General input/output (PA13)	(Initial value)
1	ATU-II input capture input/output compare output (TIO5B)	

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PA12 Mode Bit (PA12MD): Selects the function of pin PA12/TIO5A.

Bit 8: PA12MD	Description	
0	General input/output (PA12)	(Initial value)
1	ATU-II input capture input/output compare output (TIO5A)	

- Bits 7 and 6—PA11 Mode Bit 1 and 0 (PA11MD1, PA11MD0): Select the function of pin PA11/TIO4D/ADTO1B.

Bit 7: PA11MD1	Bit 6: PA11MD0	Description	
0	0	General input/output (PA11)	(Initial value)
0	1	ATU-II input capture input/output compare output (TIO4D)	
1	0	Setting prohibited	
1	1	Output compare 1B output (MTAD)	

- Bits 5 and 4—PA10 Mode Bit 1 and 0 (PA10MD1, PA10MD0): Select the function of pin PA10/TIO4C/ADTO1B.

Bit 5: PA10MD1	Bit 4: PA10MD0	Description	
0	0	General input/output (PA10)	(Initial value)
0	1	ATU-II input capture input/output compare output (TIO4C)	
1	0	Setting prohibited	
1	1	Output compare 1A output (MTAD)	

- Bits 3 and 2—PA9 Mode Bit 1 and 0 (PA9MD1, PA9MD0): Select the function of pin PA9/TIO4B/ADTO0B.

Bit 3: PA9MD1	Bit 2: PA9MD0	Description
0	0	General input/output (PA9) (Initial value)
0	1	ATU-II input capture input/output compare output (TIO4B)
1	0	Setting prohibited
1	1	Output compare 0B output (MTAD)

- Bits 1 and 0—PA8 Mode Bit 1 and 0 (PA8MD1, PA8MD0): Select the function of pin PA8/TIO4A/ADTO0A.

Bit 1: PA8MD1	Bit 0: PA8MD0	Description
0	0	General input/output (PA8) (Initial value)
0	1	ATU-II input capture input/output compare output (TIO4A)
1	0	Setting prohibited
1	1	Output compare 0A output (MTAD)

### Port A Control Register L (PACRL)

Bit:	15	14	13	12	11	10	9	8
	—	PA7MD	—	PA6MD	—	PA5MD	—	PA4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PA3MD	—	PA2MD	—	PA1MD	—	PA0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PA7 Mode Bit (PA7MD): Selects the function of pin PA7/TIO3D.

Bit 14: PA7MD	Description
0	General input/output (PA7) (Initial value)
1	ATU-II input capture input/output compare output (TIO3D)

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PA6 Mode Bit (PA6MD): Selects the function of pin PA6/TIO3C.

Bit 12: PA6MD	Description
0	General input/output (PA6) (Initial value)
1	ATU-II input capture input/output compare output (TIO3C)

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PA5 Mode Bit (PA5MD): Selects the function of pin PA5/TIO3B.

Bit 10: PA5MD	Description
0	General input/output (PA5) (Initial value)
1	ATU-II input capture input/output compare output (TIO3B)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PA4 Mode Bit (PA4MD): Selects the function of pin PA4/TIO3A.

Bit 8: PA4MD	Description
0	General input/output (PA4) (Initial value)
1	ATU-II input capture input/output compare output (TIO3A)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PA3 Mode Bit (PA3MD): Selects the function of pin PA3/TIO3D.

Bit 6: PA3MD	Description
0	General input/output (PA3) (Initial value)
1	ATU-II input capture input (TIO3D)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PA2 Mode Bit (PA2MD): Selects the function of pin PA2/TIO3C.

Bit 4: PA2MD	Description
0	General input/output (PA2) (Initial value)
1	ATU-II input capture input (TIO3C)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 2—PA1 Mode Bit (PA1MD): Selects the function of pin PA1/TI0B.

Bit 2: PA1MD	Description
0	General input/output (PA1) (Initial value)
1	ATU-II input capture input (TI0B)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PA0 Mode Bit (PA0MD): Selects the function of pin PA0/TI0A.

Bit 0: PA0MD	Description
0	General input/output (PA0) (Initial value)
1	ATU-II input capture input (TI0A)

### 21.3.3 Port B IO Register (PBIOR)

Bit:	15	14	13	12	11	10	9	8
	PB15 IOR	PB14 IOR	PB13 IOR	PB12 IOR	PB11 IOR	PB10 IOR	PB9 IOR	PB8 IOR
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:	7	6	5	4	3	2	1	0
	PB7 IOR	PB6 IOR	PB5 IOR	PB4 IOR	PB3 IOR	PB2 IOR	PB1 IOR	PB0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port B IO register (PBIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port B. Bits PB15IOR to PB0IOR correspond to pins PB15/PULS5/SCK2 to PB0/TO6A. PBIOR is enabled when port B pins function as general input/output pins (PB15 to PB0) or serial clock pins (SCK0, SCK1, SCK2), and disabled otherwise.

When port B pins function as PB15 to PB0 or SCK0, SCK1, and SCK2, a pin becomes an output when the corresponding bit in PBIOR is set to 1, and an input when the bit is cleared to 0.

PBIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

### 21.3.4 Port B Control Registers H and L (PBCRH, PBCRL)

Port B control registers H and L (PBCRH, PBCRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port B. PBCRH selects the functions of the pins for the upper 8 bits of port B, and PBCRL selects the functions of the pins for the lower 8 bits.

PBCRH and PBCRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

#### Port B Control Register H (PBCRH)

Bit:	15	14	13	12	11	10	9	8
	PB15 MD1	PB15 MD0	PB14 MD1	PB14 MD0	—	PB13 MD	PB12 MD1	PB12 MD0
Initial value:	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

Bit:	7	6	5	4	3	2	1	0
	PB11 MD1	PB11 MD0	PB10 MD1	PB10 MD0	PB9 MD1	PB9 MD0	PB8 MD1	PB8 MD0
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

- Bits 15 and 14—PB15 Mode Bits 1 and 0 (PB15MD1, PB15MD0): These bits select the function of pin PB15/PULS5/SCK2.

Bit 15: PB15MD1	Bit 14: PB15MD0	Description
0	0	General input/output (PB15) (Initial value)
	1	APC pulse output (PULS5)
1	0	Serial clock input/output (SCK2)
	1	Reserved (Do not set)



- Bits 13 and 12—PB14 Mode Bits 1 and 0 (PB14MD1, PB14MD0): These bits select the function of pin PB14/SCK1/TCLKB/T110.

Bit 13: PB14MD1	Bit 12: PB14MD0	Description
0	0	General input/output (PB14) (Initial value)
	1	Serial clock input/output (SCK1)
1	0	ATU-II clock input (TCLKB)
	1	ATU-II edge input (T110)

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PB13 Mode Bit (PB13MD): Selects the function of pin PB13/SCK0.

Bit 10: PB13MD	Description
0	General input/output (PB13) (Initial value)
1	Serial clock input/output (SCK0)

- Bits 9 and 8—PB12 Mode Bits 1 and 0 (PB12MD1, PB12MD0): These bits select the function of pin PB12/TCLKA/UBCTRG.

Bit 9: PB12MD1	Bit 8: PB12MD0	Description
0	0	General input/output (PB12) (Initial value)
	1	ATU-II clock input (TCLKA)
1	0	Trigger pulse output (UBCTRG)
	1	Reserved (Do not set)

- Bits 7 and 6—PB11 Mode Bits 1 and 0 (PB11MD1, PB11MD0): These bits select the function of pin PB11/RxD4/HRxD0/TO8H.

Bit 7: PB11MD1	Bit 6: PB11MD0	Description
0	0	General input/output (PB11) (Initial value)
	1	Receive data input (RxD4)
1	0	HCAN-II receive data input (HRxD0)
	1	ATU-II one-shot pulse output (TO8H)

- Bits 5 and 4—PB10 Mode Bits 1 and 0 (PB10MD1, PB10MD0): These bits select the function of pin PB10/TxD4/HTxD0/TO8G.

Bit 5: PB10MD1	Bit 4: PB10MD0	Description
0	0	General input/output (PB10) (Initial value)
	1	Transmit data output (TxD4)
1	0	HCAN-II transmit data output (HTxD0)
	1	ATU-II one-shot pulse output (TO8G)

- Bits 3 and 2—PB9 Mode Bits 1 and 0 (PB9MD1, PB9MD0): These bits select the function of pin PB9/RxD3/TO8F.

Bit 3: PB9MD1	Bit 2: PB9MD0	Description
0	0	General input/output (PB9) (Initial value)
	1	Receive data input (RxD3)
1	0	ATU-II one-shot pulse output (TO8F)
	1	Reserved (Do not set)

- Bits 1 and 0—PB8 Mode Bits 1 and 0 (PB8MD1, PB8MD0): These bits select the function of pin PB8/TxD3/TO8E.

Bit 1: PB8MD1	Bit 0: PB8MD0	Description
0	0	General input/output (PB8) (Initial value)
	1	Transmit data output (TxD3)
1	0	ATU-II one-shot pulse output (TO8E)
	1	Reserved (Do not set)

### Port B Control Register L (PBCRL)

Bit:	15	14	13	12	11	10	9	8
	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0
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:	7	6	5	4	3	2	1	0
	—	PB3MD	—	PB2MD	—	PB1MD	—	PB0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bits 15 and 14—PB7 Mode Bits 1 and 0 (PB7MD1, PB7MD0): These bits select the function of pin PB7/TO7D/TO8D.

Bit 15: PB7MD1	Bit 14: PB7MD0	Description
0	0	General input/output (PB7) (Initial value)
	1	ATU-II PWM output (TO7D)
1	0	ATU-II one-shot pulse output (TO8D)
	1	Reserved (Do not set)

- Bits 13 and 12—PB6 Mode Bits 1 and 0 (PB6MD1, PB6MD0): These bits select the function of pin PB6/TO7C/TO8C.

Bit 13: PB6MD1	Bit 12: PB6MD0	Description
0	0	General input/output (PB6) (Initial value)
	1	ATU-II PWM output (TO7C)
1	0	ATU-II one-shot pulse output (TO8C)
	1	Reserved (Do not set)

- Bits 11 and 10—PB5 Mode Bits 1 and 0 (PB5MD1, PB5MD0): These bits select the function of pin PB5/TO7B/TO8B.

Bit 11: PB5MD1	Bit 10: PB5MD0	Description
0	0	General input/output (PB5) (Initial value)
	1	ATU-II PWM output (TO7B)
1	0	ATU-II one-shot pulse output (TO8B)
	1	Reserved (Do not set)

- Bits 9 and 8—PB4 Mode Bits 1 and 0 (PB4MD1, PB4MD0): These bits select the function of pin PB4/TO7A/TO8A.

Bit 9: PB4MD1	Bit 8: PB4MD0	Description
0	0	General input/output (PB4) (Initial value)
	1	ATU-II PWM output (TO7A)
1	0	ATU-II one-shot pulse output (TO8A)
	1	Reserved (Do not set)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 6—PB3 Mode Bit (PB3MD): Selects the function of pin PB3/TO6D.

Bit 6: PB3MD	Description	
0	General input/output (PB3)	(Initial value)
1	ATU-II PWM output (TO6D)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PB2 Mode Bit (PB2MD): Selects the function of pin PB2/TO6C.

Bit 4: PB2MD	Description	
0	General input/output (PB2)	(Initial value)
1	ATU-II PWM output (TO6C)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PB1 Mode Bit (PB1MD): Selects the function of pin PB1/TO6B.

Bit 2: PB1MD	Description	
0	General input/output (PB1)	(Initial value)
1	ATU-II PWM output (TO6B)	

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PB0 Mode Bit (PB0MD): Selects the function of pin PB0/TO6A.

Bit 0: PB0MD	Description	
0	General input/output (PB0)	(Initial value)
1	ATU-II PWM output (TO6A)	

### 21.3.5 Port B Invert Register (PBIR)

Bit:	15	14	13	12	11	10	9	8
	PB15IR	PB14IR	PB13IR	—	PB11IR	PB10IR	PB9IR	PB8IR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PB7IR	PB6IR	PB5IR	PB4IR	PB3IR	PB2IR	PB1IR	PB0IR
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

The port B invert register (PBIR) is a 16-bit readable/writable register that sets the port B inversion function. Bits PB15IR to PB13IR and PB11IR to PB0IR correspond to pins PB15/PULS5/SCK2 to PB13/SCK0 and PB11/RxD4/HRxD0/TO8H to PB0/TO6A. PBIR is enabled when port B pins function as ATU-II outputs or serial clock pins, and disabled otherwise.

When port B pins function as ATU-II outputs or serial clock pins, the value of a pin is inverted when the corresponding bit in PBIR is set to 1.

PBIR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PBnIR	Description
0	Value is not inverted (Initial value)
1	Value is inverted

n = 15 to 13, 11 to 0

### 21.3.6 Port C IO Register (PCIOR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	—	—	—	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR
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

The port C IO register (PCIOR) is a 16-bit readable/writable register that selects the input/output direction of the five pins in port C. Bits PC4IOR to PC0IOR correspond to pins PC4/ $\overline{\text{IRQ}}_0$  to PC0/TxD1. PCIOR is enabled when port C pins function as general input/output pins (PC4 to PC0), and disabled otherwise.

When port C pins function as PC4 to PC0, a pin becomes an output when the corresponding bit in PCIOR is set to 1, and an input when the bit is cleared to 0.

PCIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

### 21.3.7 Port C Control Register (PCCR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	PC4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W
Bit:	7	6	5	4	3	2	1	0
	—	PC3MD	—	PC2MD	—	PC1MD	—	PC0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

The port C control register (PCCR) is a 16-bit readable/writable register that selects the functions of the five multiplex pins in port C.

PCCR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bits 15 to 9—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 8—PC4 Mode Bit (PC4MD): Selects the function of pin PC4/ $\overline{\text{IRQ0}}$ .

Bit 8: PC4MD	Description
0	General input/output (PC4) (Initial value)
1	Interrupt request input ( $\overline{\text{IRQ0}}$ )

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PC3 Mode Bit (PC3MD): Selects the function of pin PC3/RxD2.

Bit 6: PC3MD	Description
0	General input/output (PC3) (Initial value)
1	Receive data input (RxD2)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PC2 Mode Bit (PC2MD): Selects the function of pin PC2/TxD2.

Bit 4: PC2MD	Description
0	General input/output (PC2) (Initial value)
1	Transmit data output (TxD2)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 2—PC1 Mode Bit (PC1MD): Selects the function of pin PC1/RxD1.

Bit 2: PC1MD	Description
0	General input/output (PC1) (Initial value)
1	Receive data input (RxD1)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PC0 Mode Bit (PC0MD): Selects the function of pin PC0/TxD1.

Bit 0: PC0MD	Description
0	General input/output (PC0) (Initial value)
1	Transmit data output (TxD1)

### 21.3.8 Port D IO Register (PDIOR)

Bit:	15	14	13	12	11	10	9	8
	—	—	PD13 IOR	PD12 IOR	PD11 IOR	PD10 IOR	PD9 IOR	PD8 IOR
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:	7	6	5	4	3	2	1	0
	PD7 IOR	PD6 IOR	PD5 IOR	PD4 IOR	PD3 IOR	PD2 IOR	PD1 IOR	PD0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port D IO register (PDIOR) is a 16-bit readable/writable register that selects the input/output direction of the 14 pins in port D. Bits PD13IOR to PD0IOR correspond to pins PD13/PULS6/HTxD0/HTxD1 to PD0/TIO1A. PDIOR is enabled when port D pins function as general input/output pins (PD13 to PD0) or timer input/output pins, and disabled otherwise.

When port D pins function as PD13 to PD0 or timer input/output pins, a pin becomes an output when the corresponding bit in PDIOR is set to 1, and an input when the bit is cleared to 0.

PDIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.



### 21.3.9 Port D Control Registers H and L (PDCRH, PDCRL)

Port D control registers H and L (PDCRH, PDCRL) are 16-bit readable/writable registers that select the functions of the 14 multiplex pins in port D. PDCRH selects the functions of the pins for the upper 6 bits of port D, and PDCRL selects the functions of the pins for the lower 8 bits.

PDCRH and PDCRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

#### Port D Control Register H (PDCRH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	PD13 MD1	PD13 MD0	—	PD12 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R	R/W
Bit:	7	6	5	4	3	2	1	0
	—	PD11 MD	—	PD10 MD	—	PD9 MD	—	PD8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bits 15 to 12—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 11 and 10—PD13 Mode Bits 1 and 0 (PD13MD1, PD13MD0): These bits select the function of pin PD13/PULS6/HTxD0/HTxD1.

Bit 11: PD13MD1	Bit 10: PD13MD0	Description
0	0	General input/output (PD13) (Initial value)
	1	APC pulse output (PULS6)
1	0	HCAN-II transmit data output (HTxD0)
	1	HCAN-II transmit data output (HTxD1)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PD12 Mode Bit (PD12MD): Selects the function of pin PD12/PULS4.

Bit 8: PD12MD	Description
0	General input/output (PD12) (Initial value)
1	APC pulse output (PULS4)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PD11 Mode Bit (PD11MD): Selects the function of pin PD11/PULS3.

Bit 6: PD11MD	Description	
0	General input/output (PD11)	(Initial value)
1	APC pulse output (PULS3)	

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PD10 Mode Bit (PD10MD): Selects the function of pin PD10/PULS2.

Bit 4: PD10MD	Description	
0	General input/output (PD10)	(Initial value)
1	APC pulse output (PULS2)	

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PD9 Mode Bit (PD9MD): Selects the function of pin PD9/PULS1.

Bit 2: PD9MD	Description	
0	General input/output (PD9)	(Initial value)
1	APC pulse output (PULS1)	

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PD8 Mode Bit (PD8MD): Selects the function of pin PD8/PULS0.

Bit 0: PD8MD	Description	
0	General input/output (PD8)	(Initial value)
1	APC pulse output (PULS0)	

## Port D Control Register L (PDCRL)

Bit:	15	14	13	12	11	10	9	8
	—	PD7MD	—	PD6MD	—	PD5MD	—	PD4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PD3MD	—	PD2MD	—	PD1MD	—	PD0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PD7 Mode Bit (PD7MD): Selects the function of pin PD7/TIO1H.

Bit 14: PD7MD	Description
0	General input/output (PD7) (Initial value)
1	ATU-II input capture input/output compare output (TIO1H)

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PD6 Mode Bit (PD6MD): Selects the function of pin PD6/TIO1G.

Bit 12: PD6MD	Description
0	General input/output (PD6) (Initial value)
1	ATU-II input capture input/output compare output (TIO1G)

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PD5 Mode Bit (PD5MD): Selects the function of pin PD5/TIO1F.

Bit 10: PD5MD	Description
0	General input/output (PD5) (Initial value)
1	ATU-II input capture input/output compare output (TIO1F)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 8—PD4 Mode Bit (PD4MD): Selects the function of pin PD4/TIO1E.

Bit 8: PD4MD	Description
0	General input/output (PD4) (Initial value)
1	ATU-II input capture input/output compare output (TIO1E)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PD3 Mode Bit (PD3MD): Selects the function of pin PD3/TIO1D.

Bit 6: PD3MD	Description
0	General input/output (PD3) (Initial value)
1	ATU-II input capture input/output compare output (TIO1D)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PD2 Mode Bit (PD2MD): Selects the function of pin PD2/TIO1C.

Bit 4: PD2MD	Description
0	General input/output (PD2) (Initial value)
1	ATU-II input capture input/output compare output (TIO1C)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PD1 Mode Bit (PD1MD): Selects the function of pin PD1/TIO1B.

Bit 2: PD1MD	Description
0	General input/output (PD1) (Initial value)
1	ATU-II input capture input/output compare output (TIO1B)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PD0 Mode Bit (PD0MD): Selects the function of pin PD0/TIO1A.

Bit 0: PD0MD	Description
0	General input/output (PD0) (Initial value)
1	ATU-II input capture input/output compare output (TIO1A)

### 21.3.10 Port E IO Register (PEIOR)

Bit:	15	14	13	12	11	10	9	8
	PE15 IOR	PE14 IOR	PE13 IOR	PE12 IOR	PE11 IOR	PE10 IOR	PE9 IOR	PE8 IOR
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:	7	6	5	4	3	2	1	0
	PE7 IOR	PE6 IOR	PE5 IOR	PE4 IOR	PE3 IOR	PE2 IOR	PE1 IOR	PE0 IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port E IO register (PEIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port E. Bits PE15IOR to PE0IOR correspond to pins PE15/A15 to PE0/A0. PEIOR is enabled when port E pins function as general input/output pins (PE15 to PE0), and disabled otherwise.

When port E pins function as PE15 to PE0, a pin becomes an output when the corresponding bit in PEIOR is set to 1, and an input when the bit is cleared to 0.

PEIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

### 21.3.11 Port E Control Register (PECR)

Bit:	15	14	13	12	11	10	9	8
	PE15 MD	PE14 MD	PE13 MD	PE12 MD	PE11 MD	PE10 MD	PE9 MD	PE8 MD
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:	7	6	5	4	3	2	1	0
	PE7 MD	PE6 MD	PE5 MD	PE4 MD	PE3 MD	PE2 MD	PE1 MD	PE0 MD
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

The port E control register (PECR) is a 16-bit readable/writable register that selects the functions of the 16 multiplex pins in port E. PECR settings are not valid in all operating modes.

- Expanded mode with on-chip ROM disabled  
Port E pins function as address output pins, and PECR settings are invalid.
- Expanded mode with on-chip ROM enabled  
Port E pins are multiplexed as address output pins and general input/output pins. PECR settings are valid.
- Single-chip mode  
Port E pins function as general input/output pins, and PECR settings are invalid.

PECR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bit 15—PE15 Mode Bit (PE15MD): Selects the function of pin PE15/A15.

Bit 15: PE15MD	Description		
	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A15) (Initial value)	General input/output (PE15) (Initial value)	General input/output (PE15) (Initial value)
1	Address output (A15)	Address output (A15)	General input/output (PE15)

- Bit 14—PE14 Mode Bit (PE14MD): Selects the function of pin PE14/A14.

Description			
Bit 14: PE14MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A14) (Initial value)	General input/output (PE14) (Initial value)	General input/output (PE14) (Initial value)
1	Address output (A14)	Address output (A14)	General input/output (PE14)

- Bit 13—PE13 Mode Bit (PE13MD): Selects the function of pin PE13/A13.

Description			
Bit 13: PE13MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A13) (Initial value)	General input/output (PE13) (Initial value)	General input/output (PE13) (Initial value)
1	Address output (A13)	Address output (A13)	General input/output (PE13)

- Bit 12—PE12 Mode Bit (PE12MD): Selects the function of pin PE12/A12.

Description			
Bit 12: PE12MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A12) (Initial value)	General input/output (PE12) (Initial value)	General input/output (PE12) (Initial value)
1	Address output (A12)	Address output (A12)	General input/output (PE12)

- Bit 11—PE11 Mode Bit (PE11MD): Selects the function of pin PE11/A11.

Description			
Bit 11: PE11MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A11) (Initial value)	General input/output (PE11) (Initial value)	General input/output (PE11) (Initial value)
1	Address output (A11)	Address output (A11)	General input/output (PE11)

- Bit 10—PE10 Mode Bit (PE10MD): Selects the function of pin PE10/A10.

Description			
Bit 10: PE10MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A10) (Initial value)	General input/output (PE10) (Initial value)	General input/output (PE10) (Initial value)
1	Address output (A10)	Address output (A10)	General input/output (PE10)

- Bit 9—PE9 Mode Bit (PE9MD): Selects the function of pin PE9/A9.

Description			
Bit 9: PE9MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A9) (Initial value)	General input/output (PE9) (Initial value)	General input/output (PE9) (Initial value)
1	Address output (A9)	Address output (A9)	General input/output (PE9)

- Bit 8—PE8 Mode Bit (PE8MD): Selects the function of pin PE8/A8.

Description			
Bit 8: PE8MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A8) (Initial value)	General input/output (PE8) (Initial value)	General input/output (PE8) (Initial value)
1	Address output (A8)	Address output (A8)	General input/output (PE8)

- Bit 7—PE7 Mode Bit (PE7MD): Selects the function of pin PE7/A7.

Description			
Bit 7: PE7MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A7) (Initial value)	General input/output (PE7) (Initial value)	General input/output (PE7) (Initial value)
1	Address output (A7)	Address output (A7)	General input/output (PE7)



- Bit 6—PE6 Mode Bit (PE6MD): Selects the function of pin PE6/A6.

Description			
Bit 6: PE6MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A6) (Initial value)	General input/output (PE6) (Initial value)	General input/output (PE6) (Initial value)
1	Address output (A6)	Address output (A6)	General input/output (PE6)

- Bit 5—PE5 Mode Bit (PE5MD): Selects the function of pin PE5/A5.

Description			
Bit 5: PE5MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A5) (Initial value)	General input/output (PE5) (Initial value)	General input/output (PE5) (Initial value)
1	Address output (A5)	Address output (A5)	General input/output (PE5)

- Bit 4—PE4 Mode Bit (PE4MD): Selects the function of pin PE4/A4.

Description			
Bit 4: PE4MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A4) (Initial value)	General input/output (PE4) (Initial value)	General input/output (PE4) (Initial value)
1	Address output (A4)	Address output (A4)	General input/output (PE4)

- Bit 3—PE3 Mode Bit (PE3MD): Selects the function of pin PE3/A3.

Description			
Bit 3: PE3MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A3) (Initial value)	General input/output (PE3) (Initial value)	General input/output (PE3) (Initial value)
1	Address output (A3)	Address output (A3)	General input/output (PE3)

- Bit 2—PE2 Mode Bit (PE2MD): Selects the function of pin PE2/A2.

<b>Description</b>			
<b>Bit 2: PE2MD</b>	<b>Expanded Mode with ROM Disabled</b>	<b>Expanded Mode with ROM Enabled</b>	<b>Single-Chip Mode</b>
0	Address output (A2) (Initial value)	General input/output (PE2) (Initial value)	General input/output (PE2) (Initial value)
1	Address output (A2)	Address output (A2)	General input/output (PE2)

- Bit 1—PE1 Mode Bit (PE1MD): Selects the function of pin PE1/A1.

<b>Description</b>			
<b>Bit 1: PE1MD</b>	<b>Expanded Mode with ROM Disabled</b>	<b>Expanded Mode with ROM Enabled</b>	<b>Single-Chip Mode</b>
0	Address output (A1) (Initial value)	General input/output (PE1) (Initial value)	General input/output (PE1) (Initial value)
1	Address output (A1)	Address output (A1)	General input/output (PE1)

- Bit 0—PE0 Mode Bit (PE0MD): Selects the function of pin PE0/A0.

<b>Description</b>			
<b>Bit 0: PE0MD</b>	<b>Expanded Mode with ROM Disabled</b>	<b>Expanded Mode with ROM Enabled</b>	<b>Single-Chip Mode</b>
0	Address output (A0) (Initial value)	General input/output (PE0) (Initial value)	General input/output (PE0) (Initial value)
1	Address output (A0)	Address output (A0)	General input/output (PE0)

### 21.3.12 Port F IO Register (PFIOR)

Bit:	15	14	13	12	11	10	9	8
	PF15 IOR	PF14 IOR	PF13 IOR	PF12 IOR	PF11 IOR	PF10 IOR	PF9 IOR	PF8 IOR
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:	7	6	5	4	3	2	1	0
	PF7 IOR	PF6 IOR	PF5 IOR	PF4 IOR	PF3 IOR	PF2 IOR	PF1 IOR	PF0 IOR
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

The port F IO register (PFIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port F. Bits PF15IOR to PF0IOR correspond to pins PF15/ $\overline{\text{BREQ}}$  to PF0/A16. PFIOR is enabled when port F pins function as general input/output pins (PF15 to PF0), and disabled otherwise.

When port F pins function as PF15 to PF0, a pin becomes an output when the corresponding bit in PFIOR is set to 1, and an input when the bit is cleared to 0.

PFIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

### 21.3.13 Port F Control Registers H and L (PFCRH, PFCRL)

Port F control registers H and L (PFCRH, PFCRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port F and the function of the CK pin. PFCRH selects the functions of the pins for the upper 8 bits of port F, and PFCRL selects the functions of the pins for the lower 8 bits.

PFCRH and PFCRL are initialized to H'0015 and H'5000, respectively, by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

#### Port F Control Register H (PFCRH)

Bit:	15	14	13	12	11	10	9	8
	CKHIZ	PF15MD	—	PF14MD	—	PF13MD	—	PF12MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R/W	R	R/W
Bit:	7	6	5	4	3	2	1	0
	—	PF11MD	—	PF10MD	—	PF9MD	—	PF8MD
Initial value:	0	0	0	1	0	1	0	1
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—CKHIZ Bit: Selects the function of pin CK.

Bit: CKHIZ	Description
0	CK pin output (Initial value)
1	CK pin Hi-Z

- Bit 14—PF15 Mode Bit (PF15MD): Selects the function of pin PF15/ $\overline{\text{BREQ}}$ .

Bit 14: PF15MD	Description	
	Expanded Mode	Single-Chip Mode
0	General input/output (PF15) (Initial value)	General input/output (PF15) (Initial value)
1	Bus request input ( $\overline{\text{BREQ}}$ )	General input/output (PF15)

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 12—PF14 Mode Bit (PF14MD): Selects the function of pin PF14/ $\overline{\text{BACK}}$ .

Bit 12: PF14MD	Description	
	Expanded Mode	Single-Chip Mode
0	General input/output (PF14) (Initial value)	General input/output (PF14) (Initial value)
1	Bus acknowledge output ( $\overline{\text{BACK}}$ )	General input/output (PF14)

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PF13 Mode Bit (PF13MD): Selects the function of pin PF13/ $\overline{\text{CS3}}$ .

Bit 10: PF13MD	Description	
	Expanded Mode	Single-Chip Mode
0	General input/output (PF13) (Initial value)	General input/output (PF13) (Initial value)
1	Chip select output ( $\overline{\text{CS3}}$ )	General input/output (PF13)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PF12 Mode Bit (PF12MD): Selects the function of pin PF12/ $\overline{\text{CS2}}$ .

Bit 8: PF12MD	Description	
	Expanded Mode	Single-Chip Mode
0	General input/output (PF12) (Initial value)	General input/output (PF12) (Initial value)
1	Chip select output ( $\overline{\text{CS2}}$ )	General input/output (PF12)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PF11 Mode Bit (PF11MD): Selects the function of pin PF11/ $\overline{\text{CS1}}$ .

Bit 6: PF11MD	Description	
	Expanded Mode	Single-Chip Mode
0	General input/output (PF11) (Initial value)	General input/output (PF11) (Initial value)
1	Chip select output ( $\overline{\text{CS1}}$ )	General input/output (PF11)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 4—PF10 Mode Bit (PF10MD): Selects the function of pin PF10/ $\overline{CS0}$ .

Bit 4: PF10MD	Description	
	Expanded Mode	Single-Chip Mode
0	General input/output (PF10)	General input/output (PF10)
1	Chip select output ( $\overline{CS0}$ ) (Initial value)	General input/output (PF10) (Initial value)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PF9 Mode Bit (PF9MD): Selects the function of pin PF9/ $\overline{RD}$ .

Bit 2: PF9MD	Description	
	Expanded Mode	Single-Chip Mode
0	General input/output (PF9)	General input/output (PF9)
1	Read output ( $\overline{RD}$ ) (Initial value)	General input/output (PF9) (Initial value)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PF8 Mode Bit (PF8MD): Selects the function of pin PF8/ $\overline{WAIT}$ .

Bit 0: PF8MD	Description	
	Expanded Mode	Single-Chip Mode
0	General input/output (PF8)	General input/output (PF8)
1	Wait state input ( $\overline{WAIT}$ ) (Initial value)	General input/output (PF8) (Initial value)

### Port F Control Register L (PFCRL)

Bit:	15	14	13	12	11	10	9	8
	—	PF7MD	—	PF6MD	PF5MD1	PF5MD0	—	PF4MD
Initial value:	0	1	0	1	0	0	0	0
R/W:	R	R/W	R	R/W	R/W	R/W	R	R/W
Bit:	7	6	5	4	3	2	1	0
	—	PF3MD	—	PF2MD	—	PF1MD	—	PF0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 14—PF7 Mode Bit (PF7MD): Selects the function of pin PF7/ $\overline{\text{WRH}}$ .

Bit 14: PF7MD	Description	
	Expanded Mode	Single-Chip Mode
0	General input/output (PF7)	General input/output (PF7)
1	Upper write ( $\overline{\text{WRH}}$ ) (Initial value)	General input/output (PF7) (Initial value)

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PF6 Mode Bit (PF6MD): Selects the function of pin PF6/ $\overline{\text{WRL}}$ .

Bit 12: PF6MD	Description	
	Expanded Mode	Single-Chip Mode
0	General input/output (PF6)	General input/output (PF6)
1	Lower write ( $\overline{\text{WRL}}$ ) (Initial value)	General input/output (PF6) (Initial value)

- Bits 11 and 10—PF5 Mode Bits 1 and 0 (PF5MD1, PF5MD0): These bits select the function of pin PF5/A21/ $\overline{\text{POD}}$ .

		Description		
Bit 11: PF5MD1	Bit 10: PF5MD0	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	0	Address output (A21) (Initial value)	General input/output (PF5) (Initial value)	General input/output (PF5) (Initial value)
	1	Address output (A21)	Address output (A21)	General input/output (PF5)
1	0	Address output (A21)	Port output disable input ( $\overline{\text{POD}}$ )	Port output disable input ( $\overline{\text{POD}}$ )
	1	Reserved (Do not set)	Reserved (Do not set)	Reserved (Do not set)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 8—PF4 Mode Bit (PF4MD): Selects the function of pin PF4/A20.

Description			
Bit 8: PF4MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A20) (Initial value)	General input/output (PF4) (Initial value)	General input/output (PF4) (Initial value)
1	Address output (A20)	Address output (A20)	General input/output (PF4)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PF3 Mode Bit (PF3MD): Selects the function of pin PF3/A19.

Description			
Bit 6: PF3MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A19) (Initial value)	General input/output (PF3) (Initial value)	General input/output (PF3) (Initial value)
1	Address output (A19)	Address output (A19)	General input/output (PF3)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PF2 Mode Bit (PF2MD): Selects the function of pin PF2/A18.

Description			
Bit 4: PF2MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A18) (Initial value)	General input/output (PF2) (Initial value)	General input/output (PF2) (Initial value)
1	Address output (A18)	Address output (A18)	General input/output (PF2)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PF1 Mode Bit (PF1MD): Selects the function of pin PF1/A17.

Description			
Bit 2: PF1MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A17) (Initial value)	General input/output (PF1) (Initial value)	General input/output (PF1) (Initial value)
1	Address output (A17)	Address output (A17)	General input/output (PF1)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.



- Bit 0—PF0 Mode Bit (PF0MD): Selects the function of pin PF0/A16.

Bit 0: PF0MD	Description		
	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Address output (A16) (Initial value)	General input/output (PF0) (Initial value)	General input/output (PF0) (Initial value)
1	Address output (A16)	Address output (A16)	General input/output (PF0)

### 21.3.14 Port G IO Register (PGIOR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	PG3IOR	PG2IOR	PG1IOR	PG0IOR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

The port G IO register (PGIOR) is a 16-bit readable/writable register that selects the input/output direction of the four pins in port G. Bits PG3IOR to PG0IOR correspond to pins  $\overline{\text{PG3}}/\overline{\text{IRQ3}}/\overline{\text{ADTRG0}}$  to PG0/PULS7/HRxD0/HRxD1.

When port G pins function as PG3 to PG0, a pin becomes an output when the corresponding bit in PGIOR is set to 1, and an input when the bit is cleared to 0.

PGIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

### 21.3.15 Port G Control Register (PGCR)

The port G control register (PGCR) is a 16-bit readable/writable register that selects the functions of the four multiplex pins in port G.

PGCR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	PG3MD1	PG3MD0	PG2MD1	PG2MD0	—	PG1MD	PG0MD1	PG0MD0
Initial value:	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

- Bits 15 to 8—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 7 and 6—PG3 Mode Bits 1 and 0 (PG3MD1, PG3MD0): These bits select the function of pin PG3/ $\overline{\text{IRQ3}}$ / $\overline{\text{ADTRG0}}$ .

Bit 7: PG3MD1	Bit 6: PG3MD0	Description
0	0	General input/output (PG3) (Initial value)
	1	Interrupt request input ( $\overline{\text{IRQ3}}$ )
1	0	A/D conversion trigger input ( $\overline{\text{ADTRG0}}$ )
	1	Reserved (Do not set)

- Bits 5 and 4—PG2 Mode Bits 1 and 0 (PG2MD1, PG2MD0): These bits select the function of pin PG2/ $\overline{\text{IRQ2}}$ / $\overline{\text{ADEND}}$ .

Bit 5: PG2MD1	Bit 4: PG2MD0	Description
0	0	General input/output (PG2) (Initial value)
	1	Interrupt request input ( $\overline{\text{IRQ2}}$ )
1	0	A/D conversion end output (ADEND)
	1	Reserved (Do not set)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 2—PG1 Mode Bit (PG1MD): Selects the function of pin PG1/ $\overline{\text{IRQ1}}$ .

Bit 2: PG1MD	Description
0	General input/output (PG1) (Initial value)
1	Interrupt request input ( $\overline{\text{IRQ1}}$ )

- Bits 1 and 0—PG0 Mode Bits 1 and 0 (PG0MD1, PG2MD0): These bits select the function of pin PG0/PULS7/HRxD0/HRxD1.

Bit 1: PG0MD1	Bit 0: PG0MD0	Description
0	0	General input/output (PG0) (Initial value)
	1	APC pulse output (PULS7)
1	0	HCAN-II receive data input (HRxD0)
	1	HCAN-II receive data input (HRxD1)

### 21.3.16 Port H IO Register (PHIOR)

Bit:	15	14	13	12	11	10	9	8
	PH15 IOR	PH14 IOR	PH13 IOR	PH12 IOR	PH11 IOR	PH10 IOR	PH9 IOR	PH8 IOR
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:	7	6	5	4	3	2	1	0
	PH7 IOR	PH6 IOR	PH5 IOR	PH4 IOR	PH3 IOR	PH2 IOR	PH1 IOR	PH0 IOR
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

The port H IO register (PHIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port H. Bits PH15IOR to PH0IOR correspond to pins PH15/D15 to PH0/D0. PHIOR is enabled when port H pins function as general input/output pins (PH15 to PH0), and disabled otherwise.

When port H pins function as PH15 to PH0, a pin becomes an output when the corresponding bit in PHIOR is set to 1, and an input when the bit is cleared to 0.

PHIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

### 21.3.17 Port H Control Register (PHCR)

Bit:	15	14	13	12	11	10	9	8
	PH15 MD	PH14 MD	PH13 MD	PH12 MD	PH11 MD	PH10 MD	PH9 MD	PH8 MD
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:	7	6	5	4	3	2	1	0
	PH7 MD	PH6 MD	PH5 MD	PH4 MD	PH3 MD	PH2 MD	PH1 MD	PH0 MD
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

The port H control register (PHCR) is a 16-bit readable/writable register that selects the functions of the 16 multiplex pins in port H. PHCR settings are not valid in all operating modes.

1. Expanded mode with on-chip ROM disabled (area 0: 8-bit bus)  
Port H pins D0 to D7 function as data input/output pins, and PHCR settings are invalid.
2. Expanded mode with on-chip ROM disabled (area 0: 16-bit bus)  
Port H pins function as data input/output pins, and PHCR settings are invalid.
3. Expanded mode with on-chip ROM enabled  
Port H pins are multiplexed as data input/output pins and general input/output pins. PHCR settings are valid.
4. Single-chip mode  
Port H pins function as general input/output pins, and PHCR settings are invalid.

PHCR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bit 15—PH15 Mode Bit (PH15MD): Selects the function of pin PH15/D15.

#### Description

Bit 15: PH15MD	Expanded Mode		Expanded Mode	
	with ROM Disabled Area 0: 8 Bits	with ROM Disabled Area 0: 16 Bits	with ROM Enabled	Single-Chip Mode
0	General input/output (PH15) (Initial value)	Data input/output (D15) (Initial value)	General input/output (PH15) (Initial value)	General input/output (PH15) (Initial value)
1	Data input/output (D15)	Data input/output (D15)	Data input/output (D15)	General input/output (PH15)

- Bit 14—PH14 Mode Bit (PH14MD): Selects the function of pin PH14/D14.

#### Description

Bit 14: PH14MD	Expanded Mode		Expanded Mode	
	with ROM Disabled Area 0: 8 Bits	with ROM Disabled Area 0: 16 Bits	with ROM Enabled	Single-Chip Mode
0	General input/output (PH14) (Initial value)	Data input/output (D14) (Initial value)	General input/output (PH14) (Initial value)	General input/output (PH14) (Initial value)
1	Data input/output (D14)	Data input/output (D14)	Data input/output (D14)	General input/output (PH14)

- Bit 13—PH13 Mode Bit (PH13MD): Selects the function of pin PH13/D13.

#### Description

Bit 13: PH13MD	Expanded Mode		Expanded Mode	
	with ROM Disabled Area 0: 8 Bits	with ROM Disabled Area 0: 16 Bits	with ROM Enabled	Single-Chip Mode
0	General input/output (PH13) (Initial value)	Data input/output (D13) (Initial value)	General input/output (PH13) (Initial value)	General input/output (PH13) (Initial value)
1	Data input/output (D13)	Data input/output (D13)	Data input/output (D13)	General input/output (PH13)

- Bit 12—PH12 Mode Bit (PH12MD): Selects the function of pin PH12/D12.

#### Description

Bit 12: PH12MD	Description			
	Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH12) (Initial value)	Data input/output (D12) (Initial value)	General input/output (PH12) (Initial value)	General input/output (PH12) (Initial value)
1	Data input/output (D12)	Data input/output (D12)	Data input/output (D12)	General input/output (PH12)

- Bit 11—PH11 Mode Bit (PH11MD): Selects the function of pin PH11/D11.

#### Description

Bit 11: PH11MD	Description			
	Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH11) (Initial value)	Data input/output (D11) (Initial value)	General input/output (PH11) (Initial value)	General input/output (PH11) (Initial value)
1	Data input/output (D11)	Data input/output (D11)	Data input/output (D11)	General input/output (PH11)

- Bit 10—PH10 Mode Bit (PH10MD): Selects the function of pin PH10/D10.

#### Description

Bit 10: PH10MD	Description			
	Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
0	General input/output (PH10) (Initial value)	Data input/output (D10) (Initial value)	General input/output (PH10) (Initial value)	General input/output (PH10) (Initial value)
1	Data input/output (D10)	Data input/output (D10)	Data input/output (D10)	General input/output (PH10)

- Bit 9—PH9 Mode Bit (PH9MD): Selects the function of pin PH9/D9.

#### Description

Bit 9: PH9MD	Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
	0	General input/output (PH9) (Initial value)	Data input/output (D9) (Initial value)	General input/output (PH9) (Initial value)
1	Data input/output (D9)	Data input/output (D9)	Data input/output (D9)	General input/output (PH9)

- Bit 8—PH8 Mode Bit (PH8MD): Selects the function of pin PH8/D8.

#### Description

Bit 8: PH8MD	Expanded Mode with ROM Disabled Area 0: 8 Bits	Expanded Mode with ROM Disabled Area 0: 16 Bits	Expanded Mode with ROM Enabled	Single-Chip Mode
	0	General input/output (PH8) (Initial value)	Data input/output (D8) (Initial value)	General input/output (PH8) (Initial value)
1	Data input/output (D8)	Data input/output (D8)	Data input/output (D8)	General input/output (PH8)

- Bit 7—PH7 Mode Bit (PH7MD): Selects the function of pin PH7/D7.

#### Description

Bit 7: PH7MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
	0	Data input/output (D7) (Initial value)	General input/output (PH7) (Initial value)
1	Data input/output (D7)	Data input/output (D7)	General input/output (PH7)

- Bit 6—PH6 Mode Bit (PH6MD): Selects the function of pin PH6/D6.

Description			
Bit 6: PH6MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D6) (Initial value)	General input/output (PH6) (Initial value)	General input/output (PH6) (Initial value)
1	Data input/output (D6)	Data input/output (D6)	General input/output (PH6)

- Bit 5—PH5 Mode Bit (PH5MD): Selects the function of pin PH5/D5.

Description			
Bit 5: PH5MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D5) (Initial value)	General input/output (PH5) (Initial value)	General input/output (PH5) (Initial value)
1	Data input/output (D5)	Data input/output (D5)	General input/output (PH5)

- Bit 4—PH4 Mode Bit (PH4MD): Selects the function of pin PH4/D4.

Description			
Bit 4: PH4MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D4) (Initial value)	General input/output (PH4) (Initial value)	General input/output (PH4) (Initial value)
1	Data input/output (D4)	Data input/output (D4)	General input/output (PH4)

- Bit 3—PH3 Mode Bit (PH3MD): Selects the function of pin PH3/D3.

Description			
Bit 3: PH3MD	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D3) (Initial value)	General input/output (PH3) (Initial value)	General input/output (PH3) (Initial value)
1	Data input/output (D3)	Data input/output (D3)	General input/output (PH3)



- Bit 2—PH2 Mode Bit (PH2MD): Selects the function of pin PH2/D2.

Bit 2: PH2MD	Description		
	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D2) (Initial value)	General input/output (PH2) (Initial value)	General input/output (PH2) (Initial value)
1	Data input/output (D2)	Data input/output (D2)	General input/output (PH2)

- Bit 1—PH1 Mode Bit (PH1MD): Selects the function of pin PH1/D1.

Bit 1: PH1MD	Description		
	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D1) (Initial value)	General input/output (PH1) (Initial value)	General input/output (PH1) (Initial value)
1	Data input/output (D1)	Data input/output (D1)	General input/output (PH1)

- Bit 0—PH0 Mode Bit (PH0MD): Selects the function of pin PH0/D0.

Bit 0: PH0MD	Description		
	Expanded Mode with ROM Disabled	Expanded Mode with ROM Enabled	Single-Chip Mode
0	Data input/output (D0) (Initial value)	General input/output (PH0) (Initial value)	General input/output (PH0) (Initial value)
1	Data input/output (D0)	Data input/output (D0)	General input/output (PH0)

### 21.3.18 Port J IO Register (PJIOR)

Bit:	15	14	13	12	11	10	9	8
	PJ15 IOR	PJ14 IOR	PJ13 IOR	PJ12 IOR	PJ11 IOR	PJ10 IOR	PJ9 IOR	PJ8 IOR
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:	7	6	5	4	3	2	1	0
	PJ7 IOR	PJ6 IOR	PJ5 IOR	PJ4 IOR	PJ3 IOR	PJ2 IOR	PJ1 IOR	PJ0 IOR
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

The port J IO register (PJIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port J. Bits PJ15IOR to PJ0IOR correspond to pins PJ15/TI9F to PJ0/TIO2A. PJIOR is enabled when port J pins function as general input/output pins (PJ15 to PJ0) or ATU-II input/output pins, and disabled otherwise. When ATU-II event counter input is selected, however, the bits 10 to 15 of the PJIOR should be cleared to 0.

When port J pins function as PJ15 to PJ0 or ATU-II input/output pins, a pin becomes an output when the corresponding bit in PJIOR is set to 1, and an input when the bit is cleared to 0.

PJIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

### 21.3.19 Port J Control Registers H and L (PJCRH, PJCRL)

Port J control registers H and L (PJCRH, PJCRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port J. PJCRH selects the functions of the pins for the upper 8 bits of port J, and PJCRL selects the functions of the pins for the lower 8 bits.

PJCRH and PJCRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

#### Port J Control Register H (PJCRH)

Bit:	15	14	13	12	11	10	9	8
	—	PJ15MD	—	PJ14MD	—	PJ13MD	—	PJ12MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PJ11MD	—	PJ10MD	—	PJ9MD	—	PJ8MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PJ15 Mode Bit (PJ15MD): Selects the function of pin PJ15/TI9F.

#### Bit 14: PJ15MD Description

0	General input/output (PJ15)	(Initial value)
1	ATU-II event counter input (TI9F)	

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PJ14 Mode Bit (PJ14MD): Selects the function of pin PJ14/TI9E.

#### Bit 12: PJ14MD Description

0	General input/output (PJ14)	(Initial value)
1	ATU-II event counter input (TI9E)	

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 10—PJ13 Mode Bit (PJ13MD): Selects the function of pin PJ13/TI9D.

Bit 10: PJ13MD	Description
0	General input/output (PJ13) (Initial value)
1	ATU-II event counter input (TI9D)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PJ12 Mode Bit (PJ12MD): Selects the function of pin PJ12/TI9C.

Bit 8: PJ12MD	Description
0	General input/output (PJ12) (Initial value)
1	ATU-II event counter input (TI9C)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PJ11 Mode Bit (PJ11MD): Selects the function of pin PJ11/TI9B.

Bit 6: PJ11MD	Description
0	General input/output (PJ11) (Initial value)
1	ATU-II event counter input (TI9B)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PJ10 Mode Bit (PJ10MD): Selects the function of pin PJ10/TI9A.

Bit 4: PJ10MD	Description
0	General input/output (PJ10) (Initial value)
1	ATU-II event counter input (TI9A)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PJ9 Mode Bit (PJ9MD): Selects the function of pin PJ9/TIO5D.

Bit 2: PJ9MD	Description
0	General input/output (PJ9) (Initial value)
1	ATU-II input capture input/output compare output (TIO5D)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 0—PJ8 Mode Bit (PJ8MD): Selects the function of pin PJ8/TIO5C.

Bit 0: PJ8MD	Description
0	General input/output (PJ8) (Initial value)
1	ATU-II input capture input/output compare output (TIO5C)

### Port J Control Register L (PJCRL)

Bit:	15	14	13	12	11	10	9	8
	—	PJ7MD	—	PJ6MD	—	PJ5MD	—	PJ4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PJ3MD	—	PJ2MD	—	PJ1MD	—	PJ0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PJ7 Mode Bit (PJ7MD): Selects the function of pin PJ7/TIO2H.

Bit 14: PJ7MD	Description
0	General input/output (PJ7) (Initial value)
1	ATU-II input capture input/output compare output (TIO2H)

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PJ6 Mode Bit (PJ6MD): Selects the function of pin PJ6/TIO2G.

Bit 12: PJ6MD	Description
0	General input/output (PJ6) (Initial value)
1	ATU-II input capture input/output compare output (TIO2G)

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PJ5 Mode Bit (PJ5MD): Selects the function of pin PJ5/TIO2F.

Bit 10: PJ5MD	Description
0	General input/output (PJ5) (Initial value)
1	ATU-II input capture input/output compare output (TIO2F)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PJ4 Mode Bit (PJ4MD): Selects the function of pin PJ4/TIO2E.

Bit 8: PJ4MD	Description
0	General input/output (PJ4) (Initial value)
1	ATU-II input capture input/output compare output (TIO2E)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PJ3 Mode Bit (PJ3MD): Selects the function of pin PJ3/TIO2D.

Bit 6: PJ3MD	Description
0	General input/output (PJ3) (Initial value)
1	ATU-II input capture input/output compare output (TIO2D)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PJ2 Mode Bit (PJ2MD): Selects the function of pin PJ2/TIO2C.

Bit 4: PJ2MD	Description
0	General input/output (PJ2) (Initial value)
1	ATU-II input capture input/output compare output (TIO2C)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PJ1 Mode Bit (PJ1MD): Selects the function of pin PJ1/TIO2B.

Bit 2: PJ1MD	Description
0	General input/output (PJ1) (Initial value)
1	ATU-II input capture input/output compare output (TIO2B)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PJ0 Mode Bit (PJ0MD): Selects the function of pin PJ0/TIO2A.

Bit 0: PJ0MD	Description
0	General input/output (PJ0) (Initial value)
1	ATU-II input capture input/output compare output (TIO2A)

### 21.3.20 Port K IO Register (PKIOR)

Bit:	15	14	13	12	11	10	9	8
	PK15 IOR	PK14 IOR	PK13 IOR	PK12 IOR	PK11 IOR	PK10 IOR	PK9 IOR	PK8 IOR
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:	7	6	5	4	3	2	1	0
	PK7 IOR	PK6 IOR	PK5 IOR	PK4 IOR	PK3 IOR	PK2 IOR	PK1 IOR	PK0 IOR
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

The port K IO register (PKIOR) is a 16-bit readable/writable register that selects the input/output direction of the 16 pins in port K. Bits PK15IOR to PK0IOR correspond to pins PK15/TO8P to PK0/TO8A. PKIOR is enabled when port K pins function as general input/output pins (PK15 to PK0), and disabled otherwise.

When port K pins function as PK15 to PK0, a pin becomes an output when the corresponding bit in PKIOR is set to 1, and an input when the bit is cleared to 0.

PKIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

### 21.3.21 Port K Control Registers H and L (PKCRH, PKCRL)

Port K control registers H and L (PKCRH, PKCRL) are 16-bit readable/writable registers that select the functions of the 16 multiplex pins in port K. PKCRH selects the functions of the pins for the upper 8 bits of port K, and PKCRL selects the functions of the pins for the lower 8 bits.

PKCRH and PKCRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

## Port K Control Register H (PKCRH)

Bit:	15	14	13	12	11	10	9	8
	—	PK15 MD	—	PK14 MD	—	PK13 MD	—	PK12 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PK11 MD	—	PK10 MD	—	PK9 MD	—	PK8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PK15 Mode Bit (PK15MD): Selects the function of pin PK15/TO8P.

Bit 14: PK15MD	Description
0	General input/output (PK15) (Initial value)
1	ATU-II one-shot pulse output (TO8P)

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PK14 Mode Bit (PK14MD): Selects the function of pin PK14/TO8O.

Bit 12: PK14MD	Description
0	General input/output (PK14) (Initial value)
1	ATU-II one-shot pulse output (TO8O)

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PK13 Mode Bit (PK13MD): Selects the function of pin PK13/TO8N.

Bit 10: PK13MD	Description
0	General input/output (PK13) (Initial value)
1	ATU-II one-shot pulse output (TO8N)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.



- Bit 8—PK12 Mode Bit (PK12MD): Selects the function of pin PK12/TO8M.

Bit 8: PK12MD	Description
0	General input/output (PK12) (Initial value)
1	ATU-II one-shot pulse output (TO8M)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PK11 Mode Bit (PK11MD): Selects the function of pin PK11/TO8L.

Bit 6: PK11MD	Description
0	General input/output (PK11) (Initial value)
1	ATU-II one-shot pulse output (TO8L)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PK10 Mode Bit (PK10MD): Selects the function of pin PK10/TO8K.

Bit 4: PK10MD	Description
0	General input/output (PK10) (Initial value)
1	ATU-II one-shot pulse output (TO8K)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PK9 Mode Bit (PK9MD): Selects the function of pin PK9/TO8J.

Bit 2: PK9MD	Description
0	General input/output (PK9) (Initial value)
1	ATU-II one-shot pulse output (TO8J)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PK8 Mode Bit (PK8MD): Selects the function of pin PK8/TO8I.

Bit 0: PK8MD	Description
0	General input/output (PK8) (Initial value)
1	ATU-II one-shot pulse output (TO8I)

## Port K Control Register L (PKCRL)

Bit:	15	14	13	12	11	10	9	8
	—	PK7MD	—	PK6MD	—	PK5MD	—	PK4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PK3MD	—	PK2MD	—	PK1MD	—	PK0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PK7 Mode Bit (PK7MD): Selects the function of pin PK7/TO8H.

Bit 14: PK7MD	Description
0	General input/output (PK7) (Initial value)
1	ATU-II one-shot pulse output (TO8H)

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PK6 Mode Bit (PK6MD): Selects the function of pin PK6/TO8G.

Bit 12: PK6MD	Description
0	General input/output (PK6) (Initial value)
1	ATU-II one-shot pulse output (TO8G)

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PK5 Mode Bit (PK5MD): Selects the function of pin PK5/TO8F.

Bit 10: PK5MD	Description
0	General input/output (PK5) (Initial value)
1	ATU-II one-shot pulse output (TO8F)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 8—PK4 Mode Bit (PK4MD): Selects the function of pin PK4/TO8E.

Bit 8: PK4MD	Description
0	General input/output (PK4) (Initial value)
1	ATU-II one-shot pulse output (TO8E)

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PK3 Mode Bit (PK3MD): Selects the function of pin PK3/TO8D.

Bit 6: PK3MD	Description
0	General input/output (PK3) (Initial value)
1	ATU-II one-shot pulse output (TO8D)

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 4—PK2 Mode Bit (PK2MD): Selects the function of pin PK2/TO8C.

Bit 4: PK2MD	Description
0	General input/output (PK2) (Initial value)
1	ATU-II one-shot pulse output (TO8C)

- Bit 3—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 2—PK1 Mode Bit (PK1MD): Selects the function of pin PK1/TO8B.

Bit 2: PK1MD	Description
0	General input/output (PK1) (Initial value)
1	ATU-II one-shot pulse output (TO8B)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 0—PK0 Mode Bit (PK0MD): Selects the function of pin PK0/TO8A.

Bit 0: PK0MD	Description
0	General input/output (PK0) (Initial value)
1	ATU-II one-shot pulse output (TO8A)

### 21.3.22 Port K Invert Register (PKIR)

Bit:	15	14	13	12	11	10	9	8
	PK15IR	PK14IR	PK13IR	PK12IR	PK11IR	PK10IR	PK9IR	PK8IR
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:	7	6	5	4	3	2	1	0
	PK7IR	PK6IR	PK5IR	PK4IR	PK3IR	PK2IR	PK1IR	PK0IR
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

The port K invert register (PKIR) is a 16-bit readable/writable register that sets the port K inversion function. Bits PK15IR to PK0IR correspond to pins PK15/TO8P to PK0/TO8A. PKIR is enabled when port K pins function as ATU-II outputs, and disabled otherwise.

When port K pins function as ATU-II outputs, the value of a pin is inverted when the corresponding bit in PKIR is set to 1.

PKIR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PKnIR	Description
0	Value is not inverted (Initial value)
1	Value is inverted

n = 15 to 0

### 21.3.23 Port L IO Register (PLIOR)

Bit:	15	14	13	12	11	10	9	8
	—	—	PL13 IOR	PL12 IOR	PL11 IOR	PL10 IOR	PL9 IOR	PL8 IOR
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:	7	6	5	4	3	2	1	0
	PL7 IOR	PL6 IOR	PL5 IOR	PL4 IOR	PL3 IOR	PL2 IOR	PL1 IOR	PL0 IOR
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

The port L IO register (PLIOR) is a 16-bit readable/writable register that selects the input/output direction of the 14 pins in port L. Bits PL13IOR to PL0IOR correspond to pins PL13/ $\overline{\text{IRQOUT}}$  to PL0/TI10. PLIOR is enabled when port L pins function as general input/output pins (PL13 to PL0), timer input/output pins (TIO11A, TIO11B), or serial clock pins (SCK2, SCK3, SCK4), and disabled otherwise.

When port L pins function as PL13 to PL0, TIO11A and TIO11B, or SCK2, SCK3, and SCK4, a pin becomes an output when the corresponding bit in PLIOR is set to 1, and an input when the bit is cleared to 0.

PLIOR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

### 21.3.24 Port L Control Registers H and L (PLCRH, PLCRL)

Port L control registers H and L (PLCRH, PLCRL) are 16-bit readable/writable registers that select the functions of the 14 multiplex pins in port L. PLCRH selects the functions of the pins for the upper 6 bits of port L, and PLCRL selects the functions of the pins for the lower 8 bits.

PLCRH and PLCRL are initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. They are not initialized in software standby mode or sleep mode.

#### Port L Control Register H (PLCRH)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	PL13 MD1	PL13 MD0	—	PL12 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R	R/W
Bit:	7	6	5	4	3	2	1	0
	PL11 MD1	PL11 MD0	PL10 MD1	PL10 MD0	PL9 MD1	PL9 MD0	—	PL8 MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

- Bits 15 to 12—Reserved: These bits are always read as 0. The write value should always be 0.
- Bits 11 and 10—PL13 Mode Bits 1 and 0 (PL13MD1, PL13MD0): These bits select the function of pin PL13/ $\overline{\text{IRQOUT}}$ .

Bit 11: PL13MD1	Bit 10: PL13MD0	Description
0	0	General input/output (PL13) (Initial value)
	1	$\overline{\text{IRQOUT}}$ is fixed high ( $\overline{\text{IRQOUT}}$ )
1	0	$\overline{\text{IRQOUT}}$ is output by INTC interrupt request ( $\overline{\text{IRQOUT}}$ )
	1	Reserved (Do not set)

- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 8—PL12 Mode Bit (PL12MD): Selects the function of pin PL12/ $\overline{\text{IRQ4}}$ .

Bit 8: PL12MD	Description
0	General input/output (PL12) (Initial value)
1	Interrupt request input ( $\overline{\text{IRQ4}}$ )

- Bits 7 and 6—PL11 Mode Bits 1 and 0 (PL11MD1, PL11MD0): These bits select the function of pin PL11/HRxD0/HRxD1.

Bit 7: PL11MD1	Bit 6: PL11MD0	Description
0	0	General input/output (PL11) (Initial value)
	1	HCAN-II receive data input (HRxD0)
1	0	HCAN-II receive data input (HRxD1)
	1	HCAN-II receive data input (both HRxD0 and HRxD1 input)

- Bits 5 and 4—PL10 Mode Bits 1 and 0 (PL10MD1, PL10MD0): These bits select the function of pin PL10/HTxD0/HTxD1.

Bit 5: PL10MD1	Bit 4: PL10MD0	Description
0	0	General input/output (PL10) (Initial value)
	1	HCAN-II transmit data output (HTxD0)
1	0	HCAN-II transmit data output (HTxD1)
	1	HCAN-II transmit data output (AND of HTxD0 and HTxD1)

- Bits 3 and 2—PL9 Mode Bits 1 and 0 (PL9MD1, PL9MD0): These bits select the function of pin PL9/SCK4/ $\overline{\text{IRQ5}}$ .

Bit 3: PL9MD1	Bit 2: PL9MD0	Description
0	0	General input/output (PL9) (Initial value)
	1	Serial clock input/output (SCK4)
1	0	Interrupt request input ( $\overline{\text{IRQ5}}$ )
	1	Reserved (Do not set)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 0—PL8 Mode Bit (PL8MD): Selects the function of pin PL8/SCK3.

Bit 0: PL8MD	Description
0	General input/output (PL8) (Initial value)
1	Serial clock input/output (SCK3)

### Port L Control Register L (PLCRL)

Bit:	15	14	13	12	11	10	9	8
	—	PL7MD	—	PL6MD	—	PL5MD	—	PL4MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R	R/W	R	R/W	R	R/W

Bit:	7	6	5	4	3	2	1	0
	—	PL3MD	PL2MD1	PL2MD0	PL1MD1	PL1MD0	—	PL0MD
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R	R/W

- Bit 15—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 14—PL7 Mode Bit (PL7MD): Selects the function of pin PL7/SCK2.

Bit 14: PL7MD	Description
0	General input/output (PL7) (Initial value)
1	Serial clock input/output (SCK2)

- Bit 13—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 12—PL6 Mode Bit (PL6MD): Selects the function of pin PL6/ADEND.

Bit 12: PL6MD	Description
0	General input/output (PL6) (Initial value)
1	A/D conversion end output (ADEND)

- Bit 11—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 10—PL5 Mode Bit (PL5MD): Selects the function of pin PL5/ $\overline{\text{ADTRG1}}$ .

Bit 10: PL5MD	Description
0	General input/output (PL5) (Initial value)
1	A/D conversion trigger input ( $\overline{\text{ADTRG1}}$ )



- Bit 9—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 8—PL4 Mode Bit (PL4MD): Selects the function of pin PL4/ $\overline{\text{ADTRG0}}$ .

Bit 8: PL4MD	Description
0	General input/output (PL4) (Initial value)
1	A/D conversion trigger input ( $\overline{\text{ADTRG0}}$ )

- Bit 7—Reserved: This bit is always read as 0. The write value should always be 0.
- Bit 6—PL3 Mode Bit (PL3MD): Selects the function of pin PL3/TCLKB.

Bit 6: PL3MD	Description
0	General input/output (PL3) (Initial value)
1	ATU-II clock input (TCLKB)

- Bits 5 and 4—PL2 Mode Bits 1 and 0 (PL2MD1, PL2MD0): These bits select the function of pin PL2/TIO11B/ $\overline{\text{IRQ7}}$ .

Bit 5: PL2MD1	Bit 4: PL2MD0	Description
0	0	General input/output (PL2) (Initial value)
	1	ATU-II input capture input/output compare output (TIO11B)
1	0	Interrupt request input ( $\overline{\text{IRQ7}}$ )
	1	Reserved (Do not set)

- Bits 3 and 2—PL1 Mode Bits 1 and 0 (PL1MD1, PL1MD0): These bits select the function of pin PL1/TIO11A/ $\overline{\text{IRQ6}}$ .

Bit 3: PL1MD1	Bit 2: PL1MD0	Description
0	0	General input/output (PL1) (Initial value)
	1	ATU-II input capture input/output compare output (TIO11A)
1	0	Interrupt request input ( $\overline{\text{IRQ6}}$ )
	1	Reserved (Do not set)

- Bit 1—Reserved: This bit is always read as 0. The write value should always be 0.

- Bit 0—PL0 Mode Bit (PL0MD): Selects the function of pin PL0/TI10.

Bit 0: PL0MD	Description	
0	General input/output (PL0)	(Initial value)
1	ATU-II edge input (TI10)	

### 21.3.25 Port L Invert Register (PLIR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	PL9IR	PL8IR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	PL7IR	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

The port L invert register (PLIR) is a 16-bit readable/writable register that sets the port L inversion function. Bits PL9IR to PL7IR correspond to pins PL9/SCK4/ $\overline{\text{IRQ5}}$  to PL7/SCK2. PLIR is enabled when port L pins function as serial clock pins, and disabled otherwise.

When port L pins function as serial clock pins, the value of a pin is inverted when the corresponding bit in PLIR is set to 1.

PLIR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

PLnIR	Description	
0	Value is not inverted	(Initial value)
1	Value is inverted	

n = 9 to 7



# Section 22 I/O Ports (I/O)

## 22.1 Overview

The SH7058 has 11 ports: A, B, C, D, E, F, G, H, J, K, and L, all supporting both input and output.

Ports A B, E, F, H, J, and K are 16-bit ports, port C is a 5-bit port, ports D and L are 14-bit ports, and port G is a 4-bit port.

All the port pins are multiplexed as general input/output pins and special function pins. The functions of the multiplex pins are selected by means of the pin function controller (PFC). Each port is provided with a data register for storing the pin data.

Each of the ports A, B, D, J, and L is provided with a port register to read the pin values.

## 22.2 Port A

Port A is an input/output port with the 16 pins shown in figure 22.1.

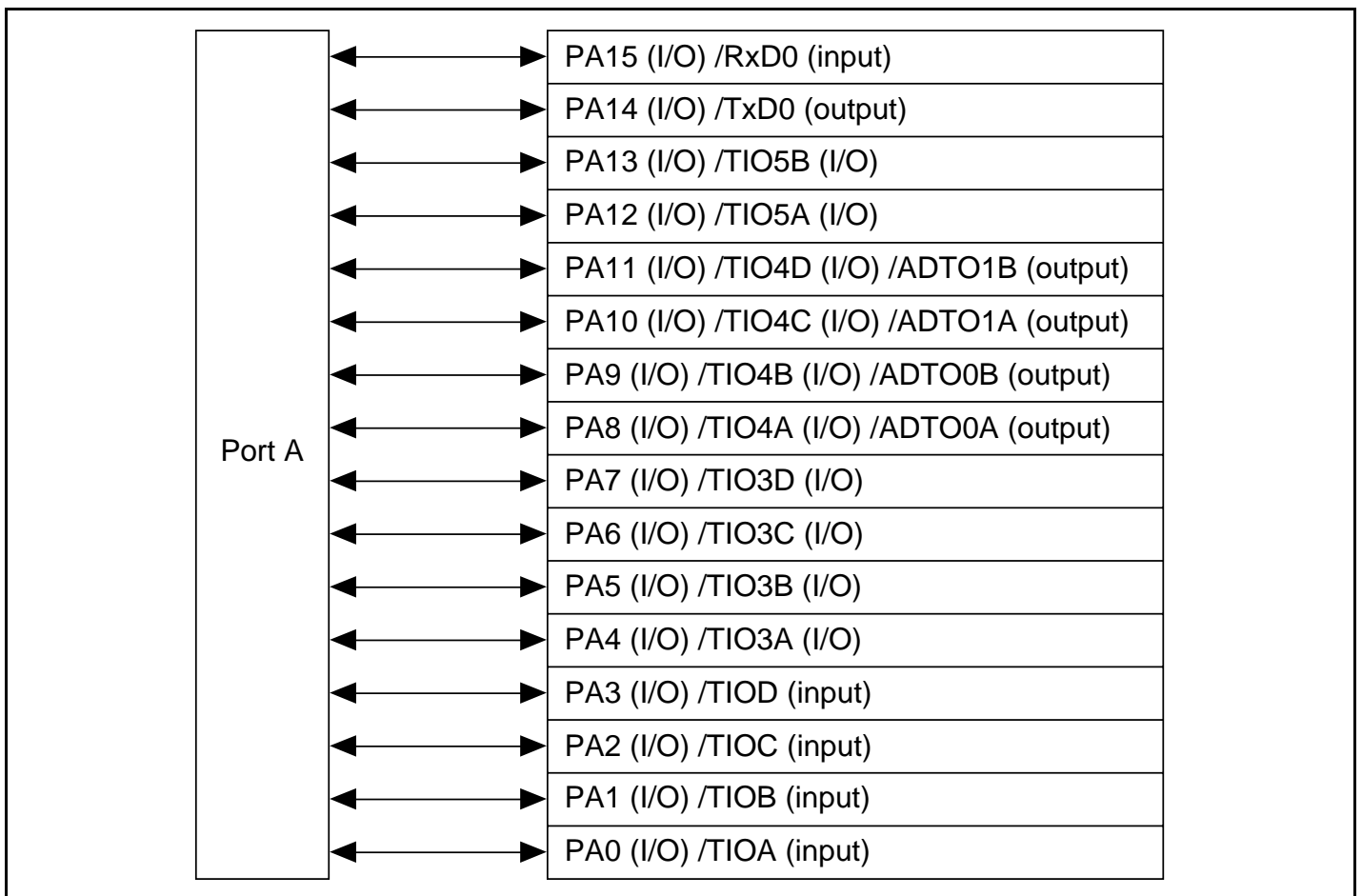


Figure 22.1 Port A

## 22.2.1 Register Configuration

The port A register configuration is shown in table 22.1.

**Table 22.1 Register Configuration**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port A data register	PADR	R/W	H'0000	H'FFFFFF726	8, 16
Port A port register	PAPR	R	Port A pin values	H'FFFFFF780	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 22.2.2 Port A Data Register (PADR)

Bit:	15	14	13	12	11	10	9	8
	PA15 DR	PA14 DR	PA13 DR	PA12 DR	PA11 DR	PA10 DR	PA9 DR	PA8 DR
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:	7	6	5	4	3	2	1	0
	PA7 DR	PA6 DR	PA5 DR	PA4 DR	PA3 DR	PA2 DR	PA1 DR	PA0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port A data register (PADR) is a 16-bit readable/writable register that stores port A data. Bits PA15DR to PA0DR correspond to pins PA15/RxD0 to PA0/TI0A.

When a pin functions as a general output, if a value is written to PADR, that value is output directly from the pin, and if PADR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PADR is read, the pin state, not the register value, is returned directly. If a value is written to PADR, although that value is written into PADR, it does not affect the pin state. Table 22.2 summarizes port A data register read/write operations.

PADR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

**Table 22.2 Port A Data Register (PADR) Read/Write Operations****Bits 15 to 0:**

PAIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PADR, but does not affect pin state
	Other than general input	Pin state	Value is written to PADR, but does not affect pin state
1	General output	PADR value	Write value is output from pin
	Other than general output	PADR value	Value is written to PADR, but does not affect pin state

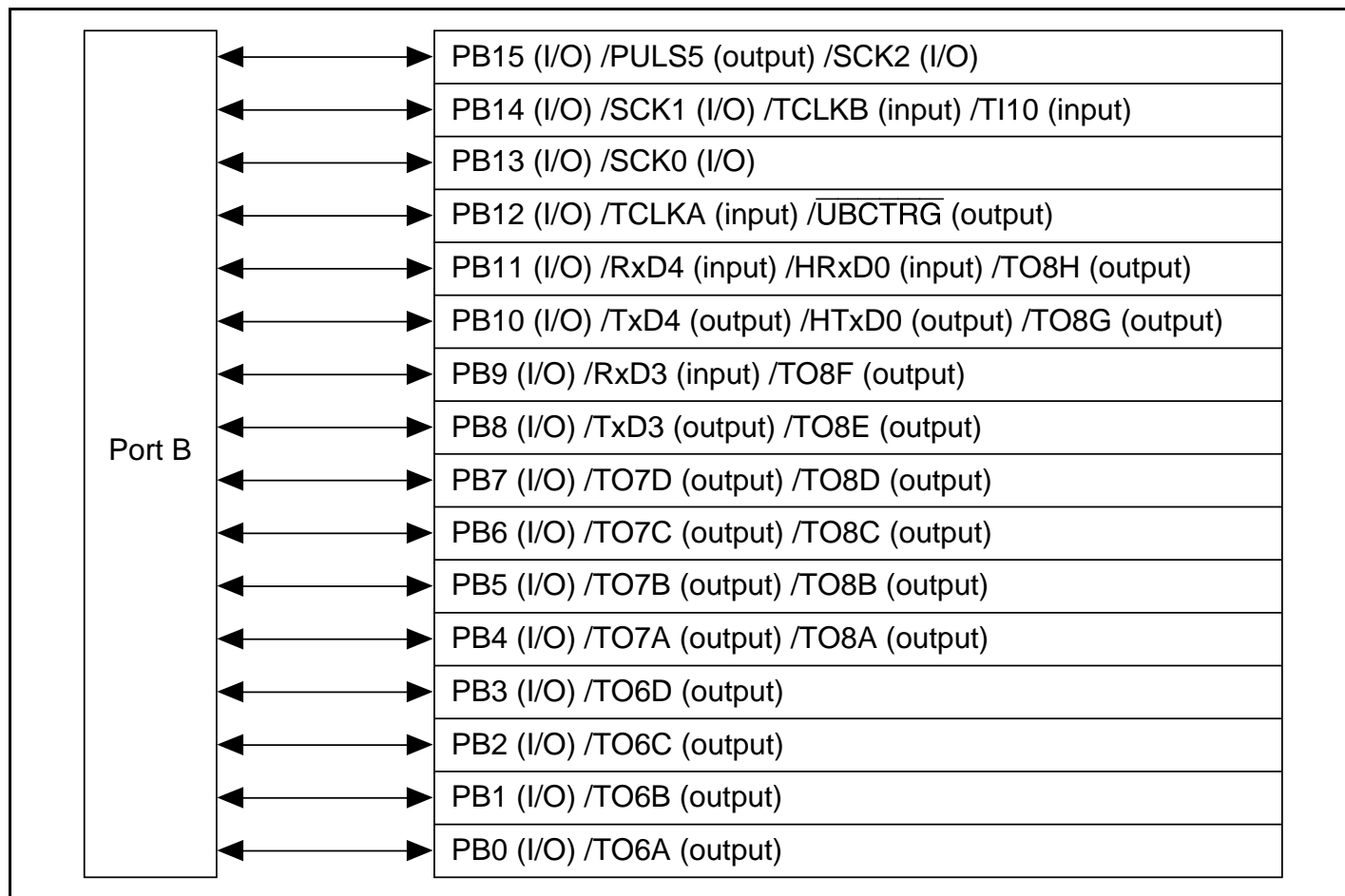
**22.2.3 Port A Port Register (PAPR)**

Bit:	15	14	13	12	11	10	9	8
	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR
Initial value:	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR
Initial value:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
R/W:	R	R	R	R	R	R	R	R

The port A port register (PAPR) is a 16-bit read-only register that always stores the value of the port A pins. The CPU cannot write data to this register. Bits PA15PR to PA0PR correspond to pins PA15/RxD0 to PA0/TIOA. If PAPR is read, the corresponding pin values are returned.

## 22.3 Port B

Port B is an input/output port with the 16 pins shown in figure 21.2.



**Figure 22.2 Port B**

### 22.3.1 Register Configuration

The port B register configuration is shown in table 22.3.

**Table 22.3 Register Configuration**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port B data register	PBDR	R/W	H'0000	H'FFFFFF738	8, 16
Port B port register	PBPR	R	Port B pin values	H'FFFFFF782	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

### 22.3.2 Port B Data Register (PBDR)

Bit:	15	14	13	12	11	10	9	8
	PB15 DR	PB14 DR	PB13 DR	PB12 DR	PB11 DR	PB10 DR	PB9 DR	PB8 DR
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:	7	6	5	4	3	2	1	0
	PB7 DR	PB6 DR	PB5 DR	PB4 DR	PB3 DR	PB2 DR	PB1 DR	PB0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port B data register (PBDR) is a 16-bit readable/writable register that stores port B data. Bits PB15DR to PB0DR correspond to pins PB15/PULS5/SCK2 to PB0/TO6A.

When a pin functions as a general output, if a value is written to PBDR, that value is output directly from the pin, and if PBDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PBDR is read, the pin state, not the register value, is returned directly. If a value is written to PBDR, although that value is written into PBDR, it does not affect the pin state. Table 22.4 summarizes port B data register read/write operations.

PBDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

**Table 22.4 Port B Data Register (PBDR) Read/Write Operations**

**Bits 15 to 0:**

PBIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PBDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PBDR, but does not affect pin state
1	General output	PBDR value	Write value is output from pin
	Other than general output	PBDR value	Value is written to PBDR, but does not affect pin state



### 22.3.3 Port B Port Register (PBPR)

Bit:	15	14	13	12	11	10	9	8
	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR
Initial value:	PB15	PB14	PB13	PB12	PB11	PB10	PB9	PB8
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR
Initial value:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W:	R	R	R	R	R	R	R	R

The port B port register (PBPR) is a 16-bit read-only register that always stores the value of the port B pins. The CPU cannot write data to this register. Bits PB15PR to PB0PR correspond to pins PB15/PULS5/SCK2 to PB0/TO6A. If PBPR is read, the corresponding pin values are returned.

## 22.4 Port C

Port C is an input/output port with the five pins shown in figure 22.3.

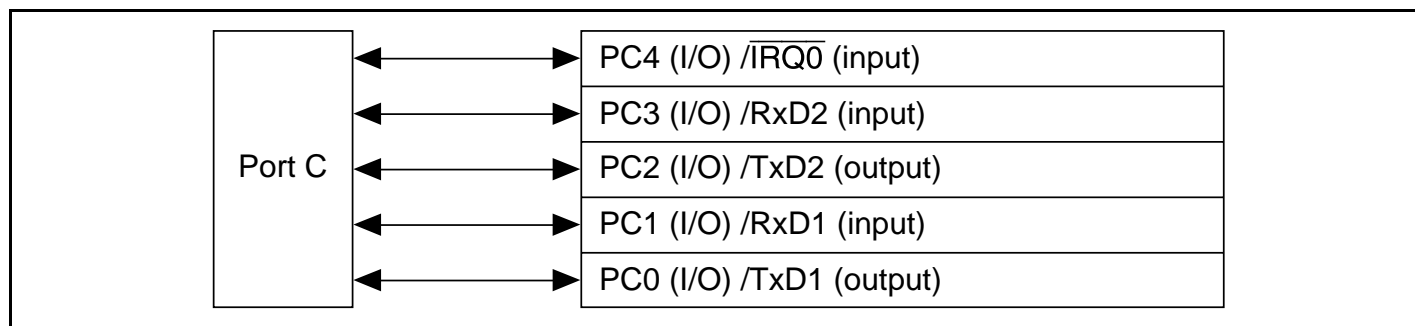


Figure 22.3 Port C

### 22.4.1 Register Configuration

The port C register configuration is shown in table 22.5.

Table 22.5 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port C data register	PCDR	R/W	H'0000	H'FFFFFF73E	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 22.4.2 Port C Data Register (PCDR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	—	—	—	PC4 DR	PC3 DR	PC2 DR	PC1 DR	PC0 DR
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

The port C data register (PCDR) is a 16-bit readable/writable register that stores port C data. Bits PC4DR to PC0DR correspond to pins PC4/ $\overline{\text{IRQ0}}$  to PC0/TxD1.

When a pin functions as a general output, if a value is written to PCDR, that value is output directly from the pin, and if PCDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PCDR is read, the pin state, not the register value, is returned directly. If a value is written to PCDR, although that value is written into PCDR, it does not affect the pin state. Table 22.6 summarizes port C data register read/write operations.

PCDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bits 15 to 5—Reserved: These bits are always read as 0. The write value should always be 0.

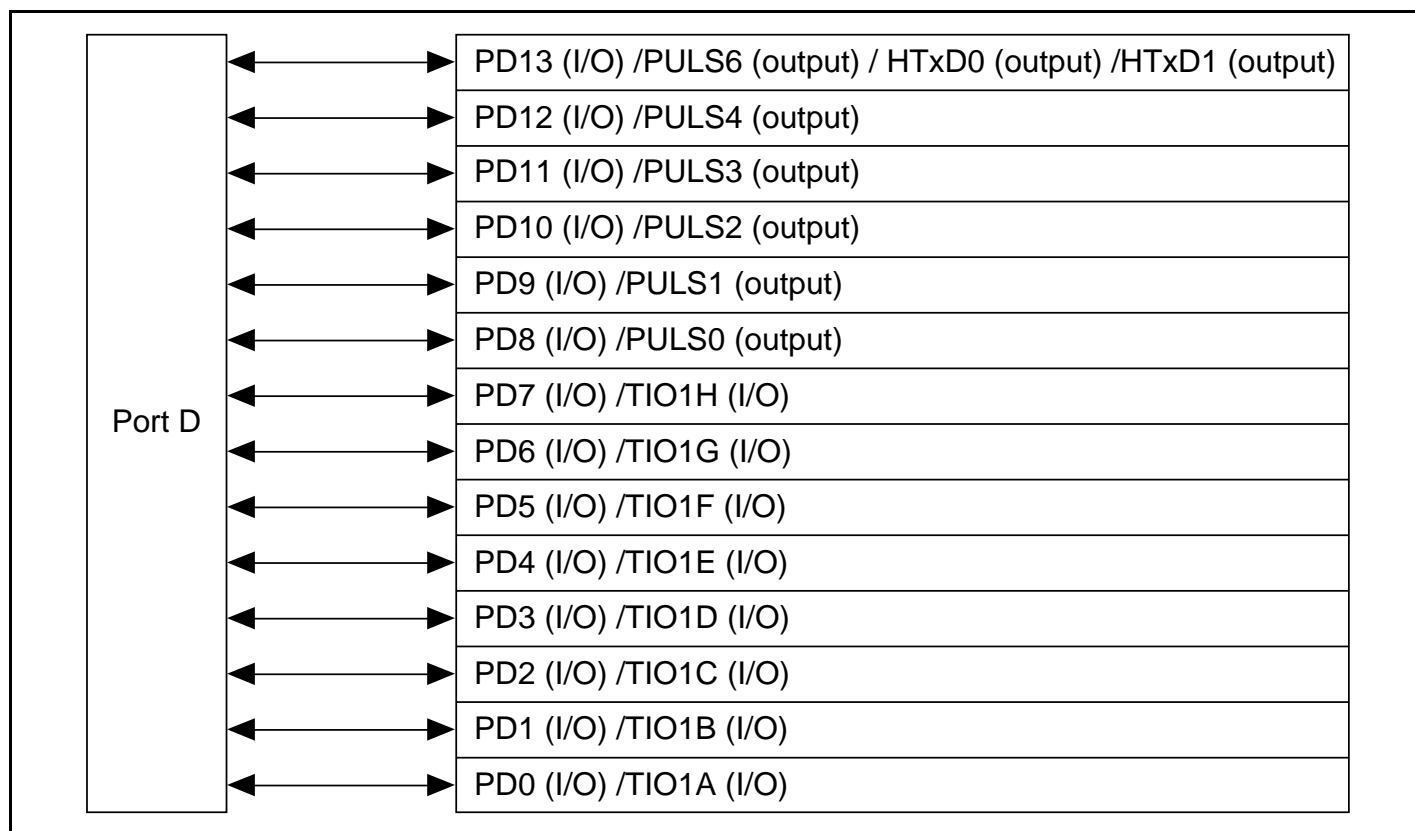
**Table 22.6 Port C Data Register (PCDR) Read/Write Operations**

**Bits 4 to 0:**

PCIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PCDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PCDR, but does not affect pin state
1	General output	PCDR value	Write value is output from pin
	Other than general output	PCDR value	Value is written to PCDR, but does not affect pin state

## 22.5 Port D

Port D is an input/output port with the 14 pins shown in figure 22.4.



**Figure 22.4 Port D**

### 22.5.1 Register Configuration

The port D register configuration is shown in table 22.7.

**Table 22.7 Register Configuration**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port D data register	PDDR	R/W	H'0000	H'FFFFFF746	8, 16
Port D port register	PDPR	R	Port D pin values	H'FFFFFF784	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 22.5.2 Port D Data Register (PDDR)

Bit:	15	14	13	12	11	10	9	8
	—	—	PD13 DR	PD12 DR	PD11 DR	PD10 DR	PD9 DR	PD8 DR
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:	7	6	5	4	3	2	1	0
	PD7 DR	PD6 DR	PD5 DR	PD4 DR	PD3 DR	PD2 DR	PD1 DR	PD0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port D data register (PDDR) is a 16-bit readable/writable register that stores port D data. Bits PD13DR to PD0DR correspond to pins PD13/PULS6/HTxD0/HTxD1 to PD0/TIO1A.

When a pin functions as a general output, if a value is written to PDDR, that value is output directly from the pin, and if PDDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PDDR is read, the pin state, not the register value, is returned directly. If a value is written to PDDR, although that value is written into PDDR, it does not affect the pin state. Table 22.8 summarizes port D data register read/write operations.

PDDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bits 15 and 14— Reserved: These bits are always read as 0. The write value should always be 0.

**Table 22.8 Port D Data Register (PDDR) Read/Write Operations****Bits 13 to 0:**

<b>PDIOR</b>	<b>Pin Function</b>	<b>Read</b>	<b>Write</b>
0	General input	Pin state	Value is written to PDDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PDDR, but does not affect pin state
1	General output	PDDR value	Write value is output from pin
	Other than general output	PDDR value	Value is written to PDDR, but does not affect pin state

**22.5.3 Port D Port Register (PDPR)**

Bit:	15	14	13	12	11	10	9	8
	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR
Initial value:	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR
Initial value:	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
R/W:	R	R	R	R	R	R	R	R

The port D port register (PDPR) is a 16-bit read-only register that always stores the value of the port D pins. The CPU cannot write data to this register. Bits PD13PR to PD0PR correspond to pins PD13/PULS6/HTxD0/HTxD1 to PD0/TIO1A. If PDPR is read, the corresponding pin values are returned.

## 22.6 Port E

Port E is an input/output port with the 16 pins shown in figure 22.5.

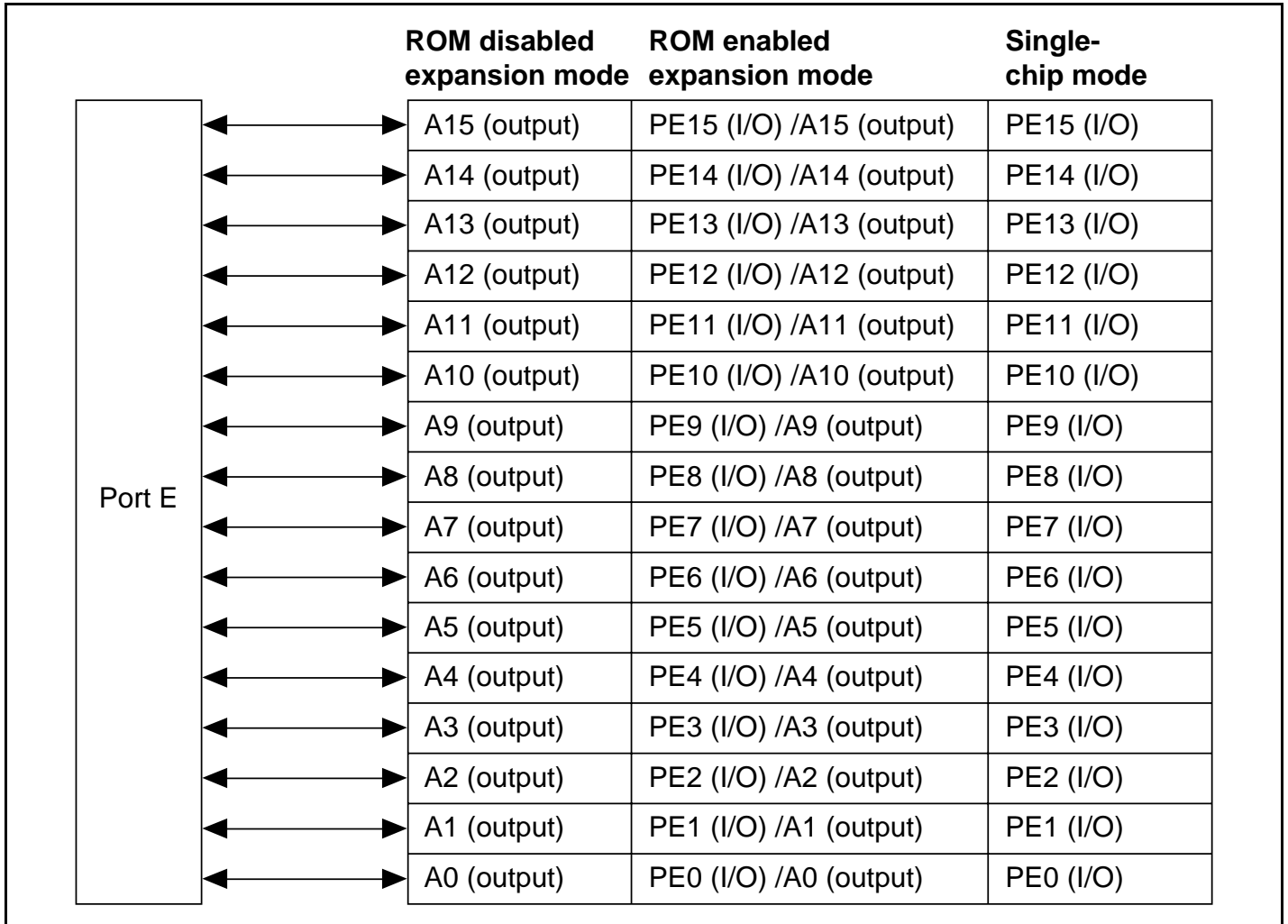


Figure 22.5 Port E

### 22.6.1 Register Configuration

The port E register configuration is shown in table 22.9.

Table 22.9 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port E data register	PEDR	R/W	H'0000	H'FFFFFF754	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 22.6.2 Port E Data Register (PEDR)

Bit:	15	14	13	12	11	10	9	8
	PE15 DR	PE14 DR	PE13 DR	PE12 DR	PE11 DR	PE10 DR	PE9 DR	PE8 DR
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:	7	6	5	4	3	2	1	0
	PE7 DR	PE6 DR	PE5 DR	PE4 DR	PE3 DR	PE2 DR	PE1 DR	PE0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The port E data register (PEDR) is a 16-bit readable/writable register that stores port E data. Bits PE15DR to PE0DR correspond to pins PE15/A15 to PE0/A0.

When a pin functions as a general output, if a value is written to PEDR, that value is output directly from the pin, and if PEDR is read, the register value is returned directly regardless of the pin state. When the  $\overline{\text{POD}}$  pin is driven low, general outputs go to the high-impedance state regardless of the PEDR value. When the  $\overline{\text{POD}}$  pin is driven high, the written value is output from the pin.

When a pin functions as a general input, if PEDR is read, the pin state, not the register value, is returned directly. If a value is written to PEDR, although that value is written into PEDR, it does not affect the pin state. Table 22.10 summarizes port E data register read/write operations.

PEDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

**Table 22.10 Port E Data Register (PEDR) Read/Write Operations****Bits 15 to 0:**

<b>PEIOR</b>	<b>Pin Function</b>	<b>Read</b>	<b>Write</b>
0	General input	Pin state	Value is written to PEDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PEDR, but does not affect pin state
1	General output	PEDR value	Write value is output from pin ( $\overline{\text{POD}}$ pin = high) High impedance regardless of PEDR value ( $\overline{\text{POD}}$ pin = low)
	Other than general output	PEDR value	Value is written to PEDR, but does not affect pin state



## 22.7 Port F

Port F is an input/output port with the 16 pins shown in figure 22.6.

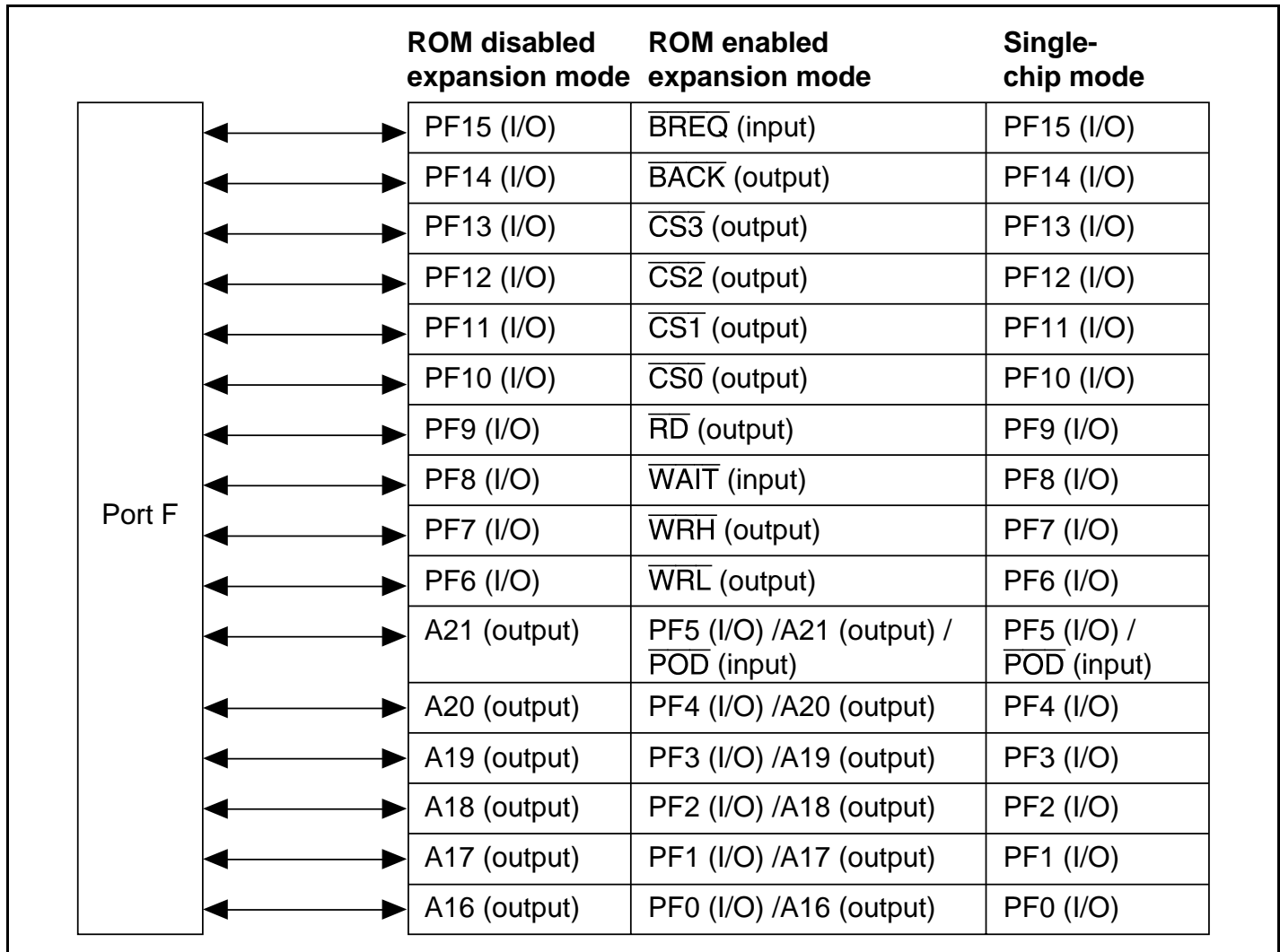


Figure 22.6 Port F

### 22.7.1 Register Configuration

The port F register configuration is shown in table 22.11.

Table 22.11 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port F data register	PFDR	R/W	H'0000	H'FFFFFF74E	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 22.7.2 Port F Data Register (PFDR)

Bit:	15	14	13	12	11	10	9	8
	PF15 DR	PF14 DR	PF13 DR	PF12 DR	PF11 DR	PF10 DR	PF9 DR	PF8 DR
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:	7	6	5	4	3	2	1	0
	PF7 DR	PF6 DR	PF5 DR	PF4 DR	PF3 DR	PF2 DR	PF1 DR	PF0 DR
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

The port F data register (PFDR) is a 16-bit readable/writable register that stores port F data. Bits PF15DR to PF0DR correspond to pins PF15/ $\overline{\text{BREQ}}$  to PF0/A16.

When a pin functions as a general output, if a value is written to PFDR, that value is output directly from the pin, and if PFDR is read, the register value is returned directly regardless of the pin state. For pins PF0 to PF4, when the  $\overline{\text{POD}}$  pin is driven low, general outputs go to the high-impedance state regardless of the PFDR value. When the  $\overline{\text{POD}}$  pin is driven high, the written value is output from the pin.

When a pin functions as a general input, if PFDR is read, the pin state, not the register value, is returned directly. If a value is written to PFDR, although that value is written into PFDR, it does not affect the pin state. Table 22.12 summarizes port F data register read/write operations.

PFDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

**Table 22.12 Port F Data Register (PFDR) Read/Write Operations**

**Bits 15 to 5:**

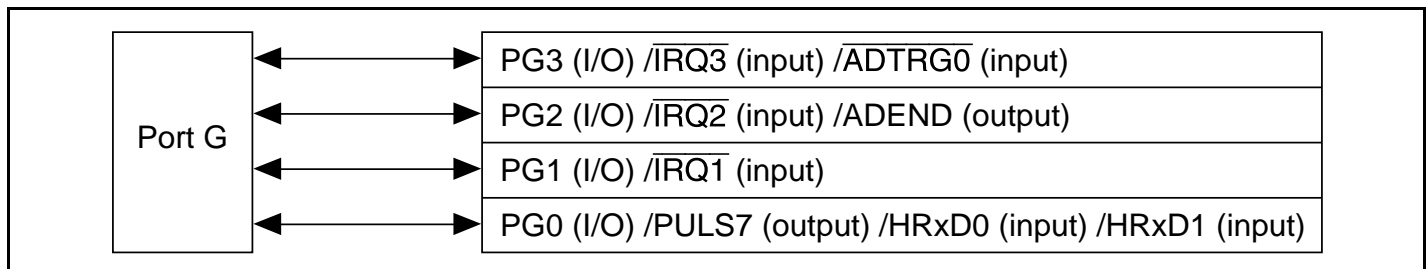
PFIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PFDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PFDR, but does not affect pin state
1	General output	PFDR value	Write value is output from pin
	Other than general output	PFDR value	Value is written to PFDR, but does not affect pin state

**Bits 4 to 0:**

PFIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PFDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PFDR, but does not affect pin state
1	General output	PFDR value	Write value is output from pin ( $\overline{\text{POD}}$ pin = high) High impedance regardless of PFDR value ( $\overline{\text{POD}}$ pin = low)
	Other than general output	PFDR value	Value is written to PFDR, but does not affect pin state

## 22.8 Port G

Port G is an input/output port with the four pins shown in figure 22.7.



**Figure 22.7 Port G**

## 22.8.1 Register Configuration

The port G register configuration is shown in table 22.13.

**Table 22.13 Register Configuration**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port G data register	PGDR	R/W	H'0000	H'FFFFFF764	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 22.8.2 Port G Data Register (PGDR)

Bit:	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	PG3 DR	PG2 DR	PG1 DR	PG0 DR
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W

The port G data register (PGDR) is a 16-bit readable/writable register that stores port G data. Bits PG3DR to PG0DR correspond to pins PG3/ $\overline{\text{IRQ3}}$ / $\overline{\text{ADTRG0}}$  to PG0/PULS7/HRxD0/HRxD1.

When a pin functions as a general output, if a value is written to PGDR, that value is output directly from the pin, and if PGDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PGDR is read, the pin state, not the register value, is returned directly. If a value is written to PGDR, although that value is written into PGDR, it does not affect the pin state. Table 22.14 summarizes port G data register read/write operations.

PGDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bits 15 to 4—Reserved: These bits are always read as 0. The write value should always be 0.

**Table 22.14 Port G Data Register (PGDR) Read/Write Operations****Bits 3 to 0:**

<b>PGIOR</b>	<b>Pin Function</b>	<b>Read</b>	<b>Write</b>
0	General input	Pin state	Value is written to PGDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PGDR, but does not affect pin state
1	General output	PGDR value	Write value is output from pin
	Other than general output	PGDR value	Value is written to PGDR, but does not affect pin state

## 22.9 Port H

Port H is an input/output port with the 16 pins shown in figure 22.8.

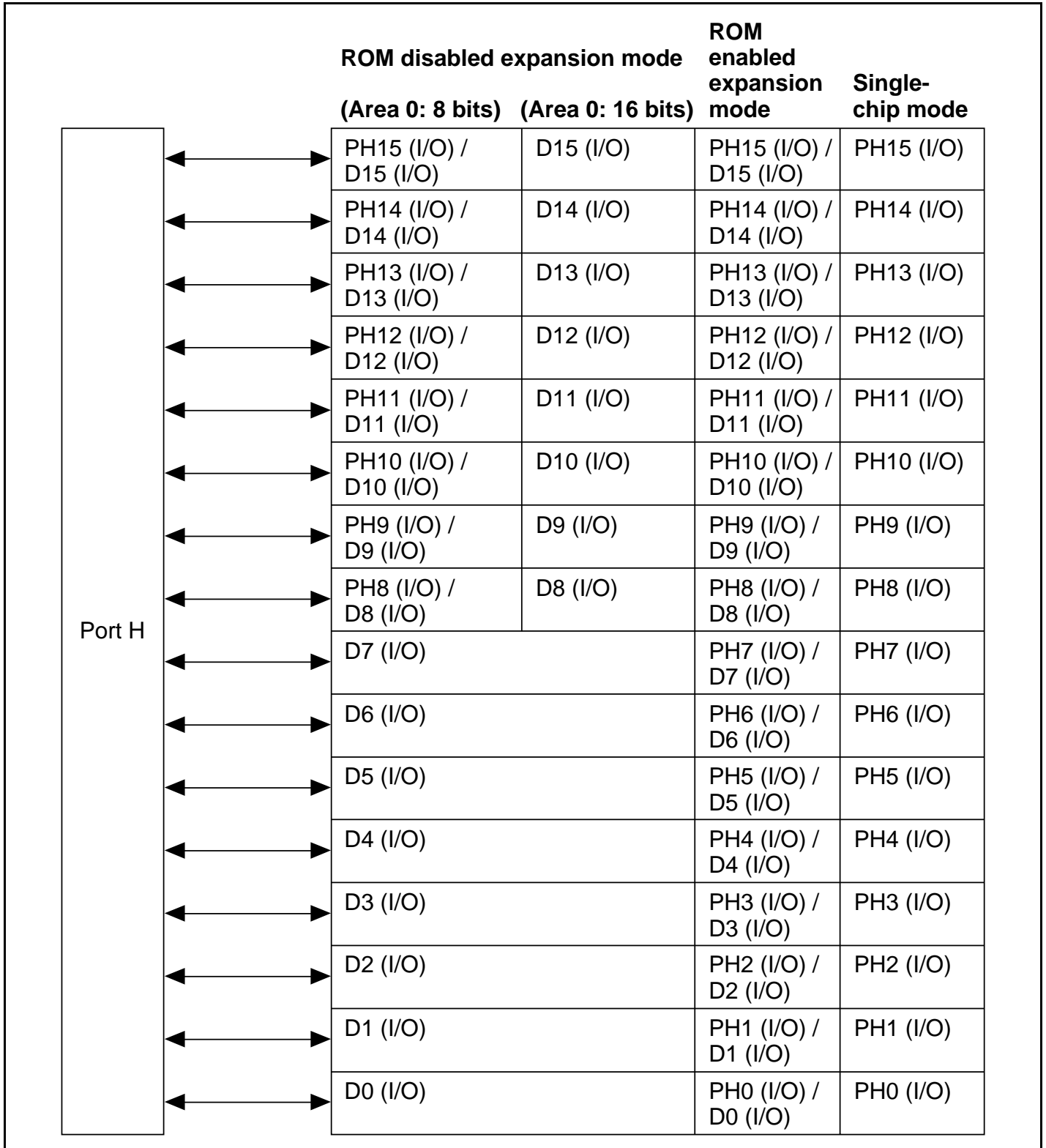


Figure 22.8 Port H

## 22.9.1 Register Configuration

The port H register configuration is shown in table 22.15.

**Table 22.15 Register Configuration**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port H data register	PHDR	R/W	H'0000	H'FFFFFF72C	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 22.9.2 Port H Data Register (PHDR)

Bit:	15	14	13	12	11	10	9	8
	PH15 DR	PH14 DR	PH13 DR	PH12 DR	PH11 DR	PH10 DR	PH9 DR	PH8 DR
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:	7	6	5	4	3	2	1	0
	PH7 DR	PH6 DR	PH5 DR	PH4 DR	PH3 DR	PH2 DR	PH1 DR	PH0 DR
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

The port H data register (PHDR) is a 16-bit readable/writable register that stores port H data. Bits PH15DR to PH0DR correspond to pins PH15/D15 to PH0/D0.

When a pin functions as a general output, if a value is written to PHDR, that value is output directly from the pin, and if PHDR is read, the register value is returned directly regardless of the pin state. When the  $\overline{\text{POD}}$  pin is driven low, general outputs go to the high-impedance state regardless of the PHDR value. When the  $\overline{\text{POD}}$  pin is driven high, the written value is output from the pin.

When a pin functions as a general input, if PHDR is read, the pin state, not the register value, is returned directly. If a value is written to PHDR, although that value is written into PHDR, it does not affect the pin state. Table 22.16 summarizes port H data register read/write operations.

PHDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

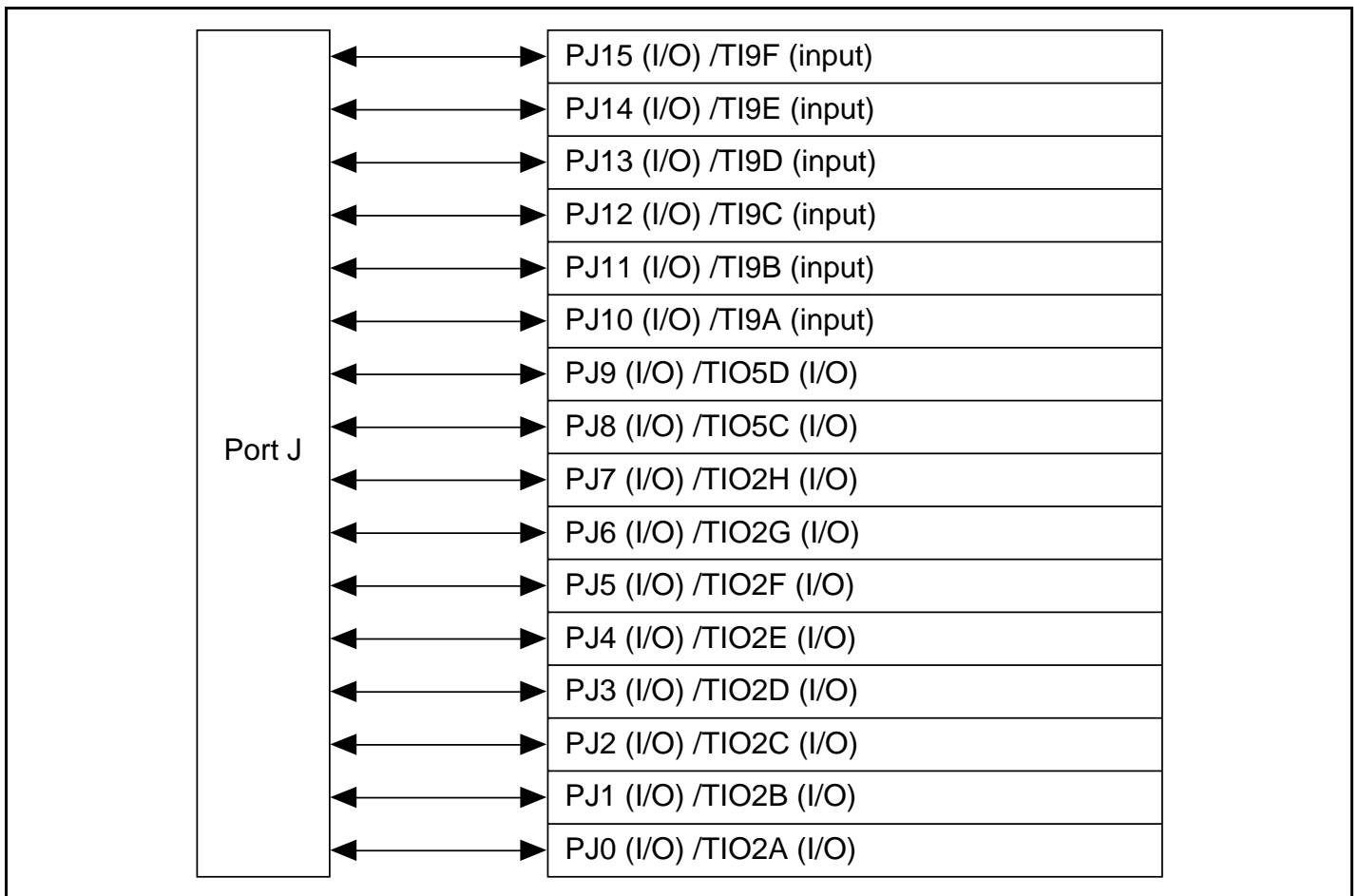
**Table 22.16 Port H Data Register (PHDR) Read/Write Operations**

**Bits 15 to 0:**

PHIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PHDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PHDR, but does not affect pin state
1	General output	PHDR value	Write value is output from pin ( $\overline{\text{POD}}$ pin = high) High impedance regardless of PHDR value ( $\overline{\text{POD}}$ pin = low)
	Other than general output	PHDR value	Value is written to PHDR, but does not affect pin state

## 22.10 Port J

Port J is an input/output port with the 16 pins shown in figure 22.9.



**Figure 22.9 Port J**



## 22.10.1 Register Configuration

The port J register configuration is shown in table 22.17.

**Table 22.17 Register Configuration**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port J data register	PJDR	R/W	H'0000	H'FFFFFF76C	8, 16
Port J port register	PJPR	R	Port J pin values	H'FFFFFF786	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 22.10.2 Port J Data Register (PJDR)

Bit:	15	14	13	12	11	10	9	8
	PJ15 DR	PJ14 DR	PJ13 DR	PJ12 DR	PJ11 DR	PJ10 DR	PJ9 DR	PJ8 DR
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:	7	6	5	4	3	2	1	0
	PJ7 DR	PJ6 DR	PJ5 DR	PJ4 DR	PJ3 DR	PJ2 DR	PJ1 DR	PJ0 DR
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

The port J data register (PJDR) is a 16-bit readable/writable register that stores port J data. Bits PJ15DR to PJ0DR correspond to pins PJ15/TI9F to PJ0/TIO2A.

When a pin functions as a general output, if a value is written to PJDR, that value is output directly from the pin, and if PJDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PJDR is read, the pin state, not the register value, is returned directly. If a value is written to PJDR, although that value is written into PJDR, it does not affect the pin state. Table 22.18 summarizes port J data register read/write operations.

PJDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

**Table 22.18 Port J Data Register (PJDR) Read/Write Operations**

**Bits 15 to 0:**

PJIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PJDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PJDR, but does not affect pin state
1	General output	PJDR value	Write value is output from pin
	Other than general output	PJDR value	Value is written to PJDR, but does not affect pin state

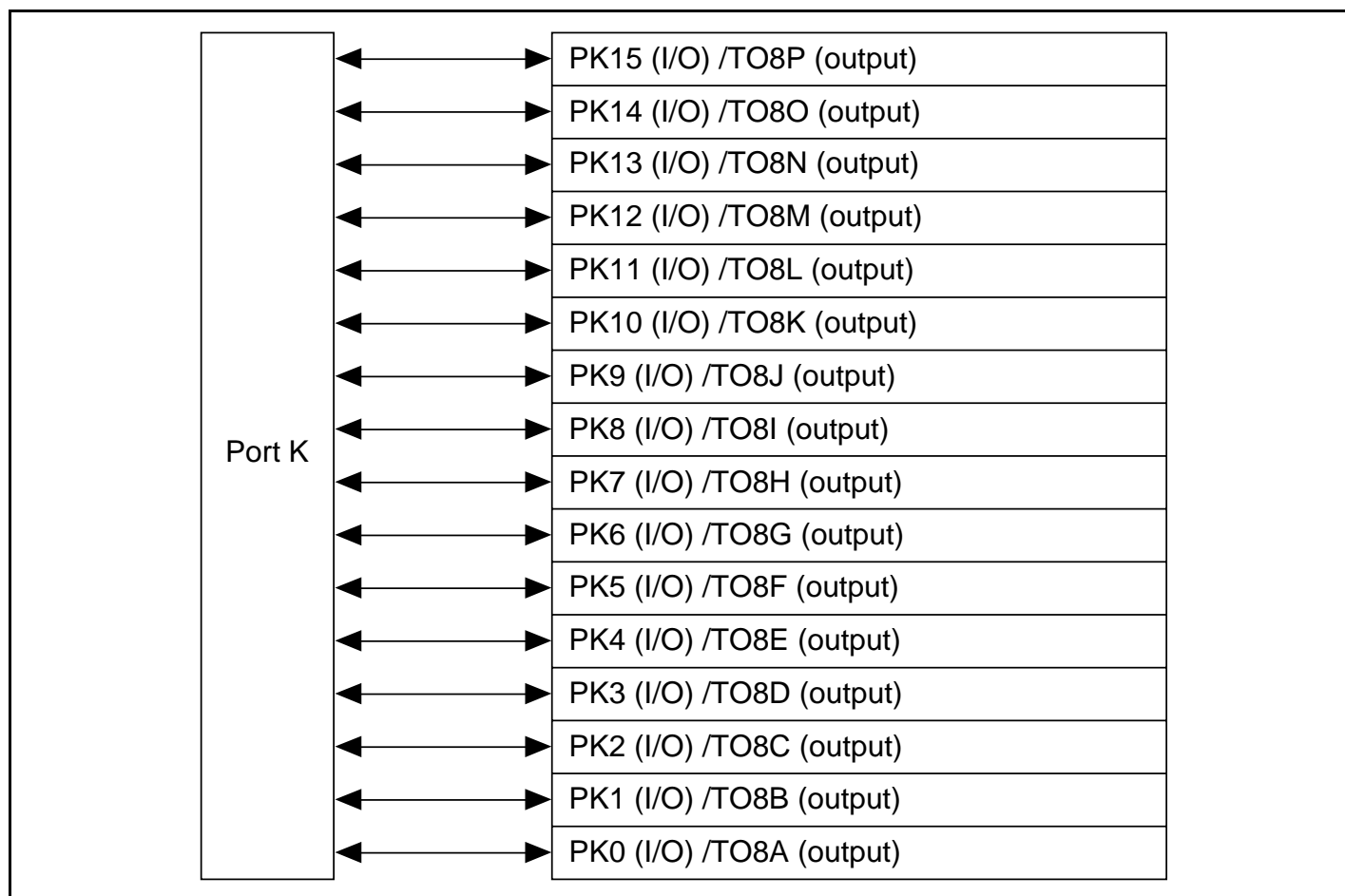
### 22.10.3 Port J Port Register (PJPR)

Bit:	15	14	13	12	11	10	9	8
	PJ15PR	PJ14PR	PJ13PR	PJ12PR	PJ11PR	PJ10PR	PJ9PR	PJ8PR
Initial value:	PJ15	PJ14	PJ13	PJ12	PJ11	PJ10	PJ9	PJ8
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR
Initial value:	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
R/W:	R	R	R	R	R	R	R	R

The port J port register (PJPR) is a 16-bit read-only register that always stores the value of the port J pins. The CPU cannot write data to this register. Bits PJ15PR to PJ0PR correspond to pins PJ15/TI9F to PJ0/TIO2A. If PJPR is read, the corresponding pin values are returned.

## 22.11 Port K

Port K is an input/output port with the 16 pins shown in figure 22.10.



**Figure 21.10 Port K**

### 22.11.1 Register Configuration

The port K register configuration is shown in table 22.19.

**Table 22.19 Register Configuration**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port K data register	PKDR	R/W	H'0000	H'FFFFFF778	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 22.11.2 Port K Data Register (PKDR)

Bit:	15	14	13	12	11	10	9	8
	PK15 DR	PK14 DR	PK13 DR	PK12 DR	PK11 DR	PK10 DR	PK9 DR	PK8 DR
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:	7	6	5	4	3	2	1	0
	PK7 DR	PK6 DR	PK5 DR	PK4 DR	PK3 DR	PK2 DR	PK1 DR	PK0 DR
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

The port K data register (PKDR) is a 16-bit readable/writable register that stores port K data. Bits PK15DR to PK0DR correspond to pins PK15/TO8P to PK0/TO8A.

When a pin functions as a general output, if a value is written to PKDR, that value is output directly from the pin, and if PKDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PKDR is read, the pin state, not the register value, is returned directly. If a value is written to PKDR, although that value is written into PKDR, it does not affect the pin state. Table 22.20 summarizes port K data register read/write operations.

PKDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

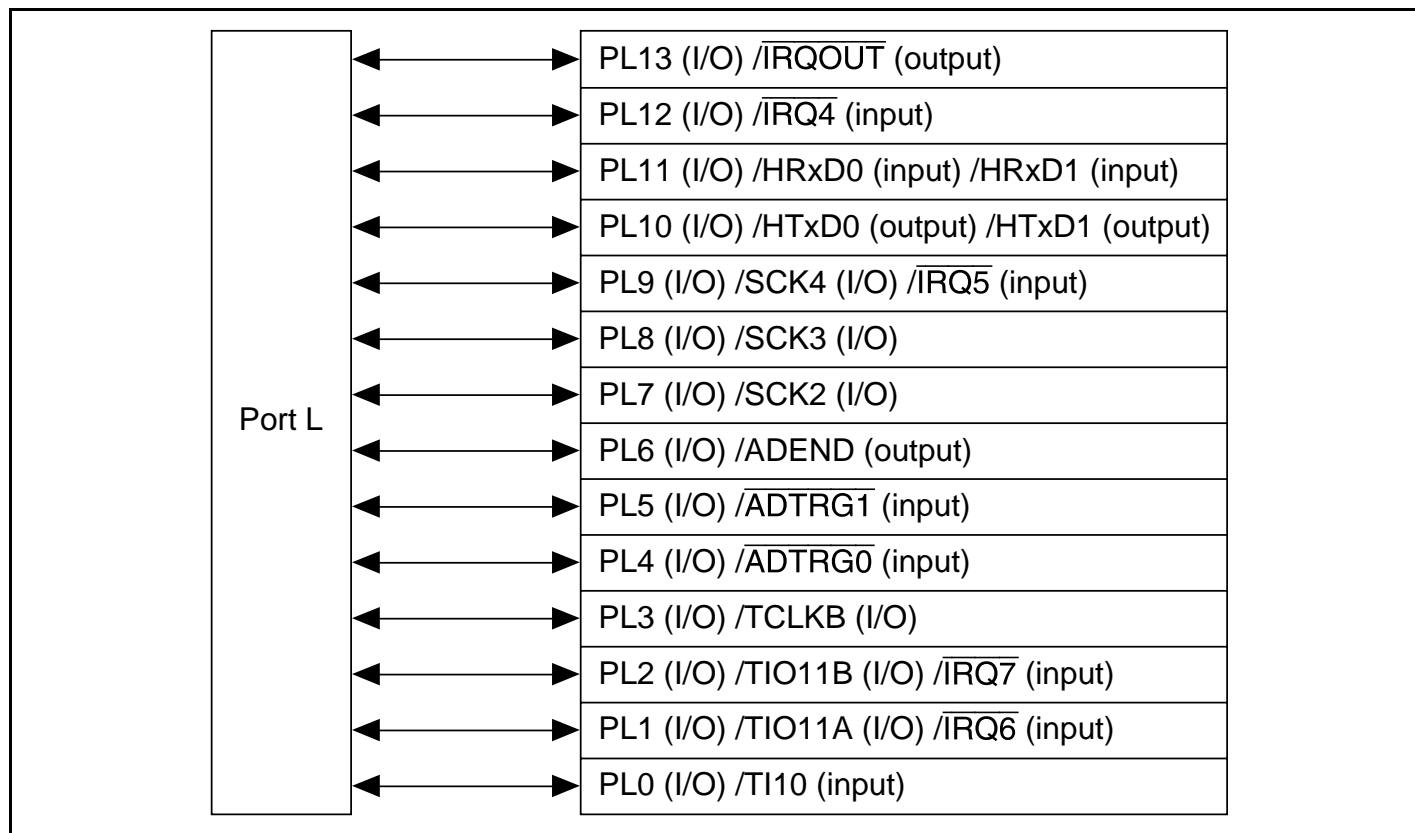
**Table 22.20 Port K Data Register (PKDR) Read/Write Operations**

### Bits 15 to 0:

PKIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PKDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PKDR, but does not affect pin state
1	General output	PKDR value	Write value is output from pin
	Other than general output	PKDR value	Value is written to PKDR, but does not affect pin state

## 22.12 Port L

Port L is an input/output port with the 14 pins shown in figure 22.11.



**Figure 22.11 Port L**

### 22.12.1 Register Configuration

The port L register configuration is shown in table 22.21.

**Table 22.21 Register Configuration**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Port L data register	PLDR	R/W	H'0000	H'FFFFFF75E	8, 16
Port L port register	PLPR	R	Port L pin values	H'FFFFFF788	8, 16

Note: Register access with an internal clock multiplication ratio of 4 requires four or five internal clock ( $\phi$ ) cycles.

## 22.12.2 Port L Data Register (PLDR)

Bit:	15	14	13	12	11	10	9	8
	—	—	PL13 DR	PL12 DR	PL11 DR	PL10 DR	PL9 DR	PL8 DR
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:	7	6	5	4	3	2	1	0
	PL7 DR	PL6 DR	PL5 DR	PL4 DR	PL3 DR	PL2 DR	PL1 DR	PL0 DR
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

The port L data register (PLDR) is a 16-bit readable/writable register that stores port L data. Bits PL13DR to PL0DR correspond to pins PL13/ $\overline{\text{IRQOUT}}$  to PL0/TI10.

When a pin functions as a general output, if a value is written to PLDR, that value is output directly from the pin, and if PLDR is read, the register value is returned directly regardless of the pin state.

When a pin functions as a general input, if PLDR is read, the pin state, not the register value, is returned directly. If a value is written to PLDR, although that value is written into PLDR, it does not affect the pin state. Table 22.22 summarizes port L data register read/write operations.

PLDR is initialized to H'0000 by a power-on reset (excluding a WDT power-on reset), and in hardware standby mode. It is not initialized in software standby mode or sleep mode.

- Bits 15 and 14—Reserved: These bits are always read as 0. The write value should always be 0.

**Table 22.22 Port L Data Register (PLDR) Read/Write Operations**

**Bits 13 to 0:**

PLIOR	Pin Function	Read	Write
0	General input	Pin state	Value is written to PLDR, but does not affect pin state
	Other than general input	Pin state	Value is written to PLDR, but does not affect pin state
1	General output	PLDR value	Write value is output from pin
	Other than general output	PLDR value	Value is written to PLDR, but does not affect pin state

### 22.12.3 Port L Port Register (PLPR)

Bit:	15	14	13	12	11	10	9	8
	PL15PR	PL14PR	PL13PR	PL12PR	PL11PR	PL10PR	PL9PR	PL8PR
Initial value:	PL15	PL14	PL13	PL12	PL11	PL10	PL9	PL8
R/W:	R	R	R	R	R	R	R	R
Bit:	7	6	5	4	3	2	1	0
	PL7PR	PL6PR	PL5PR	PL4PR	PL3PR	PL2PR	PL1PR	PL0PR
Initial value:	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
R/W:	R	R	R	R	R	R	R	R

The port L port register (PLPR) is a 16-bit read-only register that always stores the value of the port L pins. The CPU cannot write data to this register. Bits PL13PR to PL0PR correspond to pins PL13/IRQOUT to PL0/TI10. If PLPR is read, the corresponding pin values are returned.

## 22.13 POD (Port Output Disable) Control

The output port drive buffers for the address bus pins (A20 to A0) and data bus pins (D15 to D0) can be controlled by the  $\overline{\text{POD}}$  (port output disable) pin input level. However, this function is enabled only when the address bus pins (A20 to A0) and data bus pins (D15 to D0) are designated as general output ports.

Output buffer control by means of  $\overline{\text{POD}}$  is performed asynchronously from bus cycles.

$\overline{\text{POD}}$	Address Bus Pins (A20 to A0) and Data Bus Pins (D15 to D0) (when designated as output ports)
0	Enabled (high-impedance)
1	Disabled (general output)





# Section 23 ROM

## 23.1 Features

This LSI has 1-Mbyte on-chip flash memory. The flash memory has the following features.

- Two flash-memory MATs according to LSI initiation mode  
The on-chip flash memory has two memory spaces in the same address space (hereafter referred to as memory MATs). The mode setting in the initiation determines which memory MAT is initiated first. The MAT can be switched by using the bank-switching method after initiation.
  - The user MAT is initiated at a power-on reset in user mode: 1 Mbyte
  - The user boot MAT is initiated at a power-on reset in user boot mode: 8 kbytes
- Three on-board programming modes and one off-board programming mode
  - On-board programming modes  
**Boot Mode:** This mode is a program mode that uses an on-chip SCI interface. The user MAT and user boot MAT can be programmed. This mode can automatically adjust the bit rate between the host and this LSI.  
**User Program Mode:** The user MAT can be programmed by using the optional interface.  
**User Boot Mode:** The user boot program of the optional interface can be made and the user MAT can be programmed.
    - Off-board programming mode  
**Programmer Mode:** This mode uses the PROM programmer. The user MAT and user boot MAT can be programmed.
  - Programming/erasing interface by the download of on-chip program  
This LSI has a dedicated programming/erasing program. After downloading this program to the on-chip RAM, programming/erasing can be performed by setting the argument parameter. The user branch is also supported.
    - User branch  
The program processing is performed in 128-byte units. It consists the program pulse application, verify read, and several other steps. Erasing is performed in one divided-block units and consists of several steps. The user processing routine can be executed between the steps, this setting for which is called the user branch addition.
- Emulation function of flash memory by using the on-chip RAM  
As flash memory is overlapped with part of the on-chip RAM, the flash memory programming can be emulated in real time.
- Protection modes  
There are two protection modes. Software protection by the register setting and hardware protection by the FWE pin. The protection state for flash memory programming/erasing can be set.

When abnormalities, such as runaway of programming/erasing are detected, these modes enter the error protection state and the programming/erasing processing is suspended.

- Programming/erasing time

The flash memory programming time is  $t_p$  ms (typ) in 128-byte simultaneous programming and  $t_p/128$  ms per byte. The erasing time is  $t_e$  s (typ) per block.

- Number of programming

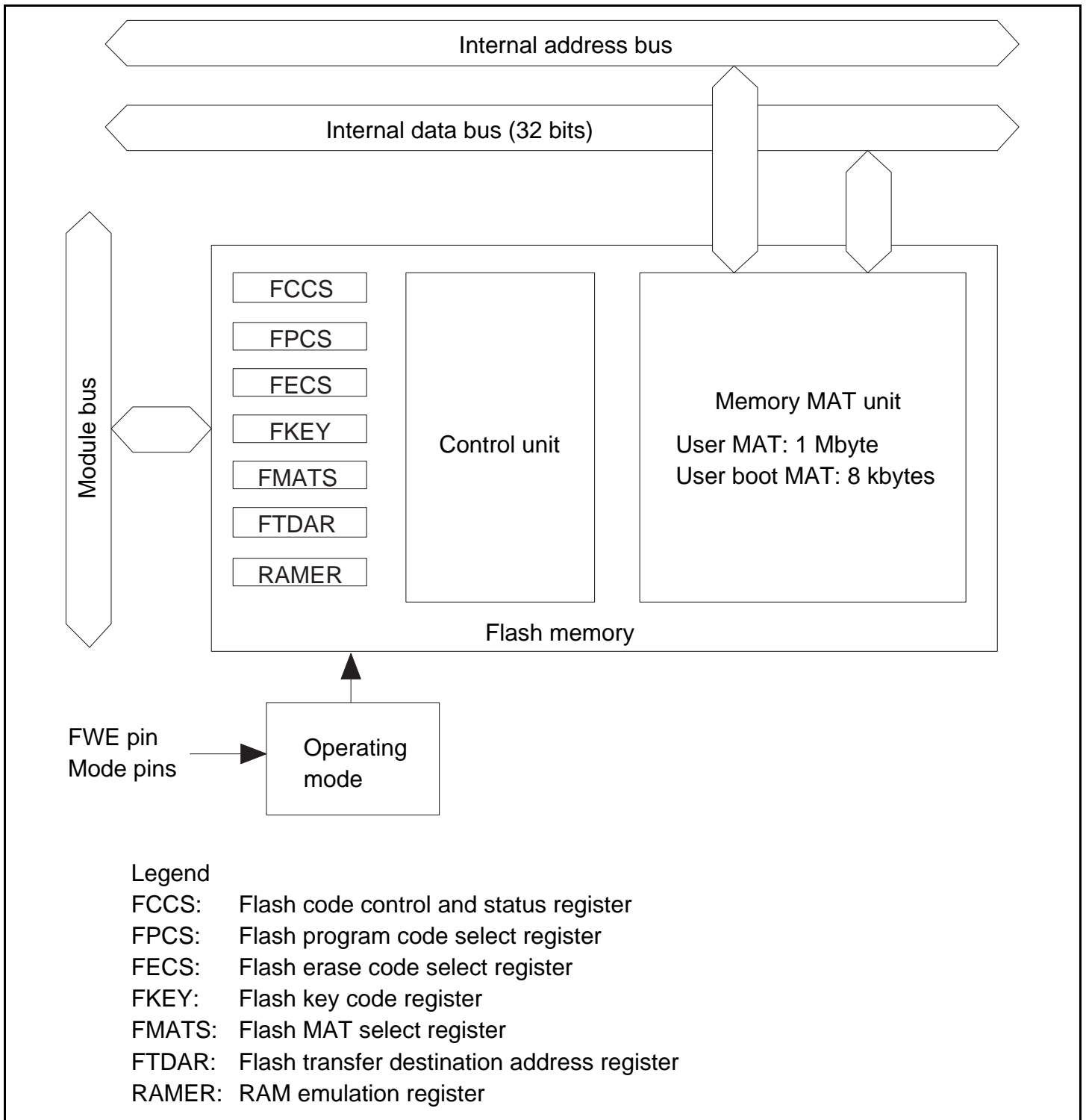
The number of flash memory programming can be up to  $N_{WEC}$  times.

- Operating frequency at programming/erasing

The operating frequency at programming/erasing is a maximum of 40 MHz.

## 23.2 Overview

### 23.2.1 Block Diagram



**Figure 23.1 Block Diagram of Flash Memory**

### 23.2.2 Operating Mode

When each mode pin and the FWE pin are set in the reset state and the reset signal is released, the microcomputer enters each operating mode as shown in figure 23.2. For the setting of each mode pin and the FWE pin, see table 23.1.

- Flash memory cannot be read, programmed, or erased in ROM invalid mode. The programming/erasing interface registers cannot be written to. When these registers are read, H'00 is always read.
- Flash memory can be read in user mode, but cannot be programmed or erased.
- Flash memory can be read, programmed, or erased on the board only in user program mode, user boot mode, and boot mode.
- Flash memory can be read, programmed, or erased by means of the PROM programmer in programmer mode.

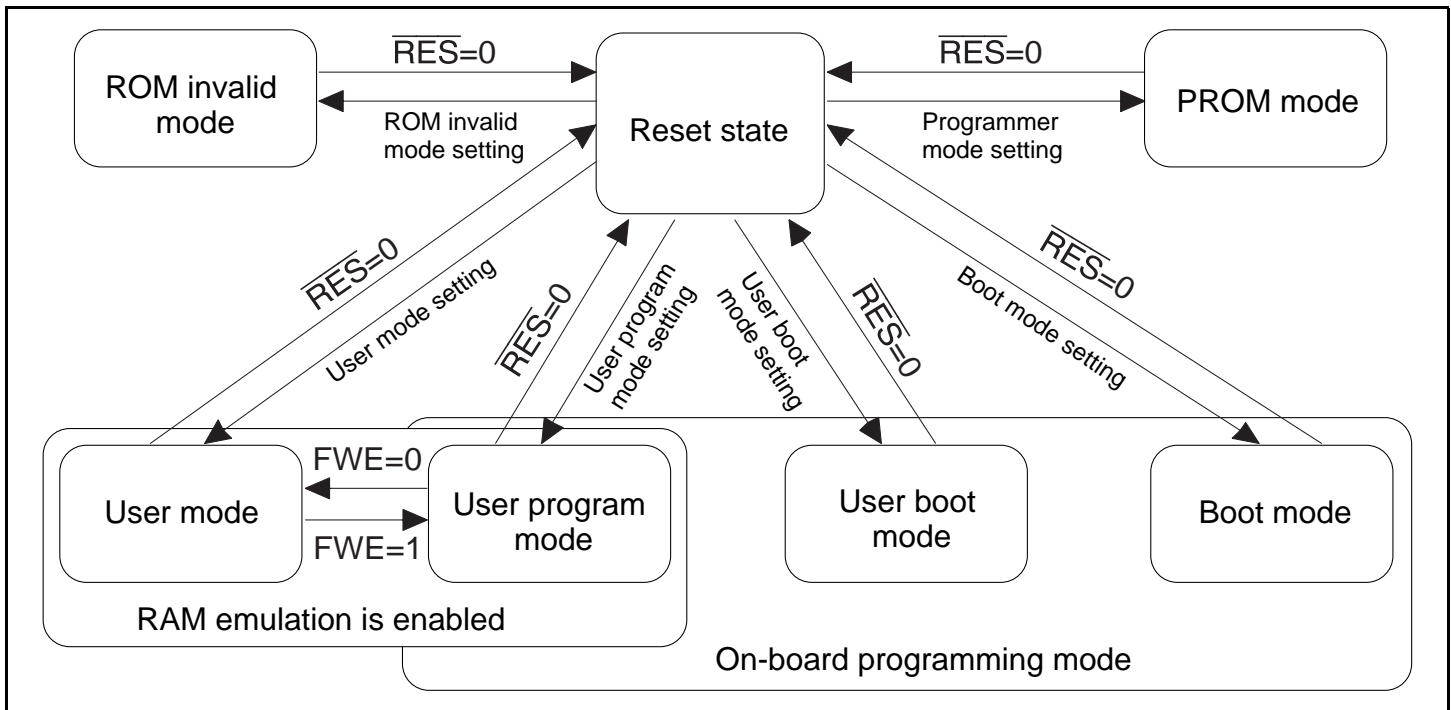


Figure 23.2 Mode Transition of Flash Memory

**Table 23.1 Relationship between FWE and MD Pins and Operating Modes**

Pin	Mode						
	Reset State	ROM Invalid Mode	ROM Valid Mode	User Program Mode	User Boot Mode	Boot Mode	Programmer Mode
$\overline{\text{RES}}$	0	1	1	1	1	1	1
FWE	0/1	0	0	1	1	1	0/1
MD0	0/1	0/1* <sup>1</sup>	0/1* <sup>2</sup>	0/1* <sup>2</sup>	0/1* <sup>2</sup>	0/1* <sup>2</sup>	1
MD1	0/1	0	1	1	0	0	1
MD2	0/1	1	1	1	0	1	0

Notes: 1. MD0 = 0: 8-bit external bus, MD0 = 1: 16-bit external bus

2. MD0 = 0: External bus can be used, MD0 = 1: Single-chip mode (external bus cannot be used)

### 23.2.3 Mode Comparison

The comparison table of programming and erasing related items about boot mode, user program mode, user boot mode, and programmer mode is shown in table 23.2.

**Table 23.2 Comparison of Programming Modes**

	<b>Boot Mode</b>	<b>User Program Mode</b>	<b>User Boot Mode</b>	<b>Programmer Mode</b>
Programming/erasing environment	On-board programming	On-board programming	On-board programming	Off-board programming
Programming/erasing enable MAT	User MAT User boot MAT	User MAT	User MAT	User MAT User boot MAT
Programming/erasing control	Command method	Programming/erasing interface	Programming/erasing interface	Command method
All erasure	O (Automatic)	O	O	O (Automatic)
Block division erasure	O* <sup>1</sup>	O	O	X
Program data transfer	From host via SCI	From optional device via RAM	From optional device via RAM	Via programmer
User branch function	X	O	O	X
RAM emulation	X	O	X	X
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT* <sup>2</sup>	Embedded program storage MAT
Transition to user mode	Mode setting change and reset	FWE setting change	Mode setting change and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.

2. Initiation starts from the embedded program storage MAT. After checking the flash-memory related registers, initiation starts from the reset vector of the user MAT.

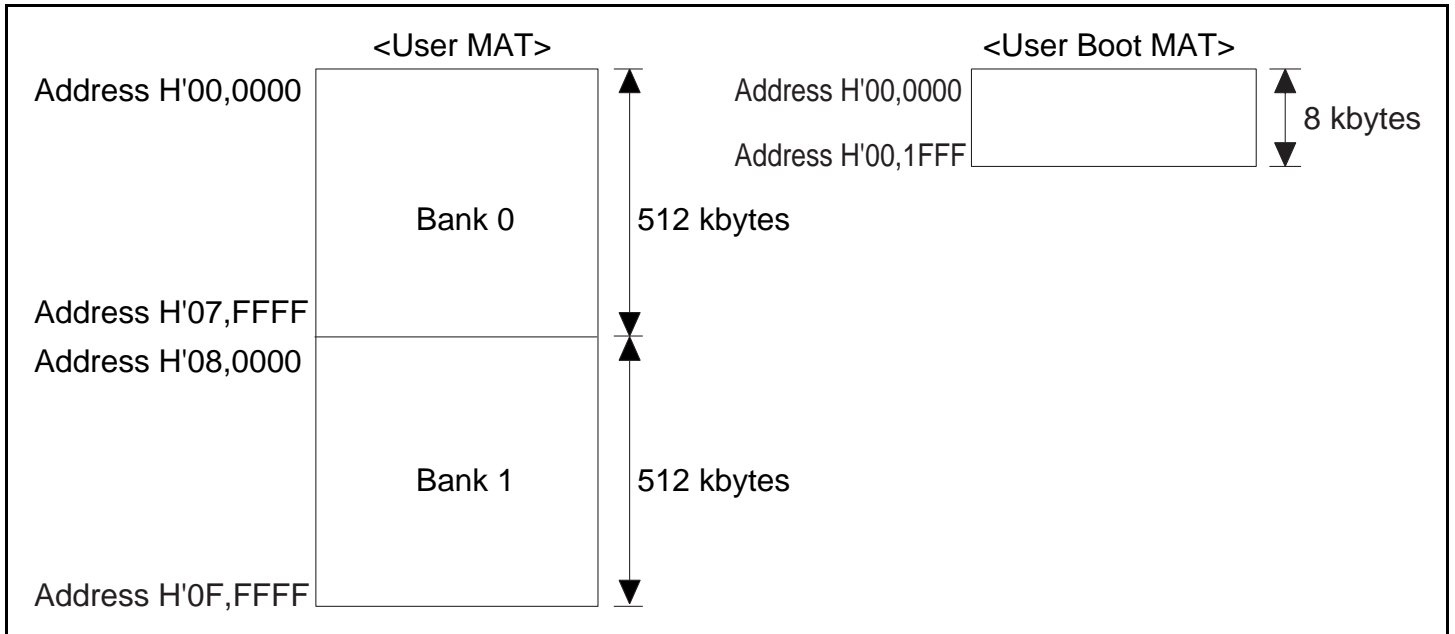
- The user boot MAT can be programmed or erased only in boot mode and programmer mode.
- The user MAT and user boot MAT are all erased in boot mode. Then, the user MAT and user boot MAT can be programmed by means of the command method. However, the contents of the MAT cannot be read until this state.  
Only user boot MAT is programmed and the user MAT is programmed in user boot mode or only user MAT is programmed because user boot mode is not used.
- In user boot mode, the boot operation of the optional interface can be performed by a mode pin setting different from user program mode.

### 23.2.4 Flash Memory Configuration

This LSI's flash memory is configured by the 1-Mbyte user MAT and 8-kbyte user boot MAT.

The start address is allocated to the same address in the user MAT and user boot MAT. Therefore, when the program execution or data access is performed between the two MATs, the MAT must be switched by using FMATS. The user MAT is divided into two 512-kbyte banks (bank 0 and bank 1).

The user MAT or user boot MAT can be read in all modes if it is in ROM valid mode. However, the user boot MAT can be programmed only in boot mode and programmer mode.



**Figure 23.3 Flash Memory Configuration**

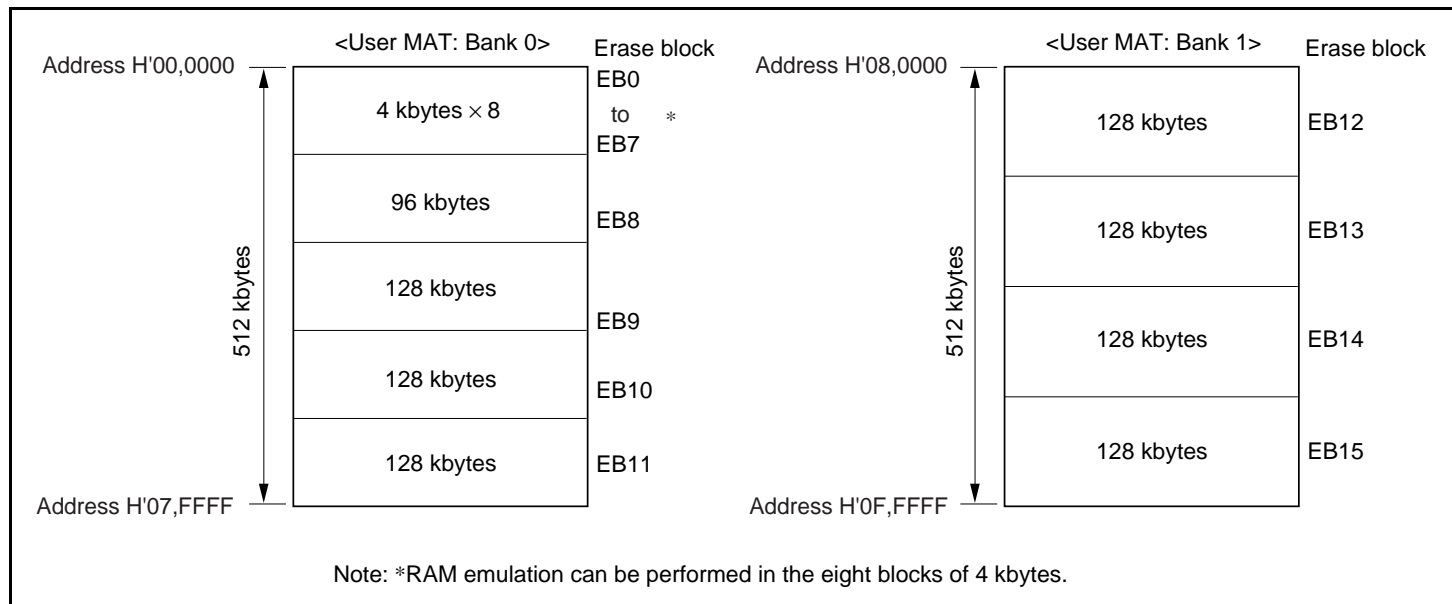
The user MAT and user boot MAT have different memory sizes. Do not access a user boot MAT that is 8 kbytes or more. When a user boot MAT exceeding 8 kbytes is read from, an undefined value is read.



## 23.2.5 Block Division

The user MAT is divided into 128 kbytes (seven blocks), 96 kbytes (one block), and 4 kbytes (eight blocks) as shown in figure 23.4. The user MAT can be erased in this divided-block units and the erase-block number of EB0 to EB15 is specified when erasing.

The RAM emulation can be performed in the eight blocks of 4 kbytes.

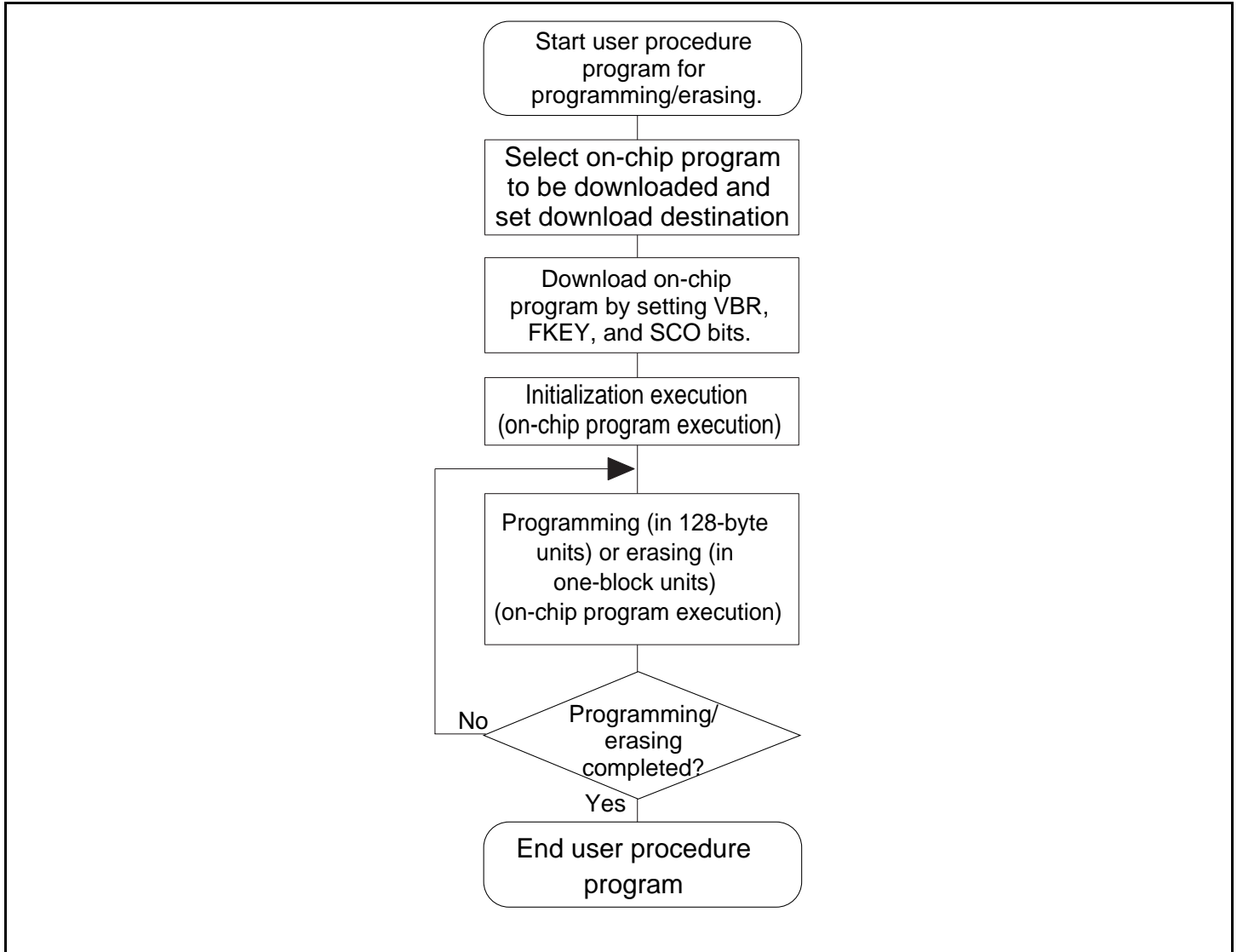


**Figure 23.4 Block Division of User MAT**

## 23.2.6 Programming/Erasing Interface

Programming/erasing is executed by downloading the on-chip program to the on-chip RAM and specifying the program address/data and erase block by using the interface registers/parameters.

The procedure program is made by the user in user program mode and user boot mode. The overview of the procedure is as follows. For details, see section 23.5.2, User Program Mode.



**Figure 23.5 Overview of User Procedure Program**

### (1) Selection of On-Chip Program to be Downloaded and Setting of Download Destination

This LSI has programming/erasing programs and they can be downloaded to the on-chip RAM. The on-chip program to be downloaded is selected by setting the corresponding bits in the programming/erasing interface registers. The download destination can be specified by FTDAR.

## (2) Download of On-Chip Program

The on-chip program is automatically downloaded by clearing VBR of the CPU to H'00000000 and then setting the SCO bit in the flash key code register (FKEY) and the flash code control and status register (FCCS), which are programming/erasing interface registers.

The user MAT is replaced to the embedded program storage area when downloading. Since the flash memory cannot be read when programming/erasing, the procedure program, which is working from download to completion of programming/erasing, must be executed in a space other than the flash memory to be programmed/erased (for example, on-chip RAM).

Since the result of download is returned to the programming/erasing interface parameters, whether the normal download is executed or not can be confirmed.

Note that VBR can be changed after download is completed.

## (3) Initialization of Programming/Erasing

The operating frequency and user branch are set before execution of programming/erasing.

The user branch destination must be in an area other than the user MAT area which is in the middle of programming and the area where the on-chip program is downloaded. These settings are performed by using the programming/erasing interface parameters.

## (4) Programming/Erasing Execution

To program or erase, the FWE pin must be brought high and user program mode must be entered.

The program data/programming destination address is specified in 128-byte units when programming.

The block to be erased is specified in erase-block units when erasing.

These specifications are set by using the programming/erasing interface parameters and the on-chip program is initiated. The on-chip program is executed by using the JSR or BSR instruction to perform the subroutine call of the specified address in the on-chip RAM. The execution result is returned to the programming/erasing interface parameters.

The area to be programmed must be erased in advance when programming flash memory.

There are limitations and notes on the interrupt processing during programming/erasing. For details, see section 23.8.2, Interrupts during Programming/Erasing.

## (5) When Programming/Erasing is Executed Consecutively

When the processing is not ended by the 128-byte programming or one-block erasure, the program address/data and erase-block number must be updated and consecutive programming/erasing is required.

Since the downloaded on-chip program is left in the on-chip RAM after the processing, download and initialization are not required when the same processing is executed consecutively.

## 23.3 Pin Configuration

Flash memory is controlled by the pins as shown in table 23.3.

**Table 23.3 Pin Configuration**

Pin Name	Abbreviation	Input/Output	Function
Power-on reset	$\overline{\text{RES}}$	Input	Reset
Flash programming enable	FWE	Input	Hardware protection when programming flash memory
Mode 2	MD2	Input	Sets operating mode of this LSI
Mode 1	MD1	Input	Sets operating mode of this LSI
Mode 0	MD0	Input	Sets operating mode of this LSI
Transmit data	TxD1	Output	Serial transmit data output (used in boot mode)
Receive data	RxD1	Input	Serial receive data input (used in boot mode)

Note: For the pin configuration in PROM mode, see section 23.9, Programmer Mode.

## 23.4 Register Configuration

### 23.4.1 Registers

The registers/parameters which control flash memory when the on-chip flash memory is valid are shown in table 23.4.

There are several operating modes for accessing flash memory, for example, read mode/program mode.

There are two memory MATs: user MAT and user boot MAT. The dedicated registers/parameters are allocated for each operating mode and MAT selection. The correspondence of operating modes and registers/parameters for use is shown in table 23.5.

**Table 23.4 (1) Register Configuration**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Flash code control status register	FCCS	R, W* <sup>1</sup>	H'00* <sup>2</sup> H'80* <sup>2</sup>	H'FFFFFFE800	8
Flash program code select register	FPCS	R/W	H'00	H'FFFFFFE801	8
Flash erase code select register	FECS	R/W	H'00	H'FFFFFFE802	8
Flash key code register	FKEY	R/W	H'00	H'FFFFFFE804	8
Flash MAT select register	FMATS	R/W	H'00* <sup>3</sup> H'AA* <sup>3</sup>	H'FFFFFFE805	8
Flash transfer destination address register	FTDAR	R/W	H'00	H'FFFFFFE806	8
RAM emulation register	RAMER	R/W	H'0000	H'FFFFFFEC26	8, 16, 32

Notes: All registers except for RAMER can be accessed only in bytes, and the access requires three cycles.

RAMER can be accessed in bytes or words, and the access requires three cycles.

1. The bits except the SCO bit are read-only bits. The SCO bit is a programming-only bit. (The value which can be read is always 0.)
2. The initial value is H'00 when the FWE pin goes low.  
The initial value is H'80 when the FWE pin goes high.
3. The initial value at initiation in user mode or user program mode is H'00.  
The initial value at initiation in user boot mode is H'AA.
4. The registers except RAMER can be accessed only in bytes, and the access requires four cycles. Since RAMER is in the BSC, when it is accessed in bytes or words, the access requires four cycles, and when it is accessed in longwords, the access requires eight cycles.

**Table 23.4 (2) Parameter Configuration**

Name	Abbreviation	R/W	Initial Value	Address	Access Size
Download pass/fail result	DPFR	R/W	Undefined	On-chip RAM*	8, 16, 32
Flash pass/fail result	FPFR	R/W	Undefined	R0 of CPU	8, 16, 32
Flash multipurpose address area	FMPAR	R/W	Undefined	R5 of CPU	8, 16, 32
Flash multipurpose data destination area	FMPDR	R/W	Undefined	R4 of CPU	8, 16, 32
Flash erase block select	FEBS	R/W	Undefined	R4 of CPU	8, 16, 32
Flash program and erase frequency control	FPEFEQ	R/W	Undefined	R4 of CPU	8, 16, 32
Flash user branch address set parameter	FUBRA	R/W	Undefined	R5 of CPU	8, 16, 32

Note: \* One byte of the start address in the on-chip RAM area specified by FTDAR is valid.

**Table 23.5 Register/Parameter and Target Mode**

		Download	Initiali- zation	Program- ming	Erase	Read	RAM Emulation
Programming/ erasing interface registers	FCCS	O	—	—	—	—	—
	FPCS	O	—	—	—	—	—
	PECS	O	—	—	—	—	—
	FKEY	O	—	O	O	—	—
	FMATS	—	—	O* <sup>1</sup>	O* <sup>1</sup>	O* <sup>2</sup>	—
	FTDAR	O	—	—	—	—	—
	Programming/ erasing interface parameters	DPFR	O	—	—	—	—
FPFR		O	O	O	O	—	—
FPEFEQ		—	O	—	—	—	—
FUBRA		—	O	—	—	—	—
FMPAR		—	—	O	—	—	—
FMPDR		—	—	O	—	—	—
FEBS		—	—	—	O	—	—
RAM emulation	RAMER	—	—	—	—	—	O

- Notes: 1. The setting is required when programming or erasing user MAT in user boot mode.  
2. The setting may be required according to the combination of initiation mode and read target MAT.

## 23.4.2 Programming/Erasing Interface Registers

The programming/erasing interface registers are as described below. They are all 8-bit registers that can be accessed in bytes. Except for the FLER bit in FCCS and FMATS, these registers are initialized at a power-on reset, in hardware standby mode, or in software standby mode. The FLER bit or FMATS is not initialized in software standby mode.

### (1) Flash Code Control and Status Register (FCCS)

FCCS is configured by bits which request the monitor of the FWE pin state and error occurrence during programming or erasing flash memory and the download of the on-chip program.

Bit :	7	6	5	4	3	2	1	0
	FWE	—	—	FLER	—	—	—	SCO
Initial value :	1/0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	(R)W

- Bit 7—Flash Programming Enable (FWE): Monitors the level which is input to the FWE pin that performs hardware protection of the flash memory programming or erasing. The initial value is 0 or 1 according to the FWE pin state.

#### Bit 7

FWE	Description
0	When the FWE pin goes low (in hardware protection state)
1	When the FWE pin goes high

- Bits 6 and 5—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 4—Flash Memory Error (FLER): Indicates an error occurs during programming and erasing flash memory.

When FLER is set to 1, flash memory enters the error protection state.

This bit is initialized at a power-on reset or in hardware standby mode.

When FLER is set to 1, high voltage is applied to the internal flash memory. To reduce the damage to flash memory, the reset signal must be released after the reset period of 100  $\mu$ s which is longer than normal.

## Bit 4

FLER	Description
0	Flash memory operates normally (Initial value) Programming/erasing protection for flash memory (error protection) is invalid. [Clearing condition] At a power-on reset or in hardware standby mode
1	Indicates an error occurs during programming/erasing flash memory. Programming/erasing protection for flash memory (error protection) is valid. [Setting condition] See section 23.6.3, Error Protection.

- Bits 3 to 1—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 0—Source Program Copy Operation (SCO): Requests the on-chip programming/erasing program to be downloaded to the on-chip RAM.

When this bit is set to 1, the on-chip program which is selected by FPCS/FECS is automatically downloaded in the on-chip RAM area specified by FTDAR.

In order to set this bit to 1, RAM emulation state must be canceled, H'A5 must be written to FKEY, and this operation must be in the on-chip RAM.

Four NOP instructions must be executed immediately after setting this bit to 1.

For interrupts during download, see section 23.8.2, Interrupts during Programming/Erasing. For the download time, see section 23.8.3, Other Notes.

Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1.

Download by setting the SCO bit to 1 requires a special interrupt processing that performs bank switching to the on-chip program storage area. Therefore, before issuing a download request (SCO = 1), set VBR to H'00000000. Otherwise, the CPU gets out of control. Once download end is confirmed, VBR can be changed to any other value.

## Bit 0

SCO	Description
0	Download of the on-chip programming/erasing program to the on-chip RAM is not executed (Initial value) [Clearing condition] When download is completed
1	Request that the on-chip programming/erasing program is downloaded to the on-chip RAM is generated [Clearing conditions] When all of the following conditions are satisfied and 1 is written to this bit <ul style="list-style-type: none"><li>• FKEY is written to H'A5</li><li>• During execution in the on-chip RAM</li><li>• Not in RAM emulation mode (RAMS in RAMCR = 0)</li></ul>



## (2) Flash Program Code Select Register (FPCS)

FPCS selects the on-chip programming program to be downloaded.

Bit :	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PPVS
Initial value :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R/W

- Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 0—Program Pulse Single (PPVS): Selects the programming program.

### Bit 0

PPVS	Description
0	On-chip programming program is not selected (Initial value) [Clearing condition] When transfer is completed
1	On-chip programming program is selected

## (3) Flash Erase Code Select Register (FECS)

FECS selects download of the on-chip erasing program.

Bit :	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	EPVB
Initial value :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R/W

- Bits 7 to 1—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 0—Erase Pulse Verify Block (EPVB): Selects the erasing program.

### Bit 0

EPVB	Description
0	On-chip erasing program is not selected (Initial value) [Clearing condition] When transfer is completed
1	On-chip erasing program is selected

## (4) Flash Key Code Register (FKEY)

FKEY is a register for software protection that enables download of the on-chip program and programming/erasing of flash memory. Before setting the SCO bit to 1 in order to download the on-chip program or executing the downloaded programming/erasing program, these processings cannot be executed if the key code is not written.

Bit :	7	6	5	4	3	2	1	0
	K7	K6	K5	K4	K3	K2	K1	K0
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

- Bits 7 to 0—Key Code (K7 to K0): Only when H'A5 is written, writing to the SCO bit is valid. When a value other than H'A5 is written to FKEY, 1 cannot be written to the SCO bit. Therefore downloading to the on-chip RAM cannot be executed.

Only when H'5A is written, programming/erasing of flash memory can be executed. Even if the on-chip programming/erasing program is executed, flash memory cannot be programmed or erased when a value other than H'5A is written to FKEY.

### Bits 7 to 0

K7 to K0	Description
H'A5	Writing to the SCO bit is enabled (The SCO bit cannot be set by a value other than H'A5.)
H'5A	Programming/erasing is enabled (A value other than H'A5 enables software protection state.)
H'00	Initial value

### (5) Flash MAT Select Register (FMATS)

FMATS specifies whether user MAT or user boot MAT is selected.

Bit :	7	6	5	4	3	2	1	0	
	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
Initial value :	0	0	0	0	0	0	0	0	(When not in user boot mode)
Initial value :	1	0	1	0	1	0	1	0	(When in user boot mode)
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- Bits 7 to 0—MAT Select (MS7 to MS0): These bits are in user-MAT selection state when a value other than H'AA is written and in user-boot-MAT selection state when H'AA is written. The MAT is switched by writing a value in FMATS.

When the MAT is switched, follow section 23.8.1, Switching between User MAT and User Boot MAT. (The user boot MAT cannot be programmed in user programming mode if user boot MAT is selected by FMATS. The user boot MAT must be programmed in boot mode or in programmer mode.)

## Bits 7 to 0

MS7 to MS0	Description
H'AA	The user boot MAT is selected (in user-MAT selection state when the value of these bits are other than H'AA) Initial value when these bits are initiated in user boot mode.
H'00	Initial value when these bits are initiated in a mode except for user boot mode (in user-MAT selection state)

[Programmable condition] These bits are in the execution state in the on-chip RAM.

### (6) Flash Transfer Destination Address Register (FTDAR)

FTDAR specifies the on-chip RAM address to which the on-chip program is downloaded.

Make settings for FTDAR before writing 1 to the SCO bit in FCCS. The initial value is H'00 which points to the start address (H'FFF0000) in on-chip RAM.

Bit :	7	6	5	4	3	2	1	0
	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0
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 7—Transfer Destination Address Setting Error: This bit is set to 1 when there is an error in the download start address set by bits 6 to 0 (TDA6 to TDA0). Whether the address setting is erroneous or not is judged by checking whether the setting of TDA6 to TDA0 is between the range of H'00 and H'05 after setting the SCO bit in FCCS to 1 and performing download. Before setting the SCO bit to 1 be sure to set the FTDAR value between H'00 to H'05 as well as clearing this bit to 0.

### Bit 7

TDER	Description (Return Value after Download)
0	Setting of TDA6 to TDA0 is normal (Initial value)
1	Setting of TDER and TDA6 to TDA0 is H'06 to H'FF and download has been aborted

- Bits 6 to 0—Transfer Destination Address (TDA6 to TDA0): These bits specify the download start address. A value from H'00 to H'05 can be set to specify the download start address in on-chip RAM in 2-kbyte units.

A value from H'06 to H'FF cannot be set. If such a value is set, the TDER bit (bit 7) in this register is set to 1 to prevent download from being executed.

TDA6 to TDA0	Description
H'00	Download start address is set to H'FFFF0000
H'01	Download start address is set to H'FFFF0800
H'02	Download start address is set to H'FFFF1000
H'03	Download start address is set to H'FFFF1800
H'04	Download start address is set to H'FFFF2000
H'05	Download start address is set to H'FFFF2800
H'06 to H'FF	Setting prohibited. If this value is set, the TDER bit (bit 7) is set to 1 to abort the download processing.

### 23.4.3 Programming/Erasing Interface Parameters

The programming/erasing interface parameters specify the operating frequency, user branch destination address, storage place for program data, programming destination address, and erase block and exchanges the processing result for the downloaded on-chip program. This parameter uses the general registers of the CPU (R4, R5, and R0) or the on-chip RAM area. The initial value is undefined at a power-on reset or in hardware standby mode.

At download all CPU registers are stored, and at initialization or when the on-chip program is executed, CPU registers except for R0 are stored. The return value of the processing result is written in R0. Since the stack area is used for storing the registers or as a work area, the stack area must be saved at the processing start. (The maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameters are used in the following four items.

- (1) Download control
- (2) Initialization before programming or erasing
- (3) Programming
- (4) Erasing

These items use different parameters. The correspondence table is shown in table 23.6.

The processing results of initialization, programming, and erasing are returned, but the bit contents have different meanings according to the processing program. See the description of FPFRR for each processing.

**Table 23.6 Usable Parameters and Target Modes**

Name of Parameter	Abbreviation	Download	Initialization	Programming	Erase	R/W	Initial Value	Allocation
Download pass/fail result	DPFR	0	—	—	—	R/W	Undefined	On-chip RAM*
Flash pass/fail result	FPFR	—	0	0	0	R/W	Undefined	R0 of CPU
Flash programming/erasing frequency control	FPEFEQ	—	0	—	—	R/W	Undefined	R4 of CPU
Flash user branch address set parameter	FUBRA	—	0	—	—	R/W	Undefined	R5 of CPU
Flash multipurpose address area	FMPAR	—	—	0	—	R/W	Undefined	R5 of CPU
Flash multipurpose data destination area	FMPDR	—	—	0	—	R/W	Undefined	R4 of CPU
Flash erase block select	FEBS	—	—	—	0	R/W	Undefined	R4 of CPU

Note: \*One byte of start address of download destination specified by FTDAR

### (1) Download Control

The on-chip program is automatically downloaded by setting the SCO bit to 1. The on-chip RAM area to be downloaded is the area as much as 2 kbytes starting from the start address specified by FTDAR. For the address map of the on-chip RAM, see figure 23.10.

The download control is set by using the programming/erasing interface registers. The return value is given by the DPFR parameter.

#### (a) Download pass/fail result parameter (DPFR: one byte of start address of on-chip RAM specified by FTDAR)

This parameter indicates the return value of the download result. The value of this parameter can be used to determine if downloading is executed or not. Since the confirmation whether the SCO bit is set to 1 is difficult, the certain determination must be performed by setting one byte of the start address of the on-chip RAM area specified by FTDAR to a value other than the return value of download (for example, H'FF) before the download start (before setting the SCO bit to 1). For the checking method of download results, see section 23.5.2, User Program Mode.

Bit :	7	6	5	4	3	2	1	0
	0	0	0	0	0	SS	FK	SF

- Bits 7 to 3—Unused: Return 0.
- Bit 2—Source Select Error Detect (SS): The on-chip program which can be downloaded can be specified as only one type. When more than two types of the program are selected, the program is not selected, or the program is selected without mapping, an error occurs.

#### Bit 2

SS	Description
0	Download program can be selected normally
1	Download error occurs (Multi-selection or program which is not mapped is selected)

- Bit 1—Flash Key Register Error Detect (FK): Returns the check result whether the value of FKEY is set to H'A5.

#### Bit 1

FK	Description
0	FKEY setting is normal (FKEY = H'A5)
1	FKEY setting is abnormal (FKEY = value other than H'A5)

- Bit 0—Success/Fail (SF): Returns the result whether download has ended normally or not.

#### Bit 0

SF	Description
0	Downloading on-chip program has ended normally (no error)
1	Downloading on-chip program has ended abnormally (error occurs)

## (2) Programming/Erasing Initialization

The on-chip programming/erasing program to be downloaded includes the initialization program.

The specified period pulse must be applied when programming or erasing. The specified pulse width is made by the method in which wait loop is configured by the CPU instruction. The operating frequency of the CPU must be set. Since the user branch function is supported, the user branch destination address must be set.

The initial program is set as a parameter of the programming/erasing program which has downloaded these settings.

### (2.1) Flash programming/erasing frequency parameter (FPEFEQ: general register R4 of CPU)

This parameter sets the operating frequency of the CPU.

For the range of the operating frequency of this LSI, see section 27.3.2, Clock Timing.

Bit :	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0

Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0

Bit :	15	14	13	12	11	10	9	8
	F15	F14	F13	F12	F11	F10	F9	F8

Bit :	7	6	5	4	3	2	1	0
	F7	F6	F5	F4	F3	F2	F1	F0

- Bits 31 to 16—Unused: Return 0.
- Bits 15 to 0—Frequency Set (F15 to F0): Set the operating frequency of the CPU. The setting value must be calculated as the following methods.
  1. The operating frequency which is shown in MHz units must be rounded in a number to three decimal places and be shown in a number of two decimal places.
  2. The centuplicated value is converted to the binary digit and is written to the FPEFEQ parameter (general register R4). For example, when the operating frequency of the CPU is 28.882 MHz, the value is as follows.
    1. The number to three decimal places of 28.882 is rounded and the value is thus 28.88.
    2. The formula that  $28.88 \times 100 = 2888$  is converted to the binary digit and b'0000, 1011, 0100, 1000 (H'0B48) is set to R4.

## (2.2) Flash user branch address setting parameter (FUBRA: general register R5 of CPU)

This parameter sets the user branch destination address. The user program which has been set can be executed in specified processing units when programming and erasing.

Bit :	31	30	29	28	27	26	25	24
	UA31	UA30	UA29	UA28	UA27	UA26	UA25	UA24

Bit :	23	22	21	20	19	18	17	16
	UA23	UA22	UA21	UA20	UA19	UA18	UA17	UA16

Bit :	15	14	13	12	11	10	9	8
	UA15	UA14	UA13	UA12	UA11	UA10	UA9	UA8

Bit :	7	6	5	4	3	2	1	0
	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0

- Bits 31 to 0—User Branch Destination Address (UA31 to UA0): When the user branch is not required, address 0 (H'00000000) must be set.

The user branch destination must be an area other than the flash memory, an area other than the RAM area in which on-chip program has been transferred, or the external bus space.

Note that the CPU must not branch to an area without the execution code and get out of control. The on-chip program download area and stack area must not be overwritten. If CPU runaway occurs or the download area or stack area is overwritten, the value of flash memory cannot be guaranteed.

The download of the on-chip program, initialization, initiation of the programming/erasing program must not be executed in the processing of the user branch destination. Programming or erasing cannot be guaranteed when returning from the user branch destination. The program data which has already been prepared must not be programmed.

Store general registers R8 to R15 and the control register GBR. General registers R0 to R7 are available without storing them.

Moreover, the programming/erasing interface registers must not be written to or RAM emulation mode must not be entered in the processing of the user branch destination.

After the processing of the user branch has ended, the programming/erasing program must be returned to by using the RTS instruction.

For the execution intervals of the user branch processing, see note 2 (User branch processing intervals) in section 23.8.3, Other Notes.

### (2.3) Flash pass/fail result parameter (FPFR: general register R0 of CPU)

This parameter indicates the return value of the initialization result.

Bit :	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0
Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
Bit :	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
Bit :	7	6	5	4	3	2	1	0
	0	0	0	0	0	BR	FQ	SF

- Bits 31 to 3—Unused: Return 0.
- Bit 2—User Branch Error Detect (BR): Returns the check result whether the specified user branch destination address is in the area other than the storage area of the programming/erasing program which has been downloaded .



## Bit 2

BR	Description
0	User branch address setting is normal
1	User branch address setting is abnormal

- Bit 1—Frequency Error Detect (FQ): Returns the check result whether the specified operating frequency of the CPU is in the range of the supported operating frequency.

## Bit 1

FQ	Description
0	Setting of operating frequency is normal
1	Setting of operating frequency is abnormal

- Bit 0—Success/Fail (SF): Indicates whether initialization is completed normally.

## Bit 0

SF	Description
0	Initialization has ended normally (no error)
1	Initialization has ended abnormally (error occurs)

### (3) Programming Execution

When flash memory is programmed, the programming destination address on the user MAT must be passed to the programming program in which the program data is downloaded.

1. The start address of the programming destination on the user MAT is set in general register R5 of the CPU. This parameter is called FMPAR (flash multipurpose address area parameter).

Since the program data is always in 128-byte units, the lower eight bits (MOA7 to MOA0) must be H'00 or H'80 as the boundary of the programming start address on the user MAT.

2. The program data for the user MAT must be prepared in the consecutive area. The program data must be in the consecutive space which can be accessed by using the MOV.B instruction of the CPU and is not the flash memory space.

When data to be programmed does not satisfy 128 bytes, the 128-byte program data must be prepared by embedding the dummy code (H'FF).

The start address of the area in which the prepared program data is stored must be set in general register R4. This parameter is called FMPDR (flash multipurpose data destination area parameter).

For details on the programming procedure, see section 23.5.2, User Program Mode.

#### (3.1) Flash multipurpose address area parameter (FMPAR: general register R5 of CPU)

This parameter indicates the start address of the programming destination on the user MAT.

When an address in an area other than the flash memory space is set, an error occurs.

The start address of the programming destination must be at the 128-byte boundary. If this boundary condition is not satisfied, an error occurs. The error occurrence is indicated by the WA bit (bit 1) in FPCR.

Bit :	31	30	29	28	27	26	25	24
	MOA31	MOA30	MOA29	MOA28	MOA27	MOA26	MOA25	MOA24
Bit :	23	22	21	20	19	18	17	16
	MOA23	MOA22	MOA21	MOA20	MOA19	MOA18	MOA17	MOA16
Bit :	15	14	13	12	11	10	9	8
	MOA15	MOA14	MOA13	MOA12	MOA11	MOA10	MOA9	MOA8
Bit :	7	6	5	4	3	2	1	0
	MOA7	MOA6	MOA5	MOA4	MOA3	MOA2	MOA1	MOA0

- Bits 31 to 0—MOA31 to MOA0: Store the start address of the programming destination on the user MAT. The consecutive 128-byte programming is executed starting from the specified start address of the user MAT. The MOA6 to MOA0 bits are always 0 because the start address of the programming destination is at the 128-byte boundary.

### (3.2) Flash multipurpose data destination parameter (FMPDR: general register R4 of CPU)

This parameter indicates the start address in the area which stores the data to be programmed in the user MAT. When the storage destination of the program data is in flash memory, an error occurs. The error occurrence is indicated by the WD bit (bit 2) in FPCR.

Bit :	31	30	29	28	27	26	25	24
	MOD31	MOD30	MOD29	MOD28	MOD27	MOD26	MOD25	MOD24
Bit :	23	22	21	20	19	18	17	16
	MOD23	MOD22	MOD21	MOD20	MOD19	MOD18	MOD17	MOD16
Bit :	15	14	13	12	11	10	9	8
	MOD15	MOD14	MOD13	MOD12	MOD11	MOD10	MOD9	MOD8
Bit :	7	6	5	4	3	2	1	0
	MOD7	MOD6	MOD5	MOD4	MOD3	MOD2	MOD1	MOD0

- Bits 31 to 0—MOD31 to MOD0: Store the start address of the area which stores the program data for the user MAT. The consecutive 128-byte data is programmed to the user MAT starting from the specified start address.

### (3.3) Flash pass/fail parameter (FPFR: general register R0 of CPU)

This parameter indicates the return value of the program processing result.

Bit :	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0
Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
Bit :	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
Bit :	7	6	5	4	3	2	1	0
	0	MD	EE	FK	0	WD	WA	SF

- Bits 31 to 7—Unused: Return 0.
- Bit 6—Programming Mode Related Setting Error Detect (MD): Returns the check result of whether the signal input to the FWE pin is high and whether the error protection state is entered.

When a low-level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The input level to the FWE pin and the error protection state can be confirmed with the FWE bit (bit 7) and the FLER bit (bit 4) in FCCS, respectively. For conditions to enter the error protection state, see section 23.6.3, Error Protection.

#### Bit 6

MD	Description
0	FWE and FLER settings are normal (FWE = 1, FLER = 0)
1	FWE = 0 or FLER = 1, and programming cannot be performed

- Bit 5—Programming Execution Error Detect (EE): 1 is returned to this bit when the specified data could not be written because the user MAT was not erased or when flash-memory related register settings are partially changed on returning from the user branch processing.

If this bit is set to 1, there is a high possibility that the user MAT is partially rewritten. In this case, after removing the error factor, erase the user MAT.

If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when programming is performed. In this case, both the user MAT and user boot MAT are not rewritten.

Programming of the user boot MAT must be executed in boot mode or programmer mode.

**Bit 5**

EE	Description
0	Programming has ended normally
1	Programming has ended abnormally (programming result is not guaranteed)

- Bit 4—Flash Key Register Error Detect (FK): Returns the check result of the value of FKEY before the start of the programming processing.

**Bit 4**

FK	Description
0	FKEY setting is normal (FKEY = H'A5)
1	FKEY setting is error (FKEY = value other than H'A5)

- Bit 3—Unused: Returns 0.
- Bit 2—Write Data Address Detect (WD): When an address in the flash memory area is specified as the start address of the storage destination of the program data, an error occurs.

**Bit 2**

WD	Description
0	Setting of write data address is normal
1	Setting of write data address is abnormal

- Bit 1—Write Address Error Detect (WA): When the following items are specified as the start address of the programming destination, an error occurs.
  1. The programming destination address is an area other than flash memory
  2. The specified address is not at the 128-byte boundary (A6 to A0 are not 0)

**Bit 1**

WA	Description
0	Setting of programming destination address is normal
1	Setting of programming destination address is abnormal

- Bit 0—Success/Fail (SF): Indicates whether the program processing has ended normally or not.

**Bit 0**

SF	Description
0	Programming has ended normally (no error)
1	Programming has ended abnormally (error occurs)

#### (4) Erasure Execution

When flash memory is erased, the erase-block number on the user MAT must be passed to the erasing program which is downloaded. This is set to the FEBS parameter (general register R4). One block is specified from the block number 0 to 15.

For details on the erasing procedure, see section 23.5.2, User Program Mode.

##### (4.1) Flash erase block select parameter (FEBS: general register R4 of CPU)

This parameter specifies the erase-block number. Several block numbers cannot be specified.

Bit :	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0
Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
Bit :	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
Bit :	7	6	5	4	3	2	1	0
	EBS7	EBS6	EBS5	EBS4	EBS3	EBS2	EBS1	EBS0

- Bits 31 to 8—Unused: Return 0.
- Bits 7 to 0—Erase Block (EB7 to EB0): Set the erase-block number in the range from 0 to 15. 0 corresponds to the EB0 block and 15 corresponds to the EB15 block. An error occurs when a number other than 0 to 15 (H'00 to H'0F) is set.

##### (4.2) Flash pass/fail result parameter (FPFR: general register R0 of CPU)

This parameter returns the value of the erasing processing result.

Bit :	31	30	29	28	27	26	25	24
	0	0	0	0	0	0	0	0
Bit :	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
Bit :	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
Bit :	7	6	5	4	3	2	1	0
	0	MD	EE	FK	EB	0	0	SF

- Bits 31 to 7—Unused: Return 0.

- Bit 6—Erasure Mode Related Setting Error Detect (MD): Returns the check result of whether the signal input to the FWE pin is high and whether the error protection state is entered.

When a low-level signal is input to the FWE pin or the error protection state is entered, 1 is written to this bit. The input level to the FWE pin and the error protection state can be confirmed with the FWE bit (bit 7) and the FLER bit (bit 4) in FCCS, respectively. For conditions to enter the error protection state, see section 23.6.3, Error Protection.

#### Bit 6

MD	Description
0	FWE and FLER settings are normal (FWE = 1, FLER = 0)
1	FWE = 0 or FLER = 1, and erasure cannot be performed

- Bit 5—Erasure Execution Error Detect (EE): 1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed on returning from the user branch processing.

If this bit is set to 1, there is a high possibility that the user MAT is partially erased. In this case, after removing the error factor, erase the user MAT.

If FMATS is set to H'AA and the user boot MAT is selected, an error occurs when erasure is performed. In this case, both the user MAT and user boot MAT are not erased.

Erasure of the user boot MAT must be executed in boot mode or programmer mode.

#### Bit 5

EE	Description
0	Erasure has ended normally
1	Erasure has ended abnormally (erasure result is not guaranteed)

- Bit 4—Flash Key Register Error Detect (FK): Returns the check result of FKEY value before start of the erasing processing.

#### Bit 4

FK	Description
0	FKEY setting is normal (FKEY = H'5A)
1	FKEY setting is error (FKEY = value other than H'5A)

- Bit 3—Erase Block Select Error Detect (EB): Returns the check result whether the specified erase-block number is in the block range of the user MAT.

### Bit 3

EB	Description
0	Setting of erase-block number is normal
1	Setting of erase-block number is abnormal

- Bits 2 and 1—Unused: Return 0.
- Bit 0—Success/Fail (SF): Indicates whether the erasing processing has ended normally or not.

### Bit 0

SF	Description
0	Erase has ended normally (no error)
1	Erase has ended abnormally (error occurs)

## 23.4.4 RAM Emulation Register (RAMER)

When the realtime programming of the user MAT is emulated, RAMER sets the area of the user MAT which is overlapped with a part of the on-chip RAM. RAMER is initialized to H'0000 at a power-on reset or in hardware standby mode and is not initialized in software standby mode. The RAMER setting must be executed in user mode or in user program mode.

For the division method of the user-MAT area, see table 23.7. In order to operate the emulation function certainly, the target MAT of the RAM emulation must not be accessed immediately after RAMER is programmed. If it is accessed, the normal access is not guaranteed.

Bit :	15	14	13	12	11	10	9	8
	—	—	—	—	—	—	—	—
Initial value :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R	R	R	R
Bit :	7	6	5	4	3	2	1	0
	—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value :	0	0	0	0	0	0	0	0
R/W :	R	R	R	R	R/W	R/W	R/W	R/W

- Bits 15 to 4—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 3—RAM Select (RAMS): Sets whether the user MAT is emulated or not. When RAMS = 1, all blocks of the user MAT are in the programming/erasing protection state.

**Bit 3**

<b>RAMS</b>	<b>Description</b>
0	Emulation is not selected Programming/erasing protection of all user-MAT blocks is invalid (Initial value)
1	Emulation is selected Programming/erasing protection of all user-MAT blocks is valid

- Bits 2 to 0—User MAT Area Select: These bits are used with bit 3 to select the user-MAT area to be overlapped with the on-chip RAM. (See table 23.7.)

**Table 23.7 Overlapping of RAM Area and User MAT Area**

<b>RAM Area</b>	<b>Block Name</b>	<b>RAMS</b>	<b>RAM2</b>	<b>RAM1</b>	<b>RAM0</b>
H'FFFF0000 to H'FFFF0FFF	RAM area (4 kbytes)	0	*	*	*
H'00000000 to H'00000FFF	EB0 (4 kbytes)	1	0	0	0
H'00001000 to H'00001FFF	EB1 (4 kbytes)	1	0	0	1
H'00002000 to H'00002FFF	EB2 (4 kbytes)	1	0	1	0
H'00003000 to H'00003FFF	EB3 (4 kbytes)	1	0	1	1
H'00004000 to H'00004FFF	EB4 (4 kbytes)	1	1	0	0
H'00005000 to H'00005FFF	EB5 (4 kbytes)	1	1	0	1
H'00006000 to H'00006FFF	EB6 (4 kbytes)	1	1	1	0
H'00007000 to H'00007FFF	EB7 (4 kbytes)	1	1	1	1

Note: \* Don't care.



## 23.5 On-Board Programming Mode

When the pin is set in on-board programming mode and the reset start is executed, the on-board programming state that can program/erase the on-chip flash memory is entered. On-board programming mode has three operating modes: user programming mode, user boot mode, and boot mode.

For details on the pin setting for entering each mode, see table 23.1. For details on the state transition of each mode for flash memory, see figure 23.2.

### 23.5.1 Boot Mode

Boot mode executes programming/erasing user MAT and user boot MAT by means of the control command and program data transmitted from the host using the on-chip SCI. The tool for transmitting the control command and program data must be prepared in the host. The SCI communication mode is set to asynchronous mode. When reset start is executed after this LSI's pin is set in boot mode, the boot program in the microcomputer is initiated. After the SCI bit rate is automatically adjusted, the communication with the host is executed by means of the control command method.

The system configuration diagram in boot mode is shown in figure 23.6. For details on the pin setting in boot mode, see table 23.1. Interrupts are ignored in boot mode, so do not generate them. Note that the AUD cannot be used during boot mode operation.

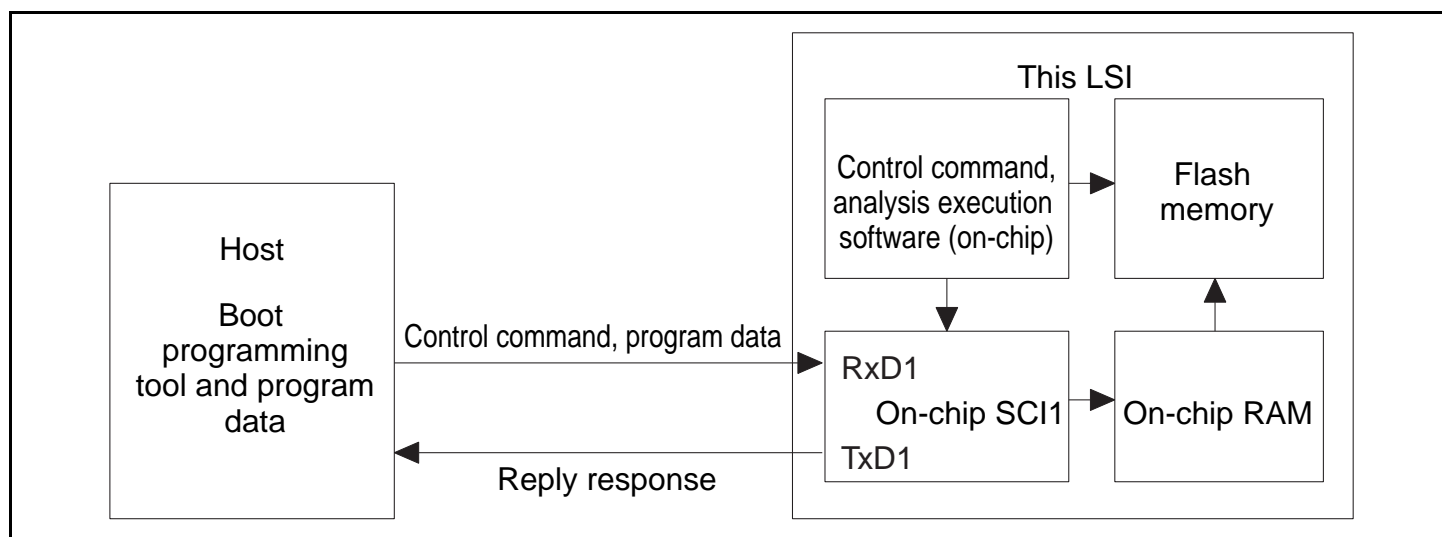
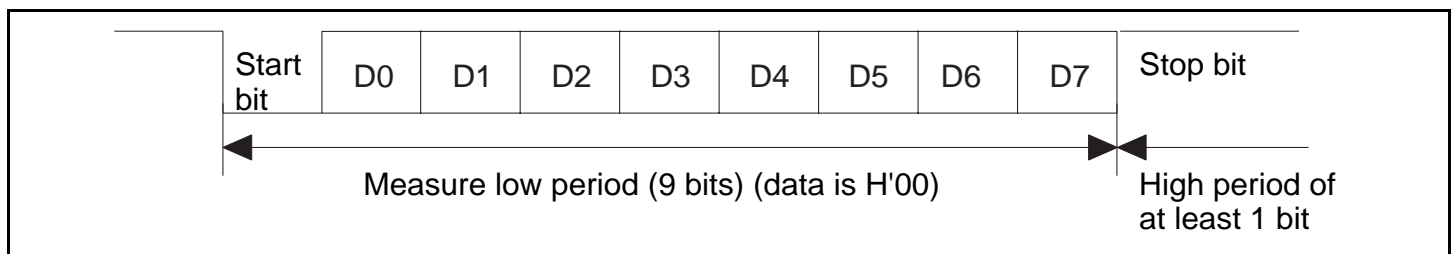


Figure 23.6 System Configuration in Boot Mode

## (1) SCI Interface Setting by Host

When boot mode is initiated, this LSI measures the low period of asynchronous SCI-communication data (H'00), which is transmitted consecutively by the host. The SCI transmit/receive format is set to 8-bit data, 1 stop bit, and no parity. This LSI calculates the bit rate of transmission by the host by means of the measured low period and transmits the bit adjustment end sign (1 byte of H'00) to the host. The host must confirm that this bit adjustment end sign (H'00) has been received normally and transmits 1 byte of H'55 to this LSI. When reception is not executed normally, boot mode is initiated again (reset) and the operation described above must be executed. The bit rate between the host and this LSI is not matched because of the bit rate of transmission by the host and system clock frequency of this LSI. To operate the SCI normally, the transfer bit rate of the host must be set to 9,600 bps or 19,200 bps.

The system clock frequency which can automatically adjust the transfer bit rate of the host and the bit rate of this LSI is shown in table 23.8. Boot mode must be initiated in the range of this system clock.



**Figure 23.7 Automatic Adjustment Operation of SCI Bit Rate**

**Table 23.8 System Clock Frequency that Can Automatically Adjust Bit Rate of This LSI**

Host Bit Rate	System Clock Frequency Which Can Automatically Adjust LSI's Bit Rate
9,600 bps	20 to 40 MHz (input frequency of 5 to 10 MHz)
19,200 bps	20 to 40 MHz (input frequency of 5 to 10 MHz)

## (2) State Transition

The overview of the state transition after boot mode is initiated is shown in figure 23.8. For details on boot mode, see section 23.10.1, Serial Communications Interface Specification for Boot Mode.

### 1. Bit rate adjustment

After boot mode is initiated, the bit rate of the SCI interface is adjusted with that of the host.

### 2. Waiting for inquiry set command

For inquiries about the user-MAT size and configuration, MAT start address, and support state, the required information is transmitted to the host.

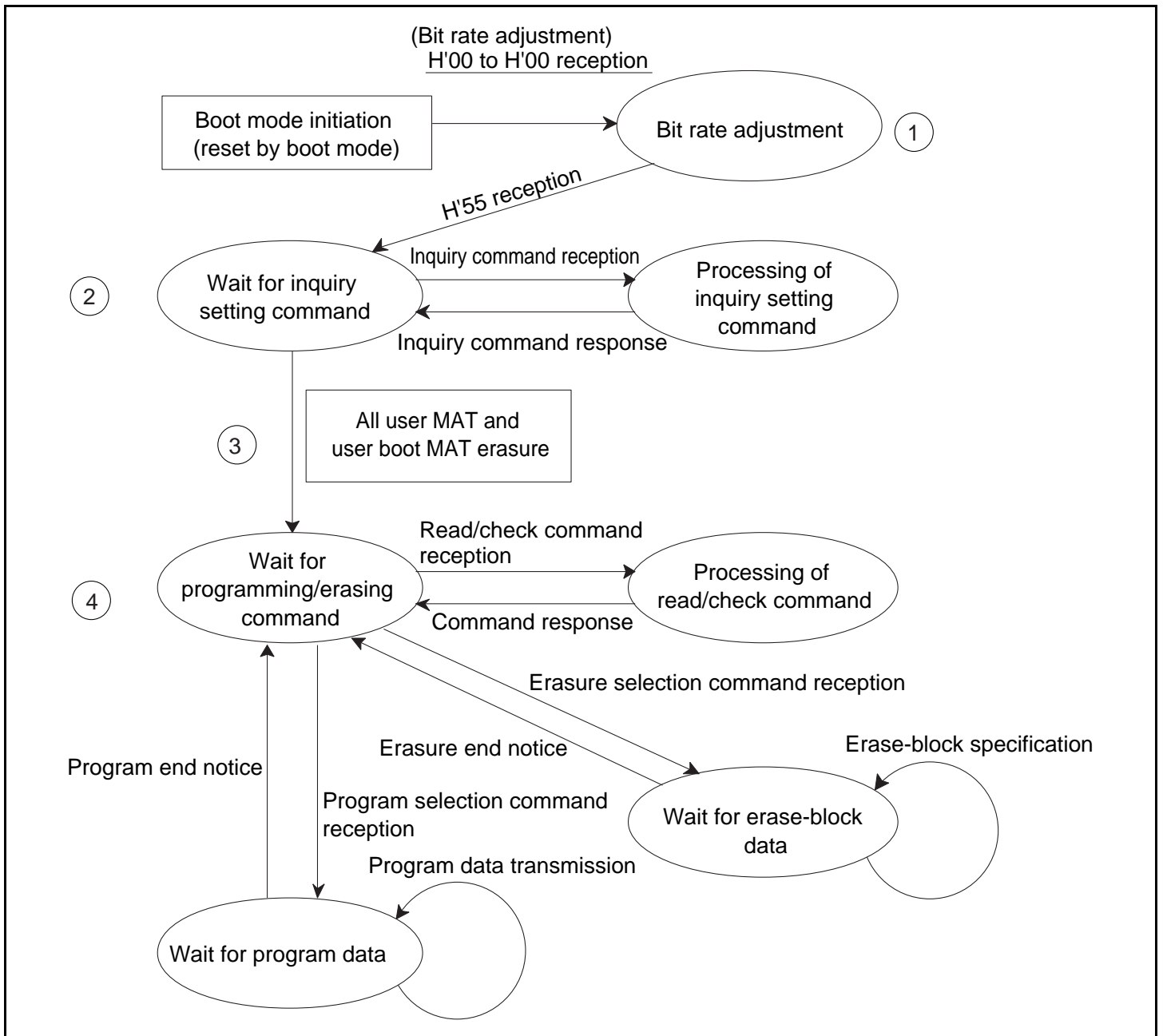
3. Automatic erasure of all user MAT and user boot MAT

After inquiries have finished, all of the user MAT and user boot MAT are automatically erased if a programming/erasing status transition command is sent.

4. Waiting for programming/erasing command

- When the program selection command is received, the state for waiting program data is entered. The programming start address and program data must be transmitted following the programming command. When programming is finished, the programming start address must be set to H'FFFFFFFF and transmitted. Then the state for waiting program data is returned to the state of programming/erasing command wait.
- When the erasure selection command is received, the state for waiting erase-block data is entered. The erase-block number must be transmitted following the erasing command. When the erasure is finished, the erase-block number must be set to H'FF and transmitted. Then the state for waiting erase-block data is returned to the state for waiting programming/erasing command. The erasure must be executed when reset start is not executed and the specified block is programmed after programming is executed in boot mode. When programming can be executed by only one operation, all blocks are erased before the state for waiting programming/erasing/other command is entered. The erasing operation is not required.
- There are many commands other than programming/erasing. Examples are sum check, blank check (erasure check), and memory read of the user MAT/user boot MAT and acquisition of current status information.

Note that memory read of the user MAT/user boot MAT can only read the program data after all user MAT/user boot MAT has automatically been erased.



**Figure 23.8 Overview of Boot Mode State Transition**

## 23.5.2 User Program Mode

The user MAT can be programmed/erased in user program mode. (The user boot MAT cannot be programmed/erased.)

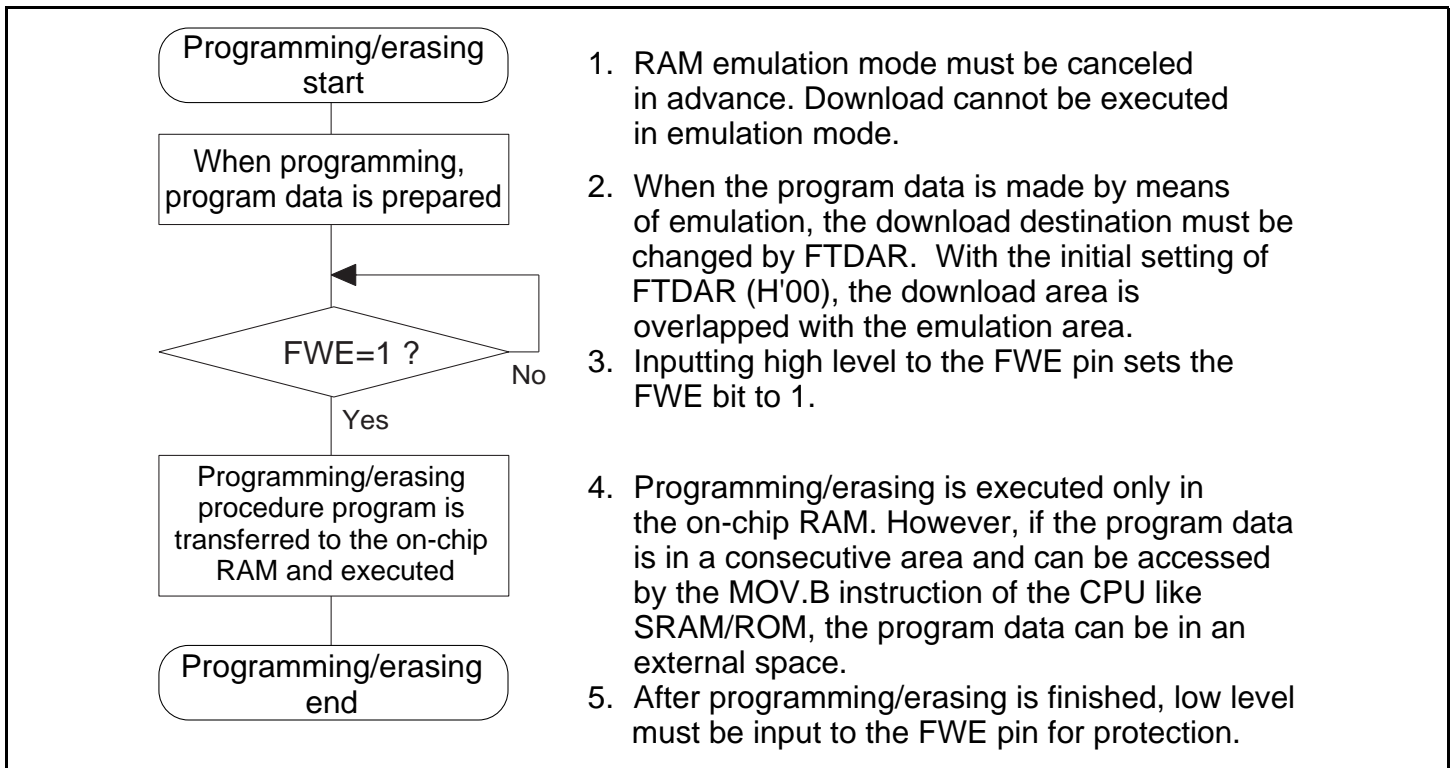
Programming/erasing is executed by downloading the program in the microcomputer.

The overview flow is shown in figure 23.9.

High voltage is applied to internal flash memory during the programming/erasing processing. Therefore, transition to reset or hardware standby mode must not be executed. Doing so may cause damage or destroy flash memory. If reset is executed accidentally, the reset signal must be released after the reset input period, which is longer than the normal 100  $\mu$ s.

For details on the programming procedure, see the description in 23.5.2 (2) Programming Procedure in User Program Mode. For details on the erasing procedure, see the description in 23.5.2 (3) Erasing Procedure in User Program Mode.

For the overview of a processing that repeats erasing and programming by downloading the programming program and the erasing program in separate on-chip ROM areas using FTDAR, see the description in 23.5.2 (4) Erasing and Programming Procedure in User Program Mode.

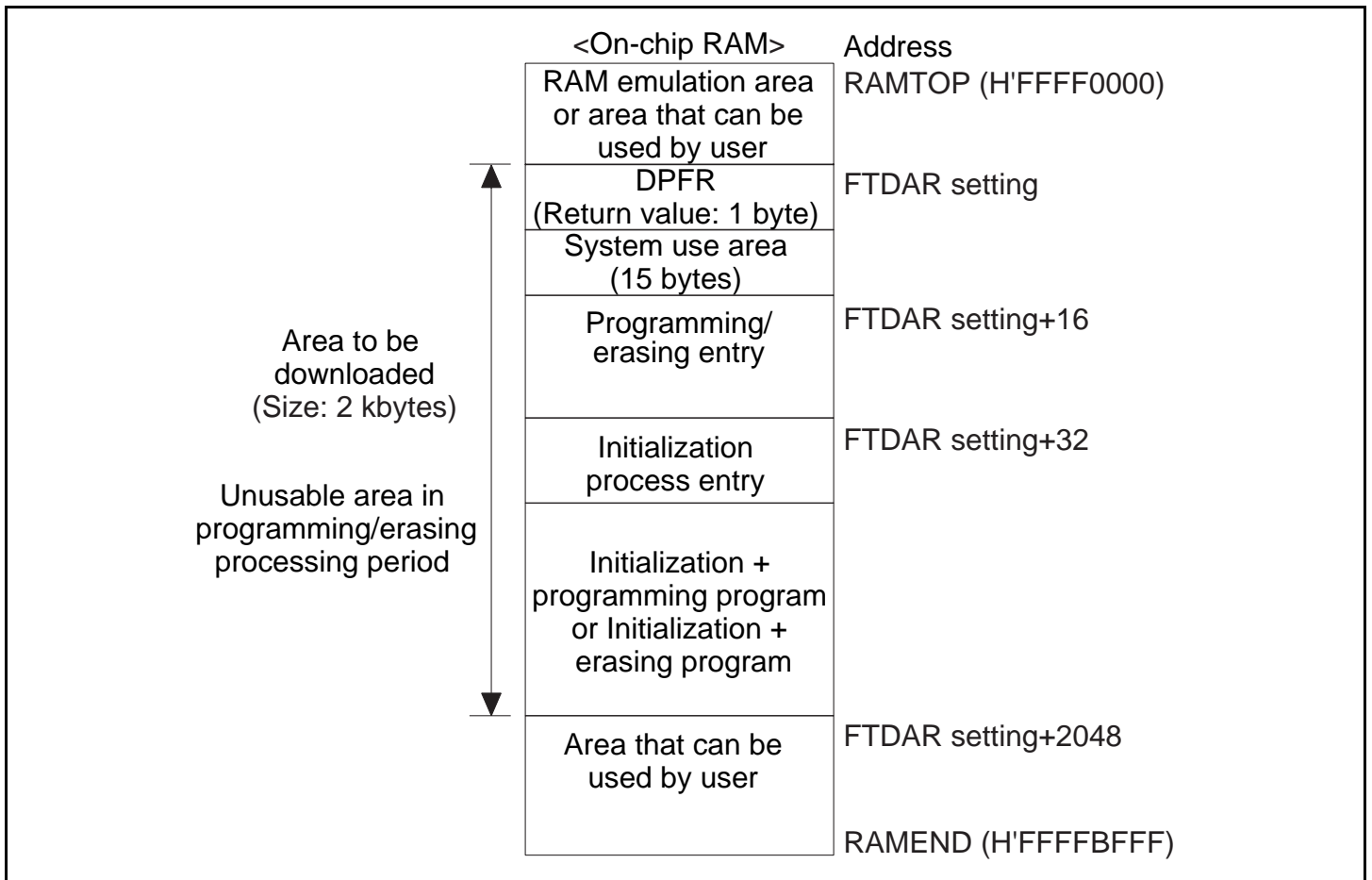


**Figure 23.9 Programming/Erasing Overview Flow**

(1) On-Chip RAM Address Map when Programming/Erasing is Executed

Parts of the procedure program that are made by the user, like download request, programming/erasing procedure, and judgement of the result, must be executed in the on-chip RAM. All of the on-chip program that is to be downloaded is in on-chip RAM. Note that on-chip RAM must be controlled so that these parts do not overlap.

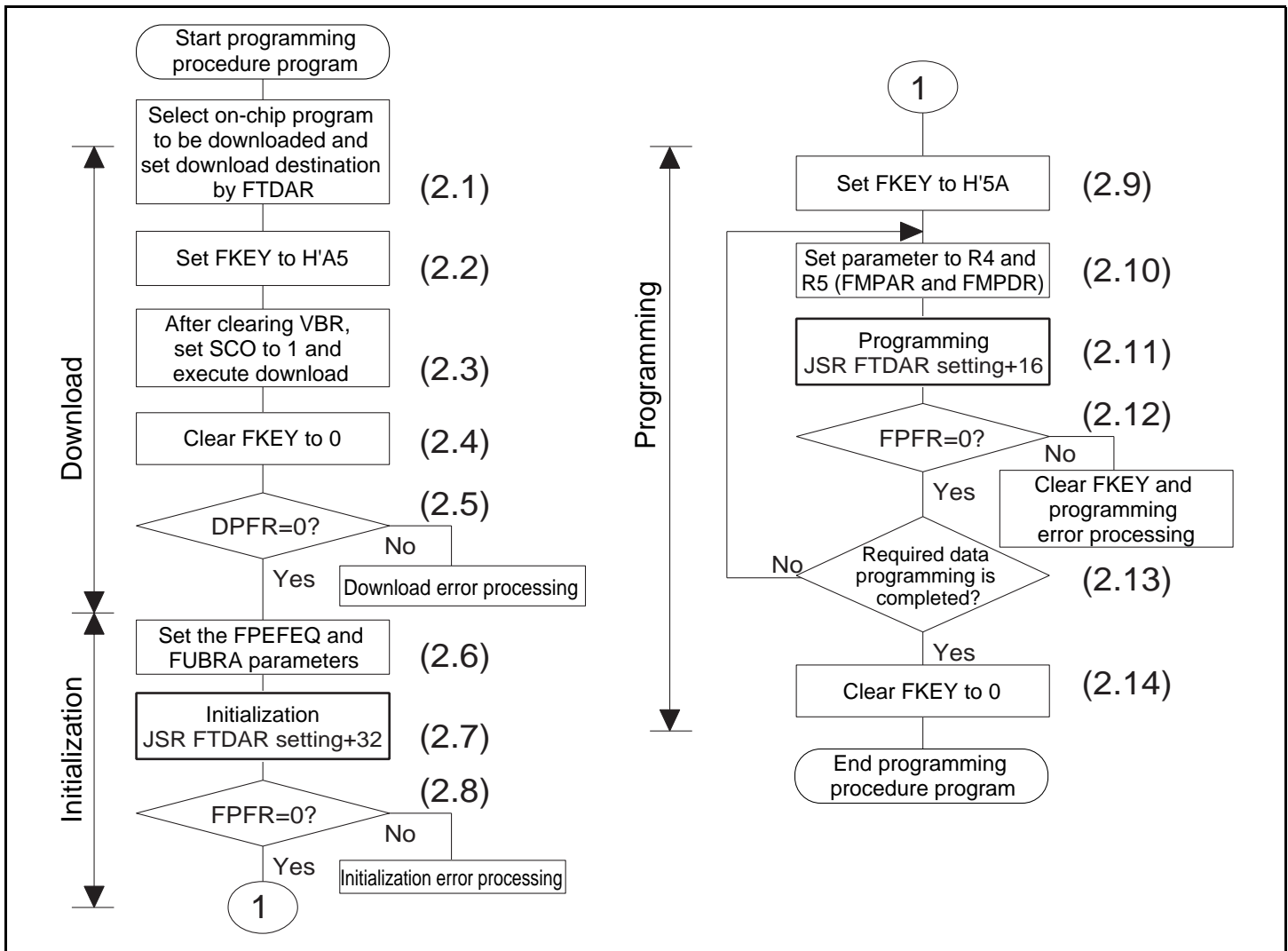
Figure 23.10 shows the program area to be downloaded.



**Figure 23.10 RAM Map after Download**

## (2) Programming Procedure in User Program Mode

The procedures for download, initialization, and programming are shown in figure 23.11.



**Figure 23.11 Programming Procedure**

The details of the programming procedure are described below. The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM. The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 23.10.3, Storable Area for Procedure Program and programming Data.

The following description assumes the area to be programmed on the user MAT is erased and program data is prepared in the consecutive area. When erasing has not been executed, carry out erasing before writing.

128-byte programming is performed in one program processing. When more than 128-byte programming is performed, programming destination address/program data parameter is updated in 128-byte units and programming is repeated.

When less than 128-byte programming is performed, data must total 128 bytes by adding the invalid data. If the invalid data to be added is H'FF, the program processing period can be shortened.

(2.1) Select the on-chip program to be downloaded

When the PPVS bit of FPCS is set to 1, the programming program is selected.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the source select error detect (SS) bit in the DPFR parameter.

Specify the start address of the download destination by FTDAR.

(2.2) Write H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be written to the SCO bit for a download request.

(2.3) VBR is cleared to 0 and 1 is written to the SCO bit of FCCS, and then download is executed.

VBR must always be cleared to H'00000000 before setting the SCO bit to 1.

To write 1 to the SCO bit, the following conditions must be satisfied.

- RAM emulation mode is canceled.
- H'A5 is written to FKEY.
- The SCO bit writing is executed in the on-chip RAM.

When the SCO bit is set to 1, download is started automatically. When execution returns to the user procedure program, the SCO bit is cleared to 0. Therefore, the SCO bit cannot be confirmed to be 1 in the user procedure program.

The download result can be confirmed only by the return value of the DPFR parameter. Before the SCO bit is set to 1, incorrect judgement must be prevented by setting the DPFR parameter, that is one byte of the start address of the on-chip RAM area specified by FTDAR, to a value other than the return value (H'FF).

When download is executed, particular interrupt processing, which is accompanied by the bank switch as described below, is performed as an internal microcomputer processing, so VBR need to be cleared to 0. Four NOP instructions are executed immediately after the instructions that set the SCO bit to 1.

- The user MAT space is switched to the on-chip program storage area.
- After the selection condition of the download program and the address set in FTDAR are checked, the transfer processing is executed starting from the on-chip RAM address specified by FTDAR.
- The SCO bits in FPCS, FECS, and FCCS are cleared to 0.
- The return value is set to the DPFR parameter.
- After the on-chip program storage area is returned to the user MAT space, execution returns to the user procedure program.

After download is completed and the user procedure program is running, the VBR setting can be changed.

The notes on download are as follows.



In the download processing, the values of the general registers of the CPU are retained.

During the download processing, the interrupt processing cannot be executed. However, the NMI, UBC, and H-UDI interrupt requests are retained, so that on returning to the user procedure program, the interrupt processing starts. For details on the relationship between download and interrupts, see section 23.8.2, Interrupts during Programming/Erasing.

Since a stack area of maximum 128 bytes is used, an area of at least 128 bytes must be saved before setting the SCO bit to 1.

If an access by the DMAC or AUD occurs during download, operation cannot be guaranteed. Therefore, access by the DMAC or AUD must not be executed.

(2.4) FKEY is cleared to H'00 for protection.

(2.5) The value of the DPFR parameter must be checked to confirm the download result.

A recommended procedure for confirming the download result is shown below.

- Check the value of the DPFR parameter (one byte of start address of the download destination specified by FTDAR). If the value is H'00, download has been performed normally. If the value is not H'00, the source that caused download to fail can be investigated by the description below.
- If the value of the DPFR parameter is the same as before downloading (e.g. H'FF), the address setting of the download destination in FTDAR may be abnormal. In this case, confirm the setting of the TDER bit (bit 7) in FTDAR.
- If the value of the DPFR parameter is different from before downloading, check the SS bit (bit 2) and the FK bit (bit 1) in the DPFR parameter to ensure that the download program selection and FKEY register setting were normal, respectively.

(2.6) The operating frequency is set to the FPEFEQ parameter and the user branch destination is set to the FUBRA parameter for initialization.

- The current frequency of the CPU clock is set to the FPEFEQ parameter (general register R4). For the settable range of the FPEFEQ parameter, see section 27.3.2, Clock Timing.

For the settable range of the FPEFEQ parameter, see section 27.3.2, Clock Timing.

When the frequency is set out of this range, an error is returned to the FPFR parameter of the initialization program and initialization is not performed. For details on the frequency setting, see the description in 23.4.3 (2.1) Flash programming/erasing frequency parameter (FPEFEQ).

- The start address in the user branch destination is set to the FUBRA parameter (general register R5).

When the user branch processing is not required, 0 must be set to FUBRA.

When the user branch is executed, the branch destination is executed in flash memory other than the one that is to be programmed. The area of the on-chip program that is downloaded cannot be set.

The program processing must be returned from the user branch processing by the RTS instruction.

See the description in 23.4.3 (2.2) Flash user branch address setting parameter (FUBRA).

## (2.7) Initialization

When a programming program is downloaded, the initialization program is also downloaded to on-chip RAM. There is an entry point of the initialization program in the area from (download start address set by FTDAR) + 32 bytes. The subroutine is called and initialization is executed by using the following steps.

```
MOV.L  #DLTOP+32,R1      ; Set entry address to R1
JSR    @R1              ; Call initialization routine
NOP
```

- The general registers other than R0 are saved in the initialization program.
- R0 is a return value of the FPFR parameter.
- Since the stack area is used in the initialization program, a stack area of maximum 128 bytes must be reserved in RAM.
- Interrupts can be accepted during the execution of the initialization program. However, the program storage area and stack area in on-chip RAM and register values must not be destroyed.

(2.8) The return value of the initialization program, FPFR (general register R0) is judged.

(2.9) FKEY must be set to H'5A and the user MAT must be prepared for programming.

(2.10) The parameter which is required for programming is set.

The start address of the programming destination of the user MAT (FMPAR) is set to general register R5. The start address of the program data storage area (FMPDR) is set to general register R4.

- FMPAR setting

FMPAR specifies the programming destination start address. When an address other than one in the user MAT area is specified, even if the programming program is executed, programming is not executed and an error is returned to the return value parameter FPFR. Since the unit is 128 bytes, the lower eight bits (MOA7 to MOA0) must be in the 128-byte boundary of H'00 or H'80.

- FMPDR setting

If the storage destination of the program data is flash memory, even when the program execution routine is executed, programming is not executed and an error is returned to the FPFR parameter. In this case, the program data must be transferred to on-chip RAM and then programming must be executed.

#### (2.11) Programming

There is an entry point of the programming program in the area from (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and programming is executed by using the following steps.

```
MOV.L #DLTOP+16,R1      ; Set entry address to R1
JSR   @R1               ; Call programming routine
NOP
```

- The general registers other than R0 are saved in the programming program.
- R0 is a return value of the FPFR parameter.
- Since the stack area is used in the programming program, a stack area of maximum 128 bytes must be reserved in RAM.

(2.12) The return value in the programming program, FPFR (general register R0) is judged.

(2.13) Determine whether programming of the necessary data has finished.

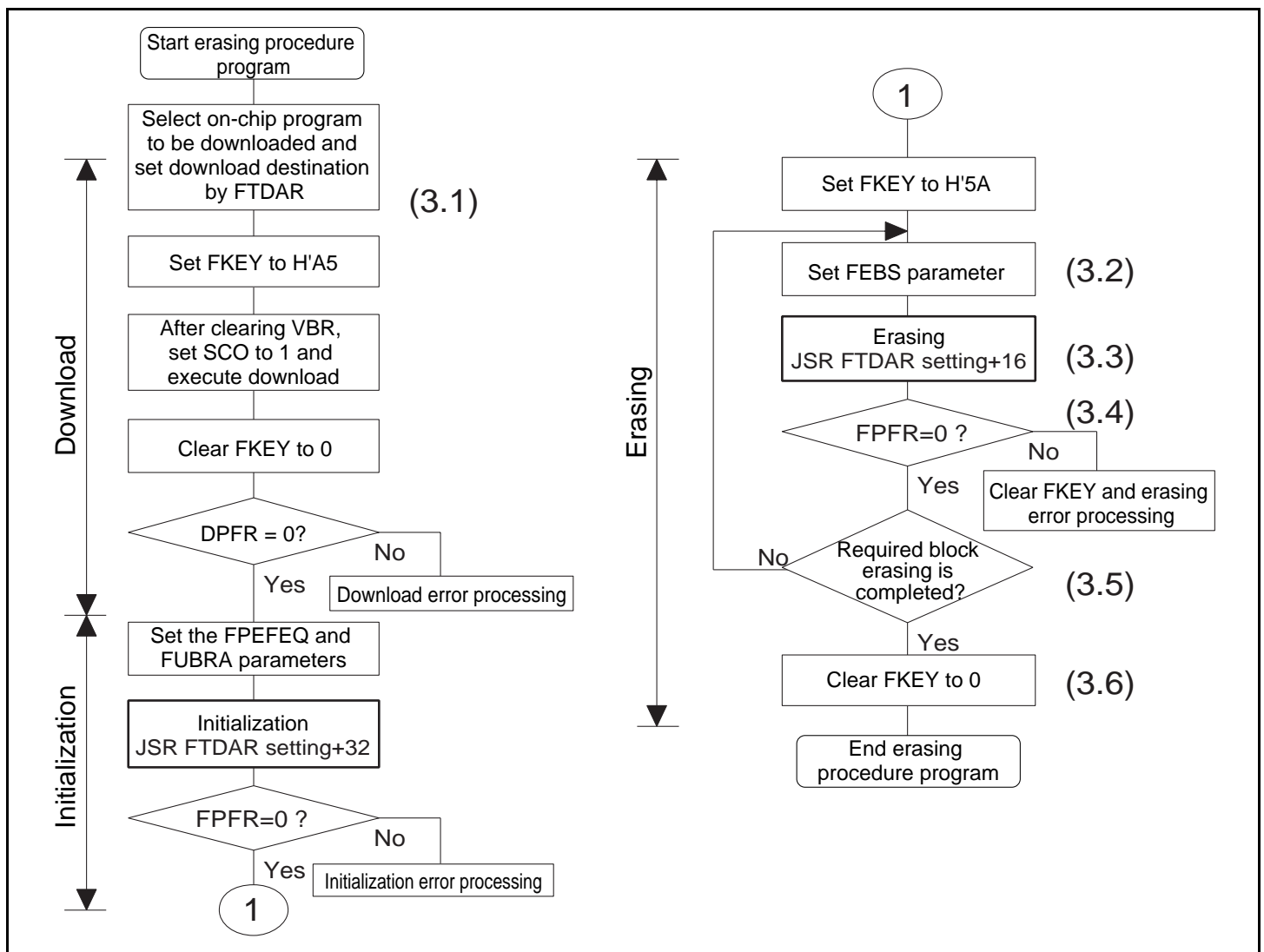
If more than 128 bytes of data are to be programmed, specify FMPAR and FMPDR in 128-byte units, and repeat steps (2.10) to (2.13). Increment the programming destination address by 128 bytes and update the programming data pointer correctly. If an address which has already been programmed is written to again, not only will a programming error occur, but also flash memory will be damaged.

(2.14) After programming finishes, clear FKEY and specify software protection.

If this LSI is restarted by a power-on reset immediately after user MAT programming has finished, secure a reset period (period of RES = 0) that is at least as long as the normal 100  $\mu$ s.

### (3) Erasing Procedure in User Program Mode

The procedures for download, initialization, and erasing are shown in figure 23.12.



**Figure 23.12 Erasing Procedure**

The details of the erasing procedure are described below. The procedure program must be executed in an area other than the user MAT to be erased.

Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 23.10.3, Storable Area for Procedure Program and Programming Data.

For the downloaded on-chip program area, see the RAM map for programming/erasing in figure 23.10.

A single divided block is erased by one erasing processing. For block divisions, see figure 23.4. To erase two or more blocks, update the erase block number and perform the erasing processing for each block.

### (3.1) Select the on-chip program to be downloaded

Set the EPVB bit in FECS to 1.

Several programming/erasing programs cannot be selected at one time. If several programs are set, download is not performed and a download error is returned to the source select error detect (SS) bit in the DPCR parameter.

Specify the start address of the download destination by FTDAR.

The procedures to be carried out after setting FKEY, e.g. download and initialization, are the same as those in the programming procedure. For details, see the description in 23.5.2 (2) Programming Procedure in User Program Mode.

### (3.2) Set the FEBS parameter necessary for erasure

Set the erase block number of the user MAT in the flash erase block select parameter (FEBS: general register R4). If a value other than an erase block number of the user MAT is set, no block is erased even though the erasing program is executed, and an error is returned to the return value parameter FPFR.

### (3.3) Erasure

Similar to as in programming, there is an entry point of the erasing program in the area from (download start address set by FTDAR) + 16 bytes of on-chip RAM. The subroutine is called and erasing is executed by using the following steps.

```
MOV.L #DLTOP+16,R1      ; Set entry address to R1
JSR   @R1               ; Call erasing routine
NOP
```

— The general registers other than R0L are saved in the erasing program.

— R0 is a return value of the FPFR parameter.

— Since the stack area is used in the erasing program, a stack area of maximum 128 bytes must be reserved in RAM.

### (3.4) The return value in the erasing program, FPFR (general register R0) is judged.

### (3.5) Determine whether erasure of the necessary blocks has finished.

If more than one block is to be erased, update the FEBS parameter and repeat steps (3.2) to (3.5). Blocks that have already been erased can be erased again.

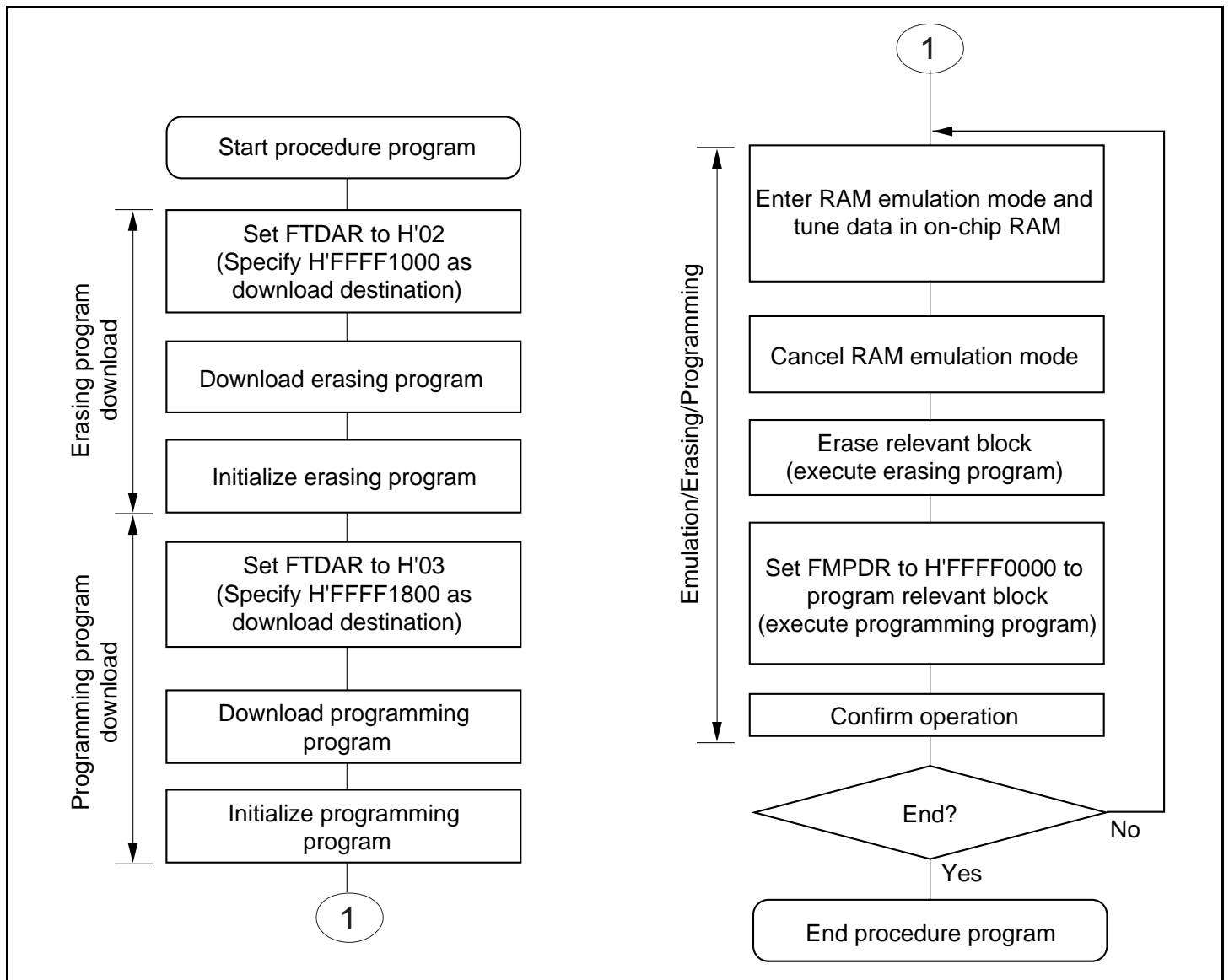
### (3.6) After erasure finishes, clear FKEY and specify software protection.

If this LSI is restarted by a power-on reset immediately after user MAT programming has finished, secure a reset period (period of  $\overline{RES} = 0$ ) that is at least as long as the normal 100  $\mu$ s.

## (4) Erasing and Programming Procedure in User Program Mode

By changing the on-chip RAM address of the download destination in FTDAR, the erasing program and programming program can be downloaded to separate on-chip RAM areas.

Figure 23.13 shows an example of repetitively executing RAM emulation, erasing, and programming.



**Figure 23.13 Sample Procedure of Repeating RAM Emulation, Erasing, and Programming (Overview)**

In the above example, the erasing program and programming program are downloaded to areas excluding the 4 kbytes (H'FFFF0000 to H'FFFF0FFF) from the start of on-chip ROM.

Download and initialization are performed only once at the beginning.

In this kind of operation, note the following:

- Be careful not to damage on-chip RAM with overlapped settings.

In addition to the RAM emulation area, erasing program area, and programming program area, areas for the user procedure programs, work area, and stack area are reserved in on-chip RAM. Do not make settings that will overwrite data in these areas.

- Be sure to initialize both the erasing program and programming program.

Initialization by setting the FPEFEQ and FUBRA parameters must be performed for both the erasing program and the programming program. Initialization must be executed for both entry addresses: (download start address for erasing program) + 32 bytes (H'FFFF1020 in this example) and (download start address for programming program) + 32 bytes (H'FFFF1820 in this example).

### 23.5.3 User Boot Mode

This LSI has user boot mode which is initiated with different mode pin settings than those in user program mode or boot mode. User boot mode is a user-arbitrary boot mode, unlike boot mode that uses the on-chip SCI.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing of the user boot MAT is only enabled in boot mode or programmer mode.

#### (1) User Boot Mode Initiation

For the mode pin settings to start up user boot mode, see table 23.1, Relationship between FWE and MD pins and Operating Modes.

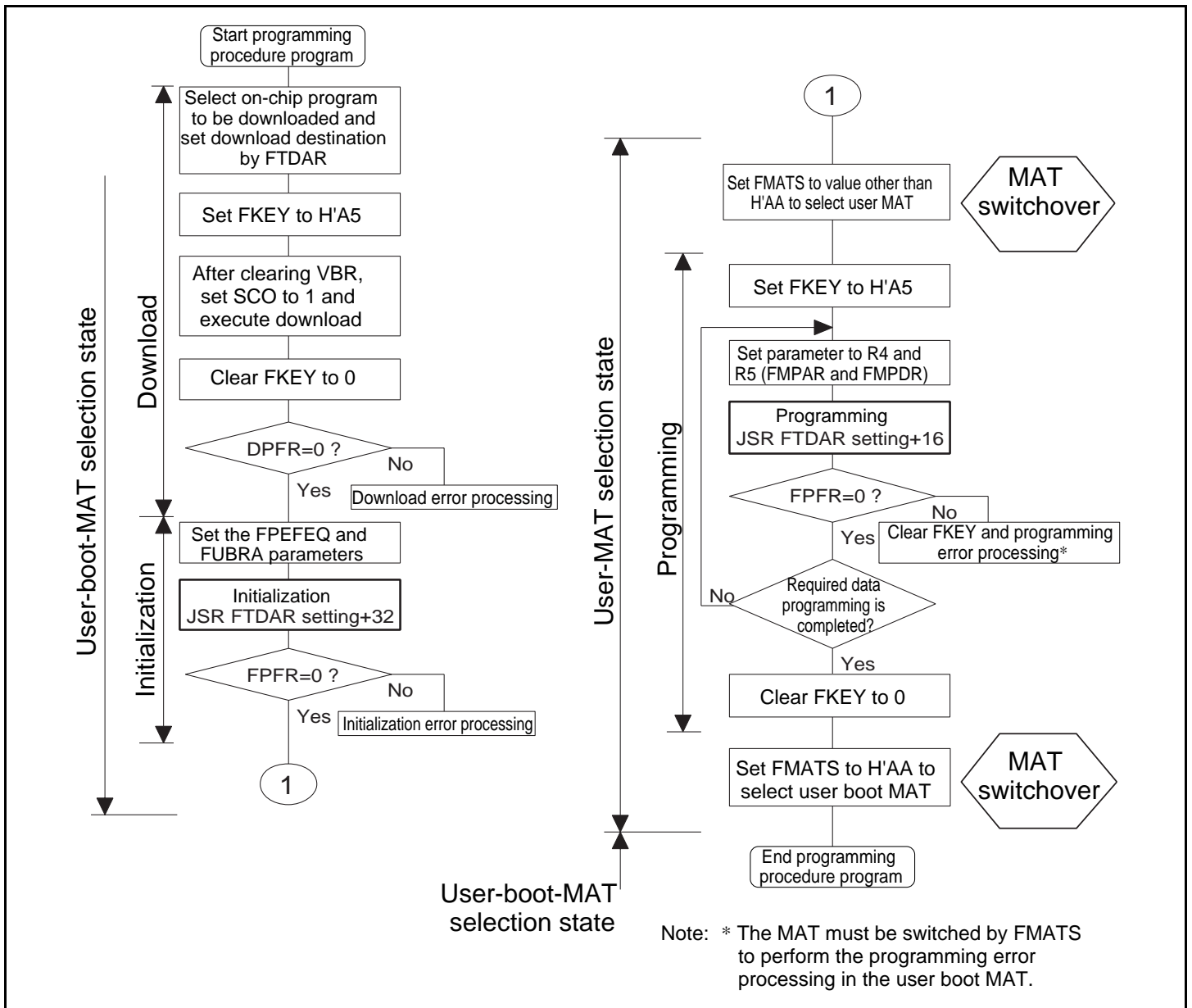
When the reset start is executed in user boot mode, the check routine for flash-memory related registers runs. The RAM area about 1.2 kbytes from H'FFFF0800 and 4 bytes from H'FFFFBFFC (a stack area) is used by the routine. While the check routine is running, NMI and all other interrupts cannot be accepted. Neither can the AUD be used in this period. This period is 100  $\mu$ s while operating at an internal frequency of 40 MHz.

Next, processing starts from the execution start address of the reset vector in the user boot MAT. At this point, H'AA is set to the flash MAT select register (FMATS) because the execution MAT is the user boot MAT.

#### (2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processings made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after programming completes.

Figure 23.14 shows the procedure for programming the user MAT in user boot mode.



**Figure 23.14 Procedure for Programming User MAT in User Boot Mode**

The difference between the programming procedures in user program mode and user boot mode is whether the MAT is switched or not as shown in figure 23.14.

In user boot mode, the user boot MAT can be seen in the flash memory space with the user MAT hidden in the background. The user MAT and user boot MAT are switched only while the user MAT is being programmed. Because the user boot MAT is hidden while the user MAT is being programmed, the procedure program must be located in an area other than flash memory. After programming finishes, switch the MATs again to return to the first state.

MAT switchover is enabled by writing a specific value to FMATS. However note that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completely finished, and if an interrupt occurs, from which MAT the interrupt vector is read from is undetermined. Perform MAT switching in accordance with the description in section 23.8.1, Switching between User MAT and User Boot MAT.

Except for MAT switching, the programming procedure is the same as that in user program mode.

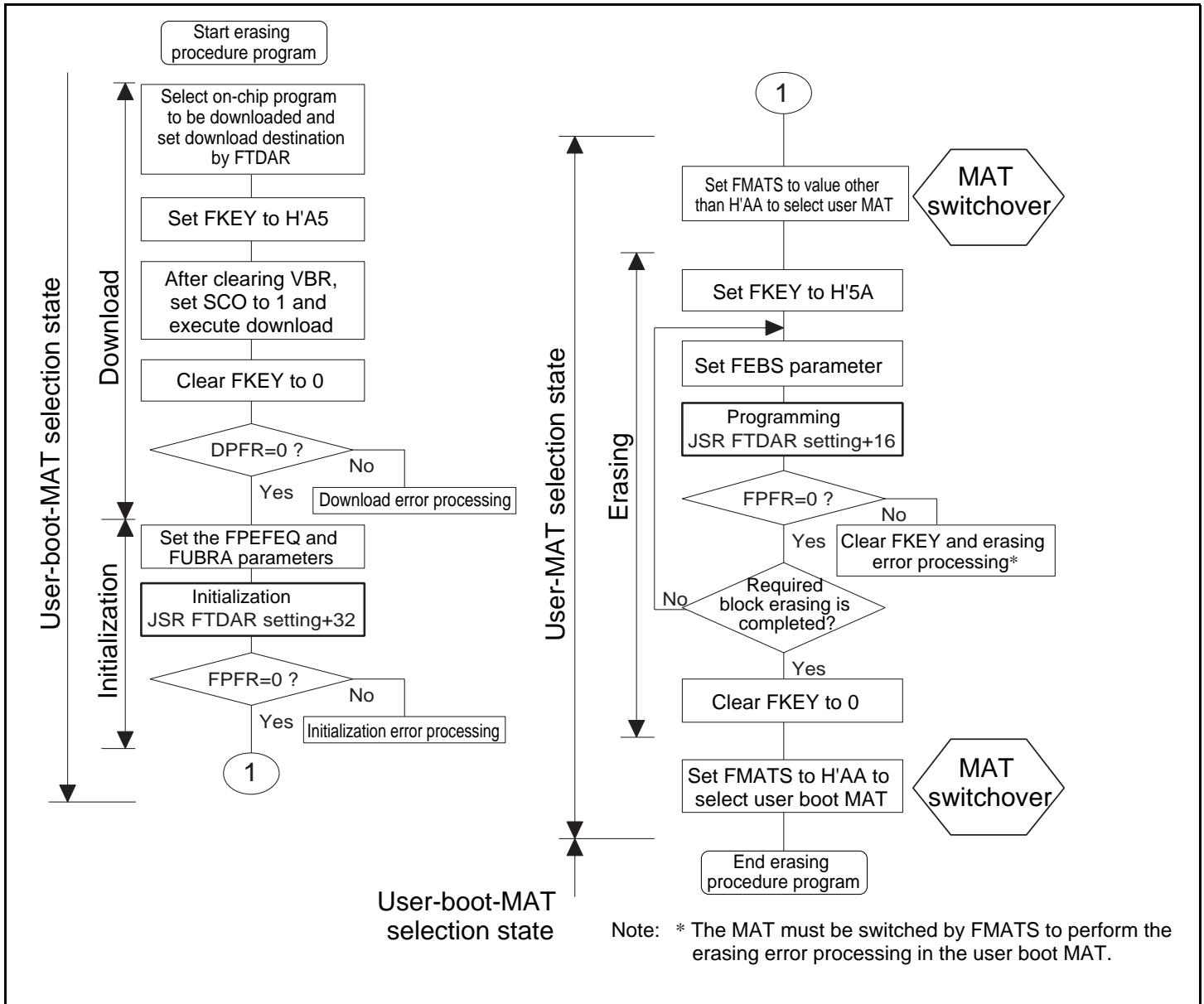


The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 23.10.3, Storable Area for Procedure Program and Programming Data.

### (3) User MAT Erasing in User Boot Mode

For erasing the user MAT in user boot mode, additional processings made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after erasing completes.

Figure 23.15 shows the procedure for erasing the user MAT in user boot mode.



**Figure 22.15 Procedure for Erasing User MAT in User Boot Mode**

The difference between the erasing procedures in user program mode and user boot mode depends on whether the MAT is switched or not as shown in figure 23.15.

MAT switching is enabled by writing a specific value to FMATS. However note that while the MATs are being switched, the LSI is in an unstable state, e.g. access to a MAT is not allowed until MAT switching is completed finished, and if an interrupt occurs, from which MAT the interrupt

vector is read from is undetermined. Perform MAT switching in accordance with the description in section 23.8.1, Switching between User MAT and User Boot MAT.

Except for MAT switching, the erasing procedure is the same as that in user program mode.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 23.10.3, Storable Area for Procedure Program and Programming Data.

## **23.6 Protection**

There are three kinds of flash memory program/erase protection: hardware, software, and error protection.

### **23.6.1 Hardware Protection**

Programming and erasing of flash memory is forcibly disabled or suspended by hardware protection. In this state, the downloading of an on-chip program and initialization of the flash memory are possible. However, an activated program for programming or erasure cannot program or erase locations in a user MAT, and the error in programming/erasing is reported in the FPFR parameter.

**Table 23.9 Hardware Protection**

Item	Description	Function to be Protected	
		Download	Programming/ Erasure
FWE-pin protection	The input of a low-level signal on the FWE pin clears the FWE bit of FCCS and the LSI enters a programming/erasing-protected state.	—	○
Reset/standby protection	<ul style="list-style-type: none"> <li>• A power-on reset (including a power-on reset by the WDT) and entry to standby mode initializes the programming/erasing interface registers and the LSI enters a programming/erasing-protected state.</li> <li>• Resetting by means of the <math>\overline{\text{RES}}</math> pin after power is initially supplied will not make the LSI enter the reset state unless the <math>\overline{\text{RES}}</math> pin is held low until oscillation has stabilized. In the case of a reset during operation, hold the <math>\overline{\text{RES}}</math> pin low for the <math>\overline{\text{RES}}</math> pulse width that is specified in the section on AC characteristics. If the LSI is reset during programming or erasure, data in the flash memory is not guaranteed. In this case, execute erasure and then execute programming again.</li> </ul>	○	○

### 23.6.2 Software Protection

Software protection is set up in any of three ways: by disabling the downloading of on-chip programs for programming and erasing, by means of a key code, and by the RAM emulation register (RAMER).

**Table 23.10 Software Protection**

Item	Description	Function to be Protected	
		Download	Programming/ Erasure
Protection by the SCO bit	Clearing the SCO bit in FCCS disables downloading of the programming/erasing program, thus making the LSI enter a programming/erasing-protected state.	O	O
Protection by FKEY	Downloading and programming/erasing are disabled unless the required key code is written in FKEY. Different key codes are used for downloading and for programming/erasing.	O	O
Emulation protection	Setting the RAMS bit in RAMER to 1 makes the LSI enter a programming/erasing-protected state.	O	O

### 23.6.3 Error Protection

Error protection is a mechanism for aborting programming or erasure when an error occurs, in the form of the microcomputer getting out of control during programming/erasing of the flash memory or operations that are not in accordance with the established procedures for programming/erasing. Aborting programming or erasure in such cases prevents damage to the flash memory due to excessive programming or erasing.

If the microcomputer malfunctions during programming/erasing of the flash memory, the FLER bit in FCCS is set to 1 and the LSI enters the error protection state, thus aborting programming or erasure.

The FLER bit is set to 1 in the following conditions:

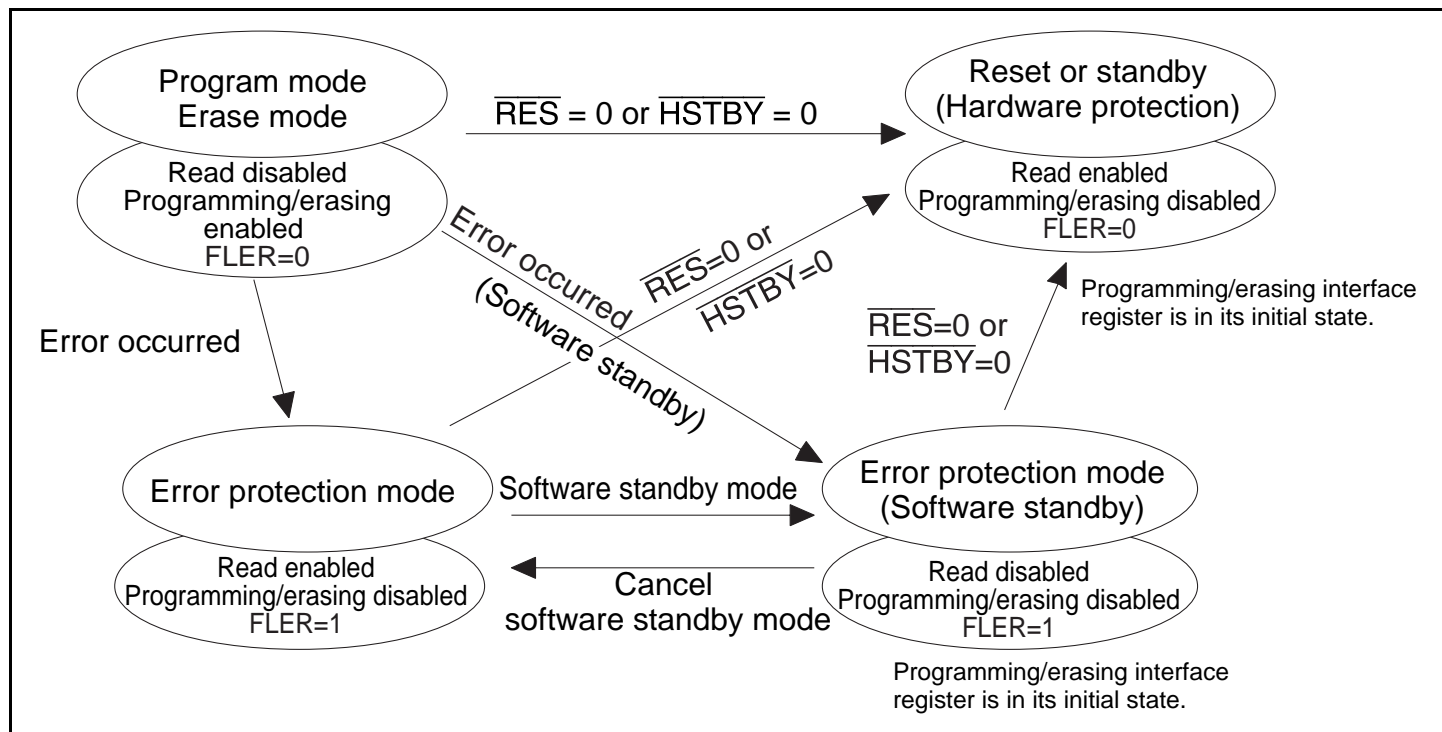
- When the relevant bank area of flash memory is read during programming/erasing (including a vector read or an instruction fetch)
- When a SLEEP instruction (including software standby mode) is executed during programming/erasing

Error protection is cancelled (FLER bit is cleared) only by a power-on reset or in hardware-standby mode.

Note that the reset signal should only be released after providing a reset input over a period longer than the normal 100  $\mu$ s. Since high voltages are applied during programming/erasing of the flash memory, some voltage may still remain even after the error protection state has been entered. For

this reason, it is necessary to reduce the risk of damage to the flash memory by extending the reset period so that the charge is released.

The state-transition diagram in figure 23.16 shows transitions to and from the error protection state.

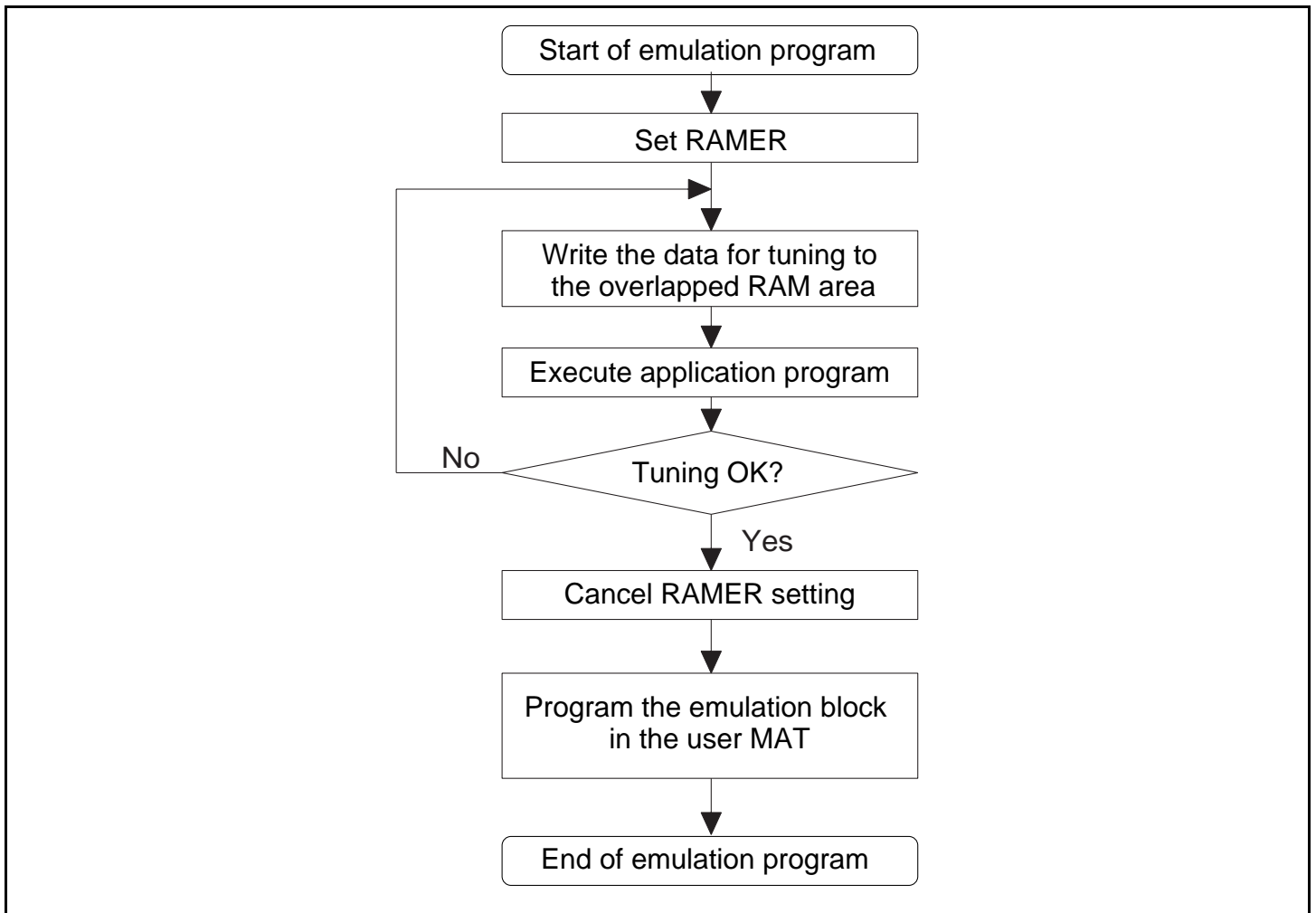


**Figure 23.16 Transitions to and from Error Protection State**

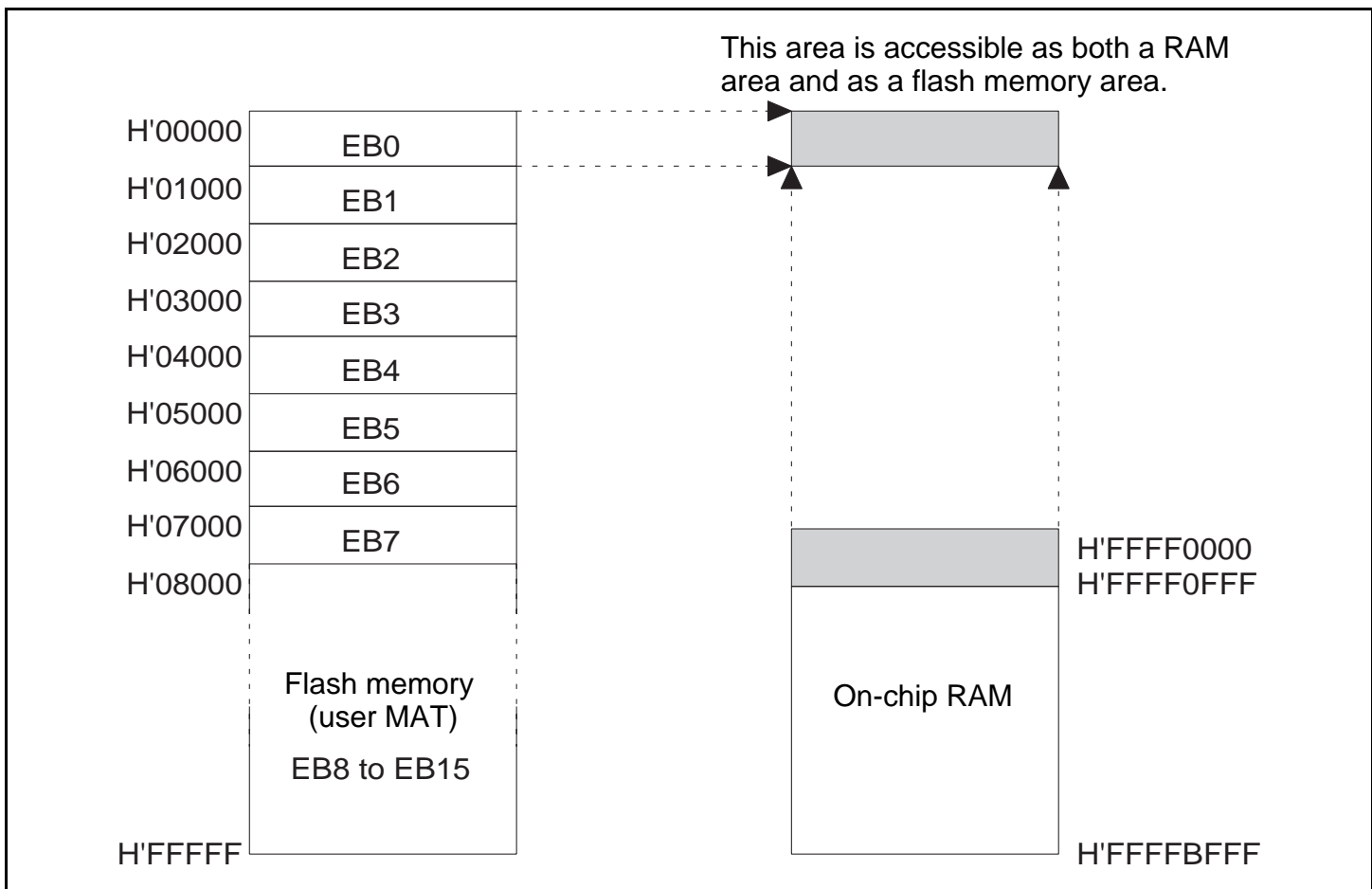
## 23.7 Flash Memory Emulation in RAM

To provide real-time emulation in RAM of data that is to be written to the flash memory, a part of the RAM can be overlaid on an area of flash memory (user MAT) that has been specified by the RAM emulation register (RAMER). After the RAMER setting is made, the RAM is accessible in both the user MAT area and as the RAM area that has been overlaid on the user MAT area. Such emulation is possible in user mode and user program mode.

Figure 23.17 shows an example of the emulation of realtime programming of the user MAT area.



**Figure 23.17 Emulation of Flash Memory in RAM**



**Figure 23.18 Example of Overlapped RAM Operation**

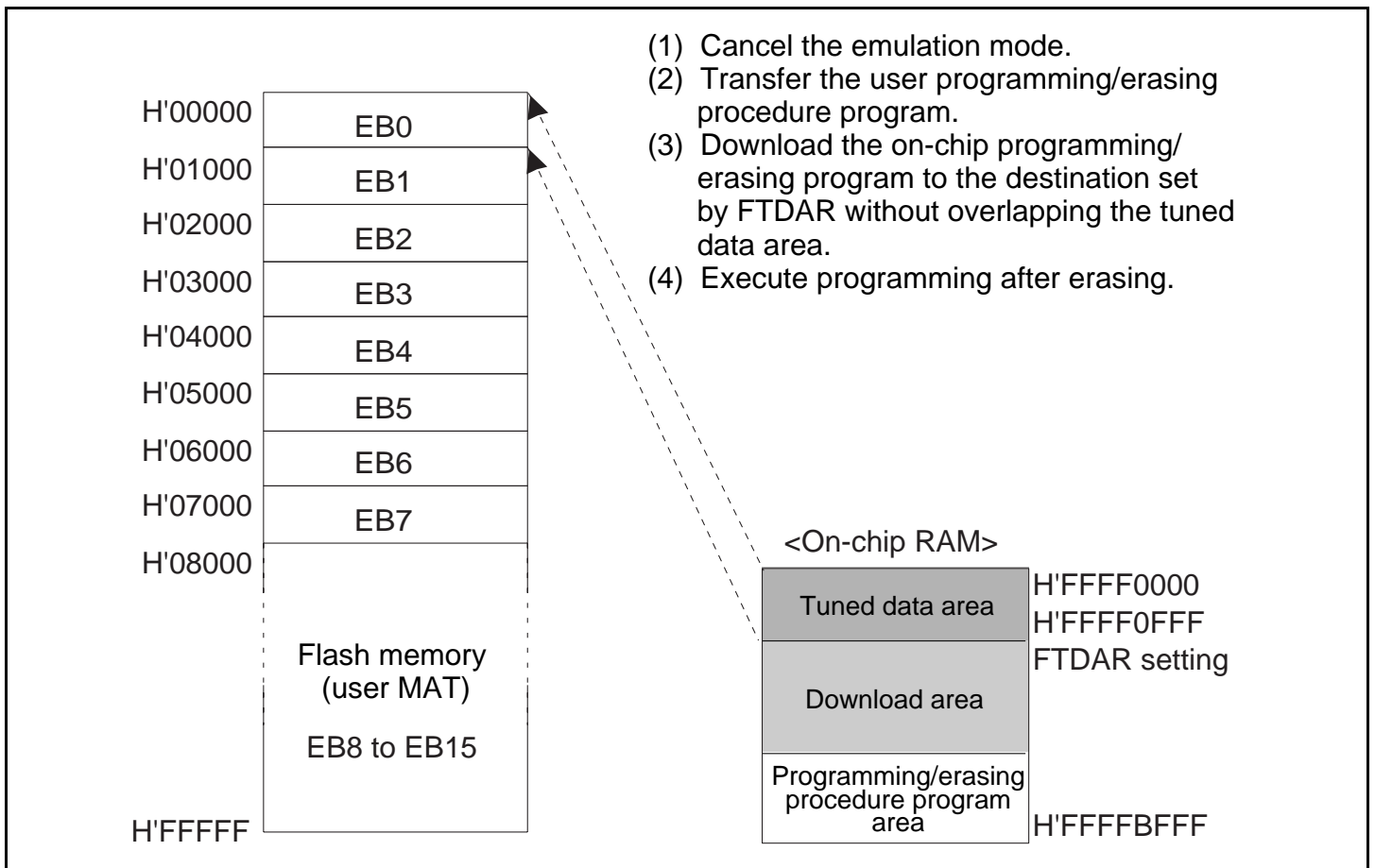
Figure 23.18 shows an example of an overlap on block area EB0 of the flash memory.

Emulation is possible for a single area selected from among the eight areas, from EB0 to EB7, of the user MAT. The area is selected by the setting of the RAM2 to RAM0 bits in RAMER.

- (1) To overlap a part of the RAM on area EB0, to allow realtime programming of the data for this area, set the RAMS bit in RAMER to 1, and each of the RAM2 to RAM0 bits to 0.
- (2) Realtime programming is carried out using the overlaid area of RAM.

In programming or erasing the user MAT, it is necessary to run a program that implements a series of procedural steps, including the downloading of an on-chip program. In this process, set the download area with FTDAR so that the overlaid RAM area and the area where the on-chip program is to be downloaded do not overlap. The initial setting (H'00) of FTDAR causes the tuned data area to overlap with the download area. When using the initial setting of FTDAR, the data that is to be programmed must be saved beforehand in an area that is not used by the system.

Figure 23.19 shows an example of programming data that has been emulated to the EB0 area in the user MAT.



**Figure 23.19 Programming of Tuned Data**

1. After the data to be programmed has fixed values, clear the RAMS bit to 0 to cancel the overlap of RAM. Emulation mode is canceled and emulation protection is also cleared.
2. Transfer the user programming/erasing procedure program to RAM.
3. Run the programming/erasing procedure program in RAM and download the on-chip programming/erasing program.  
Specify the download start address with FTDAR so that the tuned data area does not overlap with the download area.
4. When the EB0 area of the user MAT has not been erased, erasing must be performed before programming. Set the parameters FMPAR and FMPDR so that the tuned data is designated, and execute programming.

Note: Setting the RAMS bit to 1 puts all the blocks in flash memory in the programming/erasing-protected state regardless of the values of the RAM2 to RAM0 bits (emulation protection). Clear the RAMS bit to 0 before actual programming or erasure. Though RAM emulation can also be carried out with the user boot MAT selected, the user boot MAT can be erased or programmed only in boot mode or programmer mode.



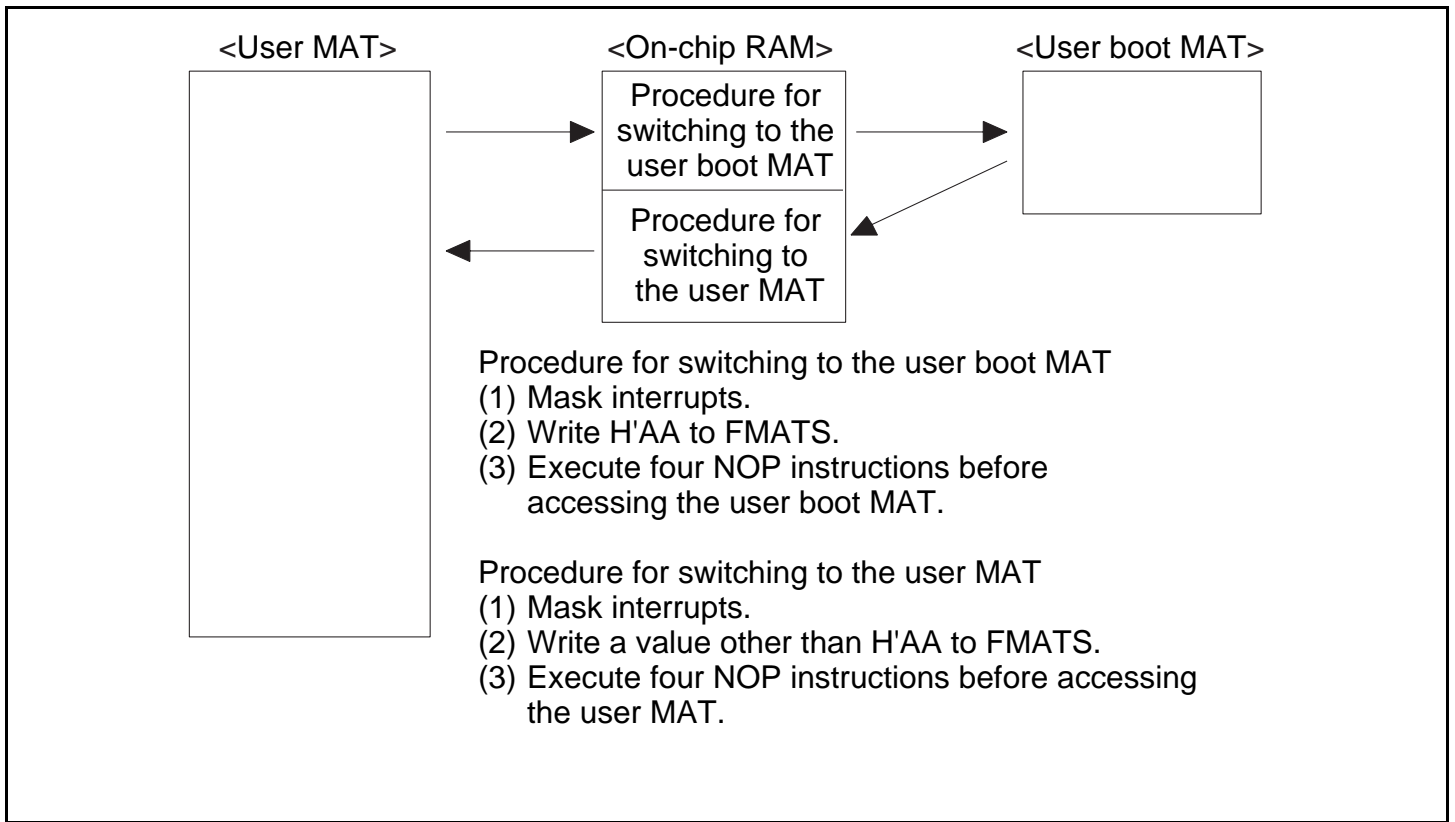
## 23.8 Usage Notes

### 23.8.1 Switching between User MAT and User Boot MAT

It is possible to switch between the user MAT and user boot MAT. However, the following procedure is required because these MATs are allocated to address 0.

(Switching to the user boot MAT disables programming and erasing. Programming of the user boot MAT must take place in boot mode or programmer mode.)

- (1) MAT switching by FMATS should always be executed from the on-chip RAM. The SH microcomputer prefetches execution instructions. Therefore, a switchover during program execution in the user MAT causes an instruction code in the user MAT to be prefetched or an instruction in the newly selected user boot MAT to be prefetched, thus resulting in unstable operation.
- (2) To ensure that the MAT that has been switched to is accessible, execute four NOP instructions in on-chip RAM immediately after writing to FMATS of on-chip RAM (this prevents access to the flash memory during MAT switching).
- (3) If an interrupt occurs during switching, there is no guarantee of which memory MAT is being accessed.  
Always mask the maskable interrupts before switching MATs. In addition, configuring the system so that NMI interrupts do not occur during MAT switching is recommended.
- (4) After the MATs have been switched, take care because the interrupt vector table will also have been switched.  
If the same interrupt processings are to be executed before and after MAT switching or interrupt requests cannot be disabled, transfer the interrupt processing routine to on-chip RAM, and use the VBR setting to place the interrupt vector table in on chip RAM. In this case, make sure the VBR setting change does not conflict with the interrupt occurrence.
- (5) Memory sizes of the user MAT and user boot MAT are different. When accessing the user boot MAT, do not access addresses exceeding the 8-kbyte memory space. If access goes beyond the 8-kbyte space, the values read are undefined.



**Figure 23.20 Switching between User MAT and User Boot MAT**

### 23.8.2 Interrupts during Programming/Erasing

#### (1) Download of On-Chip Program

##### (1.1) VBR setting change

Before downloading the on-chip program, VBR must be set to H'00000000 (initial value). If VBR is set to a value other than the initial value, the interrupt vector table is placed in the user MAT (FMATS is not H'AA) or the user boot MAT (FMATS is H'AA) on initialization of VBR.

When VBR setting change conflicts with interrupt occurrence, whether the vector table before or after VBR is changed is referenced may cause an error.

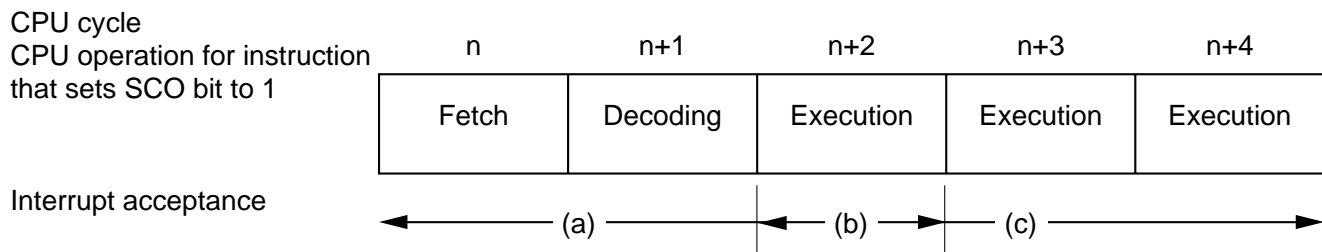
Therefore, for cases where VBR setting change may conflict with interrupt occurrence, prepare a vector table to be referenced when VBR is H'00000000 at the start of the user MAT or user boot MAT.

##### (1.2) SCO download request and interrupt request

Download of the on-chip programming/erasing program that is initiated by setting the SCO bit in FCCS to 1 generates a particular interrupt processing accompanied by MAT switchover. Operation when the SCO download request and interrupt request conflicts is described below.

##### 1. Contention between SCO download request and interrupt request

Figure 23.21 shows the timing of contention between execution of the instruction that sets the SCO bit in FCCS to 1 and interrupt acceptance.



- (a) When the interrupt is accepted at or before the (n + 1) cycle  
After the interrupt processing completes, the SCO bit is set to 1 and download is executed.
- (b) When the interrupt is accepted at the (n + 2) cycle  
The interrupt conflicts with the SCO download request. For details on operation in this case, see 2. Operation when contention occurs.
- (c) When the interrupt is accepted at or after the (N + 3) cycle  
The SCO download request occurs prior to the interrupt request, and download is executed. During download, no other interrupt processing can be handled. If an interrupt is still being requested after download completes, the interrupt processing starts. For details on interrupt requests during download, see 3. Interrupt requests generated during download.

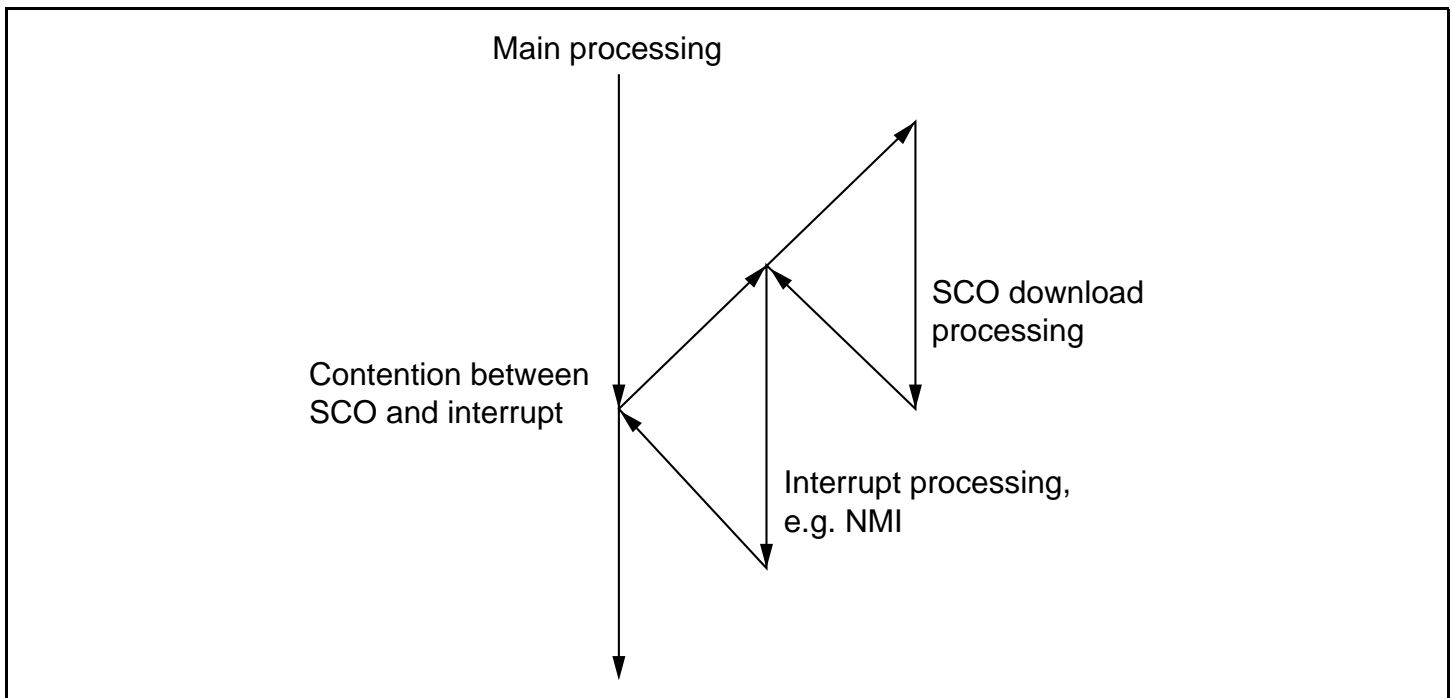
**Figure 23.21 Timing of Contention between SCO Download Request and Interrupt Request**

2. Operation when contention occurs

Operation differs according to the type of interrupt with which the SCO download request has conflicted.

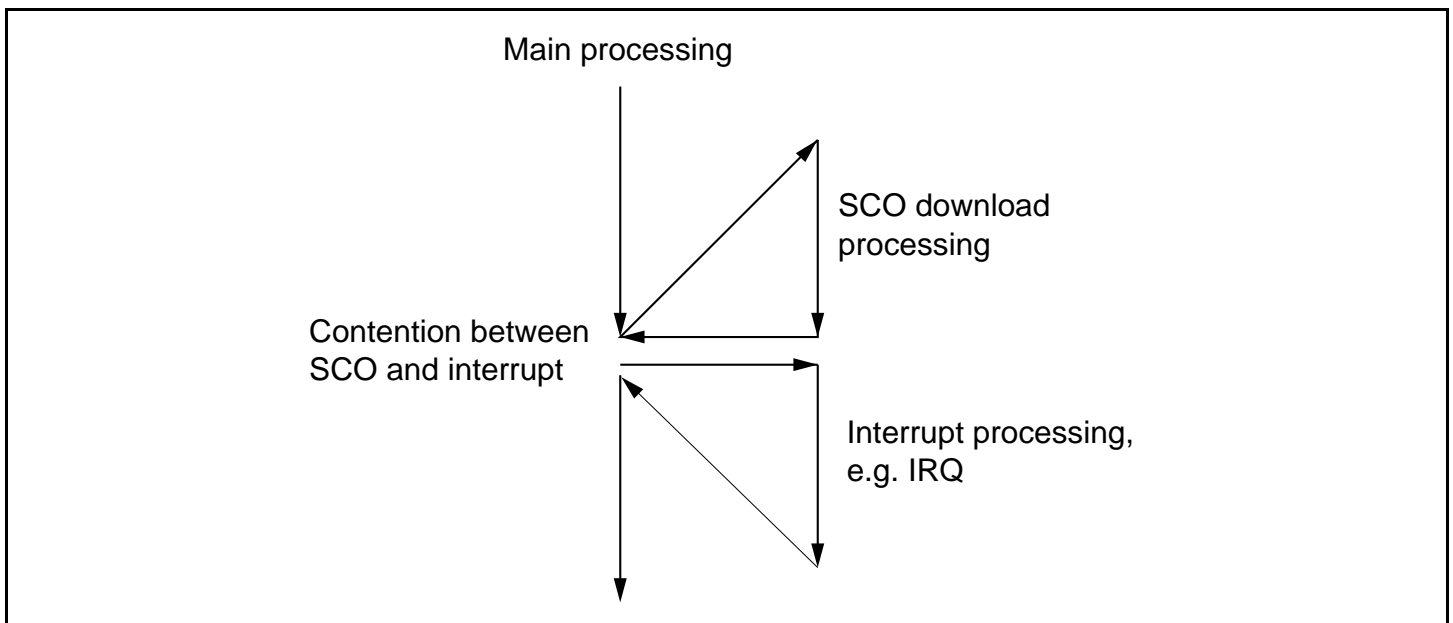
— NMI, UBC, and H-UDI interrupt requests

Operation for when these interrupts conflict with the SCO download request is described below.



**Figure 23.22 Contention between Interrupts (e.g. NMI)**

- The NMI, UBC, or H-UDI interrupt processing is started. Processing proceeds up to the point where SR and PC are saved, the vector is fetched, and the start instruction of the interrupt processing routine is fetched.
  - At this point, the SCO download request with a higher priority occurs. The SCO download processing is started.
  - After the download processing has ended, the interrupt processing routine (e.g. NMI) that was in the middle of execution resumes from the point of fetching the start instruction of the interrupt processing routine.
  - The interrupt processing routine is ended, and execution returns to the main processing.
- IRQ and on-chip peripheral module interrupt requests  
Operation for when these interrupts conflict with the SCO download request is described below.



**Figure 23.23 Contention between Interrupts (e.g. IRQ)**

- An IRQ interrupt or interrupt from an on-chip peripheral module is replaced with the SCO download request and download is executed.
  - If the IRQ or on-chip peripheral module interrupt is still being requested when the download processing has ended, the interrupt processing is executed. If these interrupt requests have been canceled, execution returns to the main processing.
  - An interrupt request is canceled when the IRQ signal, for which low-level detection is set, has been driven high before download ends. Also refer to the description below (3. Interrupt requests generated during download).
3. Interrupt requests generated during download  
Even though an interrupt is requested during SCO download, the interrupt processing is not executed until download ends. Note that interrupt requests are basically retained, so that on completion of download, the interrupt processing starts. When more than one type of interrupts are requested, their priorities are judged by the interrupt controller (INTC), and execution starts from the interrupt processing with higher priority.

— NMI, UBC, and H-UDI interrupt requests

When these interrupt requests occur during SCO download, their interrupt sources are retained.

— IRQ interrupt request

Falling-edge detection or low-level detection can be specified for an IRQ interrupt.

- Falling-edge detection is selected: When the falling-edge of IRQ is detected during SCO download, the interrupt source is retained.
- Low-level detection is selected: When the low-level of IRQ is detected during SCO download, if the IRQ remains low when download ends, the interrupt processing starts. If the IRQ is high when download ends, the interrupt source will be canceled.

— On-chip peripheral module interrupt request

An interrupt from an on-chip peripheral module is requested by input of the specified level. Since the interrupt signal continues to be output unless the interrupt flag is cleared, the interrupt source is retained.

## (2) Interrupts during programming/erasing

Though an interrupt processing can be executed at realtime during programming/erasing of the downloaded on-chip program, the following limitations and notes are applied.

1. When flash memory is being programmed or erased, both the user MAT and user boot MAT cannot be accessed. Prepare the interrupt vector table and interrupt processing routine in on-chip RAM or external memory. Make sure the flash memory being programmed or erased is not accessed by the interrupt processing routine. If flash memory is read, the read values are not guaranteed. If the relevant bank in flash memory that is being programmed or erased is accessed, the error protection state is entered, and programming or erasing is aborted. If a bank other than the relevant bank is accessed, the error protection state is not entered but the read values are not guaranteed.
2. Do not rewrite the program data specified by the FMPDR parameter. If new program data is to be provided by the interrupt processing, temporarily save the new program data in another area. After confirming the completion of programming, save the new program data in the area specified by FMPDR or change the setting in FMPDR to indicate the other area in which the new program data was temporarily saved.
3. Make sure the interrupt processing routine does not rewrite the contents of the flash-memory related registers or data in the downloaded on-chip program area. During the interrupt processing, do not simultaneously perform RAM emulation, download of the on-chip program by an SCO request, or programming/erasing.
4. At the beginning of the interrupt processing routine, save the CPU register contents. Before returning from the interrupt processing, write the saved contents in the CPU registers again.
5. When a transition is made to sleep mode or software standby mode in the interrupt processing routine, the error protection state is entered and programming/erasing is aborted.

If a transition is made to the reset state, the reset signal should only be released after

providing a reset input over a period longer than the normal 100  $\mu$ s to reduce the damage to flash memory.

### 23.8.3 Other Notes

#### 1. Download time of on-chip program

The programming program that includes the initialization routine and the erasing program that includes the initialization routine are each 2 kbytes or less. Accordingly, when the CPU clock frequency is 40 MHz, the download for each program takes approximately 75  $\mu$ s at maximum.

#### 2. User branch processing intervals

The intervals for executing the user branch processing differs in programming and erasing. The processing phase also differs. Table 23.11 lists the maximum and minimum intervals for initiating the user branch processing when the CPU clock frequency is 40 MHz.

**Table 23.11 Initiation Intervals of User Branch Processing**

Processing Name	Maximum Interval	Minimum Interval
Programming	Approximately 1 ms	Approximately 17 $\mu$ s
Erasing	Approximately 5 ms	Approximately 17 $\mu$ s

However, when operation is done with CPU clock of 40 MHz, maximum and minimum values of the time until initial user branch processing are as shown in table 23.12.

**Table 23.12 Initial User Branch Processing Time**

Processing Name	Max.	Min.
Programming	Approximately 113 $\mu$ s	Approximately 113 $\mu$ s
Erasing	Approximately 85 $\mu$ s	Approximately 45 $\mu$ s

#### 3. Write to flash-memory related registers by AUD or DMAC

While an instruction in on-chip RAM is being executed, the AUD or DMAC can write to the SCO bit in FCCS that is used for a download request or FMATS that is used for MAT switching. Make sure that these registers are not accidentally written to, otherwise an on-chip program may be downloaded and damage RAM or a MAT switchover may occur and the CPU get out of control.

#### 4. 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; they are not executed and the interrupt sources are not retained.

- Boot mode
- Programmer mode

- Checking the flash-memory related registers immediately after user boot mode is initiated (Approximately 100  $\mu$ s if operation is done at an internal frequency of 40 MHz after the reset signal is released)
5. Compatibility with programming/erasing program of conventional F-ZTAT SH microcomputer
- A programming/erasing program for flash memory used in the conventional F-ZTAT SH microcomputer which does not support download of the on-chip program by a SCO transfer request cannot run in this LSI.
- Be sure to download the on-chip program to execute programming/erasing of flash memory in this LSI.
6. Monitoring runaway by WDT
- Unlike the conventional F-ZTAT SH microcomputer, no countermeasures are available for a runaway by WDT during programming/erasing by the downloaded on-chip program.
- Prepare countermeasures (e.g. use of the user branch routine and periodic timer interrupts) for WDT while taking the programming/erasing time into consideration as required.

## 23.9 Programmer Mode

Along with its on-board programming mode, this LSI also has programmer mode as another mode for writing and erasing of programs and data. Programmer mode supports memory-read mode, auto-program mode, auto-erase mode, and status-read mode. Programming/erasing is possible on the user MAT and user boot MAT.

A status-polling system is adopted for operation in auto-program mode, auto-erase mode, and status-read mode. In status-read mode, details of the system's internal state are output after execution of automatic programming or automatic erasure.

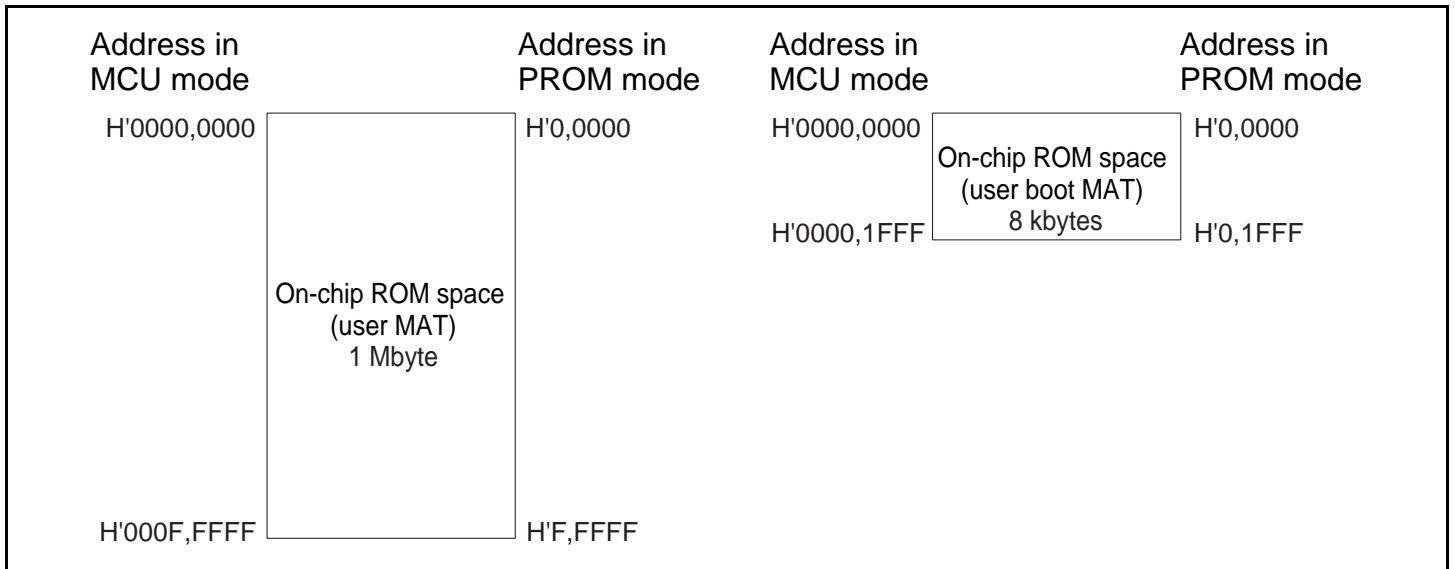
In programmer mode, set the mode pins as shown in table 23.13, and provide a 6-MHz input-clock signal.

**Table 23.13 Programmer Mode Pin Settings**

Pin Name	Settings
Mode pins: MD2, MD1, and MD0	0, 1, 1
FWE	High-level input (automatic programming and automatic erasure)
$\overline{\text{RES}}$	Power-on reset circuit
EXTAL, XTAL, PLLV <sub>CC</sub> , PLLV <sub>SS</sub> , PLLCAP	Oscillation circuit and PLL circuit
V <sub>CL</sub>	Internal stepdown stabilization capacitor

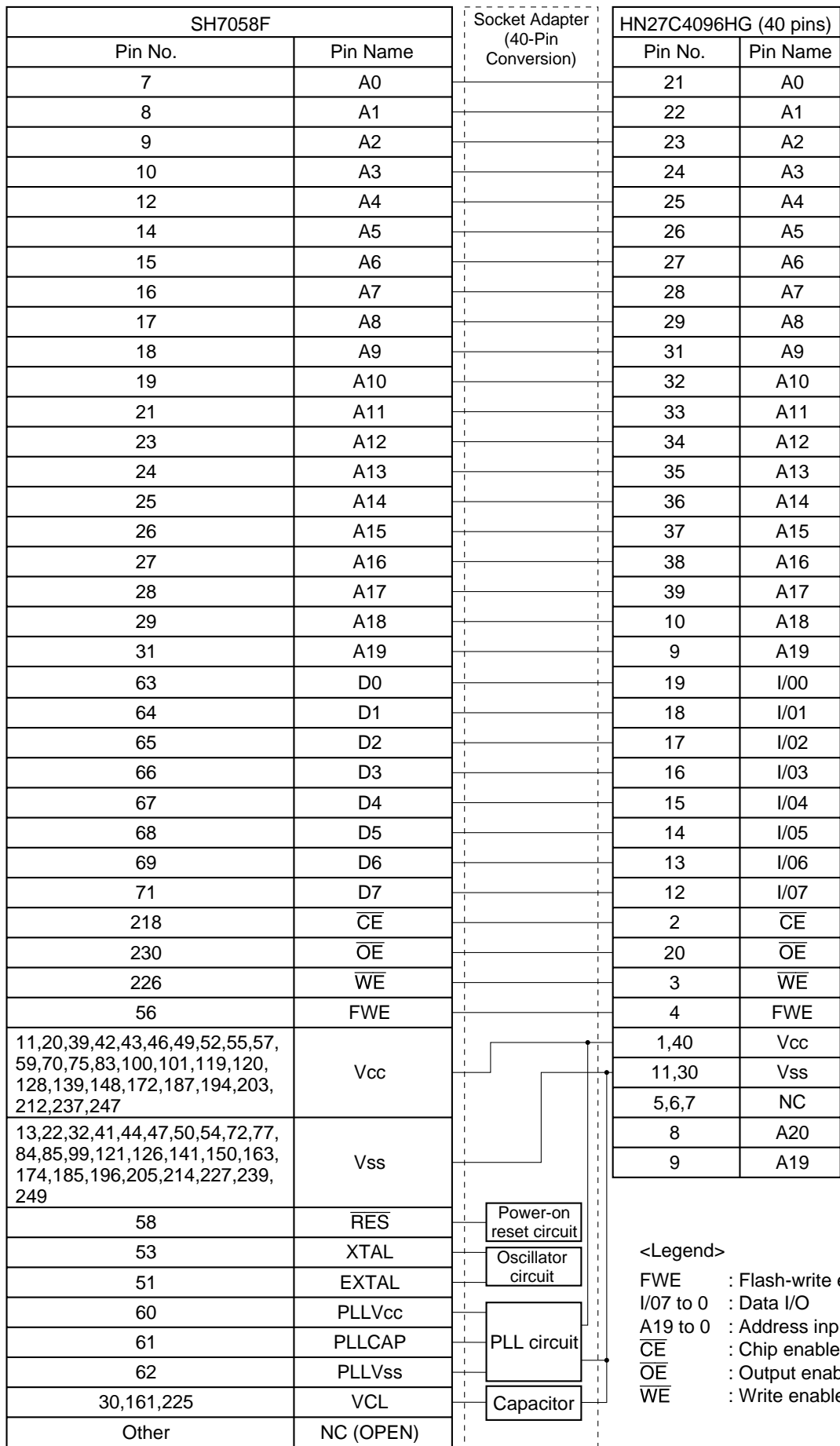
### 23.9.1 Pin Arrangement of Socket Adapter

Attach the socket adapter to the LSI in the way shown in figure 23.25. This allows conversion to 40 pins. Figure 23.24 shows the memory mapping of on-chip ROM, and figure 23.25 shows the arrangement of the socket adapter's pins.



**Figure 23.24 Mapping of On-Chip Flash Memory**





**Figure 23.25 Pin Arrangement of Socket Adapter**

## 23.9.2 Programmer Mode Operation

Table 23.14 shows the settings for the operating modes of programmer mode, and table 23.15 lists the commands used in programmer mode. The following sections provide detailed information on each mode.

- Memory-read mode  
Supports reading from the user MAT or user boot MAT in bytes.
- Auto-program mode  
Supports the simultaneous programming of the user MAT and user boot MAT in 128-byte units. Status polling is used to confirm the end of automatic programming.
- Auto-erase mode  
Supports only automatic erasure of the entire user MAT or user boot MAT. Status polling is used to confirm the end of automatic erasure.
- Status-read mode  
Status polling is used with automatic programming and automatic erasure. Normal completion can be detected by reading the signal on the I/O6 pin. In status-read mode, error information is output when an error has occurred.

**Table 23.14 Settings for Each Operating Mode of Programmer Mode**

Mode	Pin Name					
	FWE	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	I/O7 to I/O0	A19 to A0
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-Z	X
Command write	H or L	L	H	L	Data input	Ain*
Chip disable	H or L	H	X	X	Hi-Z	X

Notes: 1. The chip-disable mode is not a standby state; internally, it is an operational state.  
2. To write commands when making a transition to auto-program or auto-erase mode, input a high-level signal on the  $\overline{FWE}$  pin.

\* Ain indicates that there is also an address input in auto-program mode.

**Table 23.15 Commands in Programmer Mode**

Command	Number of Cycles	Memory MAT to be Accessed	1st Cycle			2nd Cycle		
			Mode	Address	Data	Mode	Address	Data
Memory-read mode	1+n	User MAT	Write	X	H'00	Read	RA	Dout
		User boot MAT	Write	X	H'05			
Auto-program mode	129	User MAT	Write	X	H'40	Write	WA	Din
		User boot MAT	Write	X	H'45			
Auto-erase mode	2	User MAT	Write	X	H'20	Write	X	H'20
		User boot MAT	Write	X	H'25			H'25
Status-read mode	2	Common to both MATs	Write	X	H'71	Write	X	H'71

- Notes
1. In auto-program mode, 129 cycles are required in command writing because of the simultaneous 128-byte write.
  2. In memory read mode, the number of cycles varies with the number of address writing cycles (n).
  3. In an automatic erasure command, input the same command code for the 1st and 2nd cycles (for erasing of the user boot MAT, input H'25 for the 1st and 2nd cycles).

### 23.9.3 Memory-Read Mode

- (1) On completion of automatic programming, automatic erasure, or status read, the LSI enters a command input wait state. So, to read the contents of memory after these operations, issue the command to transit to memory-read mode before reading from the memory.
- (2) In memory-read mode, the writing of commands is possible in the same way as in command input wait state.
- (3) After entering memory-read mode, continuous reading is possible.
- (4) After power has first been supplied, the LSI enters memory-read mode.

For the AC characteristics in memory read mode, see section 23.10.2, AC Characteristics and Timing in Programmer Mode.

### 23.9.4 Auto-Program Mode

- (1) In auto-program mode, programming is in 128-byte units. That is, 128 bytes of data are transferred in succession.
- (2) Even in the programming of less than 128 bytes, 128 bytes of data must be transferred. H'FF should be written to those addresses that are unnecessarily written to.
- (3) Set the lower seven bits of the address to be transferred to low level. Inputting an invalid address will result in a programming error, although processing will proceed to the memory-programming operation.
- (4) The memory address is transferred in the 2nd cycle. Do not transfer addresses in the 3rd or later cycles.
- (5) Do not issue commands while programming is in progress.
- (6) When programming, execute automatic programming once for each 128-byte block of addresses. Programming the block at an address where programming has already been performed is not possible.
- (7) To confirm the end of automatic programming, check the signal on the I/O6 pin. Confirmation in status-read mode is also possible (status polling of the I/O7 pin is used to check the end status of automatic programming).
- (8) Status-polling information on the I/O6 and I/O7 pins is retained until the next command is written. As long as no command is written, the information is made readable by enabling  $\overline{\text{CE}}$  and  $\overline{\text{OE}}$ .

For the AC characteristics in auto-program mode, see section 23.10.2, AC Characteristics and Timing in Programmer Mode.

### 23.9.5 Auto-Erase Mode

- (1) Auto-erase mode only supports erasing of the entire memory.
- (2) Do not perform command writing while auto erasing is in progress.
- (3) To confirm the end of automatic erasure, check the signal on the I/O6 pin. Confirmation in the status-read mode is also possible (status polling of the I/O7 pin is used to check the end status of automatic erasure).
- (4) Status polling information on the I/O6 and I/O7 pins is retained until the next command writing. As long as no command is written, the information is made readable by enabling  $\overline{\text{CE}}$  and  $\overline{\text{OE}}$ .

For the AC characteristics in auto-erase mode, see section 23.10.2, AC Characteristics and Timing in Programmer Mode.

### 23.9.6 Status-Read Mode

- (1) Status-read mode is used to determine the type of an abnormal termination. Use this mode when automatic programming or automatic erasure ends abnormally.
- (2) The return code is retained until writing of a command that selects a mode other than status-read mode.

Table 23.16 lists the return codes of status-read mode.

For the AC characteristics in status-read mode, see section 23.10.2, AC Characteristics and Timing in Programmer Mode.

**Table 23.16 Return Codes of Status-Read Mode**

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end indicator	Command error	Programming error	Erasure error	—	—	Programming or erase count exceeded	Invalid address error
Initial value	0	0	0	0	0	0	0	0
Indication	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise : 0	Programming error: 1 Otherwise: 0	Erasure error:1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Invalid address error: 1 Otherwise: 0

Note: I/O2 and I/O3 are undefined pins.

### 23.9.7 Status Polling

- (1) The I/O7 status-polling output is a flag that indicates the operating status in auto-program or auto-erase mode.
- (2) The I/O6 status-polling output is a flag that indicates normal/abnormal end of auto-program or auto-erase mode.

**Table 23.17 Truth Table of Status-Polling Output**

Pin Name	In Progress	Abnormal End	—	Normal End
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O0 to I/O5	0	0	0	0

## 23.9.8 Time Taken in Transition to Programmer Mode

Until oscillation has stabilized and while programmer mode is being set up, the LSI is unable to accept commands. After the programmer-mode setup time has elapsed, the LSI enters memory-read mode. For details, see section 23.10.2, AC Characteristics and Timing in Programmer Mode.

## 23.9.9 Notes on Programming in Programmer Mode

- (1) When programming addresses which have previously been programmed, apply auto-erasing before auto-programming.
- (2) When using programmer mode to program a chip that has been programmed/erased in an on-board programming mode, auto-erasing before auto-programming is recommended.
- (3) Do not take the chip out of the PROM programmer or reset the chip during programming or erasure. Flash memory is susceptible to permanent damage since a high voltage is being applied during the programming/erasing. When the reset signal is accidentally input to the chip, the period in the reset state until the reset signal is released should be longer than the normal 100  $\mu$ s.

- Notes:
1. The flash memory is initially in the erased state when the device is shipped by Renesas Technology. For other chips for which the history of erasure is unknown, auto-erasing as a check and supplement for the initialization (erase) level is recommended.
  2. Automatic programming to a single address block can only be performed once. Additional programming to an address block that has already been programmed is not allowed.

## 23.10 Further Information

### 23.10.1 Serial Communication Interface Specification for Boot Mode

Initiating boot mode enables the boot program to communicate with the host by using the on-chip SCI. The serial communication interface specifications are shown below.

- Status

The boot program has three states.

- (1) Bit-rate-adjustment state

In this state, the boot program adjusts the bit rate to communicate with the host. Initiating boot mode enables starting of the boot program and entry to the bit-rate-adjustment state. The program receives the command from the host to adjust the bit rate. After adjusting the bit rate, the program enters the inquiry/selection state.

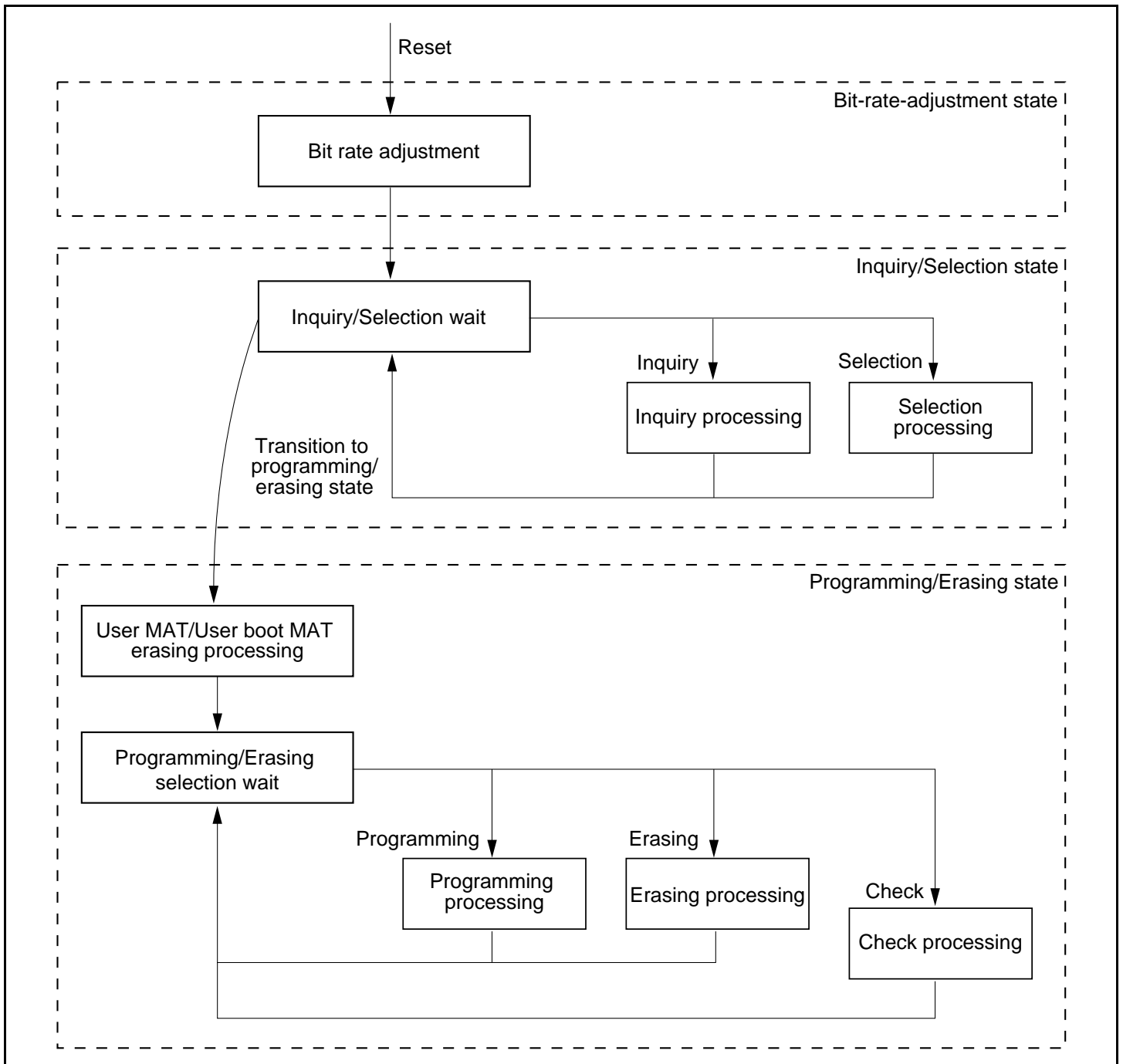
## (2) Inquiry/Selection state

In this state, the boot program responds to inquiry commands from the host. The device name, clock mode, and bit rate are selected. After selection of these settings, the program is made to enter the programming/erasing state by the command for a transition to the programming/erasing state. The boot program transfers the erasure program to RAM and erases the user MATs and user boot MATs before the transition.

## (3) Programming/erasing state

Programming and erasure by the boot program take place in this state. The boot program is made to transfer the programming/erasing program to RAM by commands from the host. Sum checks and blank checks are executed by sending these commands from the host.

These boot program states are shown in figure 23.26.

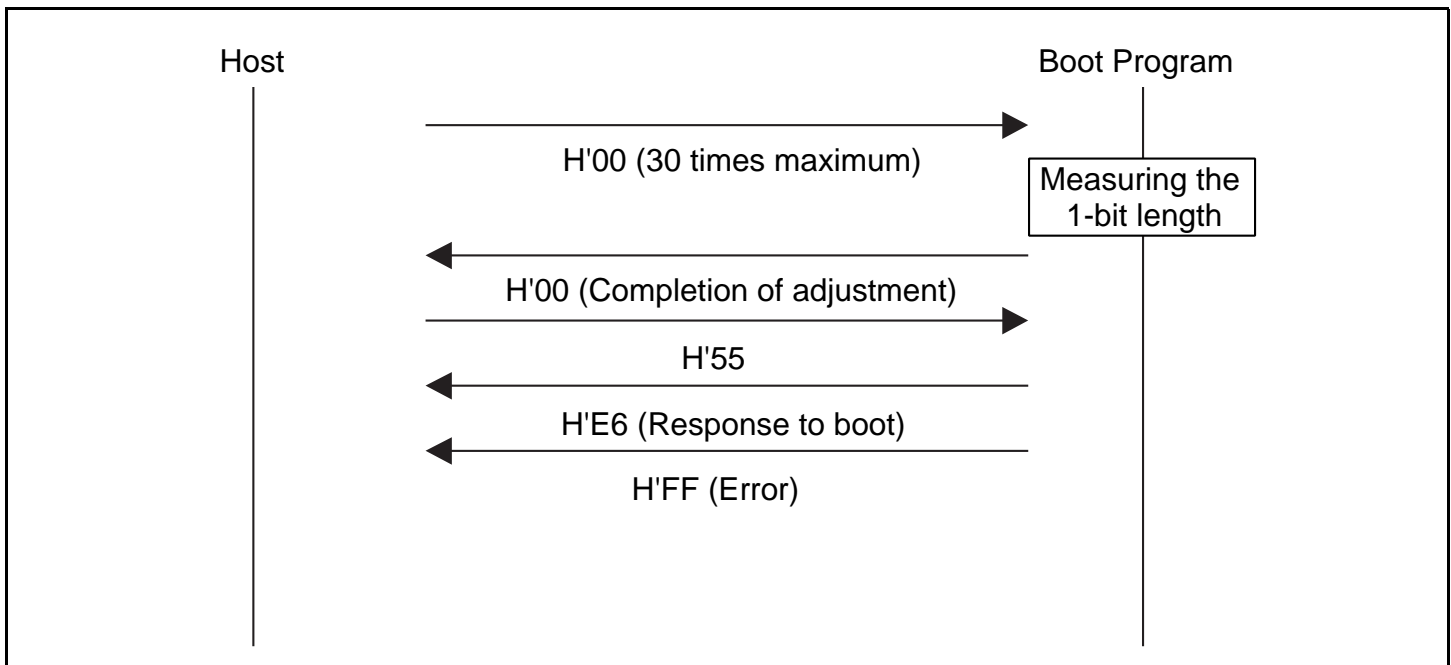


**Figure 23.26 Boot Program Processing Flow**

- Bit-rate-adjustment state

The bit rate is calculated by measuring the period of transfer of a low-level byte (H'00) from the host. The bit rate can be changed by the command for a new bit rate selection. After the bit rate has been adjusted, the boot program enters the inquiry/selection state. The bit-rate-adjustment sequence is shown in figure 23.27.





**Figure 23.27 Bit-Rate-Adjustment Sequence**

- Communications protocol

After adjustment of the bit rate, the protocol for serial communications between the host and the boot program is as shown below.

(1) One-byte commands and one-byte responses

These commands and responses are comprised of a single byte. These consists of the inquiries and ACK for successful completion.

(2) n-byte commands or n-byte responses

These commands and responses are comprised of n bytes of data. These are selections and responses to inquiries.

The amount of programming data is not included under this heading because it is determined in another command.

(3) Error response

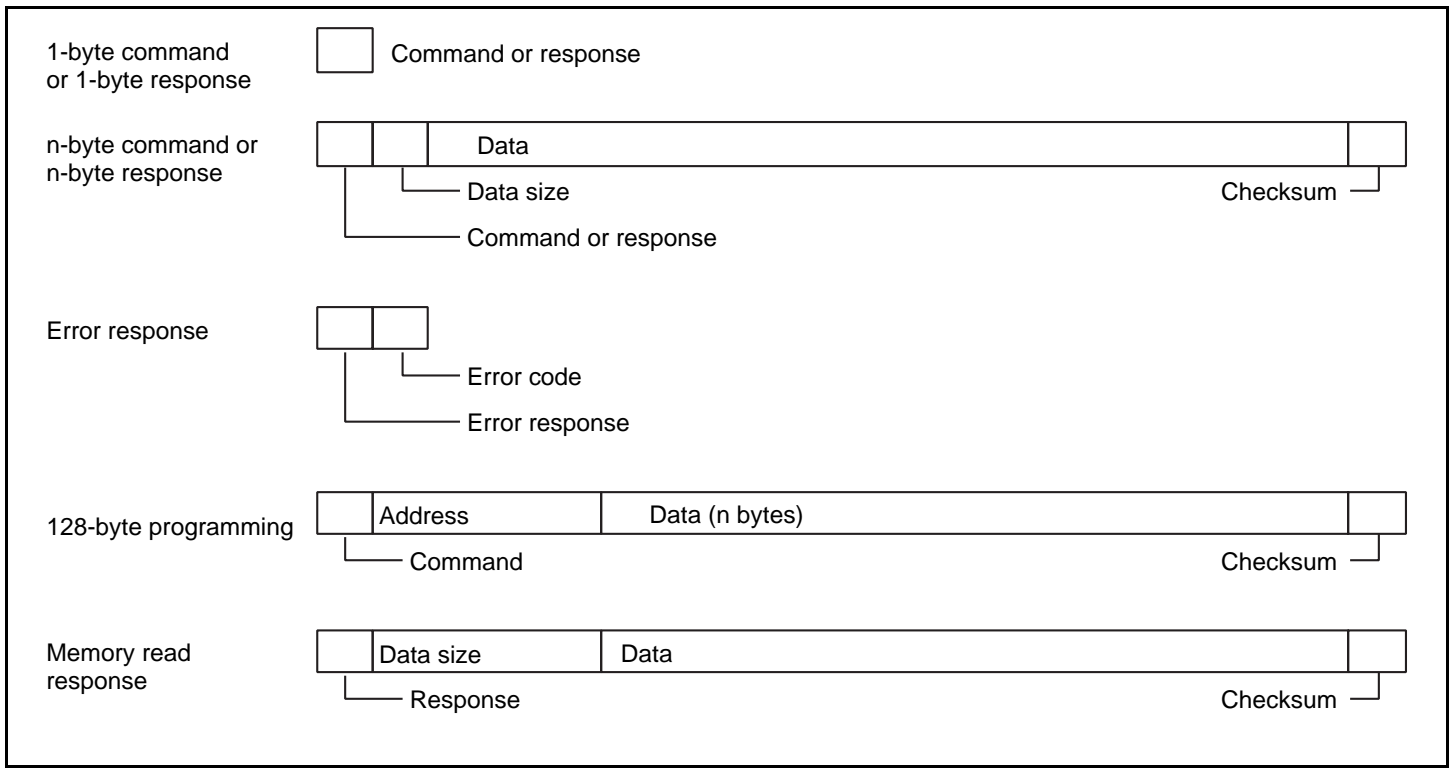
The error response is a response to inquiries. It consists of an error response and an error code and which take up two bytes.

(4) Programming of 128 bytes

The size is not specified in commands. The data size is indicated in response to the programming unit inquiry.

(5) Memory read response

This response consists of four bytes of data.



**Figure 23.28 Communications Protocol Format**

- Command (one byte): Commands including inquiries, selection, programming, erasing, and checking
- Response (one byte): Response to an inquiry
- Size (one or two bytes): The amount of data for transmission excluding the command, amount of data, and checksum
- Data (n bytes): Detailed data of a command or response
- Checksum (one byte): The checksum is calculated so that the total of all values from the command byte to the SUM byte becomes H'00.
- Error Response (one byte): Error response to a command
- Error Code (one byte): Type of the error
- Address (four bytes): Address for programming
- Data (n bytes): Data to be programmed. n is indicated in the response to the programming unit inquiry.
- Data Size (four bytes): Four-byte response to a memory read

- Inquiry/Selection State

The boot program returns information from the flash memory in response to the host's inquiry commands and sets the device code, clock mode, and bit rate in response to the host's selection command.

Table 23.18 lists the inquiry and selection commands.

**Table 23.18 Inquiry and Selection Commands**

<b>Command</b>	<b>Command Name</b>	<b>Description</b>
H'20	Supported Device Inquiry	Inquiry regarding device codes and product names of F-ZTAT
H'10	Device Selection	Selection of device code
H'21	Clock Mode Inquiry	Inquiry regarding numbers of clock modes and values of each mode
H'11	Clock Mode Selection	Indication of the selected clock mode
H'22	Multiplication Ratio Inquiry	Inquiry regarding the number of clock types, the number of multiplication/division ratios, and the multiplication/division ratios
H'23	Operating Clock Frequency Inquiry	Inquiry regarding the maximum and minimum values of the main clock and peripheral clocks
H'24	User Boot MAT Information Inquiry	Inquiry regarding the number of user boot MATs and the start and last addresses of each MAT
H'25	User MAT Information Inquiry	Inquiry regarding the a number of user MATs and the start and last addresses of each MAT
H'26	Block for Erasing Information Inquiry	Inquiry regarding the number of blocks and the start and last addresses of each block
H'27	Programming Unit Inquiry	Inquiry regarding the unit of programming data
H'3F	New Bit Rate Selection	Selection of new bit rate
H'40	Transition to Programming/Erasing State	Erasing of user MAT and user boot MAT, and entry to programming/erasing state
H'4F	Boot Program Status Inquiry	Inquiry into the operation status of the boot program

The selection commands, which are device selection (H'10), clock mode selection (H'11), and new bit rate selection (H'3F), should be sent from the host in this order. These commands are certainly required. When two or more selection commands are sent at once, the last command will be valid.

All of these commands, except for the boot program status inquiry command (H'4F), will be valid until the boot program receives the programming/erasing transition (H'40). The host can choose the needed commands out of the commands and inquiries listed above. The boot program status inquiry command (H'4F) is valid after the boot program has received the programming/erasing transition command (H'40).

## (1) Supported device inquiry

The boot program will return the device codes of supported devices in response to the supported device inquiry.

Command 

H'20
------

— Command: H'20 (one byte): Inquiry regarding supported devices

Response	H'30	Size	Number of devices	
	Number of characters	Device code		Product name
	...			
	SUM			

— Response: H'30 (one byte): Response to the supported device inquiry

— Size (one byte): Number of bytes to be transmitted, excluding the command, amount of data, and checksum, that is, the amount of data consists of the product names, the number of devices, characters, and device codes

— Number of devices (one byte): Number of device types supported by the boot program

— Number of characters (one byte): Number of characters in the device code and boot program's name

— Device code (four bytes): Supporting product (ASCII code)

— Product name (n bytes): Type name of the boot program (ASCII code)

— SUM (one byte): Checksum

The checksum is calculated so that the total number of all values from the command byte to the SUM byte becomes H'00.

## (2) Device Selection

The boot program will set the supported device to the specified device code. The program will return the selected device code in response to the inquiry after this setting has been made.

Command 

H'10	Size	Device code	SUM
------	------	-------------	-----

— Command: H'10 (one byte): Device selection

— Size (one byte): Number of characters in the device code (fixed at 2)

— Device code (four bytes): Device code returned in response to the supported device inquiry (ASCII code)

— SUM (one byte): Checksum

Response 

H'06
------

— Response: H'06, (one byte): Response to the device selection command  
ACK will be returned when the device code matches.

Error response 

H'90	ERROR
------	-------

— Error response: H'90 (one byte): Error response to the device selection command

- ERROR: (one byte): Error code
  - H'11: Sum check error
  - H'21: Device code mismatch error

### (3) Clock Mode Inquiry

The boot program will return the supported clock modes in response to the clock mode inquiry.

Command 

H'21
------

- Command: H'21 (one byte): Inquiry regarding clock mode

Response 

H'31	Size	Number of modes	Mode	SUM
------	------	-----------------	------	-----

- Response: H'31 (one byte): Response to the clock-mode inquiry
- Size (one byte): Amount of data that represents the number of modes and modes
- Number of modes (one byte): Number of supported clock modes
  - H'00 indicates no clock mode or the device allows the clock mode to be read.
- Mode (one byte): Supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (one byte): Checksum

### (4) Clock Mode Selection

The boot program will set the specified clock mode. The program will return the selected clock-mode information after this setting has been made.

The clock-mode selection command should be sent after the device selection command.

Command 

H'11	Size	Mode	SUM
------	------	------	-----

- Command: H'11 (one byte): Selection of clock mode
- Size (one byte): Number of characters that represents the mode (fixed at 1)
- Mode (one byte): Clock mode returned in reply to the supported clock mode inquiry.
- SUM (one byte): Checksum

Response 

H'06
------

- Response: H'06 (one byte): Response to the clock-mode selection command
  - ACK will be returned when the clock mode matches.

Error response 

H'91	ERROR
------	-------

- Error response: H'91 (one byte): Error response to the clock-mode selection command
- ERROR (one byte): Error code
  - H'11: Sum check error
  - H'22: Clock mode mismatch error

### (5) Multiplication Ratio Inquiry

The boot program will return the supported multiplication/division ratios.

Command 

H'22
------

- Command: H'22 (one byte): Inquiry regarding multiplication ratio

Response	H'32	Size	Number of clock types				
	Number of multiplication ratios	Multiplication ratio	...				
	...						
	SUM						

- Response: H'32 (one byte): Response to the multiplication ratio inquiry
- Size (one byte): Amount of data that represents the number of clock types, the number of multiplication ratios, and the multiplication ratios
- Number of clock types (one byte): Number of supported multiplied clock types (e.g. when there are two multiplied clock types, which are the main operating frequency and the peripheral module operating frequency, the number of types will be H'02)
- Number of multiplication ratios (one byte): Number of multiplication ratios for each operating frequency (e.g. the number of multiplication ratios to which the main operating frequency can be set and the peripheral module operating frequency can be set)
- Multiplication ratio (one byte)  
 Multiplication ratio: Value of the multiplication ratio (e.g. when the clock-frequency multiplier is four, the value of multiplication ratio will be H'04)  
 Division ratio: Value of the division ratio, inverted to be a negative number (e.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = -2)  
 The number of multiplication ratios returned is the same as the number of multiplication ratios and as many groups of data are returned as there are types.
- SUM (one byte): Checksum

#### (6) Operating Clock Frequency Inquiry

The boot program will return the number of operating clock frequencies, and the maximum and minimum values.

Command H'23

- Command: H'23, (one byte): Inquiry regarding operating clock frequencies

Response	H'33	Size	Number of operating clock frequencies
	Minimum value of operating clock frequency	Maximum value of operating clock frequency	
	...		
	SUM		

- Response: H'33 (one byte): Response to operating clock frequency inquiry
- Size (one byte): Number of bytes that represents the number of types, minimum values, and maximum values of operating clock frequencies.

- Number of types (one byte): Number of supported operating clock frequency types (e.g. when there are two operating clock frequency types, which are the main and peripheral clocks, the number of types will be H'02)
- Minimum value of operating clock frequency (two bytes): Minimum value for each multiplied or divided clock frequency.  
The minimum and maximum values represent the values in MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the value is 20.00 MHz, it will be multiplied by 100 to be 2000 which is H'07D0)
- Maximum value of operating clock frequency (two bytes): Maximum value for each multiplied or divided clock frequency.  
There are as many pairs of minimum and maximum values as there are operating clock frequencies.
- SUM (one byte): Checksum

#### (7) User Boot MAT Information Inquiry

The boot program will return the number of user boot MATs and their addresses.

Command 

H'24
------

- Command: H'24 (one byte): Inquiry regarding user boot MAT information

Response	H'34	Size	Number of areas	
	Start address of area			Last address of area
	...			
	SUM			

- Response: H'34 (one byte): Response to user boot MAT information inquiry
- Size (one byte): Amount of data that represents the number of areas, the start address of each area, and the last address of each area
- Number of areas (one byte): Number of non-consecutive user boot MAT areas  
When user boot MAT areas are consecutive, the number of areas returned is H'01.
- Start address of area (four bytes): Start address of the area
- Last address of area (four bytes): Last address of the area  
There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

#### (8) User MAT Information Inquiry

The boot program will return the number of user MATs and their addresses.

Command 

H'25
------

- Command: H'25 (one byte): Inquiry regarding user MAT information

Response	H'35	Size	Number of areas	
	Start address of area			Last address of area
	...			
	SUM			

- Response: H'35 (one byte): Response to the user MAT information inquiry
- Size (one byte): Amount of data that represents the number of areas, the start address of each area, and the last address of each area
- Number of areas (one byte): Number of non-consecutive user MAT areas  
When user MAT areas are consecutive, the number of areas returned is H'01.
- Start address of area (four bytes): Start address of the area
  - Last address of area (four bytes): Last address of the area  
There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

#### (9) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command 

H'26
------

- Command: H'26 (one byte): Inquiry regarding erased block information

Response	H'36	Size	Number of blocks	
	Start address of block			Last address of block
	...			
	SUM			

- Response: H'36 (one byte): Response to the number of erased blocks and addresses
- Size (two bytes): Amount of data that represents the number of blocks, the start address of each block, and the last address of each block
- Number of blocks (one byte): Number of erased blocks in flash memory
- Start address of block (four bytes): Start address of the block
- Last address of block (four bytes): Last address of the block  
There are as many groups of data representing the start and last addresses as there are blocks.
- SUM: Checksum

#### (10) Programming Unit Inquiry

The boot program will return the programming unit used to program data.

Command 

H'27
------

- Command: H'27 (one byte): Inquiry regarding programming unit

Response	H'37	Size	Programming unit	SUM
----------	------	------	------------------	-----

- Response: H'37 (one byte): Response to programming unit inquiry
- Size (one byte): Number of characters that indicate the programming unit (fixed at 2)



— Programming unit (two bytes): Unit for programming  
This is the unit for reception of program data.

— SUM (one byte): Checksum

### (11) Inquiry of Two-MAT Simultaneous Programming

For an inquiry of two-MAT simultaneous programming, the boot program returns the response whether two-MAT simultaneous programming is possible or not, and the start address.

Command 

H'28
------

— Command: H'28 (one byte): Inquiry of two-MAT simultaneous programming

Response	H'38	Size	Programming method		
	First MAT start address			Second MAT start address	
	SUM				

— Response: H'38 (one byte): Response to the inquiry of two-MAT simultaneous programming

— Size (one byte): Total amount of programming method and MAT start address  
5 bytes when programming to one MAT, 9 bytes when programming to two MATs simultaneously

— Programming method (one byte): H'01 = One-MAT programming  
H'02 = Two-MAT simultaneous programming

— First MAT start address (four bytes): First MAT start address

— Second side MAT start address four bytes): Second MAT start address

— Data on second MAT start address is available only when two-MAT simultaneous programming is possible.

— SUM (one byte): Checksum

### (12) New Bit Rate Selection

The boot program will set a new bit rate and return the new bit rate.

This selection should be sent after sending the clock-mode selection command.

Command	H'3F	Size	Bit rate	Input frequency
	Number of multiplication ratios	Multiplication ratio 1	Multiplication ratio 2	
	SUM			

— Command: H'3F (one byte): Selection of new bit rate

— Size (one byte): Amount of data that represents the bit rate, input frequency, number of multiplication ratios, and multiplication ratios

— Bit rate (two bytes): New bit rate

One hundredth of the value (e.g. when the value is 19200 bps, the bit rate is 192, which is H'00C0)

- Input frequency (two bytes): Frequency of the clock input to the boot program  
This value is valid to the hundredths place and represents the value in MHz multiplied by 100. (e.g. when the value is 28.88 MHz, it will be multiplied by 100 to be 2888 which is H'0B48.
- Number of multiplication ratios (one byte): Number of multiplication ratios to which the device can be set.
- Multiplication ratio 1 (one byte): Value of the multiplication or division ratio for the main operating frequency  
Multiplication ratio: Value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)  
Division ratio: Value of the division ratio, inverted to be a negative number (e.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = -2)
- Multiplication ratio 2 (one byte): Value of the multiplication or division ratio for the peripheral operating frequency  
Multiplication ratio: Value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)  
Division ratio: Value of the division ratio, inverted to be a negative number (e.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = -2)
- SUM (one byte): Checksum

Response 

H'06
------

- Response: H'06 (one byte): Response to selection of a new bit rate  
When it is possible to set the bit rate, the response will be ACK.

Error response 

H'BF	ERROR
------	-------

- Error response: H'BF (one byte): Error response to selection of new bit rate
- ERROR: (one byte): Error code
  - H'11: Sum check error
  - H'24: Bit-rate selection error  
This bit rate is not available.
  - H'25: Input frequency error  
This input frequency is not within the range set by the minimum and maximum values.
  - H'26: Multiplication ratio error  
This ratio does not match an available ratio.
  - H'27: Operating frequency error  
This operating frequency is not within the range set by the minimum and maximum values.

The methods for checking of received data are listed below.

- Input frequency  
The received value of the input frequency is checked to ensure that it is within the range of minimum to maximum frequencies which matches the clock modes of the specified device. When the value is out of this range, an input frequency error is generated.

- Multiplication ratio

The received value of the multiplication ratio or division ratio is checked to ensure that it matches the clock modes of the specified device. When the value is out of this range, a multiplication error is generated.

- Operating frequency error

The operating frequency is calculated from the received value of the input frequency and the multiplication or division ratio. The input frequency is input to the LSI and the LSI is actually operated at the operating frequency. The expression is given below.

Operating frequency = Input frequency \* Multiplication ratio, or

Operating frequency = Input frequency / Division ratio

The calculated operating frequency should be checked to ensure that it is within the range of minimum to maximum frequencies which are available with the clock modes of the specified device. When it is out of this range, an operating frequency error is generated.

- Bit rate

From peripheral operating clock ( $\phi$ ) and bit rate (B), the clock select (CKS) value (n) in the serial mode register (SMR) and the bit rate register (BRR) value (N) are obtained.

The error between n and N that is calculated by the method below is checked to ensure that it is less than 4%. When it is 4% or more, a bit-rate selection error is generated.

$$\text{Error (\%)} = \left\{ \left[ \frac{\phi \cdot 10^6}{(N+1) \cdot B \cdot 64 \cdot 2^{(2n-1)}} \right] - 1 \right\} \cdot 100$$

When the new bit rate is selectable, the new bit rate will be set in the register after sending ACK in response. The host will send ACK with the new bit rate for confirmation and the boot program will response with that rate.

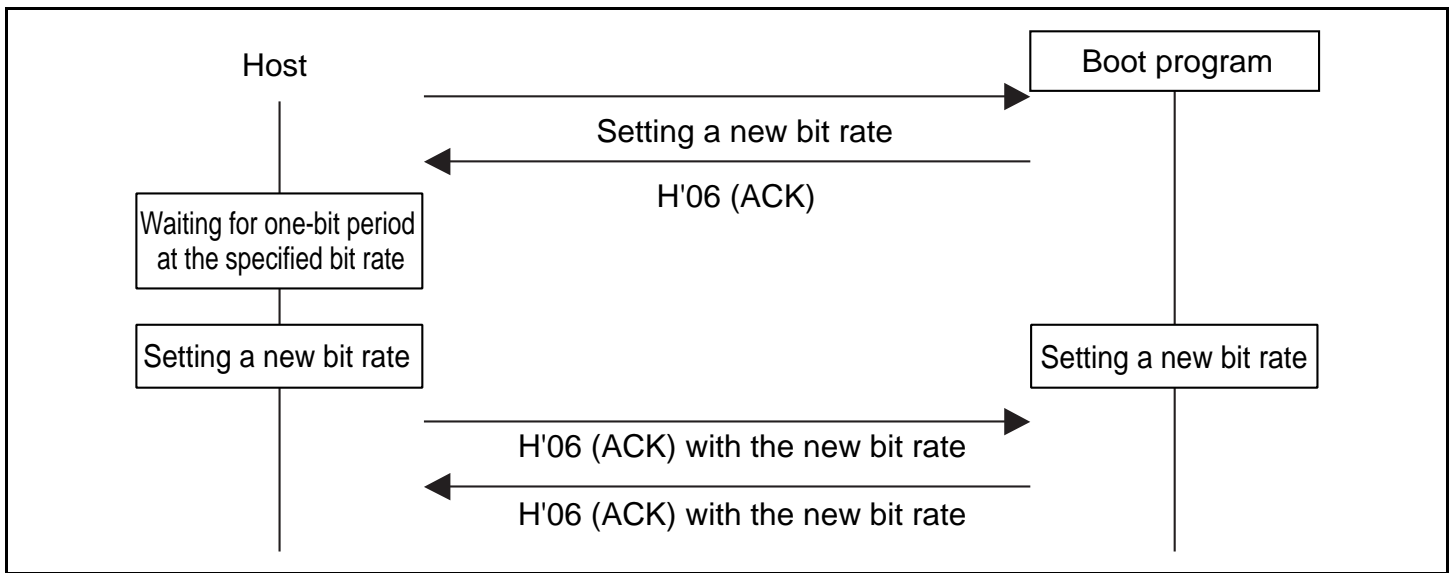
Confirmation H'06

— Confirmation: H'06 (one byte): Confirmation of a new bit rate

Response H'06

— Response: H'06 (one byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 23.29.



**Figure 23.29 New Bit-Rate Selection Sequence**

**Transition to Programming/Erasing State:** To enter the programming/erasing state, the boot program will transfer the erasing program, and erase the user MATs and user boot MATs in that order. On completion of this erasure, ACK will be returned and a transition is made to the programming/erasing state.

The host should select the device code, clock mode, and new bit rate with device selection, clock-mode selection, and new bit-rate selection commands, and then send the command for the transition to programming/erasing state. This procedure should be carried out before transferring the programming selection command or program data.

Command 

H'40
------

— Command: H'40 (one byte): Transition to programming/erasing state

Response 

H'06
------

— Response: H'06 (one byte): Response to transition to programming/erasing state

The boot program will send ACK when the user MATs and user boot MATs have been erased by the transferred erasing program.

Error response 

H'C0	H'51
------	------

— Error response: H'C0 (one byte): Error response to transition to programming/erasing state

— Error code: H'51 (one byte): Erasing error

An error occurred and erasure was not completed.

**Command Error:** A command error will occur when a command is undefined, the order of commands is incorrect, or a command is unacceptable. Issuing a clock-mode selection command before a device selection or issuing an inquiry command after the command for transition to the programming/erasing state, are examples.

Error response 

H'80	H'xx
------	------

— Error response: H'80 (one byte): Command error

— Command: H'xx (one byte): Received command

**Command Order:** The order for commands in the inquiry selection state is shown below.

- (1) A supported device inquiry (H'20) should be made to inquire about the supported devices.
- (2) The device should be selected from among those described by the returned information and set with a device selection (H'10) command.
- (3) A clock-mode inquiry (H'21) should be made to inquire about the supported clock modes.
- (4) The clock mode should be selected from among those described by the returned information and set with a clock-mode selection (H'11) command.
- (5) After selection of the device and clock mode, inquiries for other required information should be made, such as the multiplication ratio inquiry (H'22) or operating frequency inquiry (H'23).
- (6) A new bit rate should be selected with the new bit-rate selection (H'3F) command, according to the returned information on multiplication ratios and operating frequencies.
- (7) After selection of the device and clock mode, the information of the user boot MAT and user MAT should be made to inquire about the user boot MAT information inquiry (H'24), user MAT information inquiry (H'25), erased block information inquiry (H'26), programming unit inquiry (H'27), and two-MAT simultaneous programming information inquiry (H'28).
- (8) After making inquiries and selecting a new bit rate, issue the command for transition to the programming/erasing state (H'40). The boot program will then enter the programming/erasing state.

**Programming/Erasing State:** In the programming/erasing state, a programming selection command makes the boot program select the programming method, a 128-byte programming command makes it program the memory with data, and an erasing selection command and block erasing command make it erase the block. Table 23.19 lists the programming/erasing commands.

**Table 23.19 Programming/Erasing Commands**

Command	Command Name	Description
H'42	User boot MAT programming selection	Transfers the user boot MAT programming program
H'43	User MAT programming selection	Transfers the user MAT programming program
H'50	128-byte programming	Programs 128 bytes of data
H'48	Erasing selection	Transfers the erasing program
H'58	Block erasing	Erases a block of data
H'52	Memory read	Reads the contents of memory
H'4A	User boot MAT sum check	Checks the checksum of the user boot MAT
H'4B	User MAT sum check	Checks the checksum of the user MAT
H'4C	User boot MAT blank check	Checks whether the contents of the user boot MAT are blank
H'4D	User MAT blank check	Checks whether the contents of the user MAT are blank
H'4F	Boot program status inquiry	Inquires into the boot program's state

**Programming:** Programming is executed by a programming selection command and a 128-byte programming command.

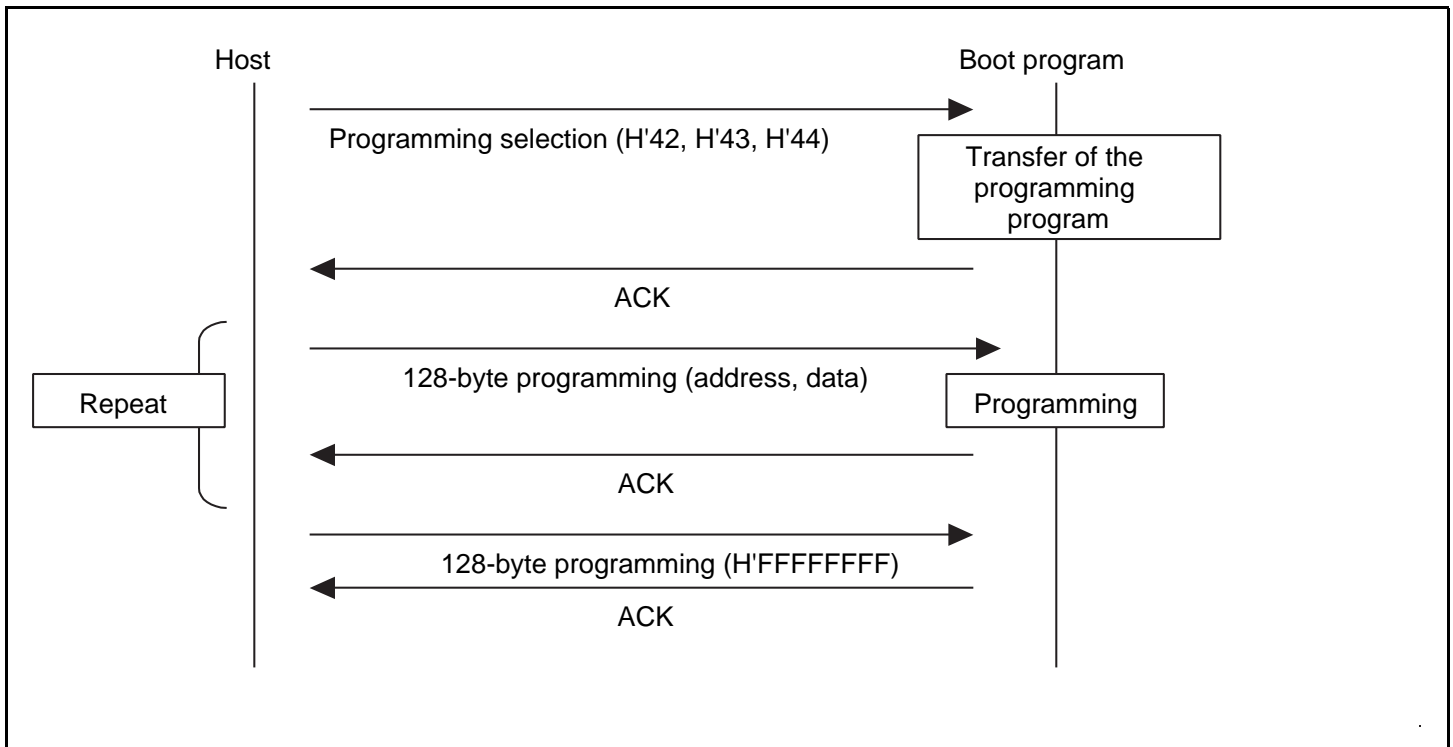
First, the host should send the programming selection command and select the programming method and programming MATs. There are three programming selection commands used according to the area and method for programming.

- (1) User boot MAT programming selection
- (2) User MAT programming selection
- (3) Two-user-MAT simultaneous programming selection

After issuing the programming selection command, the host should send the 128-byte programming command. The 128-byte programming command that follows the selection command represents the data programmed according to the method specified by the selection command. When more than 128-byte data is programmed, 128-byte commands should repeatedly be executed. Sending a 128-byte programming command with H'FFFFFFFF as the address will stop the programming. On completion of programming, the boot program will wait for selection of programming or erasing.

To continue programming with another method or of another MAT, the procedure must be repeated from the programming selection command.

The programming selection command and sequence for the 128-byte programming commands are shown in figure 23.30.



**Figure 23.30 Programming Sequence**

(1) User boot MAT programming selection

The boot program will transfer a programming program. The data is programmed to the user boot MATs by the transferred programming program.

Command H'42

— Command: H'42 (one byte): User boot MAT programming selection

Response H'06

— Response: H'06 (one byte): Response to user boot MAT programming selection

When the programming program has been transferred, the boot program will return ACK.

Error response H'C2 ERROR

— Error response: H'C2 (one byte): Error response to user boot MAT programming selection

— ERROR: (one byte): Error code

H'54: Selection processing error (transfer error occurs and processing is not completed)

(2) User MAT programming selection

The boot program will transfer a programming program. The data is programmed to the user MATs by the transferred programming program.

Command H'43

— Command: H'43 (one byte): User MAT programming selection

Response 

H'06
------

- Response: H'06 (one byte): Response to user MAT programming selection  
When the programming program has been transferred, the boot program will return ACK.

Error response 

H'C3	ERROR
------	-------

- Error response: H'C3 (one byte): Error response to user MAT programming selection
- ERROR: (one byte): Error code  
H'54: Selection processing error (transfer error occurs and processing is not completed)

### (3) Selection of Two-User-Boot MAT Simultaneous Programming

In response to the selection of two-user-boot MAT simultaneous programming, the boot program transfers a program for the two-user-boot MAT simultaneous programming. The transferred program for the two-user-boot MAT simultaneous programming is used to program data onto two user boot MATs simultaneously. Taking into account two-MAT simultaneous write, addresses and data corresponding to the two MATs should be alternately transmitted at the host side. Though the boot program returns ACK each time one 128-byte programming command is given, data is not programming until data for two user boot MATs is completely prepared.

Command 

H'44
------

- Command: Command: H'44 (one byte): Selection of two-user-boot MAT simultaneous programming

Response 

H'06
------

- Response: H'06 (one byte): Response to the selection of two-user-boot MAT simultaneous programming. ACK is given when the programming program has been transferred.

Error response 

H'C4	ERROR
------	-------

- Error response: H'C4 (one byte): Error response to the selection of two-user-boot MAT simultaneous programming
- ERROR: (one byte): Error code  
H'54: Selection processing error (transfer error occurs and processing is not completed)

### (4) 128-byte programming

The boot program will use the programming program transferred by the programming selection command for programming the user boot MATs or user MATs. When two-user-MAT simultaneous programming command is selected, programming will start after the boot program has received data for both MATs.

Command	H'50	Programming address						
	Data	...						
	...							
	SUM							

- Command: H'50 (one byte): 128-byte programming



- Programming address (four bytes): Start address for programming  
Multiple of the size specified in response to the programming unit inquiry; a 128-byte boundary (e.g. H'00, H'01, H'00, H'00: H'01000000)
- Data (n bytes): Data to be programmed  
The size is specified in response to the programming unit inquiry.
- SUM (one byte): Checksum

Response 

H'06
------

- Response: H'06 (one byte): Response to 128-byte programming  
On completion of programming, the boot program will return ACK. In two-MAT programming, when all data for the first MAT has been received, the boot program will return ACK.

Error response 

H'D0	ERROR
------	-------

- Error response: H'D0 (one byte): Error response to 128-byte programming
- ERROR: (one byte): Error code  
H'11: Sum check error  
H'2A: Address error (address is not within the specified range)  
H'53: Programming error (a programming error has occurred and programming cannot be continued)

The specified address should match the unit for programming of data. For example, when the programming is in 128-byte units, the lower byte of the address should be H'00 or H'80.

When there are less than 128 bytes of data to be programmed, the host should fill the rest with H'FF.

In two-user-MAT simultaneous programming, the host should alternately send the data for each MAT address.

Sending the 128-byte programming command with the address of H'FFFFFFFF will stop the programming operation. The boot program will interpret this as the end of programming and wait for selection of programming or erasing. When the most recently received data has not been programmed in two-user-MAT simultaneous programming, the most recent data is programmed before programming is stopped.

Command 

H'50	Programming address	SUM
------	---------------------	-----

- Command: H'50 (one byte): 128-byte programming
- Programming address (four bytes): End code is H'FF, H'FF, H'FF, H'FF.
- SUM (one byte): Checksum

Error response 

H'D0	ERROR
------	-------

- Error response: H'D0 (one byte): Error response to 128-byte programming

— ERROR: (one byte): Error code

H'11: Sum check error

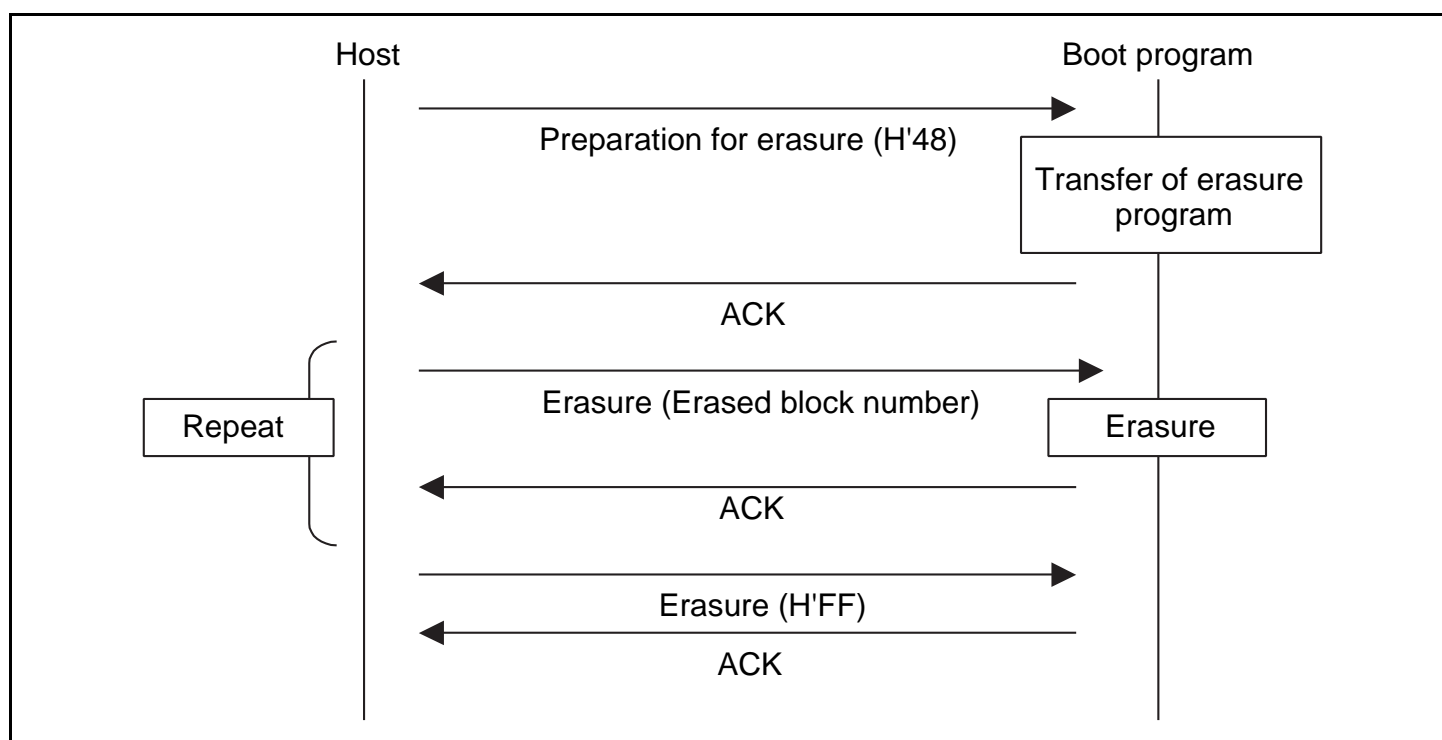
H'53: Programming error

An error has occurred in programming, and programming cannot be continued (in two-user-MAT simultaneous programming, when programming to the last MAT has not been completed.)

**Erase:** Erasure is performed with the erasing selection and block erasing command.

First, erasure is selected by the erasing selection command and the boot program then erases the block specified by the block erasing command. The command should be repeatedly executed if two or more blocks are to be erased. Sending a block erasing command from the host with the block number H'FF will stop erasure. On completion of erasing, the boot program will wait for selection of programming or erasing.

The erasing selection command and sequence for erasing data are shown in figure 23.31.



**Figure 23.31 Erasing Sequence**

#### (1) Erasing selection

The boot program will transfer the erasing program. User MAT data is erased by the transferred erasing program.

Command H'48

— Command: H'48 (one byte): Erasing selection

Response 

H'06
------

- Response: H'06 (one byte): Response to erasing selection  
After the erasing program has been transferred, the boot program will return ACK.

Error response 

H'C8	ERROR
------	-------

- Error response: H'C8 (one byte): Error response to erasing selection
- ERROR: (one byte): Error code  
H'54: Selection processing error (transfer error occurs and processing is not completed)

## (2) Block erasing

The boot program will erase the contents of the specified block.

Command 

H'58	Size	Block number	SUM
------	------	--------------	-----

- Command: H'58 (one byte): Erasing
- Size (one byte): Number of characters that represents the erasure block number (fixed at 1)
- Block number (one byte): Number of the block whose data is to be erased
- SUM (one byte): Checksum

Response 

H'06
------

- Response: H'06 (one byte): Response to erasing  
After erasure has been completed, the boot program will return ACK.

Error response 

H'D8	ERROR
------	-------

- Error response: H'D8 (one byte): Error response to erasing  
H'11: Sum check error  
H'29: Block number error  
Block number is incorrect.  
H'51: Erasure error  
An error has occurred during erasure.

On receiving block number H'FF, the boot program will stop erasure and wait for a selection command.

Command 

H'58	Size	Block number	SUM
------	------	--------------	-----

- Command: H'58 (one byte): Erasure
- Size (one byte): Number of characters that represents the block number (fixed at 1)
- Block number (one byte): H'FF (stop code for erasure)
- SUM (one byte): Checksum

Response 

H'06
------

- Response: H'06 (one byte): Response to end of erasure (ACK)  
When erasure is to be performed again after the block number H'FF has been sent, the procedure should be executed from the erasure selection command.

**Memory Read:** The boot program will return the data in the specified address.

Command	H'52	Size	Area	Read start address		
	Read size			SUM		

- Command: H'52 (one byte): Memory read
- Size (one byte): Amount of data that represents the area, read address, and read size (fixed at 9)
- Area (one byte)
  - H'11: User boot MAT
  - H'01: User MAT
 An address error occurs when the area setting is incorrect.
- Read start address (four bytes): Start address to be read from
- Read size (four bytes): Size of data to be read
- SUM (one byte): Checksum

Response	H'52	Read size						
	Data	...						
	SUM							

- Response: H'52 (one byte): Response to memory read
- Read size (four bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (one byte): Checksum

Error response	H'D2	ERROR
----------------	------	-------

- Error response: H'D2 (one byte): Error response to memory read
- ERROR: (one byte): Error code
  - H'11: Sum check error
  - H'2A: Address error  
The read address is not in the MAT.
  - H'2B: Size error  
The read size exceeds the MAT. Or, the read end address calculated from the read start address and read size is out of the MAT range, or the read size is 0.

**User Boot MAT Sum Check:** The boot program will add the amount of data in user boot MATs and return the result.

Command	H'4A
---------	------

- Command: H'4A (one byte): Sum check of user boot MATs

Response	H'5A	Size	MAT checksum	SUM
----------	------	------	--------------	-----

- Response: H'5A (one byte): Response to sum check of user boot MATs
- Size (one byte): Number of characters in checksum data (fixed at 4)

- MAT checksum (four bytes): Checksum of user boot MATs  
The total amount of data is obtained in byte units.
- SUM (one byte): Checksum (for transmit data)

**User MAT Sum Check:** The boot program will add the amount of data in user MATs and return the result.

Command 

H'4B
------

- Command: H'4B (one byte): Sum check of user MATs

Response 

H'5B	Size	MAT checksum	SUM
------	------	--------------	-----

- Response: H'5B (one byte): Response to sum check of user MATs
- Size (one byte): Number of characters in checksum data (fixed at 4)
- MAT checksum (four bytes): Checksum of user MATs  
The total amount of data is obtained in byte units.
- SUM (one byte): Checksum (for transmit data)

**User Boot MAT Blank Check:** The boot program will check whether or not all user boot MATs are blank and return the result.

Command 

H'4C
------

- Command: H'4C (one byte): Blank check of user boot MATs

Response 

H'06
------

- Response: H'06 (one byte): Response to blank check of user boot MATs  
If all user boot MATs are blank (H'FF), the boot program will return ACK.

Error response 

H'CC	H'52
------	------

- Error response: H'CC (one byte): Error response to blank check of user boot MATs
- Error code: H'52 (one byte): Erasure has not been completed

**User MAT Blank Check:** The boot program will check whether or not all user MATs are blank and return the result.

Command 

H'4D
------

- Command: H'4D (one byte): Blank check of user MATs

Response 

H'06
------

- Response: H'06 (one byte): Response to blank check of user MATs  
If all user MATs are blank (H'FF), the boot program will return ACK.

Error response 

H'CD	H'52
------	------

- Error response: H'CD (one byte): Error response to blank check of user MATs
- Error code: H'52 (one byte): Erasure has not been completed.

**Boot Program Status Inquiry:** The boot program will return indications of its present state and error condition. This inquiry can be made in the inquiry/selection state or the programming/erasing state.

Command 

H'4F
------

— Command: H'4F (one byte): Inquiry regarding boot program status

Response 

H'5F	Size	STATUS	ERROR	SUM
------	------	--------	-------	-----

— Response: H'5F (one byte): Response to inquiry regarding boot program status

— Size (one byte): Number of characters in data (fixed at 2)

— STATUS (one byte): Standard boot program status

For details, see table 23.20, Status Code

— ERROR (one byte): Error state

ERROR = 0 indicates normal operation.

ERROR = 1 indicates error has occurred

For details, see table 23.21, Error Code.

— SUM (one byte): Checksum

**Table 23.20 Status Code**

Code	Description
H'01	Device Selection Wait
H'02	Clock Mode Selection Wait
H'03	Bit Rate Selection Wait
H'0F	Programming/Erasing State Transition Wait (bit rate selection is completed)
H'31	Programming State for Erasing User MAT and User Boot MAT
H'3F	Programming/Erasing Selection Wait (Erasure is completed)
H'4F	Programming Data Receive Wait
H'5F	Erasure Block Specification Wait (erasure is completed)

**Table 23.21 Error Code**

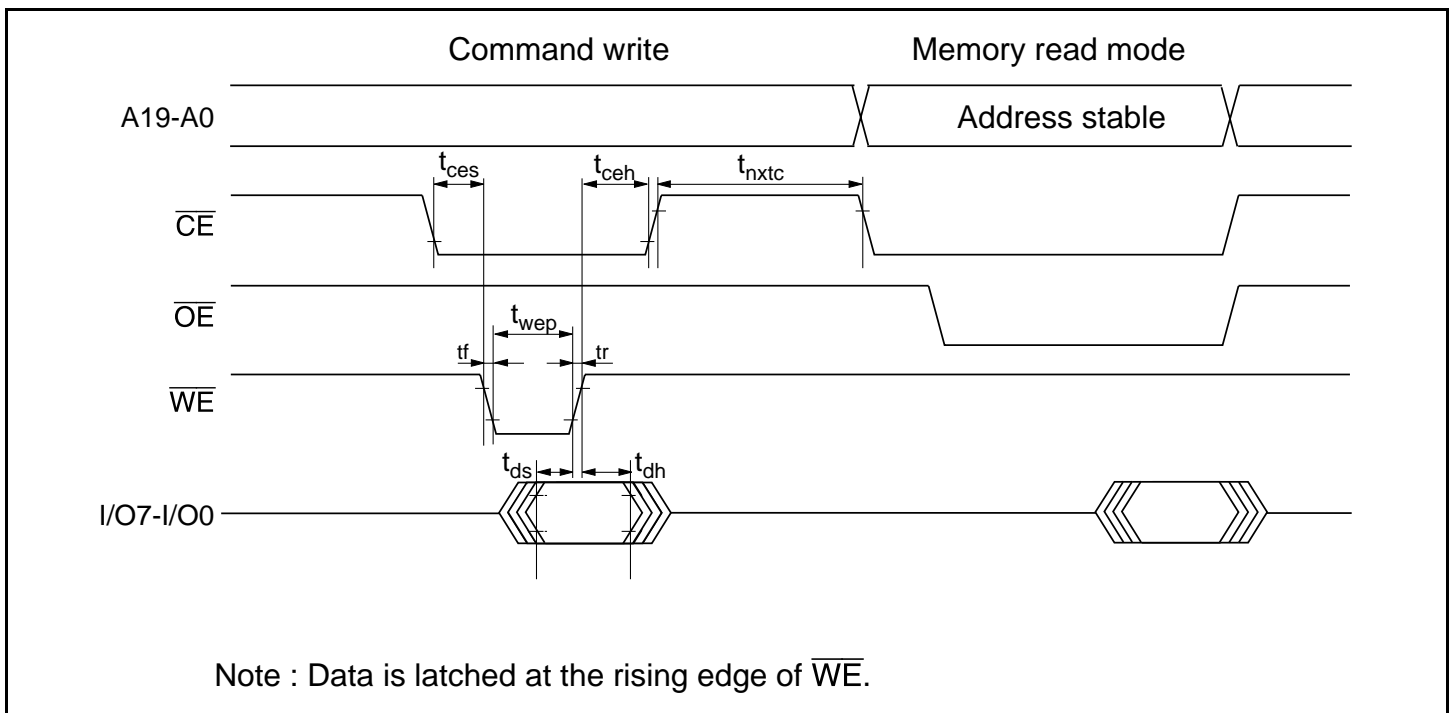
<b>Code</b>	<b>Description</b>
H'00	No Error
H'11	Sum Check Error
H'21	Device Code Mismatch Error
H'22	Clock Mode Mismatch Error
H'24	Bit Rate Selection Error
H'25	Input Frequency Error
H'26	Multiplication Ratio Error
H'27	Operating Frequency Error
H'29	Block Number Error
H'2A	Address Error
H'2B	Data Length Error
H'51	Erase Error
H'52	Erase Incompletion Error
H'53	Programming Error
H'54	Selection Error
H'80	Command Error
H'FF	Bit-Rate-Adjustment Confirmation Error

## 23.10.2 AC Characteristics and Timing in Programmer Mode

**Table 23.22 AC Characteristics in Memory Read Mode**

Condition:  $V_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $T_a = 25^\circ\text{C} \pm 5^\circ\text{C}$

Code	Symbol	Min	Max	Unit	Note
Command write cycle	$t_{nxtc}$	20		$\mu\text{s}$	
$\overline{\text{CE}}$ hold time	$t_{ceh}$	0		ns	
$\overline{\text{CE}}$ setup time	$t_{ces}$	0		ns	
Data hold time	$t_{dh}$	50		ns	
Data setup time	$t_{ds}$	50		ns	
Programming pulse width	$t_{wep}$	70		ns	
$\overline{\text{WE}}$ rise time	$t_r$		30	ns	
$\overline{\text{WE}}$ fall time	$t_f$		30	ns	



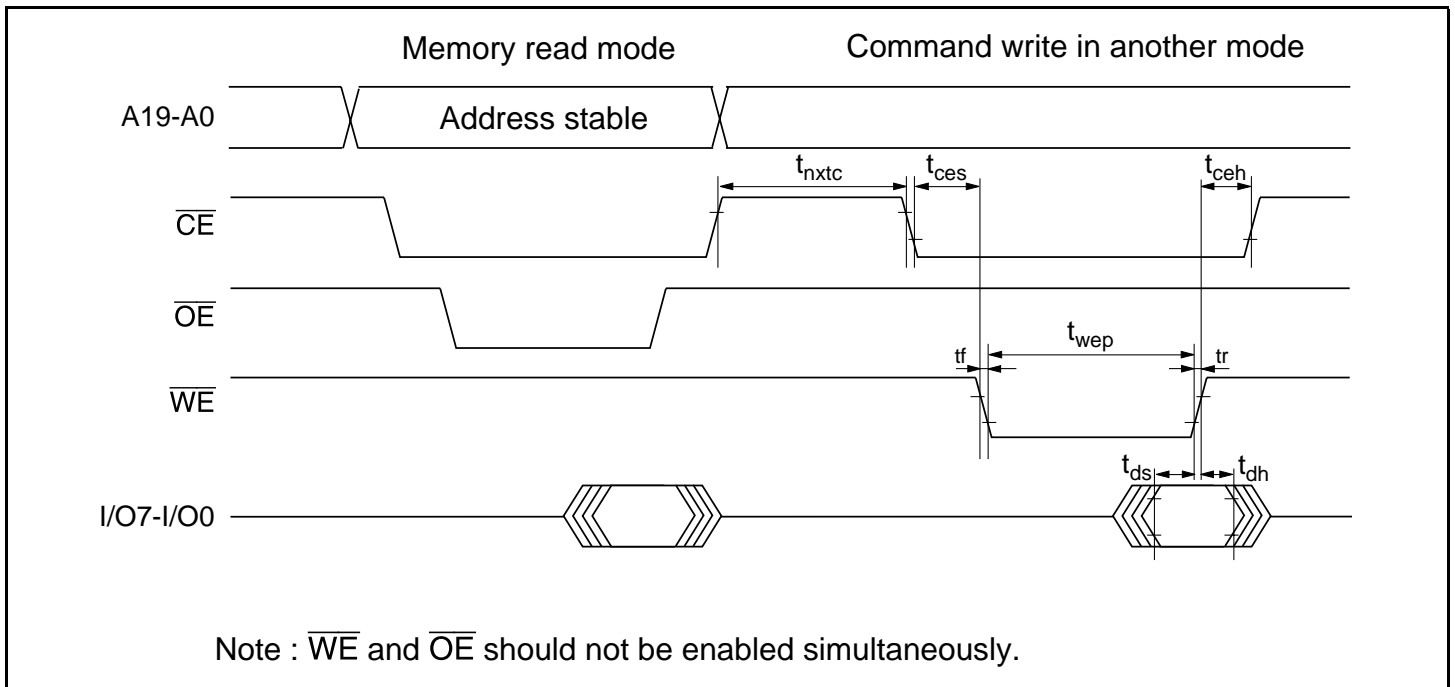
**Figure 23.32 Memory Read Timing after Command Write**



**Table 23.23 AC Characteristics in Transition from Memory Read Mode to Others**

Condition:  $V_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $T_a = 25^\circ\text{C} \pm 5^\circ\text{C}$

Code	Symbol	Min	Max	Unit	Note
Command write cycle	$t_{nxtc}$	20		$\mu\text{s}$	
$\overline{\text{CE}}$ hold time	$t_{ceh}$	0		ns	
$\overline{\text{CE}}$ setup time	$t_{ces}$	0		ns	
Data hold time	$t_{dh}$	50		ns	
Data setup time	$t_{ds}$	50		ns	
Programming pulse width	$t_{wep}$	70		ns	
$\overline{\text{WE}}$ rise time	$t_r$		30	ns	
$\overline{\text{WE}}$ fall time	$t_f$		30	ns	



**Figure 23.33 Timing at Transition from Memory Read Mode to Other Modes**

### Table 23.24 AC Characteristics in Memory Read Mode

Condition:  $V_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ ,  $T_a = 25^\circ\text{C} \pm 5^\circ\text{C}$

Code	Symbol	Min	Max	Unit	Note
Access time	$t_{acc}$		20	$\mu\text{s}$	
$\overline{\text{CE}}$ output delay time	$t_{ce}$		150	ns	
$\overline{\text{OE}}$ output delay time	$t_{oe}$		150	ns	
Output disable delay time	$t_{df}$		100	ns	
Data output hold time	$t_{oh}$	5		ns	

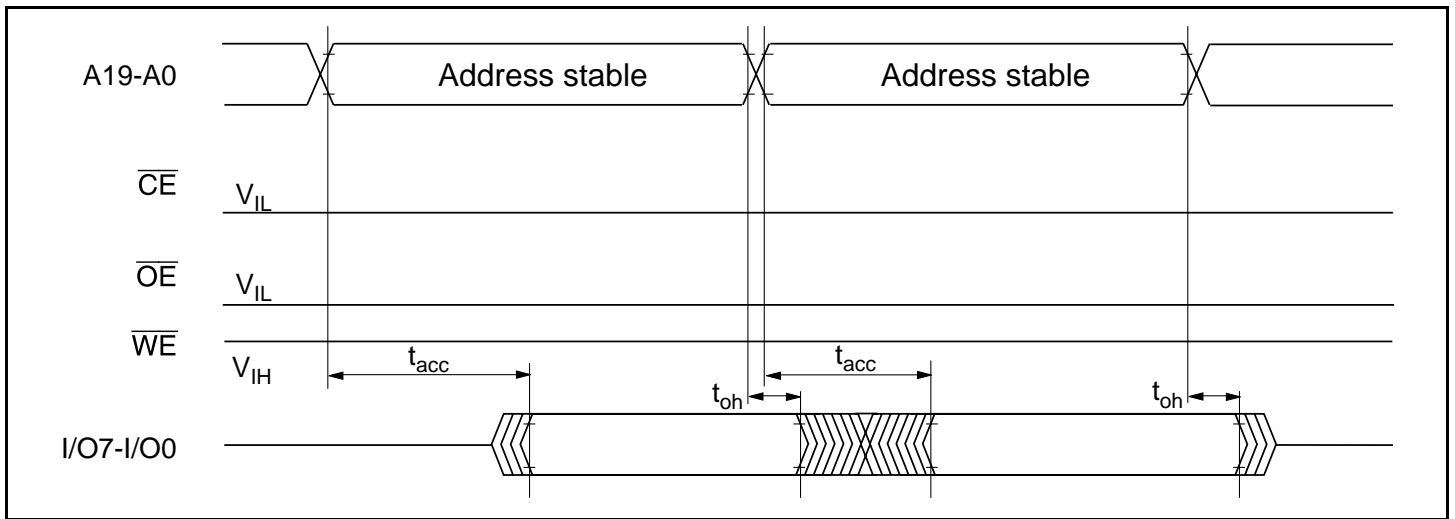


Figure 23.34  $\overline{\text{CE}}/\overline{\text{OE}}$  Enable State Read

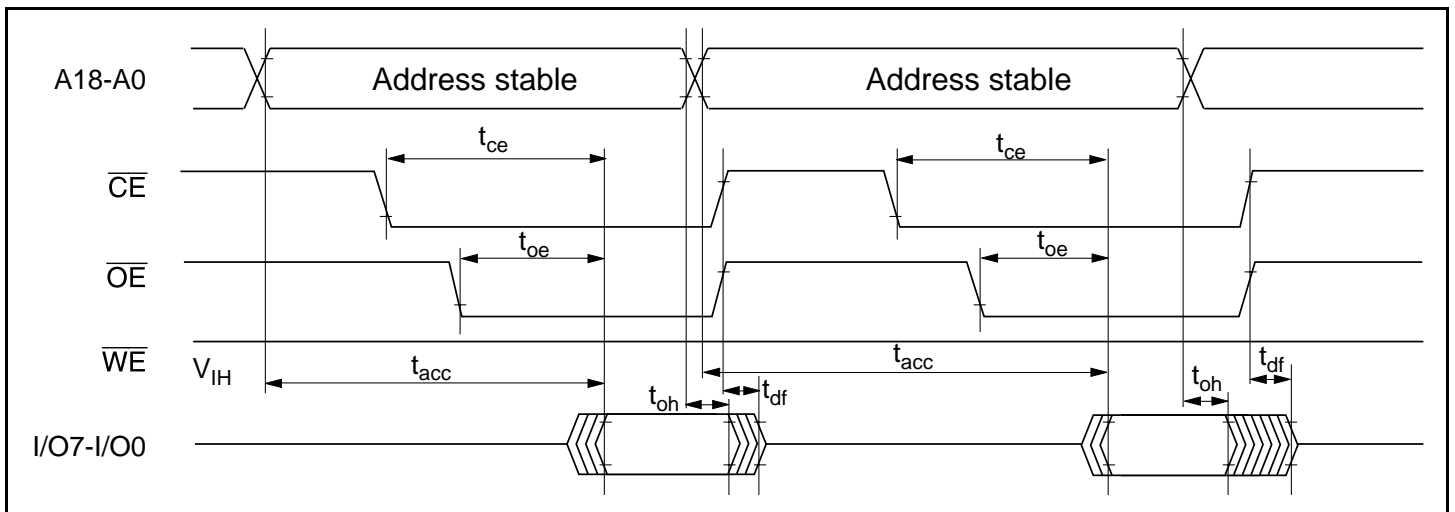


Figure 23.35  $\overline{\text{CE}}/\overline{\text{OE}}$  Clock Read

### Table 23.25 AC Characteristics in Auto-Program Mode

Condition:  $V_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{SS} = 0\text{ V}$ ,  $T_a = 25^\circ\text{C} \pm 5^\circ\text{C}$

Code	Symbol	Min	Max	Unit	Note
Command write cycle	$t_{nxtc}$	20		$\mu\text{s}$	
$\overline{\text{CE}}$ hold time	$t_{ceh}$	0		ns	
$\overline{\text{CE}}$ setup time	$t_{ces}$	0		ns	
Data hold time	$t_{dh}$	50		ns	
Data setup time	$t_{ds}$	50		ns	
Programming pulse width	$t_{wep}$	70		ns	
Status polling start time	$t_{wsts}$	1		ms	
Status polling access time	$t_{spa}$		150	ns	
Address setup time	$t_{as}$	0		ns	
Address hold time	$t_{ah}$	60		ns	
Memory programming time	$t_{write}$	1	3000	ms	
Programming setup time	$t_{pns}$	100		ns	
Programming end setup time	$t_{pnh}$	100		ns	
$\overline{\text{WE}}$ rise time	$t_r$		30	ns	
$\overline{\text{WE}}$ fall time	$t_f$		30	ns	

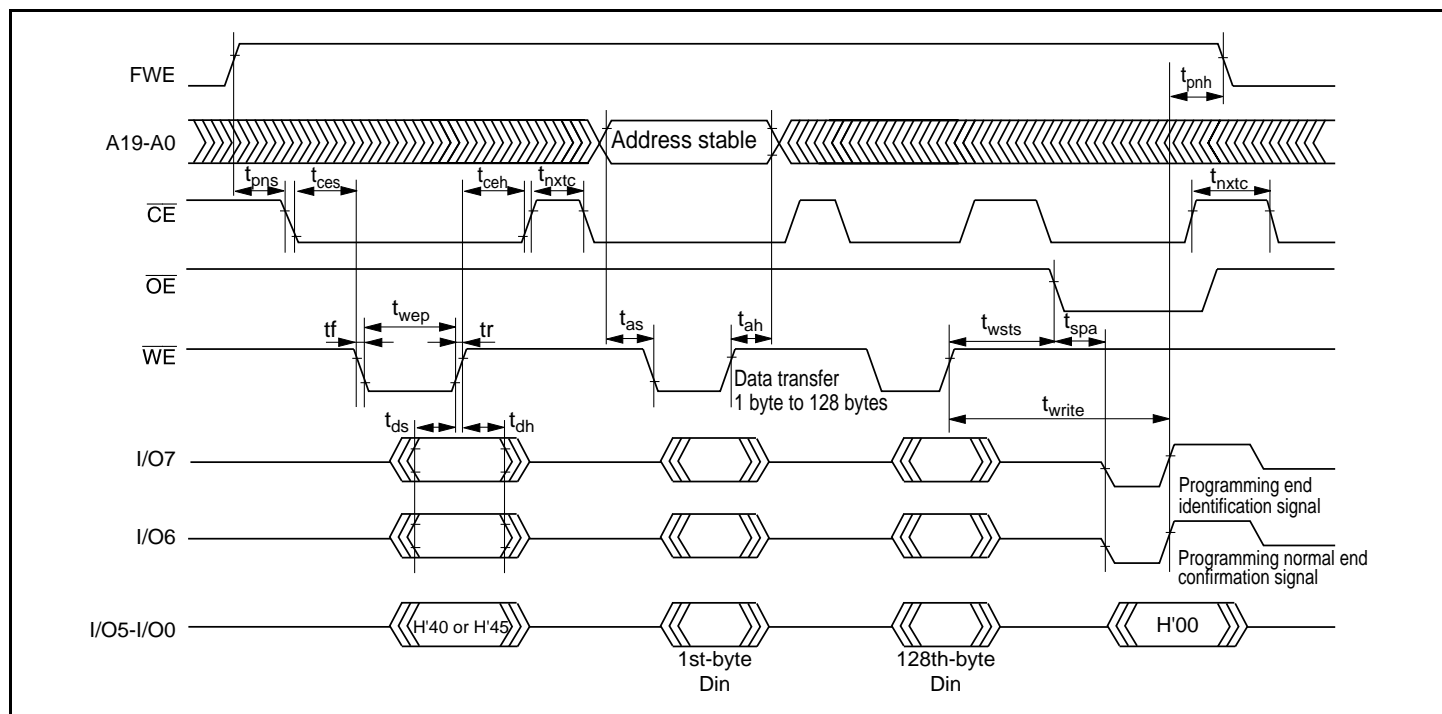


Figure 23.36 Timing in Auto-Program Mode

### Table 23.26 AC Characteristics in Auto-Erase Mode

Condition:  $V_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ ,  $T_a = 25^\circ\text{C} \pm 5^\circ\text{C}$

Code	Symbol	Min	Max	Unit	Note
Command write cycle	$t_{nxtc}$	20		$\mu\text{s}$	
CE hold time	$t_{ceh}$	0		ns	
CE setup time	$t_{ces}$	0		ns	
Data hold time	$t_{dh}$	50		ns	
Data setup time	$t_{ds}$	50		ns	
Programming pulse width	$t_{wep}$	70		ns	
Status polling start time	$t_{ests}$	1		ms	
Status polling access time	$t_{spa}$		150	ns	
Memory erase time	$t_{erase}$	100	40000	ms	
Erase setup time	$t_{ens}$	100		ns	
Erase end setup time	$t_{enh}$	100		ns	
$\overline{\text{WE}}$ rise time	$t_r$		30	ns	
$\overline{\text{WE}}$ fall time	$t_f$		30	ns	

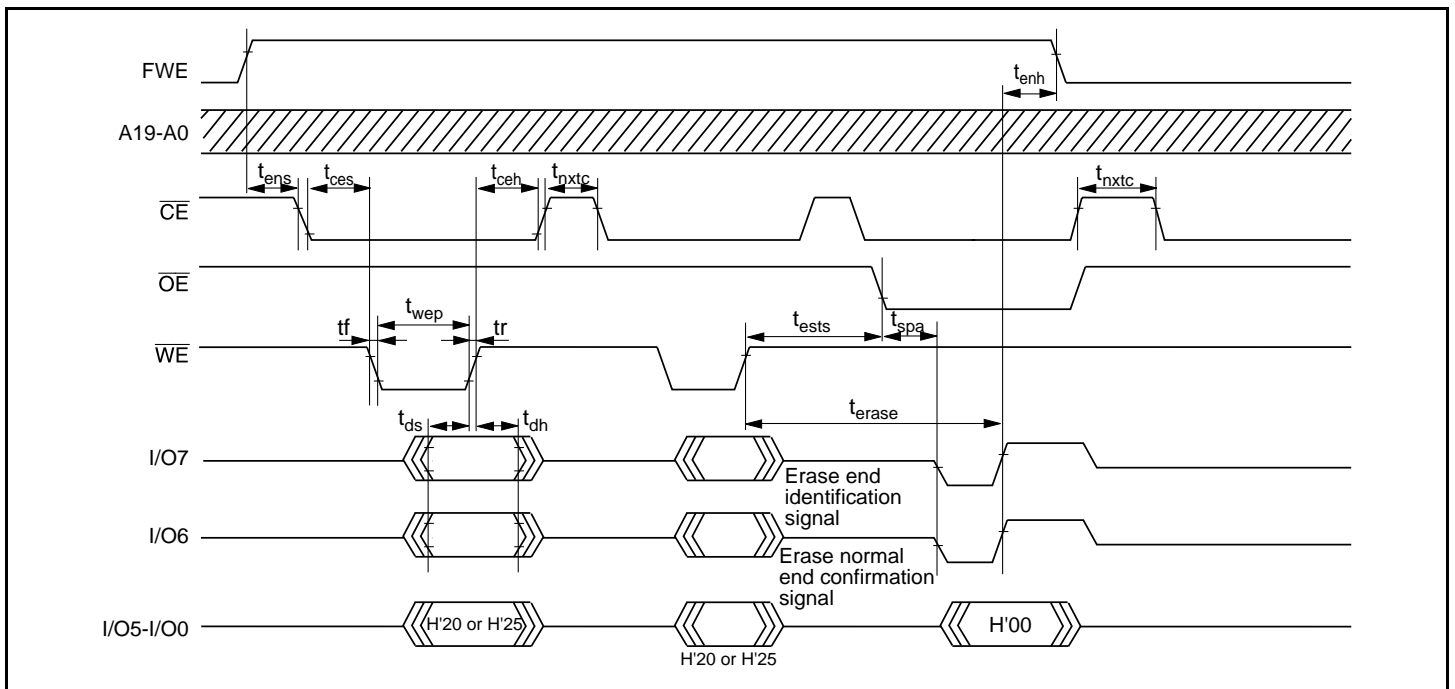
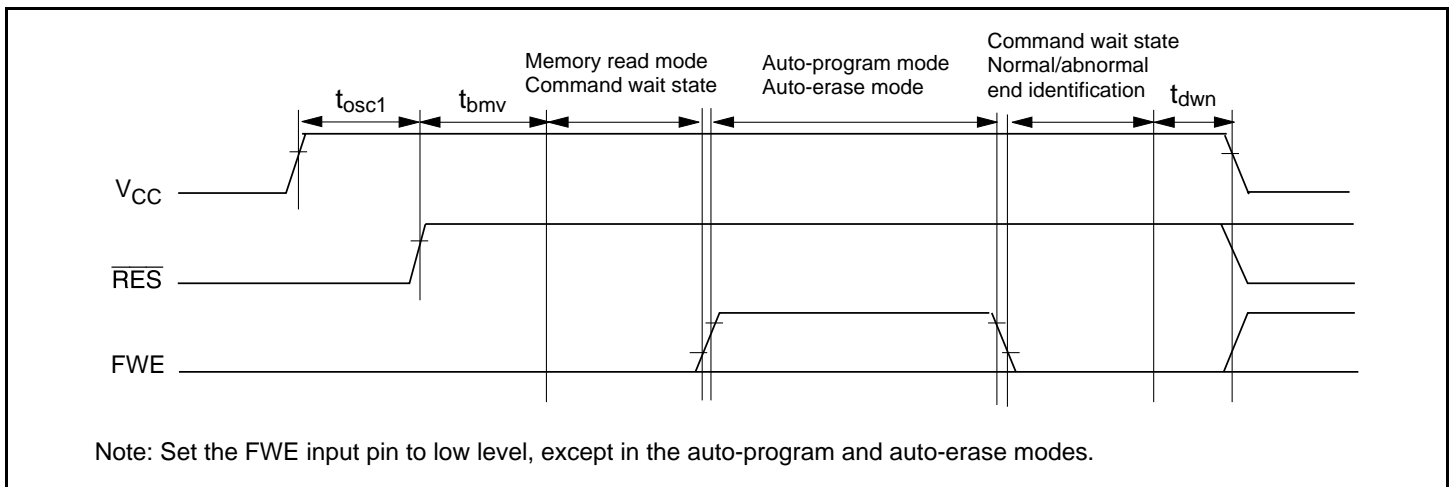


Figure 23.37 Timing in Auto-Erase Mode



**Table 23.28 Stipulated Transition Times to Command Wait State**Condition:  $V_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{SS} = 0 \text{ V}$ ,  $T_a = 25^\circ\text{C} \pm 5^\circ\text{C}$ 

Code	Symbol	Min	Max	Unit	Note
Standby release (oscillation stabilization time)	$t_{osc1}$	30		ms	
Programmer mode setup time	$t_{bmv}$	10		ms	
$V_{CC}$ hold time	$t_{dwn}$	0		ms	

**Figure 23.39 Oscillation Stabilization Time, Programmer Mode Setup Time, and Power-Down Sequence**

### 23.10.3 Storable Area for Procedure Program and Programming Data

In the descriptions in the previous section, storable areas for the programming/erasing procedure programs and program data are assumed to be in on-chip RAM. However, the procedure programs and data can be stored in and executed from other areas (e.g. external address space) as long as the following conditions are satisfied.

- (1) The on-chip programming/erasing program is downloaded from the address set by FTDAR in on-chip RAM, therefore, this area is not available for use.
- (2) The on-chip programming/erasing program will use 128 bytes or more as a stack. Make sure this area is reserved.
- (3) Since download by setting the SCO bit to 1 will cause the MATs to be switched, it should be executed in on-chip RAM.
- (4) The flash memory is accessible until the start of programming or erasing, that is, until the result of downloading has been judged. When in a mode in which the external address space is not accessible, such as single-chip mode, the required procedure programs, interrupt vector

table, interrupt processing routine, and user branch program should be transferred to on-chip RAM before programming/erasing of the flash memory starts.

- (5) The flash memory is not accessible during programming/erasing operations. Therefore, the programming/erasing program must be downloaded to on-chip RAM in advance. Areas for executing each procedure program for initiating programming/erasing, the user program at the user branch destination for programming/erasing, the interrupt vector table, and the interrupt processing routine must be located in on-chip memory other than flash memory or the external address space.
- (6) After programming/erasing, access to flash memory is inhibited until FKEY is cleared. A reset state ( $\overline{\text{RES}} = 0$ ) for more than at least 100  $\mu\text{s}$  must be taken when the LSI mode is changed to reset on completion of a programming/erasing operation.  
Transitions to the reset state or hardware standby mode during programming/erasing are inhibited. When the reset signal is accidentally input to the LSI, a longer period in the reset state than usual (100  $\mu\text{s}$ ) is needed before the reset signal is released.
- (7) Switching of the MATs by FMATS is needed for programming/erasing of the user MAT in user boot mode. The program which switches the MATs should be executed from the on-chip RAM. For details, see section 23.8.1, Switching between User MAT and User Boot MAT. Please make sure you know which MAT is selected when switching the MATs.
- (8) When the program data storage area indicated by the FMPDR parameter in the programming processing is within the flash memory area, an error will occur. Therefore, temporarily transfer the program data to on-chip RAM to change the address set in FMPDR to an address other than flash memory.

Based on these conditions, tables 23.29 and 23.30 show the areas in which the program data can be stored and executed according to the operation type and mode.

**Table 22.29 Executable MAT**

Operation	Initiated Mode	
	User Program Mode	User Boot Mode*
Programming	Table 23.30 (1)	Table 23.30 (3)
Erasing	Table 23.30 (2)	Table 23.30 (4)

Note: \*Programming/Erasing is possible to user MATs.

**Table 23.30 (1) Usable Area for Programming in User Program Mode**

Item	Storable /Executable Area			Selected MAT	
	On-Chip RAM	User MAT	External Space (Expanded Mode with MD0 = 0)	User MAT	Embedded Program Storage MAT
Program data storage area	O	X*	O	—	—
Selecting on-chip program to be downloaded	O	O	O	O	
Writing H'A5 to key register	O	O	O	O	
Writing 1 to SCO in FCCS (download)	O	X	X		O
Key register clearing	O	O	O	O	
Judging download result	O	O	O	O	
Download error processing	O	O	O	O	
Setting initialization parameters	O	O	O	O	
Initialization	O	X	X	O	
Judging initialization result	O	O	O	O	
Initialization error processing	O	O	O	O	
Interrupt processing routine	O	X	O	O	
Writing H'5A to key register	O	O	O	O	
Setting programming parameters	O	X	O	O	
Programming	O	X	X	O	
Judging programming result	O	X	O	O	
Programming error processing	O	X	O	O	
Key register clearing	O	X	O	O	

Pro-gram-ming pro-ce-dure

Note: \* If the data has been transferred to on-chip RAM in advance, this area can be used.



**Table 23.30 (2) Usable Area for Erasure in User Program Mode**

Item	Storable /Executable Area			Selected MAT	
	On-Chip RAM	User MAT	External Space (Expanded Mode with MD0 = 0)	User MAT	Embedded Program Storage MAT
Selecting on-chip program to be downloaded	O	O	O	O	
Writing H'A5 to key register	O	O	O	O	
Writing 1 to SCO in FCCS (download)	O	X	X		O
Key register clearing	O	O	O	O	
Judging download result	O	O	O	O	
Download error processing	O	O	O	O	
Setting initialization parameters	O	O	O	O	
Initialization	O	X	X	O	
Judging initialization result	O	O	O	O	
Initialization error processing	O	O	O	O	
Interrupt processing routine	O	X	O	O	
Writing H'5A to key register	O	O	O	O	
Setting erasure parameters	O	X	O	O	
Erasure	O	X	X	O	
Judging erasure result	O	X	O	O	
Erasing error processing	O	X	O	O	
Key register clearing	O	X	O	O	

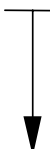
Erasing procedure

**Table 23.30 (3) Usable Area for Programming in User Boot Mode**

Item	Storable/Executable Area			Selected MAT		
	On-Chip RAM	User MAT	External Space (Expanded Mode with MD0 = 0)	User MAT	User Boot Mat	Embedded Program Storage Area
Program data storage area	O	X* <sup>1</sup>	O	—	—	—
Selecting on-chip program to be downloaded	O	O	O		O	
Writing H'A5 to key register	O	O	O		O	
Writing 1 to SCO in FCCS (download)	O	X	X			O
Key register clearing	O	O	O		O	
Judging download result	O	O	O		O	
Download error processing	O	O	O		O	
Setting initialization parameters	O	O	O		O	
Initialization	O	X	X		O	
Judging initialization result	O	O	O		O	
Initialization error processing	O	O	O		O	
Interrupt processing routine	O	X	O		O	
Switching MATs by FMATS	O	X	X	O		
Writing H'5A to Key Register	O	X	O	O		

Pro-gram-ming pro-ce-dure

**Table 23.30 (3) Usable Area for Programming in User Boot Mode (cont)**

	Item	Storable/Executable Area			Selected MAT		
		On-Chip RAM	User MAT	External Space (Expanded Mode with MD0 = 0)	User MAT	User Boot Mat	Embedded Program Storage Area
Programming procedure 	Setting programming parameters	O	X	O	O		
	Programming	O	X	X	O		
	Judging programming result	O	X	O	O		
	Programming error processing	O	X* <sup>2</sup>	O	O		
	Key register clearing	O	X	O	O		
	Switching MATs by FMATS	O	X	X		O	

- Notes
1. If the data has been transferred to on-chip RAM in advance, this area can be used.
  2. If the MATs have been switched by FMATS in on-chip RAM, this MAT can be used.

**Table 23.30 (4) Usable Area for Erasure in User Boot Mode**

Item	Storable/Executable Area			Selected MAT		
	On-Chip RAM	User MAT	External Space (Expanded Mode with MD0 = 0)	User MAT	User Boot Mat	Embedded Program Storage Area
Selecting on-chip program to be downloaded	O	O	O		O	
Writing H'A5 to key register	O	O	O		O	
Writing 1 to SCO in FCCS (download)	O	X	X			O
Key register clearing	O	O	O		O	
Judging download result	O	O	O		O	
Download error processing	O	O	O		O	
Setting initialization parameters	O	O	O		O	
Initialization	O	X	X		O	
Judging initialization result	O	O	O		O	
Initialization error processing	O	O	O		O	
Interrupt processing routine	O	X	O		O	
Switching MATs by FMATS	O	X	X		O	
Writing H'5A to key register	O	X	O	O		
Setting erasure parameters	O	X	O	O		

Erasing procedure

**Table 23.30 (4) Usable Area for Erasure in User Boot Mode (cont)**

	Item	Storable/Executable Area			Selected MAT	
		On-Chip RAM	User MAT	External Space (Expanded Mode with MD0 = 0)	User MAT	User Boot Mat
Erasing procedure	Erasure	O	X	X	O	
	Judging erasure result	O	X	O	O	
	Erasing error processing	O	X	O	O	
	Key register clearing	O	X	O	O	
	Switching MATs by FMATS	O	X	X		O

Note: If the MATs have been switched by FMATS in on-chip RAM, this MAT can be used.

# Section 24 RAM

## 24.1 Overview

The SH7058 has 48 kbytes of on-chip RAM. The on-chip RAM is linked to the CPU, direct memory access controller (DMAC), and advanced user debugger (AUD) with a 32-bit data bus (figure 24.1).

The CPU, DMAC, and AUD can access data in the on-chip RAM in 8, 16, or 32 bit widths. On-chip RAM data can always be accessed in one cycle for a read and two states for a write, making the RAM ideal for use as a program area, stack area, or data area, which require high-speed access. The contents of the on-chip RAM are held in both the sleep and software standby modes. When the RAME bit (see below) is cleared to 0, the on-chip RAM contents are also held in hardware standby mode.

The on-chip RAM is allocated to addresses H'FFFF0000 to H'FFFFBFFF.

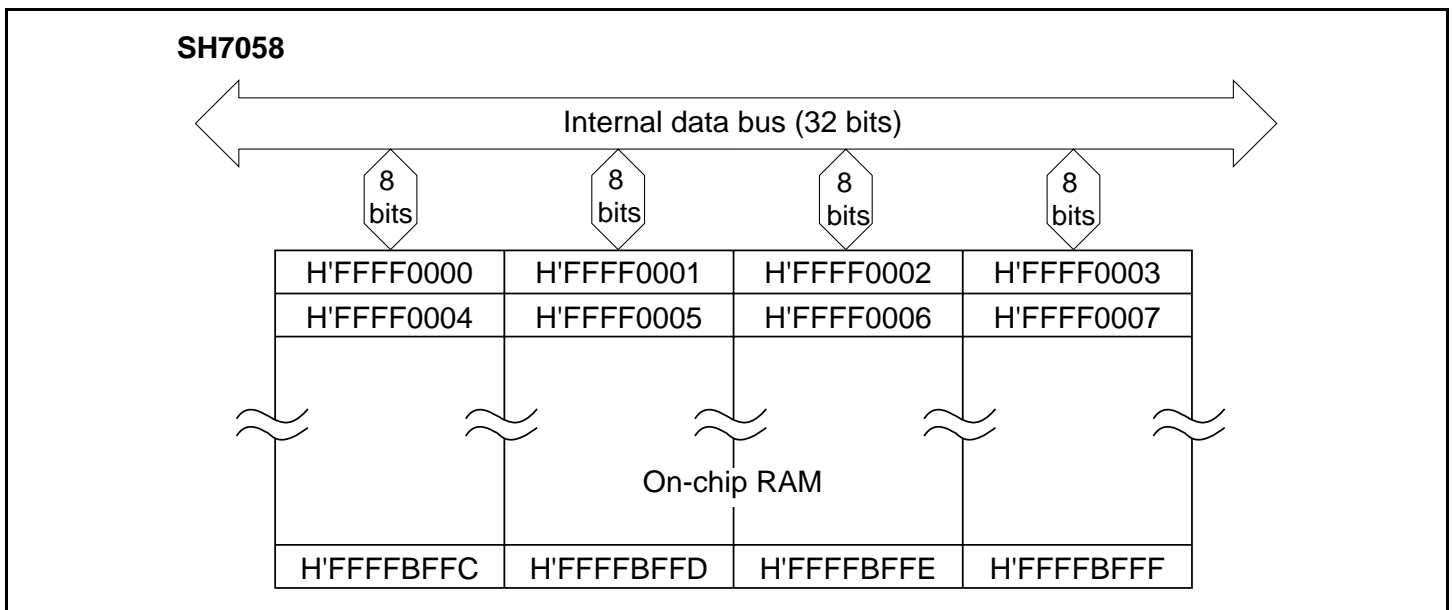


Figure 24.1 Block Diagram of RAM

## 24.2 Operation

The on-chip RAM is controlled by means of the system control register (SYSCR).

When the RAME bit in SYSCR is set to 1, the on-chip RAM is enabled. Accesses to addresses H'FFFF0000–H'FFFFBFFF are then directed to the on-chip RAM.

When the RAME bit in SYSCR is cleared to 0, the on-chip RAM is not accessed. A read will return an undefined value, and a write is invalid. If a transition is made to hardware standby mode after the RAME bit in SYSCR is cleared to 0, the contents of the on-chip RAM are held.

For details of SYSCR, see section 25.2.2, System Control Register1 (SYSCR1), in section 25, Power-Down State.

# Section 25 Power-Down State

## 25.1 Overview

Three modes are provided as power-save modes, namely, the hardware standby, software standby and sleep modes. Also, a module standby function is available to stop some modules. These standby modes can be selected depending on applications to reduce the power consumption of the SH7058.

### 25.1.1 Power-Down States

The power-down state is effected by the following modes:

1. Hardware standby mode

A transition to hardware standby mode is made according to the input level of the  $\overline{\text{RES}}$  and  $\overline{\text{HSTBY}}$  pins.

In hardware standby mode, all SH7058 functions are halted.

This state is exited by means of a power-on reset.

2. Software standby mode

A transition to software standby mode is made by means of software (a CPU instruction).

In software standby mode, all SH7058 functions are halted.

This state is exited by means of a power-on reset or an NMI interrupt.

3. Sleep mode

A transition to sleep mode is made by means of a CPU instruction.

In software standby mode, basically only the CPU is halted, and all on-chip peripheral modules operate.

This state is exited by means of a power-on reset, a manual reset, interrupt, or DMA address error.

4. Module standby mode

Operation of the on-chip peripheral modules\* which can be placed in a standby mode can be stopped by stopping the clock supply. Clock supply to the individual modules can be controlled by setting bits in system control register 2 (SYSCR2).

Note: \* AUD, H-UDI, FPU, and UBC



Table 25.1 shows the transition conditions for entering the modes from the program execution state, as well as the CPU and peripheral module status in each mode and the procedures for canceling each mode.

**Table 25.1 Power-Down State Conditions**

Mode	Entering Procedure	State						Canceling Procedure
		Clock	CPU	CPU Registers	On-Chip Peripheral Modules	RAM	Pins	
Hardware standby	Low-level input at $\overline{\text{HSTBY}}$ pin	Halted	Halted	Undefined	Halted	Held <sup>*2</sup>	Initialized	High-level input at $\overline{\text{HSTBY}}$ pin, executing power-on reset
Software standby	Execute SLEEP instruction with SSBY bit set to 1 in SBYCR	Halted	Halted	Held	Halted <sup>*1</sup>	Held	Held or high impedance <sup>*3</sup>	<ul style="list-style-type: none"> <li>• NMI interrupt</li> <li>• Power-on reset</li> </ul>
Sleep	Execute SLEEP instruction with SSBY bit cleared to 0 in SBYCR	Runs	Halted	Held	Runs	Held	Held	<ul style="list-style-type: none"> <li>• Interrupt</li> <li>• DMA address error</li> <li>• Power-on reset</li> <li>• Manual reset</li> </ul>

Notes: SBYCR: Standby control register

SSBY: Software standby bit

1. Some bits within on-chip peripheral module registers are initialized in software standby mode, and some are not. Refer to the register descriptions for each peripheral module.
2. Clear the RAME bit in SYSCR1 to 0 in advance when changing the state from the program execution state in hardware standby mode.
3. The state of the I/O ports in standby mode is set by the port high impedance bit (HIZ) in SBYCR. See section 25.2.1, Standby Control Register (SBYCR).

## 25.1.2 Pin Configuration

Pins related to power-down modes are shown in table 25.2.

**Table 25.2 Pin Configuration**

Pin Name	Abbreviation	I/O	Function
Hardware standby input pin	$\overline{\text{HSTBY}}$	Input	Input level determines transition to hardware standby mode
Power-on reset input pin	$\overline{\text{RES}}$	Input	Power-on reset signal input pin

## 25.1.3 Related Registers

Table 25.3 shows the registers used for power-down state control.

**Table 25.3 Related Registers**

Name	Abbreviation	R/W	Initial Value	Address		Access Size
				Write	Read	
Standby control register	SBYCR* <sup>1</sup>	R/W	H'1F		H'FFFFFFE14	8
System control register 1	SYSCR1* <sup>1</sup>	R/W	H'01		H'FFFFFF708	8
System control register 2	SYSCR2* <sup>1</sup>	R/W	H'01	H'FFFFFF70A* <sup>2</sup>	H'FFFFFF70B* <sup>3</sup>	8, 16

- Notes:
1. Register access with an internal clock multiplication ratio of 4 requires four internal clock ( $\phi$ ) cycles for SBYCR, and four or five internal clock ( $\phi$ ) cycles for SYSCR1 and SYSCR2.
  2. Write data in words. Data cannot be written in bytes or longwords.
  3. Read data in bytes. Values cannot be read correctly if data is read in words or longwords.

## 25.2 Register Descriptions

### 25.2.1 Standby Control Register (SBYCR)

The standby control register (SBYCR) is an 8-bit readable/writable register that sets the transition to standby mode, and the port state in standby mode. SBYCR is initialized to H'1F by a power-on reset.

Bit:	7	6	5	4	3	2	1	0
	SSBY	HIZ	—	—	—	—	—	—
Initial value:	0	0	0	1	1	1	1	1
R/W:	R/W	R/W	R	R	R	R	R	R

- Bit 7—Software Standby (SSBY): Specifies transition to software standby mode. The SSBY bit cannot be set to 1 while the watchdog timer is running (when the timer enable bit (TME) in the WDT timer control/status register (TCSR) is set to 1). To enter software standby mode, always halt the WDT by clearing the TME bit to 0, then set the SSBY bit.

Bit 7: SSBY	Description
0	Executing SLEEP instruction puts the SH7058 into sleep mode (Initial value)
1	Executing SLEEP instruction puts the SH7058 into standby mode

- Bit 6—Port High Impedance (HIZ): In software standby mode, this bit selects whether to set I/O port pins to high impedance or hold the pin state. The HIZ bit cannot be set to 1 when the TME bit in the WDT timer control/status register (TCSR) is set to 1. When making the I/O port pin state high impedance, always clear the TME bit to 0 before setting the HIZ bit.

Bit 6: HIZ	Description
0	Pin states held in software standby mode (Initial value)
1	Pins go to high impedance in software standby mode

- Bit 5—Reserved: This bit is always read as 0. The write value should always be 0.
- Bits 4 to 0—Reserved: These bits are always read as 1. The write value should always be 1.

### 25.2.2 System Control Register 1 (SYSCR1)

Bit:	7	6	5	4	3	2	1	0
	OSCSTOP	INOSCE	—	—	—	—	AUDSRST	RAME
Initial value:	—	0	0	0	0	0	0	1
R/W:	R	R/W	R	R	R	R	R/W	R/W

System control register 1 (SYSCR1) is an 8-bit readable/writable register that performs AUD software reset control and enables or disables access to the on-chip RAM.

SYSCR1 is initialized to H'01 by a power-on reset (at the rising edge).

- Bits 7 and 6: Refer to section 5.4, Precautions for Performing Crystal Resonator Stoppage Detection Function.

- Bits 5 to 2—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit1— AUD Software Reset (AUDSRST): This bit controls AUD reset using software. Setting AUDSRST bit to 1 places the AUD module in the power-on reset state.

**Bit 1: AUDSRST Description**

0	AUD reset state cleared	
1	AUD reset state entered	(Initial value)

- Bit 0—RAME Enable (RAME): Selects enabling or disabling of the on-chip RAM. When RAME is set to 1, on-chip RAM is enabled. When RAME is cleared to 0, on-chip RAM cannot be accessed. In this case, a read or instruction fetch from on-chip RAM will return an undefined value, and a write to on-chip RAM will be ignored. The initial value of RAME is 1.

When on-chip RAM is disabled by clearing RAME to 0, do not place an instruction that attempts to access on-chip RAM immediately after the SYSCR1 write instruction, as normal access cannot be guaranteed in this case.

When on-chip RAM is enabled by setting RAME to 1, place an SYSCR1 read instruction immediately after the SYSCR1 write instruction. Normal access cannot be guaranteed if an on-chip RAM access instruction is placed immediately after the SYSCR1 write instruction.

**Bit 0: RAME Description**

0	On-chip RAM disabled	
1	On-chip RAM enabled	(Initial value)

**25.2.3 System Control Register 2 (SYSCR2)**

Bit:	7	6	5	4	3	2	1	0
	CKSEL	—	—	—	MSTOP3	MSTOP2	MSTOP1	MSTOP0
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R	R	R	R/W	R/W	R/W	R/W

System control register 2 (SYSCR2) is an 8-bit readable/writable register that controls the standby state of the AUD, H-UDI, FPU, and UBC on-chip modules.

SYSCR2 is initialized to H'01 by a power-on reset.

Note: The method of writing to SYSCR2 is different from that of ordinary registers to prevent inadvertent rewriting. See section 25.2.4, Notes on Register Access, for more information.

- Bit 7—Internal Clock ( $\phi$ ) Select (CKSEL): See section 5, Clock Pulse Generator (CPG).

- Bits 6 to 4—Reserved: These bits are always read as 0. The write value should always be 0.
- Bit 3—Module Stop 3 (MSTOP3): Specifies halting of the clock supply to the AUD on-chip peripheral module. Setting the MSTOP3 bit to 1 stops the clock supply to the AUD. To cancel halting of the clock supply to the AUD, first set the AUD software reset bit (AUDSRST) in the system control register 1 (SYSCR1) to the AUD reset state value. Use of the AUD will then be enabled by clearing the AUD reset.

Bit 3: MSTOP3	Description	
0	AUD operates	(Initial value)
1	Clock supply to AUD stopped	

- Bit 2—Module Stop 2 (MSTOP2): Specifies halting of the clock supply to the H-UDI on-chip peripheral module. Setting the MSTOP2 bit to 1 stops the clock supply to the H-UDI.

Bit 2: MSTOP2	Description	
0	H-UDI operates	
1	Clock supply to H-UDI stopped	(Initial value)

- Bit 1—Module Stop 1 (MSTOP1): Specifies halting of the clock supply to the FPU on-chip peripheral module. Setting the MSTOP1 bit to 1 stops the clock supply to the FPU.  
The MSTOP1 bit cannot be cleared by writing 0 after it has been set to 1. In other words, once the MSTOP1 bit has been set to 1 and the clock supply to the FPU has been stopped, the clock supply to the FPU cannot be resumed by clearing the MSTOP1 bit to 0.  
An SH7058 power-on reset is necessary to restart the FPU clock supply after it has been stopped.

Bit 1: MSTOP1	Description	
0	FPU operates	(Initial value)
1	Clock supply to FPU stopped	

- Bit 0—Module Stop 0 (MSTOP0): Specifies halting of the clock supply to the UBC on-chip peripheral module.  
Clearing the MSTOP0 bit to 0 starts the clock supply to the UBC.  
Stopping clock supply to the UBC will reset the internal state of the UBC including its registers.

Bit 0: MSTOP0	Description	
0	UBC operates	
1	Clock supply to UBC stopped	(Initial value)

## 25.2.4 Notes on Register Access

The method of writing to system control register 2 (SYSCR2) is different from that of ordinary registers to prevent inadvertent rewriting.

Be certain to use a word transfer instruction when writing data to SYSCR2. Data cannot be written by a byte transfer instruction. As shown in figure 25.1, set the upper byte to H'3C and transfer data using the lower byte as write data.

Data can be read by the same method as for ordinary registers.

SYSCR2 is allocated to address H'FFFFFF70A. Always use a byte transfer instruction to read data.

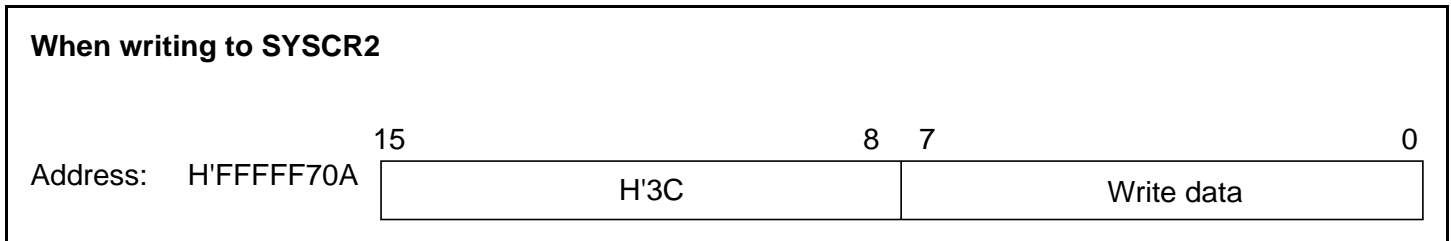


Figure 25.1 Writing to SYSCR2

## 25.3 Hardware Standby Mode

### 25.3.1 Transition to Hardware Standby Mode

The chip enters hardware standby mode when the  $\overline{\text{HSTBY}}$  and  $\overline{\text{RES}}$  pins go low. The mode pin should be set according to the pin settings described in section 4, Operating Modes. If other settings are applied to the mode pin, operation cannot be guaranteed. Hardware standby mode reduces power consumption drastically by halting all SH7058 functions. As the transition to hardware standby mode is made by means of external pin input, the transition is made asynchronously, regardless of the current state of the SH7058, and therefore the chip state prior to the transition is not preserved. However, on-chip RAM data is retained as long as the specified voltage is supplied. To retain on-chip RAM data, clear the RAM enable bit (RAME) to 0 in the system control register 1 (SYSCR1) before driving the  $\overline{\text{HSTBY}}$  pin low.

### 25.3.2 Canceling Hardware Standby Mode

Hardware standby mode is canceled by means of the  $\overline{\text{HSTBY}}$  pin and  $\overline{\text{RES}}$  pin. When  $\overline{\text{HSTBY}}$  is driven high while  $\overline{\text{RES}}$  is low, the clock oscillator starts running. The  $\overline{\text{RES}}$  pin should be held low long enough for clock oscillation to stabilize. When  $\overline{\text{RES}}$  is driven high, power-on reset exception processing is started and a transition is made to the program execution state.

### 25.3.3 Hardware Standby Mode Timing

Figure 25.2 shows sample pin timings for hardware standby mode. A transition to hardware standby mode is made by driving the  $\overline{\text{HSTBY}}$  pin low after driving the  $\overline{\text{RES}}$  pin low. Hardware standby mode is canceled by driving  $\overline{\text{HSTBY}}$  high, waiting for clock oscillation to stabilize, then switching  $\overline{\text{RES}}$  from low to high.

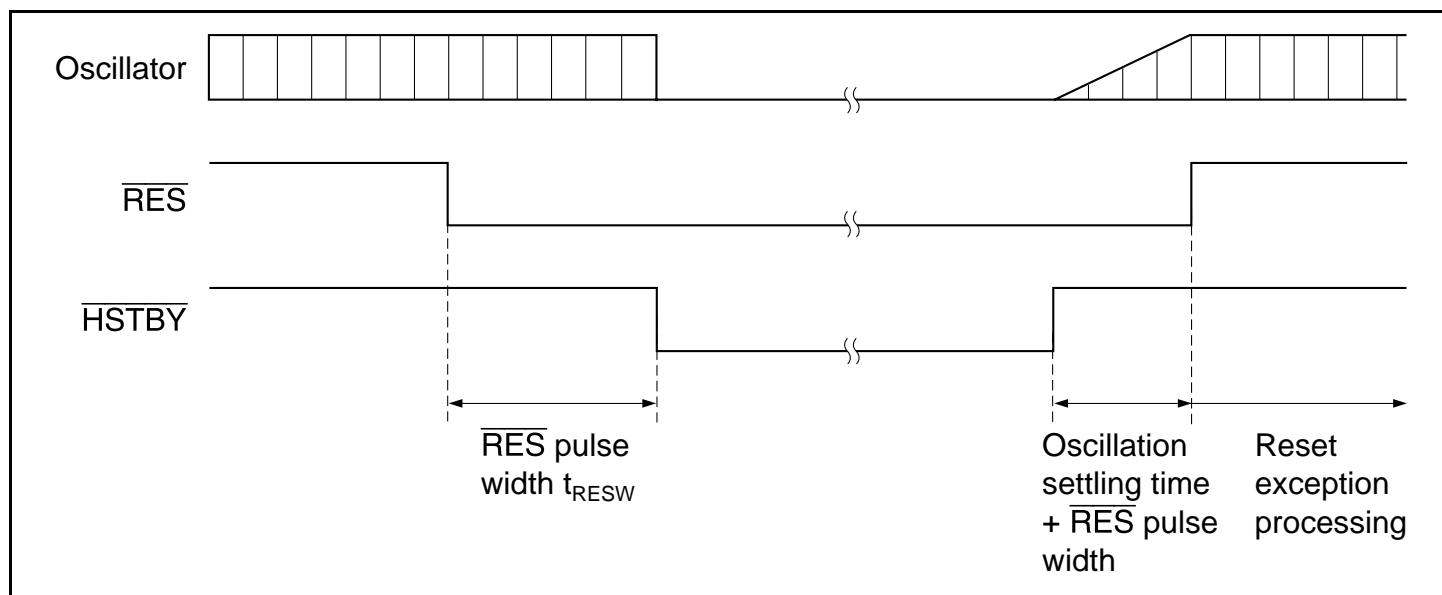


Figure 25.2 Hardware Standby Mode Timing

## 25.4 Software Standby Mode

### 25.4.1 Transition to Software Standby Mode

To enter software standby mode, set the software standby bit (SSBY) to 1 in SBYCR, then execute the SLEEP instruction. The SH7058 switches from the program execution state to software standby mode. In software standby mode, power consumption is greatly reduced by halting not only the CPU, but the clock and on-chip peripheral modules as well. CPU register contents and on-chip RAM data are held as long as the prescribed voltages are applied (when the RAME bit in SYSCR1 is 0). The register contents of some on-chip peripheral modules are initialized, but some are not. The I/O port state can be selected as held or high impedance by the port high impedance bit (HIZ) in SBYCR.

### 25.4.2 Canceling Software Standby Mode

Software standby mode is canceled by an NMI interrupt or a power-on reset.

**Cancellation by NMI:** Clock oscillation starts when a rising edge or falling edge (selected by the NMI edge select bit (NMIE) in the interrupt control register (ICR) of the INTC) is detected in the

NMI signal. This clock is supplied only to the oscillation settling counter which counts the oscillation stabilizing time.

Counting the oscillation settling time by the oscillation settling counter is used to indicate that the clock has stabilized, so the clock is supplied to the entire chip, software standby mode is canceled, and NMI exception processing begins.

The oscillation settling counter overflows when it counts  $2^{14}=16384$  with the input clock frequency. Since the frequency of this counting clock is unstable until the PLL multiplication circuit is locked in, the absolute time is not fixed, and the CK pin signal output is in the high level for the meantime.

When canceling standby mode with an NMI pin set for falling edge, be sure that the NMI pin level upon entering software standby (when the clock is halted) is high, and that the NMI pin level upon returning from software standby (when the clock starts after oscillation stabilization) is low. When canceling software standby mode with an NMI pin set for rising edge, be sure that the NMI pin level upon entering software standby (when the clock is halted) is low, and that the NMI pin level upon returning from software standby (when the clock starts after oscillation stabilization) is high.

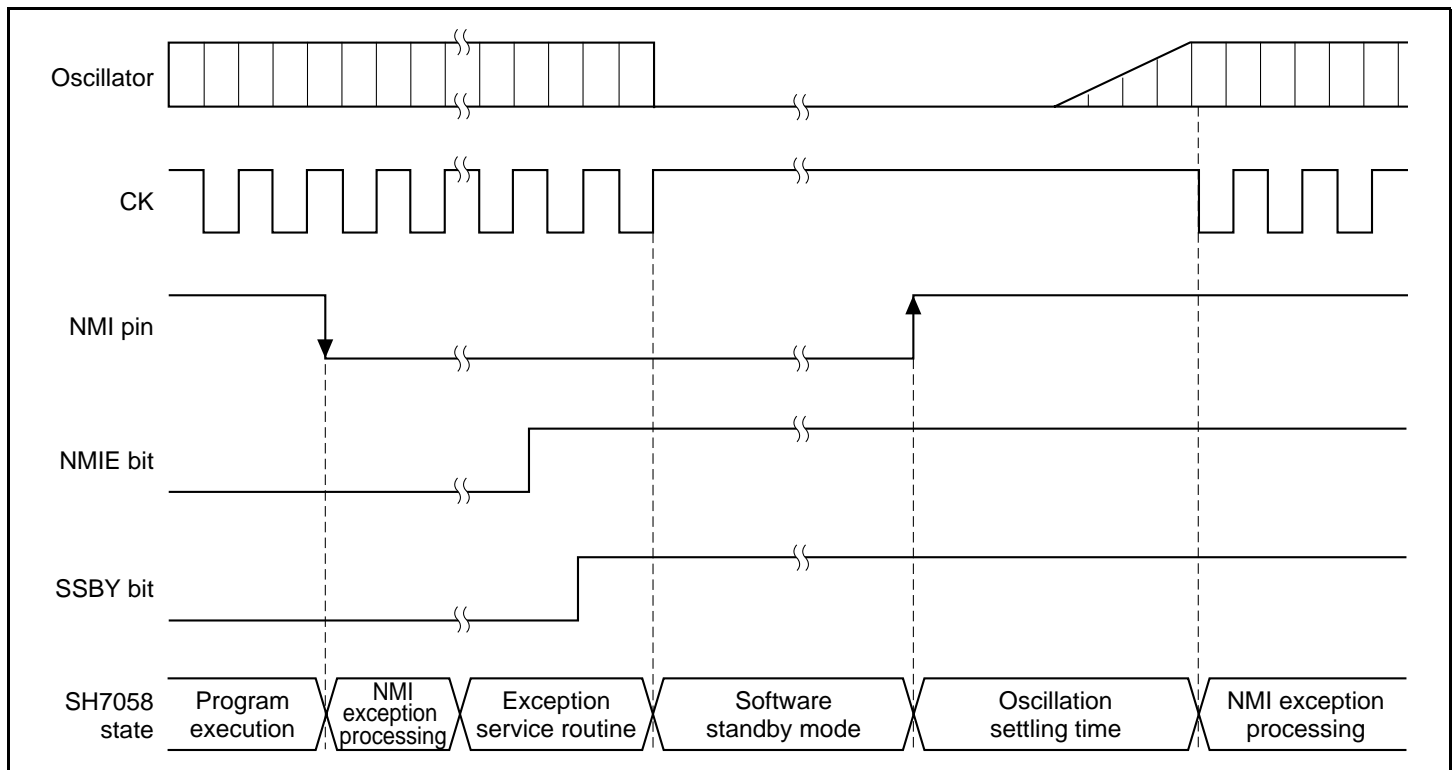
**Cancellation by Power-On Reset:** A power-on reset of the SH7058 caused by driving the  $\overline{\text{RES}}$  pin low cancels software standby mode.



### 25.4.3 Software Standby Mode Application Example

This example describes a transition to software standby mode on the falling edge of the NMI signal, and cancellation on the rising edge of the NMI signal. The timing is shown in figure 25.3.

When the NMI pin is changed from high to low level while the NMI edge select bit (NMIE) in ICR is set to 0 (falling edge detection), the NMI interrupt is accepted. When the NMIE bit is set to 1 (rising edge detection) by the NMI exception service routine, the software standby bit (SSBY) in SBYCR is set to 1, and a SLEEP instruction is executed, software standby mode is entered. Thereafter, software standby mode is canceled when the NMI pin is changed from low to high level.



**Figure 25.3 Software Standby Mode NMI Timing (Application Example)**

## 25.5 Sleep Mode

### 25.5.1 Transition to Sleep Mode

Executing the SLEEP instruction after the software standby bit (SSBY) in SBYCR has been cleared to 0 causes a transition from the program execution state to sleep mode. Although the CPU halts immediately after executing the SLEEP instruction, the contents of its internal registers remain unchanged. The on-chip peripheral modules continue to run during sleep mode.

### 25.5.2 Canceling Sleep Mode

**Cancellation by Interrupt:** When an interrupt occurs, sleep mode is canceled and interrupt exception processing is executed. The sleep mode is not canceled if the interrupt cannot be accepted because its priority level is equal to or less than the mask level set in the CPU's status register (SR) or if an interrupt by an on-chip peripheral module is disabled by the peripheral module.

**Cancellation by DMA Address Error:** If a DMA address error occurs, sleep mode is canceled and DMA address error exception processing is executed.

**Cancellation by Manual Reset:** When an internal manual reset is triggered by the WDT and the CPU acquires the bus during the internal manual reset period, the state of the SH7058 changes to the manual reset state and sleep mode will be released.

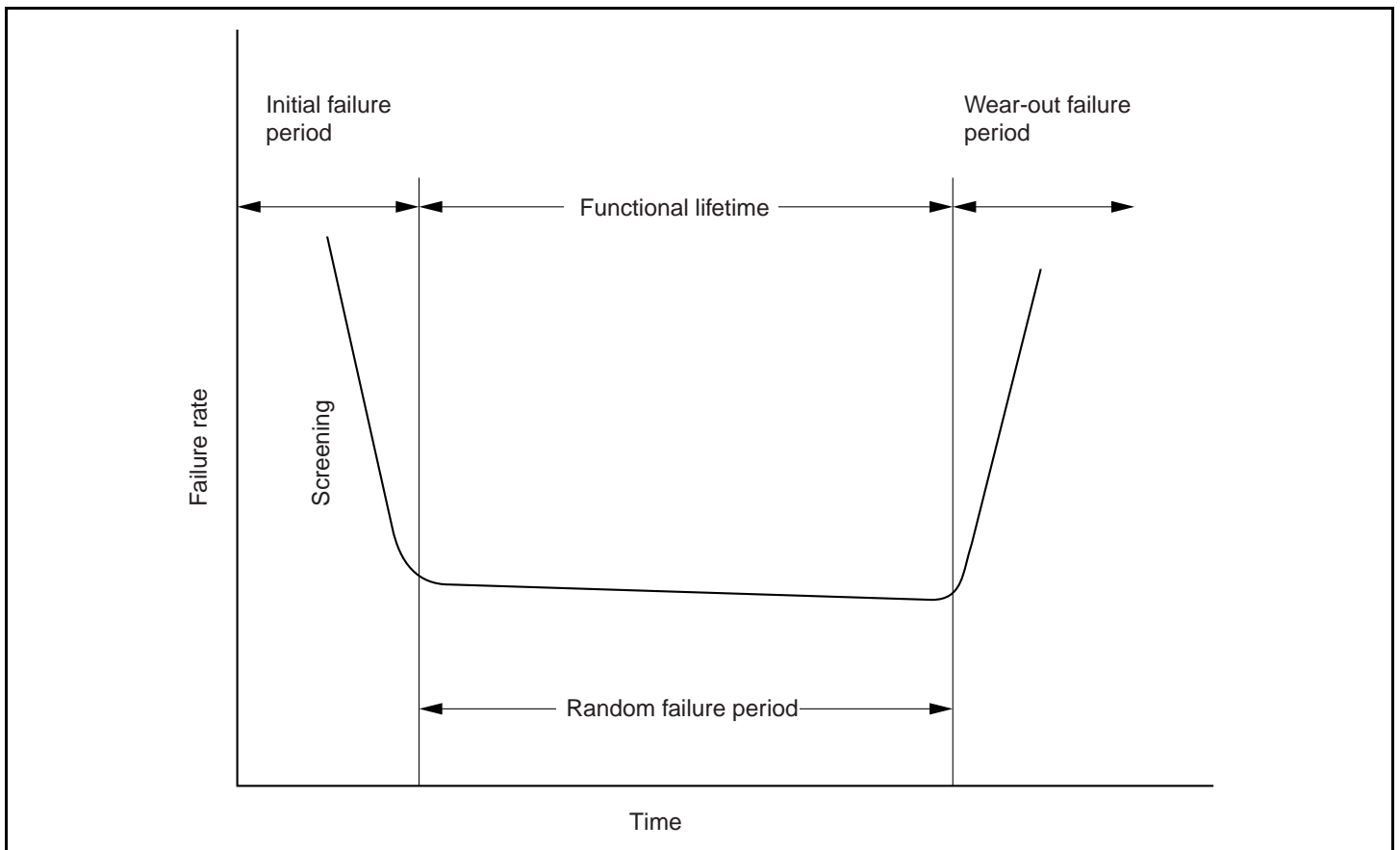
**Cancellation by Power-On Reset:** A power-on reset of the SH7058 resulting from driving the  $\overline{\text{RES}}$  pin low, or caused by the WDT, cancels sleep mode.



# Section 26 Reliability

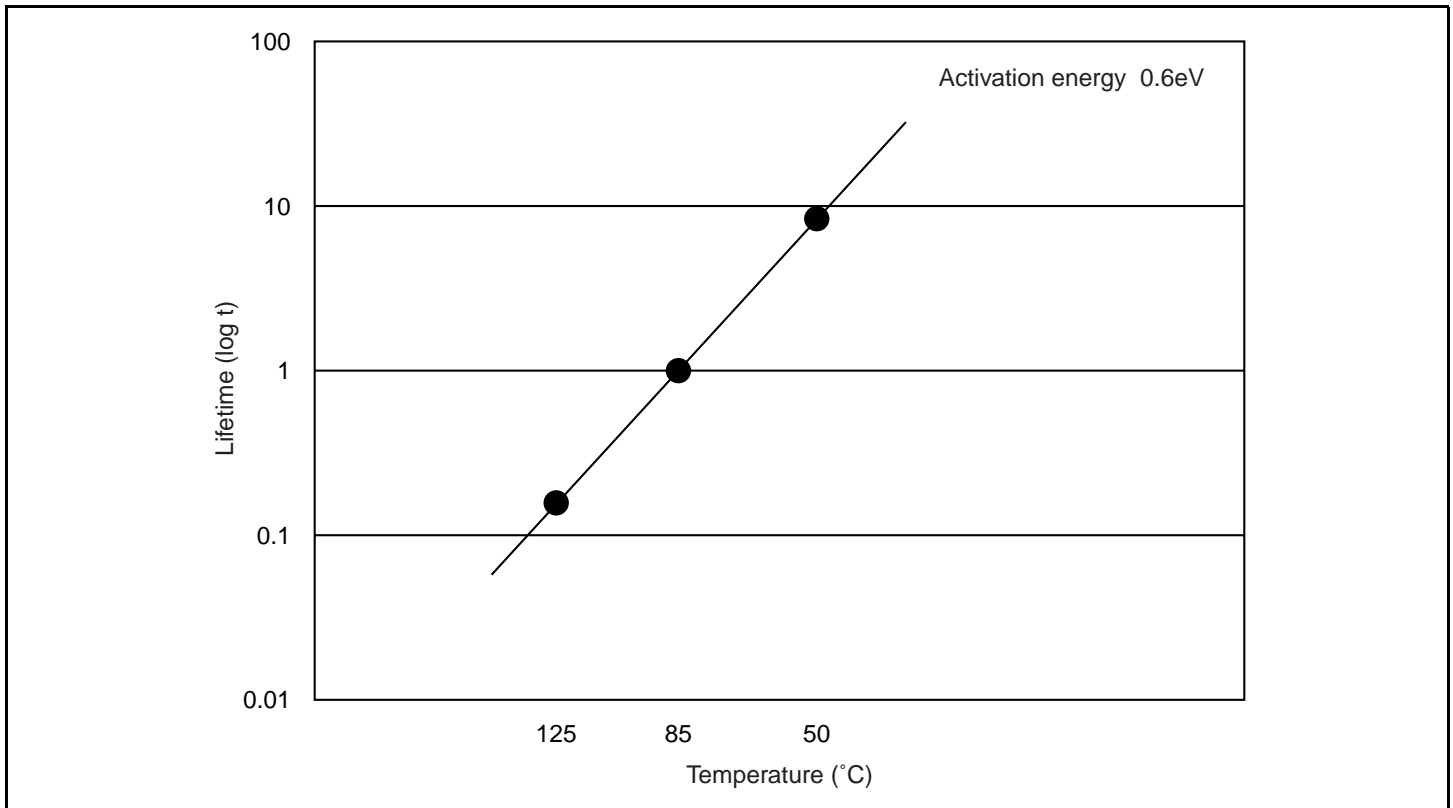
## 26.1 Reliability

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 26.1. 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.



**Figure 26.1 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  $-40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ , 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  $85^{\circ}\text{C}$  leads to failure within a short time, since high temperatures induce failures in semiconductor devices. Figure 26.2 shows the temperature dependence of semiconductor device lifetimes. The type of failure in this figure is a wear-out failure, i.e. the dielectric breakdown of oxide film. According to figure 26.2, the life at  $125^{\circ}\text{C}$  is 1/10 of life at  $85^{\circ}\text{C}$ , and operation at the higher temperature leads to a correspondingly higher probability of a failure in the field. Therefore, the reliability of operation at a temperature in excess of  $85^{\circ}\text{C}$  is checked on the assumption that the period of operation at the upper-limit temperature of the range for guaranteed operation is 3000 hours.



**Figure 26.2 Temperature Reliability of Dielectric Breakdown of Oxide Film**

# Section 27 Electrical Characteristics

## 27.1 Absolute Maximum Ratings

Table 27.1 shows the absolute maximum ratings.

**Table 27.1 Absolute Maximum Ratings**

Item		Symbol	Rating	Unit	Remarks
Power supply voltage*	V <sub>CC</sub> and PLLV <sub>CC</sub> pins	V <sub>CC</sub>	-0.3 to +4.3	V	The PLLCAP, EXTAL, XTAL, CK, and H-UDI pins are concerned. (V <sub>CC</sub> and PLLV <sub>CC</sub> are the same voltage)
	PV <sub>CC1</sub> and PV <sub>CC2</sub> pins	PV <sub>CC</sub>	-0.3 to + 6.5	V	Except for the PLLCAP, EXTAL, XTAL, CK, and H-UDI pins and the analog input pin
Input voltage	EXTAL and H-UDI pins	V <sub>in</sub>	-0.3 to V <sub>CC</sub> + 0.3	V	
	All pins other than analog input, EXTAL, and H-UDI pins	V <sub>in</sub>	-0.3 to PV <sub>CC</sub> + 0.3	V	Refer to table 27.2, Correspondence between Power Supply Names and Pins
Analog supply voltage		AV <sub>CC</sub>	-0.3 to +7.0	V	
Analog reference voltage		AV <sub>ref</sub>	-0.3 to AV <sub>CC</sub> + 0.3	V	
Analog input voltage		V <sub>AN</sub>	-0.3 to AV <sub>CC</sub> + 0.3	V	
Operating temperature (except writing or erasing flash memory)		Topr	-40 to + 125	°C	
Operating temperature (writing or erasing flash memory)		TWEopr	-40 to +85	°C	
Storage temperature		Tstg	-55 to +125	°C	

### [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 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.

Note: \* Do not apply any power supply voltage to the  $V_{CL}$  pin. Connect to GND through an external capacitor (0.33 to 0.47  $\mu$ F).

## 27.2 DC Characteristics

Table 27.2 shows the correspondence between power supply names and pins.

Table 27.4 shows DC characteristics.

**Table 27.2 Correspondence between Power Supply Names and Pins**

Pin No. (FP-256H)	Power Supply Pin	Dedicated Pin	User Pin				Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
			Function 1	Function 2	Function 3	Function 4			
1			PD8	PULS0			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
2			PD9	PULS1			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
3			PD10	PULS2			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
4			PD11	PULS3			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
5			PD12	PULS4			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
6			PD13	PULS6	HTxD0	HTxD1	PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
7			PE0	A0			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
8			PE1	A1			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
9			PE2	A2			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
10			PE3	A3			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
11	V <sub>cc</sub>								
12			PE4	A4			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
13	V <sub>ss</sub>								
14			PE5	A5			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
15			PE6	A6			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
16			PE7	A7			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
17			PE8	A8			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
18			PE9	A9			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
19			PE10	A10			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
20	PV <sub>cc</sub> 1								
21			PE11	A11			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
22	V <sub>ss</sub>								
23			PE12	A12			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
24			PE13	A13			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
25			PE14	A14			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	



**Table 27.2 Correspondence between Power Supply Names and Pins (cont)**

Pin No. (FP-256H)	Power Supply Pin	Dedicated Pin	User Pin				Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
			Function 1	Function 2	Function 3	Function 4			
26			PE15	A15			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
27			PF0	A16			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
28			PF1	A17			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
29			PF2	A18			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
30	V <sub>CL</sub>								
31			PF3	A19			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
32	V <sub>SS</sub>								
33			PF4	A20			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
34			PF5	A21	$\overline{\text{POD}}$		PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
35			PF6	$\overline{\text{WRL}}$			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
36			PF7	$\overline{\text{WRH}}$			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
37			PF8	$\overline{\text{WAIT}}$			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
38			PF9	$\overline{\text{RD}}$			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
39	PV <sub>cc</sub> 1								
40			PF10	$\overline{\text{CS0}}$			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
41	V <sub>SS</sub>								
42			PF11	$\overline{\text{CS1}}$			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
43			PF12	$\overline{\text{CS2}}$			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
44			PF13	$\overline{\text{CS3}}$			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
45			PF14	$\overline{\text{BACK}}$			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
46			PF15	$\overline{\text{BREQ}}$			PV <sub>cc</sub> 1	PV <sub>cc</sub> 1+0.3	
47	V <sub>SS</sub>								
48			CK				V <sub>CC</sub>		
49	V <sub>CC</sub>								
50		MD2						5.5+0.3	
51		EXTAL						V <sub>CC</sub> +0.3	
52	V <sub>CC</sub>								
53		XTAL					V <sub>CC</sub>		
54	V <sub>SS</sub>								
55		MD1						5.5+0.3	

**Table 27.2 Correspondence between Power Supply Names and Pins (cont)**

Pin No. (FP-256H)	Power Supply Pin	Dedicated Pin	User Pin				Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
			Function 1	Function 2	Function 3	Function 4			
56		FWE						5.5+0.3	
57		$\overline{\text{HSTBY}}$						5.5+0.3	
58		$\overline{\text{RES}}$						5.5+0.3	
59		MD0						5.5+0.3	
60	PLL <sub>V<sub>CC</sub></sub>								
61		PLLCAP							
62	PLL <sub>V<sub>SS</sub></sub>								
63			PH0	D0			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
64			PH1	D1			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
65			PH2	D2			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
66			PH3	D3			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
67			PH4	D4			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
68			PH5	D5			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
69			PH6	D6			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
70	PV <sub>CC</sub> 1								
71			PH7	D7			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
72	V <sub>SS</sub>								
73			PH8	D8			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
74			PH9	D9			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
75	V <sub>CC</sub>								
76			PH10	D10			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
77	V <sub>SS</sub>								
78			PH11	D11			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
79			PH12	D12			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
80			PH13	D13			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
81			PH14	D14			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
82			PH15	D15			PV <sub>CC</sub> 1	PV <sub>CC</sub> 1+0.3	
83	PV <sub>CC</sub> 1								
84		NMI						5.5+0.3	
85	V <sub>SS</sub>								

**Table 27.2 Correspondence between Power Supply Names and Pins (cont)**

Pin No. (FP-256H)	Power Supply Pin	Dedicated Pin	User Pin				Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
			Function 1	Function 2	Function 3	Function 4			
86			AN0					$AV_{CC}+0.3$	
87			AN1					$AV_{CC}+0.3$	
88			AN2					$AV_{CC}+0.3$	
89			AN3					$AV_{CC}+0.3$	
90			AN4					$AV_{CC}+0.3$	
91			AN5					$AV_{CC}+0.3$	
92			AN6					$AV_{CC}+0.3$	
93			AN7					$AV_{CC}+0.3$	
94			AN8					$AV_{CC}+0.3$	
95			AN9					$AV_{CC}+0.3$	
96			AN10					$AV_{CC}+0.3$	
97			AN11					$AV_{CC}+0.3$	
98			AN12					$AV_{CC}+0.3$	
99	$AV_{SS}$								
100		AVref							
101	$AV_{CC}$								
102			AN13					$AV_{CC}+0.3$	
103			AN14					$AV_{CC}+0.3$	
104			AN15					$AV_{CC}+0.3$	
105			AN16					$AV_{CC}+0.3$	
106			AN17					$AV_{CC}+0.3$	
107			AN18					$AV_{CC}+0.3$	
108			AN19					$AV_{CC}+0.3$	
109			AN20					$AV_{CC}+0.3$	
110			AN21					$AV_{CC}+0.3$	
111			AN22					$AV_{CC}+0.3$	
112			AN23					$AV_{CC}+0.3$	
113			AN24					$AV_{CC}+0.3$	
114			AN25					$AV_{CC}+0.3$	
115			AN26					$AV_{CC}+0.3$	

**Table 27.2 Correspondence between Power Supply Names and Pins (cont)**

Pin No. (FP-256H)	Power Supply Pin	Dedicated Pin	User Pin				Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
			Function 1	Function 2	Function 3	Function 4			
116			AN27					$AV_{cc}+0.3$	
117			AN28					$AV_{cc}+0.3$	
118			AN29					$AV_{cc}+0.3$	
119	$AV_{cc}$								
120		AVref							
121	$AV_{ss}$								
122			AN30					$AV_{cc}+0.3$	
123			AN31					$AV_{cc}+0.3$	
124		WDTOVF					$PV_{cc2}$		
125			PA0	TIOA			$PV_{cc2}$	$PV_{cc2}+0.3$	Schmitt-trigger input pin
126	$V_{ss}$								
127			PA1	TIOB			$PV_{cc2}$	$PV_{cc2}+0.3$	Schmitt-trigger input pin
128	$PV_{cc2}$								
129			PA2	TIOC			$PV_{cc2}$	$PV_{cc2}+0.3$	Schmitt-trigger input pin
130			PA3	TIOD			$PV_{cc2}$	$PV_{cc2}+0.3$	
131			PA4	TIO3A			$PV_{cc2}$	$PV_{cc2}+0.3$	
132			PA5	TIO3B			$PV_{cc2}$	$PV_{cc2}+0.3$	
133			PA6	TIO3C			$PV_{cc2}$	$PV_{cc2}+0.3$	
134			PA7	TIO3D			$PV_{cc2}$	$PV_{cc2}+0.3$	
135			PA8	TIO4A			$PV_{cc2}$	$PV_{cc2}+0.3$	
136			PA9	TIO4B			$PV_{cc2}$	$PV_{cc2}+0.3$	
137			PA10	TIO4C			$PV_{cc2}$	$PV_{cc2}+0.3$	
138			PA11	TIO4D			$PV_{cc2}$	$PV_{cc2}+0.3$	
139	$V_{cc}$								
140			PA12	TIO5A			$PV_{cc2}$	$PV_{cc2}+0.3$	Schmitt-trigger input pin

**Table 27.2 Correspondence between Power Supply Names and Pins (cont)**

Pin No. (FP-256H)	Power Supply Pin	Dedicated Pin	User Pin				Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
			Function 1	Function 2	Function 3	Function 4			
141	V <sub>SS</sub>								
142			PA13	TIO5B			PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3 Schmitt-trigger input pin	
143			PA14	TxD0			PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
144			PA15	RxD0			PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
145			PB0	TO6A			PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
146			PB1	TO6B			PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
147			PB2	TO6C			PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
148	PV <sub>CC2</sub>								
149			PB3	TO6D			PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
150	V <sub>SS</sub>								
151			PB4	TO7A	TO8A		PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
152			PB5	TO7B	TO8B		PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
153			PB6	TO7C	TO8C		PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
154			PB7	TO7D	TO8D		PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
155			PB8	TxD3	TO8E		PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
156			PB9	RxD3	TO8F		PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
157			PB10	TxD4	HTxD0	TO8G	PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
158			PB11	RxD4	HRxD0	TO8H	PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
159			PB12	TCLKA	UBCTRG		PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3 Schmitt-trigger input pin	
160			PB13	SCK0			PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	
161	V <sub>CL</sub>								
162			PB14	SCK1	TCLKB	TI10	PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3 Schmitt-trigger input pin	
163	V <sub>SS</sub>								
164			PB15	PULS5	SCK2		PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3 Schmitt-trigger input pin	
165			PC0	TxD1			PV <sub>CC2</sub>	PV <sub>CC2</sub> +0.3	

**Table 27.2 Correspondence between Power Supply Names and Pins (cont)**

Pin No. (FP-256H)	Power Supply Pin	Dedicated Pin	User Pin				Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
			Function 1	Function 2	Function 3	Function 4			
166			PC1	RxD1			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
167			PC2	TxD2			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
168			PC3	RxD2			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
169			PC4	$\overline{\text{IRQ0}}$			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
170			PG0	PULS7	HRxD0	HRxD1	PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
171			PG1	$\overline{\text{IRQ1}}$			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
172	PV <sub>cc2</sub>								
173			PG2	$\overline{\text{IRQ2}}$	ADEND		PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
174	V <sub>ss</sub>								
175			PG3	$\overline{\text{IRQ3}}$	$\overline{\text{ADTRG0}}$		PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
176			PJ0	TIO2A			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
177			PJ1	TIO2B			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
178			PJ2	TIO2C			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
179			PJ3	TIO2D			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
180			PJ4	TIO2E			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
181			PJ5	TIO2F			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
182			PJ6	TIO2G			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
183			PJ7	TIO2H			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
184			PJ8	TIO5C			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
185	V <sub>ss</sub>								
186			PJ9	TIO5D			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
187	V <sub>cc</sub>								
188			PJ10	TI9A			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
189			PJ11	TI9B			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin

**Table 27.2 Correspondence between Power Supply Names and Pins (cont)**

Pin No. (FP-256H)	Power Supply Pin	Dedicated Pin	User Pin				Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
			Function 1	Function 2	Function 3	Function 4			
190			PJ12	TI9C			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	Schmitt-trigger input pin
191			PJ13	TI9D			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
192			PJ14	TI9E			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
193			PJ15	TI9F			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
194	PV <sub>cc</sub> 2								
195			PK0	TO8A			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
196	V <sub>ss</sub>								
197			PK1	TO8B			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
198			PK2	TO8C			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
199			PK3	TO8D			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
200			PK4	TO8E			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
201			PK5	TO8F			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
202			PK6	TO8G			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
203	V <sub>cc</sub>								
204			PK7	TO8H			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
205	V <sub>ss</sub>								
206			PK8	TO8I			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
207			PK9	TO8J			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
208			PK10	TO8K			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
209			PK11	TO8L			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
210			PK12	TO8M			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
211			PK13	TO8N			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
212	PV <sub>cc</sub> 2								
213			PK14	TO8O			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
214	V <sub>ss</sub>								
215			PK15	TO8P			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	
216			PL0	TI10			PV <sub>cc</sub> 2	PV <sub>cc</sub> 2+0.3	Schmitt-trigger input pin

**Table 27.2 Correspondence between Power Supply Names and Pins (cont)**

Pin No. (FP-256H)	Power Supply Pin	Dedicated Pin	User Pin				Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
			Function 1	Function 2	Function 3	Function 4			
217			PL1	TIO11A	$\overline{\text{IRQ6}}$		PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
218			PL2	TIO11B	$\overline{\text{IRQ7}}$		PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
219			PL3	TCLKB			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
220			PL4	$\overline{\text{ADTRG0}}$			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
221			PL5	$\overline{\text{ADTRG1}}$			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
222			PL6	ADEND			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
223			PL7	SCK2			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
224			PL8	SCK3			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
225	V <sub>CL</sub>								
226			PL9	SCK4	$\overline{\text{IRQ5}}$		PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
227	V <sub>SS</sub>								
228			PL10	HTxD0	HTxD1	HTxD0 and 1	PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
229			PL11	HRxD0	HRxD1	HRxD0, 1	PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
230			PL12	$\overline{\text{IRQ4}}$			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
231			PL13	$\overline{\text{IRQOUT}}$	$\overline{\text{IRQOUT}}$		PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
232				TMS				V <sub>cc</sub> +0.3	
233				$\overline{\text{TRST}}$				V <sub>cc</sub> +0.3	
234				TDI				V <sub>cc</sub> +0.3	
235				TDO			V <sub>cc</sub>		
236				TCK				V <sub>cc</sub> +0.3	
237	V <sub>CC</sub>								
238				$\overline{\text{AUDRST}}$				PV <sub>cc2</sub> +0.3	
239	V <sub>SS</sub>								
240				AUDMD				PV <sub>cc2</sub> +0.3	
241				AUDATA0			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	



**Table 27.2 Correspondence between Power Supply Names and Pins (cont)**

Pin No. (FP-256H)	Power Supply Pin	Dedicated Pin	User Pin				Output Circuit Power Supply Name	Input Voltage Upper Limit (V)	Notes
			Function 1	Function 2	Function 3	Function 4			
242			AUDATA1				PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
243			AUDATA2				PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
244			AUDATA3				PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
245			AUDCK				PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
246			AUDSYNC				PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
247	PV <sub>cc2</sub>								
248			PD0	TIO1A			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
249	V <sub>SS</sub>								
250			PD1	TIO1B			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	Schmitt-trigger input pin
251			PD2	TIO1C			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
252			PD3	TIO1D			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
253			PD4	TIO1E			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
254			PD5	TIO1F			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
255			PD6	TIO1G			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	
256			PD7	TIO1H			PV <sub>cc2</sub>	PV <sub>cc2</sub> +0.3	

## [Usage Notes]

Set power supply voltages during LSI operation as shown below.

$$V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V} / 3.3 \text{ V} \pm 0.3 \text{ V}, PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}, \\ AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}, AV_{ref} = 4.5 \text{ V to } AV_{CC}, V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$$

When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$

The  $PV_{CC1}$  power supply voltage depends on the operating mode as shown below. Operation cannot be guaranteed with other  $PV_{CC1}$  power supply voltages.

**Table 27.3  $PV_{CC1}$  Voltage in Each Operating Mode**

Operating Mode No	Pin Setting				Mode Name	$PV_{CC1}$ Voltage
	FEW	MD2	MD1	MD0		
Mode 0	0	1	0	0	MCU expanded mode	3.3 V $\pm$ 0.3 V
Mode 1	0	1	0	1		
Mode 2	0	1	1	0		
Mode 3	0	1	1	1	MCU Single-chip mode	5.0 V $\pm$ 0.5 V
Mode 4	1	1	0	0	Boot mode	3.3 V $\pm$ 0.3 V
Mode 5	1	1	0	1		5.0 V $\pm$ 0.5 V
Mode 6	1	1	1	0	User program mode	3.3 V $\pm$ 0.3 V
Mode 7	1	1	1	1		5.0 V $\pm$ 0.5 V
Mode 8	1	0	0	0	User boot mode	3.3 V $\pm$ 0.3 V
Mode 9	1	0	0	1		5.0 V $\pm$ 0.5 V

**Table 27.4 DC Characteristics**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}$ ,  
 $PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{ref} = 4.5 \text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .  
 When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$ .  
 When writing or erasing flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input high-level voltage (except Schmitt trigger input voltage)	$\overline{\text{RES}}$ , $\text{NMI}$ , $\text{FWE}$ , $\text{MD2-0}$ , $\overline{\text{HSTBY}}$	$V_{CC}$ - 0.4	—	5.8	V	$2.4\text{V} \leq V_{CC} < 2.7\text{V}$
		$V_{CC}$ - 0.5		5.8		$2.7\text{V} \leq V_{CC} < 3.6\text{V}$
	EXTAL	$V_{CC}$ $\times 0.7$	—	$V_{CC}$ + 0.3	V	
	$\text{D15-D0}$ , $\overline{\text{WAIT}}$ , $\text{BREQ}$ (When in MCU expanded mode)	2.2	—	$PV_{CC1}$ + 0.3	V	$PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$
	$\text{PE15-PE0}$ , $\text{PF15-PF0}$ , $\text{PH15-PH0}$ (When in MCU expanded mode)	2.2	—	$PV_{CC1}$ + 0.3	V	$PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$
	$\overline{\text{TRST}}$	$PV_{CC}$ - 0.5	—	$V_{CC}$ + 0.3	V	
	$\text{TMS}$ , $\text{TDI}$ , $\text{TCK}$	2.2	—	$V_{CC}$ + 0.3	V	
	$\overline{\text{AUDRST}}$ , $\text{AUDMD}$	$PV_{CC}$ - 0.5	—	$PV_{CC2}$ + 0.3	V	
$\text{PG0}$ , $\text{PL11}$	$PV_{CC2}$ $\times 0.7$		$PV_{CC2}$ + 0.3	V		
Other input pins	2.2	—	$PV_{CC}$ + 0.3	V		
Input low-level voltage (except Schmitt trigger input voltage)	$\overline{\text{RES}}$ , $\text{NMI}$ , $\text{FWE}$ , $\text{MD2-0}$ , $\overline{\text{HSTBY}}$ , $\overline{\text{TRST}}$ , $\overline{\text{AUDRST}}$ , $\text{AUDMD}$	-0.3	—	0.4	V	$2.4\text{V} \leq V_{CC} < 2.7\text{V}$
		-0.3	—	0.5		$2.7\text{V} \leq V_{CC} < 3.6\text{V}$
	$\text{PG0}$ , $\text{PL11}$	-0.3	—	$PV_{CC2}$ $\times 0.3$	V	
	Other input pins	-0.3	—	0.8	V	

**Table 27.4 DC Characteristics (cont)**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V} / 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  
 $PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{ref} = 4.5 \text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .  
 When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$ .  
 When writing or erasing flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item	Symbol	Min	Typ	Max	Unit	Measurement Conditions	
Schmitt trigger input voltage	TI0A–TI0D, TIO1A–TIO1H, TIO2A–TIO2H, TIO3A–TIO3D, TIO4A–TIO4D, TIO5A–TIO5D, TI9A–TI9F, TI10, TIO11A–TIO11B, TCLKA, TCLKB, ADTRG0, ADTRG1, SCK0–SCK4, IRQ0–IRQ7 and when these pins are selected as I/O ports	$(V_{IH})$ $V_T^+$	4.0	—	$(PV_{CC2} + 0.3)$	V	Refer to table 27.2, Correspondence between Power Supply Names and Pins
		$(V_{IL})$ $V_T^-$	(–0.3)	—	1.0	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input leak current	$\overline{RES}$ , NMI, FWE, MD2–0, $\overline{HSTBY}$ , EXTAL (Standby)	lin	—	—	$3.0^{*1}$ $6.0^{*2}$	$\mu\text{A}$	$V_{in} = 0.3 \text{ V}$ to 5.8 V
	TMS, $\overline{TRST}$ , TDI, TCK (Standby)		—	—	$3.0^{*1}$ $6.0^{*2}$	$\mu\text{A}$	$V_{in} = 0.3 \text{ V}$ to $V_{CC} - 0.3 \text{ V}$
	AUDMD, AUDCK, AUDSYNC, AUDATA3–0 (Standby)		—	—	$3.0^{*1}$ $6.0^{*2}$	$\mu\text{A}$	$V_{in} = 0.3 \text{ V}$ to $PV_{CC2} - 0.3 \text{ V}$
	$\overline{AUDRST}$ (Standby)		—	—	$3.0^{*1}$ $6.0^{*2}$	$\mu\text{A}$	$V_{in} = 0.3 \text{ V}$ to $PV_{CC2} - 0.3 \text{ V}$
	A/D port		—	—	$0.1^{*1}$ $0.2^{*2}$	$\mu\text{A}$	$V_{in} = 0.3 \text{ V}$ to $AV_{CC} - 0.3 \text{ V}$

**Table 27.4 DC Characteristics (cont)**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V}/3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .  
 When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .  
 When writing or erasing flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item	Symbol	Min	Typ	Max	Unit	Measurement Conditions
Input leak current	D15–D0, $\overline{\text{WAIT}}$ , $\overline{\text{BREQ}}$   lin   (When in MCU expanded mode)	—	—	3.0* <sup>1</sup> 6.0* <sup>2</sup>	μA	Vin = 0.3 V to PV <sub>CC1</sub> – 0.3 V PV <sub>CC1</sub> = 3.3 V ± 0.3 V
	PE15–PE0, PF15– PF0, PH15–PH0 (When in MCU expanded mode)	—	—	3.0* <sup>1</sup> 6.0* <sup>2</sup>	μA	Vin = 0.3 V to PV <sub>CC1</sub> – 0.3 V PV <sub>CC1</sub> = 3.3 V ± 0.3 V
	Other input pins	—	—	3.0* <sup>1</sup> 6.0* <sup>2</sup>	μA	Vin = 0.3 V to PV <sub>CC2</sub> – 0.3 V
Input pull-up MOS current	TMS, $\overline{\text{TRST}}$ , TDI, TCK –Ipu (pull-up characteristic)	—	—	350	μA	Vin = 0 V
	AUDMD, AUDCK, $\overline{\text{AUDSYNC}}$ , AUDATA3-0 (pull-up characteristic)	—	—	800	μA	Vin = 0 V
Input pull-down MOS current	$\overline{\text{AUDRST}}$ (pull-down characteristic)	Ipd	—	800	μA	Vin = PV <sub>CC2</sub>
Three-state leak current (while OFF)	A21–A0, D15–D0, $\overline{\text{CS3}}\text{--}\overline{\text{CS0}}$ , $\overline{\text{WRH}}$ , $\overline{\text{WRL}}$ , $\overline{\text{RD}}$ , $\overline{\text{BACK}}$ (When in MCU expanded mode)	I <sub>ts</sub>	—	3.0* <sup>1</sup> 6.0* <sup>2</sup>	μA	Vin = 0.3 V to PV <sub>CC1</sub> – 0.3 V PV <sub>CC1</sub> = 3.3 V ± 0.3 V
Output high-level voltage	A21–A0, D15–D0, $\overline{\text{CS3}}\text{--}\overline{\text{CS0}}$ , $\overline{\text{WRH}}$ , $\overline{\text{WRL}}$ , $\overline{\text{RD}}$ , $\overline{\text{BACK}}$ (When in MCU expanded mode)	V <sub>OH</sub>	PV <sub>CC1</sub> – 0.5	—	V	I <sub>OH</sub> = 200 μA PV <sub>CC1</sub> = 3.3 V ± 0.3 V
	PE15–PE0, PF15– PF0, PH15–PH0 (When in MCU expanded mode)		PV <sub>CC1</sub> – 0.5	—	V	I <sub>OH</sub> = 200 μA PV <sub>CC1</sub> = 3.3 V ± 0.3 V
	CK, TDO		V <sub>CC</sub> – 0.5	—	V	I <sub>OH</sub> = 200 μA

**Table 27.4 DC Characteristics (cont)**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V} / 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  
 $PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{ref} = 4.5 \text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .  
 When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$ .  
 When writing or erasing flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Output high-level voltage	Other output pins	$V_{OH}$	$PV_{CC} - 0.5$	—	—	V	$I_{OH} = 200 \mu\text{A}$
			$PV_{CC} - 1.0$	—	—	V	$I_{OH} = 1 \text{ mA}$
Output low-level voltage	A21–A0, D15–D0, CS3–CS0, WRH, WRL, RD, BACK (When in MCU expanded mode)	$V_{OL}$	—	—	0.4	V	$I_{OL} = 1.6 \text{ mA}$ $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$
	PE15–PE0, PF15–PF0, PH15–PH0 (When in MCU expanded mode)		—	—	0.4	V	$I_{OL} = 1.6 \text{ mA}$ $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$
	Other output pins (except XTAL)		—	—	0.4	V	$I_{OL} = 1.6 \text{ mA}$
			—	—	1.2	V	$I_{OL} = 6 \text{ mA}$
Input capacitance	RES	$C_{in}$	—	—	60	pF	$V_{in} = 0 \text{ V}$
	NMI		—	—	30	pF	$f = 1 \text{ MHz}$
	All other input pins		—	—	20	pF	$T_a = 25^\circ\text{C}$
Current consumption	Normal operation	$I_{CC}$	—	100	150	mA	$f = 80 \text{ MHz}$
	Sleep		—	80	130	mA	
	Standby ( $2.7 \text{ V} \leq V_{CC} \leq 3.6 \text{ V}$ )		—	—	300	$\mu\text{A}$	$T_a \leq 50^\circ\text{C}$
			—	—	750	$\mu\text{A}$	$50^\circ\text{C} < T_a \leq 105^\circ\text{C}$
			—	—	1000	$\mu\text{A}$	$105^\circ\text{C} < T_a \leq 125^\circ\text{C}$
	RAM standby ( $2.4 \text{ V} \leq V_{CC} \leq 2.7 \text{ V}$ )		—	—	600	$\mu\text{A}$	$T_a \leq 50^\circ\text{C}$
			—	—	1000	$\mu\text{A}$	$50^\circ\text{C} < T_a \leq 105^\circ\text{C}$
			—	—	1000	$\mu\text{A}$	$105^\circ\text{C} < T_a \leq 125^\circ\text{C}$
Write operation	—	80	130	mA	$V_{CC} = 3.3 \text{ V}$ $f = 40 \text{ MHz}$		

**Table 27.4 DC Characteristics (cont)**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}$ ,  
 $PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{ref} = 4.5 \text{ V to } AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -40^\circ\text{C to } 125^\circ\text{C}$ .  
 When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$ .  
 When writing or erasing flash memory,  $T_a = -40^\circ\text{C to } 85^\circ\text{C}$ .

Item		Symbol	Min	Typ	Max	Unit	Measurement Conditions
Analog supply current	During A/D conversion	$AI_{CC}$	—	1.2	5	mA	
	Awaiting A/D conversion, standby		—	1.0	30	$\mu\text{A}$	
Reference power supply current	During A/D conversions, awaiting A/D conversion	$Al_{ref}$	—	1.1	5	mA	$AV_{ref} = 5.0 \text{ V}$
	Standby		—	1.1	30	$\mu\text{A}$	
RAM standby voltage		$V_{RAM}$	2.4	—	—	V	$V_{CC}$

Notes: 1.  $T_a \leq 105^\circ\text{C}$   
 2.  $T_a > 105^\circ\text{C}$

### [Operating precautions]

1. When the A/D converter is not used (including during standby), do not leave the  $AV_{CC}$ ,  $AV_{ref}$ , and  $AV_{SS}$  pins open.
2. The current consumption is measured when  $V_{IH\text{min}} = V_{CC} - 0.5 \text{ V}/PV_{CC} - 0.5 \text{ V}$ ,  $V_{IL} = 0.5 \text{ V}$ , with all output pins unloaded.
3. The guaranteed operating range of power supply  $PV_{CC1}$  in the MCU expanded modes is only  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ . Do not use a voltage outside this range.
4. The guaranteed operating range of power supply  $PV_{CC1}$  in MCU single-chip mode is only  $PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V}$ . Do not use a voltage outside this range.

## Table 27.5 Permitted Output Current Values

Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}$ ,  
 $PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{ref} = 4.5 \text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .

When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item	Symbol	Min	Typ	Max	Unit
Output low-level permissible current (per pin)	$I_{OL}$	—	—	6.0	mA
Output low-level permissible current (total)	$\Sigma I_{OL}$	—	—	80	mA
Output high-level permissible current (per pin)	$I_{OH}$	—	—	2.0	mA
Output high-level permissible current (total)	$\Sigma I_{OL}$	—	—	25	mA

### [Operating precautions]

To assure LSI reliability, do not exceed the output values listed in this table.

## 27.3 AC Characteristics

### 27.3.1 Timing for swicthing the power supply on/off

Table 27.6 Timing for swicthing the power supply on/off

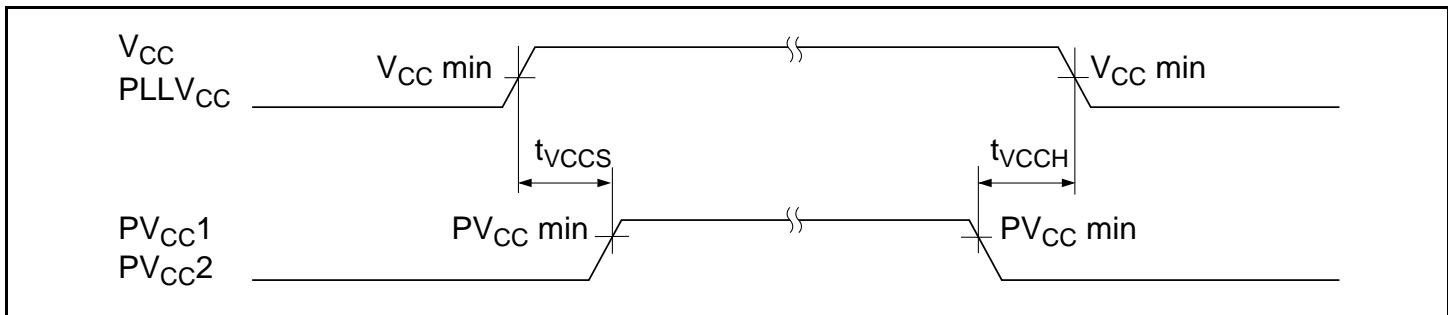
Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V}/3.3 \text{ V} \pm 0.3 \text{ V}$ ,  
 $PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{ref} = 4.5 \text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .

When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item	Symbol	Min	Max	Unit	Figures
Time taken to switch $V_{CC}$ on	$t_{VCCS}$	0	—	ms	Figure 27.1
$V_{CC}$ hold-time when $PV_{CC}$ is switched off	$t_{VCH}$	0	—	ms	





**Figure 27.1 Power-On/Off Timing**

### 27.3.2 Clock timing

Table 27.7 shows the clock timing.

**Table 27.7 Clock Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V} / 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  
 $PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{ref} = 4.5 \text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .

When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

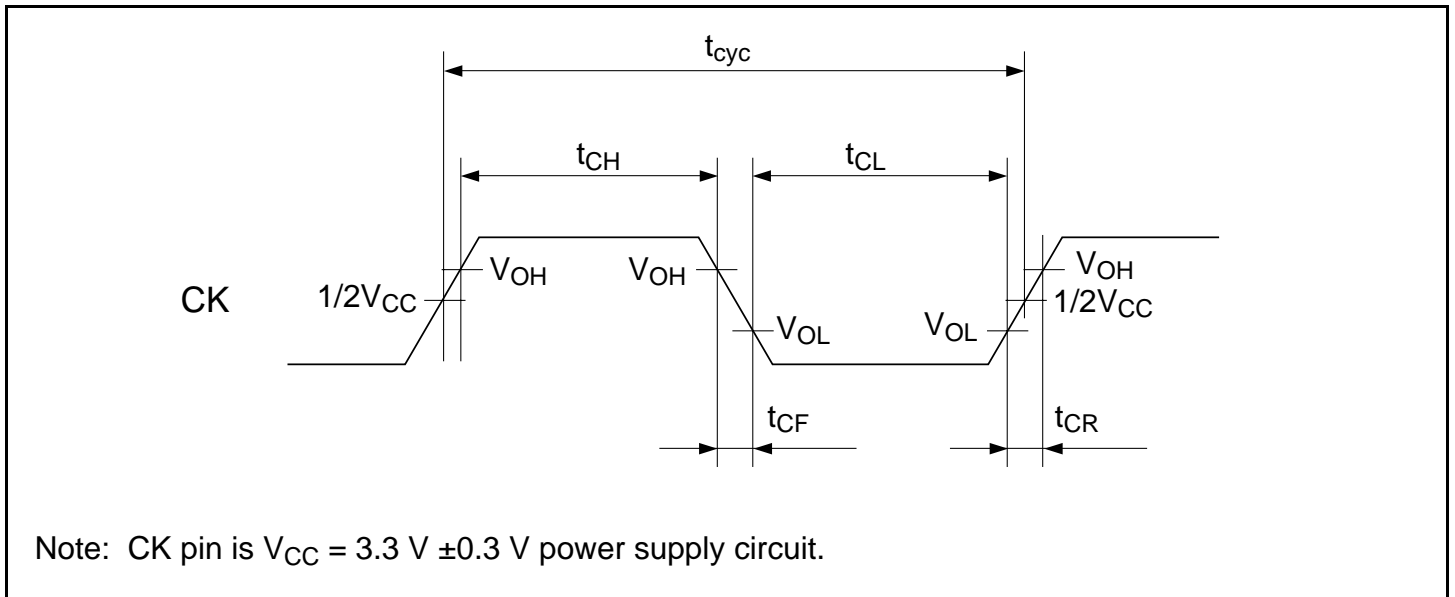
When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item	Symbol	Min	Max	Unit	Figures
Clock frequency	$f_{op}$	10	20	MHz	Figure 27.2
Clock cycle time	$t_{cyc}$	50	100	ns	
Clock low-level pulse width	$t_{CL}$	12	—	ns	
Clock high-level pulse width	$t_{CH}$	12	—	ns	
Clock rise time	$t_{CR}$	—	10	ns	
Clock fall time	$t_{CF}$	—	10	ns	
EXTAL clock input frequency	$f_{EX}$	5	10	MHz	Figure 27.3
EXTAL clock input cycle time	$t_{EXCyc}$	100	200	ns	
EXTAL clock input low-level pulse width	$t_{EXL}$	30	—	ns	
EXTAL clock input high-level pulse width	$t_{EXH}$	30	—	ns	
EXTAL clock input rise time	$t_{EXR}$	—	8	ns	
EXTAL clock input fall time	$t_{EXF}$	—	8	ns	
Reset oscillation settling time	$t_{osc1}$	30	—	ms	Figure 27.4
Standby return clock settling time	$t_{osc2}$	30	—	ms	

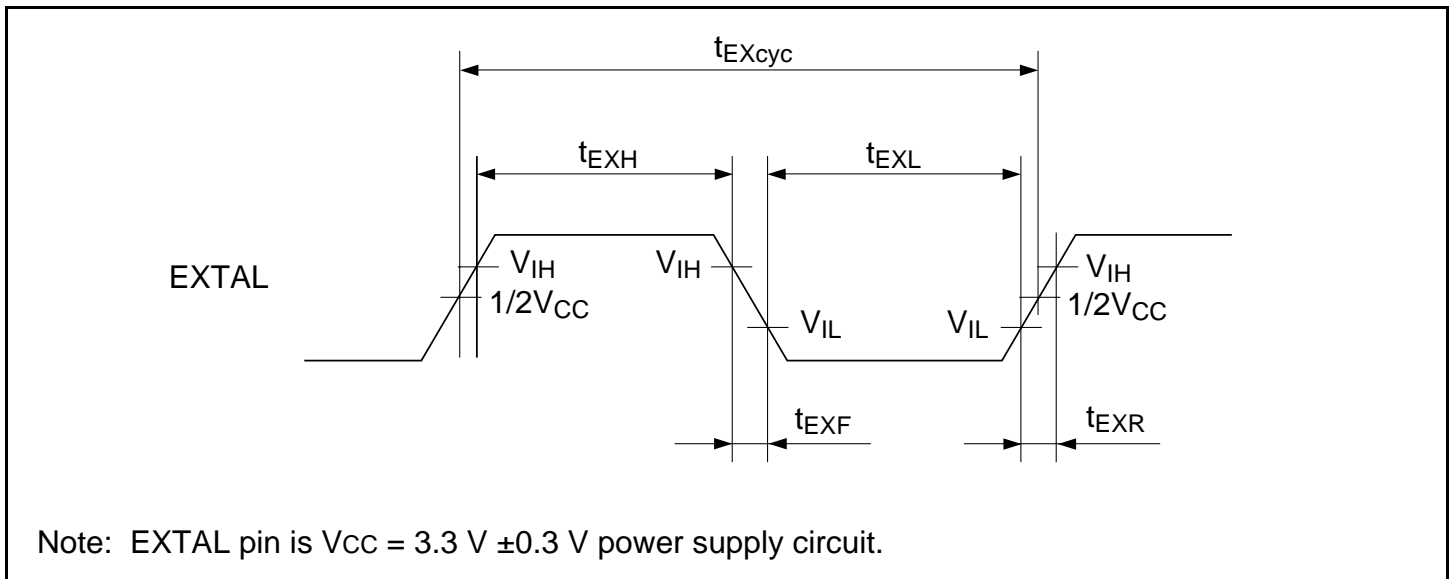
The CK pin outputs the peripheral clock signal (P $\phi$ ).

## [Operating precautions]

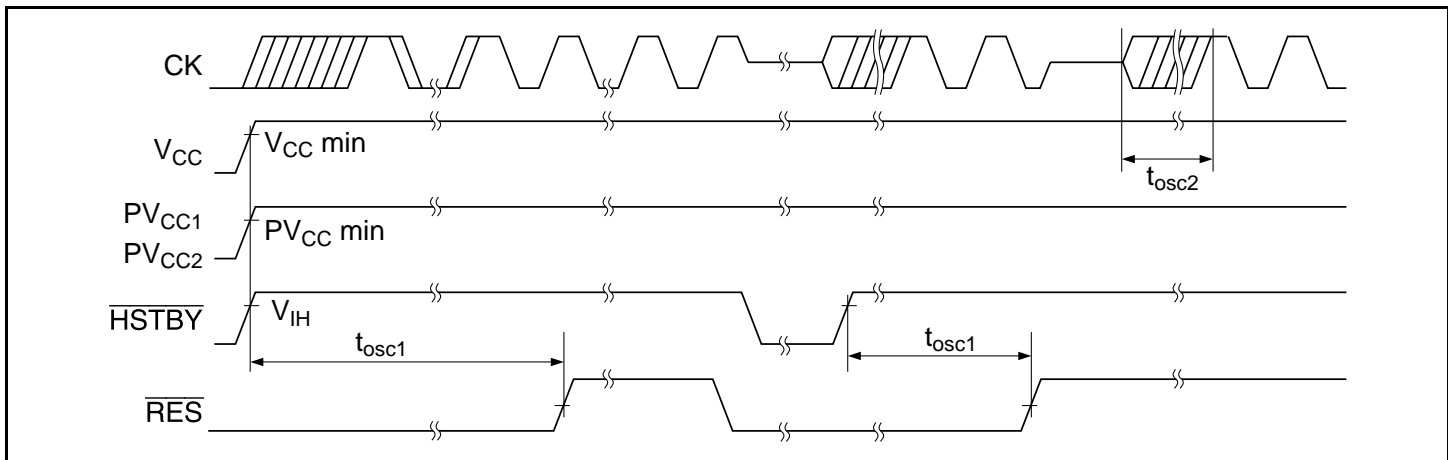
The EXTAL, XTAL, and CK pins constitute a circuit requiring a power supply voltage of  $V_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ . Comply with the input and output voltages specified in the DC characteristics.



**Figure 27.2 Peripheral Clock Timing**



**Figure 27.3 EXTAL Clock Input Timing**



**Figure 27.4 Oscillation Settling Time**

### 27.3.3 Control Signal Timing

Table 27.8 shows control signal timing.

**Table 27.8 Control Signal Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V} / 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  
 $PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{ref} = 4.5 \text{ V to } AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -40^\circ\text{C to } 125^\circ\text{C}$ .

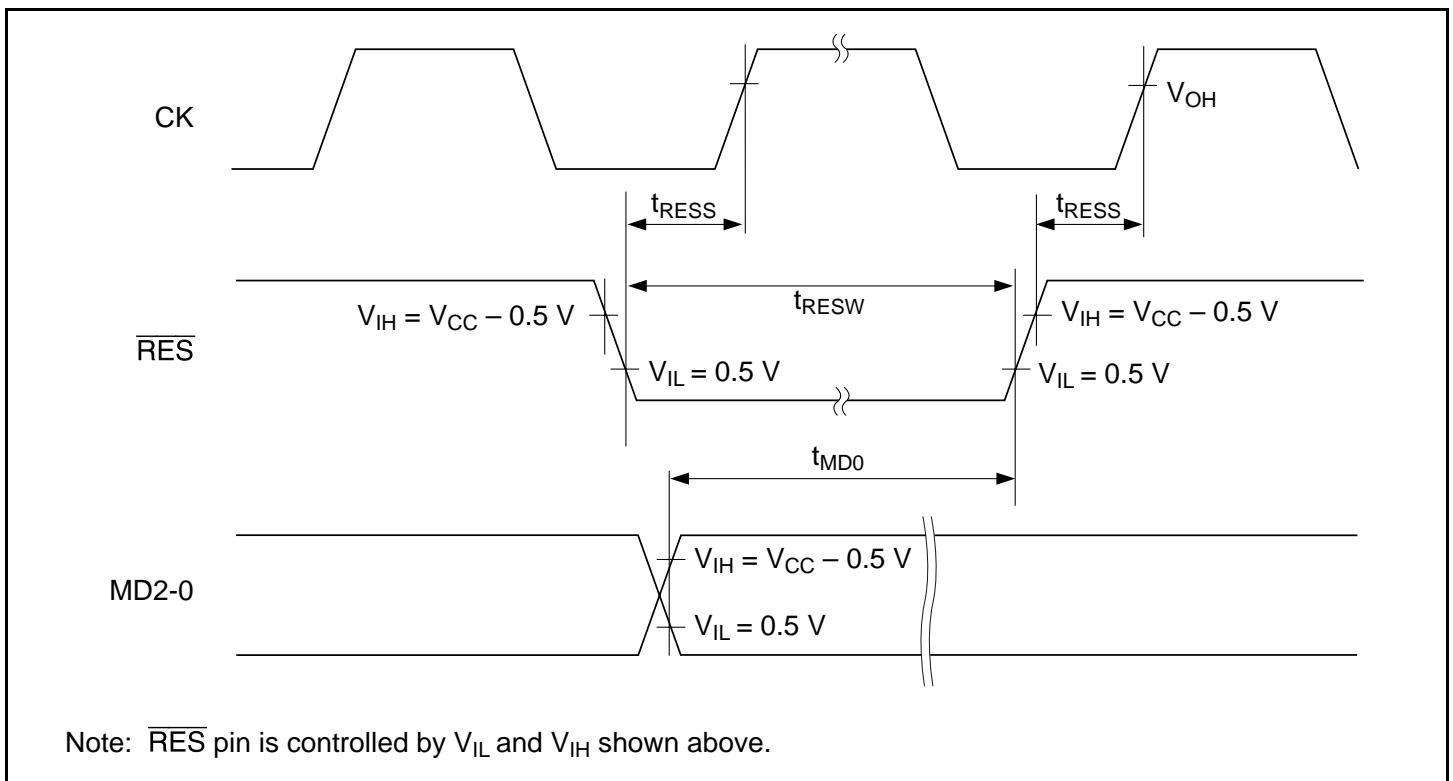
When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C to } 85^\circ\text{C}$ .

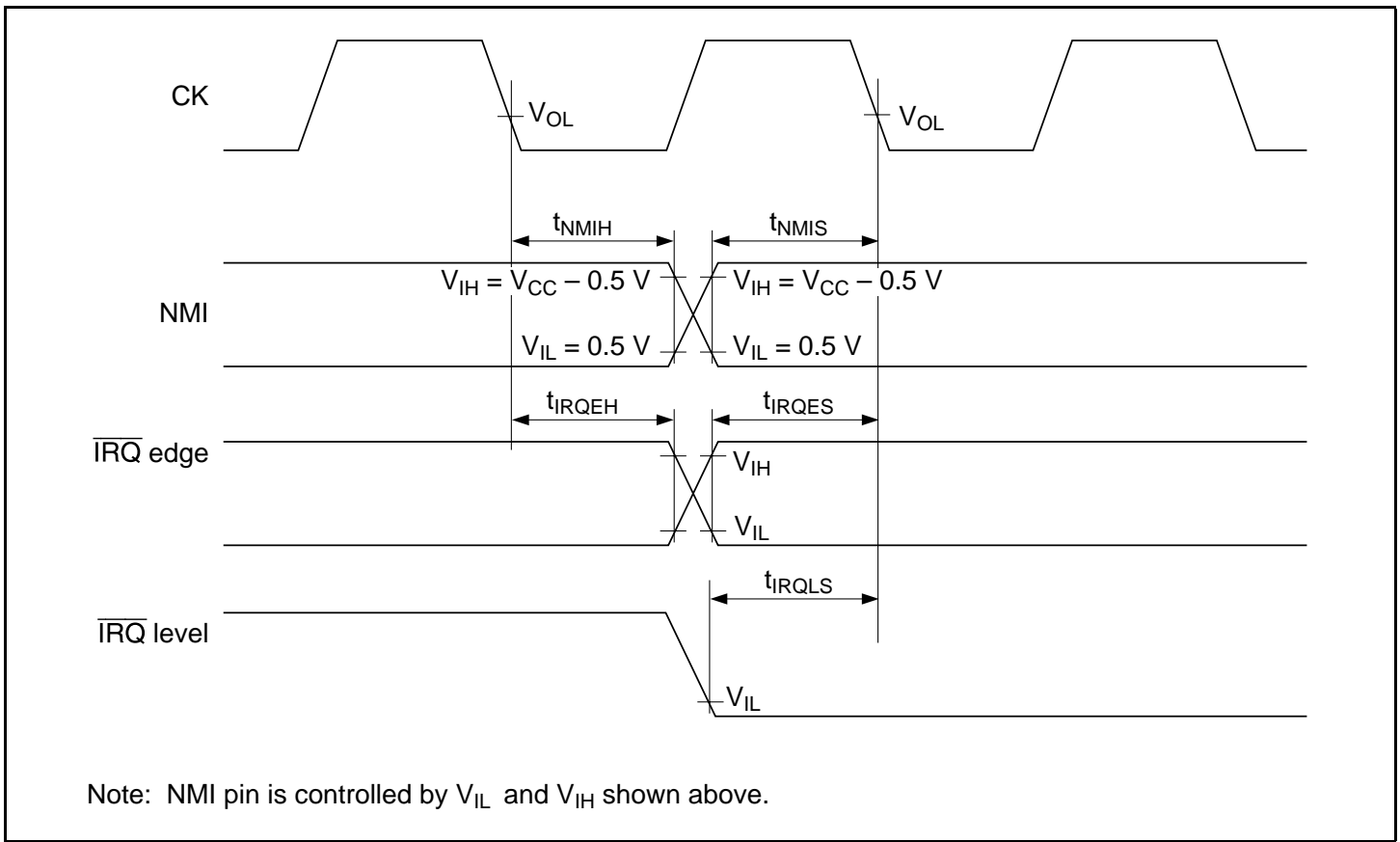
Item	Symbol	Min	Max	Unit	Figures
$\overline{\text{RES}}$ pulse width	$t_{\text{RESW}}$	10	—	$t_{\text{cyc}}$	Figure 27.5
$\overline{\text{RES}}$ setup time	$t_{\text{RESS}}$	30	—	ns	
MD2 to MD0 setup time 2* <sup>1</sup>	$t_{\text{MDS}}$	10	—	$t_{\text{cyc}}$	
NMI setup time	$t_{\text{NMIS}}$	30	—	ns	Figure 27.6
$\overline{\text{IRQ7}}\text{--}\overline{\text{IRQ0}}$ setup time* <sup>2</sup> (edge detection)	$t_{\text{IRQES}}$	30	—	ns	
$\overline{\text{IRQ7}}\text{--}\overline{\text{IRQ0}}$ setup time* <sup>2</sup> (level detection)	$t_{\text{IRQLS}}$	30	—	ns	
NMI hold time	$t_{\text{NMIH}}$	30	—	ns	
$\overline{\text{IRQ7}}\text{--}\overline{\text{IRQ0}}$ hold time	$t_{\text{IRQEH}}$	30	—	ns	
$\overline{\text{IRQOUT}}$ output delay time	$t_{\text{IRQOD}}$	—	100	ns	Figure 27.7
Bus request setup time	$t_{\text{BRQS}}$	30	—	ns	Figure 27.8* <sup>3</sup>
Bus acknowledge delay time 1	$t_{\text{BACKD1}}$	—	30	ns	
Bus acknowledge delay time 2	$t_{\text{BACKD2}}$	—	30	ns	
Bus three-state delay time	$t_{\text{BZD}}$	—	30	ns	

## [Operating precautions]

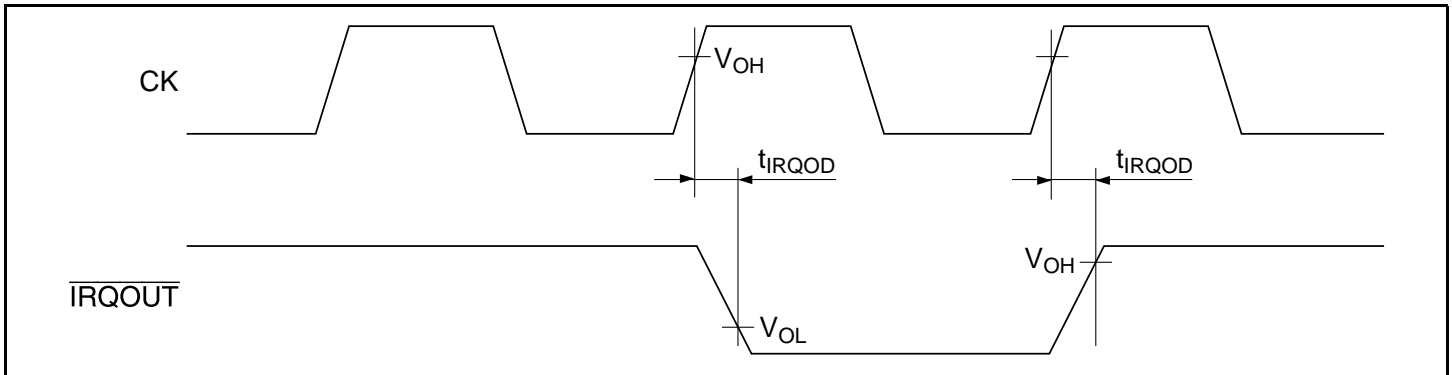
1. Mode setup time during power-on reset by the RES pin depends on the combination of signals to be input to the FWE and MD2 to MD0 pins. If a low-level signal is input to the RES pin while this LSI operates by inputting a mode specified in table 27.3 to the FWE and MD2 to MD0 pins, the mode setup time is defined by  $t_{MDS2}$ . If a signal other than the combination of signals specified in table 27.3 (undefined mode) is input to the FWE and MD2 to MD0 pins, the mode setup time is defined by  $t_{MDS1}$ . See section 27.6.2, Notes on Mode Pin Input.
2. The  $\overline{RES}$ , NMI, and  $\overline{IRQ7}$ – $\overline{IRQ0}$  signals are asynchronous inputs, but when the setup times shown here are provided, the signals are considered to have been changed at clock fall. If the setup times are not provided, recognition is delayed until the next clock rise or fall.
3. The guaranteed operating range of power supply  $PV_{CC1}$  in the MCU expanded modes is only  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ . Do not use a voltage outside this range.



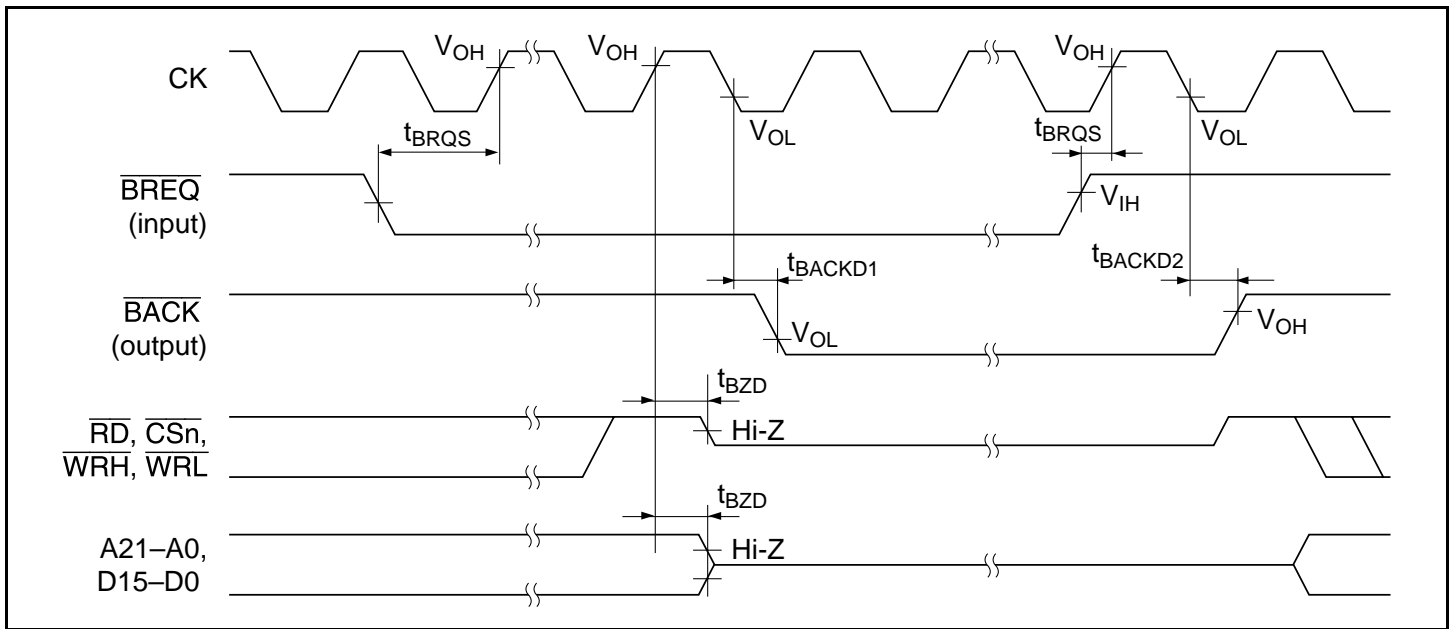
**Figure 27.5 Reset Input Timing**



**Figure 27.6 Interrupt Signal Input Timing**



**Figure 27.7 Interrupt Signal Output Timing**



**Figure 27.8 Bus Right Release Timing**

## 27.3.4 Bus Timing

Table 27.9 shows bus timing.

**Table 27.9 Bus Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  
 $PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{ref} = 4.5 \text{ V to } AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -40^\circ\text{C to } 125^\circ\text{C}$ .  
 When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$ .  
 When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C to } 85^\circ\text{C}$ .

Item	Symbol	Min	Max	Unit	Figures
Address delay time	$t_{AD}$	—	35	ns	Figures 27.9,
CS delay time 1	$t_{CSD1}$	—	30	ns	27.10
CS delay time 2	$t_{CSD2}$	—	30	ns	
Read strobe delay time 1	$t_{RSD1}$	—	30	ns	
Read strobe delay time 2	$t_{RSD2}$	—	30	ns	
Read data setup time	$t_{RDS}$	15	—	ns	
Read data hold time	$t_{RDH}$	0	—	ns	
Write strobe delay time 1	$t_{WSD1}$	—	30	ns	
Write strobe delay time 2	$t_{WSD2}$	—	30	ns	
Write data delay time	$t_{WDD}$	—	30	ns	
Write data hold time	$t_{WDH}$	$t_{cyc} \times m$	—	ns	
WAIT setup time	$t_{WTS}$	15	—	ns	Figure 27.11
WAIT hold time	$t_{WTH}$	0	—	ns	
Read data access time	$t_{ACC}$	$t_{cyc} \times (n+1.5)-39$	—	ns	Figures 27.9,
Access time from read strobe	$t_{OE}$	$t_{cyc} \times (n+1.0)-39$	—	ns	27.10
Write address setup time	$t_{AS}$	0	—	ns	
Write address hold time	$t_{WR}$	5	—	ns	

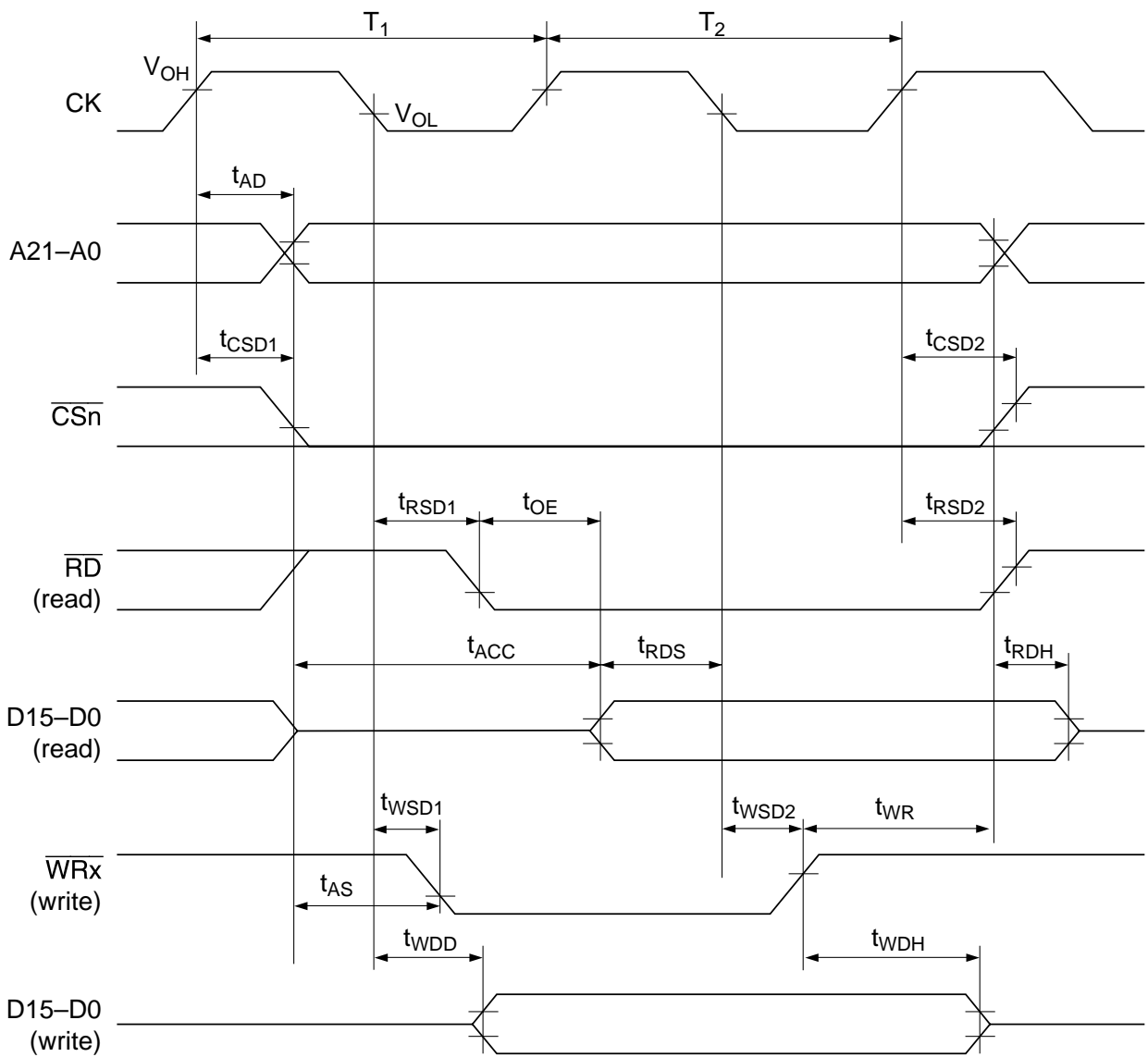
n: Number of waits

m = 1: CS assertion extension cycle

m = 0: Normal cycle (CS assertion non-extension cycle)

### [Operating precautions]

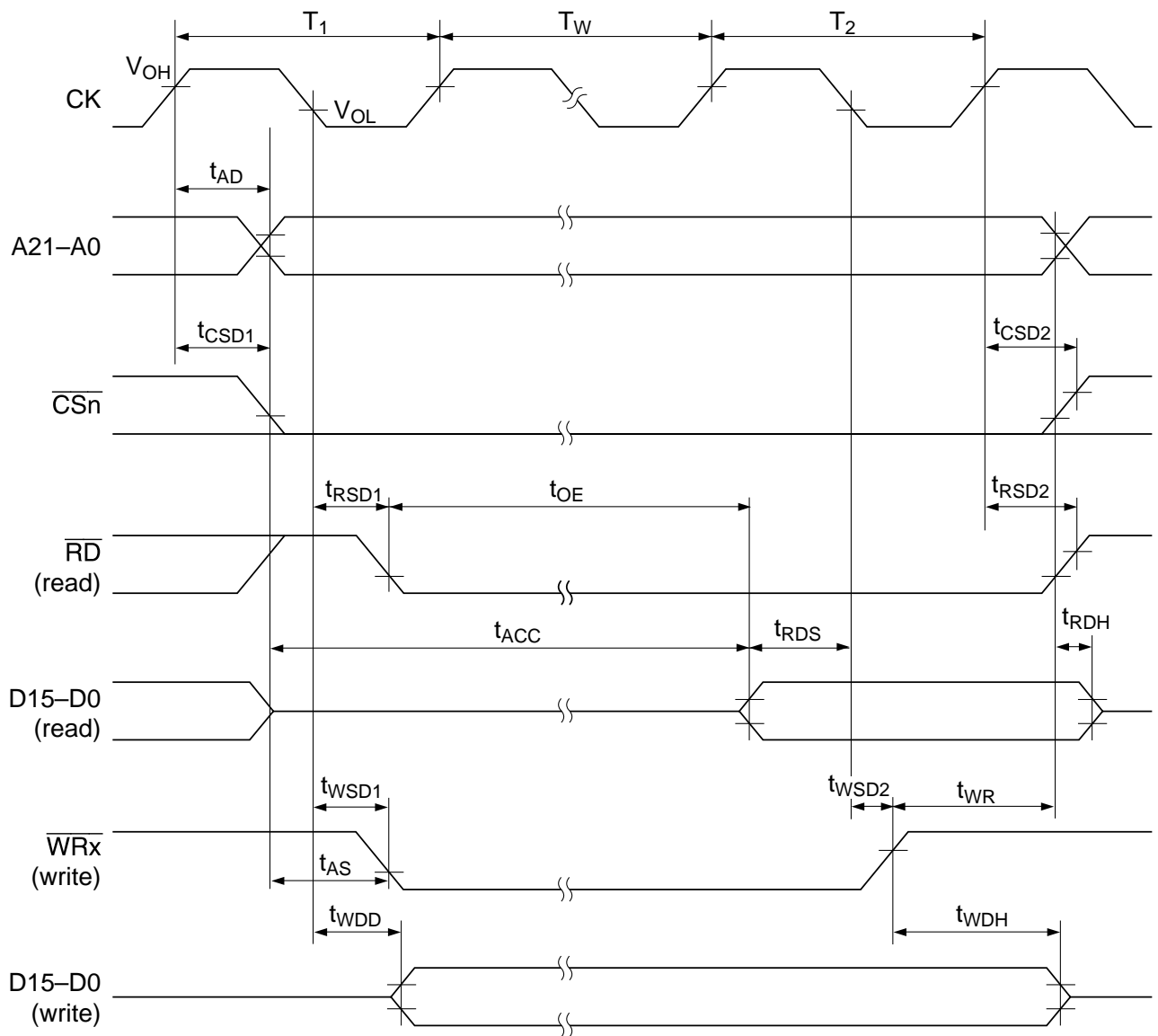
The guaranteed operating range of power supply  $PV_{CC1}$  in the MCU expanded modes is only  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ . Do not use a voltage outside this range.



Note:  $t_{RDH}$ : Specified from the negate timing of A21-A0,  $\overline{CSn}$ , or  $\overline{RD}$ , whichever is first.

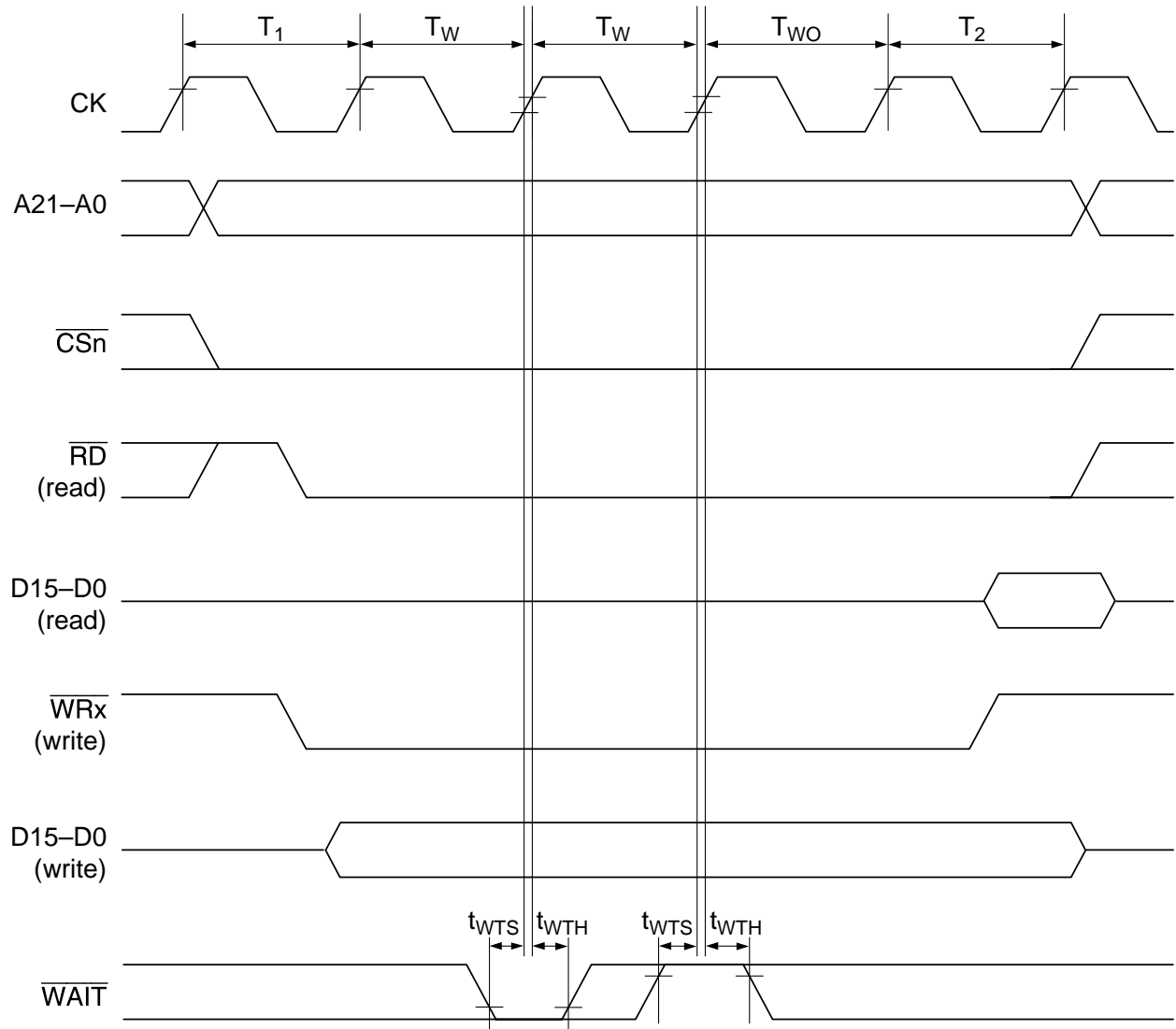
**Figure 27.9 Basic Cycle (No Waits)**





Note:  $t_{RDH}$ : Specified from the negate timing of A21-A0,  $\overline{CSn}$ , or  $\overline{RD}$ , whichever is first.

**Figure 27.10 Basic Cycle (One Software Wait)**



Note:  $t_{RDH}$ : Specified from the negate timing of  $A_{21-A0}$ ,  $\overline{CSn}$ , or  $\overline{RD}$ , whichever is first.

**Figure 27.11 Basic Cycle (Two Software Waits + Waits by  $\overline{WAIT}$  Signal)**

## 27.3.5 Advanced Timer Unit Timing and Advance Pulse Controller Timing

Table 27.10 shows advanced timer unit timing and advanced pulse controller timing.

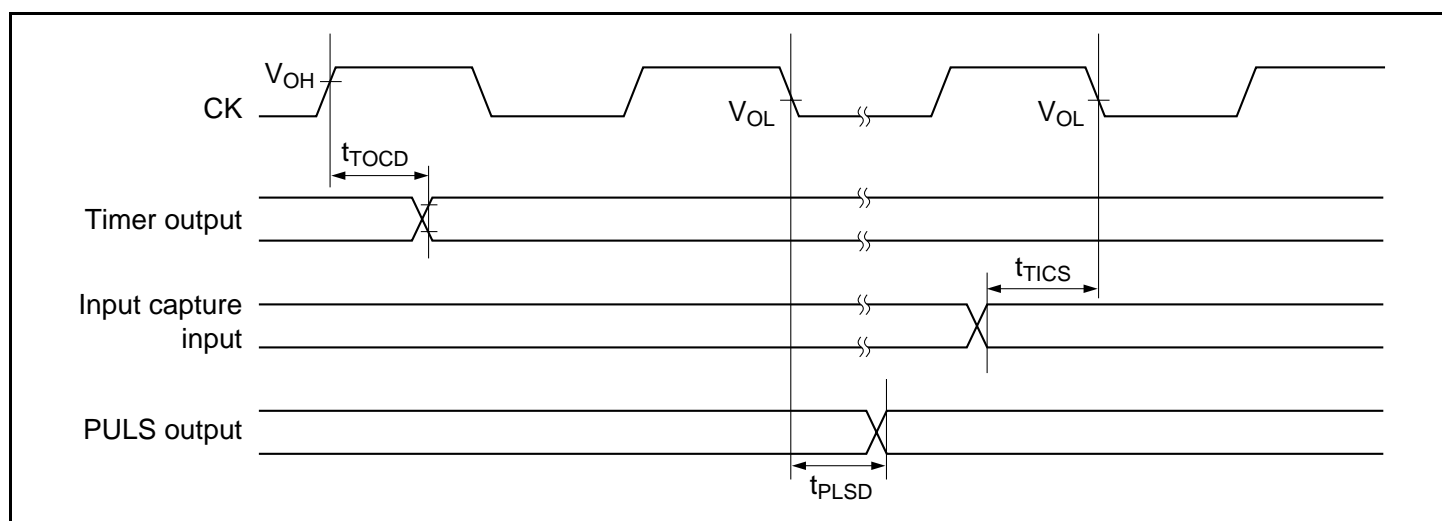
**Table 27.10 Advanced Timer Unit Timing and Advanced Pulse Controller Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V}/3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .

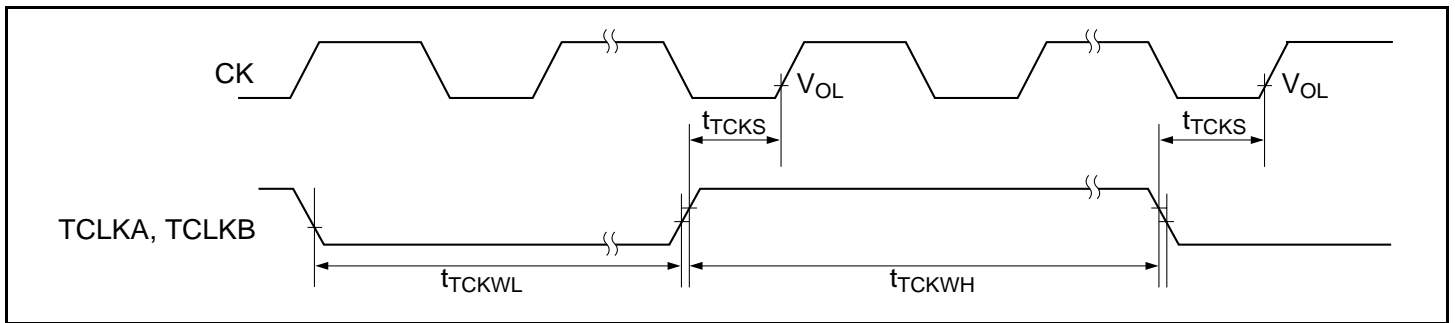
When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item	Symbol	Min	Max	Unit	Figures
Output compare output delay time	$t_{TOCD}$	—	100	ns	Figure 27.12
Input capture input setup time	$t_{TICS}$	24	—	ns	
PULS output delay time	$t_{PLSD}$	—	100	ns	
Timer clock input setup time	$t_{TCKS}$	24	—	ns	Figure 27.13
Timer clock pulse width (single edge specified)	$t_{TCKWH/L}$	1.5	—	$t_{cyc}$	
Timer clock pulse width (both edges specified)	$t_{TCKWH/L}$	2.5	—	$t_{cyc}$	



**Figure 27.12 ATU Input/Output Timing and APC Output Timing**



**Figure 27.13 ATU Clock Input Timing**

### 27.3.6 I/O Port Timing

Table 27.11 shows I/O port timing.

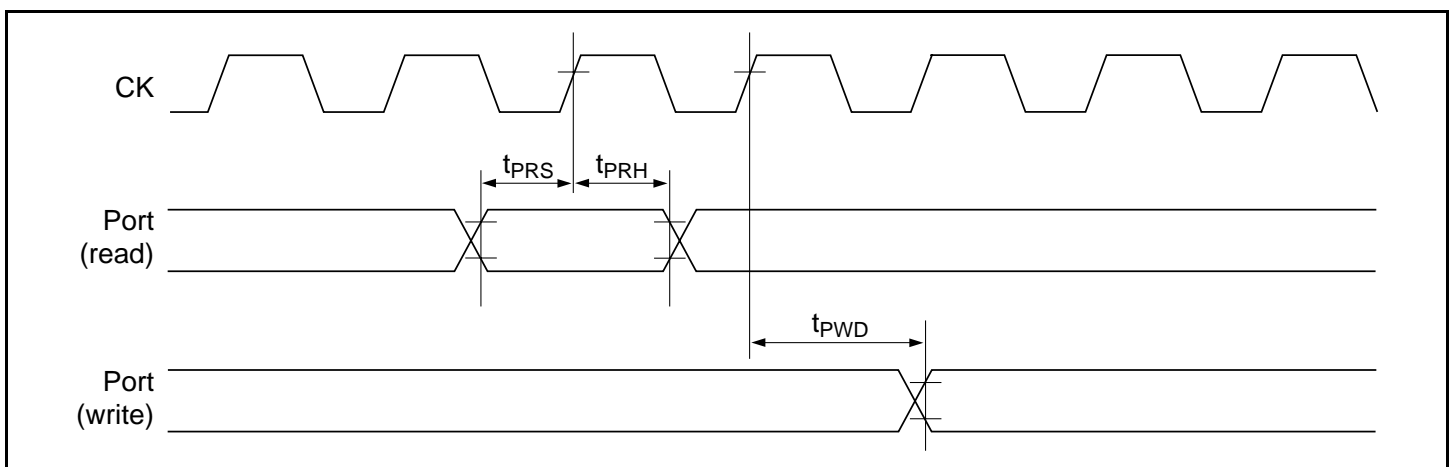
**Table 27.11 I/O Port Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V} / 3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .  
 When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .  
 When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item	Symbol	Min	Max	Unit	Figures
Port output data delay time	$t_{PWD}$	—	100	ns	Figure 27.14
Port input hold time	$t_{PRH}$	30	—	ns	
Port input setup time	$t_{PRS}$	30	—	ns	

#### [Operating precautions]

The guaranteed operating range of power supply  $PV_{CC1}$  in MCU single-chip mode is only  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V}$ . Do not use a voltage outside this range.



**Figure 27.14 I/O Port Input/Output timing**

### 27.3.7 Watchdog Timer Timing

Table 27.12 shows watchdog timer timing.

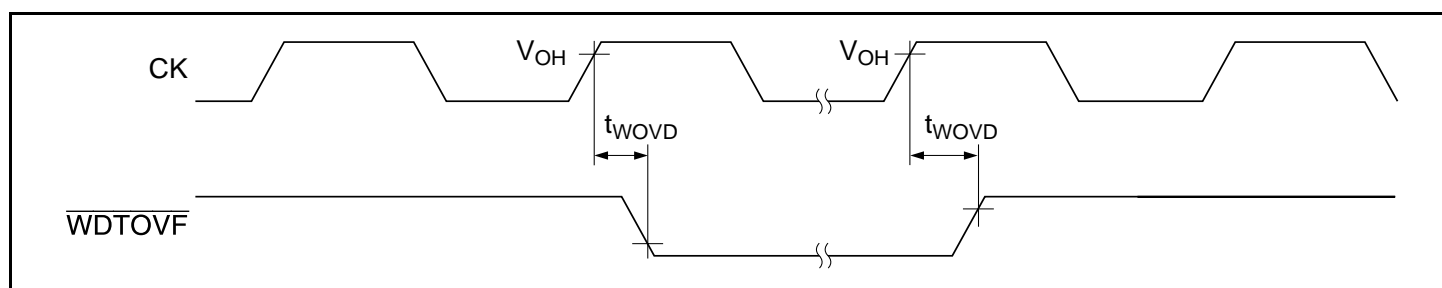
**Table 27.12 Watchdog Timer Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V}/3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V to } AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C to } 125^\circ\text{C}$ .

When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C to } 85^\circ\text{C}$ .

Item	Symbol	Min	Max	Unit	Figures
WDTOVF delay time	$t_{WOVD}$	—	100	ns	Figure 27.15



**Figure 27.15 Watchdog Timer Timing**

### 27.3.8 Serial Communication Interface Timing

Table 27.13 shows serial communication interface timing.

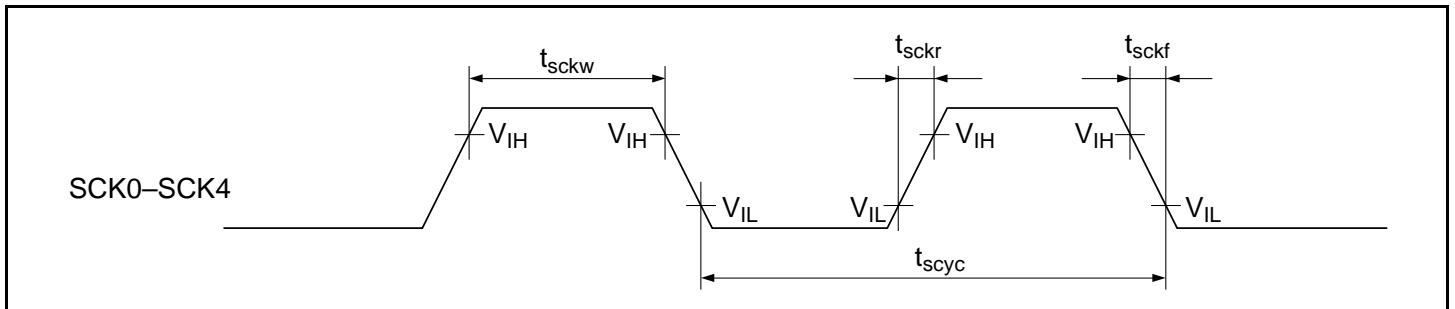
**Table 27.13 Serial Communication Interface Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V}/3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V to } AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C to } 125^\circ\text{C}$ .

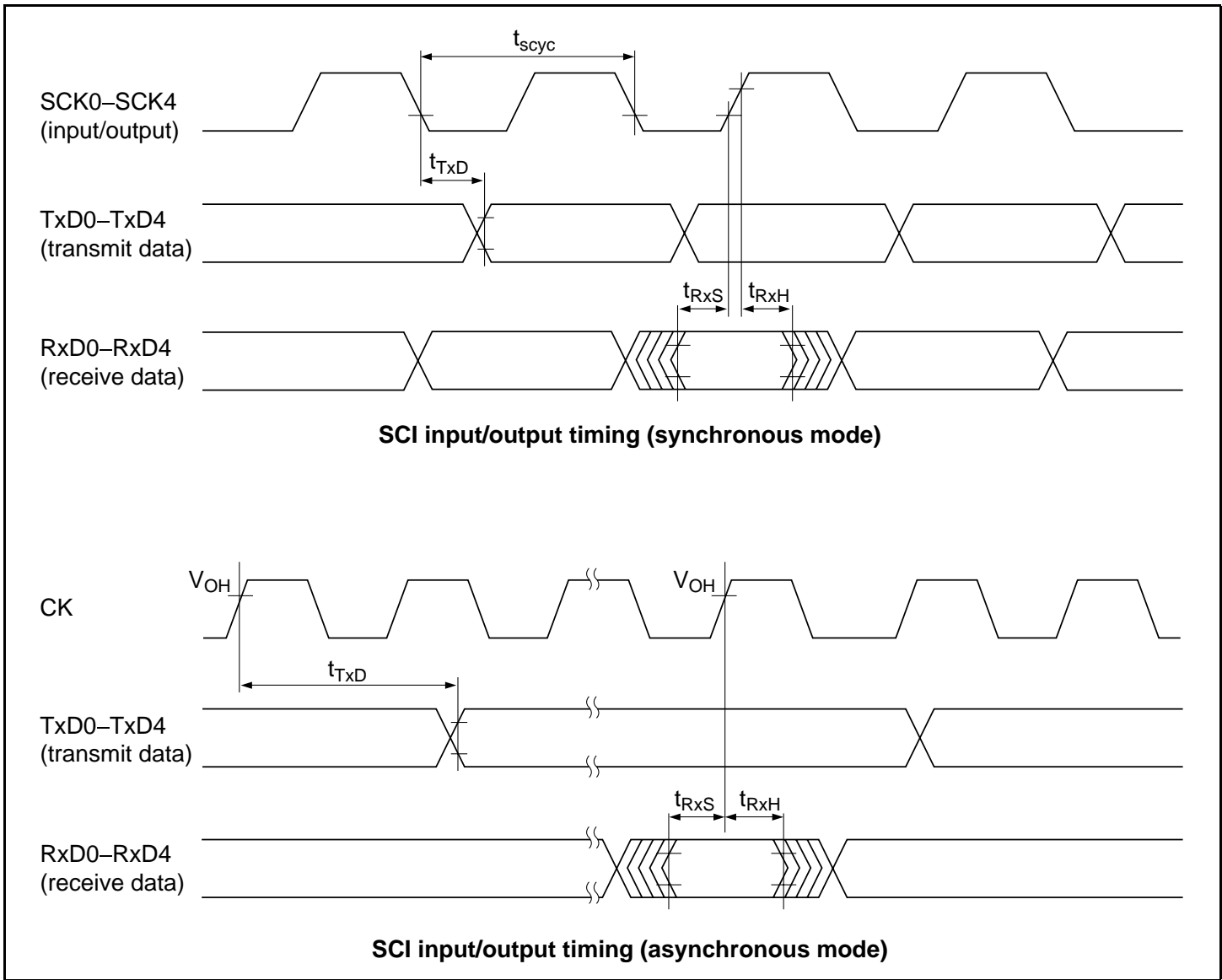
When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C to } 85^\circ\text{C}$ .

Item	Symbol	Min	Max	Unit	Figures
Clock cycle	$t_{scyc}$	4	—	$t_{cyc}$	Figure 27.16
Clock cycle (clock sync)	$t_{scyc}$	6	—	$t_{cyc}$	
Clock pulse width	$t_{sckw}$	0.4	0.6	$t_{scyc}$	
Input clock rise time	$t_{sckr}$	—	1.5	$t_{cyc}$	
Input clock fall time	$t_{sckf}$	—	1.5	$t_{cyc}$	
Transmit data delay time	$t_{TxD}$	—	100	ns	Figure 27.17
Transmit data setup time	$t_{Rxs}$	100	—	ns	
Transmit data hold time	$t_{RxH}$	100	—	ns	



**Figure 27.16 SCI Input/Output Timing**



**Figure 27.17 SCI Input/Output Timing**

### 27.3.9 HCAN Timing

Table 27.14 shows HCAN timing.

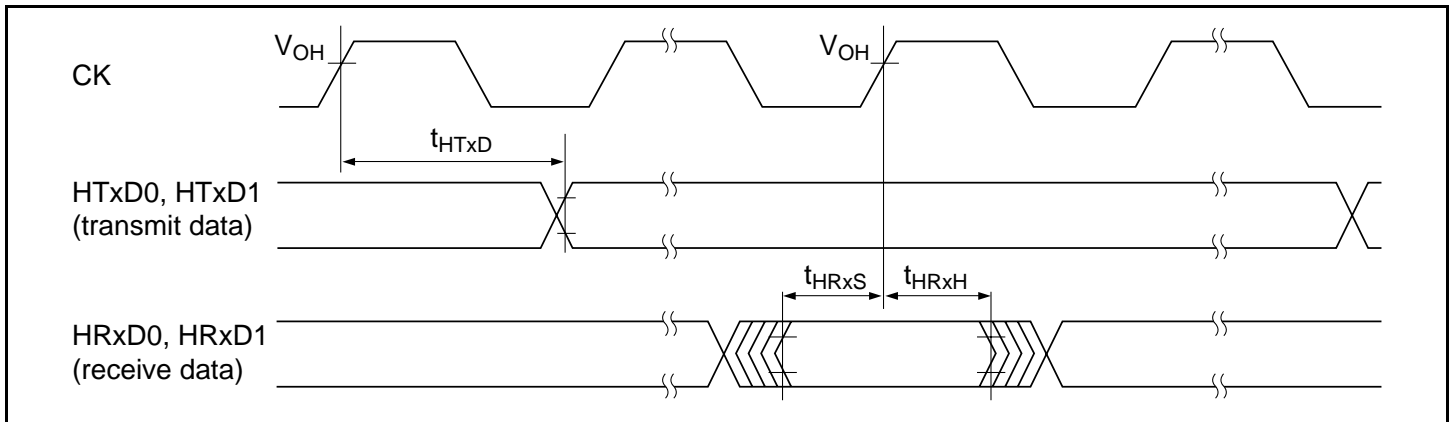
**Table 27.14 HCAN Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V} / 3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .

When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item	Symbol	Min	Max	Unit	Figures
Transmit data delay time	$t_{HTxD}$	—	100	ns	Figure 27.18
Transmit data setup time	$t_{HRxS}$	100	—	ns	
Transmit data hold time	$t_{HRxH}$	100	—	ns	



**Figure 27.18 HCAN Input/Output timing**



## 27.3.10 A/D Converter Timing

Table 27.15 shows A/D converter timing.

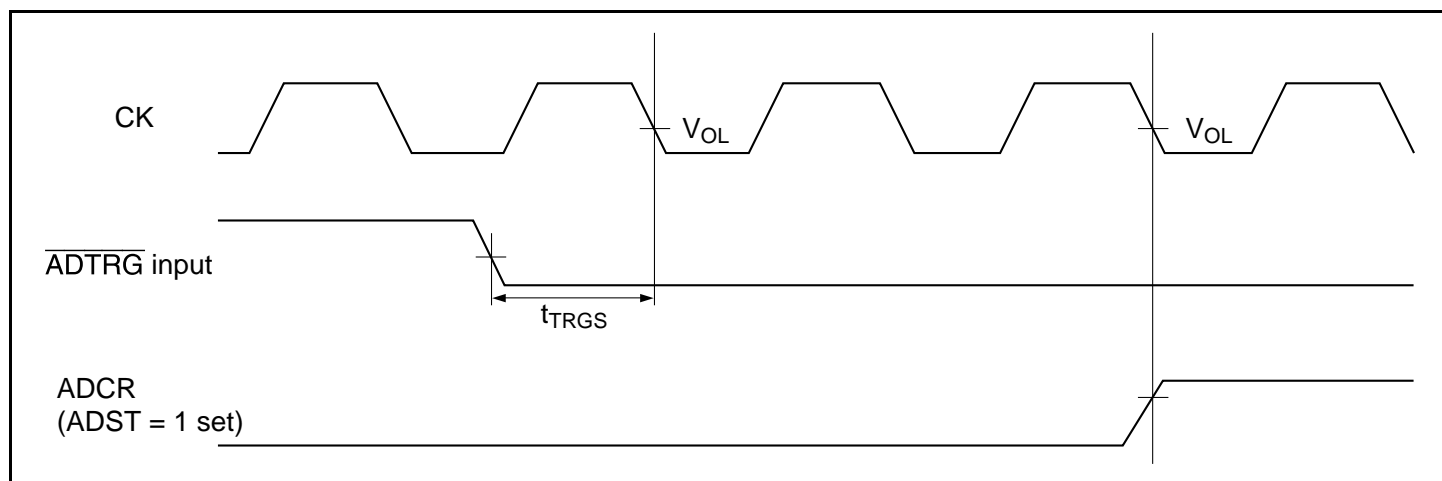
**Table 27.15 A/D Converter Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V}/3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V to } AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C to } 125^\circ\text{C}$ .

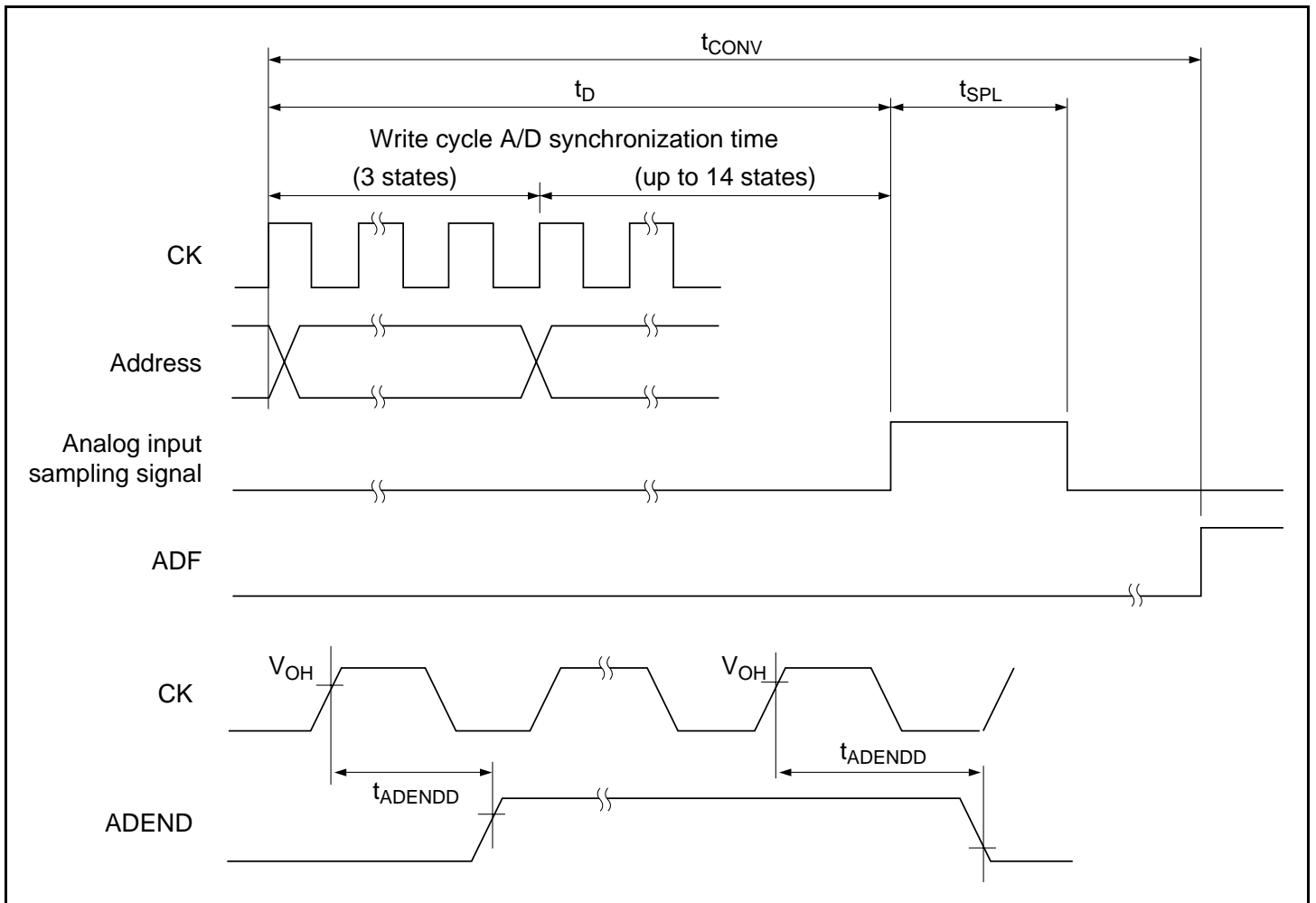
When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C to } 85^\circ\text{C}$ .

Item	Symbol	CSK = 0: fop = T.B.D.			CSK = 1: fop = T.B.D.			Unit	Figure
		Min	Typ	Max	Min	Typ	Max		
External trigger input start delay time	$t_{TRGS}$	50	—	—	50	—	—	ns	Figure 27.19
A/D conversion time	$t_{CONV}$	259	—	266	131	—	134	$t_{cyc}$	Figure 27.20
A/D conversion start delay time	$t_D$	10	—	17	6	—	9	$t_{cyc}$	
Input sampling time	$t_{SPL}$	—	64	—	—	32		$t_{cyc}$	
ADEND output delay time	$t_{ADENDD}$	—	—	100	—	—	100	ns	



**Figure 27.19 External Trigger Input Timing**



**Figure 27.20 Analog Conversion Timing**

## 27.3.11 H-UDI Timing

Table 27.16 shows H-UDI timing.

**Table 27.16 H-UDI Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V}/3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .

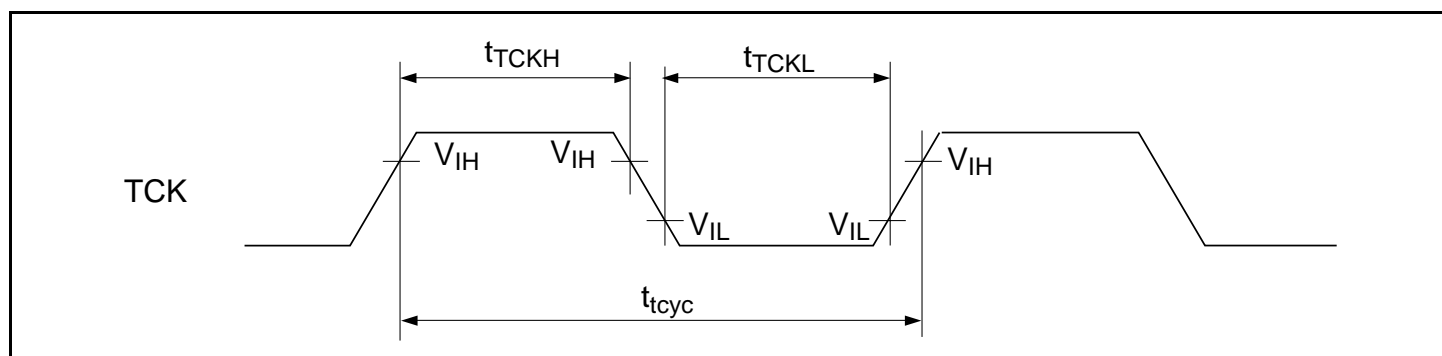
When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

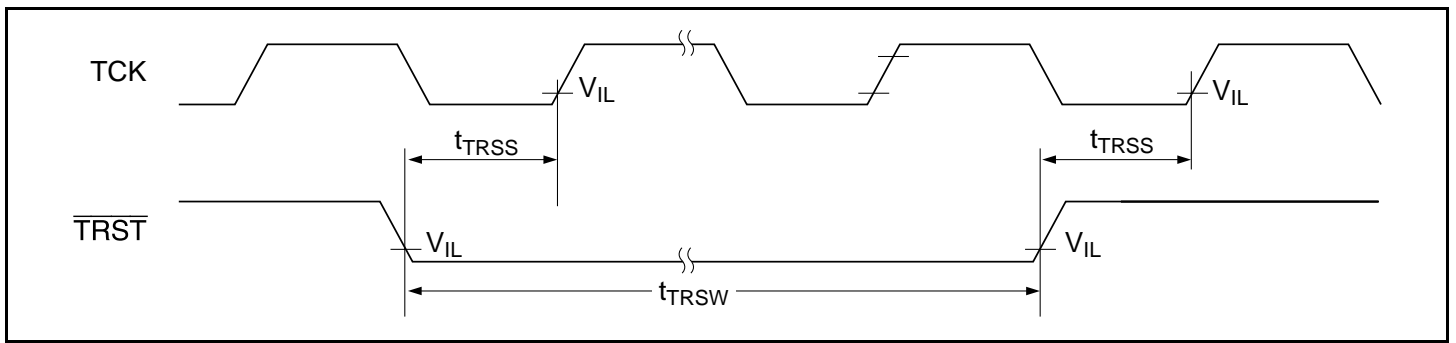
Item	Symbol	Min	Max	Unit	Figures
TCK clock cycle	$t_{tcyc}$	2	—	$t_{tcyc}$	Figure 27.21
TCK clock high-level width	$t_{TCKH}$	0.4	0.6	$t_{tcyc}$	
TCK clock low-level width	$t_{TCKL}$	0.4	0.6	$t_{tcyc}$	
$\overline{TRST}$ pulse width	$t_{TRSW}$	20	—	$t_{cyc}$	Figure 27.22
$\overline{TRST}$ setup time	$t_{TRSS}$	30	—	ns	
TMS setup time	$t_{TMSS}$	30	—	ns	Figure 27.23
TMS hold time	$t_{TMSh}$	10	—	ns	
TDI setup time	$t_{TDis}$	30	—	ns	
TDI hold time	$t_{TDIH}$	10	—	ns	
TDO delay time 1	$t_{TDOD1}$	—	30	ns	
TDO delay time 2	$t_{TDOD2}$	—	30	ns	Figure 27.24

### [Operating precautions]

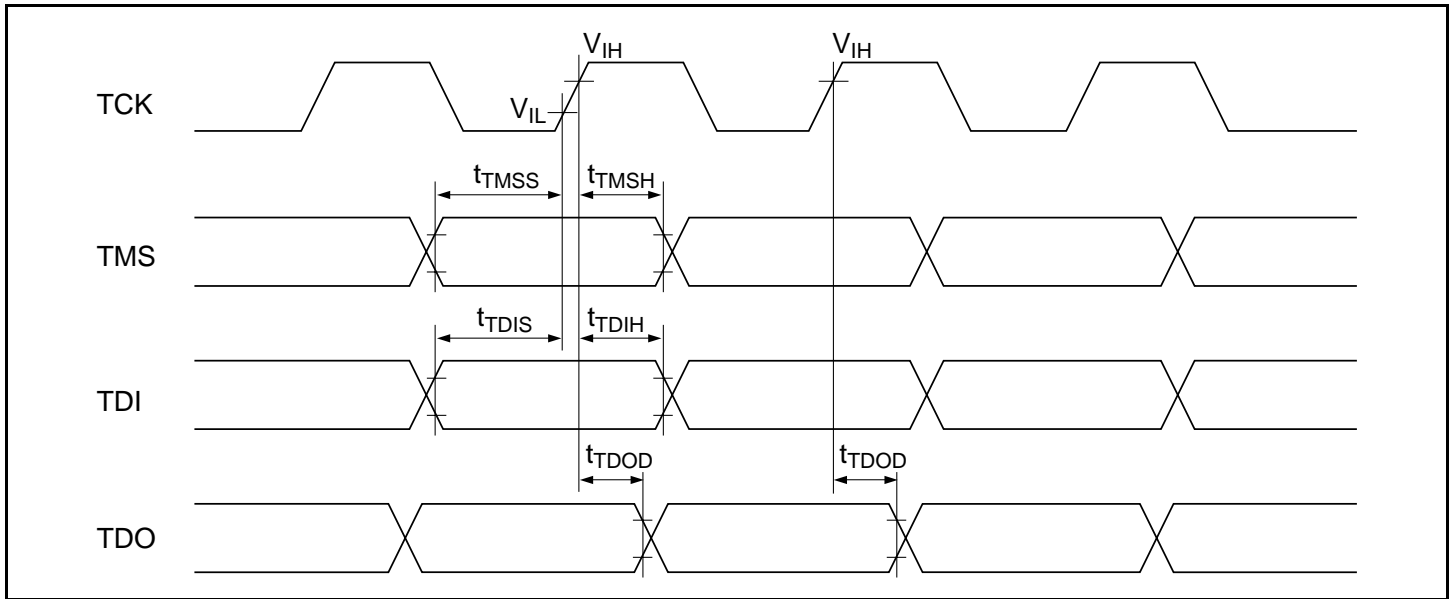
The H-UDI pins constitute a circuit requiring the voltage of  $V_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ . Comply with the input and output voltages specified in the DC characteristics, for operation.



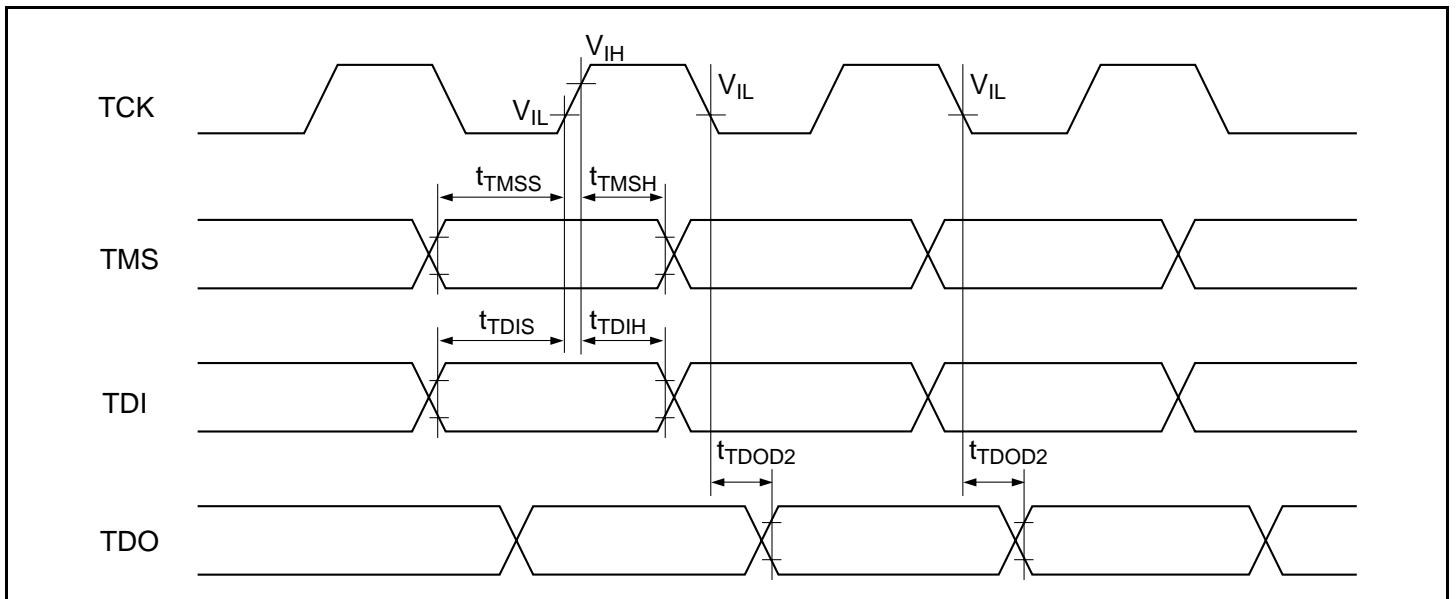
**Figure 27.21 H-UDI Clock Timing**



**Figure 27.22 H-UDI  $\overline{\text{TRST}}$  Timing**



**Figure 27.23 H-UDI Input/Output Timing**



**Figure 27.24 H-UDI Input/Output Timing (Instruction Corresponding to IEEE1149.1 is Executed)**

## 27.3.12 AUD Timing

Table 27.17 shows AUD timing.

**Table 27.17 AUD Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V}/3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .

When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

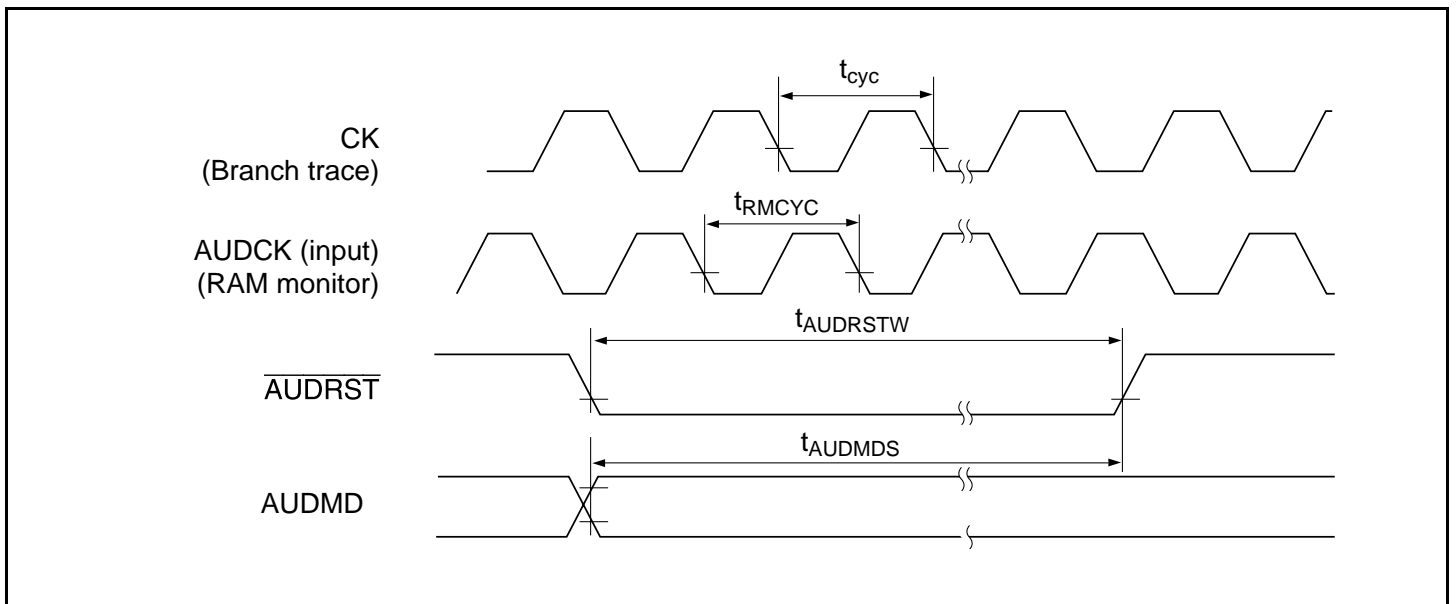
When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

Item	Symbol	Min	Max	Unit	Figures
AUDRST $\bar$ pulse width (Branch trace)	$t_{AUDRSTW}$	10	—	$t_{cyc}$	Figure 27.25
AUDRST $\bar$ pulse width (RAM monitor)	$t_{AUDRSTW}$	5	—	$t_{RMCYC}$	
AUDMD setup time (Branch trace)	$t_{AUDMDS}$	10	—	$t_{cyc}$	
AUDMD setup time (RAM monitor)	$t_{AUDMDS}$	5	—	$t_{RMCYC}$	
Branch trace clock cycle	$t_{BTCYC}$	1	1	$t_{cyc}$	Figure 27.26
Branch trace clock duty	$t_{BTCKW}$	40	60	%	
Branch trace data delay time	$t_{BTDD}$	—	40	ns	
Branch trace data hold time	$t_{BTDH}$	0	—	ns	
Branch trace SYNC delay time	$t_{BTSD}$	—	40	ns	
Branch trace SYNC hold time	$t_{BTSH}$	0	—	ns	
RAM monitor clock cycle	$t_{RMCYC}$	100	—	ns	Figure 27.27
RAM monitor clock low pulse width	$t_{RMCKW}$	45	—	ns	
RAM monitor output data delay time	$t_{RMDD}$	7	$t_{RMCYC} - 20$	ns	
RAM monitor output data hold time	$t_{RMDHD}$	5	—	ns	
RAM monitor input data setup time	$t_{RMDS}$	20	—	ns	
RAM monitor input data hold time	$t_{RMDH}$	5	—	ns	
RAM monitor SYNC setup time	$t_{RMSS}$	20	—	ns	
RAM monitor SYNC hold time	$t_{RMSH}$	5	—	ns	

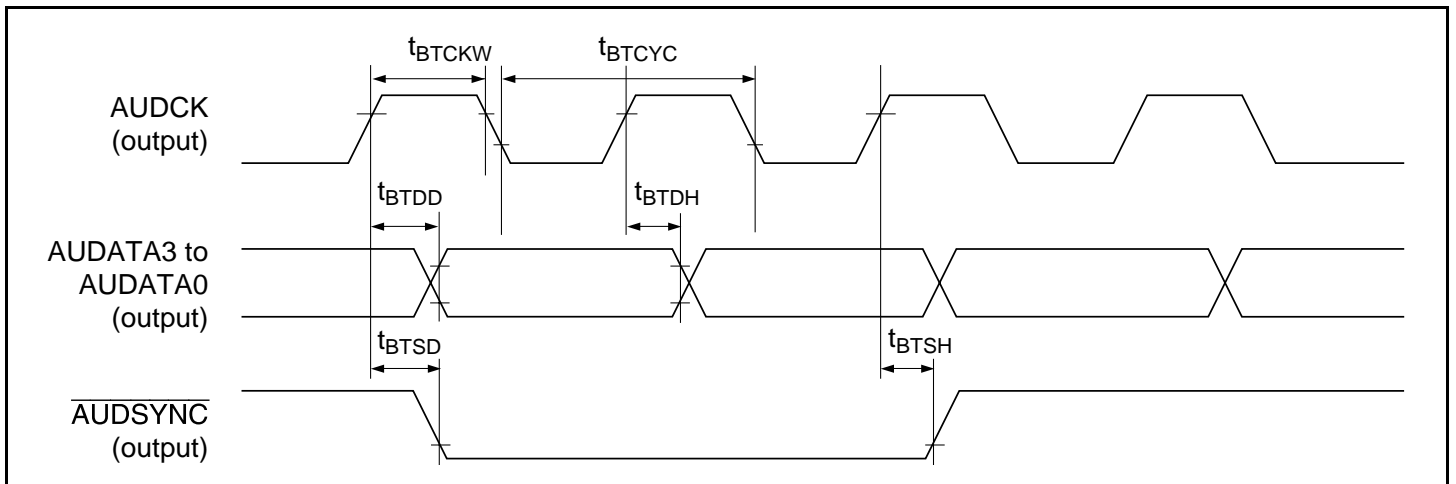
Load conditions: AUDCK (branch trace): CL = 30 pF; otherwise CL = 100 pF

AUDSYNC: CL = 100 pF

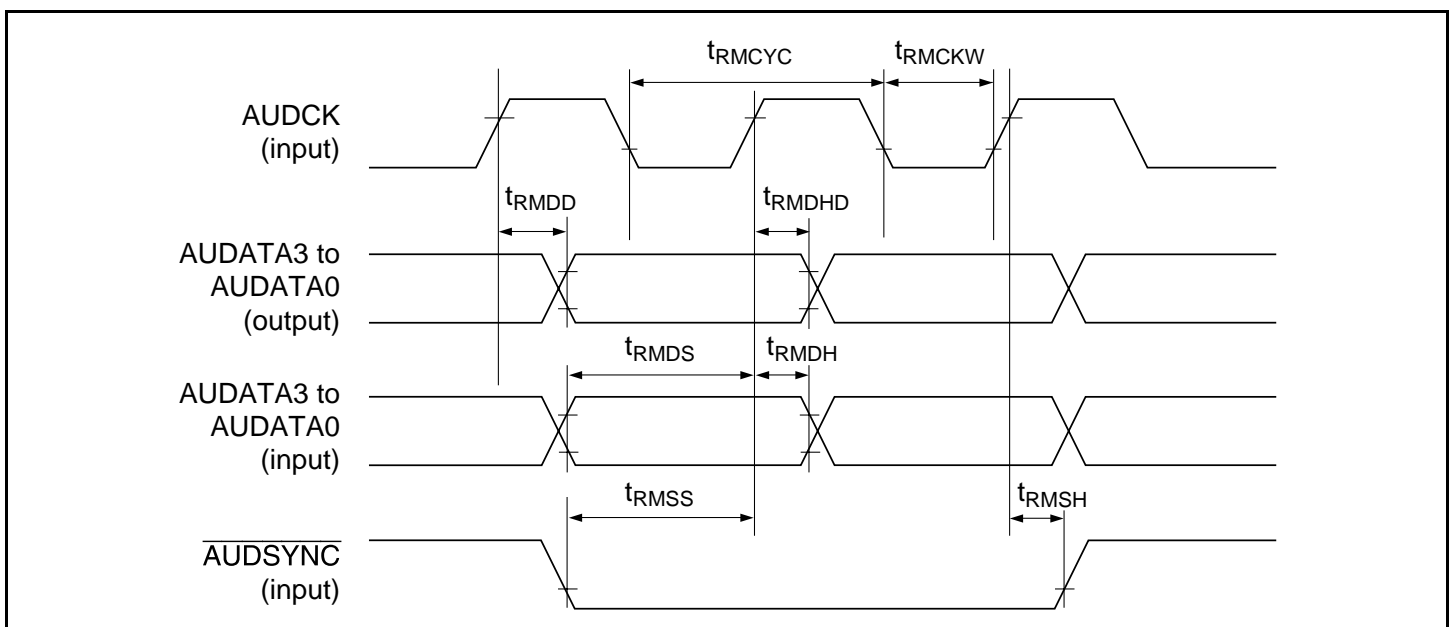
AUDATA3 to AUDATA0: CL = 100 pF



**Figure 27.25 AUD Reset Timing**



**Figure 27.26 Branch Trace Timing**



**Figure 27.27 RAM Monitor Timing**

### 27.3.13 UBC Trigger Timing

Table 27.18 shows UBC trigger timing.

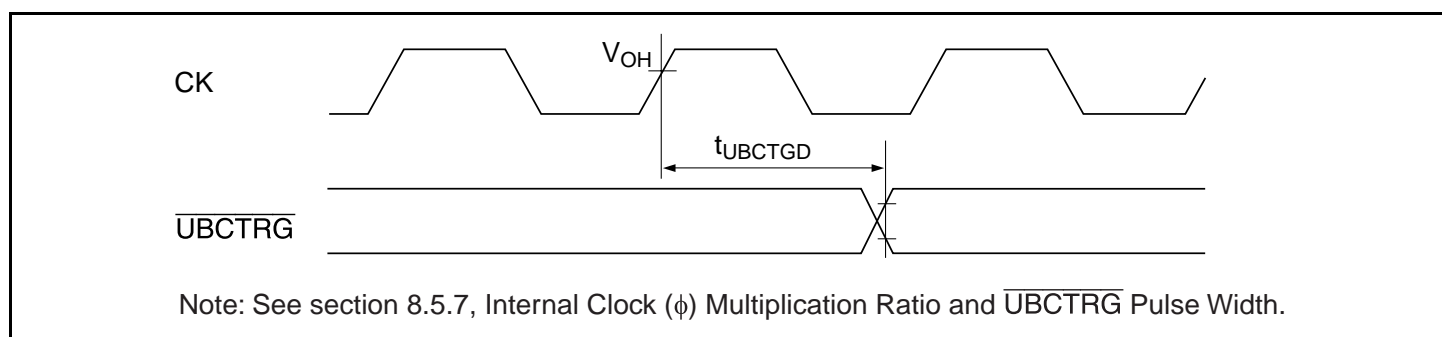
**Table 27.18 UBC Trigger Timing**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V}/3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V to } AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C to } 125^\circ\text{C}$ .

When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C to } 85^\circ\text{C}$ .

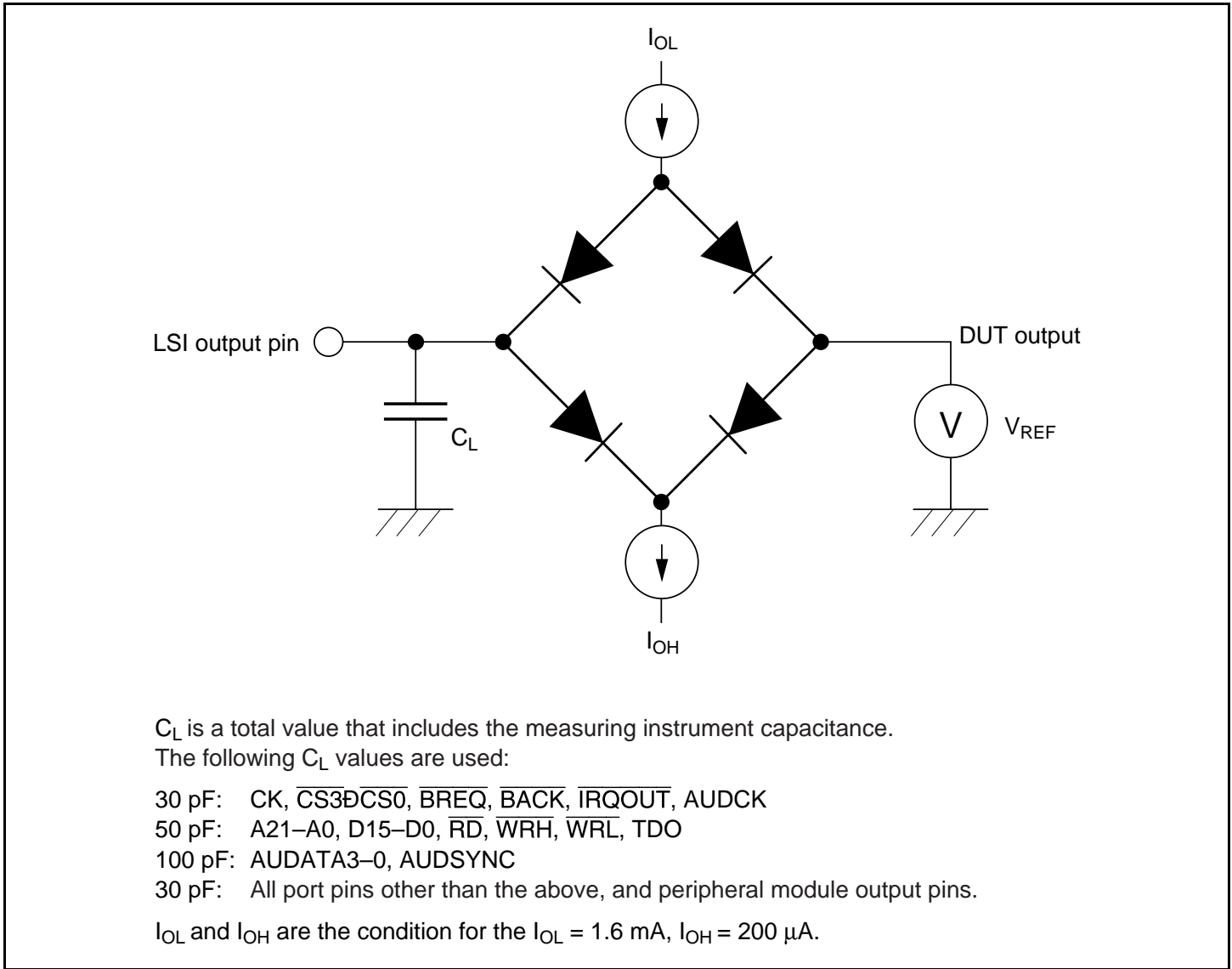
Item	Symbol	Min	Max	Unit	Figures
$\overline{UBCTR\overline{G}}$ delay time	$t_{UBCTGD}$	—	35	ns	Figure 27.28



**Figure 27.28 UBC Trigger Timing**

### 27.3.14 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



**Figure 27.29 Output Test Circuit**



## 27.4 A/D Converter Characteristics

Table 27.19 shows A/D converter characteristics.

**Table 27.19 A/D Converter Characteristics**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $PV_{CC1} = 5.0 \text{ V} \pm 0.5 \text{ V} / 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  
 $PV_{CC2} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{CC} = 5.0 \text{ V} \pm 0.5 \text{ V}$ ,  $AV_{ref} = 4.5 \text{ V to } AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0 \text{ V}$ ,  $T_a = -40^\circ\text{C to } 125^\circ\text{C}$ .

When  $PV_{CC1} = 3.3 \text{ V} \pm 0.3 \text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C to } 85^\circ\text{C}$ .

Item	CSK = 0: fop = 10 to 20 MHz			CSK = 1: fop = 10 MHz			Unit
	Min	Typ	Max	Min	Typ	Max	
Resolution	10	10	10	10	10	10	bit
A/D conversion time	—	—	13.3	—	—	13.4	$\mu\text{s}$
Analog input capacitance	—	—	20	—	—	20	pF
Permitted analog signal source impedance	—	—	3	—	—	3	k $\Omega$
Non-linear error	—	—	$\pm 1.5^{*1}$ $\pm 2.0^{*2}$	—	—	$\pm 1.5^{*1}$ $\pm 2.0^{*2}$	LSB
Offset error	—	—	$\pm 1.5^{*1}$ $\pm 2.0^{*2}$	—	—	$\pm 1.5^{*1}$ $\pm 2.0^{*2}$	LSB
Full-scale error	—	—	$\pm 1.5^{*1}$ $\pm 2.0^{*2}$	—	—	$\pm 1.5^{*1}$ $\pm 2.0^{*2}$	LSB
Quantization error	—	—	$\pm 0.5$	—	—	$\pm 0.5$	LSB
Absolute error	—	—	$\pm 2.0^{*1}$ $\pm 2.5^{*2}$	—	—	$\pm 2.0^{*1}$ $\pm 2.5^{*2}$	LSB

Notes: 1.  $T_a \leq 105^\circ\text{C}$

2.  $T_a > 105^\circ\text{C}$

## 27.5 Flash Memory Characteristics

Table 27.20 shows the flash memory characteristics.

**Table 27.20 Flash Memory Characteristics**

Conditions:  $V_{CC} = PLLV_{CC} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $PV_{CC1} = 5.0\text{ V} \pm 0.5\text{ V}/3.3\text{ V} \pm 0.3\text{ V}$ ,  
 $PV_{CC2} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V} \pm 0.5\text{ V}$ ,  $AV_{ref} = 4.5\text{ V}$  to  $AV_{CC}$ ,  
 $V_{SS} = PLLV_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = -40^\circ\text{C}$  to  $125^\circ\text{C}$ .

When  $PV_{CC1} = 3.3\text{ V} \pm 0.3\text{ V}$ ,  $V_{CC} = PV_{CC1}$ .

When writing or erasing on-chip flash memory,  $T_a = -40^\circ\text{C}$  to  $85^\circ\text{C}$ .

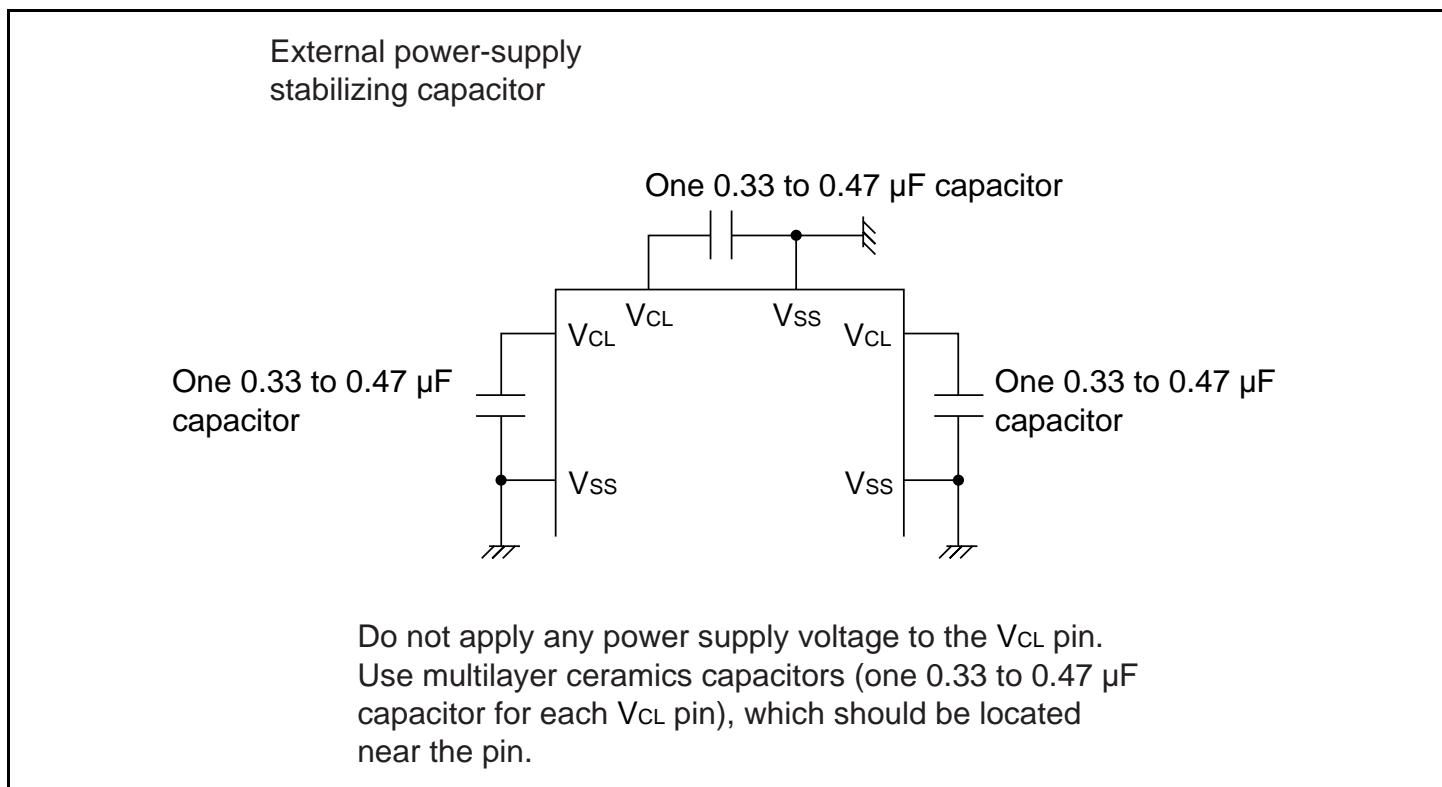
Item	Symbol	Min	Typ	Max	Unit
Programming time* <sup>1</sup> * <sup>2</sup> * <sup>4</sup>	$t_p$	—	3	200	ms/128 bytes
Erase time* <sup>1</sup> * <sup>3</sup> * <sup>5</sup>	$t_E$	—	2	20	s/block
Reprogramming count	$N_{WEC}$	100	—	—	Times

- Notes:
1. Use the on-chip programming/erasing routine for programming/erasure.
  2. When all 0 are programmed.
  3. 128 kbytes of block
  4. The total reprogramming time (programming time + erasing time) is as follows.  
40 s (typ.), reference value: 60 s, 80 s (max.)  
However, 90% of the values are within the reference value.
  5.  $t_p$ ,  $t_E$  distributes focusing on near the typ. value.

## 27.6 Usage Note

### 27.6.1 Notes on Connecting External Capacitor for Current Stabilization

The SH7058 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 ( $V_{CL}$  pin) and the  $V_{SS}$  pin, an capacitor (0.33 to 0.47  $\mu\text{F}$ ) for stabilizing the internal voltage. Connection of the external capacitor is shown in figure 27.30. The external capacitor should be located near the pin. Do not apply any power supply voltage to the  $V_{CL}$  pin.



**Figure 27.30 Connection of  $V_{CL}$  Capacitor**

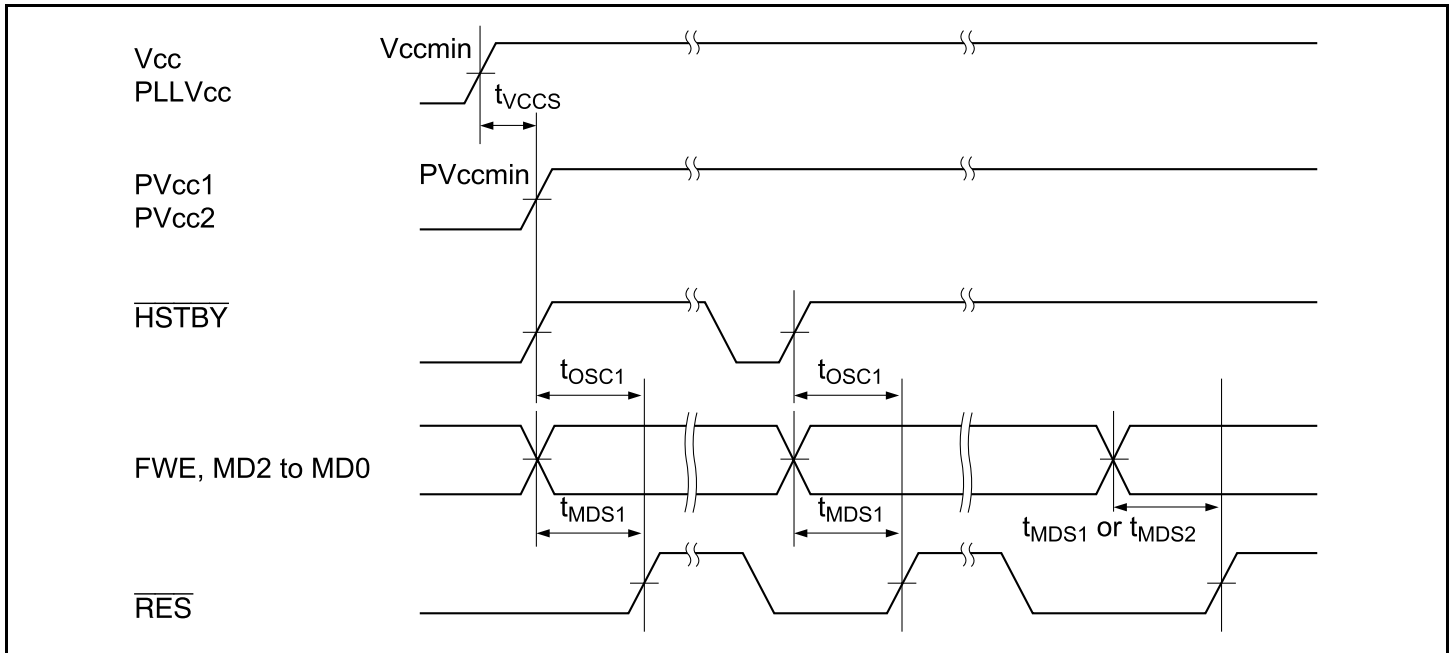
### 27.6.2 Notes on Mode Pin Input

This electrical characteristics are specified for the combination of mode pins (FWE, MD2 to MD0) specified in table 27.3. Characteristics of combinations other than those in table 27.3 cannot be guaranteed.

When power is supplied and in hardware standby mode, mode setup time is determined by  $t_{MDS1}$ . When power-on reset is performed only by the  $\overline{\text{RES}}$  pin, mode setup time is differs according to the combination of input to the FWE and MD2 to MD0. When low is input to the  $\overline{\text{RES}}$  pin with the pins FWE and MD2 to MD0 operated in mode specified in table 27.3, the mode setup time is determined by  $t_{MDS2}$ . When combination which is not specified in table 27.3 is input, the mode setup time is determined by  $t_{MDS1}$ .

**Table 27.21 Mode Pin Input Timing**

Item	Symbol	Min	Typ	Max	Unit	Remark
Mode setup time 1	$t_{MDS1}$	30	—	—	ms	Figure 27.31
Mode setup time 2	$t_{MDS2}$	10	—	—	$t_{cyc}$	



**Figure 27.31 Mode Pin Input Timing**



# Appendix A On-chip peripheral module Registers

## A.1 Address

On-chip peripheral module register addresses and bit names are shown in the following table. 16-bit and 32-bit registers are shown in two and four rows of 8 bits, respectively.

**Table A.1 Address**

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD000	MCR	—	—	—	—	—	—	—	—	HCAN0 (Channel 0)
H'FFFFD001		MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	
H'FFFFD002	GSR	—	—	—	—	—	—	—	—	
H'FFFFD003		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
H'FFFFD004	BCR1	TSEG13	TSEG12	TSEG11	TSEG10	—	TSEG22	TSEG21	TSEG20	
H'FFFFD005		—	—	SJW1	SJW0	—	—	—	BSP	
H'FFFFD006	BCR0	—	—	—	—	—	—	—	—	
H'FFFFD007		BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	
H'FFFFD008	IRR	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8	
H'FFFFD009		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
H'FFFFD00A	IMR	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	
H'FFFFD00B		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0	
H'FFFFD00C	TEC/	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	
H'FFFFD00D	REC	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
H'FFFFD020	TXPR1	TXPR1[15]	TXPR1[14]	TXPR1[13]	TXPR1[12]	TXPR1[11]	TXPR1[10]	TXPR1[9]	TXPR1[8]	
H'FFFFD021		TXPR1[7]	TXPR1[6]	TXPR1[5]	TXPR1[4]	TXPR1[3]	TXPR1[2]	TXPR1[1]	TXPR1[0]	
H'FFFFD022	TXPR0	TXPR0[15]	TXPR0[14]	TXPR0[13]	TXPR0[12]	TXPR0[11]	TXPR0[10]	TXPR0[9]	TXPR0[8]	
H'FFFFD023		TXPR0[7]	TXPR0[6]	TXPR0[5]	TXPR0[4]	TXPR0[3]	TXPR0[2]	TXPR0[1]	—	
H'FFFFD028	TXCR1	TXCR1[15]	TXCR1[14]	TXCR1[13]	TXCR1[12]	TXCR1[11]	TXCR1[10]	TXCR1[9]	TXCR1[8]	
H'FFFFD029		TXCR1[7]	TXCR1[6]	TXCR1[5]	TXCR1[4]	TXCR1[3]	TXCR1[2]	TXCR1[1]	TXCR1[0]	
H'FFFFD02A	TXCR0	TXCR0[15]	TXCR0[14]	TXCR0[13]	TXCR0[12]	TXCR0[11]	TXCR0[10]	TXCR0[9]	TXCR0[8]	
H'FFFFD02B		TXCR0[7]	TXCR0[6]	TXCR0[5]	TXCR0[4]	TXCR0[3]	TXCR0[2]	TXCR0[1]	—	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD030	TXACK1	TXACK1 [15]	TXACK1 [14]	TXACK1 [13]	TXACK1 [12]	TXACK1 [11]	TXACK1 [10]	TXACK1 [9]	TXACK1 [8]	HCAN0 (Channel 0)
H'FFFFD031		TXACK1[7]	TXACK1[6]	TXACK1[5]	TXACK1[4]	TXACK1[3]	TXACK1[2]	TXACK1[1]	TXACK1[0]	
H'FFFFD032	TXACK0	TXACK0 [15]	TXACK0 [14]	TXACK0 [13]	TXACK0 [12]	TXACK0 [11]	TXACK0 [10]	TXACK0 [9]	TXACK0 [8]	
H'FFFFD033		TXACK0[7]	TXACK0[6]	TXACK0[5]	TXACK0[4]	TXACK0[3]	TXACK0[2]	TXACK0[1]	TXACK0[0]	
H'FFFFD038	ABACK1	ABACK1 [15]	ABACK1 [14]	ABACK1 [13]	ABACK1 [12]	ABACK1 [11]	ABACK1 [10]	ABACK1 [9]	ABACK1 [8]	
H'FFFFD039		ABACK1 [7]	ABACK1 [6]	ABACK1 [5]	ABACK1 [4]	ABACK1 [3]	ABACK1 [2]	ABACK1 [1]	ABACK1 [0]	
H'FFFFD03A	ABACK0	ABACK0 [15]	ABACK0 [14]	ABACK0 [13]	ABACK0 [12]	ABACK0 [11]	ABACK0 [10]	ABACK0 [9]	ABACK0 [8]	
H'FFFFD03B		ABACK0 [7]	ABACK0 [6]	ABACK0 [5]	ABACK0 [4]	ABACK0 [3]	ABACK0 [2]	ABACK0 [1]	—	
H'FFFFD040	RXPR1	RXPR1[15]	RXPR1[14]	RXPR1[13]	RXPR1[12]	RXPR1[11]	RXPR1[10]	RXPR1[9]	RXPR1[8]	
H'FFFFD041		RXPR1[7]	RXPR1[6]	RXPR1[5]	RXPR1[4]	RXPR1[3]	RXPR1[2]	RXPR1[1]	RXPR1[0]	
H'FFFFD042	RXPR0	RXPR0[15]	RXPR0[14]	RXPR0[13]	RXPR0[12]	RXPR0[11]	RXPR0[10]	RXPR0[9]	RXPR0[8]	
H'FFFFD043		RXPR0 [7]	RXPR0 [6]	RXPR0[5]	RXPR0[4]	RXPR0 [3]	RXPR0[2]	RXPR0[1]	RXPR0 [0]	
H'FFFFD048	RFPR1	RFPR1 [15]	RFPR1 [14]	RFPR1 [13]	RFPR1 [12]	RFPR1 [11]	RFPR1 [10]	RFPR1[9]	RFPR1[8]	
H'FFFFD049		RFPR1[7]	RFPR1[6]	RFPR1[5]	RFPR1[4]	RFPR1[3]	RFPR1[2]	RFPR1[1]	RFPR1[0]	
H'FFFFD04A	RFPR0	RFPR0 [15]	RFPR0 [14]	RFPR0 [13]	RFPR0 [12]	RFPR0 [11]	RFPR0 [10]	RFPR0 [9]	RFPR0 [8]	
H'FFFFD04B		RFPR0[7]	RFPR0[6]	RFPR0[5]	RFPR0[4]	RFPR0[3]	RFPR0[2]	RFPR0[1]	RFPR0[0]	
H'FFFFD050	MBIMR1	MBIMR1 [15]	MBIMR1 [14]	MBIMR1 [13]	MBIMR1 [12]	MBIMR1 [11]	MBIMR1 [10]	MBIMR1 [9]	MBIMR1 [8]	
H'FFFFD051		MBIMR1[7]	MBIMR1[6]	MBIMR1[5]	MBIMR1[4]	MBIMR1[3]	MBIMR1[2]	MBIMR1[1]	MBIMR1[0]	
H'FFFFD052	MBIMR0	MBIMR0 [15]	MBIMR0 [14]	MBIMR0 [13]	MBIMR0 [12]	MBIMR0 [11]	MBIMR0 [10]	MBIMR0 [9]	MBIMR0 [8]	
H'FFFFD053		MBIMR0[7]	MBIMR0[6]	MBIMR0[5]	MBIMR0[4]	MBIMR0[3]	MBIMR0[2]	MBIMR0[1]	MBIMR0[0]	
H'FFFFD058	UMSR1	UMSR1 [15]	UMSR1 [14]	UMSR1 [13]	UMSR1 [12]	UMSR1 [11]	UMSR1 [10]	UMSR1 [9]	UMSR1 [8]	
H'FFFFD059		UMSR1[7]	UMSR1[6]	UMSR1[5]	UMSR1[4]	UMSR1[3]	UMSR1[2]	UMSR1[1]	UMSR1[0]	
H'FFFFD05A	UMSR0	UMSR0 [15]	UMSR0 [14]	UMSR0 [13]	UMSR0 [12]	UMSR0 [11]	UMSR0 [10]	UMSR0 [9]	UMSR0 [8]	
H'FFFFD05B		UMSR0[7]	UMSR0[6]	UMSR0[5]	UMSR0[4]	UMSR0[3]	UMSR0[2]	UMSR0[1]	UMSR0[0]	
H'FFFFD05C-7F		—	—	—	—	—	—	—	—	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD080	TCNTR	TCNTR15	TCNTR14	TCNTR13	TCNTR12	TCNTR11	TCNTR10	TCNTR9	TCNTR8	HCAN0 (Channel 0)
H'FFFFD081		TCNTR7	TCNTR6	TCNTR5	TCNTR4	TCNTR3	TCNTR2	TCNTR1	TCNTR0	
H'FFFFD082	TCR	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	TCR9	—	
H'FFFFD083		TCR7	—	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0	
H'FFFFD084	TSR	—	—	—	—	—	—	—	—	
H'FFFFD085		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0	
H'FFFFD086	TDCR	TDCR15	TDCR14	TDCR13	TDCR12	TDCR11	TDCR10	TDCR9	TDCR8	
H'FFFFD087		TDCR7	TDCR6	TDCR5	TDCR4	TDCR3	TDCR2	TDCR1	TDCR0	
H'FFFFD088	LOSR	LOSR15	LOSR14	LOSR13	LOSR12	LOSR11	LOSR10	LOSR9	LOSR8	
H'FFFFD089		LOSR7	LOSR6	LOSR5	LOSR4	LOSR3	LOSR2	LOSR1	LOSR0	
H'FFFFD08A	ICR0_cc	—	—	—	—	—	—	—	—	
H'FFFFD08B		—	—	—	—	ICCR0_cc 3	ICCR0_cc 2	ICCR0_cc 1	ICCR0_cc 0	
H'FFFFD08C	ICR0_tm	ICR0_tm	ICR0_tm	ICR0_tm	ICR0_tm	ICR0_tm	ICR0_tm	ICR0_tm9	ICR0_tm8	
H'FFFFD08D		15	14	13	12	11	10			
H'FFFFD08E	ICR1	ICR0_tm7	ICR0_tm6	ICR0_tm5	ICR0_tm4	ICR0_tm3	ICR0_tm2	ICR0_tm1	ICR0_tm0	
H'FFFFD08F		ICR1[15]	ICR1[14]	ICR1[13]	ICR1[12]	ICR1[11]	ICR1[10]	ICR1[9]	ICR1[8]	
H'FFFFD090	TCMR0	ICR1[7]	ICR1[6]	ICR1[5]	ICR1[4]	ICR1[3]	ICR1[2]	ICR1[1]	ICR1[0]	
H'FFFFD091		TCMR0	TCMR0	TCMR0	TCMR0	TCMR0	TCMR0	TCMR0	TCMR0	TCMR0
H'FFFFD092	TCMR1	TCMR0[15]	TCMR0[14]	TCMR0[13]	TCMR0[12]	TCMR0[11]	TCMR0[10]	TCMR0[9]	TCMR0[8]	
H'FFFFD093		TCMR0[7]	TCMR0[6]	TCMR0[5]	TCMR0[4]	TCMR0[3]	TCMR0[2]	TCMR0[1]	TCMR0[0]	
H'FFFFD094	TCMR2	TCMR1	TCMR1	TCMR1	TCMR1	TCMR1	TCMR1	TCMR1	TCMR1	
H'FFFFD095		TCMR1[15]	TCMR1[14]	TCMR1[13]	TCMR1[12]	TCMR1[11]	TCMR1[10]	TCMR1[9]	TCMR1[8]	
H'FFFFD096	CCR	TCMR1[7]	TCMR1[6]	TCMR1[5]	TCMR1[4]	TCMR1[3]	TCMR1[2]	TCMR1[1]	TCMR1[0]	
H'FFFFD097		TCMR2	TCMR2	TCMR2	TCMR2	TCMR2	TCMR2	TCMR2	TCMR2	TCMR2
H'FFFFD098	CMAX	TCMR2[15]	TCMR2[14]	TCMR2[13]	TCMR2[12]	TCMR2[11]	TCMR2[10]	TCMR2[9]	TCMR2[8]	
H'FFFFD099		TCMR2[7]	TCMR2[6]	TCMR2[5]	TCMR2[4]	TCMR2[3]	TCMR2[2]	TCMR2[1]	TCMR2[0]	
H'FFFFD09A	TMR	—	—	—	—	—	—	—	—	
H'FFFFD09B		—	—	—	—	TMR3	TMR2	TMR1	—	
H'FFFFD09C	CCR-buf	—	—	—	—	—	—	—	—	
H'FFFFD09D		—	—	—	—	CCR- buf3	CCR- buf2	CCR- buf1	CCR- buf0	
H'FFFFD09E	ICR0-buf	—	—	—	—	—	—	—	—	
H'FFFFD09F		—	—	—	—	ICR0- buf3	ICR0- buf2	ICR0- buf1	ICR0- buf0	
H'FFFFD0A0-FF	—	—	—	—	—	—	—	—	—	



Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD100	MB0[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD101	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD102	MB0[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD103	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD104	MB0[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD105	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD106	MB0[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD107		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD108	MB0[7],	MSG_DATA_0								
H'FFFFD109	[8]*	MSG_DATA_1								
H'FFFFD10A	MB0[9],	MSG_DATA_2								
H'FFFFD10B	[10]	MSG_DATA_3								
H'FFFFD10C	MBx[11],	MSG_DATA_4								
H'FFFFD10D	[12]	MSG_DATA_5								
H'FFFFD10E	MB0[13],	MSG_DATA_6								
H'FFFFD10F	[14]	MSG_DATA_7								
H'FFFFD110	MB0[15],	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD111	[16]	STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD112	MB0[17],	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD113	[18]	EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD114- 1F	—	—	—	—	—	—	—	—	—	—
H'FFFFD120	MB1[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD121	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD122	MB1[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD123	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD124	MB1[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD125	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD126	MB1[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD127		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD128	MB1[7],	MSG_DATA_0								
H'FFFFD129	[8]*	MSG_DATA_1								
H'FFFFD12A	MB1[9],	MSG_DATA_2								
H'FFFFD12B	[10]	MSG_DATA_3								
H'FFFFD12C	MB1[11],	MSG_DATA_4								
H'FFFFD12D	[12]	MSG_DATA_5								
H'FFFFD12E	MB1[13],	MSG_DATA_6								
H'FFFFD12F	[14]	MSG_DATA_7								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD130	MB1[15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	HCAN0 (Channel 0)
H'FFFFD131		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD132	MB1[17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD133		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD134 -3F		—	—	—	—	—	—	—	—	—
H'FFFFD140	MB2[0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD141		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD142	MB2[2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD143		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD144	MB2[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD145		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD146	MB2[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD147		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD148	MB2[7], [8]*	MSG_DATA_0								
H'FFFFD149		MSG_DATA_1								
H'FFFFD14A	MB2[9], [10]	MSG_DATA_2								
H'FFFFD14B		MSG_DATA_3								
H'FFFFD14C	MB2[11], [12]	MSG_DATA_4								
H'FFFFD14D		MSG_DATA_5								
H'FFFFD14E	MB2[13], [14]	MSG_DATA_6								
H'FFFFD14F		MSG_DATA_7								
H'FFFFD150	MB2[15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD151		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD152	MB2[17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD153		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD154 -5F		—	—	—	—	—	—	—	—	—
H'FFFFD160	MB3[0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD161		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD162	MB3[2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD163		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD164	MB3[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD165		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD166	MB3[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	HCAN0  (Channel 0)
H'FFFFD167		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD168	MB3[7],	MSG_DATA_0								
H'FFFFD169	[8]*	MSG_DATA_1								
H'FFFFD16A	MB3[9],	MSG_DATA_2								
H'FFFFD16B	[10]	MSG_DATA_3								
H'FFFFD16C	MB3[11],	MSG_DATA_4								
H'FFFFD16D	[12]	MSG_DATA_5								
H'FFFFD16E	MB3[13],	MSG_DATA_6								
H'FFFFD16F	[14]	MSG_DATA_7								
H'FFFFD170	MB3[15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD171		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD172	MB3[17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD173		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD174- 7F		—	—	—	—	—	—	—	—	
H'FFFFD180	MB4[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD181	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD182	MB4[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD183	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD184	MB4[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD185	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD186	MB4[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD187		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD188	MB4[7],	MSG_DATA_0								
H'FFFFD189	[8]*	MSG_DATA_1								
H'FFFFD18A	MB4[9],	MSG_DATA_2								
H'FFFFD18B	[10]	MSG_DATA_3								
H'FFFFD18C	MB4[11],	MSG_DATA_4								
H'FFFFD18D	[12]	MSG_DATA_5								
H'FFFFD18E	MB4[13],	MSG_DATA_6								
H'FFFFD18F	[14]	MSG_DATA_7								
H'FFFFD190	MB4[15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD191		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD192	MB4[17, 18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	HCAN0 (Channel 0)
H'FFFFD193		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD194-9F		—	—	—	—	—	—	—	—	
H'FFFFD1A0	MB5[0, 1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD1A1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD1A2	MB5[2, 3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD1A3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD1A4	MB5[4, 5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD1A5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD1A6	MB5[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD1A7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD1A8	MB5 [7, 8]*	MSG_DATA_0								
H'FFFFD1A9		MSG_DATA_1								
H'FFFFD1AA	MB5 [9, 10]	MSG_DATA_2								
H'FFFFD1AB		MSG_DATA_3								
H'FFFFD1AC	MB5[11, 12]	MSG_DATA_4								
H'FFFFD1AD		MSG_DATA_5								
H'FFFFD1AE	MB5[13, 14]	MSG_DATA_6								
H'FFFFD1AF		MSG_DATA_7								
H'FFFFD1B0	MB5[15, 16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD1B1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD1B2	MB5[17, 18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD1B3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD1B4-BF		—	—	—	—	—	—	—	—	
H'FFFFD1C0	MB6[0, 1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD1C1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD1C2	MB6[2, 3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD1C3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD1C4	MB6[4, 5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD1C5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD1C6	MB6[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD1C7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD1C8	MB6[7, 8]*	MSG_DATA_0								
H'FFFFD1C9		MSG_DATA_1								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD1CA	MB6[9],	MSG_DATA_2								HCAN0 (Channel 0)
H'FFFFD1CB	[10]	MSG_DATA_3								
H'FFFFD1CC	MB6[11],	MSG_DATA_4								
H'FFFFD1CD	[12]	MSG_DATA_5								
H'FFFFD1CE	MB6[13],	MSG_DATA_6								
H'FFFFD1CF	[14]	MSG_DATA_7								
H'FFFFD1D0	MB6[15],	—	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	
	[16]		FM[10]	FM[9]	FM[8]	FM[7]	FM[6]	FM[5]	FM[4]	
H'FFFFD1D1		STDID_LA	STDID_LA	STDID_LA	STDID_LA	—	—	EXTID_LA	EXTID_LA	
		FM[3]	FM[2]	FM[1]	FM[0]			FM[17]	FM[16]	
H'FFFFD1D2	MB6[17],	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
	[18]	FM[15]	FM[14]	FM[13]	FM[12]	FM[11]	FM[10]	FM[9]	FM[8]	
H'FFFFD1D3		EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
		FM[7]	FM[6]	FM[5]	FM[4]	FM[3]	FM[2]	FM[1]	FM[0]	
H'FFFFD1D4	-DF	—	—	—	—	—	—	—	—	
H'FFFFD1E0	MB7[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD1E1	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD1E2	MB7[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD1E3	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD1E4	MB7[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD1E5	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD1E6	MB7[6]	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	
		[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	
H'FFFFD1E7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD1E8	MB7[7],	MSG_DATA_0								
H'FFFFD1E9	[8]*	MSG_DATA_1								
H'FFFFD1EA	MB7[9],	MSG_DATA_2								
H'FFFFD1EB	[10]	MSG_DATA_3								
H'FFFFD1EC	MB7[11],	MSG_DATA_4								
H'FFFFD1ED	[12]	MSG_DATA_5								
H'FFFFD1EE	MB7[13],	MSG_DATA_6								
H'FFFFD1EF	[14]	MSG_DATA_7								
H'FFFFD1F0	MB7[15],	—	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	
	[16]		FM[10]	FM[9]	FM[8]	FM[7]	FM[6]	FM[5]	FM[4]	
H'FFFFD1F1		STDID_LA	STDID_LA	STDID_LA	STDID_LA	—	—	EXTID_LA	EXTID_LA	
		FM[3]	FM[2]	FM[1]	FM[0]			FM[17]	FM[16]	
H'FFFFD1F2	MB7[17],	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
	[18]	FM[15]	FM[14]	FM[13]	FM[12]	FM[11]	FM[10]	FM[9]	FM[8]	
H'FFFFD1F3		EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
		FM[7]	FM[6]	FM[5]	FM[4]	FM[3]	FM[2]	FM[1]	FM[0]	
H'FFFFD1F4	-FF	—	—	—	—	—	—	—	—	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD200	MB8[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD201	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD202	MB8[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD203	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD204	MB8[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD205	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD206	MB8[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD207		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD208	MB8[7],	MSG_DATA_0								
H'FFFFD209	[8]*	MSG_DATA_1								
H'FFFFD20A	MB8[9],	MSG_DATA_2								
H'FFFFD20B	[10]	MSG_DATA_3								
H'FFFFD20C	MB8[11],	MSG_DATA_4								
H'FFFFD20D	[12]	MSG_DATA_5								
H'FFFFD20E	MB8[13],	MSG_DATA_6								
H'FFFFD20F	[14]	MSG_DATA_7								
H'FFFFD210	MB8[15],	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD211	[16]	STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD212	MB8[17],	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD213	[18]	EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD214- 1F		—	—	—	—	—	—	—	—	
H'FFFFD220	MB9[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD221	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD222	MB9[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD223	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD224	MB9[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD225	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD226	MB9[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD227		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD228	MB9[7],	MSG_DATA_0								
H'FFFFD229	[8]*	MSG_DATA_1								
H'FFFFD22A	MB9[9],	MSG_DATA_2								
H'FFFFD22B	[10]	MSG_DATA_3								
H'FFFFD22C	MB9[11],	MSG_DATA_4								
H'FFFFD22D	[12]	MSG_DATA_5								
H'FFFFD22E	MB9[13],	MSG_DATA_6								
H'FFFFD22F	[14]	MSG_DATA_7								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD230	MB9[15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	HCAN0 (Channel 0)
H'FFFFD231		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD232	MB9[17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD233		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD234- 3F		—	—	—	—	—	—	—	—	—
H'FFFFD240	MB10[0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD241		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD242	MB10[2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD243		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD244	MB10[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD245		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD246	MB10[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD247		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD248	MB10[7], [8]*	MSG_DATA_0								
H'FFFFD249		MSG_DATA_1								
H'FFFFD24A	MB10[9], [10]	MSG_DATA_2								
H'FFFFD24B		MSG_DATA_3								
H'FFFFD24C	MB10 [11], [12]	MSG_DATA_4								
H'FFFFD24D		MSG_DATA_5								
H'FFFFD24E	MB10 [13],[14]	MSG_DATA_6								
H'FFFFD24F		MSG_DATA_7								
H'FFFFD250	MB10 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD251		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD252	MB10 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD253		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD254- 5F		—	—	—	—	—	—	—	—	—
H'FFFFD260	MB11 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD261		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD262	MB11 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD263		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD264	MB11[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD265		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD266	MB11[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	HCAN0  (Channel 0)
H'FFFFD267		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD268	MB11[7], [8]*	MSG_DATA_0								
H'FFFFD269		MSG_DATA_1								
H'FFFFD26A	MB11[9], [10]	MSG_DATA_2								
H'FFFFD26B		MSG_DATA_3								
H'FFFFD26C	MB11 [11], [12]	MSG_DATA_4								
H'FFFFD26D		MSG_DATA_5								
H'FFFFD26E	MB11 [13], [14]	MSG_DATA_6								
H'FFFFD26F		MSG_DATA_7								
H'FFFFD270	MB11 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD271		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD272	MB11 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD273		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD274- 7F		—	—	—	—	—	—	—	—	—
H'FFFFD280	MB12[0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0  (Channel 0)
H'FFFFD281		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD282	MB12[2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD283		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD284	MB12[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD285		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD286	MB12[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD287		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD288	MB12[7], [8]*	MSG_DATA_0								
H'FFFFD289		MSG_DATA_1								
H'FFFFD28A	MB12[9], [10]	MSG_DATA_2								
H'FFFFD28B		MSG_DATA_3								
H'FFFFD28C	MB12 [11], [12]	MSG_DATA_4								
H'FFFFD28D		MSG_DATA_5								
H'FFFFD28E	MB12 [13], [14]	MSG_DATA_6								
H'FFFFD28F		MSG_DATA_7								
H'FFFFD290	MB12 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD291		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	



Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD292	MB12 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	HCAN0  (Channel 0)
H'FFFFD293		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD294-9F		—	—	—	—	—	—	—	—	
H'FFFFD2A0	MB13 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD2A1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD2A2	MB13 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD2A3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD2A4	MB13[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD2A5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD2A6	MB13[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD2A7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD2A8	MB13[7], [8]*	MSG_DATA_0								
H'FFFFD2A9		MSG_DATA_1								
H'FFFFD2AA	MB13[9], [10]	MSG_DATA_2								
H'FFFFD2AB		MSG_DATA_3								
H'FFFFD2AC	MB13 [11], [12]	MSG_DATA_4								
H'FFFFD2AD		MSG_DATA_5								
H'FFFFD2AE	MB13 [13], [14]	MSG_DATA_6								
H'FFFFD2AF		MSG_DATA_7								
H'FFFFD2B0	MB13 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD2B1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD2B2	MB13 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD2B3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD2B4-BF		—	—	—	—	—	—	—	—	
H'FFFFD2C0	MB14 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0
H'FFFFD2C1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	(Channel 0)
H'FFFFD2C2	MB14 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD2C3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD2C4	MB14[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD2C5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD2C6	MB14[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD2C7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD2C8	MB14[7], [8]*	MSG_DATA_0								
H'FFFFD2C9		MSG_DATA_1								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD2CA	MB14[9],	MSG_DATA_2								HCAN0
H'FFFFD2CB	[10]	MSG_DATA_3								(Channel 0)
H'FFFFD2CC	MB14	MSG_DATA_4								
H'FFFFD2CD	[11], [12]	MSG_DATA_5								
H'FFFFD2CE	MB14	MSG_DATA_6								
H'FFFFD2CF	[13], [14]	MSG_DATA_7								
H'FFFFD2D0	MB14	—	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	
	[15], [16]		FM[10]	FM[9]	FM[8]	FM[7]	FM[6]	FM[5]	FM[4]	
H'FFFFD2D1		STDID_LA	STDID_LA	STDID_LA	STDID_LA	—	—	EXTID_LA	EXTID_LA	
		FM[3]	FM[2]	FM[1]	FM[0]			FM[17]	FM[16]	
H'FFFFD2D2	MB14	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
	[17], [18]	FM[15]	FM[14]	FM[13]	FM[12]	FM[11]	FM[10]	FM[9]	FM[8]	
H'FFFFD2D3		EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
		FM[7]	FM[6]	FM[5]	FM[4]	FM[3]	FM[2]	FM[1]	FM[0]	
H'FFFFD2D4		—	—	—	—	—	—	—	—	—
-DF										
H'FFFFD2E0	MB15	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0
H'FFFFD2E1	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	(Channel 0)
H'FFFFD2E2	MB15	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD2E3	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD2E4	MB15[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD2E5	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD2E6	MB15[6]	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	
		[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	
H'FFFFD2E7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD2E8	MB15[7],	MSG_DATA_0								
H'FFFFD2E9	[8]*	MSG_DATA_1								
H'FFFFD2EA	MB15[9],	MSG_DATA_2								
H'FFFFD2EB	[10]	MSG_DATA_3								
H'FFFFD2EC	MB15	MSG_DATA_4								
H'FFFFD2ED	[11], [12]	MSG_DATA_5								
H'FFFFD2EE	MB15	MSG_DATA_6								
H'FFFFD2EF	[13], [14]	MSG_DATA_7								
H'FFFFD2F0	MB15	—	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	
	[15], [16]		FM[10]	FM[9]	FM[8]	FM[7]	FM[6]	FM[5]	FM[4]	
H'FFFFD2F1		STDID_LA	STDID_LA	STDID_LA	STDID_LA	—	—	EXTID_LA	EXTID_LA	
		FM[3]	FM[2]	FM[1]	FM[0]			FM[17]	FM[16]	
H'FFFFD2F2	MB15	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
	[17], [18]	FM[15]	FM[14]	FM[13]	FM[12]	FM[11]	FM[10]	FM[9]	FM[8]	
H'FFFFD2F3		EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
		FM[7]	FM[6]	FM[5]	FM[4]	FM[3]	FM[2]	FM[1]	FM[0]	
H'FFFFD2F4		—	—	—	—	—	—	—	—	—
-FF										

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD300	MB16	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD301	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD302	MB16	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD303	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD304	MB16[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD305	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD306	MB16[6]	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	
H'FFFFD307		[15] [14]	[13]	[12]	[11]	[10]	[9]	[8]		
H'FFFFD308	MB16[7],	MSG_DATA_0								
H'FFFFD309	[8]*	MSG_DATA_1								
H'FFFFD30A	MB16[9],	MSG_DATA_2								
H'FFFFD30B	[10]	MSG_DATA_3								
H'FFFFD30C	MB16	MSG_DATA_4								
H'FFFFD30D	[11], [12]	MSG_DATA_5								
H'FFFFD30E	MB16	MSG_DATA_6								
H'FFFFD30F	[13], [14]	MSG_DATA_7								
H'FFFFD310	MB16	—	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	
H'FFFFD311	[15], [16]		FM[10]	FM[9]	FM[8]	FM[7]	FM[6]	FM[5]	FM[4]	
H'FFFFD312	MB16	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
H'FFFFD313	[17], [18]	FM[15]	FM[14]	FM[13]	FM[12]	FM[11]	FM[10]	FM[9]	FM[8]	
H'FFFFD314-1F		—	—	—	—	—	—	—	—	
H'FFFFD320	MB17[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD321	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD322	MB17[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD323	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD324	MB17[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD325	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD326	MB17[6]	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	
H'FFFFD327		[15] [14]	[13]	[12]	[11]	[10]	[9]	[8]		
H'FFFFD328	MB17[7],	MSG_DATA_0								
H'FFFFD329	[8]*	MSG_DATA_1								
H'FFFFD32A	MB17[9],	MSG_DATA_2								
H'FFFFD32B	[10]	MSG_DATA_3								
H'FFFFD32C	MB17	MSG_DATA_4								
H'FFFFD32D	[11], [12]	MSG_DATA_5								
H'FFFFD32E	MB17	MSG_DATA_6								
H'FFFFD32F	[13], [14]	MSG_DATA_7								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD330	MB17 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	HCAN0 (Channel 0)
H'FFFFD331		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD332	MB17 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD333		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD334-3F		—	—	—	—	—	—	—	—	
H'FFFFD340	MB18 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD341		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD342	MB18 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD343		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD344	MB18 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD345		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD346	MB18[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD347		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD348	MB18[7], [8]*	MSG_DATA_0								
H'FFFFD349		MSG_DATA_1								
H'FFFFD34A	MB18[9], [10]	MSG_DATA_2								
H'FFFFD34B		MSG_DATA_3								
H'FFFFD34C	MB18 [11], [12]	MSG_DATA_4								
H'FFFFD34D		MSG_DATA_5								
H'FFFFD34E	MB18 [13], [14]	MSG_DATA_6								
H'FFFFD34F		MSG_DATA_7								
H'FFFFD350	MB18 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD351		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD352	MB18 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD353		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD354-5F		—	—	—	—	—	—	—	—	
H'FFFFD360	MB19 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD361		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD362	MB19 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD363		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD364	MB19 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD365		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD366	MB19[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	HCAN0  (Channel 0)
H'FFFFD367		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD368	MB19[7], [8]*	MSG_DATA_0								
H'FFFFD369		MSG_DATA_1								
H'FFFFD36A	MB19[9], [10]	MSG_DATA_2								
H'FFFFD36B		MSG_DATA_3								
H'FFFFD36C	MB19 [11], [12]	MSG_DATA_4								
H'FFFFD36D		MSG_DATA_5								
H'FFFFD36E	MB19 [13], [14]	MSG_DATA_6								
H'FFFFD36F		MSG_DATA_7								
H'FFFFD370	MB19 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD371		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD372	MB19 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD373		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD374- 7F		—	—	—	—	—	—	—	—	—
H'FFFFD380	MB20 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0  (Channel 0)
H'FFFFD381		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD382	MB20 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD383		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD384	MB20[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD385		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD386	MB20[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD387		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD388	MB20[7], [8]*	MSG_DATA_0								
H'FFFFD389		MSG_DATA_1								
H'FFFFD38A	MB20[9], [10]	MSG_DATA_2								
H'FFFFD38B		MSG_DATA_3								
H'FFFFD38C	MB20 [11], [12]	MSG_DATA_4								
H'FFFFD38D		MSG_DATA_5								
H'FFFFD38E	MB20 [13], [14]	MSG_DATA_6								
H'FFFFD38F		MSG_DATA_7								
H'FFFFD390	MB20 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD391		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD392	MB20 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	HCAN0 (Channel 0)
H'FFFFD393		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD394-9F		—	—	—	—	—	—	—	—	—
H'FFFFD3A0	MB21 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD3A1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD3A2	MB21 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD3A3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD3A4	MB21 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD3A5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD3A6	MB21[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD3A7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD3A8	MB21 [7], [8]*	MSG_DATA_0								
H'FFFFD3A9		MSG_DATA_1								
H'FFFFD3AA	MB21 [9], [10]	MSG_DATA_2								
H'FFFFD3AB		MSG_DATA_3								
H'FFFFD3AC	MB21 [11], [12]	MSG_DATA_4								
H'FFFFD3AD		MSG_DATA_5								
H'FFFFD3AE	MB21 [13], [14]	MSG_DATA_6								
H'FFFFD3AF		MSG_DATA_7								
H'FFFFD3B0	MB21 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD3B1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD3B2	MB21 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD3B3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD3B4-BF		—	—	—	—	—	—	—	—	—
H'FFFFD3C0	MB22 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD3C1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD3C2	MB22 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD3C3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD3C4	MB22 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD3C5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD3C6	MB22[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD3C7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD3C8	MB22 [7], [8]*	MSG_DATA_0								
H'FFFFD3C9		MSG_DATA_1								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD3CA	MB22	MSG_DATA_2								HCAN0 (Channel 0)
H'FFFFD3CB	[9], [10]	MSG_DATA_3								
H'FFFFD3CC	MB22	MSG_DATA_4								
H'FFFFD3CD	[11], [12]	MSG_DATA_5								
H'FFFFD3CE	MB22	MSG_DATA_6								
H'FFFFD3CF	[13], [14]	MSG_DATA_7								
H'FFFFD3D0	MB22 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD3D1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD3D2	MB22 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD3D3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD3D4 -DF		—	—	—	—	—	—	—	—	—
H'FFFFD3E0	MB23 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD3E1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD3E2	MB23 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD3E3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD3E4	MB23 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD3E5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD3E6	MB23[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD3E7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD3E8	MB23	MSG_DATA_0								
H'FFFFD3E9	[7], [8]*	MSG_DATA_1								
H'FFFFD3EA	MB23	MSG_DATA_2								
H'FFFFD3EB	[9], [10]	MSG_DATA_3								
H'FFFFD3EC	MB23	MSG_DATA_4								
H'FFFFD3ED	[11], [12]	MSG_DATA_5								
H'FFFFD3EE	MB23	MSG_DATA_6								
H'FFFFD3EF	[13], [14]	MSG_DATA_7								
H'FFFFD3F0	MB23 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD3F1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD3F2	MB23 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD3F3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD3F4 -FF		—	—	—	—	—	—	—	—	—

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD400	MB24	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD401	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD402	MB24	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD403	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD404	MB24	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD405	[4], [5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD406	MB24[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD407		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD408	MB24	MSG_DATA_0								
H'FFFFD409	[7], [8]*	MSG_DATA_1								
H'FFFFD40A	MB24	MSG_DATA_2								
H'FFFFD40B	[9], [10]	MSG_DATA_3								
H'FFFFD40C	MB24	MSG_DATA_4								
H'FFFFD40D	[11], [12]	MSG_DATA_5								
H'FFFFD40E	MB24	MSG_DATA_6								
H'FFFFD40F	[13], [14]	MSG_DATA_7								
H'FFFFD410	MB24 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD411		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD412	MB24 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD413		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD414- 1F		—	—	—	—	—	—	—	—	
H'FFFFD420	MB25	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD421	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD422	MB25	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD423	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD424	MB25	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD425	[4], [5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD426	MB25[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD427		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD428	MB25	MSG_DATA_0								
H'FFFFD429	[7], [8]*	MSG_DATA_1								
H'FFFFD42A	MB25	MSG_DATA_2								
H'FFFFD42B	[9], [10]	MSG_DATA_3								
H'FFFFD42C	MB25	MSG_DATA_4								
H'FFFFD42D	[11], [12]	MSG_DATA_5								
H'FFFFD42E	MB25	MSG_DATA_6								
H'FFFFD42F	[13], [14]	MSG_DATA_7								



Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD430	MB25 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	HCAN0 (Channel 0)
H'FFFFD431		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD432	MB25 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD433		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD434-3F		—	—	—	—	—	—	—	—	
H'FFFFD440	MB26 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD441		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD442	MB26 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD443		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD444	MB26 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD445		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD446	MB26[6] [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]		
H'FFFFD447		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD448	MB26 [7], [8]*	MSG_DATA_0								
H'FFFFD449		MSG_DATA_1								
H'FFFFD44A	MB26 [9], [10]	MSG_DATA_2								
H'FFFFD44B		MSG_DATA_3								
H'FFFFD44C	MB26 [11], [12]	MSG_DATA_4								
H'FFFFD44D		MSG_DATA_5								
H'FFFFD44E	MB26 [13], [14]	MSG_DATA_6								
H'FFFFD44F		MSG_DATA_7								
H'FFFFD450	MB26 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD451		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD452	MB26 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD453		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD454-5F		—	—	—	—	—	—	—	—	
H'FFFFD460	MB27 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0
H'FFFFD461		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	(Channel 0)
H'FFFFD462	MB27 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD463		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD464	MB27 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD465		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD466	MB27[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	HCAN0 (Channel 0)
H'FFFFD467		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD468	MB27	MSG_DATA_0								
H'FFFFD469	[7], [8]*	MSG_DATA_1								
H'FFFFD46A	MB27	MSG_DATA_2								
H'FFFFD46B	[9], [10]	MSG_DATA_3								
H'FFFFD46C	MB27	MSG_DATA_4								
H'FFFFD46D	[11], [12]	MSG_DATA_5								
H'FFFFD46E	MB27	MSG_DATA_6								
H'FFFFD46F	[13], [14]	MSG_DATA_7								
H'FFFFD470	MB27 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD471		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD472	MB27 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD473		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD474- 7F		—	—	—	—	—	—	—	—	—
H'FFFFD480	MB28	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD481	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD482	MB28	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD483	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD484	MB28	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD485	[4], [5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD486	MB28[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD487		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD488	MB28	MSG_DATA_0								
H'FFFFD489	[7], [8]*	MSG_DATA_1								
H'FFFFD48A	MB28	MSG_DATA_2								
H'FFFFD48B	[9], [10]	MSG_DATA_3								
H'FFFFD48C	MB28	MSG_DATA_4								
H'FFFFD48D	[11], [12]	MSG_DATA_5								
H'FFFFD48E	MB28	MSG_DATA_6								
H'FFFFD48F	[13], [14]	MSG_DATA_7								
H'FFFFD490	MB28 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD491		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD492	MB28 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	HCAN0 (Channel 0)
H'FFFFD493		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD494- 7F		—	—	—	—	—	—	—	—	
H'FFFFD4A0	MB29 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD4A1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD4A2	MB29 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD4A3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD4A4	MB29 [4],[5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD4A5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD4A6	MB29[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD4A7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD4A8	MB29 [7],[8]*	MSG_DATA_0								
H'FFFFD4A9		MSG_DATA_1								
H'FFFFD4AA	MB29 [9],[10]	MSG_DATA_2								
H'FFFFD4AB		MSG_DATA_3								
H'FFFFD4AC	MB29 [11],[12]	MSG_DATA_4								
H'FFFFD4AD		MSG_DATA_5								
H'FFFFD4AE	MB29 [13],[14]	MSG_DATA_6								
H'FFFFD4AF		MSG_DATA_7								
H'FFFFD4B0	MB29 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD4B1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD4B2	MB29 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD4B3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD4B4 -BF		—	—	—	—	—	—	—	—	
H'FFFFD4C0	MB30 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0
H'FFFFD4C1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	(Channel 0)
H'FFFFD4C2	MB30 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD4C3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD4C4	MB30 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD4C5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD4C6	MB30[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD4C7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD4C8	MB30 [7], [8]*	MSG_DATA_0								
H'FFFFD4C9		MSG_DATA_1								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD4CA	MB30	MSG_DATA_2								HCAN0 (Channel 0)
H'FFFFD4CB	[9],[10]	MSG_DATA_3								
H'FFFFD4CC	MB30	MSG_DATA_4								
H'FFFFD4CD	[11],[12]	MSG_DATA_5								
H'FFFFD4CE	MB30	MSG_DATA_6								
H'FFFFD4CF	[13],[14]	MSG_DATA_7								
H'FFFFD4D0	MB30	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD4D1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD4D2	MB30	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD4D3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD4D4 -DF		—	—	—	—	—	—	—	—	—
H'FFFFD4E0	MB31	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN0 (Channel 0)
H'FFFFD4E1	[0],[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD4E2	MB31	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD4E3	[2],[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD4E4	MB31	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD4E5	[4],[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD4E6	MB31[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD4E7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD4E8	MB31	MSG_DATA_0								
H'FFFFD4E9	[7],[8]*	MSG_DATA_1								
H'FFFFD4EA	MB31	MSG_DATA_2								
H'FFFFD4EB	[9],[10]	MSG_DATA_3								
H'FFFFD4EC	MB31	MSG_DATA_4								
H'FFFFD4ED	[11],[12]	MSG_DATA_5								
H'FFFFD4EE	MB31	MSG_DATA_6								
H'FFFFD4EF	[13],[14]	MSG_DATA_7								
H'FFFFD4F0	MB31	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD4F1	[15],[16]	STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD4F2	MB31	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD4F3	[17],[18]	EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD4F4 -7FF		—	—	—	—	—	—	—	—	—

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD800	MCR	—	—	—	—	—	—	—	—	HCAN1 (Channel 1)
H'FFFFD801		MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	
H'FFFFD802	GSR	—	—	—	—	—	—	—	—	
H'FFFFD803		—	—	GSR5	GSR4	GSR3	GSR2	GSR1	GSR0	
H'FFFFD804	BCR1	TSEG13	TSEG12	TSEG11	TSEG10	—	TSEG22	TSEG21	TSEG20	
H'FFFFD805		—	—	SJW1	SJW0	—	—	—	BSP	
H'FFFFD806	BCR0	—	—	—	—	—	—	—	—	
H'FFFFD807		BRP7	BRP6	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	
H'FFFFD808	IRR	IRR15	IRR14	IRR13	IRR12	IRR11	IRR10	IRR9	IRR8	
H'FFFFD809		IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0	
H'FFFFD80A	IMR	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	
H'FFFFD80B		IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0	
H'FFFFD80C	TEC/ REC	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	
H'FFFFD80D		REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0	
H'FFFFD820	TXPR1	TXPR1[15]	TXPR1[14]	TXPR1[13]	TXPR1[12]	TXPR1[11]	TXPR1[10]	TXPR1[9]	TXPR1[8]	
H'FFFFD821		TXPR1[7]	TXPR1[6]	TXPR1[5]	TXPR1[4]	TXPR1[3]	TXPR1[2]	TXPR1[1]	TXPR1[0]	
H'FFFFD822	TXPR0	TXPR0[15]	TXPR0[14]	TXPR0[13]	TXPR0[12]	TXPR0[11]	TXPR0[10]	TXPR0[9]	TXPR0[8]	
H'FFFFD823		TXPR0[7]	TXPR0[6]	TXPR0[5]	TXPR0[4]	TXPR0[3]	TXPR0[2]	TXPR0[1]	—	
H'FFFFD828	TXCR1	TXCR1[15]	TXCR1[14]	TXCR1[13]	TXCR1[12]	TXCR1[11]	TXCR1[10]	TXCR1[9]	TXCR1[8]	
H'FFFFD829		TXCR1[7]	TXCR1[6]	TXCR1[5]	TXCR1[4]	TXCR1[3]	TXCR1[2]	TXCR1[1]	TXCR1[0]	
H'FFFFD82A	TXCR0	TXCR0[15]	TXCR0[14]	TXCR0[13]	TXCR0[12]	TXCR0[11]	TXCR0[10]	TXCR0[9]	TXCR0[8]	
H'FFFFD82B		TXCR0[7]	TXCR0[6]	TXCR0[5]	TXCR0[4]	TXCR0[3]	TXCR0[2]	TXCR0[1]	—	
H'FFFFD830	TXACK1	TXACK1 [15]	TXACK1 [14]	TXACK1 [13]	TXACK1 [12]	TXACK1 [11]	TXACK1 [10]	TXACK1 [9]	TXACK1 [8]	
H'FFFFD831		TXACK1[7]	TXACK1[6]	TXACK1[5]	TXACK1[4]	TXACK1[3]	TXACK1[2]	TXACK1[1]	TXACK1[0]	
H'FFFFD832	TXACK0	TXACK0 [15]	TXACK0 [14]	TXACK0 [13]	TXACK0 [12]	TXACK0 [11]	TXACK0 [10]	TXACK0 [9]	TXACK0 [8]	
H'FFFFD833		TXACK0[7]	TXACK0[6]	TXACK0[5]	TXACK0[4]	TXACK0[3]	TXACK0[2]	TXACK0[1]	TXACK0[0]	
H'FFFFD838	ABACK1	ABACK1 [15]	ABACK1 [14]	ABACK1 [13]	ABACK1 [12]	ABACK1 [11]	ABACK1 [10]	ABACK1 [9]	ABACK1 [8]	
H'FFFFD839		ABACK1 [7]	ABACK1 [6]	ABACK1 [5]	ABACK1 [4]	ABACK1 [3]	ABACK1 [2]	ABACK1 [1]	ABACK1 [0]	
H'FFFFD83A	ABACK0	ABACK0 [15]	ABACK0 [14]	ABACK0 [13]	ABACK0 [12]	ABACK0 [11]	ABACK0 [10]	ABACK0 [9]	ABACK0 [8]	
H'FFFFD83B		ABACK0 [7]	ABACK0 [6]	ABACK0 [5]	ABACK0 [4]	ABACK0 [3]	ABACK0 [2]	ABACK0 [1]	—	
H'FFFFD840	RXPR1	RXPR1[15]	RXPR1[14]	RXPR1[13]	RXPR1[12]	RXPR1[11]	RXPR1[10]	RXPR1[9]	RXPR1[8]	
H'FFFFD841		RXPR1[7]	RXPR1[6]	RXPR1[5]	RXPR1[4]	RXPR1[3]	RXPR1[2]	RXPR1[1]	RXPR1[0]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD842	RXPR0	RXPR0[15]	RXPR0[14]	RXPR0[13]	RXPR0[12]	RXPR0[11]	RXPR0[10]	RXPR0[9]	RXPR0[8]	HCAN1 (Channel 1)
H'FFFFD843		RXPR0[7]	RXPR0[6]	RXPR0[5]	RXPR0[4]	RXPR0[3]	RXPR0[2]	RXPR0[1]	RXPR0[0]	
H'FFFFD848	RFPR1	RFPR1[15]	RFPR1[14]	RFPR1[13]	RFPR1[12]	RFPR1[11]	RFPR1[10]	RFPR1[9]	RFPR1[8]	
H'FFFFD849		RFPR1[7]	RFPR1[6]	RFPR1[5]	RFPR1[4]	RFPR1[3]	RFPR1[2]	RFPR1[1]	RFPR1[0]	
H'FFFFD84A	RFPR0	RFPR0[15]	RFPR0[14]	RFPR0[13]	RFPR0[12]	RFPR0[11]	RFPR0[10]	RFPR0[9]	RFPR0[8]	
H'FFFFD84B		RFPR0[7]	RFPR0[6]	RFPR0[5]	RFPR0[4]	RFPR0[3]	RFPR0[2]	RFPR0[1]	RFPR0[0]	
H'FFFFD850	MBIMR1	MBIMR1[15]	MBIMR1[14]	MBIMR1[13]	MBIMR1[12]	MBIMR1[11]	MBIMR1[10]	MBIMR1[9]	MBIMR1[8]	
H'FFFFD851		MBIMR1[7]	MBIMR1[6]	MBIMR1[5]	MBIMR1[4]	MBIMR1[3]	MBIMR1[2]	MBIMR1[1]	MBIMR1[0]	
H'FFFFD852	MBIMR0	MBIMR0[15]	MBIMR0[14]	MBIMR0[13]	MBIMR0[12]	MBIMR0[11]	MBIMR0[10]	MBIMR0[9]	MBIMR0[8]	
H'FFFFD853		MBIMR0[7]	MBIMR0[6]	MBIMR0[5]	MBIMR0[4]	MBIMR0[3]	MBIMR0[2]	MBIMR0[1]	MBIMR0[0]	
H'FFFFD858	UMSR1	UMSR1[15]	UMSR1[14]	UMSR1[13]	UMSR1[12]	UMSR1[11]	UMSR1[10]	UMSR1[9]	UMSR1[8]	
H'FFFFD859		UMSR1[7]	UMSR1[6]	UMSR1[5]	UMSR1[4]	UMSR1[3]	UMSR1[2]	UMSR1[1]	UMSR1[0]	
H'FFFFD85A	UMSR0	UMSR0[15]	UMSR0[14]	UMSR0[13]	UMSR0[12]	UMSR0[11]	UMSR0[10]	UMSR0[9]	UMSR0[8]	
H'FFFFD85B		UMSR0[7]	UMSR0[6]	UMSR0[5]	UMSR0[4]	UMSR0[3]	UMSR0[2]	UMSR0[1]	UMSR0[0]	
H'FFFFD85C-7F		—	—	—	—	—	—	—	—	
H'FFFFD880	TCNTR	TCNTR15	TCNTR14	TCNTR13	TCNTR12	TCNTR11	TCNTR10	TCNTR9	TCNTR8	
H'FFFFD881		TCNTR7	TCNTR6	TCNTR5	TCNTR4	TCNTR3	TCNTR2	TCNTR1	TCNTR0	
H'FFFFD882	TCR	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	TCR9	—	
H'FFFFD883		TCR7	—	TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0	
H'FFFFD884	TSR	—	—	—	—	—	—	—	—	
H'FFFFD885		—	—	—	TSR4	TSR3	TSR2	TSR1	TSR0	
H'FFFFD886	TDCR	TDCR15	TDCR14	TDCR13	TDCR12	TDCR11	TDCR10	TDCR9	TDCR8	
H'FFFFD887		TDCR7	TDCR6	TDCR5	TDCR4	TDCR3	TDCR2	TDCR1	TDCR0	
H'FFFFD888	LOSR	LOSR15	LOSR14	LOSR13	LOSR12	LOSR11	LOSR10	LOSR9	LOSR8	
H'FFFFD889		LOSR7	LOSR6	LOSR5	LOSR4	LOSR3	LOSR2	LOSR1	LOSR0	
H'FFFFD88A	ICR0_cc	—	—	—	—	—	—	—	—	
H'FFFFD88B		—	—	—	—	ICCR0_cc 3	ICCR0_cc 2	ICCR0_cc 1	ICCR0_cc 0	
H'FFFFD88C	ICR0_tm	ICR0_tm 15	ICR0_tm 14	ICR0_tm 13	ICR0_tm 12	ICR0_tm 11	ICR0_tm 10	ICR0_tm9	ICR0_tm8	
H'FFFFD88D		ICR0_tm7	ICR0_tm6	ICR0_tm5	ICR0_tm4	ICR0_tm3	ICR0_tm2	ICR0_tm1	ICR0_tm0	
H'FFFFD88E	ICR1	ICR1[15]	ICR1[14]	ICR1[13]	ICR1[12]	ICR1[11]	ICR1[10]	ICR1[9]	ICR1[8]	
H'FFFFD88F		ICR1[7]	ICR1[6]	ICR1[5]	ICR1[4]	ICR1[3]	ICR1[2]	ICR1[1]	ICR1[0]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD890	TCMR0	TCMR0 [15]	TCMR0 [14]	TCMR0 [13]	TCMR0 [12]	TCMR0 [11]	TCMR0 [10]	TCMR0 [9]	TCMR0 [8]	HCAN1 (Channel 1)
H'FFFFD891		TCMR0[7]	TCMR0[6]	TCMR0[5]	TCMR0[4]	TCMR0[3]	TCMR0[2]	TCMR0[1]	TCMR0[0]	
H'FFFFD892	TCMR1	TCMR1 [15]	TCMR1 [14]	TCMR1 [13]	TCMR1 [12]	TCMR1 [11]	TCMR1 [10]	TCMR1 [9]	TCMR1 [8]	
H'FFFFD893		TCMR1[7]	TCMR1[6]	TCMR1[5]	TCMR1[4]	TCMR1[3]	TCMR1[2]	TCMR1[1]	TCMR1[0]	
H'FFFFD894	TCMR2	TCMR2 [15]	TCMR2 [14]	TCMR2 [13]	TCMR2 [12]	TCMR2 [11]	TCMR2 [10]	TCMR2 [9]	TCMR2 [8]	
H'FFFFD895		TCMR2[7]	TCMR2[6]	TCMR2[5]	TCMR2[4]	TCMR2[3]	TCMR2[2]	TCMR2[1]	TCMR2[0]	
H'FFFFD896	CCR	—	—	—	—	—	—	—	—	
H'FFFFD897		—	—	—	—	CCR3	CCR2	CCR1	CCR0	
H'FFFFD898	CMAx	—	—	—	—	—	—	—	—	
H'FFFFD899		—	—	—	—	CMAx3	CMAx2	CMAx1	CMAx0	
H'FFFFD89A	TMR	—	—	—	—	—	—	—	—	
H'FFFFD89B		—	—	—	—	TMR3	TMR2	TMR1	—	
H'FFFFD89C	CCR- buf	—	—	—	—	—	—	—	—	
H'FFFFD89D		—	—	—	—	CCR- buf3	CCR- buf2	CCR- buf1	CCR- buf0	
H'FFFFD89E	ICR0- buf	—	—	—	—	—	—	—	—	
H'FFFFD89F		—	—	—	—	ICR0- buf3	ICR0- buf2	ICR0- buf1	ICR0- buf0	
H'FFFFD8A0- -FF	—	—	—	—	—	—	—	—	—	—
H'FFFFD900	MB0[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFD901	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD902	MB0[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD903	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD904	MB0[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD905	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD906	MB0[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD907		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD908	MB0[7],	MSG_DATA_0								
H'FFFFD909	[8]*	MSG_DATA_1								
H'FFFFD90A	MB0[9],	MSG_DATA_2								
H'FFFFD90B	[10]	MSG_DATA_3								
H'FFFFD90C	MBx[11],	MSG_DATA_4								
H'FFFFD90D	[12]	MSG_DATA_5								
H'FFFFD90E	MB0[13],	MSG_DATA_6								
H'FFFFD90F	[14]	MSG_DATA_7								
H'FFFFD910	MB0[15],	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD911	[16]	—	STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD912	MB0[17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	HCAN1 (Channel 1)
H'FFFFD913		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD914-1F	—	—	—	—	—	—	—	—	—	—
H'FFFFD920	MB1[0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFD921		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD922	MB1[2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD923		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD924	MB1[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD925		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD926	MB1[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD927		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD928	MB1[7], [8]*	MSG_DATA_0								
H'FFFFD929		MSG_DATA_1								
H'FFFFD92A	MB1[9], [10]	MSG_DATA_2								
H'FFFFD92B		MSG_DATA_3								
H'FFFFD92C	MB1[11], [12]	MSG_DATA_4								
H'FFFFD92D		MSG_DATA_5								
H'FFFFD92E	MB1[13], [14]	MSG_DATA_6								
H'FFFFD92F		MSG_DATA_7								
H'FFFFD930	MB1[15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD931		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD932	MB1[17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD933		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD934-3F	—	—	—	—	—	—	—	—	—	—
H'FFFFD940	MB2[0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFD941		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD942	MB2[2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD943		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD944	MB2[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD945		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD946	MB2[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD947		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD948	MB2[7], [8]*	MSG_DATA_0								
H'FFFFD949		MSG_DATA_1								



Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD94A	MB2[9],	MSG_DATA_2								HCAN1 (Channel 1)
H'FFFFD94B	[10]	MSG_DATA_3								
H'FFFFD94C	MB2[11],	MSG_DATA_4								
H'FFFFD94D	[12]	MSG_DATA_5								
H'FFFFD94E	MB2[13],	MSG_DATA_6								
H'FFFFD94F	[14]	MSG_DATA_7								
H'FFFFD950	MB2[15],	—	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	
H'FFFFD951	[16]		FM[10]	FM[9]	FM[8]	FM[7]	FM[6]	FM[5]	FM[4]	
H'FFFFD952	MB2[17],	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
H'FFFFD953	[18]	FM[15]	FM[14]	FM[13]	FM[12]	FM[11]	FM[10]	FM[9]	FM[8]	
H'FFFFD954-5F		—	—	—	—	—	—	—	—	
H'FFFFD960	MB3[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFD961	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD962	MB3[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD963	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD964	MB3[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD965	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD966	MB3[6]	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	
H'FFFFD967	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]		
H'FFFFD968	MB3[7],	MSG_DATA_0								
H'FFFFD969	[8]*	MSG_DATA_1								
H'FFFFD96A	MB3[9],	MSG_DATA_2								
H'FFFFD96B	[10]	MSG_DATA_3								
H'FFFFD96C	MB3[11],	MSG_DATA_4								
H'FFFFD96D	[12]	MSG_DATA_5								
H'FFFFD96E	MB3[13],	MSG_DATA_6								
H'FFFFD96F	[14]	MSG_DATA_7								
H'FFFFD970	MB3[15],	—	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	
H'FFFFD971	[16]		FM[10]	FM[9]	FM[8]	FM[7]	FM[6]	FM[5]	FM[4]	
H'FFFFD972	MB3[17],	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
H'FFFFD973	[18]	FM[15]	FM[14]	FM[13]	FM[12]	FM[11]	FM[10]	FM[9]	FM[8]	
H'FFFFD974-7F		—	—	—	—	—	—	—	—	
H'FFFFD980	MB4[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD981	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD982	MB4[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	HCAN1 (Channel 1)
H'FFFFD983	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD984	MB4[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD985	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD986	MB4[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD987		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD988	MB4[7],	MSG_DATA_0								
H'FFFFD989	[8]*	MSG_DATA_1								
H'FFFFD98A	MB4[9],	MSG_DATA_2								
H'FFFFD98B	[10]	MSG_DATA_3								
H'FFFFD98C	MB4[11],	MSG_DATA_4								
H'FFFFD98D	[12]	MSG_DATA_5								
H'FFFFD98E	MB4[13],	MSG_DATA_6								
H'FFFFD98F	[14]	MSG_DATA_7								
H'FFFFD990	MB4[15],	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD991	[16]	STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD992	MB4[17],	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD993	[18]	EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD994-9F		—	—	—	—	—	—	—	—	
H'FFFFD9A0	MB5[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFD9A1	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD9A2	MB5[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD9A3	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD9A4	MB5[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD9A5	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD9A6	MB5[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD9A7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD9A8	MB5 [7],	MSG_DATA_0								
H'FFFFD9A9	[8]*	MSG_DATA_1								
H'FFFFD9AA	MB5 [9],	MSG_DATA_2								
H'FFFFD9AB	[10]	MSG_DATA_3								
H'FFFFD9AC	MB5[11],	MSG_DATA_4								
H'FFFFD9AD	[12]	MSG_DATA_5								
H'FFFFD9AE	MB5[13],	MSG_DATA_6								
H'FFFFD9AF	[14]	MSG_DATA_7								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD9B0	MB5[15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	HCAN1 (Channel 1)
H'FFFFD9B1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD9B2	MB5[17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD9B3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD9B4 -BF		—	—	—	—	—	—	—	—	—
H'FFFFD9C0	MB6[0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFD9C1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD9C2	MB6[2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD9C3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD9C4	MB6[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD9C5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFD9C6	MB6[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFD9C7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD9C8	MB6[7], [8]*	MSG_DATA_0								
H'FFFFD9C9		MSG_DATA_1								
H'FFFFD9CA	MB6[9], [10]	MSG_DATA_2								
H'FFFFD9CB		MSG_DATA_3								
H'FFFFD9CC	MB6[11], [12]	MSG_DATA_4								
H'FFFFD9CD		MSG_DATA_5								
H'FFFFD9CE	MB6[13], [14]	MSG_DATA_6								
H'FFFFD9CF		MSG_DATA_7								
H'FFFFD9D0	MB6[15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD9D1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD9D2	MB6[17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD9D3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD9D4 -DF		—	—	—	—	—	—	—	—	—
H'FFFFD9E0	MB7[0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFD9E1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFD9E2	MB7[2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFD9E3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFD9E4	MB7[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFD9E5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFD9E6	MB7[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	HCAN1  (Channel 1)
H'FFFFD9E7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFD9E8	MB7[7],	MSG_DATA_0								
H'FFFFD9E9	[8]*	MSG_DATA_1								
H'FFFFD9EA	MB7[9],	MSG_DATA_2								
H'FFFFD9EB	[10]	MSG_DATA_3								
H'FFFFD9EC	MB7[11],	MSG_DATA_4								
H'FFFFD9ED	[12]	MSG_DATA_5								
H'FFFFD9EE	MB7[13],	MSG_DATA_6								
H'FFFFD9EF	[14]	MSG_DATA_7								
H'FFFFD9F0	MB7[15],	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFD9F1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFD9F2	MB7[17],	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFD9F3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFD9F4	-FF	—	—	—	—	—	—	—	—	—
H'FFFFDA00	MB8[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1
H'FFFFDA01	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	(Channel 1)
H'FFFFDA02	MB8[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDA03	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDA04	MB8[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDA05	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDA06	MB8[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDA07		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDA08	MB8[7],	MSG_DATA_0								
H'FFFFDA09	[8]*	MSG_DATA_1								
H'FFFFDA0A	MB8[9],	MSG_DATA_2								
H'FFFFDA0B	[10]	MSG_DATA_3								
H'FFFFDA0C	MB8[11],	MSG_DATA_4								
H'FFFFDA0D	[12]	MSG_DATA_5								
H'FFFFDA0E	MB8[13],	MSG_DATA_6								
H'FFFFDA0F	[14]	MSG_DATA_7								
H'FFFFDA10	MB8[15],	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDA11		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDA12	MB8[17, 18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	HCAN1 (Channel 1)
H'FFFFDA13		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDA14-1F		—	—	—	—	—	—	—	—	
H'FFFFDA20	MB9[0, 1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFDA21		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDA22	MB9[2, 3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDA23		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDA24	MB9[4, 5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDA25		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDA26	MB9[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDA27		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDA28	MB9[7, 8]*	MSG_DATA_0								
H'FFFFDA29		MSG_DATA_1								
H'FFFFDA2A	MB9[9, 10]	MSG_DATA_2								
H'FFFFDA2B		MSG_DATA_3								
H'FFFFDA2C	MB9[11, 12]	MSG_DATA_4								
H'FFFFDA2D		MSG_DATA_5								
H'FFFFDA2E	MB9[13, 14]	MSG_DATA_6								
H'FFFFDA2F		MSG_DATA_7								
H'FFFFDA30	MB9[15, 16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDA31		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDA32	MB9[17, 18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDA33		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDA34-3F		—	—	—	—	—	—	—	—	
H'FFFFDA40	MB10[0, 1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFDA41		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDA42	MB10[2, 3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDA43		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDA44	MB10[4, 5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDA45		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDA46	MB10[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDA47		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDA48	MB10[7, 8]*	MSG_DATA_0								
H'FFFFDA49		MSG_DATA_1								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDA4A	MB10[9],	MSG_DATA_2								HCAN1 (Channel 1)
H'FFFFDA4B	[10]	MSG_DATA_3								
H'FFFFDA4C	MB10	MSG_DATA_4								
H'FFFFDA4D	[11], [12]	MSG_DATA_5								
H'FFFFDA4E	MB10	MSG_DATA_6								
H'FFFFDA4F	[13],[14]	MSG_DATA_7								
H'FFFFDA50	MB10 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDA51		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDA52	MB10 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDA53		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDA54 -5F		—	—	—	—	—	—	—	—	—
H'FFFFDA60	MB11	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFDA61	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDA62	MB11	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDA63	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDA64	MB11[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDA65	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDA66	MB11[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDA67		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDA68	MB11[7],	MSG_DATA_0								
H'FFFFDA69	[8]*	MSG_DATA_1								
H'FFFFDA6A	MB11[9],	MSG_DATA_2								
H'FFFFDA6B	[10]	MSG_DATA_3								
H'FFFFDA6C	MB11	MSG_DATA_4								
H'FFFFDA6D	[11], [12]	MSG_DATA_5								
H'FFFFDA6E	MB11	MSG_DATA_6								
H'FFFFDA6F	[13], [14]	MSG_DATA_7								
H'FFFFDA70	MB11 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDA71		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDA72	MB11 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDA73		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDA74 -7F		—	—	—	—	—	—	—	—	—

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDA80	MB12[0],	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFDA81	[1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDA82	MB12[2],	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDA83	[3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDA84	MB12[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDA85	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDA86	MB12[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDA87		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDA88	MB12[7],	MSG_DATA_0								
H'FFFFDA89	[8]*	MSG_DATA_1								
H'FFFFDA8A	MB12[9],	MSG_DATA_2								
H'FFFFDA8B	[10]	MSG_DATA_3								
H'FFFFDA8C	MB12	MSG_DATA_4								
H'FFFFDA8D	[11], [12]	MSG_DATA_5								
H'FFFFDA8E	MB12	MSG_DATA_6								
H'FFFFDA8F	[13], [14]	MSG_DATA_7								
H'FFFFDA90	MB12 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDA91		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDA92	MB12 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDA93		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDA94 -9F		—	—	—	—	—	—	—	—	
H'FFFFDAA0	MB13	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFDAA1	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDAA2	MB13	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDAA3	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDAA4	MB13[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDAA5	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDAA6	MB13[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDAA7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDAA8	MB13[7],	MSG_DATA_0								
H'FFFFDAA9	[8]*	MSG_DATA_1								
H'FFFFDAAA	MB13[9],	MSG_DATA_2								
H'FFFFDAAB	[10]	MSG_DATA_3								
H'FFFFDAAC	MB13	MSG_DATA_4								
H'FFFFDAAD	[11], [12]	MSG_DATA_5								
H'FFFFDAAE	MB13	MSG_DATA_6								
H'FFFFDAAF	[13], [14]	MSG_DATA_7								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDAB0	MB13 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	HCAN1 (Channel 1)
H'FFFFDAB1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDAB2	MB13 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	HCAN1 (Channel 1)
H'FFFFDAB3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDAB4 -BF		—	—	—	—	—	—	—	—	—
H'FFFFDAC0	MB14 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFDAC1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDAC2	MB14 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	HCAN1 (Channel 1)
H'FFFFDAC3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDAC4	MB14[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	HCAN1 (Channel 1)
H'FFFFDAC5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDAC6	MB14[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	HCAN1 (Channel 1)
H'FFFFDAC7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDAC8	MB14[7], [8]*	MSG_DATA_0								
H'FFFFDAC9		MSG_DATA_1								
H'FFFFDAC A	MB14[9], [10]	MSG_DATA_2								
H'FFFFDAC B		MSG_DATA_3								
H'FFFFDAC C	MB14 [11], [12]	MSG_DATA_4								
H'FFFFDAC D		MSG_DATA_5								
H'FFFFDAC E	MB14 [13], [14]	MSG_DATA_6								
H'FFFFDACF		MSG_DATA_7								
H'FFFFDAD0	MB14 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	HCAN1 (Channel 1)
H'FFFFDAD1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDAD2	MB14 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	HCAN1 (Channel 1)
H'FFFFDAD3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDAD4 -DF		—	—	—	—	—	—	—	—	—
H'FFFFDAE0	MB15 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFDAE1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDAE2	MB15 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	HCAN1 (Channel 1)
H'FFFFDAE3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDAE4	MB15[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	HCAN1 (Channel 1)
H'FFFFDAE5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	



Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDAE6	MB15[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	HCAN1 (Channel 1)
H'FFFFDAE7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDAE8	MB15[7],	MSG_DATA_0								
H'FFFFDAE9	[8]*	MSG_DATA_1								
H'FFFFDAEA	MB15[9],	MSG_DATA_2								
H'FFFFDAEB	[10]	MSG_DATA_3								
H'FFFFDAEC	MB15 [11], [12]	MSG_DATA_4								
H'FFFFDAED		MSG_DATA_5								
H'FFFFDAEE	MB15	MSG_DATA_6								
H'FFFFDAEF	[13], [14]	MSG_DATA_7								
H'FFFFDAF0	MB15 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDAF1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDAF2	MB15 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDAF3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDAF4-FF		—	—	—	—	—	—	—	—	—
H'FFFFDB00	MB16	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1
H'FFFFDB01	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	(Channel 1)
H'FFFFDB02	MB16	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDB03	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDB04	MB16[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDB05	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDB06	MB16[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDB07		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDB08	MB16[7],	MSG_DATA_0								
H'FFFFDB09	[8]*	MSG_DATA_1								
H'FFFFDB0A	MB16[9],	MSG_DATA_2								
H'FFFFDB0B	[10]	MSG_DATA_3								
H'FFFFDB0C	MB16	MSG_DATA_4								
H'FFFFDB0D	[11], [12]	MSG_DATA_5								
H'FFFFDB0E	MB16	MSG_DATA_6								
H'FFFFDB0F	[13], [14]	MSG_DATA_7								
H'FFFFDB10	MB16 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDB11		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFDB12	MB16 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	HCAN1 (Channel 1)
H'FFFDB13		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFDB14 -1F		—	—	—	—	—	—	—	—	—
H'FFFDB20	MB17[0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFDB21		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFDB22	MB17[2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFDB23		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFDB24	MB17[4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFDB25		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFDB26	MB17[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFDB27		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFDB28	MB17[7], [8]*	MSG_DATA_0								
H'FFFDB29		MSG_DATA_1								
H'FFFDB2A	MB17[9], [10]	MSG_DATA_2								
H'FFFDB2B		MSG_DATA_3								
H'FFFDB2C	MB17 [11], [12]	MSG_DATA_4								
H'FFFDB2D		MSG_DATA_5								
H'FFFDB2E	MB17 [13], [14]	MSG_DATA_6								
H'FFFDB2F		MSG_DATA_7								
H'FFFDB30	MB17 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFDB31		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFDB32	MB17 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFDB33		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFDB34 -3F		—	—	—	—	—	—	—	—	—
H'FFFDB40	MB18 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFDB41		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFDB42	MB18 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFDB43		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFDB44	MB18 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFDB45		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFDB46	MB18[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFDB47		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFDB48	MB18[7], [8]*	MSG_DATA_0								
H'FFFDB49		MSG_DATA_1								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDB4A	MB18[9],	MSG_DATA_2								HCAN1 (Channel 1)
H'FFFFDB4B	[10]	MSG_DATA_3								
H'FFFFDB4C	MB18	MSG_DATA_4								
H'FFFFDB4D	[11], [12]	MSG_DATA_5								
H'FFFFDB4E	MB18	MSG_DATA_6								
H'FFFFDB4F	[13], [14]	MSG_DATA_7								
H'FFFFDB50	MB18 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDB51		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDB52	MB18 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDB53		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDB54 -5F		—	—	—	—	—	—	—	—	
H'FFFFDB60	MB19	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFDB61	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDB62	MB19	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDB63	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDB64	MB19	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDB65	[4], [5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDB66	MB19[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDB67		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDB68	MB19[7],	MSG_DATA_0								
H'FFFFDB69	[8]*	MSG_DATA_1								
H'FFFFDB6A	MB19[9],	MSG_DATA_2								
H'FFFFDB6B	[10]	MSG_DATA_3								
H'FFFFDB6C	MB19	MSG_DATA_4								
H'FFFFDB6D	[11], [12]	MSG_DATA_5								
H'FFFFDB6E	MB19	MSG_DATA_6								
H'FFFFDB6F	[13], [14]	MSG_DATA_7								
H'FFFFDB70	MB19 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDB71		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDB72	MB19 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDB73		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDB74 -7F		—	—	—	—	—	—	—	—	

Register Name	Abbreviation	Bit Names								Module	
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
H'FFFDB80	MB20	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)	
H'FFFDB81	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]		
H'FFFDB82	MB20	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]		
H'FFFDB83	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]		
H'FFFDB84	MB20[4],	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]		
H'FFFDB85	[5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]		
H'FFFDB86	MB20[6]	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP		
H'FFFDB87	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]			
H'FFFDB88	MB20[7],	MSG_DATA_0									
H'FFFDB89	[8]*	MSG_DATA_1									
H'FFFDB8A	MB20[9],	MSG_DATA_2									
H'FFFDB8B	[10]	MSG_DATA_3									
H'FFFDB8C	MB20	MSG_DATA_4									
H'FFFDB8D	[11], [12]	MSG_DATA_5									
H'FFFDB8E	MB20	MSG_DATA_6									
H'FFFDB8F	[13], [14]	MSG_DATA_7									
H'FFFDB90	MB20	—	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA		
H'FFFDB91	[15], [16]	FM[10]	FM[9]	FM[8]	FM[7]	FM[6]	FM[5]	FM[4]			
H'FFFDB92	MB20	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA		
H'FFFDB93	[17], [18]	FM[15]	FM[14]	FM[13]	FM[12]	FM[11]	FM[10]	FM[9]	FM[8]		
H'FFFDB94		EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA		
-9F		FM[7]	FM[6]	FM[5]	FM[4]	FM[3]	FM[2]	FM[1]	FM[0]		
H'FFFDBA0	MB21	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)	
H'FFFDBA1	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]		
H'FFFDBA2	MB21	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]		
H'FFFDBA3	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]		
H'FFFDBA4	MB21	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]		
H'FFFDBA5	[4], [5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]		
H'FFFDBA6	MB21[6]	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP		
H'FFFDBA7	[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]			
H'FFFDBA8	MB21	MSG_DATA_0									
H'FFFDBA9	[7], [8]*	MSG_DATA_1									
H'FFFDBAA	MB21	MSG_DATA_2									
H'FFFDBAB	[9], [10]	MSG_DATA_3									
H'FFFDBAC	MB21	MSG_DATA_4									
H'FFFDBAD	[11], [12]	MSG_DATA_5									
H'FFFDBAE	MB21	MSG_DATA_6									
H'FFFDBAF	[13], [14]	MSG_DATA_7									

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDBB0	MB21 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	HCAN1 (Channel 1)
H'FFFFDBB1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDBB2	MB21 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDBB3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDBB4 -BF		—	—	—	—	—	—	—	—	—
H'FFFFDBC0	MB22 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFDBC1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDBC2	MB22 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDBC3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDBC4	MB22 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDBC5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDBC6	MB22[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDBC7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDBC8	MB22 [7], [8]*	MSG_DATA_0								
H'FFFFDBC9		MSG_DATA_1								
H'FFFFDBC A	MB22 [9], [10]	MSG_DATA_2								
H'FFFFDBC B		MSG_DATA_3								
H'FFFFDBC C	MB22 [11], [12]	MSG_DATA_4								
H'FFFFDBC D		MSG_DATA_5								
H'FFFFDBC E	MB22 [13], [14]	MSG_DATA_6								
H'FFFFDBCF		MSG_DATA_7								
H'FFFFDBD0	MB22 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDBD1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDBD2	MB22 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDBD3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDBD4 -DF		—	—	—	—	—	—	—	—	—
H'FFFFDBE0	MB23 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFDBE1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDBE2	MB23 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDBE3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDBE4	MB23 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDBE5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFDBE6	MB23[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	HCAN1 (Channel 1)
H'FFFDBE7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFDBE8	MB23 [7], [8]*	MSG_DATA_0								
H'FFFDBE9		MSG_DATA_1								
H'FFFDBEA	MB23 [9], [10]	MSG_DATA_2								
H'FFFDBEB		MSG_DATA_3								
H'FFFDBEC	MB23 [11], [12]	MSG_DATA_4								
H'FFFDBED		MSG_DATA_5								
H'FFFDBEE	MB23 [13], [14]	MSG_DATA_6								
H'FFFDBEF		MSG_DATA_7								
H'FFFDBF0	MB23 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFDBF1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFDBF2	MB23 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFDBF3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFDBF4-FF		—	—	—	—	—	—	—	—	—
H'FFFDC00	MB24 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFDC01		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFDC02	MB24 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFDC03		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFDC04	MB24 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFDC05		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFDC06	MB24[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFDC07		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFDC08	MB24 [7], [8]*	MSG_DATA_0								
H'FFFDC09		MSG_DATA_1								
H'FFFDC0A	MB24 [9], [10]	MSG_DATA_2								
H'FFFDC0B		MSG_DATA_3								
H'FFFDC0C	MB24 [11], [12]	MSG_DATA_4								
H'FFFDC0D		MSG_DATA_5								
H'FFFDC0E	MB24 [13], [14]	MSG_DATA_6								
H'FFFDC0F		MSG_DATA_7								
H'FFFDC10	MB24 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFDC11		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDC12	MB24 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	HCAN1 (Channel 1)
H'FFFFDC13		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDC14 -1F		—	—	—	—	—	—	—	—	
H'FFFFDC20	MB25 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFDC21		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDC22	MB25 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDC23		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDC24	MB25 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDC25		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDC26	MB25[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDC27		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDC28	MB25 [7], [8]*	MSG_DATA_0								
H'FFFFDC29		MSG_DATA_1								
H'FFFFDC2A	MB25 [9], [10]	MSG_DATA_2								
H'FFFFDC2B		MSG_DATA_3								
H'FFFFDC2C	MB25 [11], [12]	MSG_DATA_4								
H'FFFFDC2D		MSG_DATA_5								
H'FFFFDC2E	MB25 [13], [14]	MSG_DATA_6								
H'FFFFDC2F		MSG_DATA_7								
H'FFFFDC30	MB25 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDC31		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDC32	MB25 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDC33		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDC34 -3F		—	—	—	—	—	—	—	—	
H'FFFFDC40	MB26 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFDC41		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDC42	MB26 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDC43		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDC44	MB26 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDC45		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDC46	MB26[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDC47		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDC48	MB26 [7], [8]*	MSG_DATA_0								
H'FFFFDC49		MSG_DATA_1								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDC4A	MB26	MSG_DATA_2								HCAN1 (Channel 1)
H'FFFFDC4B	[9], [10]	MSG_DATA_3								
H'FFFFDC4C	MB26	MSG_DATA_4								
H'FFFFDC4D	[11], [12]	MSG_DATA_5								
H'FFFFDC4E	MB26	MSG_DATA_6								
H'FFFFDC4F	[13], [14]	MSG_DATA_7								
H'FFFFDC50	MB26	—	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA
	[15], [16]		FM[10]	FM[9]	FM[8]	FM[7]	FM[6]	FM[5]	FM[4]	
H'FFFFDC51		STDID_LA	STDID_LA	STDID_LA	STDID_LA	—	—	EXTID_LA	EXTID_LA	
		FM[3]	FM[2]	FM[1]	FM[0]			FM[17]	FM[16]	
H'FFFFDC52	MB26	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
	[17], [18]	FM[15]	FM[14]	FM[13]	FM[12]	FM[11]	FM[10]	FM[9]	FM[8]	
H'FFFFDC53		EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
		FM[7]	FM[6]	FM[5]	FM[4]	FM[3]	FM[2]	FM[1]	FM[0]	
H'FFFFDC54		—	—	—	—	—	—	—	—	—
-5F										
H'FFFFDC60	MB27	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFDC61	[0], [1]*	STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDC62	MB27	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDC63	[2], [3]	EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDC64	MB27	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDC65	[4], [5]	—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDC66	MB27[6]	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	TMSTP	
		[15]	[14]	[13]	[12]	[11]	[10]	[9]	[8]	
H'FFFFDC67		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDC68	MB27	MSG_DATA_0								
H'FFFFDC69	[7], [8]*	MSG_DATA_1								
H'FFFFDC6A	MB27	MSG_DATA_2								
H'FFFFDC6B	[9], [10]	MSG_DATA_3								
H'FFFFDC6C	MB27	MSG_DATA_4								
H'FFFFDC6D	[11], [12]	MSG_DATA_5								
H'FFFFDC6E	MB27	MSG_DATA_6								
H'FFFFDC6F	[13], [14]	MSG_DATA_7								
H'FFFFDC70	MB27	—	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	STDID_LA	
	[15], [16]		FM[10]	FM[9]	FM[8]	FM[7]	FM[6]	FM[5]	FM[4]	
H'FFFFDC71		STDID_LA	STDID_LA	STDID_LA	STDID_LA	—	—	EXTID_LA	EXTID_LA	
		FM[3]	FM[2]	FM[1]	FM[0]			FM[17]	FM[16]	
H'FFFFDC72	MB27	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
	[17], [18]	FM[15]	FM[14]	FM[13]	FM[12]	FM[11]	FM[10]	FM[9]	FM[8]	
H'FFFFDC73		EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	EXTID_LA	
		FM[7]	FM[6]	FM[5]	FM[4]	FM[3]	FM[2]	FM[1]	FM[0]	
H'FFFFDC74		—	—	—	—	—	—	—	—	—
-7F										



Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDC80	MB28 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFFDC81		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDC82	MB28 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDC83		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDC84	MB28 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDC85		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDC86	MB28[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDC87		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDC88	MB28 [7], [8]*	MSG_DATA_0								
H'FFFFDC89		MSG_DATA_1								
H'FFFFDC8A	MB28 [9], [10]	MSG_DATA_2								
H'FFFFDC8B		MSG_DATA_3								
H'FFFFDC8C	MB28 [11], [12]	MSG_DATA_4								
H'FFFFDC8D		MSG_DATA_5								
H'FFFFDC8E	MB28 [13], [14]	MSG_DATA_6								
H'FFFFDC8F		MSG_DATA_7								
H'FFFFDC90	MB28 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDC91		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDC92	MB28 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDC93		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDC94 -7F		—	—	—	—	—	—	—	—	
H'FFFFDCA0	MB29 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	
H'FFFFDCA1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFFDCA2	MB29 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFFDCA3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFFDCA4	MB29 [4],[5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFFDCA5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFFDCA6	MB29[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFFDCA7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDCA8	MB29 [7],[8]*	MSG_DATA_0								
H'FFFFDCA9		MSG_DATA_1								
H'FFFFDCA A	MB29 [9],[10]	MSG_DATA_2								
H'FFFFDCA B		MSG_DATA_3								
H'FFFFDCA C	MB29 [11],[12]	MSG_DATA_4								
H'FFFFDCA D		MSG_DATA_5								
H'FFFFDCA E	MB29 [13],[14]	MSG_DATA_6								
H'FFFFDCAF		MSG_DATA_7								

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFDCB0	MB29 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	HCAN1 (Channel 1)
H'FFFDCB1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFDCB2	MB29 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFDCB3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFDCB4 -BF		—	—	—	—	—	—	—	—	—
H'FFFDCC0	MB30 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFDCC1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFDCC2	MB30 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFDCC3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFDCC4	MB30 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFDCC5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	
H'FFFDCC6	MB30[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	
H'FFFDCC7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFDCC8	MB30 [7], [8]*	MSG_DATA_0								
H'FFFDCC9		MSG_DATA_1								
H'FFFDCCA	MB30 [9],[10]	MSG_DATA_2								
H'FFFDCCB		MSG_DATA_3								
H'FFFDCCC	MB30 [11],[12]	MSG_DATA_4								
H'FFFDCCD		MSG_DATA_5								
H'FFFDCC E	MB30 [13],[14]	MSG_DATA_6								
H'FFFDCC F		MSG_DATA_7								
H'FFFD CD0	MB30 [15], [16]	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFD CD1		STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFD CD2	MB30 [17], [18]	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFD CD3		EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFD CD4 -DF		—	—	—	—	—	—	—	—	—
H'FFFD CE0	MB31 [0], [1]*	—	STDID[10]	STDID[9]	STDID[8]	STDID[7]	STDID[6]	STDID[5]	STDID[4]	HCAN1 (Channel 1)
H'FFFD CE1		STDID[3]	STDID[2]	STDID[1]	STDID[0]	RTR	IDE	EXTID[17]	EXTID[16]	
H'FFFD CE2	MB31 [2], [3]	EXTID[15]	EXTID[14]	EXTID[13]	EXTID[12]	EXTID[11]	EXTID[10]	EXTID[9]	EXTID[8]	
H'FFFD CE3		EXTID[7]	EXTID[6]	EXTID[5]	EXTID[4]	EXTID[3]	EXTID[2]	EXTID[1]	EXTID[0]	
H'FFFD CE4	MB31 [4], [5]	CCM	TTE	NMC	ATX	DART	MBC[2]	MBC[1]	MBC[0]	
H'FFFD CE5		—	TCT	—	CLE	DLC[3]	DLC[2]	DLC[1]	DLC[0]	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFDCE6	MB31[6]	TMSTP [15]	TMSTP [14]	TMSTP [13]	TMSTP [12]	TMSTP [11]	TMSTP [10]	TMSTP [9]	TMSTP [8]	HCAN1 (Channel 1)
H'FFFFDCE7		TMSTP[7]	TMSTP[6]	TMSTP[5]	TMSTP[4]	TMSTP[3]	TMSTP[2]	TMSTP[1]	TMSTP[0]	
H'FFFFDCE8	MB31	MSG_DATA_0								
H'FFFFDCE9	[7], [8]*	MSG_DATA_1								
H'FFFFDCEA	MB31	MSG_DATA_2								
H'FFFFDCEB	[9], [10]	MSG_DATA_3								
H'FFFFDCEC	MB31	MSG_DATA_4								
H'FFFFDCE	[11], [12]	MSG_DATA_5								
H'FFFFDCEE	MB31	MSG_DATA_6								
H'FFFFDCEF	[13], [14]	MSG_DATA_7								
H'FFFFDCF0	MB31	—	STDID_LA FM[10]	STDID_LA FM[9]	STDID_LA FM[8]	STDID_LA FM[7]	STDID_LA FM[6]	STDID_LA FM[5]	STDID_LA FM[4]	
H'FFFFDCF1	[15], [16]	STDID_LA FM[3]	STDID_LA FM[2]	STDID_LA FM[1]	STDID_LA FM[0]	—	—	EXTID_LA FM[17]	EXTID_LA FM[16]	
H'FFFFDCF2	MB31	EXTID_LA FM[15]	EXTID_LA FM[14]	EXTID_LA FM[13]	EXTID_LA FM[12]	EXTID_LA FM[11]	EXTID_LA FM[10]	EXTID_LA FM[9]	EXTID_LA FM[8]	
H'FFFFDCF3	[17], [18]	EXTID_LA FM[7]	EXTID_LA FM[6]	EXTID_LA FM[5]	EXTID_LA FM[4]	EXTID_LA FM[3]	EXTID_LA FM[2]	EXTID_LA FM[1]	EXTID_LA FM[0]	
H'FFFFDCF4	-7FF	—	—	—	—	—	—	—	—	—
H'FFFFE730		—	—	—	—	—	—	—	—	—
H'FFFFE77F		—	—	—	—	—	—	—	—	—
H'FFFFE800	FCCS	FWE	—	—	FLER	—	—	—	SCO	FLASH
H'FFFFE801	FPCS	—	—	—	—	—	—	—	PPVS	
H'FFFFE802	FECS	—	—	—	—	—	—	—	EPVB	
H'FFFFE803	—	System area (access prohibited)								
H'FFFFE804	FKEY	K7	K6	K5	K4	K3	K2	K1	K0	
H'FFFFE805	FMATS	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
H'FFFFE806	FTDAR	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0	
H'FFFFE807		System area (access prohibited)								
H'FFFFEBFF	—	System area (access prohibited)								
H'FFFFEC00	UBARH	UBA31	UBA30	UBA29	UBA28	UBA27	UBA26	UBA25	UBA24	UBC
H'FFFFEC01		UBA23	UBA22	UBA21	UBA20	UBA19	UBA18	UBA17	UBA16	
H'FFFFEC02	UBARL	UBA15	UBA14	UBA13	UBA12	UBA11	UBA10	UBA9	UBA8	
H'FFFFEC03		UBA7	UBA6	UBA5	UBA4	UBA3	UBA2	UBA1	UBA0	
H'FFFFEC04	UBAMR	UBM31	UBM30	UBM29	UBM28	UBM27	UBM26	UBM25	UBM24	
H'FFFFEC05	H	UBM23	UBM22	UBM21	UBM20	UBM19	UBM18	UBM17	UBM16	
H'FFFFEC06	UBAMRL	UBM15	UBM14	UBM13	UBM12	UBM11	UBM10	UBM9	UBM8	
H'FFFFEC07		UBM7	UBM6	UBM5	UBM4	UBM3	UBM2	UBM1	UBM0	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFEC08	UBBR	—	—	—	—	—	—	—	—	UBC
H'FFFFEC09		CP1	CP0	ID1	ID0	RW1	RW0	SZ1	SZ0	
H'FFFFEC0A	UBCR	—	—	—	—	—	—	—	—	
H'FFFFEC0B		—	—	—	—	—	CKS1	CKS0	UBID	
H'FFFFEC0C to H'FFFFEC0F	—	—	—	—	—	—	—	—	—	—
H'FFFFEC10	TCSR*	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0	WDT
H'FFFFEC11	TCNT*									
H'FFFFEC12	—	—	—	—	—	—	—	—	—	
H'FFFFEC13	RSTCSR*	WOVF	RSTE	RSTS	—	—	—	—	—	
H'FFFFEC14	SBYCR	SSBY	HIZ	—	—	—	—	—	—	Power-Down state
H'FFFFEC15 to H'FFFFEC1F	—	—	—	—	—	—	—	—	—	—
H'FFFFEC20	BCR1	—	—	—	—	—	—	—	—	BSC
H'FFFFEC21		—	—	—	—	A3SZ	A2SZ	A1SZ	A0SZ	
H'FFFFEC22	BCR2	IW31	IW30	IW21	IW20	IW11	IW10	IW01	IW00	
H'FFFFEC23		CW3	CW2	CW1	CW0	SW3	SW2	SW1	SW0	
H'FFFFEC24	WCR	—	W32	W31	W30	—	W22	W21	W20	
H'FFFFEC25		—	W12	W11	W10	—	W02	W01	W00	
H'FFFFEC26	RAMER	—	—	—	—	—	—	—	—	
H'FFFFEC27		—	—	—	—	RAMS	RAM2	RAM1	RAM0	
H'FFFFEC28 to H'FFFFECAAF	—	—	—	—	—	—	—	—	—	—

Note: \* This is the read address. The Write Address is H'FFFFEC10 for TCSR and TCNT, and H'FFFFEC12 for RSTCSR. For details, see section 13.2.4, Notes on Register Access.

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFECB0	DMAOR	—	—	—	—	—	—	—	—	DMAC (Common)
H'FFFECB1		—	—	—	—	—	AE	NMIF	DME	
H'FFFECB2 to H'FFFECBF	—	—	—	—	—	—	—	—	—	—
H'FFFEC0	SAR0	—	—	—	—	—	—	—	—	DMAC (Channel 0)
H'FFFEC1		—	—	—	—	—	—	—	—	
H'FFFEC2		—	—	—	—	—	—	—	—	
H'FFFEC3		—	—	—	—	—	—	—	—	
H'FFFEC4	DAR0	—	—	—	—	—	—	—	—	
H'FFFEC5		—	—	—	—	—	—	—	—	
H'FFFEC6		—	—	—	—	—	—	—	—	
H'FFFEC7		—	—	—	—	—	—	—	—	
H'FFFEC8	DMATCR0	—	—	—	—	—	—	—	—	
H'FFFEC9		—	—	—	—	—	—	—	—	
H'FFFECA		—	—	—	—	—	—	—	—	
H'FFFECB		—	—	—	—	—	—	—	—	
H'FFFEC0	CHCR0	—	—	—	DI	—	—	—	RO	
H'FFFEC1		—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFEC2		—	—	SM1	SM0	—	—	DM1	DM0	
H'FFFEC3		—	—	TS1	TS0	TM	IE	TE	DE	
H'FFFEC4	SAR1	—	—	—	—	—	—	—	—	DMAC (Channel 1)
H'FFFEC5		—	—	—	—	—	—	—	—	
H'FFFEC6		—	—	—	—	—	—	—	—	
H'FFFEC7		—	—	—	—	—	—	—	—	
H'FFFEC8	DAR1	—	—	—	—	—	—	—	—	
H'FFFEC9		—	—	—	—	—	—	—	—	
H'FFFEDA		—	—	—	—	—	—	—	—	
H'FFFECB		—	—	—	—	—	—	—	—	
H'FFFEC0	DMATCR1	—	—	—	—	—	—	—	—	
H'FFFEC1		—	—	—	—	—	—	—	—	
H'FFFEDA		—	—	—	—	—	—	—	—	
H'FFFECB		—	—	—	—	—	—	—	—	
H'FFFEC0	CHCR1	—	—	—	DI	—	—	—	RO	
H'FFFEC1		—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFEC2		—	—	SM1	SM0	—	—	DM1	DM0	
H'FFFEC3		—	—	TS1	TS0	TM	IE	TE	DE	
H'FFFEC4	SAR2	—	—	—	—	—	—	—	—	DMAC (Channel 2)
H'FFFEC5		—	—	—	—	—	—	—	—	
H'FFFEC6		—	—	—	—	—	—	—	—	
H'FFFEC7		—	—	—	—	—	—	—	—	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFCE4	DAR2									DMAC (Channel 2)
H'FFFCE5										
H'FFFCE6										
H'FFFCE7										
H'FFFCE8	DMATCR2	—	—	—	—	—	—	—	—	
H'FFFCE9										
H'FFFCEA										
H'FFFCEB										
H'FFFCEC	CHCR2	—	—	—	DI	—	—	—	RO	
H'FFFCECD		—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFCEE		—	—	SM1	SM0	—	—	DM1	DM0	
H'FFFCEEF		—	—	TS1	TS0	TM	IE	TE	DE	
H'FFFCEF0	SAR3									DMAC (Channel 3)
H'FFFCEF1										
H'FFFCEF2										
H'FFFCEF3										
H'FFFCEF4	DAR3									
H'FFFCEF5										
H'FFFCEF6										
H'FFFCEF7										
H'FFFCEF8	DMATCR3	—	—	—	—	—	—	—	—	
H'FFFCEF9										
H'FFFCEFA										
H'FFFCEFB										
H'FFFCEFC	CHCR3	—	—	—	DI	—	—	—	RO	
H'FFFCEFD		—	—	—	RS4	RS3	RS2	RS1	RS0	
H'FFFCEFE		—	—	SM1	SM0	—	—	DM1	DM0	
H'FFFCEFF		—	—	TS1	TS0	TM	IE	TE	DE	
H'FFFED00	IPRA									INTC
H'FFFED01										
H'FFFED02	IPRB									
H'FFFED03										
H'FFFED04	IPRC									
H'FFFED05										
H'FFFED06	IPRD									
H'FFFED07										
H'FFFED08	IPRE									
H'FFFED09										
H'FFFED0A	IPRF									
H'FFFED0B										
H'FFFED0C	IPRG									
H'FFFED0D										

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFED0E	IPRH									INTC
H'FFFFED0F										
H'FFFFED10	IPRI									
H'FFFFED11										
H'FFFFED12	IPRJ									
H'FFFFED13										
H'FFFFED14	IPRK									
H'FFFFED15										
H'FFFFED16	IPRL									
H'FFFFED17										
H'FFFFED18	ICR	NMIL	—	—	—	—	—	—	NMIE	
H'FFFFED19		IRQ0S	IRQ1S	IRQ2S	IRQ3S	IRQ4S	IRQ5S	IRQ6S	IRQ7S	
H'FFFFED1A	ISR	—	—	—	—	—	—	—	—	
H'FFFFED1B		IRQ0F	IRQ1F	IRQ2F	IRQ3F	IRQ4F	IRQ5F	IRQ6F	IRQ7F	
H'FFFFED1C to H'FFFFEFFF	—	—	—	—	—	—	—	—	—	—
H'FFFFF000	SMR0	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI
H'FFFFF001	BRR0									(Channel 0)
H'FFFFF002	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFFF003	TDR0									
H'FFFFF004	SSR0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFFF005	RDR0									
H'FFFFF006	SDCR0	—	—	—	—	DIR	—	—	—	
H'FFFFF007	—	—	—	—	—	—	—	—	—	
H'FFFFF008	SMR1	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI
H'FFFFF009	BRR1									(Channel 1)
H'FFFFF00A	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFFF00B	TDR1									
H'FFFFF00C	SSR1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFFF00D	RDR1									
H'FFFFF00E	SDCR1	—	—	—	—	DIR	—	—	—	
H'FFFFF00F	—	—	—	—	—	—	—	—	—	
H'FFFFF010	SMR2	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI
H'FFFFF011	BRR2									(Channel 2)
H'FFFFF012	SCR2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFFF013	TDR2									
H'FFFFF014	SSR2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFFF015	RDR2									
H'FFFFF016	SDCR2	—	—	—	—	DIR	—	—	—	
H'FFFFF017	—	—	—	—	—	—	—	—	—	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF018	SMR3	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI
H'FFFFFF019	BRR3									(Channel 3)
H'FFFFFF01A	SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFFFF01B	TDR3									
H'FFFFFF01C	SSR3	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFFFF01D	RDR3									
H'FFFFFF01E	SDCR3	—	—	—	—	DIR	—	—	—	
H'FFFFFF01F	—	—	—	—	—	—	—	—	—	
H'FFFFFF020	SMR4	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0	SCI
H'FFFFFF021	BRR4									(Channel 4)
H'FFFFFF022	SCR4	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFFFFF023	TDR4									
H'FFFFFF024	SSR4	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'FFFFFF025	RDR4									
H'FFFFFF026	SDCR4	—	—	—	—	DIR	—	—	—	
H'FFFFFF027 to H'FFFFFF3FF	—	—	—	—	—	—	—	—	—	—
H'FFFFFF400	TSTR2	STR7D	STR7C	STR7B	STR7A	STR6D	STR6C	STR6B	STR6A	ATU-II
H'FFFFFF401	TSTR1	STR10	STR5	STR4	STR3	STR1B,2B	STR2A	STR1A	STR0	(Common)
H'FFFFFF402	TSTR3	—	—	—	—	—	—	—	STR11	
H'FFFFFF403	—	—	—	—	—	—	—	—	—	
H'FFFFFF404	PSCR1	—	—	—	PSC1E	PSC1D	PSC1C	PSC1B	PSC1A	
H'FFFFFF405	—	—	—	—	—	—	—	—	—	
H'FFFFFF406	PSCR2	—	—	—	PSC2E	PSC2D	PSC2C	PSC2B	PSC2A	
H'FFFFFF407	—	—	—	—	—	—	—	—	—	
H'FFFFFF408	PSCR3	—	—	—	PSC3E	PSC3D	PSC3C	PSC3B	PSC3A	
H'FFFFFF409	—	—	—	—	—	—	—	—	—	
H'FFFFFF40A	PSCR4	—	—	—	PSC4E	PSC4D	PSC4C	PSC4B	PSC4A	
H'FFFFFF40B	—	—	—	—	—	—	—	—	—	
H'FFFFFF40C to H'FFFFFF41F	—	—	—	—	—	—	—	—	—	—
H'FFFFFF420	ICR0DH									ATU-II
H'FFFFFF421										(Channel 0)
H'FFFFFF422	ICR0DL									
H'FFFFFF423										
H'FFFFFF424	ITVRR1	ITVA9	ITVA8	ITVA7	ITVA6	ITVE9	ITVE8	ITVE7	ITVE6	
H'FFFFFF425	—	—	—	—	—	—	—	—	—	
H'FFFFFF426	ITVRR2A	ITVA13A	ITVA12A	ITVA11A	ITVA10A	ITVE13A	ITVE12A	ITVE11A	ITVE10A	
H'FFFFFF427	—	—	—	—	—	—	—	—	—	
H'FFFFFF428	ITVRR2B	ITVA13B	ITVA12B	ITVA11B	ITVA10B	ITVE13B	ITVE12B	ITVE11B	ITVE10B	
H'FFFFFF429	—	—	—	—	—	—	—	—	—	



Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF42A	TIOR0	IO0D1	IO0D0	IO0C1	IO0C0	IO0B1	IO0B0	IO0A1	IO0A0	ATU-II (Channel 0)
H'FFFFFF42B	—	—	—	—	—	—	—	—	—	
H'FFFFFF42C	TSR0	—	—	—	—	—	—	—	—	
H'FFFFFF42D		IIF2B	IIF2A	IIF1	OVF0	ICF0D	ICF0C	ICF0B	ICF0A	
H'FFFFFF42E	TIER0	—	—	—	—	—	—	—	—	
H'FFFFFF42F		—	—	—	OVE0	ICE0D	ICE0C	ICE0B	ICE0A	
H'FFFFFF430	TCNT0H									
H'FFFFFF431										
H'FFFFFF432	TCNT0L									
H'FFFFFF433										
H'FFFFFF434	ICR0AH									
H'FFFFFF435										
H'FFFFFF436	ICR0AL									
H'FFFFFF437										
H'FFFFFF438	ICR0BH									
H'FFFFFF439										
H'FFFFFF43A	ICR0BL									
H'FFFFFF43B										
H'FFFFFF43C	ICR0CH									
H'FFFFFF43D										
H'FFFFFF43E	ICR0CL									
H'FFFFFF43F										
H'FFFFFF440	TCNT1A									ATU-II (Channel 1)
H'FFFFFF441										
H'FFFFFF442	TCNT1B									
H'FFFFFF443										
H'FFFFFF444	GR1A									
H'FFFFFF445										
H'FFFFFF446	GR1B									
H'FFFFFF447										
H'FFFFFF448	GR1C									
H'FFFFFF449										
H'FFFFFF44A	GR1D									
H'FFFFFF44B										
H'FFFFFF44C	GR1E									
H'FFFFFF44D										
H'FFFFFF44E	GR1F									
H'FFFFFF44F										
H'FFFFFF450	GR1G									
H'FFFFFF451										
H'FFFFFF452	GR1H									
H'FFFFFF453										

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFF454	OCR1									ATU-II (Channel 1)
H'FFFFF455										
H'FFFFF456	OSBR1									
H'FFFFF457										
H'FFFFF458	TIOR1B	—	IO1D2	IO1D1	IO1D0	—	IO1C2	IO1C1	IO1C0	
H'FFFFF459	TIOR1A	—	IO1B2	IO1B1	IO1B0	—	IO1A2	IO1A1	IO1A0	
H'FFFFF45A	TIOR1D	—	IO1H2	IO1H1	IO1H0	—	IO1G2	IO1G1	IO1G0	
H'FFFFF45B	TIOR1C	—	IO1F2	IO1F1	IO1F0	—	IO1E2	IO1E1	IO1E0	
H'FFFFF45C	TCR1B	—	—	CKEGB1	CKEGB0	CKSELB3	CKSELB2	CKSELB1	CKSELB0	
H'FFFFF45D	TCR1A	—	—	CKEGA1	CKEGA0	CKSELA3	CKSELA2	CKSELA1	CKSELA0	
H'FFFFF45E	TSR1A	—	—	—	—	—	—	—	OVF1A	
H'FFFFF45F		IMF1H	IMF1G	IMF1F	IMF1E	IMF1D	IMF1C	IMF1B	IMF1A	
H'FFFFF460	TSR1B	—	—	—	—	—	—	—	OVF1B	
H'FFFFF461		—	—	—	—	—	—	—	CMF1	
H'FFFFF462	TIER1A	—	—	—	—	—	—	—	OVE1A	
H'FFFFF463		IME1H	IME1G	IME1F	IME1E	IME1D	IME1C	IME1B	IME1A	
H'FFFFF464	TIER1B	—	—	—	—	—	—	—	OVE1B	
H'FFFFF465		—	—	—	—	—	—	—	CME1	
H'FFFFF466	TRGMDR	TRGMD	—	—	—	—	—	—	—	
H'FFFFF467 to H'FFFFF47F	—	—	—	—	—	—	—	—	—	—
H'FFFFF480	TSR3	—	OVF5	IMF5D	IMF5C	IMF5B	IMF5A	OVF4	IMF4D	ATU-II (Channel 3 to 5)
H'FFFFF481		IMF4C	IMF4B	IMF4A	OVF3	IMF3D	IMF3C	IMF3B	IMF3A	
H'FFFFF482	TIER3	—	OVE5	IME5D	IME5C	IME5B	IME5A	OVE4	IME4D	
H'FFFFF483		IME4C	IME4B	IME4A	OVE3	IME3D	IME3C	IME3B	IME3A	
H'FFFFF484	TMDR	—	—	—	—	—	T5PWM	T4PWM	T3PWM	
H'FFFFF485 to H'FFFFF49F	—	—	—	—	—	—	—	—	—	—
H'FFFFF4A0	TCNT3									ATU-II (Channel 3)
H'FFFFF4A1										
H'FFFFF4A2	GR3A									
H'FFFFF4A3										
H'FFFFF4A4	GR3B									
H'FFFFF4A5										
H'FFFFF4A6	GR3C									
H'FFFFF4A7										
H'FFFFF4A8	GR3D									
H'FFFFF4A9										
H'FFFFF4AA	TIOR3B	CCI3D	IO3D2	IO3D1	IO3D0	CCI3C	IO3C2	IO3C1	IO3C0	
H'FFFFF4AB	TIOR3A	CCI3B	IO3B2	IO3B1	IO3B0	CCI3A	IO3A2	IO3A1	IO3A0	
H'FFFFF4AC	TCR3	—	—	CKEG1	CKEG0	CKSEL3	CKSEL2	CKSEL1	CKSEL0	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFF4AD to H'FFFFF4BF	—	—	—	—	—	—	—	—	—	—
H'FFFFF4C0	TCNT4									ATU-II
H'FFFFF4C1										(Channel 4)
H'FFFFF4C2	GR4A									
H'FFFFF4C3										
H'FFFFF4C4	GR4B									
H'FFFFF4C5										
H'FFFFF4C6	GR4C									
H'FFFFF4C7										
H'FFFFF4C8	GR4D									
H'FFFFF4C9										
H'FFFFF4CA	TIOR4B	CCI4D	IO4D2	IO4D1	IO4D0	CCI4C	IO4C2	IO4C1	IO4C0	
H'FFFFF4CB	TIOR4A	CCI4B	IO4B2	IO4B1	IO4B0	CCI4A	IO4A2	IO4A1	IO4A0	
H'FFFFF4CC	TCR4	—	—	CKEG1	CKEG0	CKSEL3	CKSEL2	CKSEL1	CKSEL0	
H'FFFFF4CD to H'FFFFF4DF	—	—	—	—	—	—	—	—	—	—
H'FFFFF4E0	TCNT5									ATU-II
H'FFFFF4E1										(Channel 5)
H'FFFFF4E2	GR5A									
H'FFFFF4E3										
H'FFFFF4E4	GR5B									
H'FFFFF4E5										
H'FFFFF4E6	GR5C									
H'FFFFF4E7										
H'FFFFF4E8	GR5D									
H'FFFFF4E9										
H'FFFFF4EA	TIOR5B	CCI5D	IO5D2	IO5D1	IO5D0	CCI5C	IO5C2	IO5C1	IO5C0	
H'FFFFF4EB	TIOR5A	CCI5B	IO5B2	IO5B1	IO5B0	CCI5A	IO5A2	IO5A1	IO5A0	
H'FFFFF4EC	TCR5	—	—	CKEG1	CKEG0	CKSEL3	CKSEL2	CKSEL1	CKSEL0	
H'FFFFF4ED to H'FFFFF4EF	—	—	—	—	—	—	—	—	—	—
H'FFFFF500	TCNT6A									ATU-II
H'FFFFF501										(Channel 6)
H'FFFFF502	TCNT6B									
H'FFFFF503										
H'FFFFF504	TCNT6C									
H'FFFFF505										
H'FFFFF506	TCNT6D									
H'FFFFF507										

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF508	CYLR6A									ATU-II (Channel 6)
H'FFFFFF509										
H'FFFFFF50A	CYLR6B									
H'FFFFFF50B										
H'FFFFFF50C	CYLR6C									
H'FFFFFF50D										
H'FFFFFF50E	CYLR6D									
H'FFFFFF50F										
H'FFFFFF510	BFR6A									
H'FFFFFF511										
H'FFFFFF512	BFR6B									
H'FFFFFF513										
H'FFFFFF514	BFR6C									
H'FFFFFF515										
H'FFFFFF516	BFR6D									
H'FFFFFF517										
H'FFFFFF518	DTR6A									
H'FFFFFF519										
H'FFFFFF51A	DTR6B									
H'FFFFFF51B										
H'FFFFFF51C	DTR6C									
H'FFFFFF51D										
H'FFFFFF51E	DTR6D									
H'FFFFFF51F										
H'FFFFFF520	TCR6B	—	CKSEL2	CKSEL1	CKSEL0	—	CKSEL2	CKSEL1	CKSEL0	
H'FFFFFF521	TCR6A	—	CKSEL2	CKSEL1	CKSEL0	—	CKSEL2	CKSEL1	CKSEL0	
H'FFFFFF522	TSR6	—	—	—	—	—	—	—	—	
H'FFFFFF523		UD6D	UD6C	UD6B	UD6A	CMF6D	CMF6C	CMF6B	CMF6A	
H'FFFFFF524	TIER6	—	—	—	—	—	—	—	—	
H'FFFFFF525		—	—	—	—	CME6D	CME6C	CME6B	CME6A	
H'FFFFFF526	PMDR	DTSELD	DTSELC	DTSELB	DTSELA	CNTSELD	CNTSELC	CNTSELB	CNTSELA	
H'FFFFFF527 to H'FFFFFF57F	—	—	—	—	—	—	—	—	—	
H'FFFFFF580	TCNT7A									ATU-II (Channel 7)
H'FFFFFF581										
H'FFFFFF582	TCNT7B									
H'FFFFFF583										
H'FFFFFF584	TCNT7C									
H'FFFFFF585										
H'FFFFFF586	TCNT7D									
H'FFFFFF587										

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFF588	CYLR7A									ATU-II (Channel 7)
H'FFFFF589										
H'FFFFF58A	CYLR7B									
H'FFFFF58B										
H'FFFFF58C	CYLR7C									
H'FFFFF58D										
H'FFFFF58E	CYLR7D									
H'FFFFF58F										
H'FFFFF590	BFR7A									
H'FFFFF591										
H'FFFFF592	BFR7B									
H'FFFFF593										
H'FFFFF594	BFR7C									
H'FFFFF595										
H'FFFFF596	BFR7D									
H'FFFFF597										
H'FFFFF598	DTR7A									
H'FFFFF599										
H'FFFFF59A	DTR7B									
H'FFFFF59B										
H'FFFFF59C	DTR7C									
H'FFFFF59D										
H'FFFFF59E	DTR7D									
H'FFFFF59F										
H'FFFFF5A0	TCR7B	—	CKSELD2	CKSELD1	CKSELD0	—	CKSELC2	CKSELC1	CKSELC0	
H'FFFFF5A1	TCR7A	—	CKSELB2	CKSELB1	CKSELB0	—	CKSELA2	CKSELA1	CKSELA0	
H'FFFFF5A2	TSR7	—	—	—	—	—	—	—	—	
H'FFFFF5A3		UD7D	UD7C	UD7B	UD7A	CMF7D	CMF7C	CMF7B	CMF7A	
H'FFFFF5A4	TIER7	—	—	—	—	—	—	—	—	
H'FFFFF5A5		—	—	—	—	CME7D	CME7C	CME7B	CME7A	
H'FFFFF5A6 to H'FFFFF5BF	—	—	—	—	—	—	—	—	—	—
H'FFFFF5C0	TCNT11									ATU-II (Channel 11)
H'FFFFF5C1										
H'FFFFF5C2	GR11A									
H'FFFFF5C3										
H'FFFFF5C4	GR11B									
H'FFFFF5C5										
H'FFFFF5C6	TIOR11	—	IO11B2	IO11B1	IO11B0	—	IO11A2	IO11A1	IO11A0	
H'FFFFF5C7	—	—	—	—	—	—	—	—	—	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF5C8	TCR11	—	—	CKEG1	CKEG0	—	CKSELA2	CKSELA1	CKSELA0	ATU-II (Channel 11)
H'FFFFFF5C9	—	—	—	—	—	—	—	—	—	
H'FFFFFF5CA	TSR11	—	—	—	—	—	—	—	OVF11	
H'FFFFFF5CB	—	—	—	—	—	—	—	IMF11B	IMF11A	
H'FFFFFF5CC	TIER11	—	—	—	—	—	—	—	OVE11	
H'FFFFFF5CD	—	—	—	—	—	—	—	IME11B	IME11A	
H'FFFFFF5CE to H'FFFFFF5FF	—	—	—	—	—	—	—	—	—	—
H'FFFFFF600	TCNT2A									ATU-II (Channel 2)
H'FFFFFF601	—									
H'FFFFFF602	TCNT2B									
H'FFFFFF603	—									
H'FFFFFF604	GR2A									
H'FFFFFF605	—									
H'FFFFFF606	GR2B									
H'FFFFFF607	—									
H'FFFFFF608	GR2C									
H'FFFFFF609	—									
H'FFFFFF60A	GR2D									
H'FFFFFF60B	—									
H'FFFFFF60C	GR2E									
H'FFFFFF60D	—									
H'FFFFFF60E	GR2F									
H'FFFFFF60F	—									
H'FFFFFF610	GR2G									
H'FFFFFF611	—									
H'FFFFFF612	GR2H									
H'FFFFFF613	—									
H'FFFFFF614	OCR2A									
H'FFFFFF615	—									
H'FFFFFF616	OCR2B									
H'FFFFFF617	—									
H'FFFFFF618	OCR2C									
H'FFFFFF619	—									
H'FFFFFF61A	OCR2D									
H'FFFFFF61B	—									
H'FFFFFF61C	OCR2E									
H'FFFFFF61D	—									
H'FFFFFF61E	OCR2F									
H'FFFFFF61F	—									

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFF620	OCR2G									ATU-II (Channel 2)
H'FFFFF621										
H'FFFFF622	OCR2H									
H'FFFFF623										
H'FFFFF624	OSBR2									
H'FFFFF625										
H'FFFFF626	TIOR2B	—	IO2D2	IO2D1	IO2D0	—	IO2C2	IO2C1	IO2C0	
H'FFFFF627	TIOR2A	—	IO2B2	IO2B1	IO2B0	—	IO2A2	IO2A1	IO2A0	
H'FFFFF628	TIOR2D	—	IO2H2	IO2H1	IO2H0	—	IO2G2	IO2G1	IO2G0	
H'FFFFF629	TIOR2C	—	IO2F2	IO2F1	IO2F0	—	IO2E2	IO2E1	IO2E0	
H'FFFFF62A	TCR2B	—	—	CKEGB1	CKEGB0	CKSELB3	CKSELB2	CKSELB1	CKSELB0	
H'FFFFF62B	TCR2A	—	—	CKEGA1	CKEGA0	CKSELA3	CKSELA2	CKSELA1	CKSELA0	
H'FFFFF62C	TSR2A	—	—	—	—	—	—	—	OVF2A	
H'FFFFF62D		IMF2H	IMF2G	IMF2F	IMF2E	IMF2D	IMF2C	IMF2B	IMF2A	
H'FFFFF62E	TSR2B	—	—	—	—	—	—	—	OVF2B	
H'FFFFF62F		CMF2H	CMF2G	CMF2F	CMF2E	CMF2D	CMF2C	CMF2B	CMF2A	
H'FFFFF630	TIER2A	—	—	—	—	—	—	—	OVE2A	
H'FFFFF631		IME2H	IME2G	IME2F	IME2E	IME2D	IME2C	IME2B	IME2A	
H'FFFFF632	TIER2B	—	—	—	—	—	—	—	OVE2B	
H'FFFFF633		CME2H	CME2G	CME2F	CME2E	CME2D	CME2C	CME2B	CME2A	
H'FFFFF634 to H'FFFFF63F	—	—	—	—	—	—	—	—	—	—
H'FFFFF640	DCNT8A									ATU-II (Channel 8)
H'FFFFF641										
H'FFFFF642	DNCT8B									
H'FFFFF643										
H'FFFFF644	DNCT8C									
H'FFFFF645										
H'FFFFF646	DCNT8D									
H'FFFFF647										
H'FFFFF648	DCNT8E									
H'FFFFF649										
H'FFFFF64A	DCNT8F									
H'FFFFF64B										
H'FFFFF64C	DCNT8G									
H'FFFFF64D										
H'FFFFF64E	DCNT8H									
H'FFFFF64F										
H'FFFFF650	DCNT8I									
H'FFFFF651										

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF652	DCNT8J									ATU-II (Channel 8)
H'FFFFFF653										
H'FFFFFF654	DCNT8K									
H'FFFFFF655										
H'FFFFFF656	DCNT8L									
H'FFFFFF657										
H'FFFFFF658	DCNT8M									
H'FFFFFF659										
H'FFFFFF65A	DCNT8N									
H'FFFFFF65B										
H'FFFFFF65C	DCNT8O									
H'FFFFFF65D										
H'FFFFFF65E	DCNT8P									
H'FFFFFF65F										
H'FFFFFF660	RLDR8									
H'FFFFFF661										
H'FFFFFF662	TCNR	CN8P	CN8O	CN8N	CN8M	CN8L	CN8K	CN8J	CN8I	
H'FFFFFF663		CN8H	CN8G	CN8F	CN8E	CN8D	CN8C	CN8B	CN8A	
H'FFFFFF664	OTR	OTEP	OTEO	OTEN	OTEM	OTEL	OTEK	OTEJ	OTEI	
H'FFFFFF665		OTEH	OTEG	OTEF	OTEE	OTED	OTEC	OTEB	OTEA	
H'FFFFFF666	DSTR	DST8P	DST8O	DST8N	DST8M	DST8L	DST8K	DST8J	DST8I	
H'FFFFFF667		DST8H	DST8G	DST8F	DST8E	DST8D	DST8C	DST8B	DST8A	
H'FFFFFF668	TCR8	—	CKSELB2	CKSELB1	CKSELB0	—	CKSELA2	CKSELA1	CKSELA0	
H'FFFFFF669	—	—	—	—	—	—	—	—	—	
H'FFFFFF66A	TSR8	OSF8P	OSF8O	OSF8N	OSF8M	OSF8L	OSF8K	OSF8J	OSF8I	
H'FFFFFF66B		OSF8H	OSF8G	OSF8F	OSF8E	OSF8D	OSF8C	OSF8B	OSF8A	
H'FFFFFF66C	TIER8	OSE8P	OSE8O	OSE8N	OSE8M	OSE8L	OSE8K	OSE8J	OSE8I	
H'FFFFFF66D		OSE8H	OSE8G	OSE8F	OSE8E	OSE8D	OSE8C	OSE8B	OSE8A	
H'FFFFFF66E	RLDENR	RLDEN	—	—	—	—	—	—	—	
H'FFFFFF66F to H'FFFFFF67F	—	—	—	—	—	—	—	—	—	—
H'FFFFFF680	ECNT9A									ATU-II (Channel 9)
H'FFFFFF681	—	—	—	—	—	—	—	—	—	
H'FFFFFF682	ECNT9B									
H'FFFFFF683	—	—	—	—	—	—	—	—	—	
H'FFFFFF684	ECNT9C									
H'FFFFFF685	—	—	—	—	—	—	—	—	—	
H'FFFFFF686	ECNT9D									
H'FFFFFF687	—	—	—	—	—	—	—	—	—	
H'FFFFFF688	ECNT9E									
H'FFFFFF689	—	—	—	—	—	—	—	—	—	



Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFF68A	ECNT9F									ATU-II
H'FFFFF68B	—	—	—	—	—	—	—	—	—	(Channel 9)
H'FFFFF68C	GR9A									
H'FFFFF68D	—	—	—	—	—	—	—	—	—	
H'FFFFF68E	GR9B									
H'FFFFF68F	—	—	—	—	—	—	—	—	—	
H'FFFFF690	GR9C									
H'FFFFF691	—	—	—	—	—	—	—	—	—	
H'FFFFF692	GR9D									
H'FFFFF693	—	—	—	—	—	—	—	—	—	
H'FFFFF694	GR9E									
H'FFFFF695	—	—	—	—	—	—	—	—	—	
H'FFFFF696	GR9F									
H'FFFFF697	—	—	—	—	—	—	—	—	—	
H'FFFFF698	TCR9A	—	TRG3BEN	EGSELB1	EGSELB0	—	TRG3AEN	EGSELA1	EGSELA0	
H'FFFFF699	—	—	—	—	—	—	—	—	—	
H'FFFFF69A	TCR9B	—	TRG3DEN	EGSELD1	EGSELD0	—	TRG3CEN	EGSELC1	EGSELC0	
H'FFFFF69B	—	—	—	—	—	—	—	—	—	
H'FFFFF69C	TCR9C	—	—	EGSELF1	EGSELF0	—	—	EGSELE1	EGSELE0	
H'FFFFF69D	—	—	—	—	—	—	—	—	—	
H'FFFFF69E	TSR9	—	—	—	—	—	—	—	—	
H'FFFFF69F	—	—	—	CMF9F	CMF9E	CMF9D	CMF9C	CMF9B	CMF9A	
H'FFFFF6A0	TIER9	—	—	—	—	—	—	—	—	
H'FFFFF6A1	—	—	—	CME9F	CME9E	CME9D	CME9C	CME9B	CME9A	
H'FFFFF6A2 to H'FFFFF6BF	—	—	—	—	—	—	—	—	—	
H'FFFFF6C0	TCNT10A									ATU-II
H'FFFFF6C1	H									(Channel 10)
H'FFFFF6C2	TCNT10A									
H'FFFFF6C3	L									
H'FFFFF6C4	TCNT10B									
H'FFFFF6C5	—	—	—	—	—	—	—	—	—	
H'FFFFF6C6	TCNT10C									
H'FFFFF6C7	—									
H'FFFFF6C8	TCNT10D									
H'FFFFF6C9	—	—	—	—	—	—	—	—	—	
H'FFFFF6CA	TCNT10E									
H'FFFFF6CB	—									
H'FFFFF6CC	TCNT10F									
H'FFFFF6CD	—									

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF6CE	TCNT10G									ATU-II (Channel 10)
H'FFFFFF6CF										
H'FFFFFF6D0	ICR10AH									
H'FFFFFF6D1										
H'FFFFFF6D2	ICR10AL									
H'FFFFFF6D3										
H'FFFFFF6D4	OCR10AH									
H'FFFFFF6D5										
H'FFFFFF6D6	OCR10AL									
H'FFFFFF6D7										
H'FFFFFF6D8	OCR10B									
H'FFFFFF6D9	—	—	—	—	—	—	—	—	—	
H'FFFFFF6DA	RLD10C									
H'FFFFFF6DB										
H'FFFFFF6DC	GR10G									
H'FFFFFF6DD										
H'FFFFFF6DE	TCNT10H									
H'FFFFFF6DF	—	—	—	—	—	—	—	—	—	
H'FFFFFF6E0	NCR10									ATU-II (Channel 10)
H'FFFFFF6E1	—	—	—	—	—	—	—	—	—	
H'FFFFFF6E2	TIOR10	RLDEN	CCS	PIM1	PIM0	—	IO10G2	IO10G1	IO10G0	
H'FFFFFF6E3	—	—	—	—	—	—	—	—	—	
H'FFFFFF6E4	TCR10	TRG2BEN	TRG1BEN	TRG2AEN	TRG1AEN	TRG0DEN	NCE	CKEG1	CKEG0	
H'FFFFFF6E5	—	—	—	—	—	—	—	—	—	
H'FFFFFF6E6	TCCLR10									
H'FFFFFF6E7										
H'FFFFFF6E8	TSR10	—	—	—	—	—	—	—	—	
H'FFFFFF6E9		—	—	—	—	CMF10G	CMF10B	ICF10A	CMF10A	
H'FFFFFF6EA	TIER10	—	—	—	—	—	—	—	—	
H'FFFFFF6EB		—	—	—	IREG	CME10G	CME10B	ICE10A	CME10A	
H'FFFFFF6EC to H'FFFFFF6FF	—	—	—	—	—	—	—	—	—	—
H'FFFFFF700	POPCR	PULS7RO E	PULS6RO E	PULS5RO E	PULS4RO E	PULS3RO E	PULS2RO E	PULS1RO E	PULS0RO E	APC
H'FFFFFF701		PULS7SO E	PULS6SO E	PULS5SO E	PULS4SO E	PULS3SO E	PULS2SO E	PULS1SO E	PULS0SO E	
H'FFFFFF702 to H'FFFFFF707	—	—	—	—	—	—	—	—	—	—
H'FFFFFF708	SYSCR1	OSCSTOP	INOSCE	—	—	—	—	AUDSRST	RAME	Power-Down State
H'FFFFFF709	—	—	—	—	—	—	—	—	—	
H'FFFFFF70A	SYCSR2* <sup>1</sup>	—	—	—	—	—	—	—	—	
H'FFFFFF70B	SYCSR2* <sup>2</sup>	CKSEL	—	—	—	MSTOP3	MSTOP2	MSTOP1	MSTOP0	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF70C to H'FFFFFF70F	—	—	—	—	—	—	—	—	—	—
H'FFFFFF710	CMSTR	—	—	—	—	—	—	—	—	CMT
H'FFFFFF711		—	—	—	—	—	—	STR1	STR0	
H'FFFFFF712	CMCSR0	—	—	—	—	—	—	—	—	
H'FFFFFF713		CMF	CMIE	—	—	—	—	CKS1	CKS0	
H'FFFFFF714	CMCNT0									

Notes:\* This is the read address. The write address is H'FFFFFF70A. For details, see section 25.2.4, Notes on Register Access.

1. Program in the word unit. Programming in the byte or longword unit is not enabled.
2. Read in the byte unit. Correct values cannot be read in the word or longword unit.

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF715	CMCNT0									CMT
H'FFFFFF716	CMCOR0									
H'FFFFFF717										
H'FFFFFF718	CMCSR1	—	—	—	—	—	—	—	—	
H'FFFFFF719		CMF	CMIE	—	—	—	—	CKS1	CKS0	
H'FFFFFF71A	CMCNT1									
H'FFFFFF71B										
H'FFFFFF71C	CMCOR1									
H'FFFFFF71D										
H'FFFFFF71E	—	—	—	—	—	—	—	—	—	
H'FFFFFF71F	—	—	—	—	—	—	—	—	—	
H'FFFFFF720	PAIOR	PA15IOR	PA14IOR	PA13IOR	PA12IOR	PA11IOR	PA10IOR	PA9IOR	PA8IOR	Port A
H'FFFFFF721		PA7IOR	PA6IOR	PA5IOR	PA4IOR	PA3IOR	PA2IOR	PA1IOR	PA0IOR	
H'FFFFFF722	PACRH	—	PA15MD	—	PA14MD	—	PA13MD	—	PA12MD	
H'FFFFFF723		PA11MD1	PA11MD0	PA10MD1	PA10MD0	PA10MD1	PA9MD0	PA8MD1	PA8MD0	
H'FFFFFF724	PACRL	—	PA7MD	—	PA6MD	—	PA5MD	—	PA4MD	
H'FFFFFF725		—	PA3MD	—	PA2MD	—	PA1MD	—	PA0MD	
H'FFFFFF726	PADR	PA15DR	PA14DR	PA13DR	PA12DR	PA11DR	PA10DR	PA9DR	PA8DR	
H'FFFFFF727		PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
H'FFFFFF728	PHIOR	PH15IOR	PH14IOR	PH13IOR	PH12IOR	PH11IOR	PH10IOR	PH9IOR	PH8IOR	Port H
H'FFFFFF729		PH7IOR	PH6IOR	PH5IOR	PH4IOR	PH3IOR	PH2IOR	PH1IOR	PH0IOR	
H'FFFFFF72A	PHCR	PH15MD	PH14MD	PH13MD	PH12MD	PH11MD	PH10MD	PH9MD	PH8MD	
H'FFFFFF72B		PH7MD	PH6MD	PH5MD	PH4MD	PH3MD	PH2MD	PH1MD	PH0MD	
H'FFFFFF72C	PHDR	PH15DR	PH14DR	PH13DR	PH12DR	PH11DR	PH10DR	PH9DR	PH8DR	
H'FFFFFF72D		PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR	
H'FFFFFF72E	ADTRGR1	EXTRG	—	—	—	—	—	—	—	A/D
H'FFFFFF72F	ADTRGR2	EXTRG	—	—	—	—	—	—	—	
H'FFFFFF730	PBIOR	PB15IOR	PB14IOR	PB13IOR	PB12IOR	PB11IOR	PB10IOR	PB9IOR	PB8IOR	Port B
H'FFFFFF731		PB7IOR	PB6IOR	PB5IOR	PB4IOR	PB3IOR	PB2IOR	PB1IOR	PB0IOR	
H'FFFFFF732	PBCRH	PB15MD1	PB15MD0	PB14MD1	PB14MD0	—	PB13MD	PB12MD1	PB12MD0	
H'FFFFFF733		PB11MD1	PB11MD0	PB10MD1	PB10MD0	PB9MD1	PB9MD0	PB8MD1	PB8MD0	
H'FFFFFF734	PBCRL	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0	
H'FFFFFF735		—	PB3MD	—	PB2MD	—	PB1MD	—	PB0MD	
H'FFFFFF736	PBIR	PB15IR	PB14IR	PB13IR	—	PB11IR	PB10IR	PB9IR	PB8IR	
H'FFFFFF737		PB7IR	PB6IR	PB5IR	PB4IR	PB3IR	PB2IR	PB1IR	PB0IR	
H'FFFFFF738	PBDR	PB15DR	PB14DR	PB13DR	PB12DR	PB11DR	PB10DR	PB9DR	PB8DR	
H'FFFFFF739		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
H'FFFFFF73A	PCIOR	—	—	—	—	—	—	—	—	Port C
H'FFFFFF73B		—	—	—	PC4IOR	PC3IOR	PC2IOR	PC1IOR	PC0IOR	
H'FFFFFF73C	PCCR	—	—	—	—	—	—	—	PC4MD	
H'FFFFFF73D		—	PC3MD	—	PC2MD	—	PC1MD	—	PC0MD	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF73E	PCDR	—	—	—	—	—	—	—	—	Port C
H'FFFFFF73F		—	—	—	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
H'FFFFFF740	PDIOR	—	—	PD13IOR	PD12IOR	PD11IOR	PD10IOR	PD9IOR	PD8IOR	Port D
H'FFFFFF741		PD7IOR	PD6IOR	PD5IOR	PD4IOR	PD3IOR	PD2IOR	PD1IOR	PD0IOR	
H'FFFFFF742	PDCRH	—	—	—	—	PD13MD1	PD13MD0	—	PD12MD	
H'FFFFFF743		—	PD11MD	—	PD10MD	—	PD9MD	—	PD8MD	
H'FFFFFF744	PDCRL	—	PD7MD	—	PD6MD	—	PD5MD	—	PD4MD	
H'FFFFFF745		—	PD3MD	—	PD2MD	—	PD1MD	—	PD0MD	
H'FFFFFF746	PDDR	—	—	PD13DR	PD12DR	PD11DR	PD10DR	PD9DR	PD8DR	
H'FFFFFF747		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
H'FFFFFF748	PFIOR	PF15IOR	PF14IOR	PF13IOR	PF12IOR	PF11IOR	PF10IOR	PF9IOR	PF8IOR	Port F
H'FFFFFF749		PF7IOR	PF6IOR	PF5IOR	PF4IOR	PF3IOR	PF2IOR	PF1IOR	PF0IOR	
H'FFFFFF74A	PFCRH	CKHIZ	PF15MD	—	PF14MD	—	PF13MD	—	PF12MD	
H'FFFFFF74B		—	PF11MD	—	PF10MD	—	PF9MD	—	PF8MD	
H'FFFFFF74C	PFCRL	—	PF7MD	—	PF6MD	PF5MD1	PF5MD0	—	PF4MD	
H'FFFFFF74D		—	PF3MD	—	PF2MD	—	PF1MD	—	PF0MD	
H'FFFFFF74E	PFDR	PF15DR	PF14DR	PF13DR	PF12DR	PF11DR	PF10DR	PF9DR	PF8DR	
H'FFFFFF74F		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
H'FFFFFF750	PEIOR	PE15IOR	PE14IOR	PE13IOR	PE12IOR	PE11IOR	PE10IOR	PE9IOR	PE8IOR	Port E
H'FFFFFF751		PE7IOR	PE6IOR	PE5IOR	PE4IOR	PE3IOR	PE2IOR	PE1IOR	PE0IOR	
H'FFFFFF752	PECR	PE15MD	PE14MD	PE13MD	PE12MD	PE11MD	PE10MD	PE9MD	PE8MD	
H'FFFFFF753		PE7MD	PE6MD	PE5MD	PE4MD	PE3MD	PE2MD	PE1MD	PE0MD	
H'FFFFFF754	PEDR	PE15DR	PE14DR	PE13DR	PE12DR	PE11DR	PE10DR	PE9DR	PE8DR	
H'FFFFFF755		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
H'FFFFFF756	PLIOR	—	—	PL13IOR	PL12IOR	PL11IOR	PL10IOR	PL9IOR	PL8IOR	Port L
H'FFFFFF757		PL7IOR	PL6IOR	PL5IOR	PL4IOR	PL3IOR	PL2IOR	PL1IOR	PL0IOR	
H'FFFFFF758	PLCRH	—	—	—	—	PL13MD1	PL13MD0	—	PL12MD	
H'FFFFFF759		PL11MD1	PL11MD0	PL10MD1	PL10MD0	PL9MD1	PL9MD0	—	PL8MD	
H'FFFFFF75A	PLCRL	—	PL7MD	—	PL6MD	—	PL5MD	—	PL4MD	
H'FFFFFF75B		—	PL3MD	PL2MD1	PL2MD0	PL1MD1	PL1MD0	—	PL0MD0	
H'FFFFFF75C	PLIR	—	—	—	—	—	—	PL9IR	PL8IR	
H'FFFFFF75D		PL7IR	—	—	—	—	—	—	—	
H'FFFFFF75E	PLDR	—	—	PL13DR	PL12DR	PL11DR	PL10DR	PL9DR	PL8DR	
H'FFFFFF75F		PL7DR	PL6DR	PL5DR	PL4DR	PL3DR	PL2DR	PL1DR	PL0DR	
H'FFFFFF760	PGIOR	—	—	—	—	—	—	—	—	Port G
H'FFFFFF761		—	—	—	—	PG3IOR	PG2IOR	PG1IOR	PG0IOR	
H'FFFFFF762	PGCR	—	—	—	—	—	—	—	—	
H'FFFFFF763		PG3MD1	PG3MD0	PG2MD1	PG2MD0	—	PG1MD	PG0MD1	PG0MD0	
H'FFFFFF764	PGDR	—	—	—	—	—	—	—	—	
H'FFFFFF765		—	—	—	—	PG3DR	PG2DR	PG1DR	PG0DR	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF766	PJIOR	PJ15IOR	PJ14IOR	PJ13IOR	PJ12IOR	PJ11IOR	PJ10IOR	PJ9IOR	PJ8IOR	Port J
H'FFFFFF767		PJ7IOR	PJ6IOR	PJ5IOR	PJ4IOR	PJ3IOR	PJ2IOR	PJ1IOR	PJ0IOR	
H'FFFFFF768	PJCRH	—	PJ15MD	—	PJ14MD	—	PJ13MD	—	PJ12MD	
H'FFFFFF769		—	PJ11MD	—	PJ10MD	—	PJ9MD	—	PJ8MD	
H'FFFFFF76A	PJCRL	—	PJ7MD	—	PJ6MD	—	PJ5MD	—	PJ4MD	
H'FFFFFF76B		—	PJ3MD	—	PJ2MD	—	PJ1MD	—	PJ0MD	
H'FFFFFF76C	PJDR	PJ15DR	PJ14DR	PJ13DR	PJ12DR	PJ11DR	PJ10DR	PJ9DR	PJ8DR	
H'FFFFFF76D		PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR	
H'FFFFFF76E	ADTRG0	EXTRG	—	—	—	—	—	—	—	A/D
H'FFFFFF76F	—	—	—	—	—	—	—	—	—	
H'FFFFFF770	PKIOR	PK15IOR	PK14IOR	PK13IOR	PK12IOR	PK11IOR	PK10IOR	PK9IOR	PK8IOR	Port K
H'FFFFFF771		PK7IOR	PK6IOR	PK5IOR	PK4IOR	PK3IOR	PK2IOR	PK1IOR	PK0IOR	
H'FFFFFF772	PKCRH	—	PK15MD	—	PK14MD	—	PK13MD	—	PK12MD	
H'FFFFFF773		—	PK11MD	—	PK10MD	—	PK9MD	—	PK8MD	
H'FFFFFF774	PKCRL	—	PK7MD	—	PK6MD	—	PK5MD	—	PK4MD	
H'FFFFFF775		—	PK3MD	—	PK2MD	—	PK1MD	—	PK0MD	
H'FFFFFF776	PKIR	PK15IR	PK14IR	PK13IR	PK12IR	PK11IR	PK10IR	PK9IR	PK8IR	
H'FFFFFF777		PK7IR	PK6IR	PK5IR	PK4IR	PK3IR	PK2IR	PK1IR	PK0IR	
H'FFFFFF778	PKDR	PK15DR	PK14DR	PK13DR	PK12DR	PK11DR	PK10DR	PK9DR	PK8DR	
H'FFFFFF779		PK7DR	PK6DR	PK5DR	PK4DR	PK3DR	PK2DR	PK1DR	PK0DR	
H'FFFFFF77A to H'FFFFFF77F	—	—	—	—	—	—	—	—	—	—
H'FFFFFF780	PAPR	PA15PR	PA14PR	PA13PR	PA12PR	PA11PR	PA10PR	PA9PR	PA8PR	Port A
H'FFFFFF781		PA7PR	PA6PR	PA5PR	PA4PR	PA3PR	PA2PR	PA1PR	PA0PR	
H'FFFFFF782	PBPR	PB15PR	PB14PR	PB13PR	PB12PR	PB11PR	PB10PR	PB9PR	PB8PR	Port B
H'FFFFFF783		PB7PR	PB6PR	PB5PR	PB4PR	PB3PR	PB2PR	PB1PR	PB0PR	
H'FFFFFF784	PDPR	PD15PR	PD14PR	PD13PR	PD12PR	PD11PR	PD10PR	PD9PR	PD8PR	Port D
H'FFFFFF785		PD7PR	PD6PR	PD5PR	PD4PR	PD3PR	PD2PR	PD1PR	PD0PR	
H'FFFFFF786	PJPR	PJ15PR	PJ14PR	PJ13PR	PJ12PR	PJ11PR	PJ10PR	PJ9PR	PJ8PR	Port J
H'FFFFFF787		PJ7PR	PJ6PR	PJ5PR	PJ4PR	PJ3PR	PJ2PR	PJ1PR	PJ0PR	
H'FFFFFF788	PLPR	PL15PR	PL14PR	PL13PR	PL12PR	PL11PR	PL10PR	PL9PR	PL8PR	Port L
H'FFFFFF789		PL7PR	PL6PR	PL5PR	PL4PR	PL3PR	PL2PR	PL1PR	PL0PR	
H'FFFFFF78A to H'FFFFFF7BF	—	—	—	—	—	—	—	—	—	—

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFFF7C0	SDIR	TS3	TS2	TS1	TS0	—	—	—	—	H-UDI
H'FFFFFF7C1		—	—	—	—	—	—	—	—	
H'FFFFFF7C2	SDSR	—	—	—	—	—	—	—	—	
H'FFFFFF7C3		—	—	—	—	—	—	—	SDTRF	
H'FFFFFF7C4	SDDRH									
H'FFFFFF7C5										
H'FFFFFF7C6	SDDRL									
H'FFFFFF7C7										
H'FFFFFF7C8 to H'FFFFFF7FF	—	—	—	—	—	—	—	—	—	—
H'FFFFFF800	ADDR0H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	A/D
H'FFFFFF801	ADDR0L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF802	ADDR1H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF803	ADDR1L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF804	ADDR2H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF805	ADDR2L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF806	ADDR3H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF807	ADDR3L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF808	ADDR4H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF809	ADDR4L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF80A	ADDR5H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF80B	ADDR5L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF80C	ADDR6H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF80D	ADDR6L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF80E	ADDR7H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF80F	ADDR7L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF810	ADDR8H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF811	ADDR8L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF812	ADDR9H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF813	ADDR9L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF814	ADDR10H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF815	ADDR10L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF816	ADDR11H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF817	ADDR11L	AD1	AD0	—	—	—	—	—	—	
H'FFFFFF818	ADCSR0	ADF	ADIE	ADM1	ADM0	CH3	CH2	CH1	CH0	
H'FFFFFF819	ADCR0	TRGE	CKS	ADST	ADCS	—	—	—	—	
H'FFFFFF81A to H'FFFFFF81F	—	—	—	—	—	—	—	—	—	
H'FFFFFF820	ADDR12H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFFF821	ADDR12L	AD1	AD0	—	—	—	—	—	—	

Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFF822	ADDR13H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	A/D
H'FFFFF823	ADDR13L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF824	ADDR14H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF825	ADDR14L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF826	ADDR15H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF827	ADDR15L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF828	ADDR16H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF829	ADDR16L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF82A	ADDR17H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF82B	ADDR17L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF82C	ADDR18H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF82D	ADDR18L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF82E	ADDR19H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF82F	ADDR19L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF830	ADDR20H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF831	ADDR20L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF832	ADDR21H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF833	ADDR21L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF834	ADDR22H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF835	ADDR22L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF836	ADDR23H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF837	ADDR23L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF838	ADCSR1	ADF	ADIE	ADM1	ADM0	CH3	CH2	CH1	CH0	
H'FFFFF839	ADCR1	TRGE	CKS	ADST	ADCS	—	—	—	—	
H'FFFFF83A to H'FFFFF83F	—	—	—	—	—	—	—	—	—	
H'FFFFF840	ADDR24H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF841	ADDR24L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF842	ADDR25H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF843	ADDR25L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF844	ADDR26H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF845	ADDR26L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF846	ADDR27H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF847	ADDR27L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF848	ADDR28H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF849	ADDR28L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF84A	ADDR29H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF84B	ADDR29L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF84C	ADDR30H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	
H'FFFFF84D	ADDR30L	AD1	AD0	—	—	—	—	—	—	
H'FFFFF84E	ADDR31H	AD9	AD8	AD7	AD6	AD5	ADR	AD3	AD2	



Register Name	Abbreviation	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFFFF84F	ADDR31L	AD1	AD0	—	—	—	—	—	—	A/D
H'FFFFF850	—	—	—	—	—	—	—	—	—	
to H'FFFFF857										
H'FFFFF858	ADCSR2	ADF	ADIE	ADM1	ADM0	—	CH2	CH1	CH0	
H'FFFFF859	ADCR2	TRGE	CKS	ADST	ADCS	—	—	—	—	
H'FFFFF85A	—	—	—	—	—	—	—	—	—	
to H'FFFFF85F										
H'FFFFF860	ADCNT0									MTAD
H'FFFFF861										
H'FFFFF862	ADCYLR0									
H'FFFFF863										
H'FFFFF864	ADDR0A									
H'FFFFF865										
H'FFFFF866	ADDR0B									
H'FFFFF867										
H'FFFFF868	ADGR0A									
H'FFFFF869										
H'FFFFF86A	ADGR0B									
H'FFFFF86B										
H'FFFFF86C	ADTCR0	CKSEL1x	CKSEL0x	—	—	DTSELxB	DTSELxA	ADSELxB	ADSELxA	
H'FFFFF86D	ADTSR0	—	TADFxB	TADFxA	ADDFxB	ADDFxA	ADCYLFx	ADCMFxB	ADCMFxA	
H'FFFFF86E	ADTIER0	ADTRGx	TADExB	TADExA	ADDExB	ADDExA	ADCYLEx	ADCMExB	ADCNExA	
H'FFFFF86F										
H'FFFFF870	ADCNT1									
H'FFFFF871										
H'FFFFF872	ADCYLR1									
H'FFFFF873										
H'FFFFF874	ADDR1A									
H'FFFFF875										
H'FFFFF876	ADDR1B									
H'FFFFF877										
H'FFFFF878	ADGR1A									
H'FFFFF879										
H'FFFFF87A	ADGR1B									
H'FFFFF87B										
H'FFFFF87C	ADTCR1	CKSEL1x	CKSEL0x	—	—	DTSELxB	DTSELxA	ADSELxB	ADSELxA	
H'FFFFF87D	ADTSR1	—	TADFxB	TADFxA	ADDFxB	ADDFxA	ADCYLFx	ADCMFxB	ADCMFxA	
H'FFFFF87E	ADTIER1	ADTRGx	TADExB	TADExA	ADDExB	ADDExA	ADCYLEx	ADCMExB	ADCNExA	
H'FFFFF87F										

## A.2 Register States in Reset and Power-Down States

**Table A.2 Register States in Reset and Power-Down States**

Type	Name	Reset State	Power-Down State					
		Power-On	Hardware Standby	Software Standby	Sleep			
CPU	R0 to R15	Initialized	Initialized	Held	Held			
	SR							
	GBR							
	VBR							
	MACH, MACL							
	PR							
	PC							
FPU	FR0 to FR15	Initialized	Initialized	Held	Held			
	FPUL							
	FPSCR							
Interrupt controller (INTC)	IPRA to IPRL	Initialized	Initialized	Held	Held			
	ICR							
	ISR							
User break controller (UBC)	UBARH, UBARL	Initialized	Initialized	Held	Held			
	UBAMRH, UBAMRL							
	UBBR							
	UBCR							
Bus state controller (BSC)	BCR1, BCR2	Initialized	Initialized	Held	Held			
	WCR							
Direct memory access controller (DMAC)	SAR0 to SAR3	Undefined	Undefined	Undefined	Held			
	DAR0 to DAR3							
	DMATCR0 to DMATCR3							
	CHCR0 to CHCR3					Initialized	Initialized	Initialized
	DMAOR							
Advanced timer unit-II (ATU-II)	BFR6A-D, BFR7A-D	Initialized	Initialized	Initialized	Held			
	CYLR6A-D, CYLR7A-D							
	DCNT8A-P							
	DSTR							

**Table A.2 Register States in Reset and Power-Down States (cont)**

Type	Name	Reset State	Power-Down State		
		Power-On	Hardware Standby	Software Standby	Sleep
Advanced timer unit-II (ATU-II)	DTR6A-D, DTR7A-D	Initialized	Initialized	Initialized	Held
	ECNT9A-F				
	GR1A-H, GR2A-H GR3A-D, GR4A-D GR5A-D, GR9A-F GR10G, GR11A, 11B				
	ICR0A-D, ICR10A				
	ITVRR1, ITVRR2A, 2B				
	NCR10				
	OCR1, OCR2A-H OCR10AH, 10AL OCR10B				
	OSBR1, OSBR2				
	OTR				
	PMDR				
	PSCR1-4				
	PSTR				
	RLD10C				
	RLDENR				
	RLDR8				
	TCCLR10				
	TCNR				
	TCNT0H, L, TCNT1A, 1B, TCNT2A, 2B TCNT3-5, TCNT6A-D TCNT7A-D TCNT10AH, 10AL TCNT10B-H, TCNT11				
	TCR1A, 1B TCR2A, 2B, TCR3-5 TCR6A, 6B, TCR7A, 7B, TCR8, TCR9A-C TCR10, TCR11				

**Table A.2 Register States in Reset and Power-Down States (cont)**

Type	Name	Reset State	Power-Down State		
		Power-On	Hardware Standby	Software Standby	Sleep
Advanced timer unit-II (ATU-II)	TIER0, TIER1A, 1B TIER2A, 2B, TIER3 TIER6-11	Initialized	Initialized	Initialized	Held
	TIOR0, TIOR1A-D TIOR2A-D, TIOR3A, 3B, TIOR4A, 4B TIOR5A, 5B TIOR10,11				
	TMDR				
	TNCT10E				
	TRGMDR				
	TSR0, TSR1A, 1B TSR2A, 2B, TSR3 TSR6-11				
	TSTR1-3				
Advanced pulse controller (APC)	POPCR	Initialized	Initialized	Held	Held
Watchdog timer (WDT)	TCNT	Initialized	Initialized	Initialized	Held
	TCSR				
	RSTCSR				
Serial communication interface (SCI)	SMR0 to SMR4	Initialized	Initialized	Held	Held
	BRR0 to BRR4				
	SCR0 to SCR4				
	TDR0 to TDR4			Initialized	
	SSR0 to SSR4				
	RDR0 to RDR4				
	SDCR0 to SDCR4			Held	
A/D converter	ADDR0 (H/L) to ADDR31 (H/L)	Initialized	Initialized	Initialized	Held
	ADSCR0, ADCSR1 ADCSR2				
	ADCR0, ADCR1 ADCR2				

**Table A.2 Register States in Reset and Power-Down States (cont)**

Type	Name	Reset State	Power-Down State		
		Power-On	Hardware Standby	Software Standby	Sleep
A/D converter	ADTRGR0, ADTRGR1 ADTRGR2	Initialized	Initialized	Held	Held
Compare match timer (CMT)	CMSTR	Initialized	Initialized	Initialized	Held
	CMCSR0, CMCSR1				
	CMCNT0, CMCNT1	Initialized	Initialized	Initialized	Held
	CMCOR0, CMCOR1				
Pin function controller (PFC)	PAIOR, PBIOR	Initialized	Initialized	Held	Held
	PCIOR, PDIOR				
	PEIOR, PFIOR				
	PGIOR, PHIOR				
	PJIOR, PKIOR, PLIOR				
	PACRH, PACRL				
	PBCRH, PBCRL				
	PBIR, PCCR, PDCRH				
	PDCRL, PECR				
	PFCRH, PFCRL				
	PGCR, PHCR, PJCRH				
	PJCRL, PKCRH				
	PKCRL, PKIR, PLCRH				
PLCRL, PLIR					
I/O ports	PADR, PBDR, PCDR	Initialized	Initialized	Held	Held
	PDDR, PEDR, PFDR				
	PGDE, PHDR, PJDR				
	PKDR, PLDR				
Flash ROM	RAMER	Initialized	Initialized	Held	Held
	FCCS			Initialized/ Held*	
	FPCS			Initialized	
	FECS				
	FKEY				
	FMATS			Held	
	FTDAR			Initialized	

**Table A.2 Register States in Reset and Power-Down States (cont)**

Type	Name	Reset State	Power-Down State		
		Power-On	Hardware Standby	Software Standby	Sleep
Power-down state related	SBYCR	Initialized	Initialized	Held	Held
	SYSR1, SYSR2				
	MSTCR				
Controller area network (HCAN)	MCR	Initialized	Initialized	Initialized	Held
	GSR				
	HCAN_BCR 0/1				
	IRP				
	IMR				
	TXPR 0/1				
	TXCR 0/1				
	TXACK 0/1				
	ABACK 0/1				
	RXPR 0/1				
	RFPR 0/1				
	MBIMR 0/1				
	UMSR 0/1				
	TCNTR				
	TCR				
	TSR				
	TMR				
	TDCR				
	LOSR				
	CCR				
	CMAX				
	ICR 0/1				
	TCMR 0-2				
MB	Undefined	Held	Held	Held	
High-performance user debug interface (H-UDI)	SDIR	Held	Held	Held	Held
	SDSR				
	SDDRH, SDDRL				

Note: \* Bit 7 (FLER) is held, and bit 0 (SCO) is initialized.



# Appendix B Pin States

Tables B.1, B.2, and B.3 show the SH7058 pin states.

**Table B.1 Pin States**

Type	Pin Name	Pin State								
		Reset State				Power-Down State				
		Power-On				Hardware Standby	Software Standby	H-JDI Module Standby	AUD Module Standby	Bus-Released State
		ROMless Expanded Mode		Expanded Mode with ROM	Single-Chip Mode					
8 Bits	16 Bits									
Clock	CK* <sup>2</sup>	O				Z	H* <sup>1</sup>	O	O	O
	XTAL	O				L	L	O	O	O
	EXTAL	I				Z	I	I	I	I
	PLLCAP	I				I	I	I	I	I
System control	RES	I				Z	I	I	I	I
	FWE	I				I	I	I	I	I
	HSTBY	I				I	I	I	I	I
	MD0	I				I	I	I	I	I
	MD1	I				I	I	I	I	I
	MD2	I				I	I	I	I	I
	WDTOVF	O				Z	O* <sup>1</sup>	O	O	O
	BREQ	—				Z	Z	I	I	I
	BACK	—				Z	Z	O	O	L
Interrupt	NMI	I				Z	I	I	I	I
	IRQ0 to IRQ7	—				Z	Z	I	I	I
	IRQOUT	—				Z	O* <sup>1</sup>	O	O	O
Address bus	A0 to A21	O		—		Z	Z	O	O	Z
Data bus	D0 to D7	Z		—		Z	Z	I/O	I/O	Z
	D8 to D15	—	Z	—		Z	Z	I/O	I/O	Z
Bus control	WAIT	I			—	Z	Z	I	I	I
	WRH, WRL	H			—	Z	Z	O	O	Z
	RD	H			—	Z	Z	O	O	Z
	CS0	H			—	Z	Z	O	O	Z
	CS1 to CS3	—				Z	Z	O	O	Z
Port	POD	—				Z	Z	I	I	I
ATU-II	TIOA to TIOD	—				Z	Z	I	I	I
	TIO1A to TIO1H	—				Z	K* <sup>1</sup>	I/O	I/O	I/O
	TIO2A to TIO2H	—				Z	K* <sup>1</sup>	I/O	I/O	I/O
	TIO3A to TIO3D	—				Z	K* <sup>1</sup>	I/O	I/O	I/O



**Table B.1 Pin States (cont)**

		Pin State								
		Reset State			Power-Down State					
		Power-On								
Type	Pin Name	ROMless Expanded Mode		Expanded Mode with ROM	Single-Chip Mode	Hardware Standby	Software Standby	H-UDI Module Standby	AUD Module Standby	Bus-Released State
		8 Bits	16 Bits							
ATU-II	TIO4A to TIO4D	—				Z	K* <sup>1</sup>	I/O	I/O	I/O
	TIO5A to TIO5D	—				Z	K* <sup>1</sup>	I/O	I/O	I/O
	TO6A to TO6D	—				Z	O* <sup>1</sup>	O	O	O
	TO7A to TO7D	—				Z	O* <sup>1</sup>	O	O	O
	TO8A to TO8P	—				Z	O* <sup>1</sup>	O	O	O
	TI9A to TI9F	—				Z	Z	I	I	I
	TI10	—				Z	Z	I	I	I
	TIO11A, TIO11B	—				Z	K* <sup>1</sup>	I/O	I/O	I/O
	TCLKA, TCLKB	—				Z	Z	I	I	I
SCI	SCK0 to SCK4	—				Z	K* <sup>1</sup>	I/O	I/O	I/O
	TxD0 to TxD4	—				Z	O* <sup>1</sup>	O	O	O
	RxD0 to RxD4	—				Z	Z	I	I	I
A/D converter	AN0 to AN31	Z				Z	Z	I	I	I
	ADTRG0, ADTRG1	—				Z	Z	I	I	I
	ADEND	—				Z	O* <sup>1</sup>	O	O	O
	AVref	I				I	I	I	I	I
APC	PULS0 to PULS7	—				Z	O* <sup>1</sup>	O	O	O
HCAN	HTxD0, HTxD1	—				Z	O* <sup>1</sup>	O	O	O
	HRxD0, HRxD1	—				Z	Z	I	I	I
UBC	UBCTRG	—				Z	O* <sup>1</sup>	O	O	O
I/O port	PA0 to PA15	Z				Z	K* <sup>1</sup>	I/O	I/O	I/O
	PB0 to PB15	Z				Z	K* <sup>1</sup>	I/O	I/O	I/O
	PC0 to PC4	Z				Z	K* <sup>1</sup>	I/O	I/O	I/O
	PD0 to PD13	Z				Z	K* <sup>1</sup>	I/O	I/O	I/O
	PE0 to PE15	—	Z			Z	K* <sup>1</sup>	I/O	I/O	I/O
	PF0 to PF5	—	Z			Z	K* <sup>1</sup>	I/O	I/O	I/O
	PF6 to PF10	—			Z	Z	K* <sup>1</sup>	I/O	I/O	I/O
	PH11 to PH15	Z				Z	K* <sup>1</sup>	I/O	I/O	I/O
	PG0 to PG3	Z				Z	K* <sup>1</sup>	I/O	I/O	I/O
	PH0 to PH7	—		Z		Z	K* <sup>1</sup>	I/O	I/O	I/O
	PH8 to PH15	Z	—	Z		Z	K* <sup>1</sup>	I/O	I/O	I/O
	PJ0 to PJ15	Z				Z	K* <sup>1</sup>	I/O	I/O	I/O
	PK0 to PK15	Z				Z	K* <sup>1</sup>	I/O	I/O	I/O
PL0 to PL13	Z				Z	K* <sup>1</sup>	I/O	I/O	I/O	

**Table B.2 Pin States**

		Pin State									
		Reset State					Power-Down State				
		Power-On									
Type	Pin Name	ROMless Expanded Mode		Expanded Mode with ROM	Single-Chip Mode	Hardware Standby	Software Standby	H-UDI Module Standby	AUD Module Standby	Bus-Released State	No Connection
		8 Bits	16 Bits								
H-UDI	TMS	I				Z	I	Z	I	I	Pulled up internally
	$\overline{\text{TRST}}$	I				Z	I	Z	I	I	Pulled up internally
	TDI	I				Z	I	Z	I	I	Pulled up internally
	TDO	O/Z				Z	O/Z	Z	O/Z	O/Z	O/Z
	TCK	I				Z	I	Z	I	I	Pulled up internally

**Table B.3 Pin States**

		Pin State			
Type	Pin Name	Hardware Standby	AUD Reset	Software Standby	No Connection
		$\overline{\text{AUD}}$ Module Standby	( $\overline{\text{AUDRST}} = \text{L}$ )	$\overline{\text{AUDSRST}} = 1/$ Normal Operation	
AUD	$\overline{\text{AUDRST}}$	Z	L input	H input	Pulled down internally
	AUDMD	Z	I	I	Pulled up internally
	AUDATA0 to AUDATA3	Z	When AUDMD = H: I When AUDMD = L: H (pulled up internally)	When AUDMD = H: I/O When AUDMD = L: O	Pulled up internally
	AUDCK	Z	When AUDMD = H: I When AUDMD = L: H (pulled up internally)	When AUDMD = H: I When AUDMD = L: O	Pulled up internally
	$\overline{\text{AUDSYNC}}$	Z	When AUDMD = H: I When AUDMD = L: H (pulled up internally)	When AUDMD = H: I When AUDMD = L: O	Pulled up internally

— : Not initial value

I : Input

O : Output

H : High-level output

L : Low-level output

Z : High impedance

K : Input pins become high-impedance, output pins retain their state.

Notes: 1. When the port impedance bit (HIZ) in the standby control register (SBYCR) is set to 1, output pins become high-impedance.

2. When the CKHIZ bit in PFCRH is set to 1, becomes high-impedance unconditionally.



# Appendix C Product Lineup

**Table C.1 SH7058 F-ZTAT Product Lineup**

<b>Product Type</b>	<b>Model Name</b>	<b>Mark Model Name</b>	<b>Package</b>	<b>Operating Temperature (Except for W/E of Flash Memory)</b>	
SH7058	F-ZTAT	HD64F7058BF80L	64F7058F80	256-pin (FP-256H)	-40°C to 105°C
		HD64F7058BF80K	64F7058F80	256-pin (FP-256H)	-40°C to 125°C
		HD64F7058BP80L	64F7058BP80	272-pin (BP-272)	-40°C to 105°C
		HD64F7058BP80K	64F7058BP80	272-pin (BP-272)	-40°C to 125°C





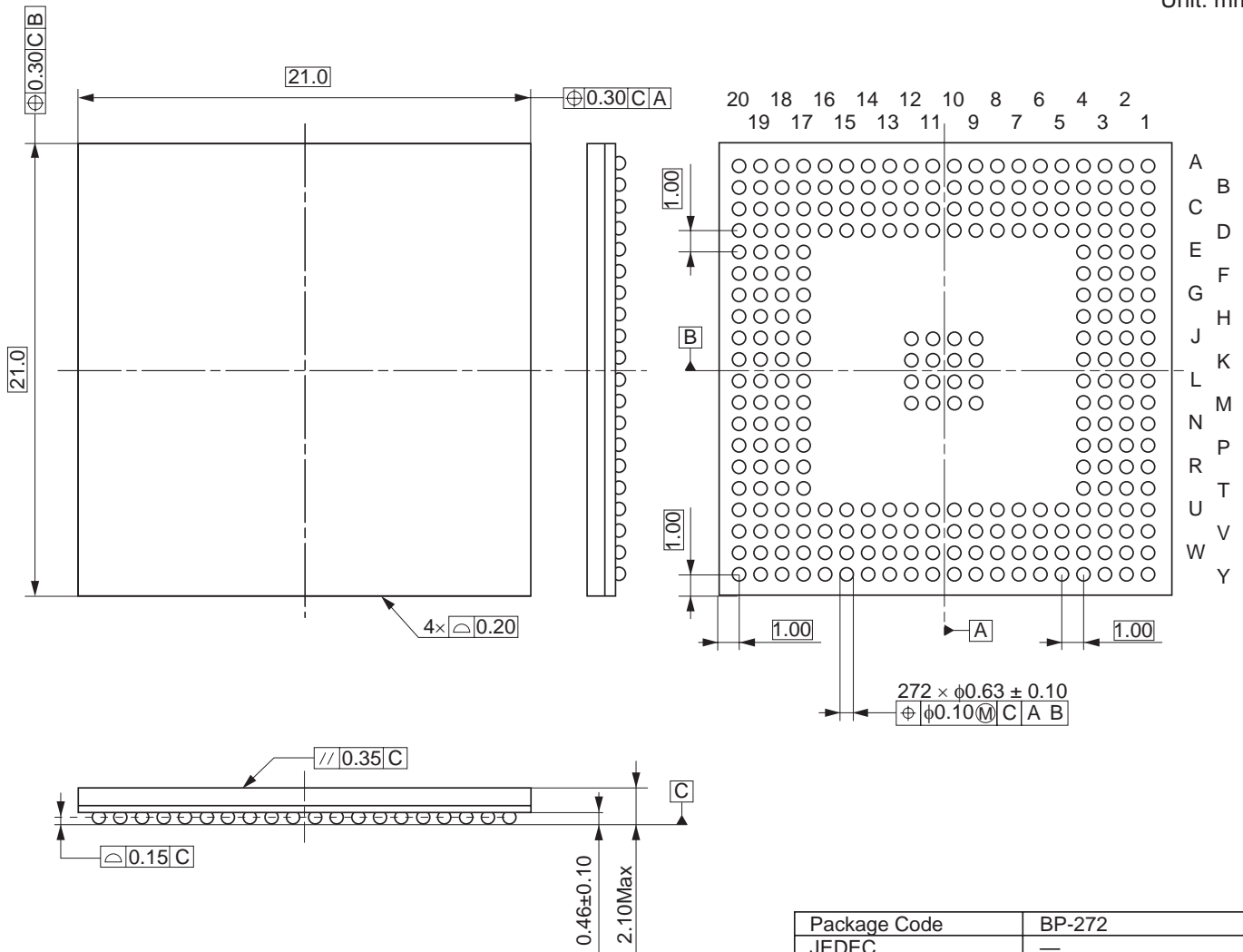


Figure D.2 SH7058 Package Dimensions (BP-272)

---

**Renesas SuperH™ RISC engine  
Hardware Manual  
SH-2E SH7058 F-ZTAT™**

Publication Date: 1st Edition, July, 2002  
Rev.3.00, September 17, 2004  
Published by: Sales Strategic Planning Div.  
Renesas Technology Corp.  
Edited by: Technical Documentation & Information Department  
Renesas Kodaïra Semiconductor Co., Ltd.

---

© 2004. Renesas Technology Corp., All rights reserved. Printed in Japan.



**Renesas Technology Corp.** Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

---



**RENESAS SALES OFFICES**

<http://www.renesas.com>

---

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

**Renesas Technology America, Inc.**

450 Holger Way, San Jose, CA 95134-1368, U.S.A  
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

**Renesas Technology Europe Limited**

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

**Renesas Technology Hong Kong Ltd.**

7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong  
Tel: <852> 2265-6688, Fax: <852> 2730-6071

**Renesas Technology Taiwan Co., Ltd.**

10th Floor, No.99, Fushing North Road, Taipei, Taiwan  
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

**Renesas Technology (Shanghai) Co., Ltd.**

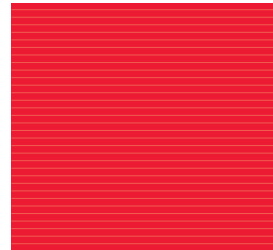
Unit2607 Ruijing Building, No.205 Maoming Road (S), Shanghai 200020, China  
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

**Renesas Technology Singapore Pte. Ltd.**

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: <65> 6213-0200, Fax: <65> 6278-8001



SH-2E SH7058 F-ZTAT™  
Hardware Manual



Renesas Technology Corp.

2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan