

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

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

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

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

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

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

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

"Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

# H8S/2368 Group

## Hardware Manual

### Renesas 16-Bit Single-Chip Microcomputer

### H8S Family/H8S/2300 Series

H8S/2368F	HD64F2368
H8S/2367F	HD64F2367
H8S/2365	HD6432365
H8S/2364F	HD64F2364
H8S/2363	HD6412363
H8S/2362F	HD64F2362
H8S/2361F	HD64F2361
H8S/2360F	HD64F2360

Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.

2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
  - (1) artificial life support devices or systems
  - (2) surgical implantations
  - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
  - (4) any other purposes that pose a direct threat to human life

Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.

9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

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

#### 1. Handling of Unused Pins

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

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

#### 2. Processing at Power-on

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

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

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

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

#### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

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

#### 4. Clock Signals

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

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

#### 5. Differences between Products

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

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

This manual comprises the following items:

1. General Precautions in the Handling of MPU/MCU Products
2. Configuration of This Manual
3. Preface
4. Main Revisions for This Edition

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

5. Contents
6. Overview
7. Description of Functional Modules
  - CPU and System-Control Modules
  - On-Chip Peripheral Modules

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

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

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

8. List of Registers
9. Electrical Characteristics
10. Appendix
11. Index

The H8S/2368 Group are microcomputers (MCU) made up of the H8S/2600 CPU employing Renesas Technology's original architecture as their cores, and the peripheral functions required to configure a system.

The H8S/2600 CPU has an internal 32-bit configuration, sixteen 16-bit general registers, and a simple and optimized instruction set for high-speed operation. The H8S/2600 CPU can handle a 16-Mbyte linear address space.

This LSI is equipped with direct memory access controller (DMAC) and data transfer controller (DTC) bus masters, ROM and RAM, a 16-bit timer pulse unit (TPU), a programmable pulse generator (PPG), 8-bit timers (TMR), a watchdog timer (WDT), serial communication interfaces (SCI and IrDA), a 10-bit A/D converter, an 8-bit D/A converter, and I/O ports as on-chip peripheral modules required for system configuration. I<sup>2</sup>C bus interface 2 (IIC2) can also be included as an optional interface.

A high-functionality bus controller is also provided, enabling fast and easy connection of DRAM and other kinds of memory.

A single-power flash memory (F-ZTAT™) version is available for this LSI's ROM. This provides flexibility as it can be reprogrammed in no time to cope with all situations from the early stages of mass production to full-scale mass production. This is particularly applicable to application devices with specifications that will most probably change.

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

**Target Users:** This manual was written for users who will be using the H8S/2368 Group in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

**Objective:** This manual was written to explain the hardware functions and electrical characteristics of the H8S/2368 Group to the target users.  
Refer to the H8S/2600 Series, H8S/2000 Series Software Manual for a detailed description of the instruction set.

Notes on reading this manual:

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

For the execution state of each instruction in this LSI, see appendix D, Bus State during Execution of Instructions.

- In order to understand the details of a register when its name is known

Read the index that is the final part of the manual to find the page number of the entry on the register. The addresses, bits, and initial values of the registers are summarized in section 24, List of Registers.

Examples: Register name: The following notation is used for cases when the same or a similar function, e.g. 16-bit timer pulse unit or serial communication, is implemented on more than one channel: XXX\_N (XXX is the register name and N is the channel number)

Bit order: The MSB is on the left and the LSB is on the right.

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

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

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

H8S/2368 Group Manuals:

Document Title	Document No.
H8S/2368 Group Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Software Manual	REJ09B0139

User's Manuals for Development Tools:

Document Title	Document No.
H8S, H8/300 Series C/C++ Compiler, Assembler, Optimizing Linkage Editor Compiler Package Ver.6.01 User's Manual	REJ10B0161
H8S, H8/300 Series Simulator/Debugger User's Manual	REJ10B0211
H8S, H8/300 Series High-performance Embedded Workshop, V.3 Tutorial	REJ10B0024
High-performance Embedded Workshop V.4.04 User's Manual	REJ10J1737



1.3.3 Pin Functions 14 Table amended

Table 1.2 Pin Functions

Type	Symbol	Pin No.		I/O	Function
		TFP-120	QFP-128 <sup>SH</sup>		
Power Supply	V <sub>cc</sub>	2, 33 <sup>SH</sup> , 60, 83, 84	6,39,66, 91,92	Input	Power supply pins. V <sub>cc</sub> pins should be connected to the system power supply.
	V <sub>ss</sub>	8, 17, 22, 58, 80, 87	3,4,12,21, 26,35,36, 64,68,88, 95,99,100	Input	Ground pins. V <sub>ss</sub> pins should be connected to the system power supply (0 V).
	PLL <sub>V<sub>cc</sub></sub>	76	84	Input	Power supply pin for the on-chip PLL oscillator.
	PLL <sub>V<sub>ss</sub></sub>	78	86	Input	Ground pin for the on-chip PLL oscillator.
	VCL	33 <sup>*3</sup>		Input	Connect to V <sub>ss</sub> via a 0.1 μF (recommended value) capacitor (placed close to the pin).

20 Note 1 amended and Notes 2 and 3 added

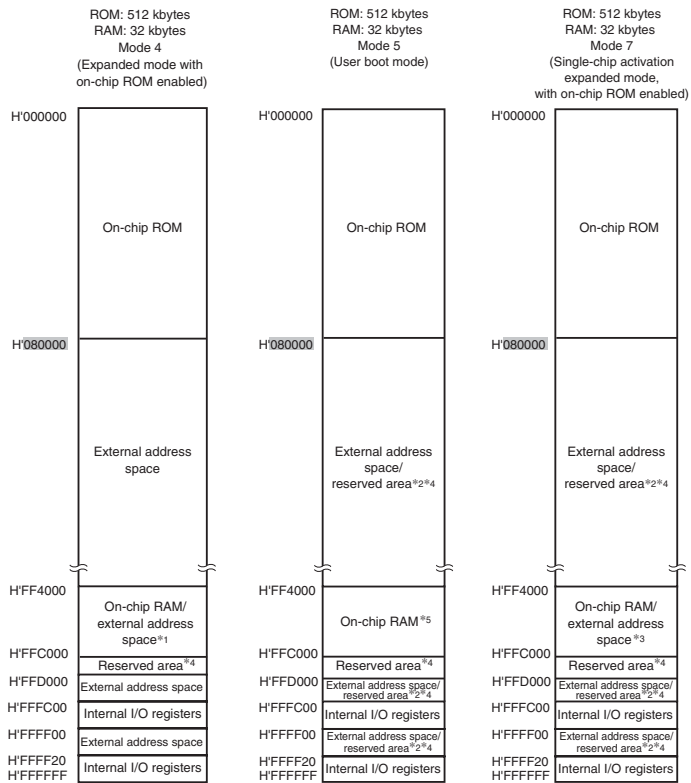
Notes: 1. Not supported by the H8S/2368 0.18 μm F-ZTAT Group.

2. VCL on the H8S/2368 0.18 μm F-ZTAT Group. Do not connect to VCL.

3. H8S/2368 0.18 μm F-ZTAT Group only.

# Operating Mode

## Figure 3.2 H8S/2368F Memory Map (2)

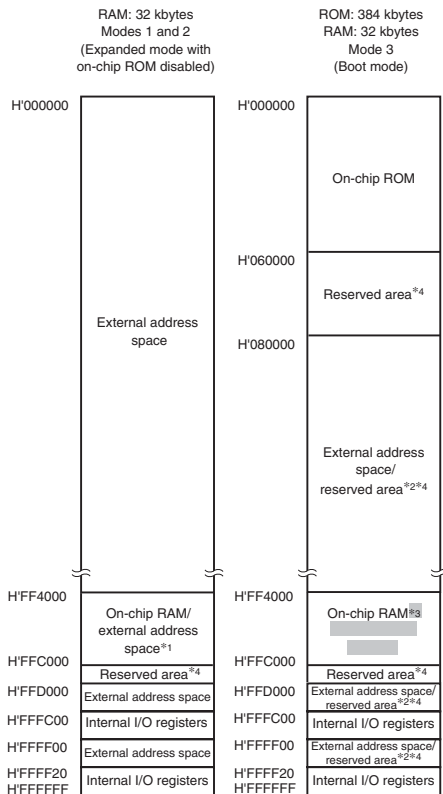


- Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.  
 2. When EXPE = 1, external address space; when EXPE = 0, reserved area.  
 3. When EXPE = 1, external address space with RAME = 0, on-chip RAM with RAME = 1.  
 4. A reserved area should not be accessed.  
 5. The on-chip RAM is used to program the flash memory. The RAME bit in SYSCR should not be cleared to 0.



# Operating Mode

## Figure 3.5 H8S/2364F Memory Map (1)



- Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.  
 2. When EXPE = 1, external address space; when EXPE = 0, reserved area.  
 3. On-chip RAM is used for flash memory programming. The RAME bit in SYSCR should not be cleared to 0.  
 4. A reserved area should not be accessed.

### 7.3.7 DMA Terminal Control Register (DMATCR)

240 Description amended  
 In short address mode, the **TEND** pin is only available for channel B.

### 8.8.5 Chain Transfer

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

• PA7/A23/CS7/IRQ7

PA7/DDR	0	1	0	1	0	1	0	1	0	1	0	1	
Pin function	PA7 input	PA7 output	PA7 input	CS7 output	PA7 output	Address input	PA7 input	PA7 output	PA7 input	PA7 output	CS7 output	PA7 input	Address output
IRQ7 interrupt input*													

371 • PA6/A22/IRQ6, PA5/A21/IRQ5

PA $\bar{n}$ /DDR	0	1	0	1	0	1	0	1	0	1
Pin function	PA $\bar{n}$ input	PA $\bar{n}$ output	PA $\bar{n}$ input	Address output	PA $\bar{n}$ input	PA $\bar{n}$ output	PA $\bar{n}$ input	PA $\bar{n}$ output	PA $\bar{n}$ input	Address output
IRQ $\bar{n}$ interrupt input*										

372 • PA3/A19, PA2/A18, PA1/A17, PA0/A16

Pin function	Address output	PA $\bar{n}$ input	PA $\bar{n}$ output	PA $\bar{n}$ input	Address output	PA $\bar{n}$ input	PA $\bar{n}$ output	PA $\bar{n}$ input	PA $\bar{n}$ output	PA $\bar{n}$ input	Address output

9.9.5 Pin Functions

375 Table amended

- PB7/A15, PB6/A14, PB5/A13, PB4/A12, PB3/A11, PB2/A10, PB1/A9, PB0/A8

PB $\bar{n}$ /DDR	—	0	1	0	1	0	1
Pin function	Address output	PB $\bar{n}$ input	Address output	PB $\bar{n}$ input	PB $\bar{n}$ output	PB $\bar{n}$ input	Address output

9.10.5 Pin Functions

379 Table amended

- PC7/A7, PC6/A6, PC5/A5, PC4/A4, PC3/A3, PC2/A2, PC1/A1, PC0/A0

PC $\bar{n}$ /DDR	—	0	1	0	1	0	1
Pin function	Address output	PC $\bar{n}$ input	Address output	PC $\bar{n}$ input	PC $\bar{n}$ output	PC $\bar{n}$ input	Address output

9.11.5 Pin Functions

383 Table amended

- PD7/D15, PD6/D14, PD5/D13, PD4/D12, PD3/D11, PD2/D10, PD1/D9, PD0/D8

PD $\bar{n}$ /DDR	—	0	1	—
Pin function	Data I/O	PD $\bar{n}$ input	PD $\bar{n}$ output	Data I/O

9.12.5 Pin Functions

387 Table amended

- PE7/D7, PE6/D6, PE5/D5, PE4/D4, PE3/D3, PE2/D2, PE1/D1, PE0/D0

PE $\bar{n}$ /DDR	0	1	—	0	1	0	1	—
Pin function	PE $\bar{n}$ input	PE $\bar{n}$ output	Data I/O	PE $\bar{n}$ input	PE $\bar{n}$ output	PE $\bar{n}$ input	PE $\bar{n}$ output	Data I/O

- PF7/ $\phi$

(Before) PFDDR → (After) PF7DDR

393 Table amended

- PF1/ $\overline{\text{CS5}}$ /UCAS

PF1DDR	—	0	1	0	1	0	1	—	0	1	0	1
Pin function	UCAS output	PF1 input	CS5 output	PF1 input	PF1 output	PF1 input	PF1 output	UCAS output	PF1 input	CS5 output	PF1 input	PF1 output

14.3.7 Serial Status Register (SSR) 558

Note amended

Note: \* Only 0 can be written, to clear the flag. Alternately, use the bit clear instruction to clear the flag.

Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Smart Card Interface Mode (When SMIF in SCMR is 1) 562

Note amended

Notes: 1. Only 0 can be written, to clear the flag. Alternately, use the bit clear instruction to clear the flag.  
2. etu: Elementary Time Unit: (time for transfer of 1 bit)

14.3.9 Bit Rate Register (BRR) 565

Table amended

Table 14.3 BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	8			9.8304			10			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	25	0.16	0	31	0.00	0	32	-1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	-1.73	0	19	-2.34
31250	0	7	0.00	0	9	-1.70	0	9	0.00	0	11	0.00
38400	—	—	—	0	7	0.00	0	7	-1.73	0	9	-2.34

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	12.288			14			14.7456			16		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	39	0.00	0	45	-0.93	0	47	0.00	0	51	0.16
19200	0	19	0.00	0	22	-0.93	0	23	0.00	0	25	0.16
31250	0	11	2.40	0	13	0.00	0	14	-1.70	0	15	0.00
38400	0	9	0.00	—	—	—	0	11	0.00	0	12	0.16

(BRR)

Table 14.3 BRR Settings  
for Various Bit Rates  
(Asynchronous Mode)

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	17.2032			18			19.6608			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32	-1.36
31250	0	16	1.20	0	17	0.00	0	19	-1.70	0	19	0.00
38400	0	13	0.00	0	14	-2.34	0	15	0.00	0	15	1.73

Bit Rate (bit/s)	Operating Frequency $\phi$ (MHz)											
	25		30		33			34*				
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
19200	0	40	-0.76	0	48	-0.35	0	53	-0.54	0	54	0.62
31250	0	24	0.00	0	29	0	0	32	0	0	33	0.00
38400	0	19	1.73	0	23	1.73	0	26	-0.54	0	27	-1.18

14.4.4 SCI Initialization 580  
(Asynchronous Mode)

Description amended

... Do not write to SMR, SCMR, IrCR, or SEMR while the SCI is operating. This also applies to writing the same data as the current register contents. ...

14.6.2 SCI Initialization 596  
(Clocks Synchronous Mode)

Description amended

... Do not write to SMR, SCMR, IrCR, or SEMR while the SCI is operating. This also applies to writing the same data as the current register contents. ...

Section 15 I<sup>2</sup>C Bus 629  
Interface2 (IIC2) (Option)

Description amended

The I<sup>2</sup>C bus interface conforms to and provides a subset of the NXP Semiconductors I<sup>2</sup>C bus (inter-IC bus) interface (Rev.03) standard and fast mode functions. The register configuration that controls the I<sup>2</sup>C bus differs partly from the NXP Semiconductors configuration, however.

15.3.5 I<sup>2</sup>C Bus Status Register (ICSR) 639

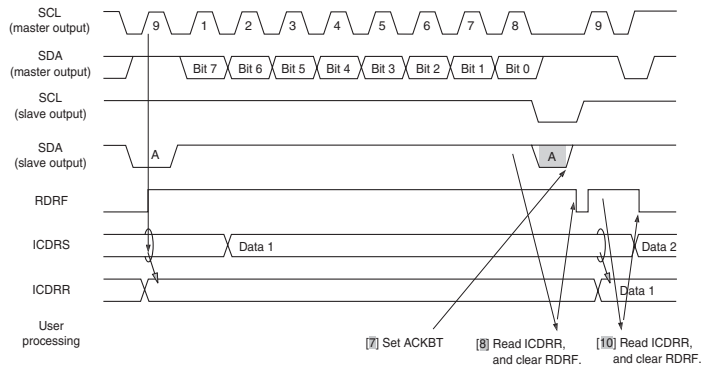
Table amended

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Setting conditions] <ul style="list-style-type: none"> <li>When data is transferred from ICDDRT to ICDDRS and ICDDRT becomes empty</li> <li>When TRS is set</li> </ul>
[Clearing conditions] <ul style="list-style-type: none"> <li>When 0 is written in TDRE after reading TDRE = 1</li> <li>When data is written to ICDDRT</li> </ul>				

15.4.5 Slave Receive Operation 652

Figure amended

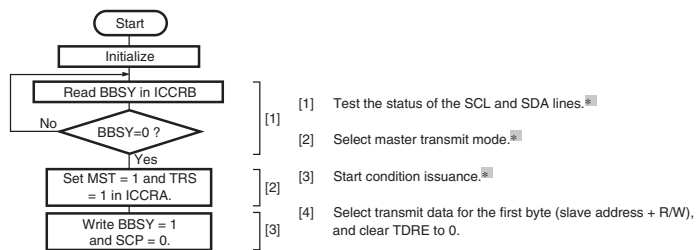
Figure 15.12 Slave Receive Mode Operation Timing 2



15.4.7 Example of Use 654

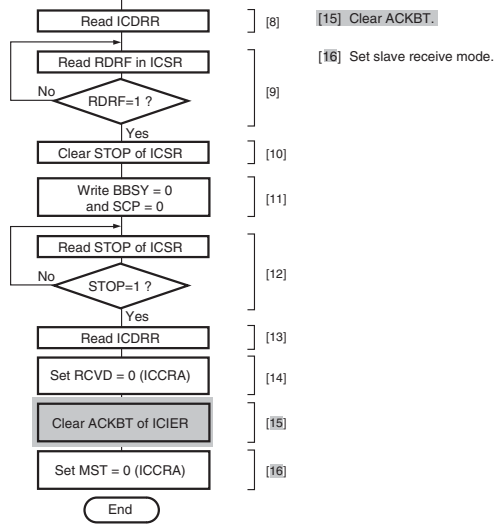
Figure amended

Figure 15.14 Sample Flowchart for Master Transmit Mode



Note: \* Ensure that no interrupts occur between when BBSY is cleared to 0 and start condition [3].

Figure 15.15 Sample Flowchart for Master Receive Mode

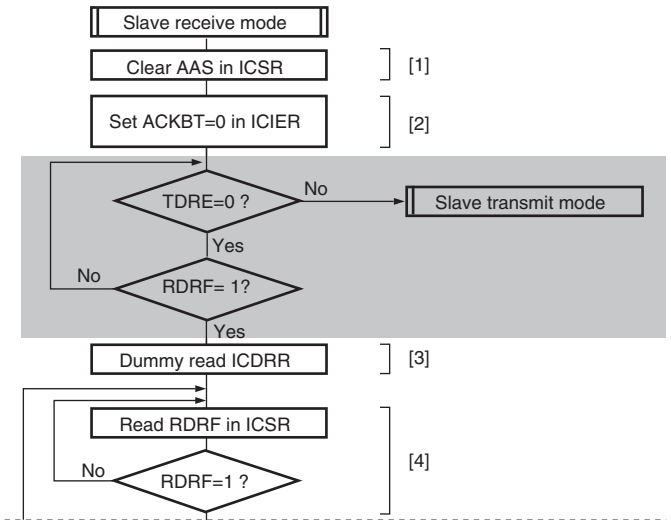


Note: \* Prevent any interrupts while steps [1] to [3] are executed.  
 Additional information: When receiving one-byte data, execute step [1], and then step [7] omitting steps [2] to [6].  
 In step [8], dummy read ICDRR.

Figure 15.17 Sample Flowchart for Slave Receive Mode

657

Figure amended





3. I<sup>2</sup>C bus interface 2 (IIC2) master receive mode

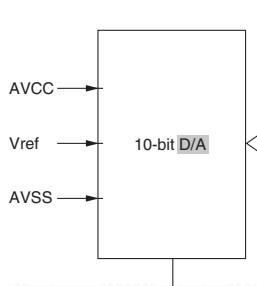
4. Limitations on transfer rate setting values when using I<sup>2</sup>C bus interface 2 (IIC2) in multi-master mode

5. Limitations on use of bit manipulation instructions to set MST and TRS when using I<sup>2</sup>C bus interface 2 (IIC2) in multi-master mode

---

16.1 Features 662 Figure amended

Figure 16.1 Block Diagram of A/D Converter



---

20.3.2 Programming/Erasing Interface Parameter 735

Description amended

... The return value of the processing result is written in ER0, ER1. Since the stack area is used for storing the registers except for ER0, ER1, the stack area must be saved at the processing start.

---

20.3.3 Flash Vector Address Control Register (FVACR) 747

Description amended

... Normally the vector table data is read from the address spaces from H'00001C to H'00001F.

---

20.4.2 User Program Mode 757

Description amended

The notes on download are as follows.

(2) Programming Procedure in User Program Mode

— In the download processing, the values of CPU general registers other than ER0 and ER1 are retained.

(b) Device Selection

...

(4) Inquiry and Selection  
States

- Size (one byte): Amount of device-code data  
This is fixed at 4

Figure 20.21  
Programming Sequence

801 Description deleted  
(Before) Programming selection (H'42, H'43, H'44)  
(After) Programming selection (H'42, H'43)

(9) Programming/Erasing  
State

- Programming
- (b) 128-byte programming

802 Description amended

- Programming Address (four bytes): Start address for programming  
Multiple of the size specified in response to the programming unit inquiry  
(i.e. H'00, H'01, H'00, H'00 : H'00010000)

24.2 Register Bits

855 Table amended

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3
DRACCR	DRMI	—	TPC1	TPC0	—

25.1.2 DC  
Characteristics

874 Table amended

Table 25.2 DC  
Characteristics (1)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions	
Schmitt trigger input voltage	Ports 1, 2, and 4 <sup>82</sup> , P50 to P53 <sup>82</sup> , PA4 to PA7 <sup>82</sup>	$V_T^-$	$V_{CC}$	0.2	—	V	
		$V_T^+$	—	—	$V_{CC}$	0.7 V	
		$V_T^+ - V_T^-$	$V_{CC}$	0.07	—	V	
Input high voltage	STBY, MD2 to MD0	$V_{IH}$	$V_{CC}$	0.9	—	$V_{CC} + 0.3$ V	
	RES, NMI, EMLE		$V_{CC}$	0.9	—	$V_{CC} + 0.3$ V	
	EXTAL		$V_{CC}$	0.7	—	$V_{CC} + 0.3$ V	
	Port 3, P50 to P53 <sup>83</sup> , port 8, ports A to G <sup>83</sup>		2.2V	—	—	$V_{CC} + 0.3$ V	
	Ports 4 and 9		2.2V	—	—	$V_{CC} + 0.3$ V	
Input low voltage	RES, STBY, MD2 to MD0, EMLE	$V_{IL}$	-0.3	—	$V_{CC}$	0.1 V	
	NMI, EXTAL		-0.3	—	$V_{CC}$	0.2 V	
	Ports 3 to 5 <sup>83</sup> , 8, 9, A to G <sup>83</sup>		-0.3	—	$V_{CC}$	0.2 V	
Output high voltage	All output pins	$V_{OH}$	$V_{CC} - 0.5$	—	—	V	$I_{OH} = -200$ A
			$V_{CC} - 1.0$	—	—	V	$I_{OH} = -1$ mA
Output low voltage	All output pins	$V_{OL}$	—	—	0.4	V	$I_{OL} = 1.6$ mA
			—	—	0.5	V	$I_{OL} = 8.0$ mA

Note added

Notes: 4. When used as SCL0 to SCL1, SDA0 to SDA1.

Characteristics

Table 25.4 Permissible Output Currents

Item	Symbol	Min	Typ	Max	Unit	
Permissible output low current (per pin)	SCL0, 1, SDA0, 1	$I_{OL}$	—	—	8.0	mA
	Output pins other than the above	—	—	2.0	—	

25.2.2 DC Characteristics

912 Table amended

Table 25.14 DC Characteristics (1)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions	
Schmitt trigger input voltage	Ports 1, 2, and 4 <sup>32</sup> , P50 to P53 <sup>32</sup> , PA4 to PA7 <sup>32</sup>	$V_T$	$V_{CC} \times 0.2$	—	—	V	
		$V_T^+$	—	—	$V_{CC} \times 0.7$	V	
		$V_T^- - V_T^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	STBY, MD2 to MD0	$V_{IH}$	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	RES, NMI, EMLE		$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Port 3, P50 to P53 <sup>33</sup> , port 8 <sup>33</sup> , ports A to G <sup>33</sup>		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Ports 4 and 9		$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V	
Input low voltage	RES, STBY, MD2 to MD0, EMLE	$V_{IL}$	-0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL		-0.3	—	$V_{CC} \times 0.2$	V	
	Ports 3 to 5 <sup>33</sup> , 8, 9, A to G <sup>33</sup>		-0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	$V_{OH}$	$V_{CC} - 0.5$	—	—	V	$I_{OH} = -200 \mu A$
			$V_{CC} - 1.0$	—	—	V	$I_{OH} = -1 \text{ mA}$
Output low voltage	All output pins	$V_{OL}$	—	—	0.4	V	$I_{OL} = 1.6 \text{ mA}$
			P32 to P35 <sup>34</sup>	—	—	0.5	V

Note added

Notes: 4. When used as SCL0 to SCL1, SDA0 to SDA1.

Table 25.16 Permissible Output Currents

914 Table amended

Item	Symbol	Min	Typ	Max	Unit	
Permissible output low current (per pin)	SCL0 to SCL1, SDA0 to SDA1	$I_{OL}$	—	—	8.0	mA
	Output pins other than the above	—	—	2.0	—	



Characteristics

Table 25.27 DC Characteristics

Item		Symbol
Schmitt trigger input voltage	Port 1, 2, 4 <sup>92</sup> , P50 to P53 <sup>92</sup> , PA4 to PA7 <sup>92</sup>	VT <sup>+</sup>
		VT <sup>+</sup>
		VT <sup>+</sup> - VT
Input high voltage	STBY, MD2 to MD0	V <sub>IH</sub>
	RES, NMI, FWE	
	EXTAL	
	Port 3, P50 to P53 <sup>93</sup> , port 8 <sup>93</sup> , ports A to G <sup>93</sup>	
	Port 4, Port 9	
Input low voltage	RES, STBY, MD2 to MD0, EMLE	V <sub>IL</sub>
	NMI, EXTAL	
	Ports 3 to 5 <sup>94</sup> , port 8, port 9, ports A to G <sup>93</sup>	
	Port 4, Port 9	
Output high voltage	All output pins	V <sub>OH</sub>
Output low voltage	All output pins	V <sub>OL</sub>
	P32 to P35 <sup>94</sup>	

Note amended

- Notes: 2. When used as  $\overline{\text{IRQ0}}$  to  $\overline{\text{IRQ7}}$ .  
 3. When used as other than  $\overline{\text{IRQ0}}$  to  $\overline{\text{IRQ7}}$

Table 25.28 DC Characteristics

928 Table amended

Item	
Input leakage current	RES
	STBY, NMI, MD2 to MD0
	Port 4, Port 9
Three-state leakage current (off state)	Ports 1 to 3, P50 to P53, port 8, ports A to G

Table 25.29 Permissible Output Currents

929 Table amended

Item	
Permissible output low current (per pin)	SCL0 to SCL1, SDA0 to SDA1
	Output pins other than the above

25.3.3 AC Characteristics

935

Table amended

Table 25.33 Bus Timing  
(2)

Item	Symbol	Min.
WAIT hold time	$t_{\text{WITH}}$	5

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

Section 1	Overview	1
1.1	Features	1
1.2	Block Diagram	3
1.3	Pin Description	5
1.3.1	Pin Arrangement	5
1.3.2	Pin Arrangement in Each Operating Mode	8
1.3.3	Pin Functions	14
Section 2	CPU	21
2.1	Features	21
2.1.1	Differences between H8S/2600 CPU and H8S/2000 CPU	22
2.1.2	Differences from H8/300 CPU	23
2.1.3	Differences from H8/300H CPU	23
2.2	CPU Operating Modes	24
2.2.1	Normal Mode	24
2.2.2	Advanced Mode	26
2.3	Address Space	28
2.4	Register Configuration	29
2.4.1	General Registers	30
2.4.2	Program Counter (PC)	31
2.4.3	Extended Control Register (EXR)	31
2.4.4	Condition-Code Register (CCR)	32
2.4.5	Initial Register Values	34
2.5	Data Formats	34
2.5.1	General Register Data Formats	34
2.5.2	Memory Data Formats	36
2.6	Instruction Set	37
2.6.1	Table of Instructions Classified by Function	38
2.6.2	Basic Instruction Formats	47
2.7	Addressing Modes and Effective Address Calculation	49
2.7.1	Register Direct—Rn	49
2.7.2	Register Indirect—@ERn	49
2.7.3	Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn)	50
2.7.4	Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn	50
2.7.5	Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32	50
2.7.6	Immediate—#xx:8, #xx:16, or #xx:32	51
2.7.7	Program-Counter Relative—@(d:8, PC) or @(d:16, PC)	51

2.8	Processing States.....	55
2.9	Usage Note.....	56
2.9.1	Note on Bit Manipulation Instructions.....	56
Section 3 MCU Operating Modes .....		57
3.1	Operating Mode Selection .....	57
3.2	Register Descriptions .....	58
3.2.1	Mode Control Register (MDCR) .....	58
3.2.2	System Control Register (SYSCR).....	58
3.3	Operating Mode Descriptions .....	60
3.3.1	Mode 1 .....	60
3.3.2	Mode 2 .....	60
3.3.3	Mode 3 .....	60
3.3.4	Mode 4 .....	60
3.3.5	Mode 5 .....	61
3.3.6	Mode 7 .....	61
3.3.7	Pin Functions .....	62
3.4	Memory Map in Each Operating Mode .....	63
Section 4 Exception Handling .....		79
4.1	Exception Handling Types and Priority .....	79
4.2	Exception Sources and Exception Vector Table .....	79
4.3	Reset.....	81
4.3.1	Reset Exception Handling.....	81
4.3.2	Interrupts after Reset.....	83
4.3.3	On-Chip Peripheral Functions after Reset Release .....	83
4.4	Traces .....	84
4.5	Interrupts .....	84
4.6	Trap Instruction.....	85
4.7	Stack Status after Exception Handling.....	86
4.8	Usage Notes .....	87
Section 5 Interrupt Controller .....		89
5.1	Features.....	89
5.2	Input/Output Pins .....	91
5.3	Register Descriptions .....	91
5.3.1	Interrupt Control Register (INTCR) .....	92
5.3.2	Interrupt Priority Registers A to K (IPRA to IPRK).....	93



5.3.5	IRQ Status Register (ISR).....	99
5.3.6	IRQ Pin Select Register (ITSR).....	100
5.3.7	Software Standby Release IRQ Enable Register (SSIER).....	101
5.4	Interrupt Sources.....	101
5.4.1	External Interrupts.....	101
5.4.2	Internal Interrupts.....	102
5.5	Interrupt Exception Handling Vector Table.....	103
5.6	Interrupt Control Modes and Interrupt Operation.....	108
5.6.1	Interrupt Control Mode 0.....	108
5.6.2	Interrupt Control Mode 2.....	110
5.6.3	Interrupt Exception Handling Sequence.....	112
5.6.4	Interrupt Response Times.....	114
5.6.5	DTC and DMAC Activation by Interrupt.....	115
5.7	Usage Notes.....	116
5.7.1	Contention between Interrupt Generation and Disabling.....	116
5.7.2	Instructions that Disable Interrupts.....	117
5.7.3	Times when Interrupts Are Disabled.....	117
5.7.4	Interrupts during Execution of EEPMOV Instruction.....	117
5.7.5	Change of IRQ Pin Select Register (ITSR) Setting.....	117
5.7.6	Note on IRQ Status Register (ISR).....	118
Section 6 Bus Controller (BSC).....		119
6.1	Features.....	119
6.2	Input/Output Pins.....	121
6.3	Register Descriptions.....	122
6.3.1	Bus Width Control Register (ABWCR).....	123
6.3.2	Access State Control Register (ASTCR).....	123
6.3.3	Wait Control Registers AH, AL, BH, and BL (WTCRAH, WTCRAL, WTCRBH, and WTCRBL).....	124
6.3.4	Read Strobe Timing Control Register (RDNCR).....	129
6.3.5	$\overline{CS}$ Assertion Period Control Registers H, L (CSACRH, CSACRL).....	130
6.3.6	Area 0 Burst ROM Interface Control Register (BROMCRH) Area 1 Burst ROM Interface Control Register (BROMCRL).....	132
6.3.7	Bus Control Register (BCR).....	133
6.3.8	DRAM Control Register (DRAMCR).....	135
6.3.9	DRAM Access Control Register (DRACCR).....	140
6.3.10	Refresh Control Register (REFCR).....	141
6.3.11	Refresh Timer Counter (RTCNT).....	144

6.4.1	Area Division.....	144
6.4.2	Bus Specifications.....	146
6.4.3	Memory Interfaces.....	148
6.4.4	Chip Select Signals.....	149
6.5	Basic Bus Interface.....	150
6.5.1	Data Size and Data Alignment.....	150
6.5.2	Valid Strobes.....	152
6.5.3	Basic Timing.....	153
6.5.4	Wait Control.....	161
6.5.5	Read Strobe ( $\overline{RD}$ ) Timing.....	162
6.5.6	Extension of Chip Select ( $\overline{CS}$ ) Assertion Period.....	163
6.6	DRAM Interface.....	165
6.6.1	Setting DRAM Space.....	165
6.6.2	Address Multiplexing.....	165
6.6.3	Data Bus.....	166
6.6.4	Pins Used for DRAM Interface.....	167
6.6.5	Basic Timing.....	168
6.6.6	Column Address Output Cycle Control.....	169
6.6.7	Row Address Output State Control.....	170
6.6.8	Precharge State Control.....	172
6.6.9	Wait Control.....	173
6.6.10	Byte Access Control.....	176
6.6.11	Burst Operation.....	177
6.6.12	Refresh Control.....	182
6.6.13	DMAC Single Address Transfer Mode and DRAM Interface.....	187
6.7	Burst ROM Interface.....	190
6.7.1	Basic Timing.....	190
6.7.2	Wait Control.....	192
6.7.3	Write Access.....	192
6.8	Idle Cycle.....	193
6.8.1	Operation.....	193
6.8.2	Pin States in Idle Cycle.....	204
6.9	Write Data Buffer Function.....	204
6.10	Bus Release.....	206
6.10.1	Operation.....	206
6.10.2	Pin States in External Bus Released State.....	207
6.10.3	Transition Timing.....	208
6.11	Bus Arbitration.....	209

6.12	Bus Controller Operation in Reset .....	211
6.13	Usage Notes .....	211
6.13.1	External Bus Release Function and All-Module-Clocks-Stopped Mode .....	211
6.13.2	External Bus Release Function and Software Standby .....	211
6.13.3	External Bus Release Function and CBR Refreshing .....	211
6.13.4	BRE $\overline{Q}$ O Output Timing .....	212
<b>Section 7 DMA Controller (DMAC) .....</b>		<b>213</b>
7.1	Features .....	213
7.2	Input/Output Pins .....	215
7.3	Register Descriptions .....	215
7.3.1	Memory Address Registers (MARA and MARB) .....	217
7.3.2	I/O Address Registers (IOARA and IOARB) .....	217
7.3.3	Execute Transfer Count Registers (ETCRA and ETCRB).....	218
7.3.4	DMA Control Registers (DMACRA and DMACRB) .....	219
7.3.5	DMA Band Control Registers H and L (DMABCRH and DMABCRL).....	226
7.3.6	DMA Write Enable Register (DMAWER) .....	238
7.3.7	DMA Terminal Control Register (DMATCR).....	240
7.4	Activation Sources .....	241
7.4.1	Activation by Internal Interrupt Request .....	242
7.4.2	Activation by External Request .....	242
7.4.3	Activation by Auto-Request .....	243
7.5	Operation.....	244
7.5.1	Transfer Modes .....	244
7.5.2	Sequential Mode .....	246
7.5.3	Idle Mode .....	249
7.5.4	Repeat Mode .....	252
7.5.5	Single Address Mode .....	256
7.5.6	Normal Mode.....	259
7.5.7	Block Transfer Mode .....	262
7.5.8	Basic Bus Cycles.....	267
7.5.9	DMA Transfer (Dual Address Mode) Bus Cycles .....	267
7.5.10	DMA Transfer (Single Address Mode) Bus Cycles.....	275
7.5.11	Write Data Buffer Function .....	281
7.5.12	Multi-Channel Operation .....	282
7.5.13	Relation between DMAC and External Bus Requests and Refresh Cycles .....	283
7.5.14	DMAC and NMI Interrupts.....	284
7.5.15	Forced Termination of DMAC Operation.....	285

7.7	Usage Notes .....	288
7.7.1	DMAC Register Access during Operation.....	288
7.7.2	Module Stop.....	289
7.7.3	Write Data Buffer Function .....	289
7.7.4	$\overline{\text{TEND}}$ Output.....	290
7.7.5	Activation by Falling Edge on $\overline{\text{DREQ}}$ Pin .....	291
7.7.6	Activation Source Acceptance.....	291
7.7.7	Internal Interrupt after End of Transfer.....	292
7.7.8	Channel Re-Setting.....	292
<b>Section 8 Data Transfer Controller (DTC).....</b>		<b>293</b>
8.1	Features.....	293
8.2	Register Descriptions .....	295
8.2.1	DTC Mode Register A (MRA) .....	295
8.2.2	DTC Mode Register B (MRB).....	297
8.2.3	DTC Source Address Register (SAR).....	297
8.2.4	DTC Destination Address Register (DAR).....	297
8.2.5	DTC Transfer Count Register A (CRA) .....	298
8.2.6	DTC Transfer Count Register B (CRB).....	298
8.2.7	DTC Enable Registers A to G (DTCERA to DTCERG) .....	299
8.2.8	DTC Vector Register (DTVECR).....	300
8.3	Activation Sources .....	301
8.4	Location of Register Information and DTC Vector Table .....	302
8.5	Operation .....	306
8.5.1	Normal Mode.....	308
8.5.2	Repeat Mode .....	309
8.5.3	Block Transfer Mode .....	310
8.5.4	Chain Transfer .....	311
8.5.5	Interrupts.....	312
8.5.6	Operation Timing.....	313
8.5.7	Number of DTC Execution States .....	314
8.6	Procedures for Using DTC.....	316
8.6.1	Activation by Interrupt.....	316
8.6.2	Activation by Software .....	316
8.7	Examples of Use of the DTC .....	317
8.7.1	Normal Mode.....	317
8.7.2	Chain Transfer .....	317
8.7.3	Chain Transfer when Counter = 0.....	318

8.8.1	Module Stop Mode Setting .....	321
8.8.2	On-Chip RAM .....	321
8.8.3	DTCE Bit Setting .....	321
8.8.4	DMAC Transfer End Interrupt.....	321
8.8.5	Chain Transfer .....	321
Section 9 I/O Ports .....		323
9.1	Port 1 .....	327
9.1.1	Port 1 Data Direction Register (P1DDR).....	327
9.1.2	Port 1 Data Register (P1DR).....	328
9.1.3	Port 1 Register (PORT1).....	328
9.1.4	Pin Functions .....	329
9.2	Port 2.....	337
9.2.1	Port 2 Data Direction Register (P2DDR).....	337
9.2.2	Port 2 Data Register (P2DR).....	338
9.2.3	Port 2 Register (PORT2).....	338
9.2.4	Pin Functions .....	339
9.3	Port 3.....	347
9.3.1	Port 3 Data Direction Register (P3DDR).....	347
9.3.2	Port 3 Data Register (P3DR).....	348
9.3.3	Port 3 Register (PORT3).....	348
9.3.4	Port 3 Open Drain Control Register (P3ODR).....	349
9.3.5	Port Function Control Register 2 (PFCR2).....	350
9.3.6	Pin Functions .....	351
9.4	Port 4.....	354
9.4.1	Port 4 Register (PORT4).....	354
9.4.2	Pin Functions .....	355
9.5	Port 5.....	357
9.5.1	Port 5 Data Direction Register (P5DDR).....	357
9.5.2	Port 5 Data Register (P5DR).....	357
9.5.3	Port 5 Register (PORT5).....	358
9.5.4	Pin Functions .....	358
9.6	Port 8.....	360
9.6.1	Port 8 Data Direction Register (P8DDR).....	360
9.6.2	Port 8 Data Register (P8DR).....	361
9.6.3	Port 8 Register (PORT8).....	361
9.6.4	Pin Functions .....	362
9.7	Port 9.....	363

9.8	Port A	364
9.8.1	Port A Data Direction Register (PADDDR)	365
9.8.2	Port A Data Register (PADR)	366
9.8.3	Port A Register (PORTA)	366
9.8.4	Port A MOS Pull-Up Control Register (PAPCR)	367
9.8.5	Port A Open Drain Control Register (PAODR)	367
9.8.6	Port Function Control Register 0 (PFCR0)	368
9.8.7	Port Function Control Register 1 (PFCR1)	369
9.8.8	Pin Functions	370
9.8.9	Port A MOS Input Pull-Up States	372
9.9	Port B	373
9.9.1	Port B Data Direction Register (PBDDR)	373
9.9.2	Port B Data Register (PBDR)	374
9.9.3	Port B Register (PORTB)	374
9.9.4	Port B MOS Pull-Up Control Register (PBPCR)	375
9.9.5	Pin Functions	375
9.9.6	Port B MOS Input Pull-Up States	376
9.10	Port C	377
9.10.1	Port C Data Direction Register (PCDDR)	377
9.10.2	Port C Data Register (PCDR)	378
9.10.3	Port C Register (PORTC)	378
9.10.4	Port C MOS Pull-Up Control Register (PCPCR)	379
9.10.5	Pin Functions	379
9.10.6	Port C MOS Input Pull-Up States	380
9.11	Port D	381
9.11.1	Port D Data Direction Register (PDDDR)	381
9.11.2	Port D Data Register (PDDR)	382
9.11.3	Port D Register (PORTD)	382
9.11.4	Port D Pull-up Control Register (PDPCR)	383
9.11.5	Pin Functions	383
9.11.6	Port D MOS Input Pull-Up States	384
9.12	Port E	385
9.12.1	Port E Data Direction Register (PEDDR)	385
9.12.2	Port E Data Register (PEDR)	386
9.12.3	Port E Register (PORTE)	386
9.12.4	Port E Pull-up Control Register (PEPCR)	387
9.12.5	Pin Functions	387
9.12.6	Port E MOS Input Pull-Up States	388

9.13.2	Port F Data Register (PFDR) .....	390
9.13.3	Port F Register (PORTF) .....	390
9.13.4	Pin Functions .....	391
9.14	Port G .....	395
9.14.1	Port G Data Direction Register (PGDDR) .....	395
9.14.2	Port G Data Register (PGDR) .....	396
9.14.3	Port G Register (PORTG) .....	396
9.14.4	Pin Functions .....	397
<b>Section 10 16-Bit Timer Pulse Unit (TPU).....</b>		<b>401</b>
10.1	Features .....	401
10.2	Input/Output Pins .....	405
10.3	Register Descriptions .....	406
10.3.1	Timer Control Register (TCR) .....	408
10.3.2	Timer Mode Register (TMDR) .....	413
10.3.3	Timer I/O Control Register (TIOR) .....	414
10.3.4	Timer Interrupt Enable Register (TIER) .....	432
10.3.5	Timer Status Register (TSR) .....	434
10.3.6	Timer Counter (TCNT) .....	437
10.3.7	Timer General Register (TGR) .....	437
10.3.8	Timer Start Register (TSTR) .....	437
10.3.9	Timer Synchronous Register (TSYR) .....	438
10.4	Operation .....	439
10.4.1	Basic Functions .....	439
10.4.2	Synchronous Operation .....	445
10.4.3	Buffer Operation .....	447
10.4.4	Cascaded Operation .....	452
10.4.5	PWM Modes .....	454
10.4.6	Phase Counting Mode .....	459
10.5	Interrupts .....	466
10.6	DTC Activation .....	468
10.7	DMAC Activation .....	468
10.8	A/D Converter Activation .....	468
10.9	Operation Timing .....	469
10.9.1	Input/Output Timing .....	469
10.9.2	Interrupt Signal Timing .....	473
10.10	Usage Notes .....	477
10.10.1	Module Stop Mode Setting .....	477

10.10.4	Contention between TCNT Write and Clear Operations .....	478
10.10.5	Contention between TCNT Write and Increment Operations.....	479
10.10.6	Contention between TGR Write and Compare Match .....	479
10.10.7	Contention between Buffer Register Write and Compare Match .....	480
10.10.8	Contention between TGR Read and Input Capture.....	481
10.10.9	Contention between TGR Write and Input Capture.....	482
10.10.10	Contention between Buffer Register Write and Input Capture .....	482
10.10.11	Contention between Overflow/Underflow and Counter Clearing.....	483
10.10.12	Contention between TCNT Write and Overflow/Underflow.....	484
10.10.13	Multiplexing of I/O Pins .....	484
10.10.14	Interrupts and Module Stop Mode .....	484
<b>Section 11</b>	<b>Programmable Pulse Generator (PPG) .....</b>	<b>485</b>
11.1	Features.....	485
11.2	Input/Output Pins .....	487
11.3	Register Descriptions .....	487
11.3.1	Next Data Enable Registers H, L (NDERH, NDERL) .....	488
11.3.2	Output Data Registers H, L (PODRH, PODRL).....	489
11.3.3	Next Data Registers H, L (NDRH, NDRL) .....	490
11.3.4	PPG Output Control Register (PCR) .....	492
11.3.5	PPG Output Mode Register (PMR) .....	493
11.4	Operation .....	495
11.4.1	Output Timing.....	496
11.4.2	Sample Setup Procedure for Normal Pulse Output.....	497
11.4.3	Example of Normal Pulse Output (Example of Five-Phase Pulse Output).....	498
11.4.4	Non-Overlapping Pulse Output.....	499
11.4.5	Sample Setup Procedure for Non-Overlapping Pulse Output.....	501
11.4.6	Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output) .....	502
11.4.7	Inverted Pulse Output .....	504
11.4.8	Pulse Output Triggered by Input Capture .....	505
11.5	Usage Notes .....	505
11.5.1	Module Stop Mode Setting.....	505
11.5.2	Operation of Pulse Output Pins.....	505
<b>Section 12</b>	<b>8-Bit Timers (TMR) .....</b>	<b>507</b>
12.1	Features.....	507
12.2	Input/Output Pins .....	509



12.3.2	Time Constant Register A (TCORA).....	510
12.3.3	Time Constant Register B (TCORB).....	510
12.3.4	Timer Control Register (TCR).....	510
12.3.5	Timer Control/Status Register (TCSR).....	512
12.4	Operation.....	515
12.4.1	Pulse Output.....	515
12.5	Operation Timing.....	516
12.5.1	TCNT Incrementation Timing.....	516
12.5.2	Timing of CMFA and CMFB Setting when Compare-Match Occurs.....	517
12.5.3	Timing of Timer Output when Compare-Match Occurs.....	518
12.5.4	Timing of Compare Match Clear.....	518
12.5.5	Timing of TCNT External Reset.....	519
12.5.6	Timing of Overflow Flag (OVF) Setting.....	519
12.6	Operation with Cascaded Connection.....	520
12.6.1	16-Bit Counter Mode.....	520
12.6.2	Compare Match Count Mode.....	520
12.7	Interrupts.....	521
12.7.1	Interrupt Sources and DTC Activation.....	521
12.7.2	A/D Converter Activation.....	521
12.8	Usage Notes.....	522
12.8.1	Contention between TCNT Write and Clear.....	522
12.8.2	Contention between TCNT Write and Increment.....	523
12.8.3	Contention between TCOR Write and Compare Match.....	524
12.8.4	Contention between Compare Matches A and B.....	525
12.8.5	Switching of Internal Clocks and TCNT Operation.....	525
12.8.6	Mode Setting with Cascaded Connection.....	527
12.8.7	Interrupts in Module Stop Mode.....	527
 Section 13 Watchdog Timer.....		 529
13.1	Features.....	529
13.2	Input/Output Pin.....	530
13.3	Register Descriptions.....	531
13.3.1	Timer Counter (TCNT).....	531
13.3.2	Timer Control/Status Register (TCSR).....	531
13.3.3	Reset Control/Status Register (RSTCSR).....	533
13.4	Operation.....	534
13.4.1	Watchdog Timer Mode.....	534
13.4.2	Interval Timer Mode.....	535

13.6.1	Notes on Register Access.....	536
13.6.2	Contention between Timer Counter (TCNT) Write and Increment .....	538
13.6.3	Changing Value of CKS2 to CKS0.....	538
13.6.4	Switching between Watchdog Timer Mode and Interval Timer Mode.....	538
13.6.5	Internal Reset in <u>Watchdog</u> Timer Mode.....	539
13.6.6	System Reset by <u>WDTOVF</u> Signal.....	539
Section 14	Serial Communication Interface (SCI, IrDA).....	541
14.1	Features.....	541
14.2	Input/Output Pins .....	544
14.3	Register Descriptions .....	545
14.3.1	Receive Shift Register (RSR) .....	546
14.3.2	Receive Data Register (RDR) .....	546
14.3.3	Transmit Data Register (TDR).....	546
14.3.4	Transmit Shift Register (TSR).....	547
14.3.5	Serial Mode Register (SMR) .....	547
14.3.6	Serial Control Register (SCR).....	551
14.3.7	Serial Status Register (SSR) .....	556
14.3.8	Smart Card Mode Register (SCMR).....	563
14.3.9	Bit Rate Register (BRR) .....	564
14.3.10	IrDA Control Register (IrCR).....	573
14.3.11	Serial Extension Mode Register (SEMR) .....	574
14.4	Operation in Asynchronous Mode .....	576
14.4.1	Data Transfer Format.....	576
14.4.2	Receive Data Sampling Timing and Reception Margin in Asynchronous Mode .....	578
14.4.3	Clock.....	579
14.4.4	SCI Initialization (Asynchronous Mode).....	580
14.4.5	Data Transmission (Asynchronous Mode).....	581
14.4.6	Serial Data Reception (Asynchronous Mode).....	584
14.5	Multiprocessor Communication Function.....	588
14.5.1	Multiprocessor Serial Data Transmission .....	589
14.5.2	Multiprocessor Serial Data Reception .....	591
14.6	Operation in Clocked Synchronous Mode .....	595
14.6.1	Clock.....	595
14.6.2	SCI Initialization (Clocked Synchronous Mode).....	596
14.6.3	Serial Data Transmission (Clocked Synchronous Mode) .....	597
14.6.4	Serial Data Reception (Clocked Synchronous Mode).....	600

14.7	Operation in Smart Card Interface Mode.....	604
14.7.1	Pin Connection Example.....	604
14.7.2	Data Format (Except for Block Transfer Mode).....	605
14.7.3	Block Transfer Mode.....	606
14.7.4	Receive Data Sampling Timing and Reception Margin.....	606
14.7.5	Initialization.....	608
14.7.6	Data Transmission (Except for Block Transfer Mode).....	608
14.7.7	Serial Data Reception (Except for Block Transfer Mode).....	612
14.7.8	Clock Output Control.....	613
14.8	IrDA Operation.....	616
14.9	SCI Interrupts.....	619
14.9.1	Interrupts in Normal Serial Communication Interface Mode.....	619
14.9.2	Interrupts in Smart Card Interface Mode.....	621
14.10	Usage Notes.....	622
14.10.1	Module Stop Mode Setting.....	622
14.10.2	Break Detection and Processing.....	622
14.10.3	Mark State and Break Sending.....	622
14.10.4	Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only).....	623
14.10.5	Relation between Writes to TDR and the TDRE Flag.....	623
14.10.6	Restrictions on Use of DMAC or DTC.....	623
14.10.7	Operation in Case of Mode Transition.....	624
Section 15 I <sup>2</sup> C Bus Interface2 (IIC2) (Option).....		629
15.1	Features.....	629
15.2	Input/Output Pins.....	631
15.3	Register Descriptions.....	632
15.3.1	I <sup>2</sup> C Bus Control Register A (ICCRA).....	633
15.3.2	I <sup>2</sup> C Bus Control Register B (ICCRB).....	635
15.3.3	I <sup>2</sup> C Bus Mode Register (ICMR).....	636
15.3.4	I <sup>2</sup> C Bus Interrupt Enable Register (ICIER).....	637
15.3.5	I <sup>2</sup> C Bus Status Register (ICSR).....	639
15.3.6	Slave Address Register (SAR).....	641
15.3.7	I <sup>2</sup> C Bus Transmit Data Register (ICDRT).....	642
15.3.8	I <sup>2</sup> C Bus Receive Data Register (ICDRR).....	642
15.3.9	I <sup>2</sup> C Bus Shift Register (ICDRS).....	642
15.4	Operation.....	643
15.4.1	I <sup>2</sup> C Bus Format.....	643

15.4.4	Slave Transmit Operation .....	648
15.4.5	Slave Receive Operation.....	650
15.4.6	Noise Canceler.....	653
15.4.7	Example of Use.....	653
15.5	Interrupt Request.....	658
15.6	Bit Synchronous Circuit.....	658
15.7	Usage Notes .....	659
<b>Section 16 A/D Converter.....</b>		<b>661</b>
16.1	Features.....	661
16.2	Input/Output Pins .....	663
16.3	Register Descriptions .....	664
16.3.1	A/D Data Registers A to H (ADDRA to ADDRH) .....	664
16.3.2	A/D Control/Status Register (ADCSR) .....	666
16.3.3	A/D Control Register (ADCR) .....	668
16.4	Operation .....	669
16.4.1	Single Mode.....	669
16.4.2	Scan Mode .....	669
16.4.3	Input Sampling and A/D Conversion Time .....	670
16.4.4	External Trigger Input Timing .....	673
16.5	Interrupts.....	673
16.6	A/D Conversion Precision Definitions.....	674
16.7	Usage Notes .....	676
16.7.1	Module Stop Mode Setting .....	676
16.7.2	Permissible Signal Source Impedance .....	676
16.7.3	Influences on Absolute Precision.....	677
16.7.4	Setting Range of Analog Power Supply and Other Pins .....	677
16.7.5	Notes on Board Design .....	677
16.7.6	Notes on Noise Countermeasures .....	678
<b>Section 17 D/A Converter.....</b>		<b>681</b>
17.1	Features.....	681
17.2	Input/Output Pins .....	683
17.3	Register Descriptions .....	683
17.3.1	D/A Data Registers 2 and 3 (DADR2 and DADR3).....	683
17.3.2	D/A Control Register 23 (DACR23) .....	684
17.4	Operation .....	685
17.5	Usage Notes .....	686

Section 18	RAM .....	687
Section 19	Flash Memory (0.35- $\mu$ m F-ZTAT Version).....	689
19.1	Features .....	689
19.2	Mode Transitions .....	691
19.3	Block Configuration.....	695
19.4	Input/Output Pins .....	697
19.5	Register Descriptions .....	697
19.5.1	Flash Memory Control Register 1 (FLMCR1).....	698
19.5.2	Flash Memory Control Register 2 (FLMCR2).....	699
19.5.3	Erase Block Register 1 (EBR1) .....	700
19.5.4	Erase Block Register 2 (EBR2) .....	700
19.6	On-Board Programming Modes .....	702
19.6.1	Boot Mode .....	702
19.6.2	User Program Mode.....	705
19.7	Flash Memory Programming/Erasing .....	706
19.7.1	Program/Program-Verify .....	706
19.7.2	Erase/Erase-Verify.....	708
19.7.3	Interrupt Handling when Programming/Erasing Flash Memory.....	708
19.8	Program/Erase Protection.....	710
19.8.1	Hardware Protection .....	710
19.8.2	Software Protection.....	710
19.8.3	Error Protection.....	710
19.9	Programmer Mode .....	711
19.10	Power-Down States for Flash Memory.....	711
19.11	Usage Notes .....	712
Section 20	Flash Memory (0.18- $\mu$ m F-ZTAT Version).....	717
20.1	Features .....	717
20.1.1	Operating Mode .....	720
20.1.2	Mode Comparison.....	721
20.1.3	Flash MAT Configuration.....	722
20.1.4	Block Division .....	723
20.1.5	Programming/Erasing Interface .....	724
20.2	Input/Output Pins .....	726
20.3	Register Descriptions .....	726
20.3.1	Programming/Erasing Interface Register .....	728

20.4	On-Board Programming Mode .....	748
20.4.1	Boot Mode .....	748
20.4.2	User Program Mode.....	753
20.4.3	User Boot Mode.....	764
20.4.4	Procedure Program and Storable Area for Programming Data .....	768
20.5	Protection .....	778
20.5.1	Hardware Protection .....	778
20.5.2	Software Protection.....	779
20.5.3	Error Protection.....	779
20.6	Switching between User MAT and User Boot MAT .....	781
20.7	Programmer Mode .....	782
20.8	Serial Communication Interface Specification for Boot Mode.....	782
20.9	Usage Notes .....	810
Section 21 Mask ROM.....		811
Section 22 Clock Pulse Generator .....		813
22.1	Register Descriptions .....	813
22.1.1	System Clock Control Register (SCKCR) .....	814
22.1.2	PLL Control Register (PLLCR).....	815
22.2	Oscillator.....	816
22.2.1	Connecting a Crystal Oscillator .....	816
22.2.2	External Clock Input.....	817
22.3	PLL Circuit .....	818
22.4	Frequency Divider .....	819
22.5	Usage Notes .....	819
22.5.1	Notes on Clock Pulse Generator.....	819
22.5.2	Notes on Oscillator .....	820
22.5.3	Notes on Board Design .....	820
Section 23 Power-Down Modes.....		823
23.1	Register Descriptions .....	826
23.1.1	Standby Control Register (SBYCR) .....	826
23.1.2	Module Stop Control Registers H and L (MSTPCRH, MSTPCRL).....	828
23.1.3	Extension Module Stop Control Registers H and L (EXMSTPCRH, EXMSTPCRL) .....	829
23.2	Operation .....	830
23.2.1	Clock Division Mode.....	830

23.2.4	Hardware Standby Mode .....	834
23.2.5	Module Stop Mode .....	835
23.2.6	All-Module-Clocks-Stop Mode .....	836
23.3	$\phi$ Clock Output Control.....	836
23.4	Usage Notes .....	837
23.4.1	I/O Port Status.....	837
23.4.2	Current Dissipation during Oscillation Stabilization Standby Period.....	837
23.4.3	DMAC/DTC Module Stop.....	837
23.4.4	On-Chip Peripheral Module Interrupts .....	837
23.4.5	Writing to MSTPCR, EXMSTPCR .....	837
23.4.6	Notes on Clock Division Mode.....	838
Section 24 List of Registers.....		839
24.1	Register Addresses (Address Order).....	840
24.2	Register Bits.....	850
24.3	Register States in Each Operating Mode.....	863
Section 25 Electrical Characteristics.....		873
25.1	Electrical Characteristics of Masked ROM and ROMless Versions .....	873
25.1.1	Absolute Maximum Ratings .....	873
25.1.2	DC Characteristics .....	874
25.1.3	AC Characteristics .....	877
25.1.4	A/D Conversion Characteristics.....	910
25.1.5	D/A Conversion Characteristics.....	910
25.2	Electrical Characteristics of 0.35 $\mu$ m F-ZTAT Version.....	911
25.2.1	Absolute Maximum Ratings .....	911
25.2.2	DC Characteristics .....	912
25.2.3	A/D Conversion Characteristics.....	923
25.2.4	D/A Conversion Characteristics.....	923
25.2.5	Flash Memory Characteristics.....	924
25.3	Electrical Characteristics for 0.18 $\mu$ m F-ZTAT Version.....	926
25.3.1	Absolute Maximum Ratings .....	926
25.3.2	DC Characteristics .....	927
25.3.3	AC Characteristics .....	930
25.3.4	A/D Conversion Characteristics.....	938
25.3.5	D/A Conversion Characteristics.....	938
25.3.6	Flash Memory Characteristics.....	939
25.4	Usage Note.....	940

B.	Product Lineup.....	949
C.	Package Dimensions .....	950
D.	Bus State during Execution of Instructions.....	952
Index	.....	975



## Section 1 Overview

Figure 1.1	Internal Block Diagram of H8S/2367F, H8S/2365, and H8S/2363.....	3
Figure 1.2	Internal Block Diagram of H8S/2368 0.18 $\mu$ m F-ZTAT Group.....	4
Figure 1.3	Pin Arrangement of H8S/2367F, H8S/2365, and H8S/2363 .....	5
Figure 1.4	Pin Arrangement of H8S/2368 0.18 $\mu$ m F-ZTAT Group.....	6
Figure 1.5	Pin Arrangement of H8S/2367F, H8S/2365, and H8S/2363 .....	7

## Section 2 CPU

Figure 2.1	Exception Vector Table (Normal Mode).....	25
Figure 2.2	Stack Structure in Normal Mode.....	25
Figure 2.3	Exception Vector Table (Advanced Mode).....	26
Figure 2.4	Stack Structure in Advanced Mode.....	27
Figure 2.5	Memory Map.....	28
Figure 2.6	CPU Internal Registers.....	29
Figure 2.7	Usage of General Registers .....	30
Figure 2.8	Stack.....	31
Figure 2.9	General Register Data Formats (1).....	34
Figure 2.9	General Register Data Formats (2).....	35
Figure 2.10	Memory Data Formats.....	36
Figure 2.11	Instruction Formats (Examples) .....	48
Figure 2.12	Branch Address Specification in Memory Indirect Addressing Mode.....	52
Figure 2.13	State Transitions.....	56

## Section 3 MCU Operating Modes

Figure 3.1	H8S/2368F Memory Map (1).....	63
Figure 3.2	H8S/2368F Memory Map (2).....	64
Figure 3.3	H8S/2367F Memory Map (1).....	65
Figure 3.4	H8S/2367F Memory Map (2).....	66
Figure 3.5	H8S/2364F Memory Map (1).....	67
Figure 3.6	H8S/2364F Memory Map (2).....	68
Figure 3.7	H8S/2362F Memory Map (1).....	69
Figure 3.8	H8S/2362F Memory Map (2).....	70
Figure 3.9	H8S/2361F Memory Map (1).....	71
Figure 3.10	H8S/2361F Memory Map (2).....	72
Figure 3.11	H8S/2360F Memory Map (1).....	73
Figure 3.12	H8S/2360F Memory Map (2).....	74
Figure 3.13	H8S/2365 Memory Map (1).....	75

## Section 4 Exception Handling

Figure 4.1	Reset Sequence (Advanced Mode with On-Chip ROM Enabled).....	82
Figure 4.2	Reset Sequence (Advanced Mode with On-Chip ROM Disabled).....	83
Figure 4.3	Stack Status after Exception Handling .....	86
Figure 4.4	Operation when SP Value Is Odd.....	87

## Section 5 Interrupt Controller

Figure 5.1	Block Diagram of Interrupt Controller.....	90
Figure 5.2	Block Diagram of Interrupts IRQ15 to IRQ0.....	102
Figure 5.3	Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0 .....	109
Figure 5.4	Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2 .....	111
Figure 5.5	Interrupt Exception Handling.....	113
Figure 5.6	Contention between Interrupt Generation and Disabling .....	116

## Section 6 Bus Controller (BSC)

Figure 6.1	Block Diagram of Bus Controller .....	120
Figure 6.2	Read Strobe Negation Timing (Example of 3-State Access Space).....	129
Figure 6.3	$\overline{CS}$ and Address Assertion Period Extension (Example of 3-State Access Space and $RDNn = 0$ ).....	131
Figure 6.4	$\overline{RAS}$ Signal Assertion Timing (2-State Column Address Output Cycle, Full Access).....	139
Figure 6.5	Area Divisions.....	145
Figure 6.6	$\overline{CSn}$ Signal Output Timing ( $n = 0$ to 7) .....	150
Figure 6.7	Access Sizes and Data Alignment Control (8-Bit Access Space) .....	151
Figure 6.8	Access Sizes and Data Alignment Control (16-bit Access Space).....	151
Figure 6.9	Bus Timing for 8-Bit, 2-State Access Space .....	153
Figure 6.10	Bus Timing for 8-Bit, 3-State Access Space .....	154
Figure 6.11	Bus Timing for 16-Bit, 2-State Access Space (Even Address Byte Access).....	155
Figure 6.12	Bus Timing for 16-Bit, 2-State Access Space (Odd Address Byte Access).....	156
Figure 6.13	Bus Timing for 16-Bit, 2-State Access Space (Word Access) .....	157
Figure 6.14	Bus Timing for 16-Bit, 3-State Access Space (Even Address Byte Access).....	158
Figure 6.15	Bus Timing for 16-Bit, 3-State Access Space (Odd Address Byte Access).....	159
Figure 6.16	Bus Timing for 16-Bit, 3-State Access Space (Word Access) .....	160
Figure 6.17	Example of Wait State Insertion Timing.....	162
Figure 6.18	Example of Read Strobe Timing .....	163

Figure 6.21	Example of Access Timing with 3-State Column Address Output Cycle (RAST = 0).....	169
Figure 6.22	Example of Access Timing when $\overline{\text{RAS}}$ Signal Goes Low from Beginning of T <sub>r</sub> State (CAST = 0) .....	170
Figure 6.23	Example of Timing with One Row Address Output Maintenance State (RAST = 0, CAST = 0) .....	171
Figure 6.24	Example of Timing with Two-State Precharge Cycle (RAST = 0, CAST = 0).....	172
Figure 6.25	Example of Wait State Insertion Timing (2-State Column Address Output) .....	174
Figure 6.26	Example of Wait State Insertion Timing (3-State Column Address Output) .....	175
Figure 6.27	2-CAS Control Timing (Upper Byte Write Access: RAST = 0, CAST = 0).....	176
Figure 6.28	Example of 2-CAS DRAM Connection .....	177
Figure 6.29	Operation Timing in Fast Page Mode (RAST = 0, CAST = 0) .....	178
Figure 6.30	Operation Timing in Fast Page Mode (RAST = 0, CAST = 1) .....	179
Figure 6.31	Example of Operation Timing in RAS Down Mode (RAST = 0, CAST = 0).....	180
Figure 6.32	Example of Operation Timing in RAS Up Mode (RAST = 0, CAST = 0).....	181
Figure 6.33	RTCNT Operation.....	182
Figure 6.34	Compare Match Timing .....	183
Figure 6.35	CBR Refresh Timing.....	183
Figure 6.36	CBR Refresh Timing (RCW1 = 0, RCW0 = 1, RLW1 = 0, RLW0 = 0).....	184
Figure 6.37	Example of CBR Refresh Timing (CBRM = 1) .....	185
Figure 6.38	Self-Refresh Timing.....	186
Figure 6.39	Example of Timing when Precharge Time after Self-Refreshing Is Extended by 2 States .....	187
Figure 6.40	Example of $\overline{\text{DACK}}$ Output Timing when DDS = 1 (RAST = 0, CAST = 0) .....	188
Figure 6.41	Example of $\overline{\text{DACK}}$ Output Timing when DDS = 0 (RAST = 0, CAST = 1) .....	189
Figure 6.42	Example of Burst ROM Access Timing (ASTn = 1, 2-State Burst Cycle) .....	191
Figure 6.43	Example of Burst ROM Access Timing (ASTn = 0, 1-State Burst Cycle) .....	192
Figure 6.44	Example of Idle Cycle Operation (Consecutive Reads in Different Areas).....	193
Figure 6.45	Example of Idle Cycle Operation (Write after Read).....	194
Figure 6.46	Example of Idle Cycle Operation (Read after Write).....	195
Figure 6.47	Relationship between Chip Select ( $\overline{\text{CS}}$ ) and Read ( $\overline{\text{RD}}$ ).....	196
Figure 6.48	Example of DRAM Full Access after External Read (CAST = 0).....	196
Figure 6.49	Example of Idle Cycle Operation in RAS Down Mode (Consecutive Reads in Different Areas) (IDLC = 0, RAST = 0, CAST = 0).....	197
Figure 6.50	Example of Idle Cycle Operation in RAS Down Mode (Write after Read) (IDLC = 0, RAST = 0, CAST = 0) .....	198
Figure 6.51	Example of Idle Cycle Operation after DRAM Access (Consecutive Reads in Different Areas) (IDLC = 0, RAST = 0, CAST = 0).....	199

Figure 6.53	Example of Idle Cycle Operation after DRAM Write Access (IDLC = 0, ICIS1 = 0, RAST = 0, CAST = 0) .....	200
Figure 6.54	Example of Timing for Idle Cycle Insertion in Case of Consecutive Read and Write Accesses to DRAM Space in RAS Down Mode .....	203
Figure 6.55	Example of Timing when Write Data Buffer Function is Used .....	205
Figure 6.56	Bus Released State Transition Timing .....	208

## Section 7 DMA Controller (DMAC)

Figure 7.1	Block Diagram of DMAC .....	214
Figure 7.2	Areas for Register Re-Setting by DTC (Channel 0A) .....	239
Figure 7.3	Operation in Sequential Mode .....	247
Figure 7.4	Example of Sequential Mode Setting Procedure .....	248
Figure 7.5	Operation in Idle Mode .....	249
Figure 7.6	Example of Idle Mode Setting Procedure .....	251
Figure 7.7	Operation in Repeat mode .....	254
Figure 7.8	Example of Repeat Mode Setting Procedure .....	255
Figure 7.9	Operation in Single Address Mode (When Sequential Mode is Specified) .....	257
Figure 7.10	Example of Single Address Mode Setting Procedure (When Sequential Mode is Specified) .....	258
Figure 7.11	Operation in Normal Mode .....	260
Figure 7.12	Example of Normal Mode Setting Procedure .....	261
Figure 7.13	Operation in Block Transfer Mode (BLKDIR = 0) .....	263
Figure 7.14	Operation in Block Transfer Mode (BLKDIR = 1) .....	264
Figure 7.15	Operation Flow in Block Transfer Mode .....	265
Figure 7.16	Example of Block Transfer Mode Setting Procedure .....	266
Figure 7.17	Example of DMA Transfer Bus Timing .....	267
Figure 7.18	Example of Short Address Mode Transfer .....	268
Figure 7.19	Example of Full Address Mode Transfer (Cycle Steal) .....	269
Figure 7.20	Example of Full Address Mode Transfer (Burst Mode) .....	270
Figure 7.21	Example of Full Address Mode Transfer (Block Transfer Mode) .....	271
Figure 7.22	Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Normal Mode Transfer .....	272
Figure 7.23	Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Block Transfer Mode Transfer .....	273
Figure 7.24	Example of $\overline{\text{DREQ}}$ Pin Low Level Activated Normal Mode Transfer .....	274
Figure 7.25	Example of $\overline{\text{DREQ}}$ Pin Low Level Activated Block Transfer Mode Transfer .....	275
Figure 7.26	Example of Single Address Mode Transfer (Byte Read) .....	276
Figure 7.27	Example of Single Address Mode (Word Read) Transfer .....	276
Figure 7.28	Example of Single Address Mode Transfer (Byte Write) .....	277
Figure 7.29	Example of Single Address Mode Transfer (Word Write) .....	278

Figure 7.32	Example of Dual Address Transfer Using Write Data Buffer Function .....	281
Figure 7.33	Example of Single Address Transfer Using Write Data Buffer Function .....	282
Figure 7.34	Example of Multi-Channel Transfer .....	283
Figure 7.35	Example of Procedure for Continuing Transfer on Channel Interrupted by NMI Interrupt .....	284
Figure 7.36	Example of Procedure for Forcibly Terminating DMAC Operation.....	285
Figure 7.37	Example of Procedure for Clearing Full Address Mode .....	286
Figure 7.38	Block Diagram of Transfer End/Transfer Break Interrupt .....	287
Figure 7.39	DMAC Register Update Timing .....	288
Figure 7.40	Contention between DMAC Register Update and CPU Read.....	289
Figure 7.41	Example in which Low Level is Not Output at $\overline{\text{TEND}}$ Pin .....	291

## Section 8 Data Transfer Controller (DTC)

Figure 8.1	Block Diagram of DTC .....	294
Figure 8.2	Block Diagram of DTC Activation Source Control .....	302
Figure 8.3	Correspondence between DTC Vector Address and Register Information .....	303
Figure 8.4	Correspondence between DTC Vector Address and Register Information .....	303
Figure 8.5	Flowchart of DTC Operation.....	307
Figure 8.6	Memory Mapping in Normal Mode .....	309
Figure 8.7	Memory Mapping in Repeat Mode .....	310
Figure 8.8	Memory Mapping in Block Transfer Mode .....	311
Figure 8.9	Operation of Chain Transfer.....	312
Figure 8.10	DTC Operation Timing (Example in Normal Mode or Repeat Mode).....	313
Figure 8.11	DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2).....	313
Figure 8.12	DTC Operation Timing (Example of Chain Transfer) .....	314
Figure 8.13	Chain Transfer when Counter = 0 .....	319

## Section 10 16-Bit Timer Pulse Unit (TPU)

Figure 10.1	Block Diagram of TPU.....	404
Figure 10.2	Example of Counter Operation Setting Procedure .....	439
Figure 10.3	Free-Running Counter Operation .....	440
Figure 10.4	Periodic Counter Operation.....	441
Figure 10.5	Example of Setting Procedure for Waveform Output by Compare Match.....	442
Figure 10.6	Example of 0 Output/1 Output Operation .....	443
Figure 10.7	Example of Toggle Output Operation .....	443
Figure 10.8	Example of Setting Procedure for Input Capture Operation.....	444
Figure 10.9	Example of Input Capture Operation.....	445

Figure 10.12 Compare Match Buffer Operation .....	448
Figure 10.13 Input Capture Buffer Operation .....	448
Figure 10.14 Example of Buffer Operation Setting Procedure.....	449
Figure 10.15 Example of Buffer Operation (1) .....	450
Figure 10.16 Example of Buffer Operation (2) .....	451
Figure 10.17 Cascaded Operation Setting Procedure .....	452
Figure 10.18 Example of Cascaded Operation (1).....	453
Figure 10.19 Example of Cascaded Operation (2).....	453
Figure 10.20 Example of PWM Mode Setting Procedure .....	456
Figure 10.21 Example of PWM Mode Operation (1) .....	457
Figure 10.22 Example of PWM Mode Operation (2) .....	457
Figure 10.23 Example of PWM Mode Operation (3) .....	458
Figure 10.24 Example of Phase Counting Mode Setting Procedure.....	460
Figure 10.25 Example of Phase Counting Mode 1 Operation .....	461
Figure 10.26 Example of Phase Counting Mode 2 Operation .....	462
Figure 10.27 Example of Phase Counting Mode 3 Operation .....	463
Figure 10.28 Example of Phase Counting Mode 4 Operation .....	464
Figure 10.29 Phase Counting Mode Application Example.....	465
Figure 10.30 Count Timing in Internal Clock Operation .....	469
Figure 10.31 Count Timing in External Clock Operation .....	469
Figure 10.32 Output Compare Output Timing .....	470
Figure 10.33 Input Capture Input Signal Timing .....	470
Figure 10.34 Counter Clear Timing (Compare Match) .....	471
Figure 10.35 Counter Clear Timing (Input Capture).....	471
Figure 10.36 Buffer Operation Timing (Compare Match) .....	472
Figure 10.37 Buffer Operation Timing (Input Capture) .....	472
Figure 10.38 TGI Interrupt Timing (Compare Match) .....	473
Figure 10.39 TGI Interrupt Timing (Input Capture).....	474
Figure 10.40 TCIV Interrupt Setting Timing.....	475
Figure 10.41 TCIU Interrupt Setting Timing.....	475
Figure 10.42 Timing for Status Flag Clearing by CPU .....	476
Figure 10.43 Timing for Status Flag Clearing by DTC/DMAC* Activation .....	476
Figure 10.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode .....	477
Figure 10.45 Contention between TCNT Write and Clear Operations.....	478
Figure 10.46 Contention between TCNT Write and Increment Operations .....	479
Figure 10.47 Contention between TGR Write and Compare Match .....	480
Figure 10.48 Contention between Buffer Register Write and Compare Match.....	480
Figure 10.49 Contention between TGR Read and Input Capture .....	481

Figure 10.52 Contention between Overflow and Counter Clearing .....	483
Figure 10.53 Contention between TCNT Write and Overflow.....	484

## Section 11 Programmable Pulse Generator (PPG)

Figure 11.1 Block Diagram of PPG.....	486
Figure 11.2 Overview Diagram of PPG.....	495
Figure 11.3 Timing of Transfer and Output of NDR Contents (Example).....	496
Figure 11.4 Setup Procedure for Normal Pulse Output (Example).....	497
Figure 11.5 Normal Pulse Output Example (Five-Phase Pulse Output).....	498
Figure 11.6 Non-Overlapping Pulse Output .....	499
Figure 11.7 Non-Overlapping Operation and NDR Write Timing .....	500
Figure 11.8 Setup Procedure for Non-Overlapping Pulse Output (Example).....	501
Figure 11.9 Non-Overlapping Pulse Output Example (Four-Phase Complementary).....	502
Figure 11.10 Inverted Pulse Output (Example) .....	504
Figure 11.11 Pulse Output Triggered by Input Capture (Example).....	505

## Section 12 8-Bit Timers (TMR)

Figure 12.1 Block Diagram of 8-Bit Timer Module .....	508
Figure 12.2 Example of Pulse Output.....	516
Figure 12.3 Count Timing for Internal Clock Input.....	516
Figure 12.4 Count Timing for External Clock Input .....	517
Figure 12.5 Timing of CMF Setting .....	517
Figure 12.6 Timing of Timer Output .....	518
Figure 12.7 Timing of Compare Match Clear .....	518
Figure 12.8 Timing of Clearance by External Reset.....	519
Figure 12.9 Timing of OVF Setting.....	519
Figure 12.10 Contention between TCNT Write and Clear .....	522
Figure 12.11 Contention between TCNT Write and Increment.....	523
Figure 12.12 Contention between TCOR Write and Compare Match.....	524

## Section 13 Watchdog Timer

Figure 13.1 Block Diagram of WDT .....	530
Figure 13.2 Operation in Watchdog Timer Mode.....	535
Figure 13.3 Operation in Interval Timer Mode .....	536
Figure 13.4 Writing to TCNT, TCSR, and RSTCSR.....	537
Figure 13.5 Contention between TCNT Write and Increment.....	538
Figure 13.6 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example).....	539

Figure 14.2	Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits).....	576
Figure 14.3	Receive Data Sampling Timing in Asynchronous Mode .....	578
Figure 14.4	Relation between Output Clock and Transfer Data Phase (Asynchronous Mode).....	579
Figure 14.5	Sample SCI Initialization Flowchart .....	580
Figure 14.6	Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit).....	582
Figure 14.7	Sample Serial Transmission Flowchart .....	583
Figure 14.8	Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit).....	584
Figure 14.9	Sample Serial Reception Data Flowchart (1) .....	586
Figure 14.9	Sample Serial Reception Data Flowchart (2) .....	587
Figure 14.10	Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A) .....	589
Figure 14.11	Sample Multiprocessor Serial Transmission Flowchart .....	590
Figure 14.12	Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit).....	592
Figure 14.13	Sample Multiprocessor Serial Reception Flowchart (1).....	593
Figure 14.13	Sample Multiprocessor Serial Reception Flowchart (2).....	594
Figure 14.14	Data Format in Clocked Synchronous Communication (For LSB-First) .....	595
Figure 14.15	Sample SCI Initialization Flowchart .....	596
Figure 14.16	Sample SCI Transmission Operation in Clocked Synchronous Mode .....	598
Figure 14.17	Sample Serial Transmission Flowchart .....	599
Figure 14.18	Example of SCI Operation in Reception .....	600
Figure 14.19	Sample Serial Reception Flowchart .....	601
Figure 14.20	Sample Flowchart of Simultaneous Serial Transmit and Receive Operations .....	603
Figure 14.21	Schematic Diagram of Smart Card Interface Pin Connections.....	604
Figure 14.22	Normal Smart Card Interface Data Format .....	605
Figure 14.23	Direct Convention (SDIR = SINV = $O/\bar{E} = 0$ ) .....	605
Figure 14.24	Inverse Convention (SDIR = SINV = $O/\bar{E} = 1$ ) .....	606
Figure 14.25	Receive Data Sampling Timing in Smart Card Mode (Using Clock of 372 Times the Bit Rate).....	607
Figure 14.26	Retransfer Operation in SCI Transmit Mode .....	609
Figure 14.27	TEND Flag Generation Timing in Transmission Operation .....	610
Figure 14.28	Example of Transmission Processing Flow .....	611
Figure 14.29	Retransfer Operation in SCI Receive Mode .....	612
Figure 14.30	Example of Reception Processing Flow.....	613



Figure 14.33 Block Diagram of IrDA.....	616
Figure 14.34 IrDA Transmit/Receive Operations.....	617
Figure 14.35 Example of Synchronous Transmission Using DTC.....	623
Figure 14.36 Sample Flowchart for Mode Transition during Transmission.....	625
Figure 14.37 Port Pin States during Mode Transition (Internal Clock, Asynchronous Transmission) .....	626
Figure 14.38 Port Pin States during Mode Transition (Internal Clock, Synchronous Transmission) .....	626
Figure 14.39 Sample Flowchart for Mode Transition during Reception .....	627
 <b>Section 15 I<sup>2</sup>C Bus Interface2 (IIC2) (Option)</b>	
Figure 15.1 Block Diagram of I <sup>2</sup> C Bus Interface2.....	630
Figure 15.2 External Circuit Connections of I/O Pins .....	631
Figure 15.3 I <sup>2</sup> C Bus Formats.....	643
Figure 15.4 I <sup>2</sup> C Bus Timing.....	643
Figure 15.5 Master Transmit Mode Operation Timing 1.....	645
Figure 15.6 Master Transmit Mode Operation Timing 2.....	646
Figure 15.7 Master Receive Mode Operation Timing 1 .....	647
Figure 15.8 Master Receive Mode Operation Timing 2 .....	648
Figure 15.9 Slave Transmit Mode Operation Timing 1 .....	649
Figure 15.10 Slave Transmit Mode Operation Timing 2.....	650
Figure 15.11 Slave Receive Mode Operation Timing 1 .....	651
Figure 15.12 Slave Receive Mode Operation Timing 2 .....	652
Figure 15.13 Block Diagram of Noise Canceler.....	653
Figure 15.14 Sample Flowchart for Master Transmit Mode.....	654
Figure 15.15 Sample Flowchart for Master Receive Mode .....	655
Figure 15.16 Sample Flowchart for Slave Transmit Mode.....	656
Figure 15.17 Sample Flowchart for Slave Receive Mode .....	657
Figure 15.18 Timing of the Bit Synchronous Circuit .....	659
 <b>Section 16 A/D Converter.....</b>	
Figure 16.1 Block Diagram of A/D Converter .....	662
Figure 16.2 A/D Conversion Timing.....	671
Figure 16.3 External Trigger Input Timing .....	673
Figure 16.4 A/D Conversion Precision Definitions .....	675
Figure 16.5 A/D Conversion Precision Definitions .....	675
Figure 16.6 Example of Analog Input Circuit .....	676
Figure 16.7 Example of Analog Input Protection Circuit.....	678

**Section 19 Flash Memory (0.35- $\mu$ m F-ZTAT Version)**

Figure 19.1 Block Diagram of Flash Memory..... 690

Figure 19.2 Flash Memory State Transitions..... 691

Figure 19.3 Boot Mode..... 693

Figure 19.4 User Program Mode ..... 694

Figure 19.5 384-kbyte Flash Memory Block Configuration (Modes 3, 4, and 7)..... 696

Figure 19.6 Programming/Erasing Flowchart Example in User Program Mode..... 705

Figure 19.7 Program/Program-Verify Flowchart ..... 707

Figure 19.8 Erase/Erase-Verify Flowchart ..... 709

Figure 19.9 Power-On/Off Timing ..... 714

Figure 19.10 Mode Transition Timing (Example: Boot Mode  $\rightarrow$  User Mode  $\leftrightarrow$  User Program Mode)..... 715

**Section 20 Flash Memory (0.18- $\mu$ m F-ZTAT Version)**

Figure 20.1 Block Diagram of Flash Memory..... 719

Figure 20.2 Mode Transition of Flash Memory ..... 720

Figure 20.3 Flash Memory Configuration ..... 722

Figure 20.4 Block Division of User MAT..... 723

Figure 20.5 Overview of User Procedure Program ..... 724

Figure 20.6 System Configuration in Boot Mode..... 749

Figure 20.7 Automatic-Bit-Rate Adjustment Operation of SCI ..... 749

Figure 20.8 Overview of Boot Mode State Transition Diagram..... 752

Figure 20.9 Programming/Erasing Overview Flow..... 753

Figure 20.10 RAM Map when Programming/Erasing Is Executed ..... 754

Figure 20.11 Programming Procedure..... 755

Figure 20.12 Erasing Procedure ..... 762

Figure 20.13 Procedure for Programming User MAT in User Boot Mode ..... 765

Figure 20.14 Procedure for Erasing User MAT in User Boot Mode ..... 767

Figure 20.15 Transitions to Error-Protection State..... 780

Figure 20.16 Switching between the User MAT and User Boot MAT ..... 781

Figure 20.17 Boot Program States..... 783

Figure 20.18 Bit-Rate-Adjustment Sequence ..... 784

Figure 20.19 Communication Protocol Format ..... 785

Figure 20.20 New Bit-Rate Selection Sequence..... 796

Figure 20.21 Programming Sequence..... 801

Figure 20.22 Erasure Sequence ..... 804

## Section 22 Clock Pulse Generator

Figure 22.1	Block Diagram of Clock Pulse Generator .....	813
Figure 22.2	Connection of Crystal Oscillator (Example) .....	816
Figure 22.3	Crystal Oscillator Equivalent Circuit .....	816
Figure 22.4	External Clock Input (Examples) .....	817
Figure 22.5	External Clock Input Timing.....	818
Figure 22.6	Note on Oscillator Board Design .....	820
Figure 22.7	Recommended External Circuitry for PLL Circuit .....	821

## Section 23 Power-Down Modes

Figure 23.1	Mode Transitions.....	825
Figure 23.2	Software Standby Mode Application Example .....	833
Figure 23.3	Hardware Standby Mode Timing .....	834
Figure 23.4	Hardware Standby Mode Timing when Power Is Supplied .....	835

## Section 25 Electrical Characteristics

Figure 25.1	Output Load Circuit.....	877
Figure 25.2	System Clock Timing.....	878
Figure 25.3	Oscillation Stabilization Timing (1).....	879
Figure 25.3	Oscillation Stabilization Timing (2).....	879
Figure 25.4	Reset Input Timing.....	880
Figure 25.5	Interrupt Input Timing.....	881
Figure 25.6	Basic Bus Timing: Two-State Access .....	886
Figure 25.7	Basic Bus Timing: Three-State Access .....	887
Figure 25.8	Basic Bus Timing: Three-State Access, One Wait .....	888
Figure 25.9	Basic Bus Timing: Two-State Access ( $\overline{CS}$ Assertion Period Extended) .....	889
Figure 25.10	Basic Bus Timing: Three-State Access ( $\overline{CS}$ Assertion Period Extended) .....	890
Figure 25.11	Burst ROM Access Timing: One-State Burst Access .....	891
Figure 25.12	Burst ROM Access Timing: Two-State Burst Access.....	892
Figure 25.13	DRAM Access Timing: Two-State Access .....	893
Figure 25.14	DRAM Access Timing: Two-State Access, One Wait.....	894
Figure 25.15	DRAM Access Timing: Two-State Burst Access .....	895
Figure 25.16	DRAM Access Timing: Three-State Access (RAST = 1) .....	896
Figure 25.17	DRAM Access Timing: Three-State Burst Access .....	897
Figure 25.18	CAS-Before-RAS Refresh Timing.....	898
Figure 25.19	CAS-Before-RAS Refresh Timing (with Wait Cycle Insertion) .....	898
Figure 25.20	Self-Refresh Timing (Return from Software Standby Mode: RAST = 0).....	899

Figure 25.23 External Bus Request Output Timing.....	900
Figure 25.24 DMAC Single Address Transfer Timing: Two-State Access.....	902
Figure 25.25 DMAC Single Address Transfer Timing: Three-State Access.....	903
Figure 25.26 DMAC $\overline{\text{TEND}}$ Output Timing .....	903
Figure 25.27 DMAC $\overline{\text{DREQ}}$ Input Timing .....	904
Figure 25.28 I/O Port Input/Output Timing.....	906
Figure 25.29 PPG Output Timing.....	906
Figure 25.30 TPU Input/Output Timing.....	906
Figure 25.31 TPU Clock Input Timing.....	907
Figure 25.32 8-Bit Timer Output Timing .....	907
Figure 25.33 8-Bit Timer Clock Input Timing .....	907
Figure 25.34 8-Bit Timer Reset Input Timing .....	907
Figure 25.35 WDT Output Timing.....	908
Figure 25.36 SCK Clock Input Timing .....	908
Figure 25.37 SCI Input/Output Timing: Synchronous Mode .....	908
Figure 25.38 A/D Converter External Trigger Input Timing.....	908
Figure 25.39 I <sup>2</sup> C Bus Interface Input/Output Timing (Option).....	909

## Appendix

Figure C.1 Package Dimensions (TFP-120).....	950
Figure C.2 Package Dimensions (FP-128B) .....	951
Figure D.1 Timing of Address Bus, $\overline{\text{RD}}$ , $\overline{\text{HWR}}$ , and $\overline{\text{LWR}}$ (8-bit bus, 3-state access, no wait).....	953

## Section 1 Overview

Table 1.1	Pin Arrangement in Each Operating Mode .....	8
Table 1.2	Pin Functions.....	14

## Section 2 CPU

Table 2.1	Instruction Classification.....	37
Table 2.2	Operation Notation .....	38
Table 2.3	Data Transfer Instructions .....	39
Table 2.4	Arithmetic Operations Instructions .....	40
Table 2.5	Logic Operations Instructions .....	42
Table 2.6	Shift Instructions .....	42
Table 2.7	Bit Manipulation Instructions.....	43
Table 2.8	Branch Instructions.....	45
Table 2.9	System Control Instructions .....	46
Table 2.10	Block Data Transfer Instructions.....	47
Table 2.11	Addressing Modes.....	49
Table 2.12	Absolute Address Access Ranges .....	51
Table 2.13	Effective Address Calculation.....	53

## Section 3 MCU Operating Modes

Table 3.1	MCU Operating Mode Selection.....	57
Table 3.2	Pin Functions in Each Operating Mode.....	62

## Section 4 Exception Handling

Table 4.1	Exception Types and Priority .....	79
Table 4.2	Exception Handling Vector Table .....	80
Table 4.3	Status of CCR and EXR after Trace Exception Handling .....	84
Table 4.4	Status of CCR and EXR after Trap Instruction Exception Handling .....	85

## Section 5 Interrupt Controller

Table 5.1	Pin Configuration .....	91
Table 5.2	Interrupt Sources, Vector Addresses, and Interrupt Priorities .....	103
Table 5.3	Interrupt Control Modes.....	108
Table 5.4	Interrupt Response Times.....	114
Table 5.5	Number of States in Interrupt Handling Routine Execution Statuses.....	115

Table 6.2	Bus Specifications for Each Area (Basic Bus Interface) .....	147
Table 6.3	Data Buses Used and Valid Strokes .....	152
Table 6.4	Relation between Settings of Bits RMTS2 to RMTS0 and DRAM Space.....	165
Table 6.5	Relation between Settings of Bits MXC2 to MXC0 and Address Multiplexing ....	166
Table 6.6	DRAM Interface Pins .....	167
Table 6.7	Idle Cycles in Mixed Accesses to Normal Space and DRAM .....	201
Table 6.8	Pin States in Idle Cycle .....	204
Table 6.9	Pin States in Bus Released State .....	207

## **Section 7 DMA Controller (DMAC)**

Table 7.1	Pin Configuration .....	215
Table 7.3	DMAC Activation Sources .....	241
Table 7.4	DMAC Transfer Modes .....	244
Table 7.5	Register Functions in Sequential Mode.....	246
Table 7.6	Register Functions in Idle Mode .....	249
Table 7.7	Register Functions in Repeat Mode .....	252
Table 7.8	Register Functions in Single Address Mode .....	256
Table 7.9	Register Functions in Normal Mode .....	259
Table 7.10	Register Functions in Block Transfer Mode.....	262
Table 7.11	DMAC Channel Priority Order .....	282
Table 7.12	Interrupt Sources and Priority Order .....	287

## **Section 8 Data Transfer Controller (DTC)**

Table 8.1	Relationship between Activation Sources and DTCER Clearing.....	301
Table 8.2	Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs.....	304
Table 8.3	Chain Transfer Conditions .....	308
Table 8.4	Register Function in Normal Mode.....	308
Table 8.5	Register Function in Repeat Mode.....	309
Table 8.6	Register Function in Block Transfer Mode.....	310
Table 8.7	DTC Execution Status.....	314
Table 8.8	Number of States Required for Each Execution Status.....	315

## **Section 9 I/O Ports**

Table 9.1	Port Functions .....	324
Table 9.2	MOS Input Pull-Up States (Port A).....	372
Table 9.3	MOS Input Pull-Up States (Port B).....	376
Table 9.4	MOS Input Pull-Up States (Port C).....	380
Table 9.5	MOS Input Pull-Up States (Port D).....	384

**Section 10 16-Bit Timer Pulse Unit (TPU)**

Table 10.1	TPU Functions.....	402
Table 10.2	Pin Configuration .....	405
Table 10.3	CCLR2 to CCLR0 (Channels 0 and 3).....	409
Table 10.4	CCLR2 to CCLR0 (Channels 1, 2, 4, and 5).....	409
Table 10.5	TPSC2 to TPSC0 (Channel 0).....	410
Table 10.6	TPSC2 to TPSC0 (Channel 1).....	410
Table 10.7	TPSC2 to TPSC0 (Channel 2).....	411
Table 10.8	TPSC2 to TPSC0 (Channel 3).....	411
Table 10.9	TPSC2 to TPSC0 (Channel 4).....	412
Table 10.10	TPSC2 to TPSC0 (Channel 5).....	412
Table 10.11	MD3 to MD0.....	414
Table 10.12	TIORH_0.....	416
Table 10.13	TIORL_0.....	417
Table 10.14	TIOR_1.....	418
Table 10.15	TIOR_2.....	419
Table 10.16	TIORH_3.....	420
Table 10.17	TIORL_3.....	421
Table 10.18	TIOR_4.....	422
Table 10.19	TIOR_5.....	423
Table 10.20	TIORH_0.....	424
Table 10.21	TIORL_0.....	425
Table 10.22	TIOR_1.....	426
Table 10.23	TIOR_2.....	427
Table 10.24	TIORH_3.....	428
Table 10.25	TIORL_3.....	429
Table 10.26	TIOR_4.....	430
Table 10.27	TIOR_5.....	431
Table 10.28	Register Combinations in Buffer Operation.....	448
Table 10.29	Cascaded Combinations .....	452
Table 10.30	PWM Output Registers and Output Pins.....	455
Table 10.31	Clock Input Pins in Phase Counting Mode.....	459
Table 10.32	Up/Down-Count Conditions in Phase Counting Mode 1 .....	461
Table 10.33	Up/Down-Count Conditions in Phase Counting Mode 2 .....	462
Table 10.34	Up/Down-Count Conditions in Phase Counting Mode 3 .....	463
Table 10.35	Up/Down-Count Conditions in Phase Counting Mode 4 .....	464
Table 10.36	TPU Interrupts.....	467

## Section 12 8-Bit Timers (TMR)

Table 12.1	Pin Configuration .....	509
Table 12.2	Clock Input to TCNT and Count Condition .....	512
Table 12.3	8-Bit Timer Interrupt Sources .....	521
Table 12.4	Timer Output Priorities .....	525
Table 12.5	Switching of Internal Clock and TCNT Operation .....	526

## Section 13 Watchdog Timer

Table 13.1	Pin Configuration .....	530
Table 13.2	WDT Interrupt Source.....	536

## Section 14 Serial Communication Interface (SCI, IrDA)

Table 14.1	Pin Configuration .....	544
Table 14.2	Relationships between N Setting in BRR and Bit Rate B .....	564
Table 14.3	BRR Settings for Various Bit Rates (Asynchronous Mode) .....	565
Table 14.4	Maximum Bit Rate for Each Frequency (Asynchronous Mode).....	567
Table 14.5	Maximum Bit Rate with External Clock Input (Asynchronous Mode).....	568
Table 14.6	BRR Settings for Various Bit Rates (Clocked Synchronous Mode) .....	569
Table 14.7	Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode).....	570
Table 14.8	Examples of Bit Rate for Various BRR Settings (Smart Card Interface Mode) (when $n = 0$ and $S = 372$ ) .....	571
Table 14.9	Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (when $S = 372$ ) .....	572
Table 14.10	Serial Transfer Formats (Asynchronous Mode) .....	577
Table 14.11	SSR Status Flags and Receive Data Handling.....	585
Table 14.12	Settings of Bits IrCKS2 to IrCKS0 .....	618
Table 14.13	SCI Interrupt Sources .....	620
Table 14.14	Interrupt Sources .....	621

## Section 15 I<sup>2</sup>C Bus Interface2 (IIC2) (Option)

Table 15.1	Pin Configuration .....	631
Table 15.2	Transfer Rate.....	634
Table 15.3	Interrupt Requests .....	658
Table 15.4	Time for monitoring SCL.....	659

## Section 16 A/D Converter

Table 16.1	Pin Configuration .....	663
------------	-------------------------	-----



Table 16.4	A/D Conversion Time (Scan Mode).....	672
Table 16.5	A/D Converter Interrupt Source .....	673
Table 16.6	Analog Pin Specifications .....	679
<b>Section 17 D/A Converter</b>		
Table 17.1	Pin Configuration .....	683
Table 17.2	Control of D/A Conversion .....	685
<b>Section 19 Flash Memory (0.35-<math>\mu</math>m F-ZTAT Version)</b>		
Table 19.1	Differences between Boot Mode and User Program Mode.....	692
Table 19.2	Pin Configuration .....	697
Table 19.3	Erase Blocks .....	701
Table 19.4	Setting On-Board Programming Mode.....	702
Table 19.5	Boot Mode Operation .....	704
Table 19.6	System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible .....	705
Table 19.7	Flash Memory Operating States .....	711
<b>Section 20 Flash Memory (0.18-<math>\mu</math>m F-ZTAT Version)</b>		
Table 20.1	Comparison of Programming Modes .....	721
Table 20.2	Pin Configuration .....	726
Table 20.3	Register/Parameter and Target Mode .....	727
Table 20.4	Parameters and Target Modes .....	736
Table 20.5	Setting On-Board Programming Mode.....	748
Table 20.6	System Clock Frequency for Automatic-Bit-Rate Adjustment by This LSI .....	750
Table 20.7	Executable MAT .....	769
Table 20.8 (1)	Useable Area for Programming in User Program Mode .....	770
Table 20.8 (2)	Useable Area for Erasure in User Program Mode .....	772
Table 20.8 (3)	Useable Area for Programming in User Boot Mode .....	774
Table 20.8 (4)	Useable Area for Erasure in User Boot Mode .....	776
Table 20.9	Hardware Protection.....	778
Table 20.10	Software Protection .....	779
Table 20.11	Inquiry and Selection Commands .....	786
Table 20.12	Programming/Erasing Command .....	799
Table 20.13	Status Code.....	808
Table 20.14	Error Code .....	809
Table 20.15	User Branch Processing Start Intervals .....	810

Table 22.2	Crystal Oscillator Characteristics .....	817
Table 22.3	External Clock Input Conditions .....	818

### Section 23 Power-Down Modes

Table 23.1	Operating Modes and Internal States of the LSI .....	824
Table 23.2	Oscillation Stabilization Time Settings .....	832
Table 23.3	$\phi$ Pin State in Each Processing State .....	836

### Section 25 Electrical Characteristics

Table 25.1	Absolute Maximum Ratings .....	873
Table 25.2	DC Characteristics (1) .....	874
Table 25.3	DC Characteristics (2) .....	875
Table 25.4	Permissible Output Currents .....	876
Table 25.5	Clock Timing .....	878
Table 25.6	Control Signal Timing .....	880
Table 25.7	Bus Timing (1) .....	882
Table 25.8	Bus Timing (2) .....	884
Table 25.9	DMAC Timing .....	901
Table 25.10	Timing of On-Chip Peripheral Modules .....	904
Table 25.11	A/D Conversion Characteristics .....	910
Table 25.12	D/A Conversion Characteristics .....	910
Table 25.13	Absolute Maximum Ratings .....	911
Table 25.14	DC Characteristics (1) .....	912
Table 25.15	DC Characteristics (2) .....	913
Table 25.16	Permissible Output Currents .....	914
Table 25.17	Clock Timing .....	915
Table 25.18	Control Signal Timing .....	916
Table 25.19	Bus Timing (1) .....	917
Table 25.20	Bus Timing (2) .....	919
Table 25.21	DMAC Timing .....	920
Table 25.22	Timing of On-Chip Peripheral Modules .....	921
Table 25.23	A/D Conversion Characteristics .....	923
Table 25.24	D/A Conversion Characteristics .....	923
Table 25.25	Flash Memory Characteristics .....	924
Table 25.26	Absolute Maximum Ratings .....	926
Table 25.27	DC Characteristics .....	927
Table 25.28	DC Characteristics .....	928
Table 25.29	Permissible Output Currents .....	929

Table 25.32	Bus Timing (1) .....	932
Table 25.33	Bus Timing (2) .....	934
Table 25.34	DMAC Timing .....	935
Table 25.35	Timing of On-Chip Peripheral Modules.....	936
Table 25.36	A/D Conversion Characteristics .....	938
Table 25.37	D/A Conversion Characteristics .....	938
Table 25.38	Flash Memory Characteristics (0.18- $\mu$ m F-ZTAT Version) .....	939

## **Appendix**

Table D.1	Execution State of Instructions.....	954
-----------	--------------------------------------	-----



## 1.1 Features

- High-speed H8S/2000 central processing unit with an internal 16-bit architecture  
Upward-compatible with H8/300 and H8/300H CPUs on an object level  
Sixteen 16-bit general registers  
65 basic instructions
- Various peripheral functions  
DMA controller (DMAC)  
Data transfer controller (DTC)  
16-bit timer-pulse unit (TPU)  
Programmable pulse generator (PPG)  
8-bit timer (TMR)  
Watchdog timer (WDT)  
Asynchronous or clocked synchronous serial communication interface (SCI)  
I<sup>2</sup>C bus interface 2 (IIC2)  
10-bit A/D converter  
8-bit D/A converter  
Clock pulse generator

ROM Type	Model	ROM	RAM	Remarks
Flash memory version	HD64F2368F	512 kbytes	32 kbytes	H8S/2368 0.18 μm F-ZTAT Group
	HD64F2367F	384 kbytes	24 kbytes	
	HD64F2364	384 kbytes	32 kbytes	H8S/2368 0.18 μm F-ZTAT Group
	HD64F2362F	256 kbytes	32 kbytes	H8S/2368 0.18 μm F-ZTAT Group
	HD64F2361	256 kbytes	24 kbytes	H8S/2368 0.18 μm F-ZTAT Group
	HD64F2360	256 kbytes	16 kbytes	H8S/2368 0.18 μm F-ZTAT Group
Masked ROM version	HD6432365	256 kbytes	16 kbytes	
ROMless version	HD6412363	—	16 kbytes	

- General I/O ports  
I/O pins: 83  
Input-only pins: 11
- Supports various power-down states
- Compact package

Package	Code	Body Size	Pin Pitch
TFP-120	TFP-120 (TFP-120V* <sup>1</sup> )	14.0 × 14.0 mm	0.4 mm
QFP-128* <sup>2</sup>	FP-128B (FP-128BV* <sup>1</sup> )	14.0 × 20.0 mm	0.5 mm

Notes: 1. Pb free version  
2. Not supported by the H8S/2368 0.18 μm F-ZTAT Group.

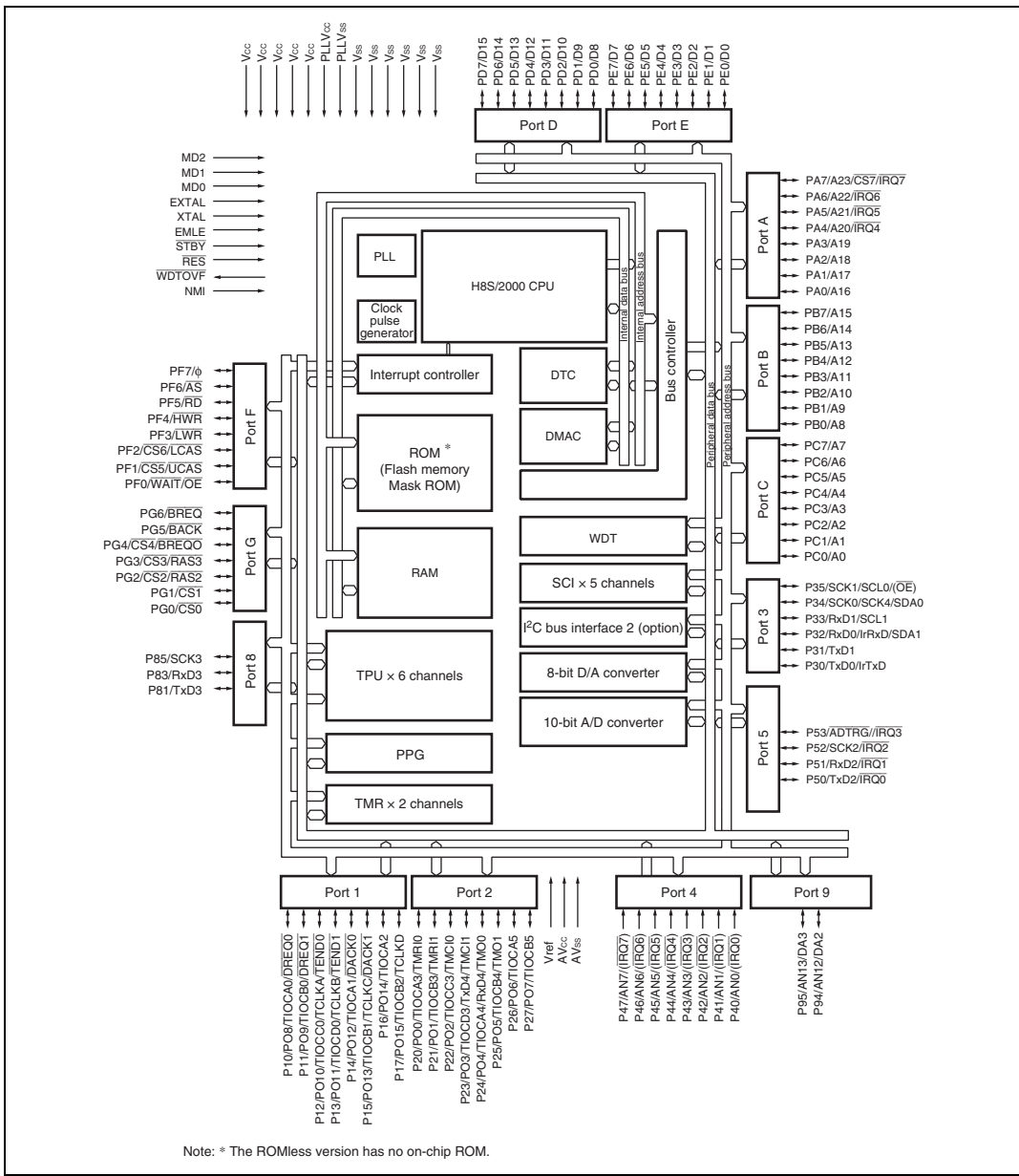


Figure 1.1 Internal Block Diagram of H8S/2367F, H8S/2365, and H8S/2363

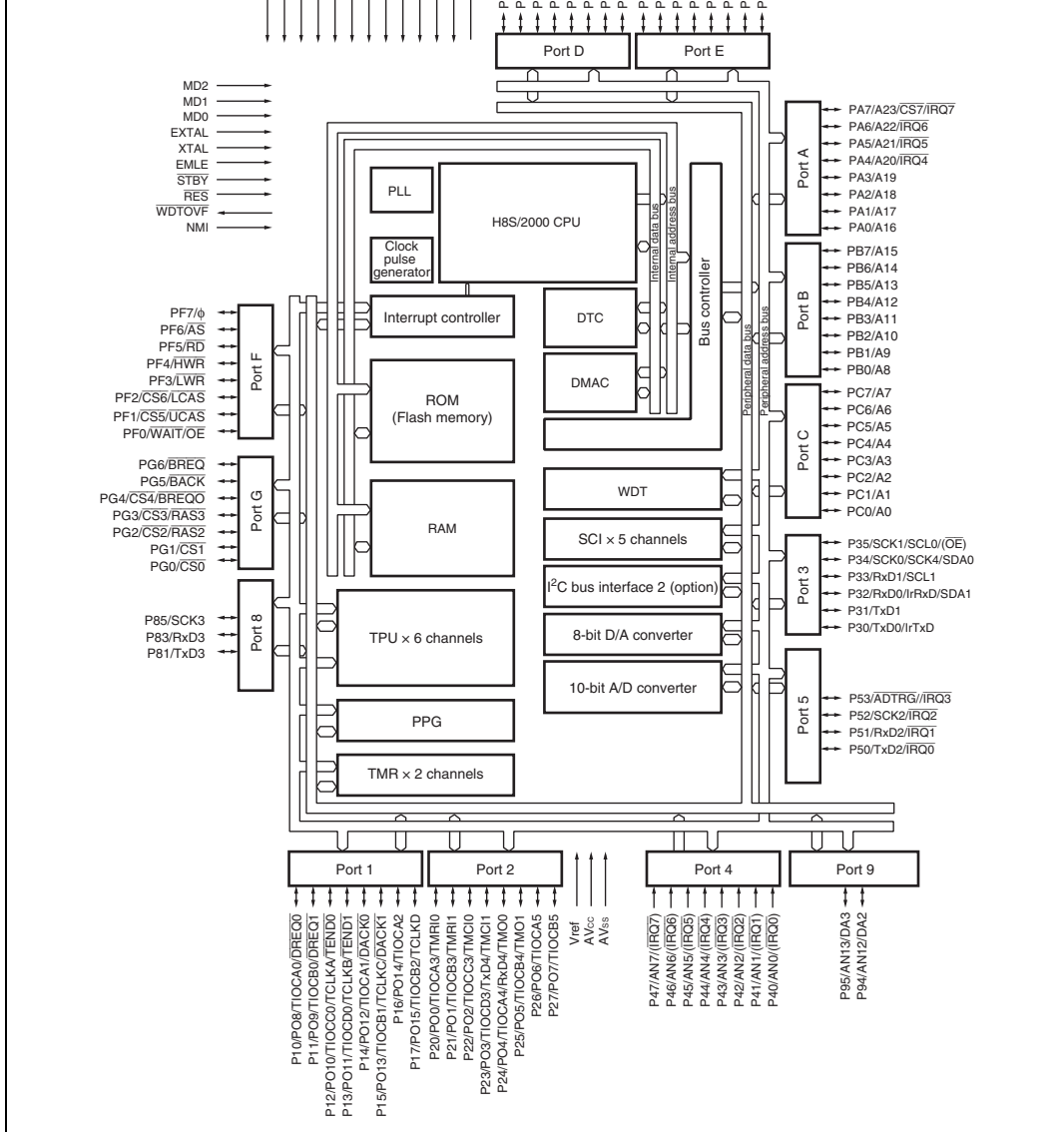
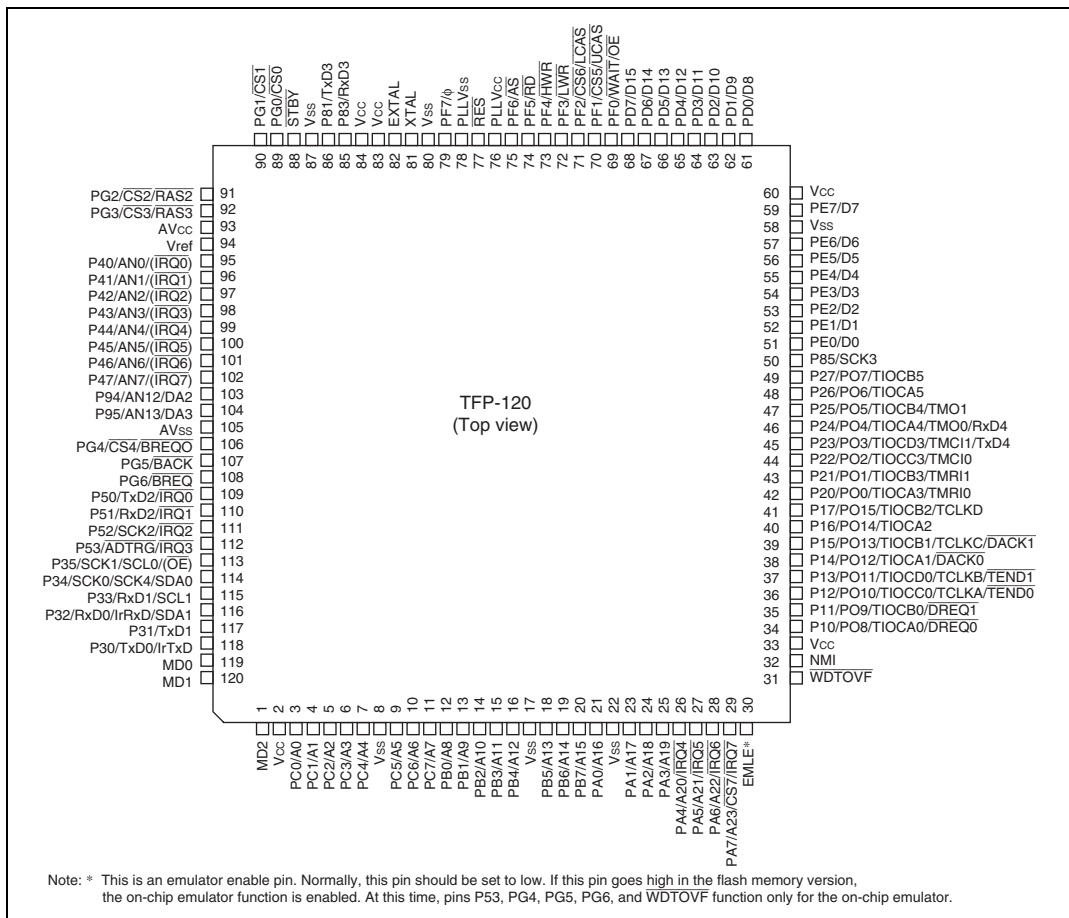


Figure 1.2 Internal Block Diagram of H8S/2368 0.18 μm F-ZTAT Group

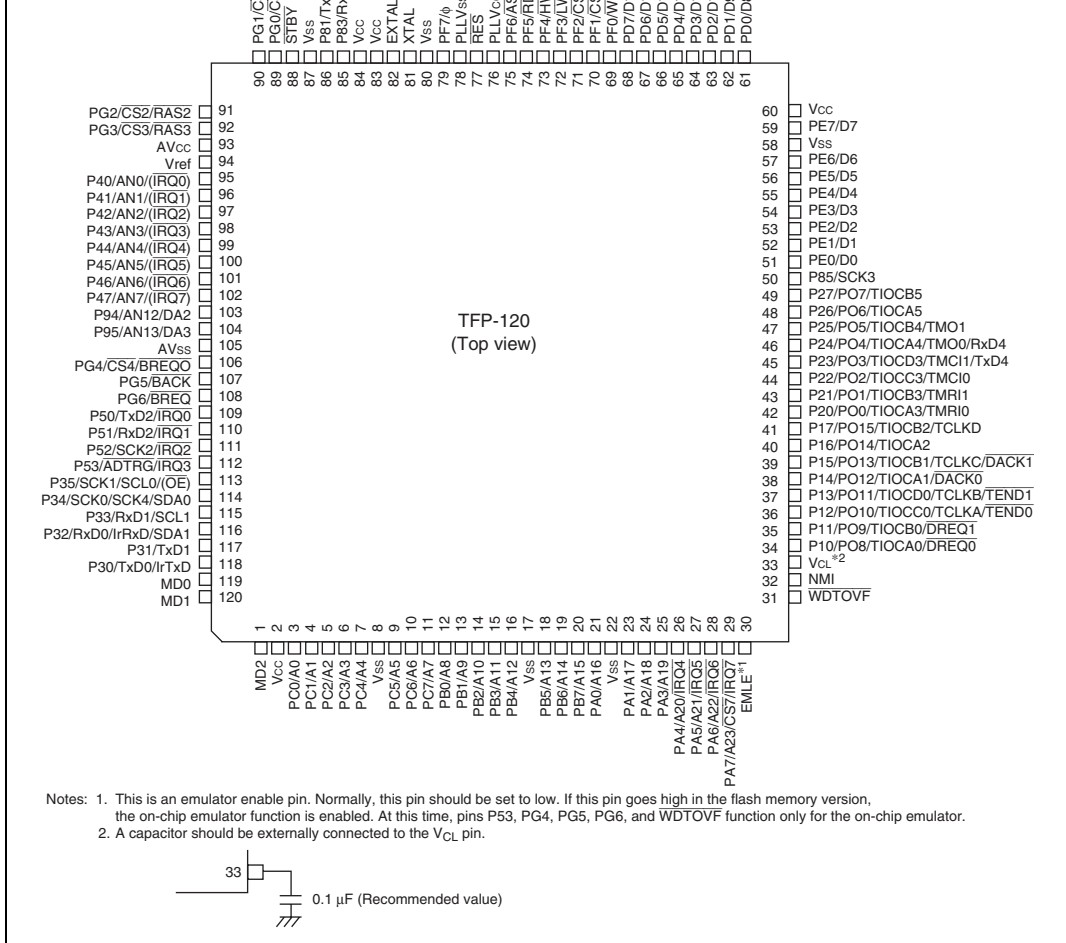


### 1.3.1 Pin Arrangement

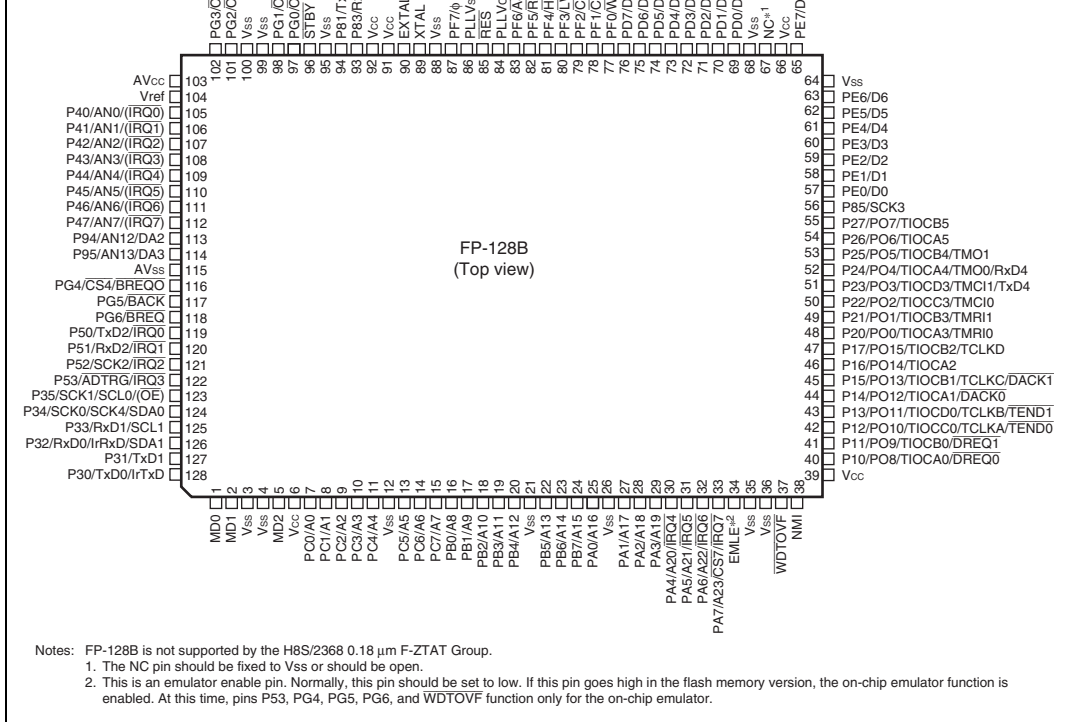
Figures 1.3 to 1.5 show the pin arrangements of this LSI.



**Figure 1.3 Pin Arrangement of H8S/2367F, H8S/2365, and H8S/2363**



**Figure 1.4 Pin Arrangement of H8S/2368 0.18  $\mu$ m F-ZTAT Group**



**Figure 1.5 Pin Arrangement of H8S/2367F, H8S/2365, and H8S/2363**

**Table 1.1 Pin Arrangement in Each Operating Mode**

Pin No.		Pin Name						Flash Memory Programmer Mode
TFP-120	QFP-128*1	Mode 1	Mode 2	Mode 4	Mode 7			
					EXPE = 1	EXPE = 0		
1	5	MD2	MD2	MD2	MD2	MD2	Vss	
2	6	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc	
3	7	A0	A0	PC0/A0	PC0/A0	PC0	A0	
4	8	A1	A1	PC1/A1	PC1/A1	PC1	A1	
5	9	A2	A2	PC2/A2	PC2/A2	PC2	A2	
6	10	A3	A3	PC3/A3	PC3/A3	PC3	A3	
7	11	A4	A4	PC4/A4	PC4/A4	PC4	A4	
8	12	Vss	Vss	Vss	Vss	Vss	Vss	
9	13	A5	A5	PC5/A5	PC5/A5	PC5	A5	
10	14	A6	A6	PC6/A6	PC6/A6	PC6	A6	
11	15	A7	A7	PC7/A7	PC7/A7	PC7	A7	
12	16	A8	A8	PB0/A8	PB0/A8	PB0	A8	
13	17	A9	A9	PB1/A9	PB1/A9	PB1	A9	
14	18	A10	A10	PB2/A10	PB2/A10	PB2	A10	
15	19	A11	A11	PB3/A11	PB3/A11	PB3	A11	
16	20	A12	A12	PB4/A12	PB4/A12	PB4	A12	
17	21	Vss	Vss	Vss	Vss	Vss	Vss	
18	22	A13	A13	PB5/A13	PB5/A13	PB5	A13	
19	23	A14	A14	PB6/A14	PB6/A14	PB6	A14	
20	24	A15	A15	PB7/A15	PB7/A15	PB7	A15	
21	25	A16	A16	PA0/A16	PA0/A16	PA0	A16	
22	26	Vss	Vss	Vss	Vss	Vss	Vss	
23	27	A17	A17	PA1/A17	PA1/A17	PA1	A17	
24	28	A18	A18	PA2/A18	PA2/A18	PA2	A18	
25	29	A19	A19	PA3/A19	PA3/A19	PA3	NC	
26	30	A20/IRQ4	A20/IRQ4	PA4/A20/IRQ4	PA4/A20/IRQ4	PA4/IRQ4	NC	
27	31	PA5/A21/IRQ5	PA5/A21/IRQ5	PA5/A21/IRQ5	PA5/IRQ5	PA5/IRQ5	NC	
28	32	PA6/A22/IRQ6	PA6/A22/IRQ6	PA6/A22/IRQ6	PA6/IRQ6	PA6/IRQ6	NC	

TFP-120	QFP-128*1	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	Programmer Mode
29	33	PA7/A23/CS7/ IRQ7	PA7/A23/CS7/ IRQ7	PA7/A23/CS7/ IRQ7	PA7/CS7/ IRQ7	PA7/IRQ7	NC
30	34	EMLE	EMLE	EMLE	EMLE	EMLE	EMLE
—	35	Vss	Vss	Vss	Vss	Vss	Vss
—	36	Vss	Vss	Vss	Vss	Vss	Vss
31	37	WDTOVF	WDTOVF	WDTOVF	WDTOVF	WDTOVF	NC
32	38	NMI	NMI	NMI	NMI	NMI	Vcc
33	39	V <sub>CC</sub> (V <sub>CL</sub> <sup>*2</sup> )	V <sub>CC</sub> (V <sub>CL</sub> <sup>*2</sup> )	V <sub>CC</sub> (V <sub>CL</sub> <sup>*2</sup> )	V <sub>CC</sub> (V <sub>CL</sub> <sup>*2</sup> )	V <sub>CC</sub> (V <sub>CL</sub> <sup>*2</sup> )	V <sub>CC</sub> (V <sub>CL</sub> <sup>*2</sup> )
34	40	P10/PO8/ TIOCA0/ DREQ0	P10/PO8/ TIOCA0/ DREQ0	P10/PO8/ TIOCA0/ DREQ0	P10/PO8/ TIOCA0/ DREQ0	P10/PO8/ TIOCA0/ DREQ0	NC
35	41	P11/PO9/ TIOCB0/ DREQ1	P11/PO9/ TIOCB0/ DREQ1	P11/PO9/ TIOCB0/ DREQ1	P11/PO9/ TIOCB0/ DREQ1	P11/PO9/ TIOCB0/ DREQ1	NC
36	42	P12/PO10/ TIOCC0/TCLKA/ TEND0	P12/PO10/ TIOCC0/TCLKA/ TEND0	P12/PO10/ TIOCC0/TCLKA/ TEND0	P12/PO10/ TIOCC0/TCLKA/ TEND0	P12/PO10/ TIOCC0/TCLKA/ TEND0	NC
37	43	P13/PO11/ TIOCD0/TCLKB/ TEND1	P13/PO11/ TIOCD0/TCLKB/ TEND1	P13/PO11/ TIOCD0/TCLKB/ TEND1	P13/PO11/ TIOCD0/TCLKB/ TEND1	P13/PO11/ TIOCD0/TCLKB/ TEND1	NC
38	44	P14/PO12/ TIOCA1/ DACK0	P14/PO12/ TIOCA1/ DACK0	P14/PO12/ TIOCA1/ DACK0	P14/PO12/ TIOCA1/ DACK0	P14/PO12/ TIOCA1/ DACK0	NC
39	45	P15/PO13/ TIOCB1/TCLKC/ DACK1	P15/PO13/ TIOCB1/TCLKC/ DACK1	P15/PO13/ TIOCB1/TCLKC/ DACK1	P15/PO13/ TIOCB1/TCLKC/ DACK1	P15/PO13/ TIOCB1/TCLKC/ DACK1	NC
40	46	P16/PO14/ TIOCA2	P16/PO14/ TIOCA2	P16/PO14/ TIOCA2	P16/PO14/ TIOCA2	P16/PO14/ TIOCA2	NC
41	47	P17/PO15/ TIOCB2/TCLKD	P17/PO15/ TIOCB2/TCLKD	P17/PO15/ TIOCB2/TCLKD	P17/PO15/ TIOCB2/TCLKD	P17/PO15/ TIOCB2/TCLKD	NC
42	48	P20/PO0/ TIOCA3/TMRI0	P20/PO0/ TIOCA3/(TMRI0)	P20/PO0/ TIOCA3/(TMRI0)	P20/PO0/ TIOCA3/(TMRI0)	P20/PO0/ TIOCA3/(TMRI0)	NC
43	49	P21/PO1/ TIOCB3/TMRI1	P21/PO1/ TIOCB3/TMRI1	P21/PO1/ TIOCB3/TMRI1	P21/PO1/ TIOCB3/TMRI1	P21/PO1/ TIOCB3/TMRI1	NC
44	50	P22/PO2/ TIOCC3/TMCI0	P22/PO2/ TIOCC3/TMCI0	P22/PO2/ TIOCC3/TMCI0	P22/PO2/ TIOCC3/TMCI0	P22/PO2/ TIOCC3/TMCI0	OE

TFP-120	QFP-128*1	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	Programmer Mode
45	51	P23/PO3/ TIOCD3/TMC11/ TxD4	P23/PO3/ TIOCD3/TMC11/ TxD4	P23/PO3/ TIOCD3/TMC11/ TxD4	P23/PO3/ TIOCD3/TMC11/ TxD4	P23/PO3/ TIOCD3/TMC11/ TxD4	$\overline{CE}$
46	52	P24/PO4/ TIOCA4/TMO0/ RxD4	P24/PO4/ TIOCA4/TMO0/ RxD4	P24/PO4/ TIOCA4/TMO0/ RxD4	P24/PO4/ TIOCA4/TMO0/ RxD4	P24/PO4/ TIOCA4/TMO0/ RxD4	$\overline{WE}$
47	53	P25/PO5/ TIOCB4/TMO1	P25/PO5/ TIOCB4/TMO1	P25/PO5/ TIOCB4/TMO1	P25/PO5/ TIOCB4/TMO1	P25/PO5/ TIOCB4/TMO1	Vss
48	54	P26/PO6/ TIOCA5	P26/PO6/ TIOCA5	P26/PO6/ TIOCA5	P26/PO6/ TIOCA5	P26/PO6/ TIOCA5	NC
49	55	P27/PO7/ TIOCB5	P27/PO7/ TIOCB5	P27/PO7/ TIOCB5	P27/PO7/ TIOCB5	P27/PO7/ TIOCB5	NC
50	56	P85/SCK3	P85/SCK3	P85/SCK3	P85/SCK3	P85/SCK3	NC
51	57	D0	PE0/D0	PE0/D0	PE0/D0	PE0	NC
52	58	D1	PE1/D1	PE1/D1	PE1/D1	PE1	NC
53	59	D2	PE2/D2	PE2/D2	PE2/D2	PE2	NC
54	60	D3	PE3/D3	PE3/D3	PE3/D3	PE3	NC
55	61	D4	PE4/D4	PE4/D4	PE4/D4	PE4	NC
56	62	D5	PE5/D5	PE5/D5	PE5/D5	PE5	NC
57	63	D6	PE6/D6	PE6/D6	PE6/D6	PE6	NC
58	64	Vss	Vss	Vss	Vss	Vss	Vss
59	65	D7	PE7/D7	PE7/D7	PE7/D7	PE7	NC
60	66	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
—	67	NC	NC	NC	NC	NC	NC
—	68	Vss	Vss	Vss	Vss	Vss	Vss
61	69	D8	D8	D8	D8	PD0	I/O0
62	70	D9	D9	D9	D9	PD1	I/O1
63	71	D10	D10	D10	D10	PD2	I/O2
64	72	D11	D11	D11	D11	PD3	I/O3
65	73	D12	D12	D12	D12	PD4	I/O4
66	74	D13	D13	D13	D13	PD5	I/O5
67	75	D14	D14	D14	D14	PD6	I/O6

TFP-120	QFP-128*1	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	Programmer Mode
68	76	D15	D15	D15	D15	PD7	I/O7
69	77	PF0/WAIT/OE	PF0/WAIT/OE	PF0/WAIT/OE	PF0/WAIT/OE	PF0	NC
70	78	PF1/CS5/ UCAS	PF1/CS5/ UCAS	PF1/CS5/ UCAS	PF1/CS5/ UCAS	PF1	NC
71	79	PF2/CS6/ LCAS	PF2/CS6/ LCAS	PF2/CS6/ LCAS	PF2/CS6/ LCAS	PF2	NC
72	80	PF3/LWR	PF3/LWR	PF3/LWR	PF3/LWR	PF3	NC
73	81	HWR	HWR	HWR	HWR	PF4	NC
74	82	RD	RD	RD	RD	PF5	NC
75	83	PF6/AS	PF6/AS	PF6/AS	PF6/AS	PF6	NC
76	84	PLLVcc	PLLVcc	PLLVcc	PLLVcc	PLLVcc	Vcc
77	85	RES	RES	RES	RES	RES	RES
78	86	PLLVss	PLLVss	PLLVss	PLLVss	PLLVss	Vss
79	87	PF7/φ	PF7/φ	PF7/φ	PF7/φ	PF7/φ	NC
80	88	Vss	Vss	Vss	Vss	Vss	Vss
81	89	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL
82	90	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL
83	91	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
84	92	Vcc	Vcc	Vcc	Vcc	Vcc	Vcc
85	93	P83/RxD3	P83/RxD3	P83/RxD3	P83/RxD3	P83/RxD3	NC
86	94	P81/TxD3	P81/TxD3	P81/TxD3	P81/TxD3	P81/TxD3	NC
87	95	Vss	Vss	Vss	Vss	Vss	Vss
88	96	STBY	STBY	STBY	STBY	STBY	Vcc
89	97	PG0/CS0	PG0/CS0	PG0/CS0	PG0/CS0	PG0	NC
90	98	PG1/CS1	PG1/CS1	PG1/CS1	PG1/CS1	PG1	NC
—	99	Vss	Vss	Vss	Vss	Vss	Vss
—	100	Vss	Vss	Vss	Vss	Vss	Vss
91	101	PG2/CS2/ RAS2	PG2/CS2/ RAS2	PG2/CS2/ RAS2	PG2/CS2/ RAS2	PG2	NC
92	102	PG3/CS3/ RAS3	PG3/CS3/ RAS3	PG3/CS3/ RAS3	PG3/CS3/ RAS3	PG3	NC
93	103	AVcc	AVcc	AVcc	AVcc	AVcc	Vcc

TFP-120	QFP-128*1	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	Programmer Mode
94	104	Vref	Vref	Vref	Vref	Vref	NC
95	105	P40/AN0/( $\overline{\text{IRQ0}}$ )	P40/AN0/( $\overline{\text{IRQ0}}$ )	P40/AN0/( $\overline{\text{IRQ0}}$ )	P40/AN0/( $\overline{\text{IRQ0}}$ )	P40/AN0/( $\overline{\text{IRQ0}}$ )	NC
96	106	P41/AN1/( $\overline{\text{IRQ1}}$ )	P41/AN1/( $\overline{\text{IRQ1}}$ )	P41/AN1/( $\overline{\text{IRQ1}}$ )	P41/AN1/( $\overline{\text{IRQ1}}$ )	P41/AN1/( $\overline{\text{IRQ1}}$ )	NC
97	107	P42/AN2/( $\overline{\text{IRQ2}}$ )	P42/AN2/( $\overline{\text{IRQ2}}$ )	P42/AN2/( $\overline{\text{IRQ2}}$ )	P42/AN2/( $\overline{\text{IRQ2}}$ )	P42/AN2/( $\overline{\text{IRQ2}}$ )	NC
98	108	P43/AN3/( $\overline{\text{IRQ3}}$ )	P43/AN3/( $\overline{\text{IRQ3}}$ )	P43/AN3/( $\overline{\text{IRQ3}}$ )	P43/AN3/( $\overline{\text{IRQ3}}$ )	P43/AN3/( $\overline{\text{IRQ3}}$ )	NC
99	109	P44/AN4/( $\overline{\text{IRQ4}}$ )	P44/AN4/( $\overline{\text{IRQ4}}$ )	P44/AN4/( $\overline{\text{IRQ4}}$ )	P44/AN4/( $\overline{\text{IRQ4}}$ )	P44/AN4/( $\overline{\text{IRQ4}}$ )	NC
100	110	P45/AN5/( $\overline{\text{IRQ5}}$ )	P45/AN5/( $\overline{\text{IRQ5}}$ )	P45/AN5/( $\overline{\text{IRQ5}}$ )	P45/AN5/( $\overline{\text{IRQ5}}$ )	P45/AN5/( $\overline{\text{IRQ5}}$ )	NC
101	111	P46/AN6/( $\overline{\text{IRQ6}}$ )	P46/AN6/( $\overline{\text{IRQ6}}$ )	P46/AN6/( $\overline{\text{IRQ6}}$ )	P46/AN6/( $\overline{\text{IRQ6}}$ )	P46/AN6/( $\overline{\text{IRQ6}}$ )	NC
102	112	P47/AN7/( $\overline{\text{IRQ7}}$ )	P47/AN7/( $\overline{\text{IRQ7}}$ )	P47/AN7/( $\overline{\text{IRQ7}}$ )	P47/AN7/( $\overline{\text{IRQ7}}$ )	P47/AN7/( $\overline{\text{IRQ7}}$ )	NC
103	113	P94/AN12/DA2	P94/AN12/DA2	P94/AN12/DA2	P94/AN12/DA2	P94/AN12/DA2	NC
104	114	P95/AN13/DA3	P95/AN13/DA3	P95/AN13/DA3	P95/AN13/DA3	P95/AN13/DA3	NC
105	115	AVss	AVss	AVss	AVss	AVss	Vss
106	116	PG4/CS4/ BREQO	PG4/CS4/ BREQO	PG4/CS4/ BREQO	PG4/CS4/ BREQO	PG4	NC
107	117	PG5/BACK	PG5/BACK	PG5/BACK	PG5/BACK	PG5	NC
108	118	PG6/BREQ	PG6/BREQ	PG6/BREQ	PG6/BREQ	PG6	NC
109	119	P50/TxD2/ $\overline{\text{IRQ0}}$	P50/TxD2/ $\overline{\text{IRQ0}}$	P50/TxD2/ $\overline{\text{IRQ0}}$	P50/TxD2/ $\overline{\text{IRQ0}}$	P50/TxD2/ $\overline{\text{IRQ0}}$	Vss
110	120	P51/RxD2/ $\overline{\text{IRQ1}}$	P51/RxD2/ $\overline{\text{IRQ1}}$	P51/RxD2/ $\overline{\text{IRQ1}}$	P51/RxD2/ $\overline{\text{IRQ1}}$	P51/RxD2/ $\overline{\text{IRQ1}}$	Vss
111	121	P52/SCK2/ $\overline{\text{IRQ2}}$	P52/SCK2/ $\overline{\text{IRQ2}}$	P52/SCK2/ $\overline{\text{IRQ2}}$	P52/SCK2/ $\overline{\text{IRQ2}}$	P52/SCK2/ $\overline{\text{IRQ2}}$	Vcc
112	122	P53/ADTRG/ $\overline{\text{IRQ3}}$	P53/ADTRG/ $\overline{\text{IRQ3}}$	P53/ADTRG/ $\overline{\text{IRQ3}}$	P53/ADTRG/ $\overline{\text{IRQ3}}$	P53/ADTRG/ $\overline{\text{IRQ3}}$	NC
113	123	P35/SCK1/ SCL0/( $\overline{\text{OE}}$ )	P35/SCK1/ SCL0/( $\overline{\text{OE}}$ )	P35/SCK1/ SCL0/( $\overline{\text{OE}}$ )	P35/SCK1/ SCL0/( $\overline{\text{OE}}$ )	P35/SCK1/SCL0	NC
114	124	P34/SCK0/ SCK4/SDA0	P34/SCK0/ SCK4/SDA0	P34/SCK0/ SCK4/SDA0	P34/SCK0/ SCK4/SDA0	P34/SCK0/ SCK4/SDA0	NC
115	125	P33/RxD1/SCL1	P33/RxD1/SCL1	P33/RxD1/SCL1	P33/RxD1/SCL1	P33/RxD1/SCL1	NC
116	126	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	P32/RxD0/ IrRxD/SDA1	Vcc
117	127	P31/TxD1	P31/TxD1	P31/TxD1	P31/TxD1	P31/TxD1	NC
118	128	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	P30/TxD0/IrTxD	NC
119	1	MD0	MD0	MD0	MD0	MD0	Vss
120	2	MD1	MD1	MD1	MD1	MD1	Vss



TFP-120	QFP-128 <sup>*1</sup>	Mode 1	Mode 2	Mode 4	EXPE = 1	EXPE = 0	Programmer Mode
—	3	Vss	Vss	Vss	Vss	Vss	Vss
—	4	Vss	Vss	Vss	Vss	Vss	Vss

- Notes: 1. Not supported by the H8S/2368 0.18 μm F-ZTAT Group.  
2. Used as the V<sub>CL</sub> pin in the H8S/2368 0.18 μm F-ZTAT Group.

**Table 1.2 Pin Functions**

Type	Symbol	Pin No.		I/O	Function
		TFP-120	QFP-128*1		
Power Supply	V <sub>cc</sub>	2, 33*2, 60, 83, 84	6,39,66, 91,92	Input	Power supply pins. V <sub>cc</sub> pins should be connected to the system power supply.
	V <sub>ss</sub>	8, 17, 22, 58, 80, 87	3,4,12,21, 26,35,36, 64,68,88, 95,99,100	Input	Ground pins. V <sub>ss</sub> pins should be connected to the system power supply (0 V).
	PLL <sub>V<sub>cc</sub></sub>	76	84	Input	Power supply pin for the on-chip PLL oscillator.
	PLL <sub>V<sub>ss</sub></sub>	78	86	Input	Ground pin for the on-chip PLL oscillator.
	VCL	33*3		Input	Connect to V <sub>ss</sub> via a 0.1 μF (recommended value) capacitor (placed close to the pin).
Clock	XTAL	81	89	Input	For connection to a crystal oscillator. See section 22, Clock Pulse Generator for typical connection diagrams for a crystal oscillator and external clock input.
	EXTAL	82	90	Input	For connection to a crystal oscillator. The EXTAL pin can also input an external clock. See section 22, Clock Pulse Generator for typical connection diagrams for a crystal oscillator and external clock input.
	0	79	87	Output	Supplies the system clock to external devices.
Operating mode control	MD2	1,	5,	Input	These pins set the operating mode. These pins should not be changed while the MCU is operating.
	MD1	120,	2,		
	MD0	119	1		

System control	RES	77	85	Input	Reset pin. When this pin is driven low, the chip is reset.
	STBY	88	96	Input	When this pin is driven low, a transition is made to hardware standby mode.
	EMLE	30	34	Input	Enables emulator. This pin should be connected to the power supply (0 V).
Address bus	A23 to A0	29 to 23, 21 to 18, 16 to 9, 7 to 3	33 to 27, 25 to 22, 20 to 13, 11 to 7	Output	Address output pins.
Data bus	D15 to D0	68 to 61, 59, 57 to 51	76 to 69, 65, 63 to 57	Input/ output	These pins constitute a bidirectional data bus.
Bus control	$\overline{CS7}$ to $\overline{CS0}$	29,71,70, 106, 92 to 89	33,79,78, 116,102, 101,98,97	Output	Signals that select division areas 7 to 0 in the external address space.
	AS	75	83	Output	When this pin is low, it indicates that address output on the address bus is valid.
	$\overline{RD}$	74	82	Output	When this pin is low, it indicates that the external address space is being read.
	HWR	73	81	Output	Strobe signal indicating that external address space is to be written, and the upper half (D15 to D8) of the data bus is enabled.  Write enable signal for accessing the DRAM space.
	$\overline{LWR}$	72	80	Output	Strobe signal indicating that external address space is to be written, and the lower half (D7 to D0) of the data bus is enabled.
	$\overline{BREQ}$	108	118	Input	The external bus master requests the bus to this LSI.
	$\overline{BREQO}$	106	116	Input	External bus request signal when the internal bus master accesses the external space in external bus release state.

Bus control	$\overline{\text{BACK}}$	107	117	Output	Indicates the bus is released to the external bus master.
	$\overline{\text{UCAS}}$	70	78	Output	Upper column address strobe signal for accessing the 16-bit DRAM space. Column address strobe signal for accessing the 8-bit DRAM space.
	$\overline{\text{LCAS}}$	71	79	Output	Lower column address strobe signal for accessing the 16-bit DRAM space.
	$\overline{\text{RAS2}}$	91	101	Output	Row address strobe signal for the DRAM interface.
	$\overline{\text{RAS3}}$	92	102		
	$\overline{\text{WAIT}}$	69	77	Input	Requests insertion of a wait state in the bus cycle when accessing external 3-state address space.
	$\overline{\text{OE}}$ ( $\overline{\text{OE}}$ )	69, 113	77, 123	Output	Output enable signal for accessing the DRAM space.  The output pins of $\overline{\text{OE}}$ and ( $\overline{\text{OE}}$ ) are selected by the port function control register 2 (PFCR2) of port 3.
Interrupt signals	NMI	32	38	Input	Nonmaskable interrupt request pin. Fix high when not used.
	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ ( $\overline{\text{IRQ7}}$ ) to ( $\overline{\text{IRQ0}}$ )	29 to 26, 112 to 109, 102 to 95	33 to 30, 122 to 119, 112 to 105	Input	These pins request a maskable interrupt.  The input pins of $\overline{\text{IRQn}}$ and ( $\overline{\text{IRQn}}$ ) are selected by the IRQ pin select register (ITSR) of the interrupt controller. (n = 0 to 7)
DMA controller (DMAC)	$\overline{\text{DREQ1}}$ $\overline{\text{DREQ0}}$	35, 34	41, 40	Input	These signals request DMAC activation.
	$\overline{\text{TEND1}}$ , $\overline{\text{TEND0}}$	37, 36	43, 42	Output	These signals indicate the end of DMAC data transfer.
	$\overline{\text{DACK1}}$ , $\overline{\text{DACK0}}$	39, 38	45, 44	Output	DMAC single address transfer acknowledge signals.

16-bit timer pulse unit (TPU)	TCLKD	41,	47,	Input	External clock input pins for the timer.	
	TCLKC	39,	45,			
	TCLKB	37,	43,			
		TCLKA	36	42		
		TIOCA0	34,	40,	Input/ output	TGRA_0 to TGRD_0 input capture input/output compare output/PWM output pins.
		TIOCB0	35,	41,		
		TIOCC0	36,	42,		
		TIOCD0	37	43		
		TIOCA1	38,	44,	Input/ output	TGRA_1 and TGRB_1 input capture input/output compare output/PWM output pins.
	TIOCB1	39	45			
	TIOCA2	40,	46,	Input/ output	TGRA_2 and TGRB_2 input capture input/output compare output/PWM output pins.	
	TIOCB2	41	47			
	TIOCA3	42,	48,	Input/ output	TGRA_3 to TGRD_3 input capture input/output compare output/PWM output pins.	
	TIOCB3	43,	49,			
	TIOCC3	44,	50,			
	TIOCD3	45	51			
	TIOCA4	46,	52,	Input/ output	TGRA_4 and TGRB_4 input capture input/output compare output/PWM output pins.	
	TIOCB4	47	53			
	TIOCA5	48,	54,	Input/ output	TGRA_5 and TGRB_5 input capture input/output compare output/PWM output pins.	
	TIOCB5	49	55			
Programmable pulse generator (PPG)	PO15 to PO0	41 to 34, 49 to 42	47 to 40, 55 to 48	Output	Pulse output pins.	
8-bit timer (TMR)	TMO0	46,	52,	Output	Waveform output pins with output compare function.	
	TMO1	47	53			
		TMCI0	44,	50,	Input	External event input pins.
		TMCI1	45	51		
		TMRI0	42,	48,	Input	Counter reset input pins.
	TMRI1	43	49			
Watchdog Timer (WDT)	WDTOVF	31	37	Output	Counter overflow signal output pin in watchdog timer mode.	

Serial communication interface(SCI)/ smart card interface (SCI_0 with IrDA function)	TxD4	45,	51,	Output	Data output pins.
	TxD3	86,	94,		
	TxD2	109,	119,		
	TxD1	117,	127,		
	TxD0/ IrTxD	118	128		
	RxD4	46,	52,	Input	Data input pins.
	RxD3	85,	93,		
	RxD2	110,	120		
	RxD1	115,	125,		
	RxD0/ IrRxD	116	126		
	SCK4	114,	124,	Input/ output	Clock input/output pins.
	SCK3	50,	56,		
	SCK2	111,	121,		
	SCK1	113,	123,		
	SCK0	114	124		
IIC bus interface2 (IIC2)	SCL1	115,	125,	Input/ output	IIC clock input/output pins.
	SCL0	113	123		
	SDA1	116,	126,	Input/ output	IIC data input/output pins.
	SDA0	114	124		
A/D converter	AN13, AN12, AN7 to AN0	104 to 95	114 to 105	Input	Analog input pins.
	$\overline{\text{ADTRG}}$	112	122		
D/A converter	DA3, DA2	104, 103	114, 113	Output	Analog output pins.
A/D converter, D/A converter	$\text{AV}_{\text{cc}}$	93	103	Input	The analog power-supply pin for the A/D converter and D/A converter. When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+3 V).
	$\text{AV}_{\text{ss}}$	105	115		
				Input	The ground pin for the A/D converter and D/A converter. This pin should be connected to the system power supply (0 V).

A/D converter, D/A converter	Vref	94	104	Input	The reference voltage input pin for the A/D converter and D/A converter.  When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+3 V).
I/O ports	P17 to P104	1 to 34	47 to 40	Input/ output	Eight-bit input/output pins.
	P27 to P204	9 to 42	55 to 48	Input/ output	Eight-bit input/output pins.
	P35 to P301	13 to 118	123 to 128	Input/ output	Six-bit input/output pins.
	P47 to P401	2 to 95	112 to 105	Input	Eight-bit input pins.
	P53 to P501	12 to 109	122 to 119	Input/ output	Four-bit input/output pins.
	P85, P83, P81	50, 85, 86	56, 93, 94	Input/ output	Three-bit input/output pins.
	P95, P94	104, 103	114, 113	Input	Two-bit input pins.
	PA7 to PA0	29 to 23, 21	33 to 27, 25	Input/ output	Eight-bit input/output pins.
	PB7 to PB0	20 to 18, 16 to 12	24 to 22, 20 to 16	Input/ output	Eight-bit input/output pins.
	PC7 to PC0	11 to 9, 7 to 3	15 to 13, 11 to 7	Input/ output	Eight-bit input/output pins.
	PD7 to PD0	68 to 61	76 to 69	Input/ output	Eight-bit input/output pins.
	PE7 to PE0	59, 57 to 51	65, 63 to 57	Input/ output	Eight-bit input/output pins.
	PF7 to PF0	79, 75 to 69	87, 83 to 77	Input/ output	Eight-bit input/output pins.

---

I/O ports	PG6 to PG0	108 to 106, 92 to 89	118 to 116, 102,101, 98,97	Input/ output	Seven-bit input/output pins.
-----------	---------------	-------------------------	----------------------------------	------------------	------------------------------

---

- Notes:
1. Not supported by the H8S/2368 0.18  $\mu\text{m}$  F-ZTAT Group.
  2. VCL on the H8S/2368 0.18  $\mu\text{m}$  F-ZTAT Group. Do not connect to VCL.
  3. H8S/2368 0.18  $\mu\text{m}$  F-ZTAT Group only.



The H8S/2000 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2000 CPU has sixteen 16-bit general registers, can address a 16-Mbyte linear address space, and is ideal for realtime control.

This section describes the H8S/2000 CPU. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.

## 2.1 Features

- Upward-compatibility with H8/300 and H8/300H CPUs
  - Can execute H8/300 and H8/300H CPU object programs
- General-register architecture
  - Sixteen 16-bit general registers also usable as sixteen 8-bit registers or eight 32-bit registers
- Sixty-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @-ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes
- High-speed operation
  - All frequently-used instructions are executed in one or two states
  - 8/16/32-bit register-register add/subtract: 1 state
  - $8 \times 8$ -bit register-register multiply: 12 states (MULXU.B), 13 states (MULXS.B)
  - $16 \div 8$ -bit register-register divide: 12 states (DIVXU.B)

- Two CPU operating modes
  - Normal mode\*
  - Advanced mode

Note: \* For this LSI, normal mode is not available.

- Power-down state
  - Transition to power-down state by SLEEP instruction
  - Selectable CPU clock speed

### 2.1.1 Differences between H8S/2600 CPU and H8S/2000 CPU

The differences between the H8S/2600 CPU and the H8S/2000 CPU are as shown below.

- Register configuration
  - The MAC register is supported only by the H8S/2600 CPU.
- Basic instructions
  - The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by the H8S/2600 CPU.
- The number of execution states of the MULXU and MULXS instructions

Instruction	Mnemonic	Execution States	
		H8S/2600	H8S/2000
MULXU	MULXU.B Rs, Rd	3	12
	MULXU.W Rs, ERd	4	20
MULXS	MULXS.B Rs, Rd	4	13
	MULXS.W Rs, ERd	5	21

In addition, there are differences in address space, CCR and EXR register functions, power-down modes, etc., depending on the model.

In comparison to the H8/300 CPU, the H8S/2000 CPU has the following enhancements.

- More general registers and control registers
  - Eight 16-bit extended registers, and one 8-bit and two 32-bit control registers, have been added.
- Expanded address space
  - Normal mode supports the same 64-kbyte address space as the H8/300 CPU.
  - Advanced mode supports a maximum 16-Mbyte address space.
- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Signed multiply and divide instructions have been added.
  - Two-bit shift and two-bit rotate instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions are executed twice as fast.

### **2.1.3 Differences from H8/300H CPU**

In comparison to the H8/300H CPU, the H8S/2000 CPU has the following enhancements.

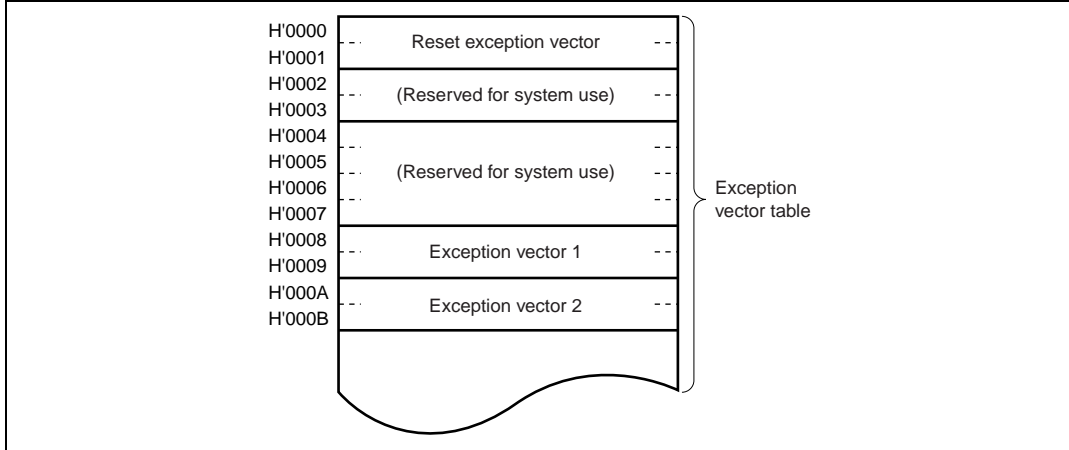
- Additional control register
  - One 8-bit control register has been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift and two-bit rotate instructions have been added.
  - Instructions for saving and restoring multiple registers have been added.
  - A test and set instruction has been added.
- Higher speed
  - Basic instructions are executed twice as fast.

The H8S/2000 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte address space. The mode is selected by the LSI's mode pins.

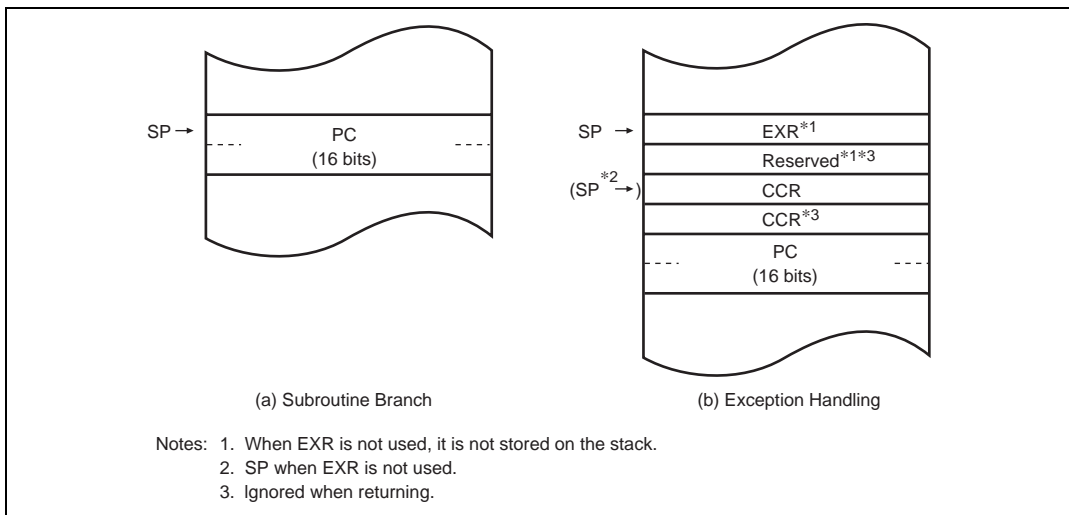
### 2.2.1 Normal Mode

The exception vector table and stack have the same structure as in the H8/300 CPU in normal mode.

- Address space  
Linear access to a maximum address space of 64 kbytes is possible.
- Extended registers (En)  
The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers.  
When extended register En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. (If general register Rn is referenced in the register indirect addressing mode with pre-decrement (@-Rn) or post-increment (@Rn+) and a carry or borrow occurs, the value in the corresponding extended register (En) will be affected.)
- Instruction set  
All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.
- Exception vector table and memory indirect branch addresses  
In normal mode, the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. The exception vector table in normal mode is shown in figure 2.1. For details of the exception vector table, see section 4, Exception Handling.  
The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode, the operand is a 16-bit (word) operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.
- Stack structure  
When the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.2. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.

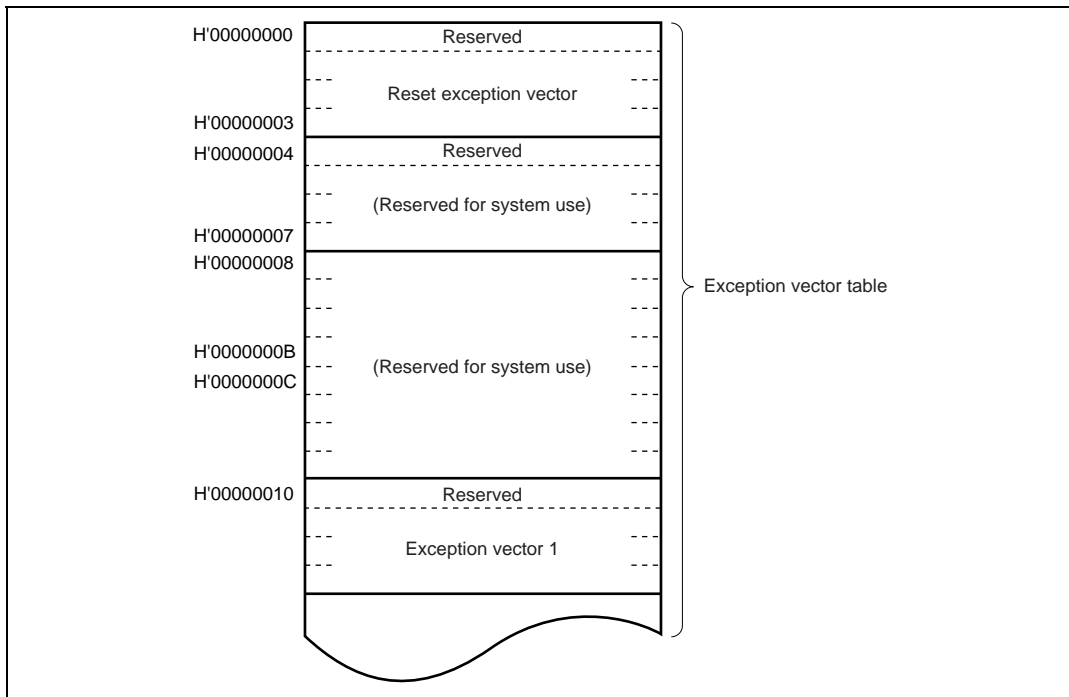


**Figure 2.1 Exception Vector Table (Normal Mode)**



**Figure 2.2 Stack Structure in Normal Mode**

- Address space  
Linear access to a maximum address space of 16 Mbytes is possible.
- Extended registers (En)  
The extended registers (E0 to E7) can be used as 16-bit registers. They can also be used as the upper 16-bit segments of 32-bit registers or address registers.
- Instruction set  
All instructions and addressing modes can be used.
- Exception vector table and memory indirect branch addresses  
In advanced mode, the top area starting at H'00000000 is allocated to the exception vector table in 32-bit units. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (see figure 2.3). For details of the exception vector table, see section 4, Exception Handling.

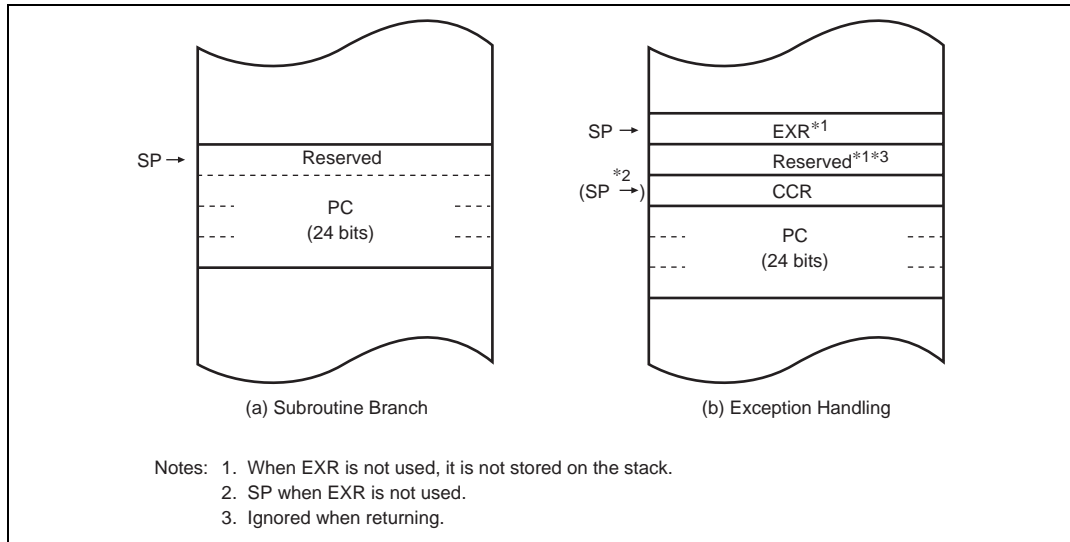


**Figure 2.3 Exception Vector Table (Advanced Mode)**

that contains a branch address. In advanced mode, the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper 8 bits of these 32 bits are a reserved area that is regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. Note that the top area of this range is also used for the exception vector table.

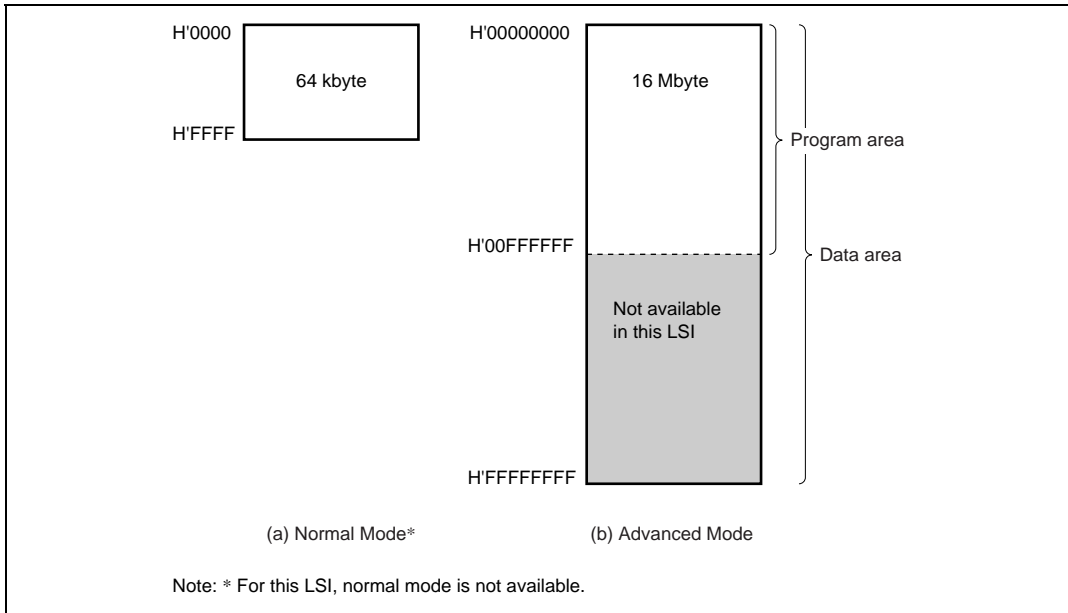
- Stack structure

In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.4. EXR is not pushed onto the stack in interrupt control mode 0. For details, see section 4, Exception Handling.



**Figure 2.4 Stack Structure in Advanced Mode**

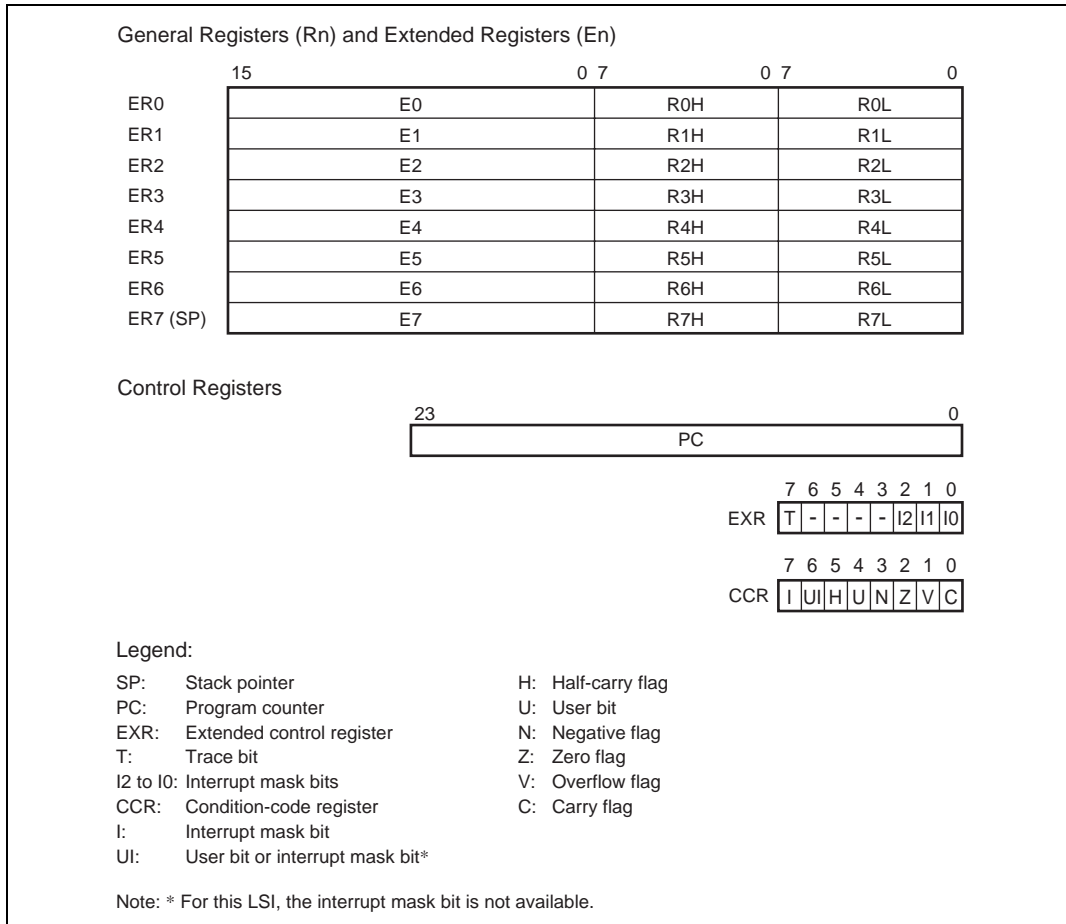
Figure 2.5 shows a memory map of the H8S/2000 CPU. The H8S/2000 CPU provides linear access to a maximum 64-kbyte address space in normal mode, and a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode. The usable modes and address spaces differ depending on the product. For details on each product, refer to section 3, MCU Operating Modes.



**Figure 2.5 Memory Map**



The H8S/2000 CPU has the internal registers shown in figure 2.6. There are two types of registers: general registers and control registers. Control registers are a 24-bit program counter (PC), an 8-bit extended control register (EXR), and an 8-bit condition code register (CCR).



**Figure 2.6 CPU Internal Registers**

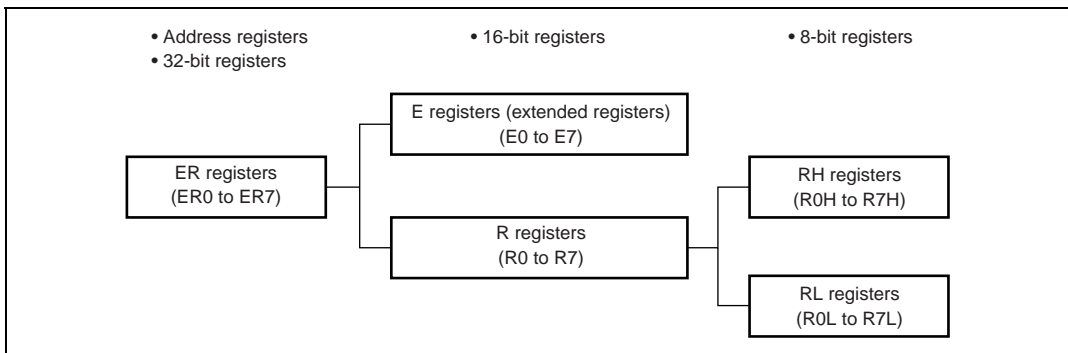
The H8S/2000 CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. Figure 2.7 illustrates the usage of the general registers. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

When the general registers are used as 16-bit registers, the ER registers are divided into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

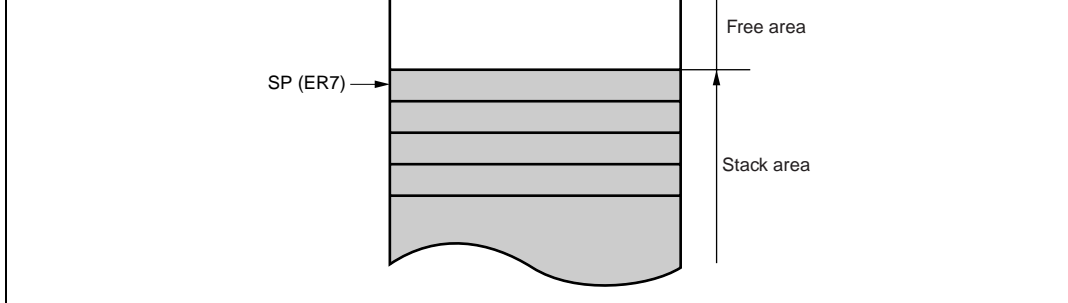
When the general registers are used as 8-bit registers, the R registers are divided into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixteen 8-bit registers.

The usage of each register can be selected independently.

General register ER7 has the function of the stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2.8 shows the stack.



**Figure 2.7 Usage of General Registers**



**Figure 2.8 Stack**

### 2.4.2 Program Counter (PC)

This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an instruction is fetched for read, the least significant PC bit is regarded as 0.)

### 2.4.3 Extended Control Register (EXR)

EXR is an 8-bit register that can be operated by the LDC, STC, ANDC, ORC, and XORC instructions. When an instruction other than STC is executed, all interrupts including NMI are masked in three states after the instruction is completed.

Bit	Bit Name	Initial Value	R/W	Description
7	T	0	R/W	Trace Bit When this bit is set to 1, trace exception processing starts every when an instruction is executed. When this bit is cleared to 0, instructions are consecutively executed.
6 to 3	–	All 1	–	Reserved These bits are always read as 1.
2 to 0	I2 I1 I0	1	R/W	Interrupt Mask Bits 2 to 0 Specify interrupt request mask levels (0 to 7). For details, see section 5, Interrupt Controller.

This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

Masks interrupts other than NMI when set to 1. NMI is accepted regardless of the I bit setting. The I bit is set to 1 at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.

6	UI	Undefined	R/W	<p>User Bit or Interrupt Mask Bit</p> <p>Can be written to and read from by software using the LDC, STC, ANDC, ORC, and XORC instructions. For this LSI, Interrupt Mask Bit is not available.</p>
5	H	Undefined	R/W	<p>Half-Carry Flag</p> <p>When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.</p>
4	U	Undefined	R/W	<p>User Bit</p> <p>Can be written to and read from by software using the LDC, STC, ANDC, ORC, and XORC instructions.</p>
3	N	Undefined	R/W	<p>Negative Flag</p> <p>Stores the value of the most significant bit of data as a sign bit.</p>
2	Z	Undefined	R/W	<p>Zero Flag</p> <p>Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.</p>
1	V	Undefined	R/W	<p>Overflow Flag</p> <p>Set to 1 when an arithmetic overflow occurs, and cleared to 0 otherwise.</p>
0	C	Undefined	R/W	<p>Carry Flag</p> <p>Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by:</p> <ul style="list-style-type: none"> <li>• Add instructions, to indicate a carry</li> <li>• Subtract instructions, to indicate a borrow</li> <li>• Shift and rotate instructions, to indicate a carry</li> </ul> <p>The carry flag is also used as a bit accumulator by bit manipulation instructions.</p>

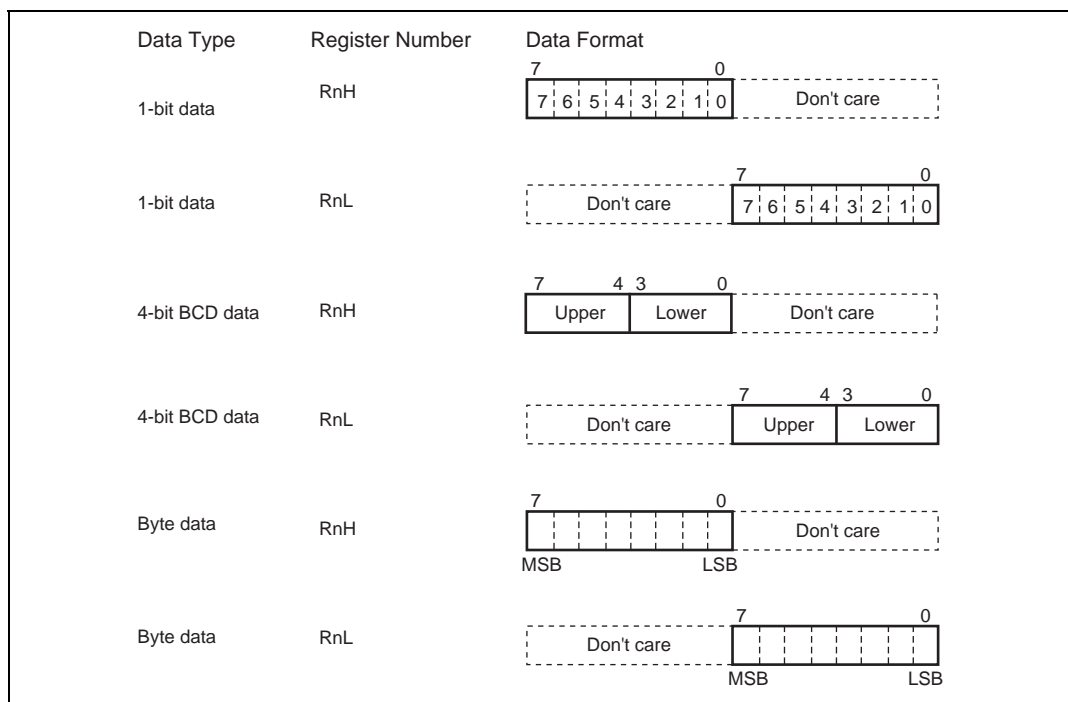
Reset exception handling loads the CPU's program counter (PC) from the vector table, clears the trace (T) bit in EXR to 0, and sets the interrupt mask (I) bits in CCR and EXR to 1. The other CCR bits and the general registers are not initialized. Note that the stack pointer (ER7) is undefined. The stack pointer should therefore be initialized by an MOV.L instruction executed immediately after a reset.

## 2.5 Data Formats

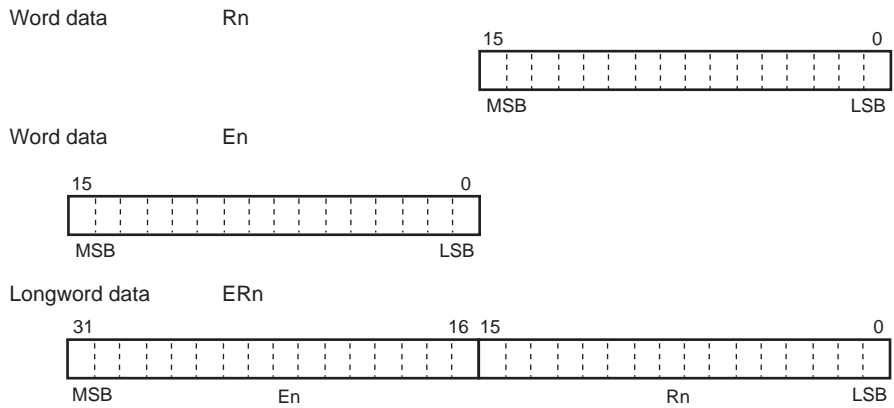
The H8S/2000 CPU can process 1-bit, 4-bit BCD, 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit  $n$  ( $n = 0, 1, 2, \dots, 7$ ) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

### 2.5.1 General Register Data Formats

Figure 2.9 shows the data formats of general registers.



**Figure 2.9 General Register Data Formats (1)**



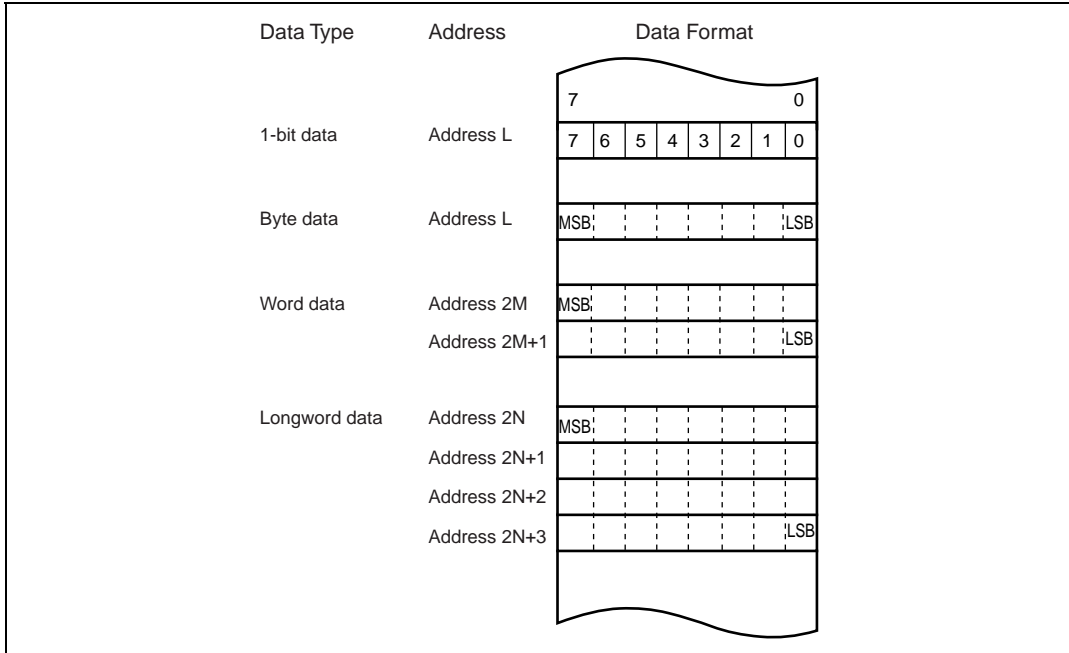
Legend:

- ERn: General register ER
- En: General register E
- Rn: General register R
- RnH: General register RH
- RnL: General register RL
- MSB: Most significant bit
- LSB: Least significant bit

**Figure 2.9 General Register Data Formats (2)**

Figure 2.10 shows the data formats in memory. The H8S/2000 CPU can access word data and longword data in memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

When SP (ER7) is used as an address register to access the stack, the operand size should be word size or longword size.



**Figure 2.10 Memory Data Formats**



The H8S/2000 CPU has 65 types of instructions. The instructions are classified by function as shown in table 2.1.

**Table 2.1 Instruction Classification**

Function	Instructions	Size	Types
Data transfer	MOV	B/W/L	5
	POP <sup>*1</sup> , PUSH <sup>*1</sup>	W/L	
	LDM, STM	L	
	MOVFP <sup>*3</sup> , MOVTP <sup>*3</sup>	B	
Arithmetic operations	ADD, SUB, CMP, NEG	B/W/L	19
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	B/W/L	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	B/W	
	EXTU, EXTS	W/L	
	TAS <sup>*4</sup>	B	
Logic operations	AND, OR, XOR, NOT	B/W/L	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	B/W/L	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BBIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	B <sub>cc</sub> <sup>*2</sup> , JMP, BSR, JSR, RTS	–	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	–	9
Block data transfer	EEPMOV	–	1

Total: 65

Notes: B: Byte size; W: Word size; L: Longword size.

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. BCC is the general name for conditional branch instructions.
3. Cannot be used in this LSI.
4. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

Tables 2.3 to 2.10 summarize the instructions in each functional category. The notation used in tables 2.3 to 2.10 is defined below.

**Table 2.2 Operation Notation**

<b>Symbol</b>	<b>Description</b>
Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
−	Subtraction
×	Multiplication
÷	Division
^	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Move
~	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: \* General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

Instruction	Size	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPPE	B	Cannot be used in this LSI.
MOVTPPE	B	Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn
PUSH	W/L	Rn → @-SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @-SP. PUSH.L ERn is identical to MOV.L ERn, @-SP.
LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM	L	Rn (register list) → @-SP Pushes two or more general registers onto the stack.

Note: \* Size refers to the operand size.

B: Byte

W: Word

L: Longword

Instruction	Size	Function
ADD SUB	B/W/L	$Rd \pm Rs \rightarrow Rd$ , $Rd \pm \#IMM \rightarrow Rd$ Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register. (Subtraction on immediate data and data in a general register cannot be performed in bytes. Use the SUBX or ADD instruction.)
ADDX SUBX	B	$Rd \pm Rs \pm C \rightarrow Rd$ , $Rd \pm \#IMM \pm C \rightarrow Rd$ Performs addition or subtraction with carry on data in two general registers, or on immediate data and data in a general register.
INC DEC	B/W/L	$Rd \pm 1 \rightarrow Rd$ , $Rd \pm 2 \rightarrow Rd$ Adds or subtracts the value 1 or 2 to or from data in a general register. (Only the value 1 can be added to or subtracted from byte operands.)
ADDS SUBS	L	$Rd \pm 1 \rightarrow Rd$ , $Rd \pm 2 \rightarrow Rd$ , $Rd \pm 4 \rightarrow Rd$ Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
DAA DAS	B	$Rd$ (decimal adjust) $\rightarrow Rd$ Decimal-adjusts an addition or subtraction result in a general register by referring to CCR to produce 4-bit BCD data.
MULXU	B/W	$Rd \times Rs \rightarrow Rd$ Performs unsigned multiplication on data in two general registers: either 8 bits $\times$ 8 bits $\rightarrow$ 16 bits or 16 bits $\times$ 16 bits $\rightarrow$ 32 bits.
MULXS	B/W	$Rd \times Rs \rightarrow Rd$ Performs signed multiplication on data in two general registers: either 8 bits $\times$ 8 bits $\rightarrow$ 16 bits or 16 bits $\times$ 16 bits $\rightarrow$ 32 bits.
DIVXU	B/W	$Rd \div Rs \rightarrow Rd$ Performs unsigned division on data in two general registers: either 16 bits $\div$ 8 bits $\rightarrow$ 8-bit quotient and 8-bit remainder or 32 bits $\div$ 16 bits $\rightarrow$ 16-bit quotient and 16-bit remainder.

Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.

CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and sets the CCR bits according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.
TAS* <sup>2</sup>	B	@ERd – 0, 1 → (<bit 7> of @ERd) Tests memory contents, and sets the most significant bit (bit 7) to 1.

Notes: 1. Size refers to the operand size.

B: Byte

W: Word

L: Longword

2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

Instruction	Size	Function
AND	B/W/L	$Rd \wedge Rs \rightarrow Rd, Rd \wedge \#IMM \rightarrow Rd$ Performs a logical AND operation on a general register and another general register or immediate data.
OR	B/W/L	$Rd \vee Rs \rightarrow Rd, Rd \vee \#IMM \rightarrow Rd$ Performs a logical OR operation on a general register and another general register or immediate data.
XOR	B/W/L	$Rd \oplus Rs \rightarrow Rd, Rd \oplus \#IMM \rightarrow Rd$ Performs a logical exclusive OR operation on a general register and another general register or immediate data.
NOT	B/W/L	$\sim Rd \rightarrow Rd$ Takes the one's complement (logical complement) of data in a general register.

Note: \* Size refers to the operand size.

B: Byte

W: Word

L: Longword

**Table 2.6 Shift Instructions**

Instruction	Size*	Function
SHAL	B/W/L	$Rd \text{ (shift)} \rightarrow Rd$
SHAR		Performs an arithmetic shift on data in a general register. 1-bit or 2 bit shift is possible.
SHLL	B/W/L	$Rd \text{ (shift)} \rightarrow Rd$
SHLR		Performs a logical shift on data in a general register. 1-bit or 2 bit shift is possible.
ROTL	B/W/L	$Rd \text{ (rotate)} \rightarrow Rd$
ROTR		Rotates data in a general register. 1-bit or 2 bit rotation is possible.
ROTXL	B/W/L	$Rd \text{ (rotate)} \rightarrow Rd$
ROTXR		Rotates data including the carry flag in a general register. 1-bit or 2 bit rotation is possible.

Note: \* Size refers to the operand size.

B: Byte

W: Word

L: Longword

Instruction	Size	Function
BSET	B	$1 \rightarrow (\text{<bit-No.> of <EAd>})$ Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BCLR	B	$0 \rightarrow (\text{<bit-No.> of <EAd>})$ Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BNOT	B	$\sim (\text{<bit-No.> of <EAd>}) \rightarrow (\text{<bit-No.> of <EAd>})$ Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BTST	B	$\sim (\text{<bit-No.> of <EAd>}) \rightarrow Z$ Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BAND	B	$C \wedge (\text{<bit-No.> of <EAd>}) \rightarrow C$ Logically ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIAND	B	$C \wedge (\sim (\text{<bit-No.> of <EAd>})) \rightarrow C$ Logically ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BOR	B	$C \vee (\text{<bit-No.> of <EAd>}) \rightarrow C$ Logically ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIOR	B	$C \vee (\sim (\text{<bit-No.> of <EAd>})) \rightarrow C$ Logically ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

		Logically exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
BIXOR	B	$C \oplus \sim (\text{<bit-No.> of <EAd>}) \rightarrow C$ Logically exclusive-ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
BLD	B	$(\text{<bit-No.> of <EAd>}) \rightarrow C$ Transfers a specified bit in a general register or memory operand to the carry flag.
BILD	B	$\sim (\text{<bit-No.> of <EAd>}) \rightarrow C$ Transfers the inverse of a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data.
BST	B	$C \rightarrow (\text{<bit-No.> of <EAd>})$ Transfers the carry flag value to a specified bit in a general register or memory operand.
BIST	B	$\sim C \rightarrow (\text{<bit-No.>. of <EAd>})$ Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.

Note: \* Size refers to the operand size.

B: Byte



Instruction	Size	Function
Bcc	–	Branches to a specified address if a specified condition is true. The branching conditions are listed below.

Mnemonic	Description	Condition
BRA (BT)	Always (true)	Always
BRN (BF)	Never (false)	Never
BHI	High	$C \vee Z = 0$
BLS	Low or same	$C \vee Z = 1$
BCC (BHS)	Carry clear (high or same)	$C = 0$
BCS (BLO)	Carry set (low)	$C = 1$
BNE	Not equal	$Z = 0$
BEQ	Equal	$Z = 1$
BVC	Overflow clear	$V = 0$
BVS	Overflow set	$V = 1$
BPL	Plus	$N = 0$
BMI	Minus	$N = 1$
BGE	Greater or equal	$N \oplus V = 0$
BLT	Less than	$N \oplus V = 1$
BGT	Greater than	$Z \vee (N \oplus V) = 0$
BLE	Less or equal	$Z \vee (N \oplus V) = 1$

JMP	–	Branches unconditionally to a specified address.
BSR	–	Branches to a subroutine at a specified address.
JSR	–	Branches to a subroutine at a specified address.
RTS	–	Returns from a subroutine.

Instruction	Size	Function
TRAPA	–	Starts trap-instruction exception handling.
RTE	–	Returns from an exception-handling routine.
SLEEP	–	Causes a transition to a power-down state.
LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves the memory operand contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory operand. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR contents with immediate data.
NOP	–	PC + 2 → PC Only increments the program counter.

Note: \* Size refers to the operand size.

B: Byte

W: Word

Instruction	Size	Function
EEPMOV.B	–	if R4L $\neq$ 0 then Repeat @ER5+ $\rightarrow$ @ER6+ R4L–1 $\rightarrow$ R4L Until R4L = 0 else next:
EEPMOV.W	–	if R4 $\neq$ 0 then Repeat @ER5+ $\rightarrow$ @ER6+ R4–1 $\rightarrow$ R4 Until R4 = 0 else next: Transfers a data block. Starting from the address set in ER5, transfers data for the number of bytes set in R4L or R4 to the address location set in ER6. Execution of the next instruction begins as soon as the transfer is completed.

## 2.6.2 Basic Instruction Formats

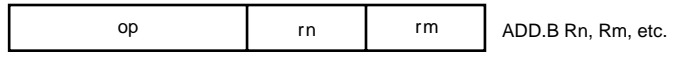
The H8S/2000 CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op), a register field (r), an effective address extension (EA), and a condition field (cc).

Figure 2.11 shows examples of instruction formats.

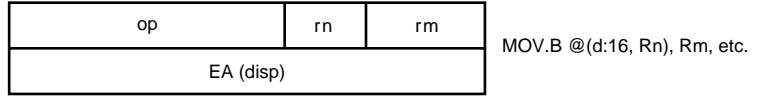
- **Operation field**  
Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first four bits of the instruction. Some instructions have two operation fields.
- **Register field**  
Specifies a general register. Address registers are specified by 3 bits, and data registers by 3 bits or 4 bits. Some instructions have two register fields, and some have no register field.
- **Effective address extension**  
8, 16, or 32 bits specifying immediate data, an absolute address, or a displacement.
- **Condition field**  
Specifies the branching condition of Bcc instructions.

op NOP, RTS, etc.

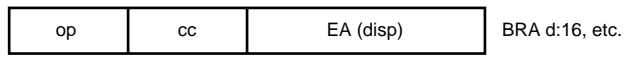
(2) Operation field and register fields



(3) Operation field, register fields, and effective address extension



(4) Operation field, effective address extension, and condition field



**Figure 2.11 Instruction Formats (Examples)**

The H8S/2000 CPU supports the eight addressing modes listed in table 2.11. Each instruction uses a subset of these addressing modes.

Arithmetic and logic operations instructions can use the register direct and immediate addressing modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions can use register direct, register indirect, or absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

**Table 2.11 Addressing Modes**

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16,ERn)/@(d:32,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @-ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

### 2.7.1 Register Direct—Rn

The register field of the instruction code specifies an 8-, 16-, or 32-bit general register which contains the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

### 2.7.2 Register Indirect—@ERn

The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

A 16-bit or 32-bit displacement contained in the instruction code is added to an address register (ERn) specified by the register field of the instruction, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

#### 2.7.4 Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn

**Register Indirect with Post-Increment—@ERn+:** The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word access, and 4 for longword access. For word or longword transfer instructions, the register value should be even.

**Register Indirect with Pre-Decrement—@-ERn:** The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the result becomes the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word access, and 4 for longword access. For word or longword transfer instructions, the register value should be even.

#### 2.7.5 Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32

The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long (@aa:32). Table 2.12 indicates the accessible absolute address ranges.

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address, the upper 16 bits are a sign extension. For a 32-bit absolute address, the entire address space is accessed.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper 8 bits are all assumed to be 0 (H'00).

		Normal Mode	Advanced Mode
Data address	8 bits (@aa:8)	H'FF00 to H'FFFF	H'FFFF00 to H'FFFFFF
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF
	32 bits (@aa:32)		H'000000 to H'FFFFFF
Program instruction address	24 bits (@aa:24)		

### 2.7.6 Immediate—#xx:8, #xx:16, or #xx:32

The 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data contained in a instruction code can be used directly as an operand.

The ADDS, SUBS, INC, and DEC instructions implicitly contain immediate data in their instruction codes. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. The TRAPA instruction contains 2-bit immediate data in its instruction code, specifying a vector address.

### 2.7.7 Program-Counter Relative—@(d:8, PC) or @(d:16, PC)

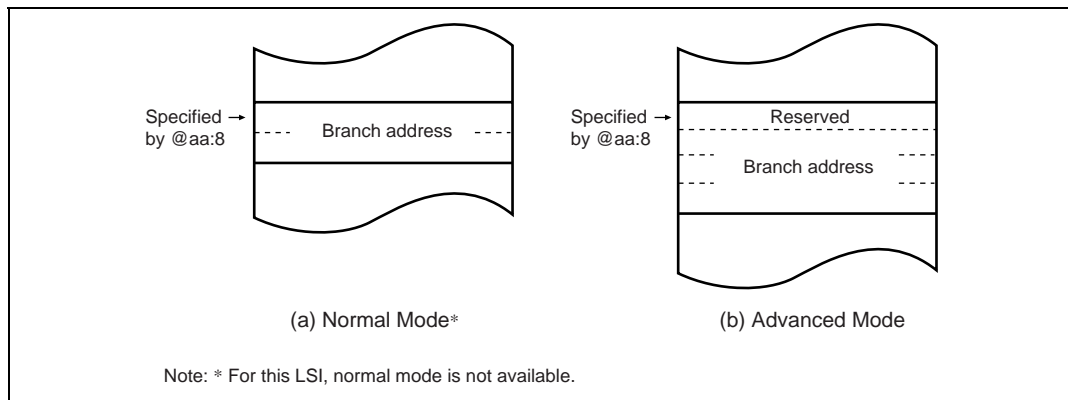
This mode can be used by the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction code is sign-extended to 24 bits and added to the 24-bit address indicated by the PC value to generate a 24-bit branch address. Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00). The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is -126 to +128 bytes (-63 to +64 words) or -32766 to +32768 bytes (-16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

### 2.7.8 Memory Indirect—@@aa:8

This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand which contains a branch address. The upper bits of the 8-bit absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF in normal mode, H'000000 to H'0000FF in advanced mode).

In normal mode, the memory operand is a word operand and the branch address is 16 bits long. In advanced mode, the memory operand is a longword operand, the first byte of which is assumed to be 0 (H'00).

If an odd address is specified in word or longword memory access, or as a branch address, the least significant bit is regarded as 0, causing data to be accessed or the instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

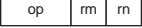

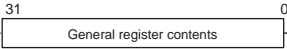
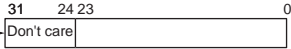

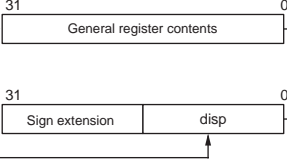
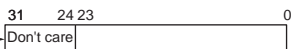


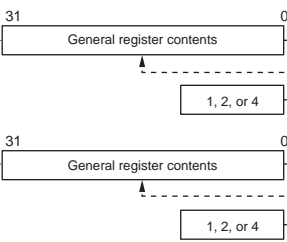
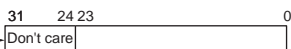



**Figure 2.12 Branch Address Specification in Memory Indirect Addressing Mode**

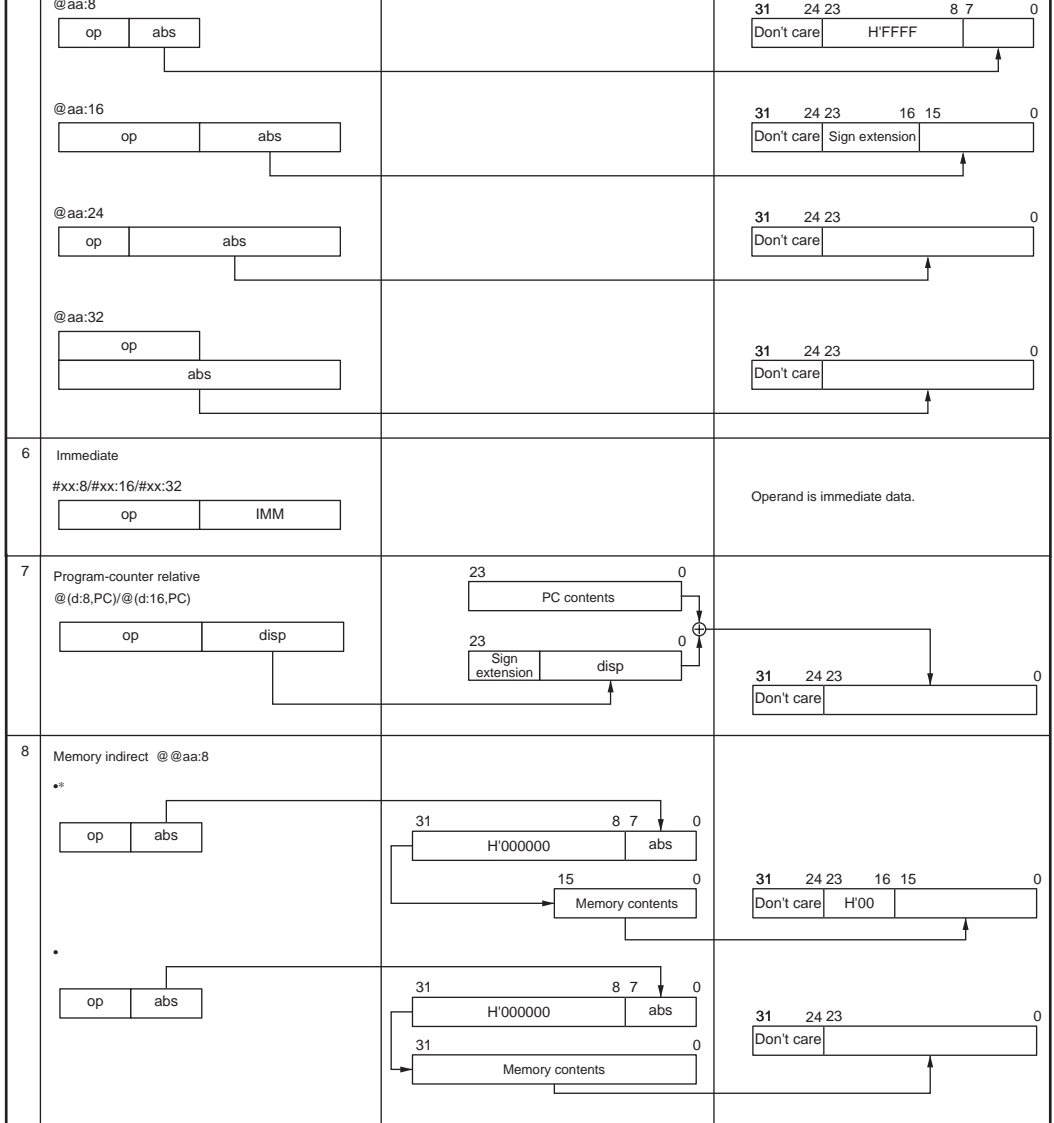
### 2.7.9 Effective Address Calculation

Table 2.13 indicates how effective addresses are calculated in each addressing mode. In normal mode, the upper 8 bits of the effective address are ignored in order to generate a 16-bit address.



No	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)								
1	Register direct (Rn) 		Operand is general register contents.								
2	Register indirect (@ERn) 										
3	Register indirect with displacement @(d:16,ERn) or @(d:32,ERn) 										
4	Register indirect with post-increment or pre-decrement •Register indirect with post-increment @ERn+  •Register indirect with pre-decrement @-ERn 	 <table border="1" data-bbox="547 678 776 758"> <thead> <tr> <th>Operand Size</th> <th>Offset</th> </tr> </thead> <tbody> <tr> <td>Byte</td> <td>1</td> </tr> <tr> <td>Word</td> <td>2</td> </tr> <tr> <td>Longword</td> <td>4</td> </tr> </tbody> </table>	Operand Size	Offset	Byte	1	Word	2	Longword	4	 
Operand Size	Offset										
Byte	1										
Word	2										
Longword	4										





Note: \* For this LSI, normal mode is not available.

The H8S/2000 CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and program stop state. Figure 2.13 indicates the state transitions.

- Reset state

In this state the CPU and internal peripheral modules are all initialized and stopped. When the  $\overline{\text{RES}}$  input goes low, all current processing stops and the CPU enters the reset state. All interrupts are masked in the reset state. Reset exception handling starts when the  $\overline{\text{RES}}$  signal changes from low to high. For details, refer to section 4, Exception Handling.

The reset state can also be entered by a watchdog timer overflow.

- Exception-handling state

The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to an exception source, such as, a reset, trace, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address. For further details, refer to section 4, Exception Handling.

- Program execution state

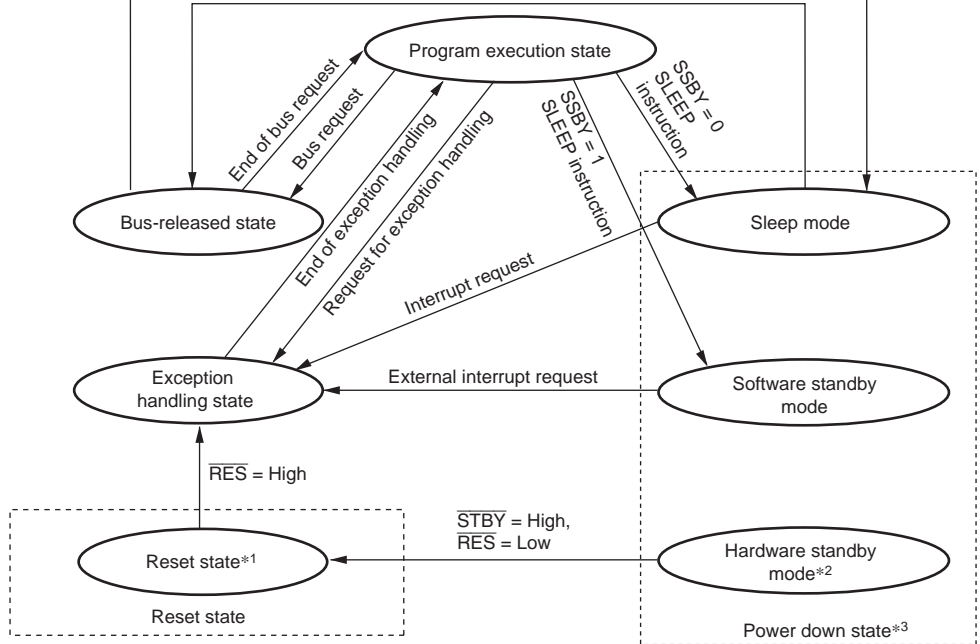
In this state the CPU executes program instructions in sequence.

- Bus-released state

In a product which has a DMA controller and a data transfer controller (DTC), the bus-released state occurs when the bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts operations.

- Program stop state

This is a power-down state in which the CPU stops operating. The program stop state occurs when a SLEEP instruction is executed or the CPU enters hardware standby mode. For details, refer to section 23, Power-Down Modes.



- Notes:
1. From any state except hardware standby mode, a transition to the reset state occurs whenever  $\overline{RES}$  goes low. A transition can also be made to the reset state when the watchdog timer overflows.
  2. In every state, when the STBY pin becomes low, the hardware standby mode is entered.
  3. For details, refer to section 23, Power-Down Modes.

**Figure 2.13 State Transitions**

## 2.9 Usage Note

### 2.9.1 Note on Bit Manipulation Instructions

Bit manipulation instructions such as BSET, BCLR, BNOT, BST, and BIST read data in byte units, perform bit manipulation, and write data in byte units. Thus, care must be taken when these bit manipulation instructions are executed for a register or port including write-only bits.

In addition, the BCLR instruction can be used to clear the flag of an internal I/O register. In this case, if the flag to be cleared has been set by an interrupt processing routine, the flag need not be read before executing the BCLR instruction.

## 3.1 Operating Mode Selection

This LSI has six operating modes (modes 1 to 5 and 7). Modes 1 to 5 and 7 are available in the H8S/2368 0.18  $\mu\text{m}$  F-ZTAT Group flash memory version. Modes 1 to 4 and 7 are available in the H8S/2367F. Modes 1, 2, 4, and 7 are available in the masked ROM version. Modes 1 and 2 are available in the ROMless version. The operating mode is selected by the setting of mode pins (MD2 to MD0).

Modes 1, 2, and 4 are externally expanded modes in which the CPU can access an external memory and peripheral devices. In the externally expanded mode, each area can be switched to 8-bit or 16-bit address space by the bus controller. If any one of the areas is set to 16-bit address space, the bus mode is 16 bits. If all areas are set to 8-bit address space, the bus mode is 8 bits.

Mode 7 is a single-chip activation externally expanded mode in which the CPU can switch to access an external memory and peripheral devices at the beginning of a program execution.

Modes 3 and 5 are a boot mode/user boot mode in which the flash memory can be programmed or erased. For details on the boot mode/user boot mode, refer to section 19, Flash Memory (0.35- $\mu\text{m}$  F-ZTAT Version), or section 20, Flash Memory (0.18- $\mu\text{m}$  F-ZTAT Version).

Do not change the MD2 to MD0 pin settings during operation.

**Table 3.1 MCU Operating Mode Selection**

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus	
							Initial Width	Max. Value
1	0	0	1	Advanced	Expanded mode with on-chip ROM disabled	Disabled	16 bits	16 bits
2	0	1	0	Advanced	Expanded mode with on-chip ROM disabled	Disabled	8 bits	16 bits
3	0	1	1	Advanced	Boot mode	Enabled	—	16 bits
4	1	0	0	Advanced	Expanded mode with on-chip ROM enabled	Enabled	8 bits	16 bits
5*	1	0	1	Advanced	User boot mode	Enabled	—	16 bits
7	1	1	1	Advanced	Single-chip mode	Enabled	—	16 bits

Note: \* Supported only by the H8S/2368 0.18  $\mu\text{m}$  F-ZTAT Group.

The following registers are related to the operating mode.

- Mode control register (MDCR)
- System control register (SYSCR)

### 3.2.1 Mode Control Register (MDCR)

MDCR monitors the current operating mode of this LSI.

Bit	Bit Name	Initial Value	R/W	Descriptions
7 to 3	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
2	MDS2	—*	R	Mode Select 2 to 0
1	MDS1	—*	R	These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD2 to MD0. MDS2 to MDS0 are read-only bits and they cannot be written to. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are canceled by a reset.
0	MDS0	—*	R	

Note: \* Determined by pins MD2 to MD0.

### 3.2.2 System Control Register (SYSCR)

SYSCR controls CPU access to the flash memory control registers, sets external bus mode, and enables or disables on-chip RAM.

5, 4	—	All 0	R/W R/W	The initial value should not be modified. Reserved The initial value should not be modified.
3	FLSHE	0	R/W	Flash Memory Control Register Enable Controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2). If this bit is set to 1, the flash memory control registers can be read/written to. If this bit is cleared to 0, the flash memory control registers are not selected. At this time, the contents of the flash memory control registers are maintained. This bit should be written to 0 in other than flash memory version.  0: Flash memory control registers are not selected for area H'FFFFC8 to H'FFFFCB  1: Flash memory control registers are selected for area H'FFFFC8 to H'FFFFCB
2	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
1	EXPE	—	R/W	External Bus Mode Enable Sets external bus mode. In modes 1, 2, and 4, this bit is fixed at 1 and cannot be modified. In mode 3 and 7, this bit has an initial value of 0, and can be read and written. Writing of 0 to EXPE when its value is 1 should only be carried out when an external bus cycle is not being executed. 0: External bus disabled 1: External bus enabled
0	RAME	1	R/W	RAM Enable Enables or disables the on-chip RAM. The RAME bit is initialized when the reset status is released. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

### **3.3.1 Mode 1**

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled. Ports A, B, and C function as an address bus, ports D and E function as a data bus, and parts of ports F, and G, carry bus control signals.

The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. However, if 8-bit access is designated for all areas by the bus controller, the bus mode switches to 8 bits.

### **3.3.2 Mode 2**

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled. Ports A, B, and C function as an address bus, ports D and E function as a data bus, and parts of ports F, and G carry bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, if 16-bit access is designated for any one of the areas by the bus controller, the bus mode switches to 16 bits and port E functions as a data bus.

### **3.3.3 Mode 3**

This mode is a boot mode of the flash memory. This mode is the same as mode 7, except for the programming and erasure on the flash memory. Mode 3 is only available in the H8S/2368 Group flash memory version.

### **3.3.4 Mode 4**

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. The program in the on-chip ROM connected to the first half of area 0 is executed.

Ports A, B, and C function as input ports immediately after a reset, but can be set to function as an address bus. Ports D and E function as a data bus, and parts of ports F, and G, carry bus control signals. For details, see section 9, I/O Ports.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, if 16-bit access is designated for any area by the bus controller, the bus mode switches to 16 bits and port E functions as a data bus.



### 3.3.5 Mode 5

This mode is a user boot mode of the flash memory. This mode is the same as mode 7, except for the programming and erasure on the flash memory. Mode 5 is only available in the H8S/2368 0.18  $\mu\text{m}$  F-ZTAT Group.

### 3.3.6 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, and the chip starts up in single-chip mode. External address space cannot be used in single-chip mode.

The initial mode after a reset is single-chip mode, with all I/O ports available for use as input/output ports. However, the mode can be switched to externally expanded mode by setting 1 to the EXPE bit of SYSCR and then the external address space is enabled. When externally expanded mode is selected, all areas are initially designated as 16-bit access space. The functions of pins in ports A to G are the same as in externally expanded mode with on-chip ROM enabled.

In the flash memory version, user program mode is entered by setting the SWE bit of FLMCR1 to 1.

Table 3.2 shows the pin functions in each operating mode.

**Table 3.2 Pin Functions in Each Operating Mode**

Port		Mode 1	Mode 2	Mode 3 <sup>*2</sup>	Mode 4 <sup>*2</sup>	Mode 5 <sup>*2*3</sup>	Mode 7 <sup>*2</sup>
Port A	PA7 to PA5	P <sup>*1</sup> /A	P <sup>*1</sup> /A	P <sup>*1</sup> /A	P <sup>*1</sup> /A	P <sup>*1</sup> /A	P <sup>*1</sup> /A
	PA4 to PA0	A	A				
Port B		A	A	P <sup>*1</sup> /A	P <sup>*1</sup> /A	P <sup>*1</sup> /A	P <sup>*1</sup> /A
Port C		A	A	P <sup>*1</sup> /A	P <sup>*1</sup> /A	P <sup>*1</sup> /A	P <sup>*1</sup> /A
Port D		D	D	P <sup>*1</sup> /D	D	P <sup>*1</sup> /D	P <sup>*1</sup> /D
Port E		P/D <sup>*1</sup>	P <sup>*1</sup> /D	P <sup>*1</sup> /D	P <sup>*1</sup> /D	P <sup>*1</sup> /D	P <sup>*1</sup> /D
Port F	PF7, PF6	P/C <sup>*1</sup>	P <sup>*1</sup> /C	P <sup>*1</sup> /C	P/C <sup>*1</sup>	P <sup>*1</sup> /C	P <sup>*1</sup> /C
	PF5, PF4	C	C		C		
	PF3	P/C <sup>*1</sup>	P/C <sup>*1</sup>		P/C <sup>*1</sup>		
	PF2 to PF0	P <sup>*1</sup> /C	P <sup>*1</sup> /C		P <sup>*1</sup> /C		
Port G	PG6 to PG1	P <sup>*1</sup> /C	P <sup>*1</sup> /C	P <sup>*1</sup> /C	P <sup>*1</sup> /C	P <sup>*1</sup> /C	P <sup>*1</sup> /C
	PG0	P/C <sup>*1</sup>	P/C <sup>*1</sup>		P <sup>*1</sup> /C		

Legend:

P: I/O port

A: Address bus output

D: Data bus input/output

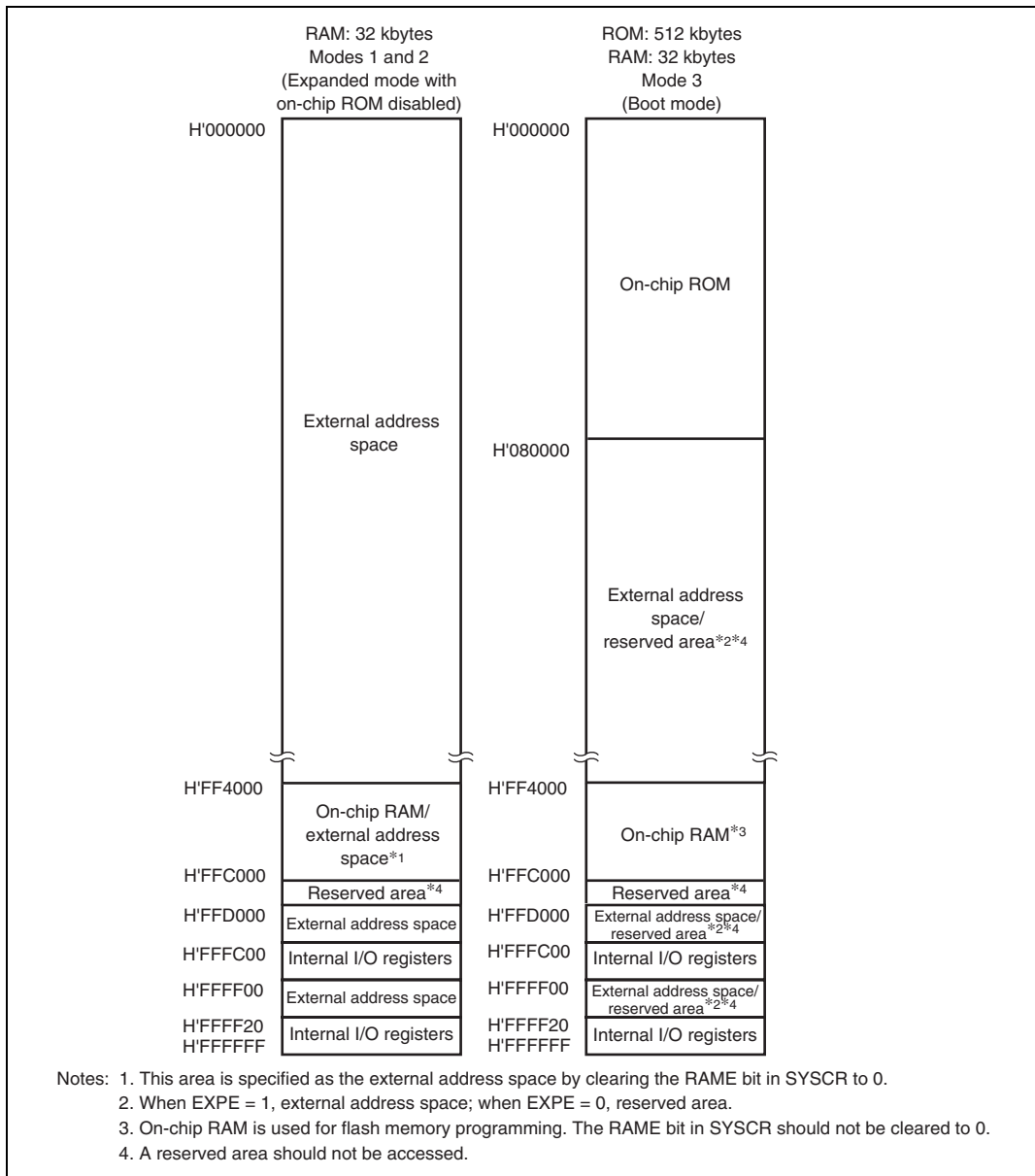
C: Control signals, clock input/output

Notes: 1. After reset

2. Setting not allowed on ROMless versions.

3. Mode 5 is available only in the H8S/2368 0.18 μm F-ZTAT Group.

Figures 3.1 to 3.15 show memory maps for each product.

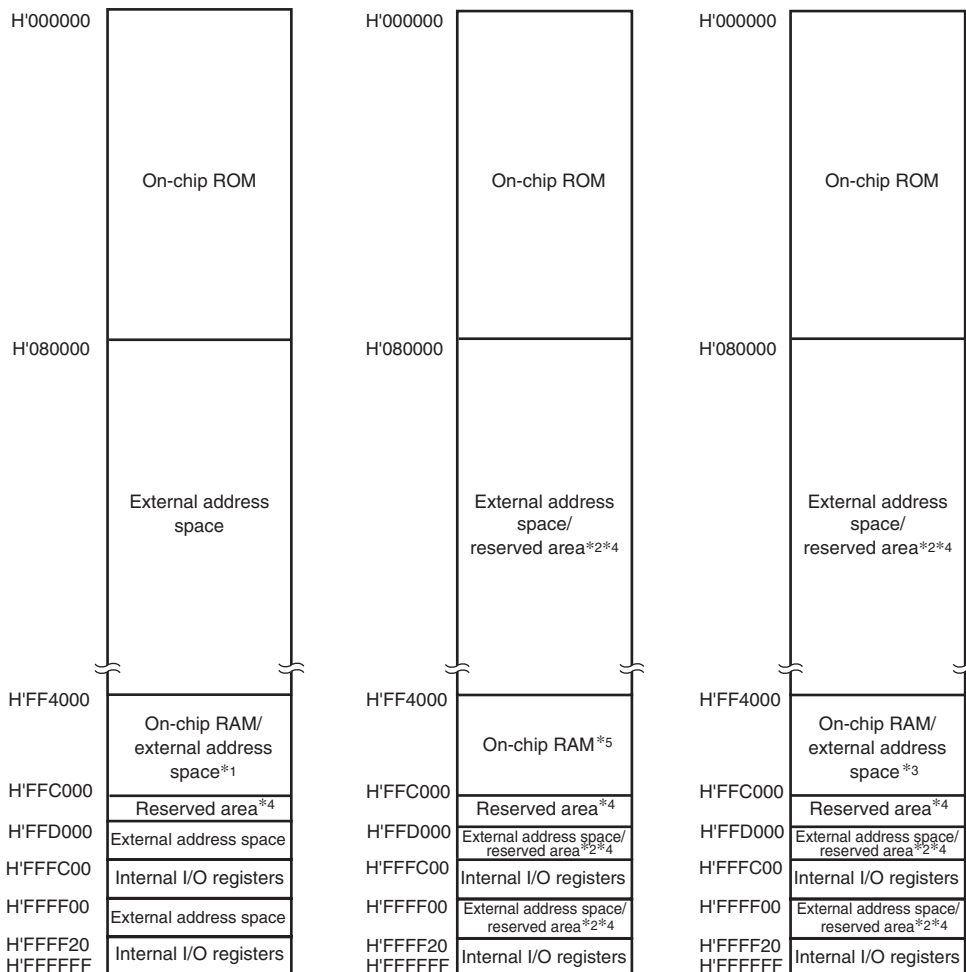


**Figure 3.1 H8S/2368F Memory Map (1)**

Mode 4  
(Expanded mode with  
on-chip ROM enabled)

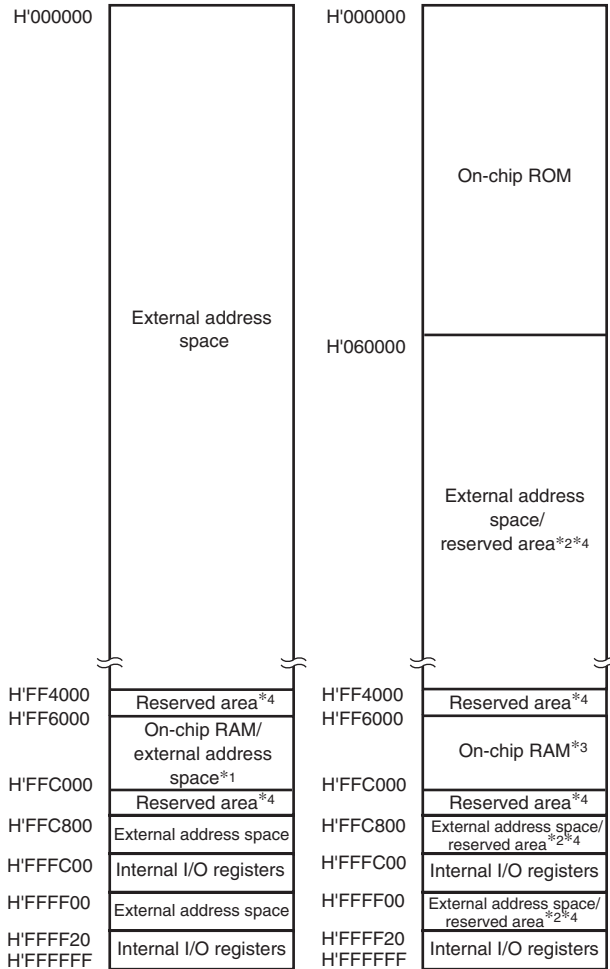
Mode 5  
(User boot mode)

Mode 7  
(Single-chip activation  
expanded mode,  
with on-chip ROM enabled)



- Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.  
 2. When EXPE = 1, external address space; when EXPE = 0, reserved area.  
 3. When EXPE = 1, external address space with RAME = 0, on-chip RAM with RAME = 1.  
 When EXPE = 0, on-chip RAM.  
 4. A reserved area should not be accessed.  
 5. The on-chip RAM is used to program the flash memory. The RAME bit in SYSCR should not be cleared to 0.

**Figure 3.2 H8S/2368F Memory Map (2)**

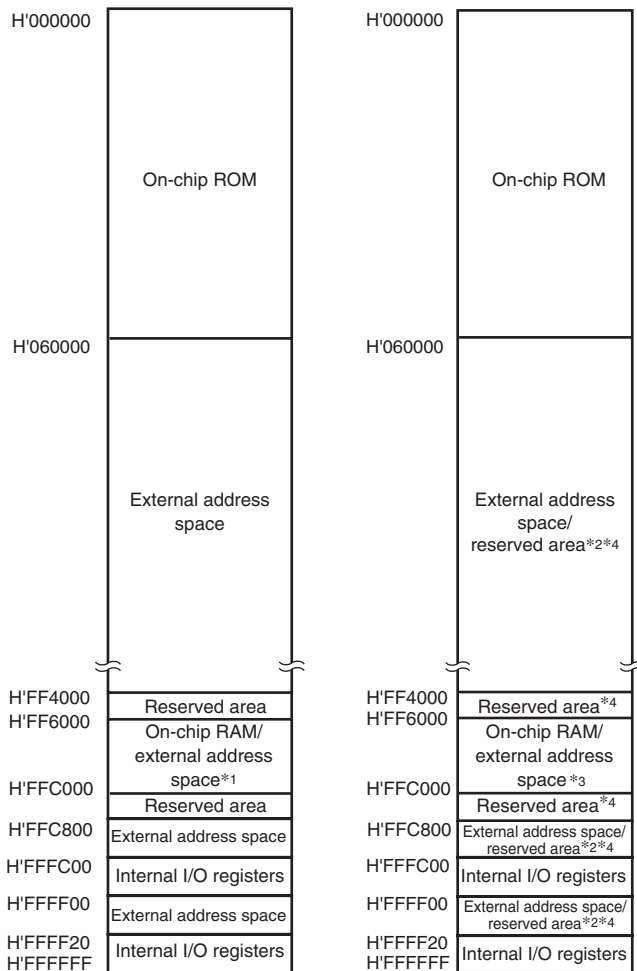


- Notes:
1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.
  2. When EXPE = 1, external address space; when EXPE = 0, reserved area.
  3. On-chip RAM is used for flash memory programming. Do not clear the RAME bit in SYSCR to 0.
  4. Do not access a reserved area.

**Figure 3.3 H8S/2367F Memory Map (1)**

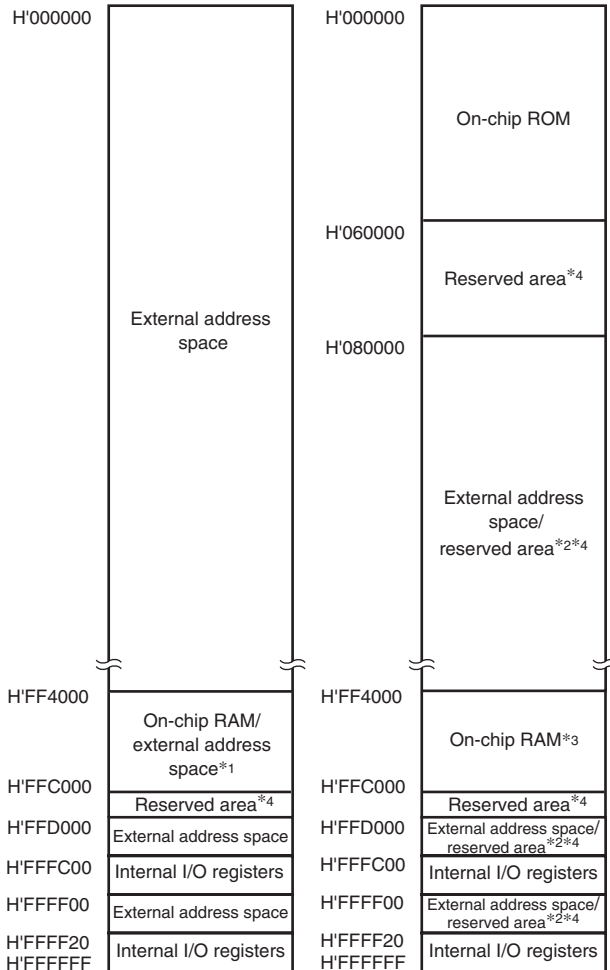
Mode 4  
(Expanded mode with  
on-chip ROM enabled)

Mode 7  
(Single-chip activation  
expanded mode,  
with on-chip ROM enabled)



- Notes:
1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.
  2. When EXPE = 1, external address space; when EXPE = 0, reserved area.
  3. When EXPE = 1, external address space with RAME = 0, on-chip RAM with RAME = 1.  
When EXPE = 0, on-chip RAM.
  4. Do not access a reserved area.

**Figure 3.4 H8S/2367F Memory Map (2)**



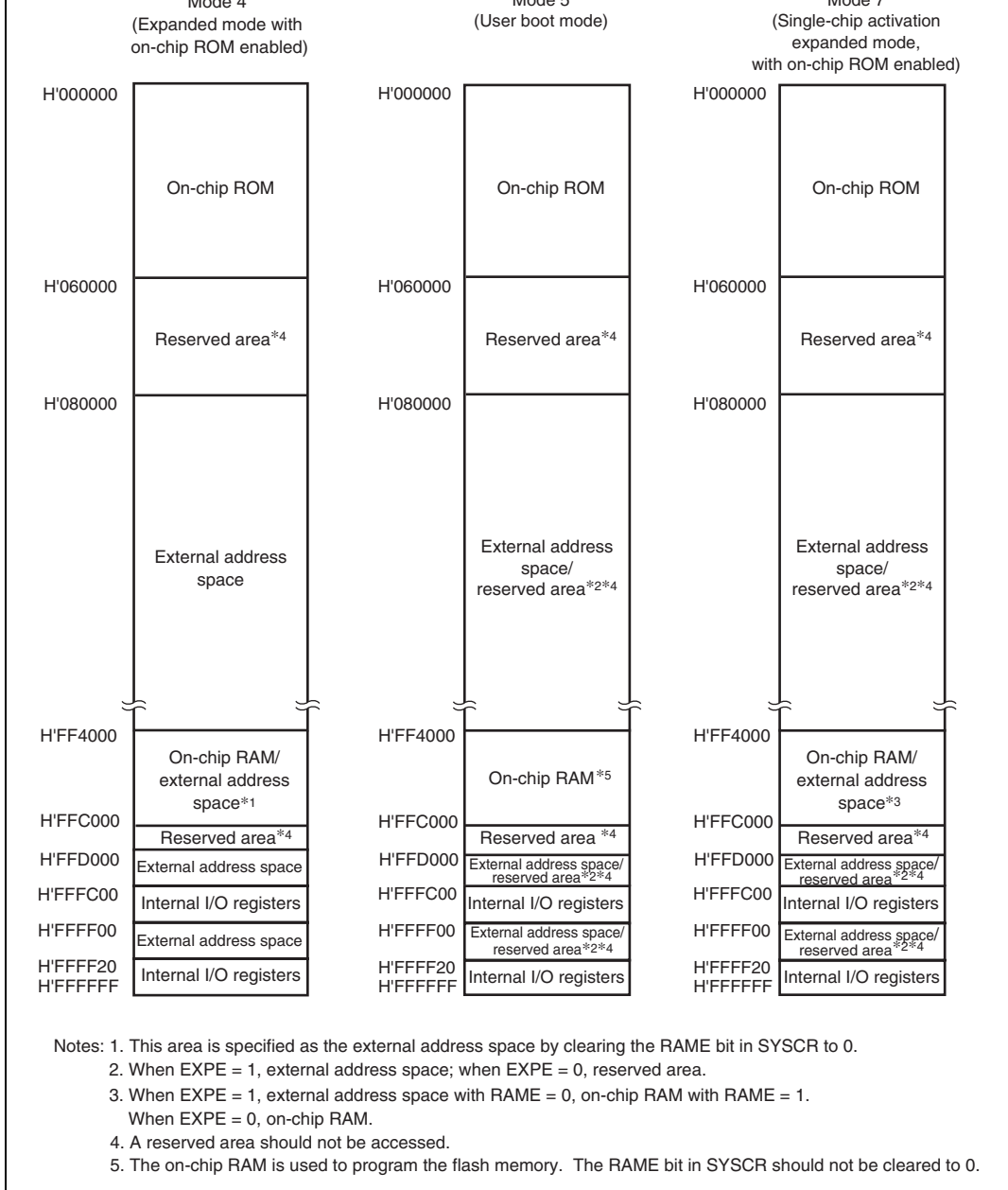
Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.

2. When EXPE = 1, external address space; when EXPE = 0, reserved area.

3. On-chip RAM is used for flash memory programming. The RAME bit in SYSCR should not be cleared to 0.

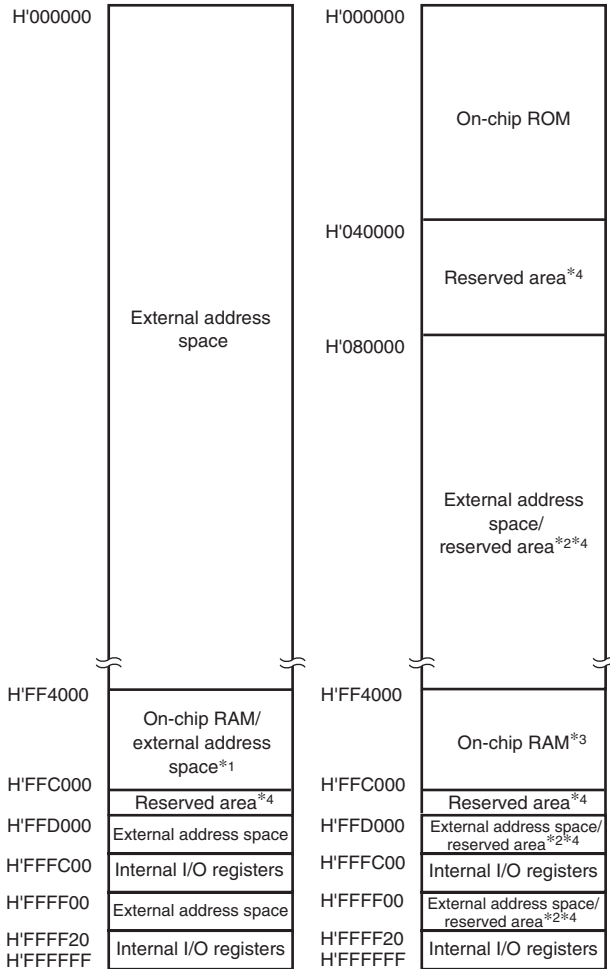
4. A reserved area should not be accessed.

Figure 3.5 H8S/2364F Memory Map (1)



**Figure 3.6 H8S/2364F Memory Map (2)**





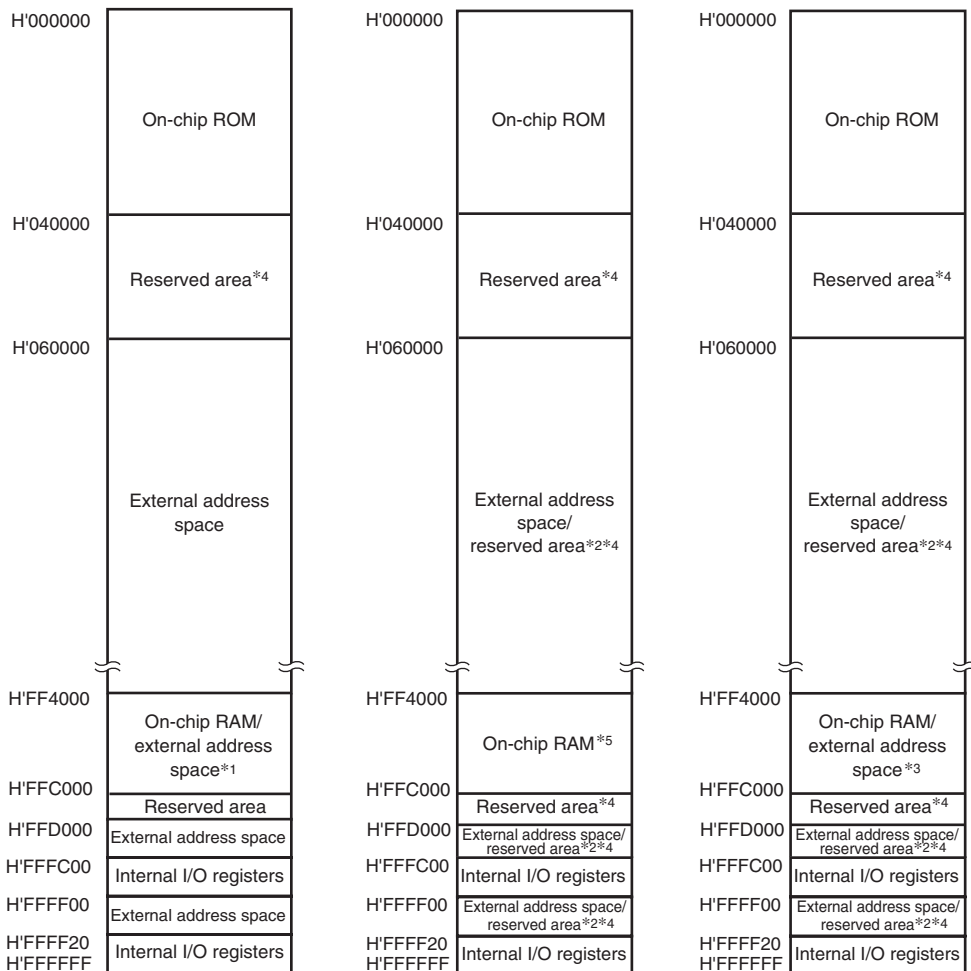
- Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.  
 2. When EXPE = 1, external address space; when EXPE = 0, reserved area.  
 3. On-chip RAM is used for flash memory programming. The RAME bit in SYSCR should not be cleared to 0.  
 4. A reserved area should not be accessed.

Figure 3.7 H8S/2362F Memory Map (1)

Mode 4  
(Expanded mode with  
on-chip ROM enabled)

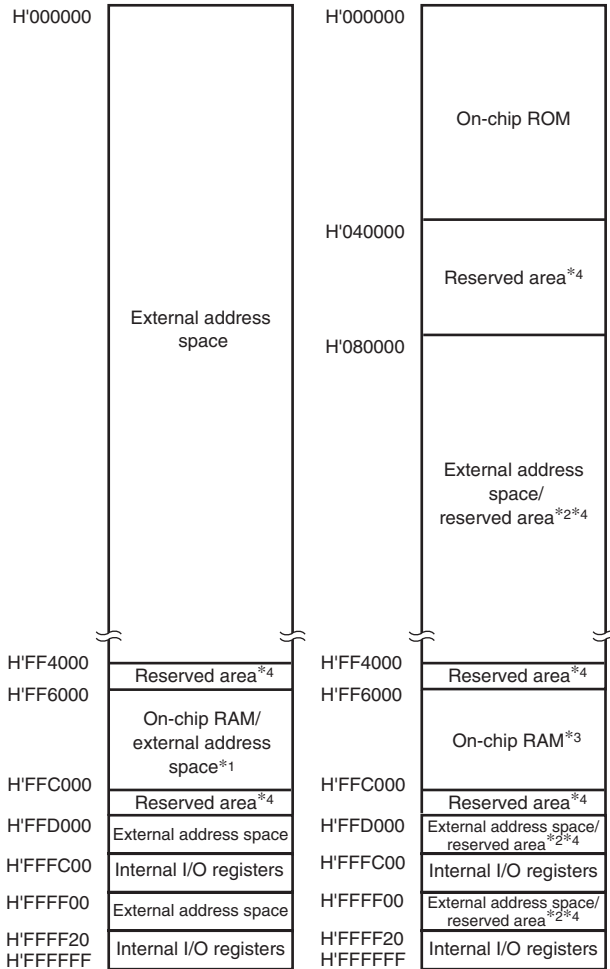
Mode 5  
(User boot mode)

Mode 7  
(Single-chip activation  
expanded mode,  
with on-chip ROM enabled)



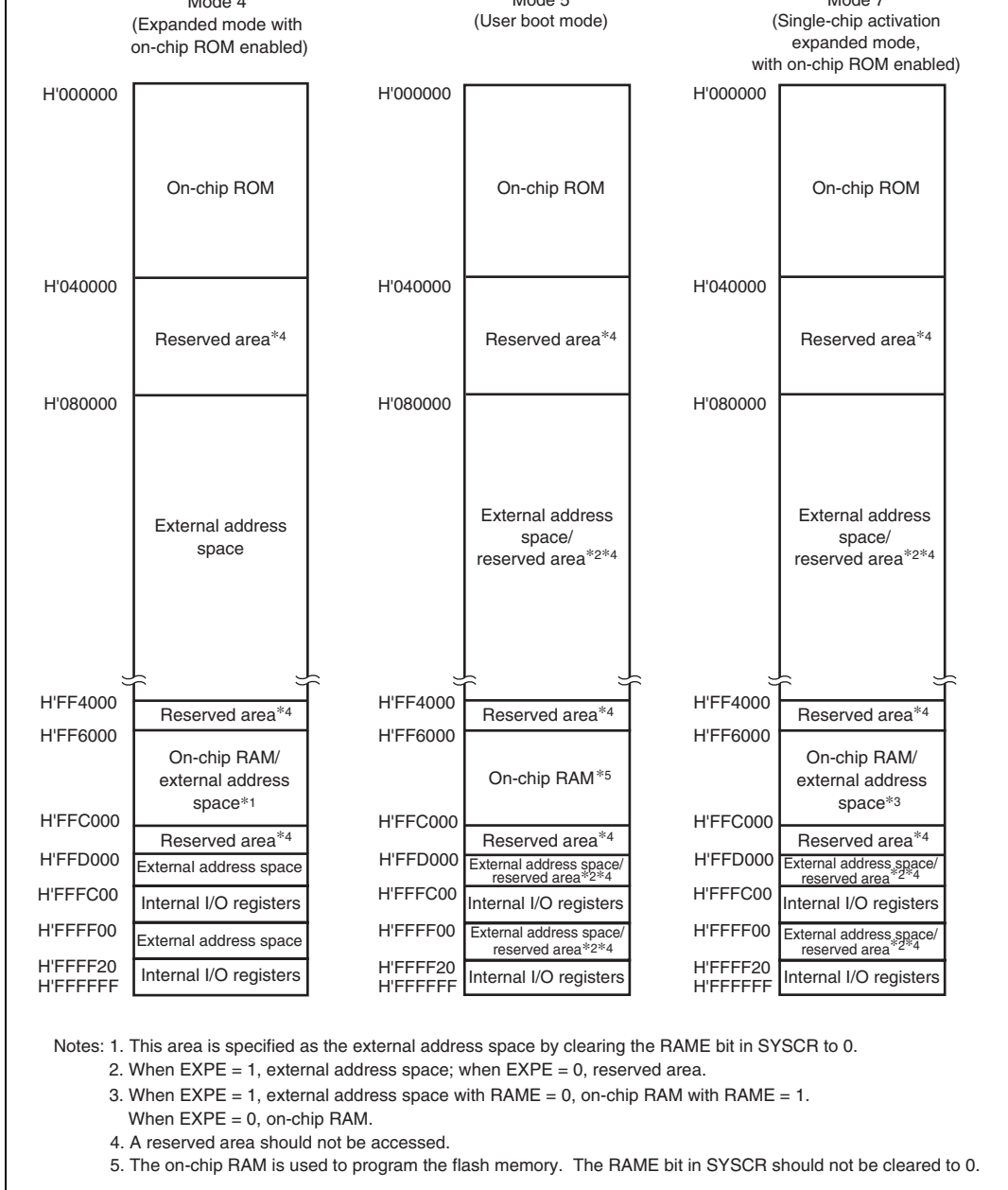
- Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.  
 2. When EXPE = 1, external address space; when EXPE = 0, reserved area.  
 3. When EXPE = 1, external address space with RAME = 0, on-chip RAM with RAME = 1.  
 When EXPE = 0, on-chip RAM.  
 4. A reserved area should not be accessed.  
 5. The on-chip RAM is used to program the flash memory. The RAME bit in SYSCR should not be cleared to 0.

**Figure 3.8 H8S/2362F Memory Map (2)**

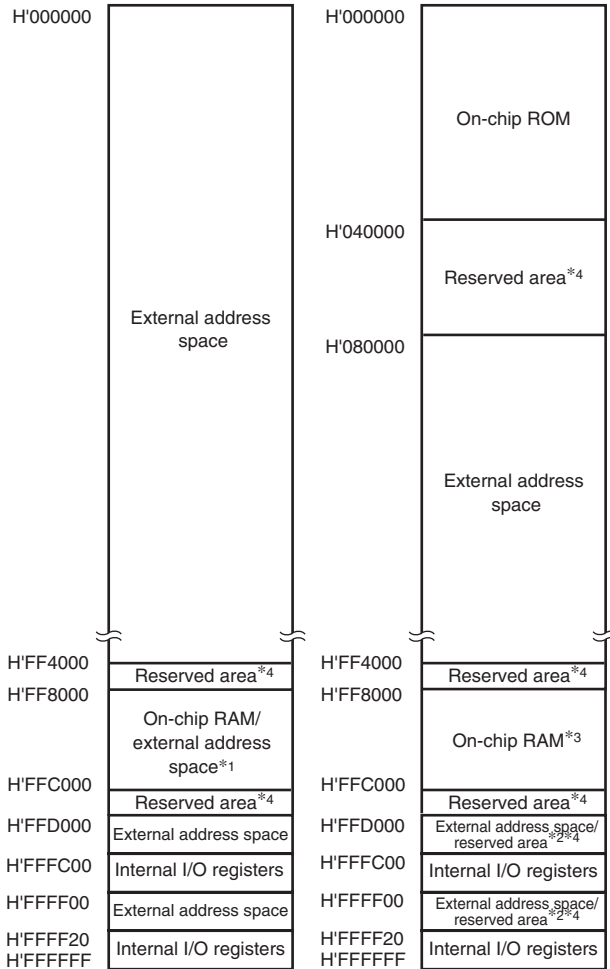


- Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.  
 2. When EXPE = 1, external address space; when EXPE = 0, reserved area.  
 3. On-chip RAM is used for flash memory programming. The RAME bit in SYSCR should not be cleared to 0.  
 4. A reserved area should not be accessed.

Figure 3.9 H8S/2361F Memory Map (1)



**Figure 3.10 H8S/2361F Memory Map (2)**



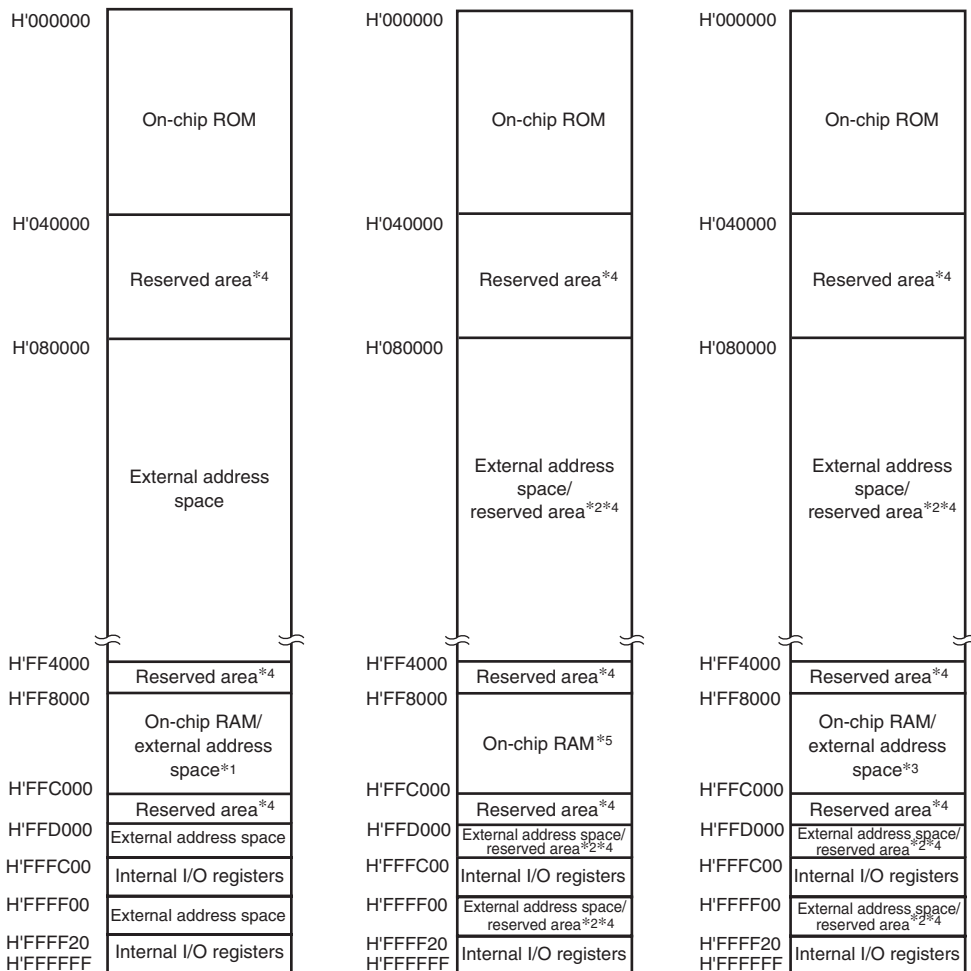
- Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.  
 2. When EXPE = 1, external address space; when EXPE = 0, reserved area.  
 3. On-chip RAM is used for flash memory programming. The RAME bit in SYSCR should not be cleared to 0.  
 4. A reserved area should not be accessed.

Figure 3.11 H8S/2360F Memory Map (1)

Mode 4  
(Expanded mode with  
on-chip ROM enabled)

Mode 5  
(User boot mode)

Mode 7  
(Single-chip activation  
expanded mode,  
with on-chip ROM enabled)

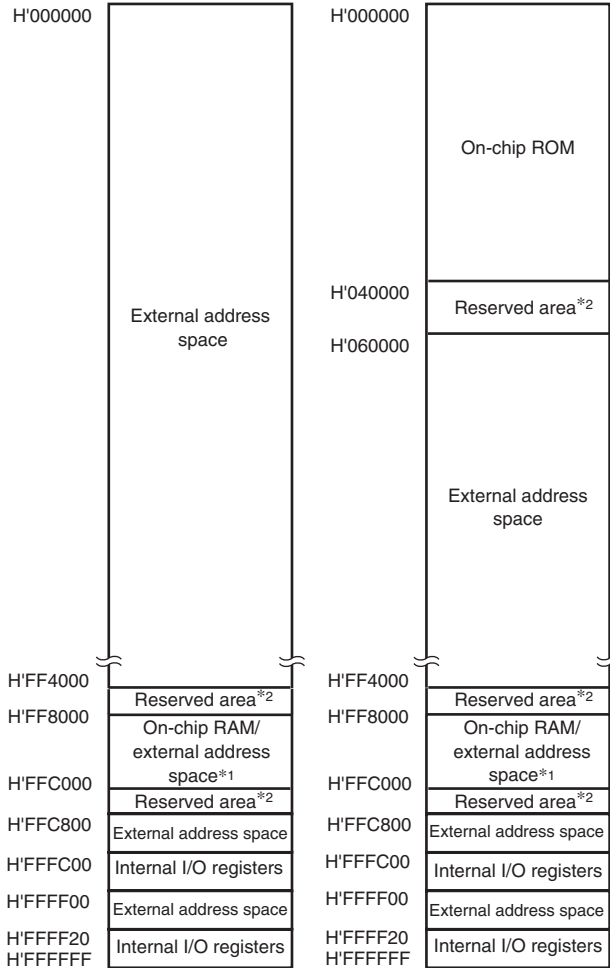


- Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.  
 2. When EXPE = 1, external address space; when EXPE = 0, reserved area.  
 3. When EXPE = 1, external address space with RAME = 0, on-chip RAM with RAME = 1.  
 When EXPE = 0, on-chip RAM.  
 4. A reserved area should not be accessed.  
 5. The on-chip RAM is used to program the flash memory. The RAME bit in SYSCR should not be cleared to 0.

**Figure 3.12 H8S/2360F Memory Map (2)**

Modes 1 and 2  
(Expanded mode with  
on-chip ROM disabled)

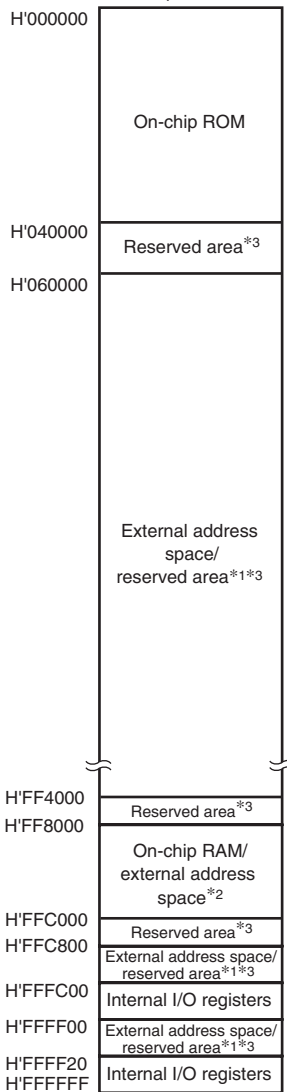
Mode 4  
(Expanded mode with  
on-chip ROM enabled)



Notes: 1. This area is specified as external address space by clearing the RAME bit in SYSCR to 0.  
2. Do not access a reserved area.

**Figure 3.13 H8S/2365 Memory Map (1)**

Mode 7  
 (Single-chip activation expanded mode,  
 with on-chip ROM enabled)

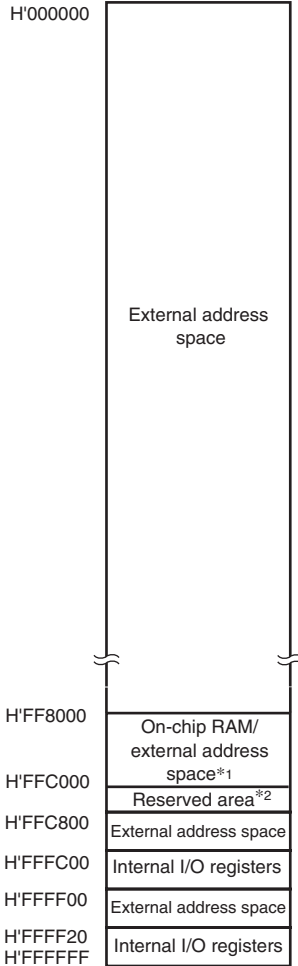


- Notes: 1. When EXPE = 1, external address space; when EXPE = 0, reserved area.  
 2. When EXPE = 1, external address space with RAME = 0, on-chip RAM with RAME = 1.  
 When EXPE = 0, on-chip RAM.  
 3. Do not access a reserved area.

**Figure 3.14 H8S/2365 Memory Map (2)**



(Expanded mode with  
on-chip ROM disabled)



- Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.  
2. Do not access a reserved area.

**Figure 3.15 H8S/2363 Memory Map**



## 4.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trace, interrupt, or trap instruction. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority. Exception sources, the stack structure, and operation of the CPU vary depending on the interrupt control mode. For details on the interrupt control mode, refer to section 5, Interrupt Controller.

**Table 4.1 Exception Types and Priority**

Priority	Exception Type	Start of Exception Handling
High ↑	Reset	Starts immediately after a low-to-high transition at the RES pin, or when the watchdog timer overflows. The CPU enters the reset state when the $\overline{\text{RES}}$ pin is low.
	Trace <sup>*1</sup>	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit in the EXR is set to 1.
	Direct transition <sup>*2</sup>	Starts when the direct transition occurs by execution of the SLEEP instruction.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued. <sup>*3</sup>
Low	Trap instruction <sup>*4</sup>	Started by execution of a trap instruction (TRAPA)

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception handling is not executed after execution of an RTE instruction.  
2. Not available in this LSI.  
3. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.  
4. Trap instruction exception handling requests are accepted at all times in program execution state.

## 4.2 Exception Sources and Exception Vector Table

Different vector addresses are assigned to different exception sources. Table 4.2 lists the exception sources and their vector addresses. Since the usable modes differ depending on the product, for details on each product, refer to section 3, MCU Operating Modes.

Exception Source		Vector Number	Normal Mode* <sup>2</sup>	Advanced Mode
Power-on reset		0	H'0000 to H'0001	H'0000 to H'0003
Manual reset* <sup>3</sup>		1	H'0002 to H'0003	H'0004 to H'0007
Reserved for system use		2	H'0004 to H'0005	H'0008 to H'000B
		3	H'0006 to H'0007	H'000C to H'000F
		4	H'0008 to H'0019	H'0010 to H'0013
Trace		5	H'000A to H'000B	H'0014 to H'0017
Interrupt (direct transition)* <sup>3</sup>		6	H'000C to H'000D	H'0018 to H'001B
Interrupt (NMI)		7	H'000E to H'000F	H'001C to H'001F
Trap instruction (#0)		8	H'0010 to H'0011	H'0020 to H'0023
(#1)		9	H'0012 to H'0013	H'0024 to H'0027
(#2)		10	H'0014 to H'0015	H'0028 to H'002B
(#3)		11	H'0016 to H'0017	H'002C to H'002F
Reserved for system use		12	H'0018 to H'0019	H'0030 to H'0033
		13	H'001A to H'001B	H'0034 to H'0037
		14	H'001C to H'001D	H'0038 to H'003B
		15	H'001E to H'001F	H'003C to H'003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'0040 to H'0043
	IRQ1	17	H'0022 to H'0023	H'0044 to H'0047
	IRQ2	18	H'0024 to H'0025	H'0048 to H'004B
	IRQ3	19	H'0026 to H'0027	H'004C to H'004F
	IRQ4	20	H'0028 to H'0029	H'0050 to H'0053
	IRQ5	21	H'002A to H'002B	H'0054 to H'0057
	IRQ6	22	H'002C to H'002D	H'0058 to H'005B
	IRQ7	23	H'002E to H'002F	H'005C to H'005F
Reserved for system use		24	H'0030 to H'0031	H'0060 to H'0063
		25	H'0032 to H'0033	H'0064 to H'0067
		26	H'0034 to H'0035	H'0068 to H'006B
		27	H'0036 to H'0037	H'006C to H'006F
		28	H'0038 to H'0039	H'0070 to H'0073
		29	H'003A to H'003B	H'0074 to H'0077

Reserved for system use	30	H'003C to H'003D	H'0078 to H'007B
	31	H'003E to H'003F	H'007C to H'007F
Internal interrupt* <sup>4</sup>	32	H'0040 to H'0041	H'0080 to H'0083
	118	H'00EC to H'00ED	H'01D8 to H'01DB

- Notes:
1. Lower 16 bits of the address.
  2. Not available in this LSI.
  3. Not available in this LSI. It is reserved for system use.
  4. For details of internal interrupt vectors, see section 5.5, Interrupt Exception Handling Vector Table.

## 4.3 Reset

A reset has the highest exception priority. When the  $\overline{\text{RES}}$  pin goes low, all processing halts and this LSI enters the reset. To ensure that this LSI is reset, hold the  $\overline{\text{RES}}$  pin low for at least 20 ms at power-up. To reset the chip during operation, hold the  $\overline{\text{RES}}$  pin low for at least 20 states. A reset initializes the internal state of the CPU and the registers of on-chip peripheral modules.

The chip can also be reset by overflow of the watchdog timer. For details see section 13, Watchdog Timer.

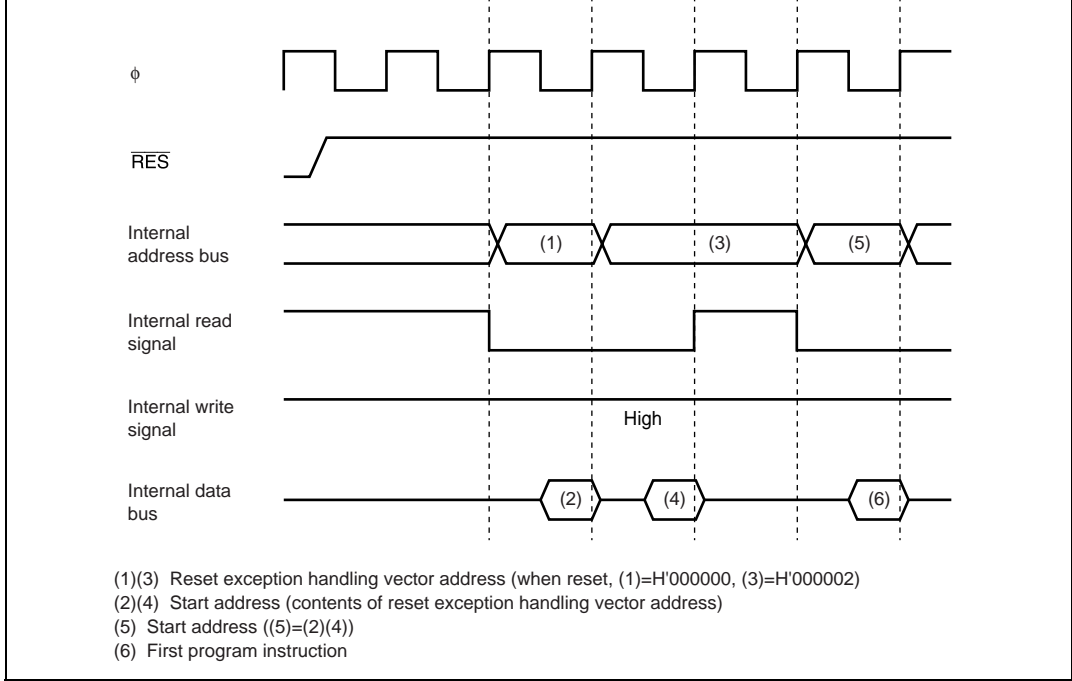
The interrupt control mode is 0 immediately after reset.

### 4.3.1 Reset Exception Handling

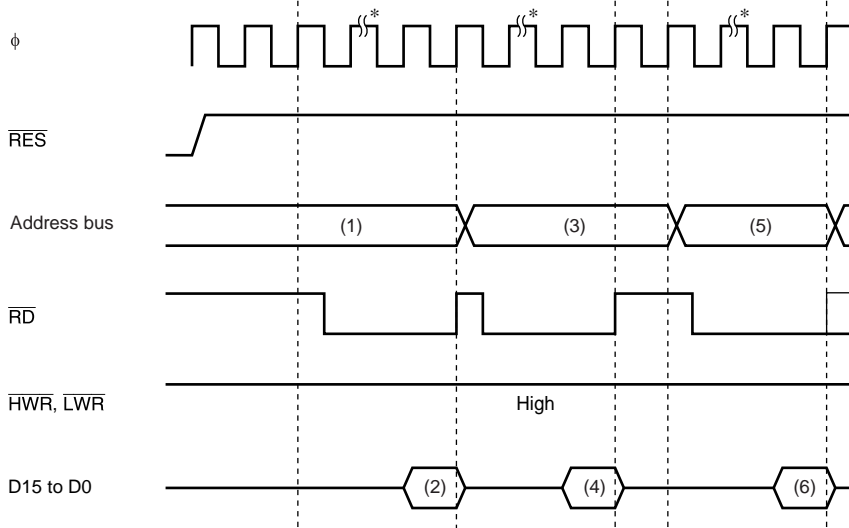
When the  $\overline{\text{RES}}$  pin goes high after being held low for the necessary time, this LSI starts reset exception handling as follows:

1. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized, the T bit is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR.
2. The reset exception handling vector address is read and transferred to the PC, and program execution starts from the address indicated by the PC.

Figures 4.1 and 4.2 show examples of the reset sequence.



**Figure 4.1 Reset Sequence (Advanced Mode with On-Chip ROM Enabled)**



- (1)(3) Reset exception handling vector address (when reset, (1)=H'000000, (3)=H'000002)
- (2)(4) Start address (contents of reset exception handling vector address)
- (5) Start address ((5)=(2)(4))
- (6) First program instruction

Note: \* Seven program wait states are inserted.

**Figure 4.2 Reset Sequence (Advanced Mode with On-Chip ROM Disabled)**

### 4.3.2 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.L #xx: 32, SP).

### 4.3.3 On-Chip Peripheral Functions after Reset Release

After reset release, MSTPCR is initialized to H'0FFF and EXMSTPCR is to H'FFFD, all modules except the DMAC and the DTC enter module stop mode.

## 4.4 Traces

Traces are enabled in interrupt control mode 2. Trace mode is not activated in interrupt control mode 0, irrespective of the state of the T bit. For details on interrupt control modes, see section 5, Interrupt Controller.

If the T bit in EXR is set to 1, trace mode is activated. In trace mode, a trace exception occurs on completion of each instruction. Trace mode is not affected by interrupt masking. Table 4.3 shows the state of CCR and EXR after execution of trace exception handling. Trace mode is canceled by clearing the T bit in EXR to 0. The T bit saved on the stack retains its value of 1, and when control is returned from the trace exception handling routine by the RTE instruction, trace mode resumes. Trace exception handling is not carried out after execution of the RTE instruction.

Interrupts are accepted even within the trace exception handling routine.

**Table 4.3 Status of CCR and EXR after Trace Exception Handling**

Interrupt Control Mode	CCR		EXR	
	I	UI	I2 to I0	T
0	Trace exception handling cannot be used.			
2	1	—	—	0

Legend:

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

## 4.5 Interrupts

Interrupts are controlled by the interrupt controller. The interrupt controller has two interrupt control modes and can assign interrupts other than NMI to eight priority/mask levels to enable multiplexed interrupt control. The source to start interrupt exception handling and the vector address differ depending on the product. For details, refer to section 5, Interrupt Controller.

The interrupt exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved in the stack.



from the vector table to the PC, and program execution starts from that address.

## 4.6 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

The trap instruction exception handling is as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended register (EXR) are saved in the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4.4 shows the status of CCR and EXR after execution of trap instruction exception handling.

**Table 4.4 Status of CCR and EXR after Trap Instruction Exception Handling**

Interrupt Control Mode	CCR		EXR	
	I	UI	I2 to I0	T
0	1	—	—	—
2	1	—	—	0

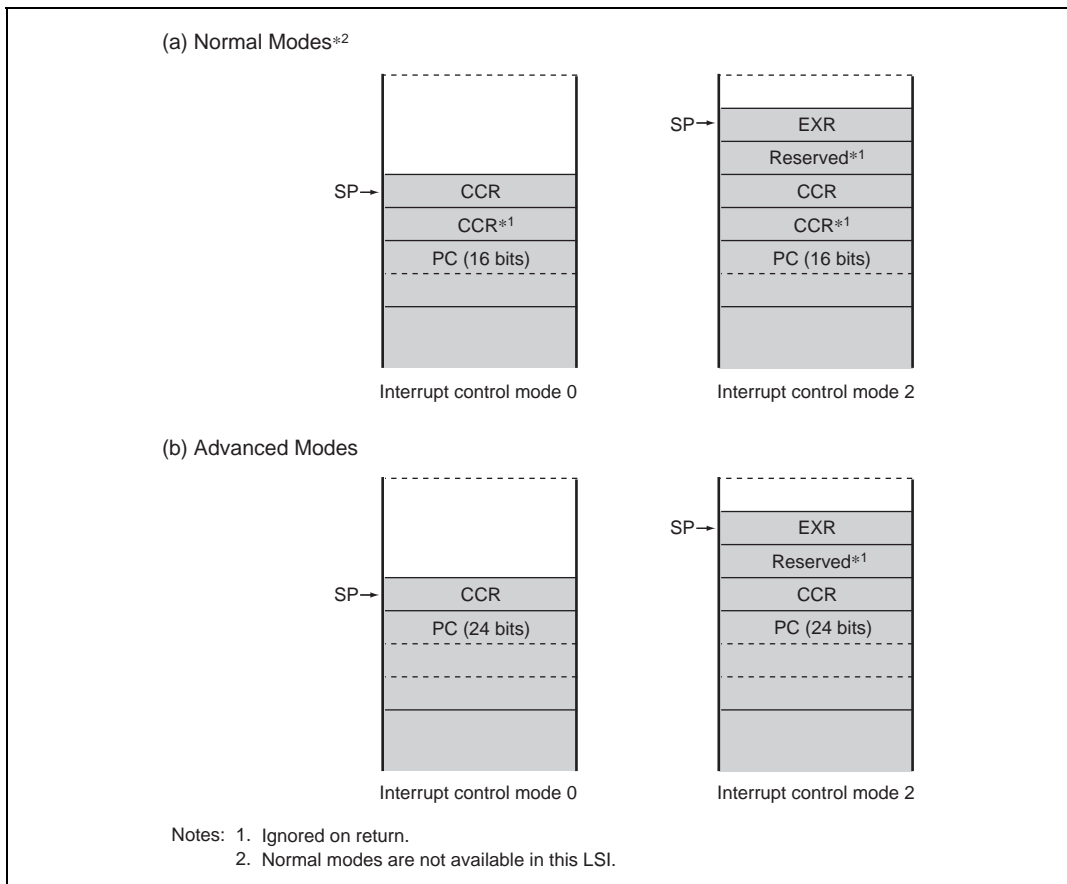
Legend:

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

Figure 4.3 shows the stack after completion of trap instruction exception handling and interrupt exception handling.



**Figure 4.3 Stack Status after Exception Handling**

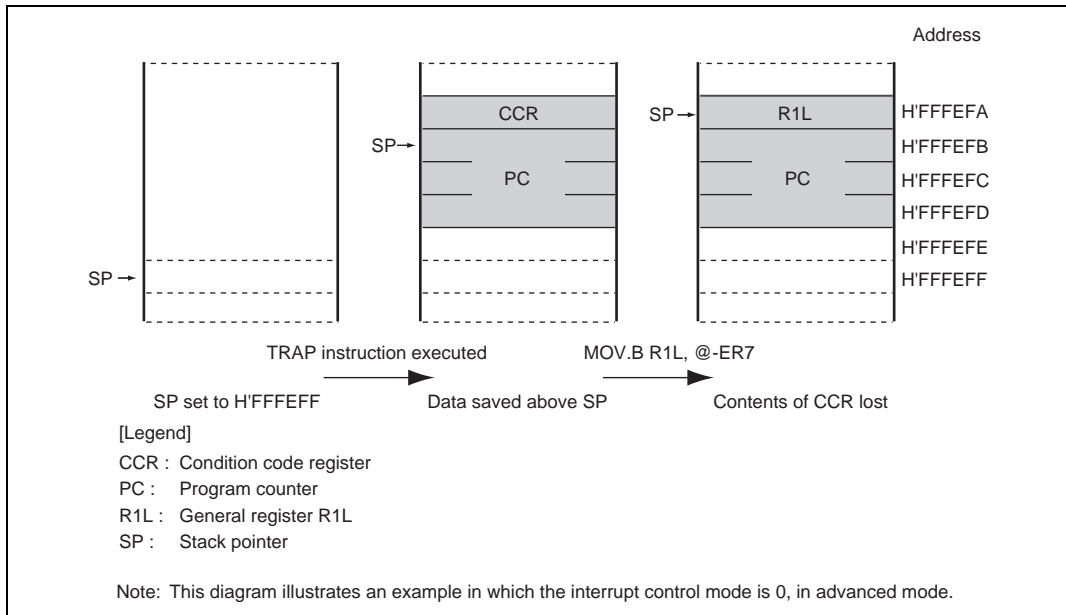
When accessing word data or longword data, this LSI assumes that the lowest address bit is 0. The stack should always be accessed by word transfer instruction or longword transfer instruction, and the value of the stack pointer (SP, ER7) should always be kept even. Use the following instructions to save registers:

```
PUSH.W   Rn      (or MOV.W Rn,  @-SP)
PUSH.L   ERn     (or MOV.L ERn,  @-SP)
```

Use the following instructions to restore registers:

```
POP.W    Rn      (or MOV.W @SP+, Rn)
POP.L    ERn     (or MOV.L @SP+, ERn)
```

Setting SP to an odd value may lead to a malfunction. Figure 4.4 shows an example of operation when the SP value is odd.

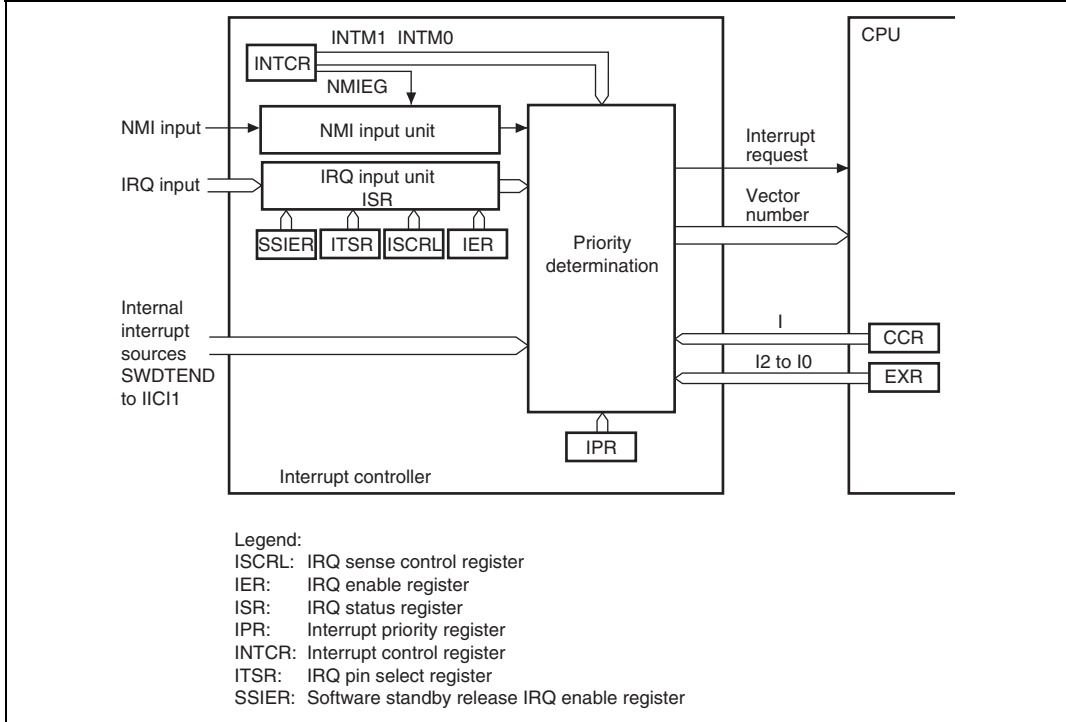


**Figure 4.4 Operation when SP Value Is Odd**



## 5.1 Features

- Two interrupt control modes  
Any of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the interrupt control register (INTCR).
- Priorities settable with IPR  
An interrupt priority register (IPR) is provided for setting interrupt priorities. Eight priority levels can be set for each module for all interrupts except NMI. NMI is assigned the highest priority level of 8, and can be accepted at all times.
- Independent vector addresses  
All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Nine external interrupts  
NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge can be selected for NMI. Falling-edge, rising-edge, or both-edge detection, or level sensing, can be selected for  $\overline{\text{IRQ7}}$  to  $\overline{\text{IRQ0}}$ .
- DTC and DMAC control  
DTC and DMAC activations are performed by means of interrupts.



**Figure 5.1 Block Diagram of Interrupt Controller**

Table 5.1 summarizes the pins of the interrupt controller.

**Table 5.1 Pin Configuration**

<b>Name</b>	<b>I/O</b>	<b>Function</b>
NMI	Input	Nonmaskable external interrupt Rising or falling edge can be selected.
$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	Input	Maskable external interrupts Rising edge, falling edge, both edges, or level sensing, can be selected.

### 5.3 Register Descriptions

The interrupt controller has the following registers.

- Interrupt control register (INTCR)
- IRQ sense control register L (ISCRL)
- IRQ enable register (IER)
- IRQ status register (ISR)
- IRQ pin select register (ITSR)
- Software standby release IRQ enable register (SSIER)
- Interrupt priority register A (IPRA)
- Interrupt priority register B (IPRB)
- Interrupt priority register C (IPRC)
- Interrupt priority register D (IPRD)
- Interrupt priority register E (IPRE)
- Interrupt priority register F (IPRF)
- Interrupt priority register G (IPRG)
- Interrupt priority register H (IPRH)
- Interrupt priority register I (IPRI)
- Interrupt priority register J (IPRJ)
- Interrupt priority register K (IPRK)

INTCR selects the interrupt control mode, and the detected edge for NMI.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	INTM1	0	R/W	Interrupt Control Select Mode 1 and 0
4	INTM0	0	R/W	These bits select either of two interrupt control modes for the interrupt controller. 00: Interrupt control mode 0 Interrupts are controlled by I bit. 01: Setting prohibited. 10: Interrupt control mode 2 Interrupts are controlled by bits I2 to I0, and IPR. 11: Setting prohibited.
3	NMIEG	0	R/W	NMI Edge Select Selects the input edge for the NMI pin. 0: Interrupt request generated at falling edge of NMI input 1: Interrupt request generated at rising edge of NMI input
2 to 0	—	0	—	Reserved These bits are always read as 0 and cannot be modified.



IPR are eleven 16-bit readable/writable registers that set priorities (levels 7 to 0) for interrupts other than NMI.

The correspondence between interrupt sources and IPR settings is shown in table 5.2. Setting a value in the range from H'0 to H'7 in the 3-bit groups of bits 14 to 12, 10 to 8, 6 to 4, and 2 to 0 sets the priority of the corresponding interrupt. IPR should be read in word size.

Bit	Bit Name	Initial Value	R/W	Description
15	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
14	IPR14	1	R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
13	IPR13	1	R/W	
12	IPR12	1	R/W	
11	—	0	—	
10	IPR10	1	R/W	Sets the priority of the corresponding interrupt source. 000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
9	IPR9	1	R/W	
8	IPR8	1	R/W	
7	—	0	—	

5	IPR5	1	R/W	source.
4	IPR4	1	R/W	000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)
3	—	0	—	Reserved This bit is always read as 0 and cannot be modified.
2	IPR2	1	R/W	Sets the priority of the corresponding interrupt source.
1	IPR1	1	R/W	
0	IPR0	1	R/W	000: Priority level 0 (Lowest) 001: Priority level 1 010: Priority level 2 011: Priority level 3 100: Priority level 4 101: Priority level 5 110: Priority level 6 111: Priority level 7 (Highest)

IER controls enabling and disabling of interrupt requests IRQ7 to IRQ0.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	–	All 0	R/W	Reserved The write value should always be 0.
7	IRQ7E	0	R/W	IRQ7 Enable The IRQ7 interrupt request is enabled when this bit is 1.
6	IRQ6E	0	R/W	IRQ6 Enable The IRQ6 interrupt request is enabled when this bit is 1.
5	IRQ5E	0	R/W	IRQ5 Enable The IRQ5 interrupt request is enabled when this bit is 1.
4	IRQ4E	0	R/W	IRQ4 Enable The IRQ4 interrupt request is enabled when this bit is 1.
3	IRQ3E	0	R/W	IRQ3 Enable The IRQ3 interrupt request is enabled when this bit is 1.
2	IRQ2E	0	R/W	IRQ2 Enable The IRQ2 interrupt request is enabled when this bit is 1.
1	IRQ1E	0	R/W	IRQ1 Enable The IRQ1 interrupt request is enabled when this bit is 1.
0	IRQ0E	0	R/W	IRQ0 Enable The IRQ0 interrupt request is enabled when this bit is 1.

ISCR1 select the source that generates an interrupt request at pins  $\overline{\text{IRQ7}}$  to  $\overline{\text{IRQ0}}$ .

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ7SCB	0	R/W	IRQ7 Sense Control B
14	IRQ7SCA	0	R/W	IRQ7 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ7}}$ input low level
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ7}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ7}}$ input
13	IRQ6SCB	0	R/W	IRQ6 Sense Control B
12	IRQ6SCA	0	R/W	IRQ6 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ6}}$ input low level
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ6}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ6}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ6}}$ input
11	IRQ5SCB	0	R/W	IRQ5 Sense Control B
10	IRQ5SCA	0	R/W	IRQ5 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ5}}$ input low level
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ5}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ5}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ5}}$ input

8	IRQ4SCA	0	R/W	IRQ4 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ4}}$ input low level
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ4}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ4}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ4}}$ input
7	IRQ3SCB	0	R/W	IRQ3 Sense Control B
6	IRQ3SCA	0	R/W	IRQ3 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ3}}$ input low level
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ3}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ3}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ3}}$ input
5	IRQ2SCB	0	R/W	IRQ2 Sense Control B
4	IRQ2SCA	0	R/W	IRQ2 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ2}}$ input low level
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ2}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ2}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ2}}$ input

2	IRQ1SCA	0	R/W	IRQ1 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ1}}$ input low level
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ1}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ1}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ1}}$ input
1	IRQ0SCB	0	R/W	IRQ0 Sense Control B
0	IRQ0SCA	0	R/W	IRQ0 Sense Control A
				00: Interrupt request generated at $\overline{\text{IRQ0}}$ input low level
				01: Interrupt request generated at falling edge of $\overline{\text{IRQ0}}$ input
				10: Interrupt request generated at rising edge of $\overline{\text{IRQ0}}$ input
				11: Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ0}}$ input

ISR is an IRQ7 to IRQ0 interrupt request flag register.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	–	All 0	R/W	Reserved The write value should always be 0.
7	IRQ7F	0	R/(W)*	[Setting condition]
6	IRQ6F	0	R/(W)*	When the interrupt source selected by ISCR occurs
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	[Clearing conditions]
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	<ul style="list-style-type: none"> <li>• Cleared by reading IRQnF flag when <math>IRQnF = 1</math>, then writing 0 to IRQnF flag</li> <li>• When interrupt exception handling is executed when low-level detection is set and <math>\overline{IRQn}</math> input is high</li> <li>• When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set</li> <li>• When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0</li> </ul>
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	R/(W)*	

Note: \* Only 0 can be written, to clear the flag.

ITSR selects input pins  $\overline{IRQ7}$  to  $\overline{IRQ0}$ .

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	–	All 0	R/W	Reserved The write value should always be 0.
7	ITS7	0	R/W	Selects $\overline{IRQ7}$ input pin. 0: PA7 1: P47
6	ITS6	0	R/W	Selects $\overline{IRQ6}$ input pin. 0: PA6 1: P46
5	ITS5	0	R/W	Selects $\overline{IRQ5}$ input pin. 0: PA5 1: P45
4	ITS4	0	R/W	Selects $\overline{IRQ4}$ input pin. 0: PA4 1: P44
3	ITS3	0	R/W	Selects $\overline{IRQ3}$ input pin. 0: P53 1: P43
2	ITS2	0	R/W	Selects $\overline{IRQ2}$ input pin. 0: P52 1: P42
1	ITS1	0	R/W	Selects $\overline{IRQ1}$ input pin. 0: P51 1: P41
0	ITS0	0	R/W	Selects $\overline{IRQ0}$ input pin. 0: P50 1: P40



SSI $\overline{ER}$  selects the  $\overline{IRQ}$  pins used to recover from the software standby state.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R/W	Reserved The write value should always be 0.
7	SSI7	0	R/W	Software Standby Release IRQ Setting
6	SSI6	0	R/W	These bits select the $\overline{IRQn}$ pins used to recover from the software standby state.
5	SSI5	0	R/W	
4	SSI4	0	R/W	0: $IRQn$ requests are not sampled in the software standby state (Initial value when n = 7 to 3)
3	SSI3	0	R/W	
2	SSI2	0	R/W	
1	SSI1	0	R/W	1: When an $IRQn$ request occurs in the software standby state, the chip recovers from the software standby state after the elapse of the oscillation settling time (Initial value when n = 2 to 0)
0	SSI0	0	R/W	

## 5.4 Interrupt Sources

### 5.4.1 External Interrupts

There are nine external interrupts: NMI and  $IRQ7$  to  $IRQ0$ . These interrupts can be used to restore the chip from software standby mode.

**NMI Interrupt:** Nonmaskable interrupt request (NMI) is the highest-priority interrupt, and is always accepted by the CPU regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in INTCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

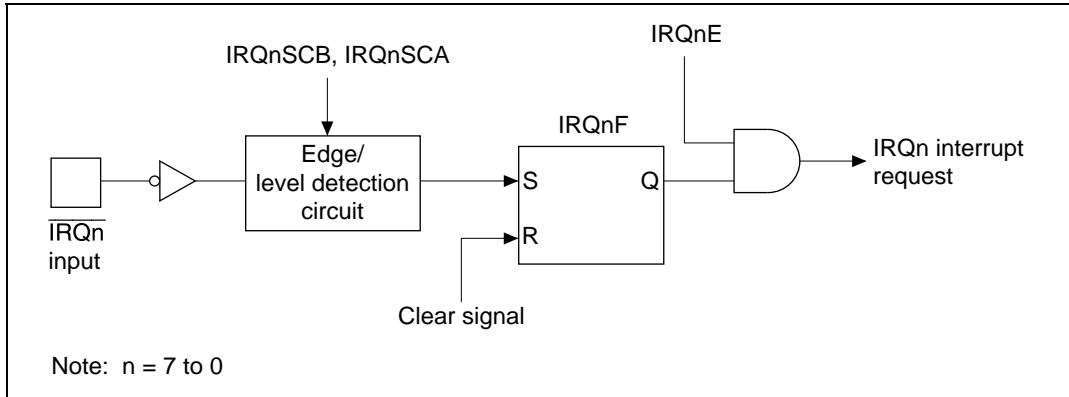
**$IRQ7$  to  $IRQ0$  Interrupts:** Interrupts  $IRQ7$  to  $IRQ0$  are requested by an input signal at pins  $\overline{IRQ7}$  to  $\overline{IRQ0}$ . Interrupts  $IRQ7$  to  $IRQ0$  have the following features:

- Using ISCRL, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins  $\overline{IRQ7}$  to  $\overline{IRQ0}$ .
- Enabling or disabling of interrupt requests  $IRQ7$  to  $IRQ0$  can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of interrupt requests  $IRQ7$  to  $IRQ0$  is indicated in ISR. ISR flags can be cleared to 0 by software.

the interrupt handling routine and clear the IRQnF bit (n = 0 to 7) in ISR to 0. Interrupts may not be executed when the corresponding  $\overline{\text{IRQ}}$  is set to high before the interrupt handling starts.

Detection of IRQ7 to IRQ0 interrupts does not depend on whether the relevant pin has been set for input or output. However, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR to 0 and use the pin as an I/O pin for another function.

A block diagram of interrupts IRQ7 to IRQ0 is shown in figure 5.2.



**Figure 5.2 Block Diagram of Interrupts IRQ15 to IRQ0**

## 5.4.2 Internal Interrupts

The sources for internal interrupts from on-chip peripheral modules have the following features:

- For each on-chip peripheral module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. They can be controlled independently. When the enable bit is set to 1, an interrupt request is issued to the interrupt controller.
- The interrupt priority level can be set by means of IPR.
- The DMAC and DTC can be activated by a TPU, SCI, or other interrupt request.
- When the DMAC or DTC is activated by an interrupt request, it is not affected by the interrupt control mode or CPU interrupt mask bit.

Table 5.2 shows interrupt exception handling sources, vector addresses, and interrupt priorities.

For default priorities, the lower the vector number, the higher the priority. When interrupt control mode 2 is set, priorities among modules can be set by means of the IPR. Modules set at the same priority will conform to their default priorities. Priorities within a module are fixed.

**Table 5.2 Interrupt Sources, Vector Addresses, and Interrupt Priorities**

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*		Priority	DTC Activation	DMAC Activation
			Advanced Mode	IPR			
External pin	NMI	7	H'001C	—	High ↑	—	—
	IRQ0	16	H'0040	IPRA14 to IPRA12		O	—
	IRQ1	17	H'0044	IPRA10 to IPRA8		O	—
	IRQ2	18	H'0048	IPRA6 to IPRA4		O	—
	IRQ3	19	H'004C	IPRA2 to IPRA0		O	—
	IRQ4	20	H'0050	IPRB14 to IPRB12		O	—
	IRQ5	21	H'0054	IPRB10 to IPRB8		O	—
	IRQ6	22	H'0058	IPRB6 to IPRB4		O	—
—	Reserved for system use	23	H'005C	IPRB2 to IPRB0	O	—	
		24	H'0060	IPRC14 to IPRC12	—	—	
		25	H'0064	IPRC10 to IPRC8	—	—	
		26	H'0068	IPRC6 to IPRC4	—	—	
		27	H'006C	IPRC2 to IPRC0	—	—	
		28	H'0070	IPRD14 to IPRD12	—	—	
		29	H'0074	IPRD10 to IPRD8	—	—	
		30	H'0078	IPRD6 to IPRD4	—	—	
		31	H'007C	IPRD2 to IPRD0	—	—	
		DTC	SWDTEND	32	H'0080	IPRE14 to IPRE12	O
WDT	WOVI	33	H'0084	IPRE10 to IPRE8	—	—	
—	Reserved for system use	34	H'0088	IPRE6 to IPRE4	Low	—	—

Interrupt Source	Interrupt Source	Vector Number	Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
Refresh controller	CMI	35	H'008C	IPRE2 to IPRE0	High	—	—
—	Reserved for system use	36	H'0090	IPRF14 to IPRF12	↑	—	—
—		37	H'0094			—	—
A/D	ADI	38	H'0098	IPRF10 to IPRF8		O	O
—	Reserved for system use	39	H'009C	—		—	—
—		—	—			—	—
TPU_0	TGI0A	40	H'00A0	IPRF6 to IPRF4		O	O
	TGI0B	41	H'00A4			O	—
	TGI0C	42	H'00A8			O	—
	TGI0D	43	H'00AC			O	—
	TCI0V	44	H'00B0			—	—
—	Reserved for system use	45	H'00B4	—	—	—	
—		46	H'00B8		—	—	
—		47	H'00BC		—	—	
TPU_1	TGI1A	48	H'00C0	IPRF2 to IPRF0	O	O	
	TGI1B	49	H'00C4		O	—	
	TCI1V	50	H'00C8		—	—	
	TCI1U	51	H'00CC		—	—	
TPU_2	TGI2A	52	H'00D0	IPRG14 to IPRG12	O	O	
	TGI2B	53	H'00D4		O	—	
	TCI2V	54	H'00D8		—	—	
	TCI2U	55	H'00DC		—	—	
TPU_3	TGI3A	56	H'00E0	IPRG10 to IPRG8	O	O	
	TGI3B	57	H'00E4		O	—	
	TGI3C	58	H'00E8		O	—	
	TGI3D	59	H'00EC		O	—	
	TCI3V	60	H'00F0		—	—	
—	Reserved for system use	61	H'00F4	—	—	—	
—		62	H'00F8		—	—	
—		63	H'00FC		—	—	

Interrupt Source	Interrupt Source	Vector Number	Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
TPU_4	TGI4A	64	H'0100	IPRG6 to IPRG4	High	O	O
	TGI4B	65	H'0104			O	—
	TCI4V	66	H'0108			—	—
	TCI4U	67	H'010C			—	—
TPU_5	TGI5A	68	H'0110	IPRG2 to IPRG0		O	O
	TGI5B	69	H'0114			O	—
	TCI5V	70	H'0118			—	—
	TCI5U	71	H'011C			—	—
TMR_0	CMIA0	72	H'0120	IPRH14 to IPRH12		O	—
	CMIB0	73	H'0124			O	—
	OVI0	74	H'0128			—	—
—	Reserved for system use	75	H'012C	IPRH14 to IPRH12		—	—
TMR_1	CMIA1	76	H'0130	IPRH10 to IPRH8		O	—
	CMIB1	77	H'0134			O	—
	OVI1	78	H'0138			—	—
—	Reserved for system use	79	H'013C			—	—
DMAC	DMTEND0A	80	H'0140	IPRH6 to IPRH4		O	—
	DMTEND0B	81	H'0144			O	—
	DMTEND1A	82	H'0148			O	—
	DMTEND1B	83	H'014C			O	—
	Reserved for system use	84	H'0150	IPRH0 to IPRH0		—	—
		85	H'0154	IPRI14 to IPRI12		—	—
		86	H'0158	IPRI10 to IPRI8		—	—
		87	H'015C	IPRI6 to IPRI4	—	—	
SCI_0	ERI0	88	H'0160	IPRI2 to IPRI0		—	—
	RXI0	89	H'0164			O	O
	TXI0	90	H'0168			O	O
	TEI0	91	H'016C			—	—

Interrupt Source	Interrupt Source	Vector Number	Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
SCI_1	ERI1	92	H'0170	IPRJ14 to IPRJ12	High	—	—
	RX11	93	H'0174			O	O
	TX11	94	H'0178			O	O
	TEI1	95	H'017C			—	—
SCI_2	ERI2	96	H'0180	IPRJ10 to IPRJ8		—	—
	RX12	97	H'0184			O	—
	TX12	98	H'0188			O	—
	TEI2	99	H'018C			—	—
SCI_3	ERI3	100	H'0190	IPRJ6 to IPRJ4		—	—
	RX13	101	H'0194			—	—
	TX13	102	H'0198			—	—
	TEI3	103	H'019C			—	—
SCI_4	ERI4	104	H'01A0	IPRJ2 to IPRJ0		—	—
	RX14	105	H'01A4			O	—
	TX14	106	H'01A8			O	—
	TEI4	107	H'01AC			—	—
—	Reserved for system use	108	H'01B0	IPRK14 to IPRK12		—	—
		109	H'01B4			—	—
		110	H'01B8			—	—
		111	H'01BC			—	—
—	Reserved for system use	112	H'01C0	IPRK10 to IPRK8		—	—
		113	H'01C4			—	—
		114	H'01C8			—	—
		115	H'01CC			—	—
IIC2	IIC10	116	H'01D0	IPRK6 to IPRK4		—	—
	Reserved for system use	117	H'01D4			—	—
	IIC11	118	H'01D8			—	—
	Reserved for system use	119	H'01DC			—	—

Interrupt Source	Interrupt Source	Vector Number	Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
—	Reserved for system use	120	H'01E0	IPRK2 to IPRK0	High	—	—
		121	H'01E4		↑	—	—
		122	H'01E8		—	—	—
		123	H'01EC		—	—	—
		124	H'01F0		—	—	—
		125	H'01F4		—	—	—
		126	H'01F8		—	—	—
		127	H'01EC		Low	—	—

Note: \* Lower 16 bits of the start address.

The interrupt controller has two modes: interrupt control mode 0 and interrupt control mode 2. Interrupt operations differ depending on the interrupt control mode. The interrupt control mode is selected by INTCR. Table 5.3 shows the differences between interrupt control mode 0 and interrupt control mode 2.

**Table 5.3 Interrupt Control Modes**

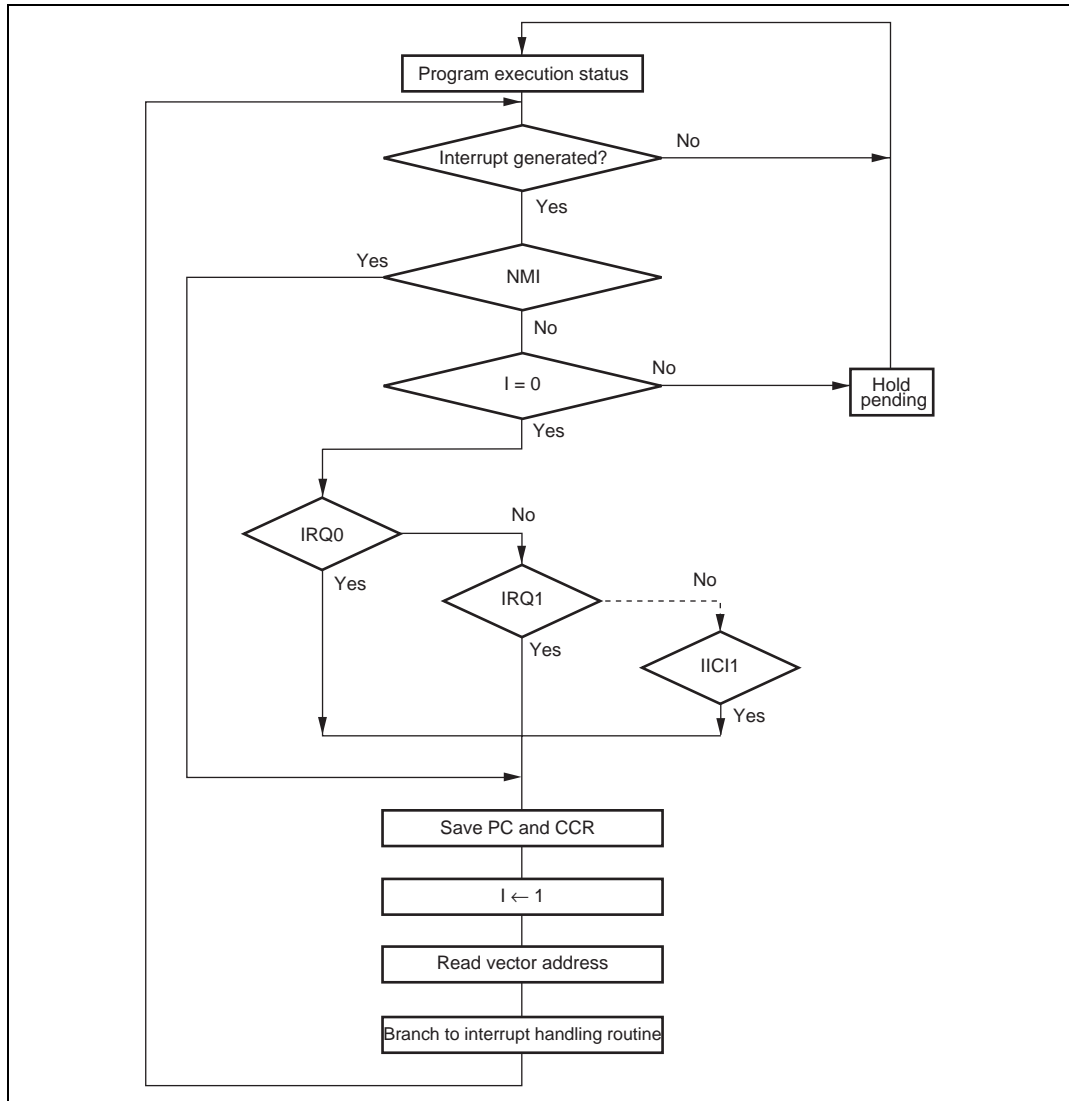
<b>Interrupt Control Mode</b>	<b>Priority Setting Registers</b>	<b>Interrupt Mask Bits</b>	<b>Description</b>
0	Default	I	The priorities of interrupt sources are fixed at the default settings. Interrupt sources except for NMI is masked by the I bit.
2	IPR	I2 to I0	8 priority levels except for NMI can be set with IPR. 8-level interrupt mask control is performed by bits I2 to I0.

### 5.6.1 Interrupt Control Mode 0

In interrupt control mode 0, interrupt requests except for NMI is masked by the I bit of CCR in the CPU. Figure 5.3 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending. If the I bit is cleared, an interrupt request is accepted.
3. Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt according to the priority system is accepted, and other interrupt requests are held pending.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

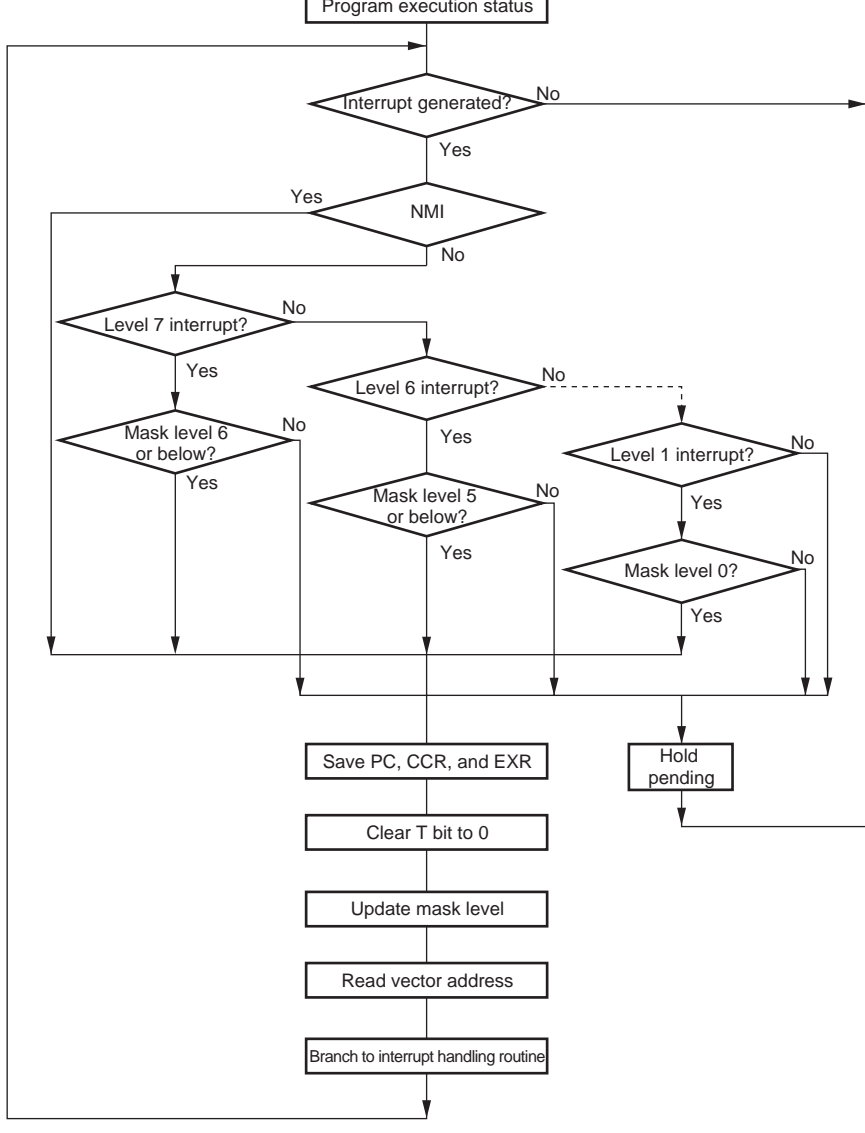




**Figure 5.3 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0**

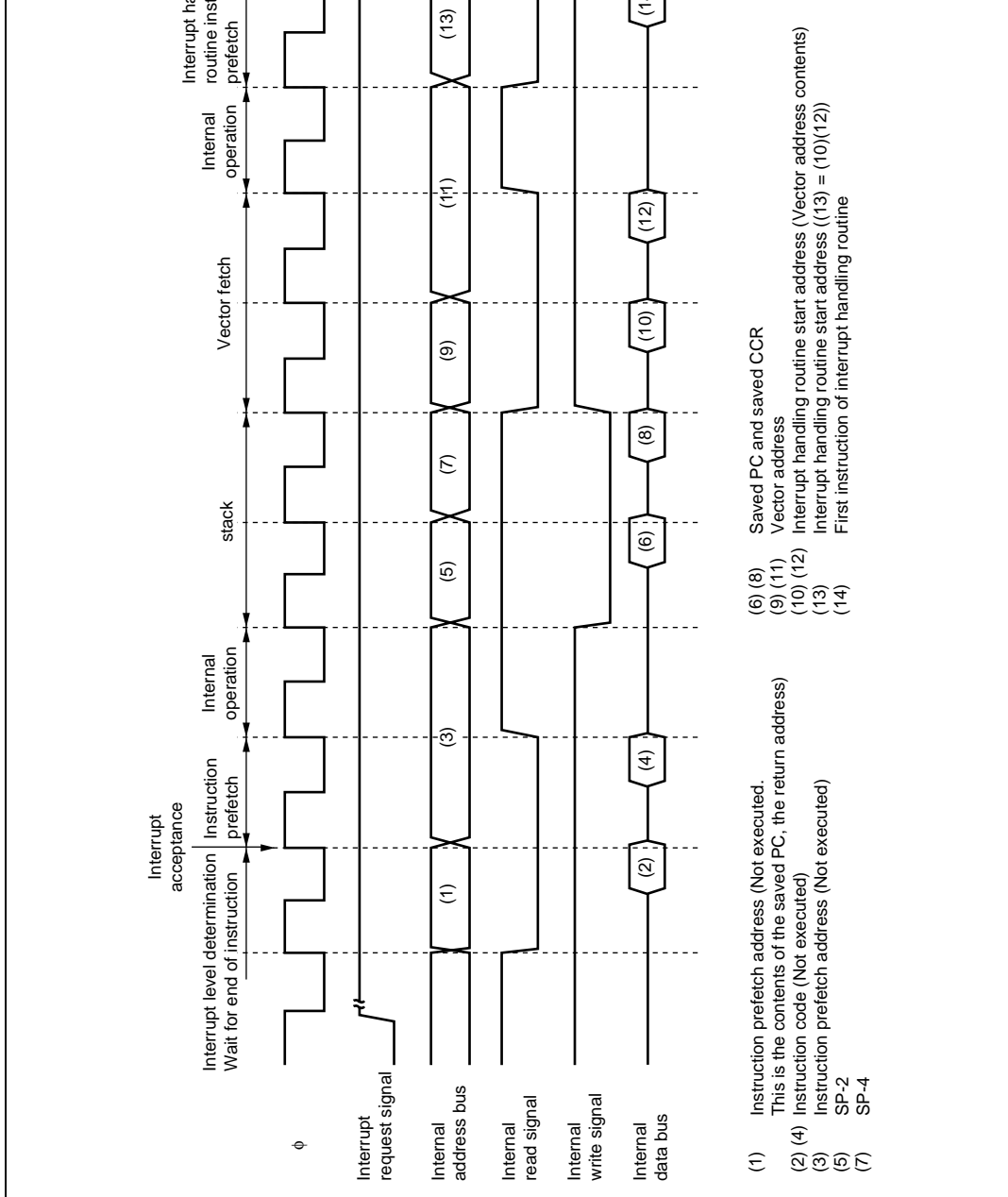
In interrupt control mode 2, mask control is done in eight levels for interrupt requests except for NMI by comparing the EXR interrupt mask level (I2 to I0 bits) in the CPU and the IPR setting. Figure 5.4 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If a number of interrupt requests with the same priority are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.2 is selected.
3. Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level is accepted.
4. When the CPU accepts an interrupt request, it starts interrupt exception handling after execution of the current instruction has been completed.
5. The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt.  
If the accepted interrupt is NMI, the interrupt mask level is set to H'7.
7. The CPU generates a vector address for the accepted interrupt and starts execution of the interrupt handling routine at the address indicated by the contents of the vector address in the vector table.



**Figure 5.4 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2**

Figure 5.5 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0 is set in advanced mode, and the program area and stack area are in on-chip memory.



**Figure 5.5 Interrupt Exception Handling**

Table 5.4 shows interrupt response times - the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine. The execution status symbols used in table 5.4 are explained in table 5.5.

This LSI is capable of fast word transfer to on-chip memory, and have the program area in on-chip ROM and the stack area in on-chip RAM, enabling high-speed processing.

**Table 5.4 Interrupt Response Times**

No.	Execution Status	Normal Mode <sup>*5</sup>		Advanced Mode	
		Interrupt control mode 0	Interrupt control mode 2	Interrupt control mode 0	Interrupt control mode 2
1	Interrupt priority determination <sup>*1</sup>	3	3	3	3
2	Number of wait states until executing instruction ends <sup>*2</sup>	1 to 19 + 2·S <sub>i</sub>	1 to 19 + 2·S <sub>i</sub>	1 to 19 + 2·S <sub>i</sub>	1 to 19 + 2·S <sub>i</sub>
3	PC, CCR, EXR stack save	2·S <sub>k</sub>	3·S <sub>k</sub>	2·S <sub>k</sub>	3·S <sub>k</sub>
4	Vector fetch	S <sub>i</sub>	S <sub>i</sub>	2·S <sub>i</sub>	2·S <sub>i</sub>
5	Instruction fetch <sup>*3</sup>	2·S <sub>i</sub>	2·S <sub>i</sub>	2·S <sub>i</sub>	2·S <sub>i</sub>
6	Internal processing <sup>*4</sup>	2	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32	13 to 33

Notes: 1. Two states in case of internal interrupt.

2. Refers to MULXS and DIVXS instructions.

3. Prefetch after interrupt acceptance and interrupt handling routine prefetch.

4. Internal processing after interrupt acceptance and internal processing after vector fetch.

5. Not available in this LSI.

Symbol	Internal Memory	External Device			
		8-Bit Bus		16-Bit Bus	
		2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch $S_i$	1	4	6 + 2 m	2	3 + m
Branch address read $S_j$					
Stack manipulation $S_k$					

Legend:

m: Number of wait states in an external device access.

### 5.6.5 DTC and DMAC Activation by Interrupt

The DTC and DMAC can be activated by an interrupt. In this case, the following options are available:

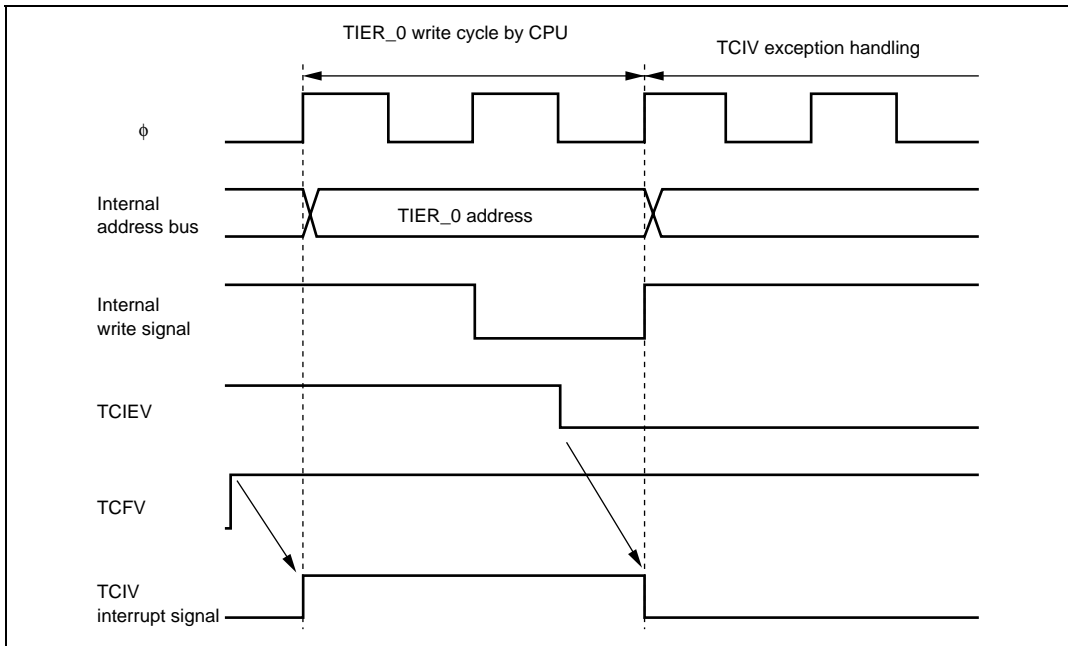
- Interrupt request to CPU
- Activation request to DTC
- Activation request to DMAC
- Selection of a number of the above

For details of interrupt requests that can be used to activate the DTC and DMAC, see table 5.2 and section 8, Data Transfer Controller (DTC) and section 7, DMA Controller (DMAC).

## 5.7.1 Contention between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to mask interrupts, the masking becomes effective after execution of the instruction.

When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored. The same also applies when an interrupt source flag is cleared to 0. Figure 5.6 shows an example in which the TCIEV bit in the TPU's TIER\_0 register is cleared to 0. The above contention will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.



**Figure 5.6 Contention between Interrupt Generation and Disabling**



Instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions is executed, all interrupts including NMI are disabled and the next instruction is always executed. When the I bit is set by one of these instructions, the new value becomes valid two states after execution of the instruction ends.

### 5.7.3 Times when Interrupts Are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU has updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

### 5.7.4 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during the transfer is not accepted until the transfer is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, interrupt exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction. Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

```
L1:   EEPMOV.W
      MOV.W   R4, R4
      BNE    L1
```

### 5.7.5 Change of IRQ Pin Select Register (ITSR) Setting

When the ITSr setting is changed, an edge occurs internally and the IRQnF bit (n = 0 to 7) of ISR may be set to 1 at the unintended timing if the selected pin level before the change is different from the selected pin level after the change. If the IRQn interrupt request (n = 0 to 7) is enabled, the interrupt exception handling is executed. To prevent the unintended interrupt, ITSr setting should be changed while the IRQn interrupt request is disabled, then the IRQnF bit should be cleared to 0.

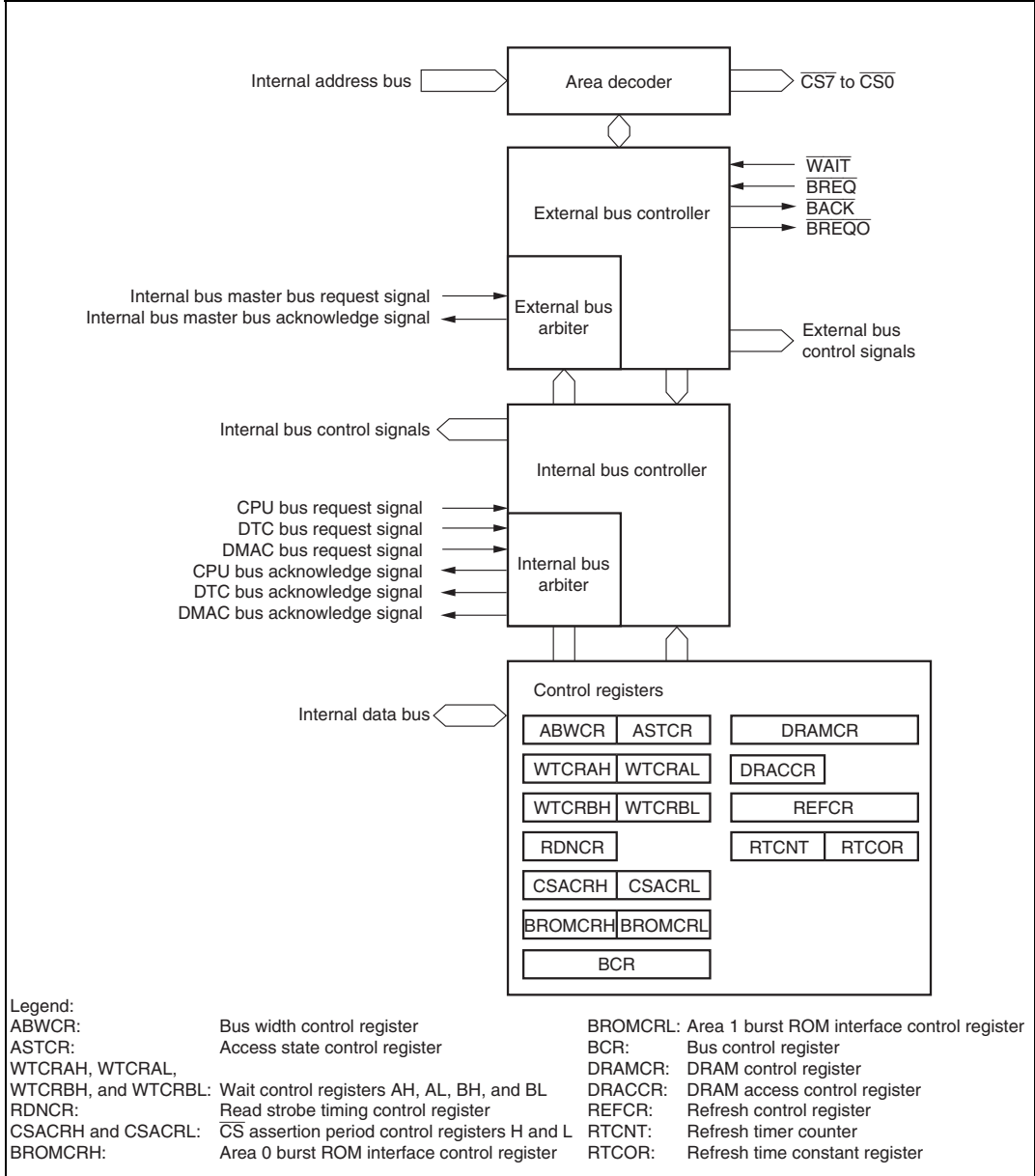
Since IRQnF flags may be set to 1 depending on the pin states after a reset, be sure to read from ISR after a reset and then write 0 to clear the IRQnF flags.

This LSI has an on-chip bus controller (BSC) that manages the external space divided into eight areas.

The bus controller also has a bus arbitration function, and controls the operation of the bus masters—the CPU, DMA controller (DMAC) and data transfer controller (DTC).

## 6.1 Features

- Manages external space in area units
  - Manages the external space divided into eight areas of 2 Mbytes
  - Bus specifications can be set independently for each area
  - Burst ROM, or DRAM, interface can be set
- Basic bus interface
  - Chip select signals ( $\overline{CS0}$  to  $\overline{CS7}$ ) can be output for areas 0 to 7
  - 8-bit access or 16-bit access can be selected for each area
  - 2-state access or 3-state access can be selected for each area
  - Program wait states can be inserted for each area
- Burst ROM interface
  - Burst ROM interface can be set independently for areas 0 and 1
- DRAM interface
  - DRAM interface can be set for areas 2 to 5
- Bus arbitration function
  - Includes a bus arbiter that arbitrates bus right between the CPU, DMAC, and DTC



**Figure 6.1 Block Diagram of Bus Controller**

Table 6.1 summarizes the pin configuration of the bus controller.

**Table 6.1 Pin Configuration**

<b>Name</b>	<b>Symbol</b>	<b>I/O</b>	<b>Function</b>
Address strobe	$\overline{AS}$	Output	Strobe signal indicating that normal space is accessed and address output on address bus is enabled.
Read	$\overline{RD}$	Output	Strobe signal indicating that normal space is being read.
High write	$\overline{HWR}$	Output	Strobe signal indicating that normal space is written to, and upper half (D15 to D8) of data bus is enabled or DRAM space write enable signal.
Low write	$\overline{LWR}$	Output	Strobe signal indicating that normal space is written to, and lower half (D7 to D0) of data bus is enabled.
Chip select 0	$\overline{CS0}$	Output	Strobe signal indicating that area 0 is selected.
Chip select 1	$\overline{CS1}$	Output	Strobe signal indicating that area 1 is selected.
Chip select 2/row address strobe 2	$\overline{CS2}/$ $\overline{RAS2}$	Output	Strobe signal indicating that area 2 is selected, DRAM row address strobe signal when area 2 is DRAM space or areas 2 to 5 are set as continuous DRAM space.
Chip select 3/row address strobe 3	$\overline{CS3}/$ $\overline{RAS3}$	Output	Strobe signal indicating that area 3 is selected, DRAM row address strobe signal when area 3 is DRAM space.
Chip select 4	$\overline{CS4}$	Output	Strobe signal indicating that area 4 is selected.
Chip select 5	$\overline{CS5}$	Output	Strobe signal indicating that area 5 is selected.
Chip select 6	$\overline{CS6}$	Output	Strobe signal indicating that area 6 is selected.
Chip select 7	$\overline{CS7}$	Output	Strobe signal indicating that area 7 is selected.
Upper column address strobe	$\overline{UCAS}$	Output	16-bit DRAM space upper column address strobe signal, 8-bit DRAM space column address strobe signal.

Output enable	$\overline{OE}$	Output	Output enable signal for the DRAM space.
Wait	$\overline{WAIT}$	Input	Wait request signal when accessing external address space.
Bus request	$\overline{BREQ}$	Input	Request signal for release of bus to external bus master.
Bus request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating that bus has been released to external bus master.
Bus request output	$\overline{BREQO}$	Output	External bus request signal used when internal bus master accesses external address space when external bus is released.
Data transfer acknowledge 1 (DMAC)	$\overline{DACK1}$	Output	Data transfer acknowledge signal for single address transfer by DMAC channel 1.
Data transfer acknowledge 0 (DMAC)	$\overline{DACK0}$	Output	Data transfer acknowledge signal for single address transfer by DMAC channel 0.

### 6.3 Register Descriptions

The bus controller has the following registers.

- Bus width control register (ABWCR)
- Access state control register (ASTCR)
- Wait control register AH (WTCRAH)
- Wait control register AL (WTCRAL)
- Wait control register BH (WTCRBH)
- Wait control register BL (WTCRBL)
- Read strobe timing control register (RDNCR)
- $\overline{CS}$  assertion period control register H (CSACRH)
- $\overline{CS}$  assertion period control register L (CSACRL)
- Area 0 burst ROM interface control register (BROMCRH)
- Area 1 burst ROM interface control register (BROMCRL)
- Bus control register (BCR)
- DRAM control register (DRAMCR)
- DRAM access control register (DRACCR)

- Refresh time constant register (RTCOR)

### 6.3.1 Bus Width Control Register (ABWCR)

ABWCR designates each area in the external address space as either 8-bit access space or 16-bit access space.

Bit	Bit Name	Initial Value*	R/W	Description
7	ABW7	1/0	R/W	Area 7 to 0 Bus Width Control
6	ABW6	1/0	R/W	These bits select whether the corresponding area is to be designated as 8-bit access space or 16-bit access space. 0: Area n is designated as 16-bit access space 1: Area n is designated as 8-bit access space (n = 7 to 0)
5	ABW5	1/0	R/W	
4	ABW4	1/0	R/W	
3	ABW3	1/0	R/W	
2	ABW2	1/0	R/W	
1	ABW1	1/0	R/W	
0	ABW0	1/0	R/W	

Note: \* In modes 2 and 4, ABWCR is initialized to 1. In modes 1 and 7, ABWCR is initialized to 0.

### 6.3.2 Access State Control Register (ASTCR)

ASTCR designates each area in the external address space as either 2-state access space or 3-state access space.

Bit	Bit Name	Initial Value	R/W	Description
7	AST7	1	R/W	Area 7 to 0 Access State Control
6	AST6	1	R/W	These bits select whether the corresponding area is to be designated as 2-state access space or 3-state access space. Wait state insertion is enabled or disabled at the same time. 0: Area n is designated as 2-state access space Wait state insertion in area n access is disabled 1: Area n is designated as 3-state access space Wait state insertion in area n access is enabled (n = 7 to 0)
5	AST5	1	R/W	
4	AST4	1	R/W	
3	AST3	1	R/W	
2	AST2	1	R/W	
1	AST1	1	R/W	
0	AST0	1	R/W	

WTCRA and WTCRB select the number of program wait states for each area in the external address space.

- WTCRAH

Bit	Bit Name	Initial Value	R/W	Description	
15	—	0	R	Reserved This bit is always read as 0 and cannot be modified.	
14	W72	1	R/W	Area 7 Wait Control 2 to 0	
13	W71	1	R/W	These bits select the number of program wait states when accessing area 7 while AST7 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted	
12	W70	1	R/W		
11	—	0	R		Reserved This bit is always read as 0 and cannot be modified.



9	W61	1	R/W	These bits select the number of program wait states when accessing area 6 while AST6 bit in ASTCR = 1.
8	W60	1	R/W	
000: Program wait not inserted				
001: 1 program wait state inserted				
010: 2 program wait states inserted				
011: 3 program wait states inserted				
100: 4 program wait states inserted				
101: 5 program wait states inserted				
110: 6 program wait states inserted				
111: 7 program wait states inserted				

---

- WTARAL

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved  This bit is always read as 0 and cannot be modified.
6	W52	1	R/W	Area 5 Wait Control 2 to 0
5	W51	1	R/W	These bits select the number of program wait states when accessing area 5 while AST5 bit in ASTCR = 1.
4	W50	1	R/W	
000: Program wait not inserted				
001: 1 program wait state inserted				
010: 2 program wait states inserted				
011: 3 program wait states inserted				
100: 4 program wait states inserted				
101: 5 program wait states inserted				
110: 6 program wait states inserted				
111: 7 program wait states inserted				

---

This bit is always read as 0 and cannot be modified.

---

2	W42	1	R/W	Area 4 Wait Control 2 to 0
1	W41	1	R/W	These bits select the number of program wait states when accessing area 4 while AST4 bit in ASTCR = 1.  000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
0	W40	1	R/W	

---

15	—	0	R	Reserved	This bit is always read as 0 and cannot be modified.	
14	W32	1	R/W	Area 3 Wait Control 2 to 0		
13	W31	1	R/W	These bits select the number of program wait states when accessing area 3 while AST3 bit in ASTCR = 1.		
12	W30	1	R/W			
					000: Program wait not inserted	
					001: 1 program wait state inserted	
					010: 2 program wait states inserted	
					011: 3 program wait states inserted	
					100: 4 program wait states inserted	
				101: 5 program wait states inserted		
				110: 6 program wait states inserted		
				111: 7 program wait states inserted		
11	—	0	R	Reserved	This bit is always read as 0 and cannot be modified.	
10	W22	1	R/W	Area 2 Wait Control 2 to 0		
9	W21	1	R/W	These bits select the number of program wait states when accessing area 2 while AST2 bit in ASTCR = 1.		
8	W20	1	R/W			
					000: Program wait not inserted	
					001: 1 program wait state inserted	
					010: 2 program wait states inserted	
					011: 3 program wait states inserted	
					100: 4 program wait states inserted	
				101: 5 program wait states inserted		
				110: 6 program wait states inserted		
				111: 7 program wait states inserted		

7	—	0	R	Reserved This bit is always read as 0 and cannot be modified.	
6	W12	1	R/W	Area 1 Wait Control 2 to 0	
5	W11	1	R/W	These bits select the number of program wait states when accessing area 1 while AST1 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted	
4	W10	1	R/W		
3	—	0	R		Reserved This bit is always read as 0 and cannot be modified.
2	W02	1	R/W		Area 0 Wait Control 2 to 0
1	W01	1	R/W		These bits select the number of program wait states when accessing area 0 while AST0 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
0	W00	1	R/W		

RDNCR selects the read strobe signal ( $\overline{RD}$ ) negation timing in a basic bus interface read access.

Bit	Bit Name	Initial Value	R/W	Description
7	RDN7	0	R/W	Read Strobe Timing Control 7 to 0
6	RDN6	0	R/W	These bits set the negation timing of the read strobe in a corresponding area read access.
5	RDN5	0	R/W	
4	RDN4	0	R/W	As shown in figure 6.2, the read strobe for an area for which the RDNn bit is set to 1 is negated one half-state earlier than that for an area for which the RDNn bit is cleared to 0. The read data setup and hold time specifications are also one half-state earlier.
3	RDN3	0	R/W	
2	RDN2	0	R/W	0: In an area n read access, the $\overline{RD}$ is negated at the end of the read cycle
1	RDN1	0	R/W	
0	RDN0	0	R/W	1: In an area n read access, the $\overline{RD}$ is negated one half-state before the end of the read cycle (n = 7 to 0)

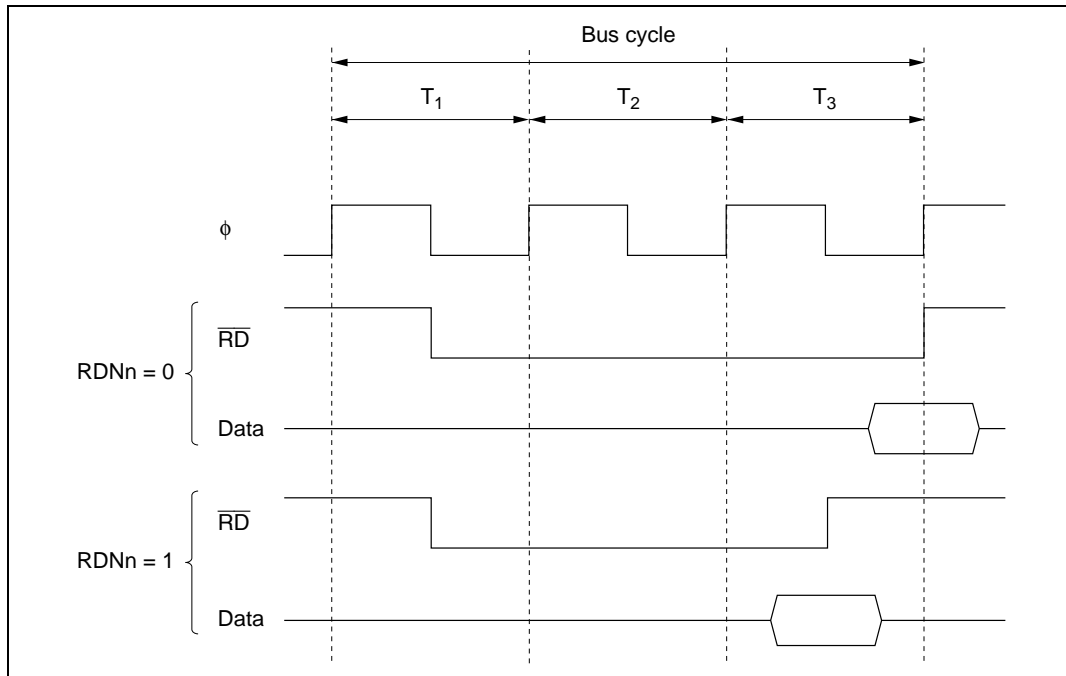


Figure 6.2 Read Strobe Negation Timing (Example of 3-State Access Space)

CSACRH and CSACRL select whether or not the assertion period of the basic bus interface chip select signals ( $\overline{CS}_n$ ) and address signals is to be extended. Extending the assertion period of the  $\overline{CS}_n$  and address signals allows flexible interfacing to external I/O devices.

- CSACRH

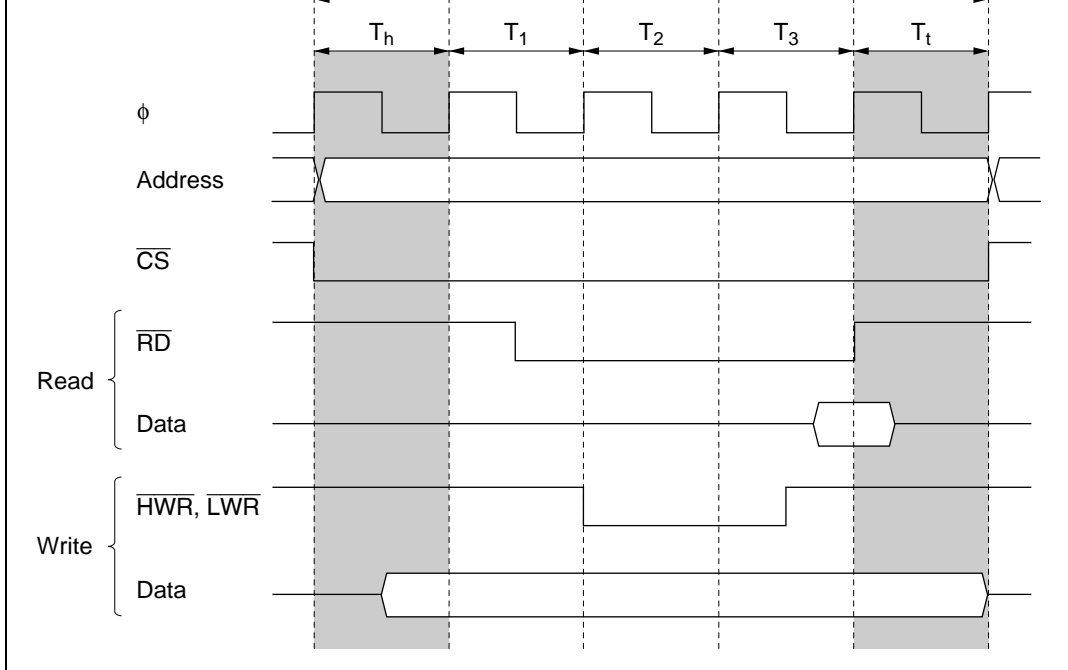
Bit	Bit Name	Initial Value	R/W	Description	
7	CSXH7	0	R/W	$\overline{CS}$ and Address Signal Assertion Period Control	
6	CSXH6	0	R/W	1	
5	CSXH5	0	R/W	These bits specify whether or not the $T_h$ cycle is to be inserted (see figure 6.3). When an area for which the CSXHn bit is set to 1 is accessed, a one-state $T_h$ cycle, in which only the $\overline{CS}_n$ and address signals are asserted, is inserted before the normal access cycle.	
4	CSXH4	0	R/W		
3	CSXH3	0	R/W		
2	CSXH2	0	R/W		
1	CSXH1	0	R/W		
0	CSXH0	0	R/W		
					0: In area n basic bus interface access, the $\overline{CS}_n$ and address assertion period ( $T_h$ ) is not extended
					1: In area n basic bus interface access, the $\overline{CS}_n$ and address assertion period ( $T_h$ ) is extended

(n = 7 to 0)

- CSACRL

Bit	Bit Name	Initial Value	R/W	Description	
7	CSXT7	0	R/W	$\overline{CS}$ and Address Signal Assertion Period Control	
6	CSXT6	0	R/W	2	
5	CSXT5	0	R/W	These bits specify whether or not the $T_l$ cycle shown in figure 6.3 is to be inserted. When an area for which the CSXTn bit is set to 1 is accessed, a one-state $T_l$ cycle, in which only the $\overline{CS}_n$ and address signals are asserted, is inserted after the normal access cycle.	
4	CSXT4	0	R/W		
3	CSXT3	0	R/W		
2	CSXT2	0	R/W		
1	CSXT1	0	R/W		
0	CSXT0	0	R/W		
					0: In area n basic bus interface access, the $\overline{CS}_n$ and address assertion period ( $T_l$ ) is not extended
					1: In area n basic bus interface access, the $\overline{CS}_n$ and address assertion period ( $T_l$ ) is extended

(n = 7 to 0)



**Figure 6.3  $\overline{CS}$  and Address Assertion Period Extension (Example of 3-State Access Space and  $RDNn = 0$ )**

BROMCRH and BROMCRL are used to make burst ROM interface settings. Area 0 and area 1 burst ROM interface settings can be made independently in BROMCRH and BROMCRL, respectively.

Bit	Bit Name	Initial Value	R/W	Description	
7	BSRMn	0	R/W	Burst ROM Interface Select Selects the basic bus interface or burst ROM interface. 0: Basic bus interface space 1: Burst ROM interface space	
6	BSTSn2	0	R/W	Burst Cycle Select	
5	BSTSn1	0	R/W	These bits select the number of burst cycle states. 000: 1 state 001: 2 states 010: 3 states 011: 4 states 100: 5 states 101: 6 states 110: 7 states 111: 8 states	
4	BSTSn0	0	R/W		
3 and 2	—	All 0	R/W		Reserved Though these bits can be read from or written to, the write value should always be 0.
1	BSWDn1	0	R/W		Burst Word Number Select
0	BSWDn0	0	R/W		These bits select the number of words that can be burst-accessed on the burst ROM interface. 00: Maximum 4 words 01: Maximum 8 words 10: Maximum 16 words 11: Maximum 32 words

(n = 1 or 0)



BCR is used for idle cycle settings, selection of the bus released state protocol, enabling or disabling of the write data buffer function, and enabling or disabling of WAIT pin input.

Bit	Bit Name	Initial Value	R/W	Description
15	BRLE	0	R/W	<p>External Bus Release Enable</p> <p>Enables or disables external bus release.</p> <p>0: External bus release disabled  <math>\overline{\text{BREQ}}</math>, BACK, and <math>\overline{\text{BREQO}}</math> pins can be used as I/O ports</p> <p>1: External bus release enabled</p>
14	BREQOE	0	R/W	<p><math>\overline{\text{BREQO}}</math> Pin Enable</p> <p>Controls outputting the bus request signal (<math>\overline{\text{BREQO}}</math>) to the external bus master in the external bus released state, when an internal bus master performs an external address space access, or when a refresh request is generated.</p> <p>0: <math>\overline{\text{BREQO}}</math> output disabled  <math>\overline{\text{BREQO}}</math> pin can be used as I/O port</p> <p>1: <math>\overline{\text{BREQO}}</math> output enabled</p>
13	—	0	R/W	<p>Reserved</p> <p>Though this bit can be read from or written to, the write value should always be 0.</p>
12	IDLC	1	R/W	<p>Idle Cycle State Number Select</p> <p>Specifies the number of states in the idle cycle set by ICIS2, ICIS1, and ICIS0.</p> <p>0: Idle cycle comprises 1 state</p> <p>1: Idle cycle comprises 2 states</p>
11	ICIS1	1	R/W	<p>Idle Cycle Insert 1</p> <p>When consecutive external read cycles are performed in different areas, an idle cycle can be inserted between the bus cycles.</p> <p>0: Idle cycle not inserted</p> <p>1: Idle cycle inserted</p>

When an external read cycle and external write cycle are performed consecutively, an idle cycle can be inserted between the bus cycles.

0: Idle cycle not inserted

1: Idle cycle inserted

9	WDBE	0	R/W	<p>Write Data Buffer Enable</p> <p>The write data buffer function can be used for an external write cycle or DMAC single address transfer cycle.</p> <p>0: Write data buffer function not used</p> <p>1: Write data buffer function used</p>
8	WAITE	0	R/W	<p><math>\overline{\text{WAIT}}</math> Pin Enable</p> <p>Selects enabling or disabling of wait input by the <math>\overline{\text{WAIT}}</math> pin.</p> <p>0: Wait input by <math>\overline{\text{WAIT}}</math> pin disabled <math>\overline{\text{WAIT}}</math> pin can be used as I/O port</p> <p>1: Wait input by <math>\overline{\text{WAIT}}</math> pin enabled</p>
7 to 3	—	All 0	R/W	<p>Reserved</p> <p>Though these bits can be read from or written to, the write value should always be 0.</p>
2	ICIS2	0	R/W	<p>Idle Cycle Insert 2</p> <p>When an external write cycle and external read cycle are performed consecutively, an idle cycle can be inserted between the bus cycles.</p> <p>0: Idle cycle not inserted</p> <p>1: Idle cycle inserted</p>
1 and 0	—	0	R/W	<p>Reserved</p> <p>Though these bits can be read from or written to, the write value should always be 0.</p>

DRAMCR is used to make DRAM/synchronous DRAM interface settings.

Bit	Bit Name	Initial Value	R/W	Description
15	OEE	0	R/W	<p><math>\overline{OE}</math> Output Enable</p> <p>The OE signal used when EDO page mode DRAM is connected can be output from the (<math>\overline{OE}</math>) pin. The <math>\overline{OE}</math> signal is common to all areas designated as DRAM space.</p> <p>0: <math>\overline{OE}</math> signal output disabled  <math>\overline{OE}</math> pin can be used as I/O port</p> <p>1: <math>\overline{OE}</math> signal output enabled</p>
14	RAST	0	R/W	<p><math>\overline{RAS}</math> Assertion Timing Select</p> <p>Selects whether, in DRAM access, the <math>\overline{RAS}</math> signal is asserted from the start of the <math>T_r</math> cycle (rising edge of <math>\phi</math>) or from the falling edge of <math>\phi</math>.</p> <p>Figure 6.4 shows the relationship between the RAST bit setting and the <math>\overline{RAS}</math> assertion timing.</p> <p>The setting of this bit applies to all areas designated as DRAM space.</p> <p>0: <math>\overline{RAS}</math> is asserted from <math>\phi</math> falling edge in <math>T_r</math> cycle</p> <p>1: <math>\overline{RAS}</math> is asserted from start of <math>T_r</math> cycle</p>
13	—	0	R/W	<p>Reserved</p> <p>Though this bit can be read from or written to, the write value should always be 0.</p>
12	CAST	0	R/W	<p>Column Address Output Cycle Number Select</p> <p>Selects whether the column address output cycle in DRAM access comprises 3 states or 2 states. The setting of this bit applies to all areas designated as DRAM space.</p> <p>0: 2-state column address output cycle</p> <p>1: 3-state column address output cycle</p>
11	—	0	R/W	<p>Reserved</p> <p>Though this bit can be read from or written to, the write value should always be 0.</p>

9	RMTS1	0	R/W	These bits designate DRAM space for areas 2 to 5.  When continuous DRAM space is set, it is possible to connect large-capacity DRAM exceeding 2 Mbytes per area. In this case, the $\overline{\text{RAS}}$ signal is output from the $\overline{\text{CS2}}$ pin.
8	RMTS0	0	R/W	
<p>000: Normal space</p> <p>001: Normal space in areas 3 to 5 DRAM space in area 2</p> <p>010: Normal space in areas 4 and 5 DRAM space in areas 2 and 3</p> <p>011: Reserved (setting prohibited)</p> <p>100: Reserved (setting prohibited)</p> <p>101: Reserved (setting prohibited)</p> <p>110: Reserved (setting prohibited)</p> <p>111: Continuous DRAM space in areas 2 to 5</p>				
7	BE	0	R/W	<p>Burst Access Enable</p> <p>Selects enabling or disabling of burst access to areas designated as DRAM space. DRAM space burst access is performed in fast page mode. When using EDO page mode DRAM, the <math>\overline{\text{OE}}</math> signal must be connected.</p> <p>0: Full access</p> <p>1: Access in fast page mode</p>

When access to DRAM space is interrupted by an access to normal space, an access to an internal I/O register, etc., this bit selects whether the  $\overline{\text{RAS}}$  signal is held low while waiting for the next DRAM access ( $\overline{\text{RAS}}$  down mode), or is driven high again ( $\overline{\text{RAS}}$  up mode). The setting of this bit is valid only when the BE bit is set to 1.

If this bit is cleared to 0 when set to 1 in the  $\overline{\text{RAS}}$  down state, the  $\overline{\text{RAS}}$  down state is cleared at that point, and  $\overline{\text{RAS}}$  goes high.

0:  $\overline{\text{RAS}}$  up mode selected for DRAM space access

1:  $\overline{\text{RAS}}$  down mode selected for DRAM space access

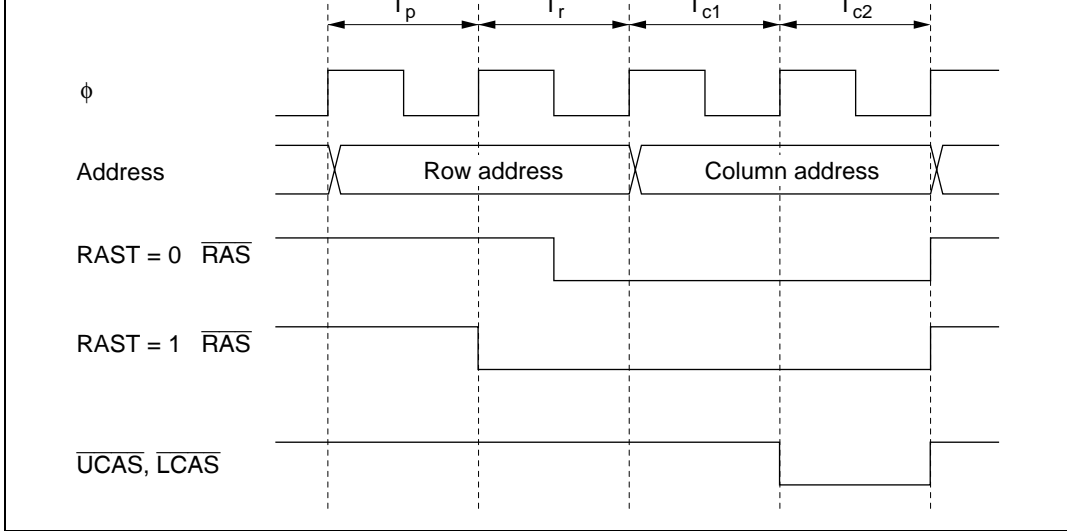
5	DDS	0	R/W	<p>DMAC Single Address Transfer Option</p> <p>Selects whether full access is always performed or burst access is enabled when DMAC single address transfer is performed on the DRAM</p> <p>When the BE bit is cleared to 0 in DRAMCR, disabling DRAM burst access, DMAC single address transfer is performed in full access mode regardless of the setting of this bit.</p> <p>This bit has no effect on other bus master external accesses or DMAC dual address transfers.</p> <p>0: Full access is always executed</p> <p>1: Burst access is enabled</p>
4 and 3	—	All 0	R/W	<p>Reserved</p> <p>Though these bits can be read from or written to, the write value should always be 0.</p>

1	MXC1	0	R/W	These bits select the size of the shift toward the lower half of the row address in row address/column address multiplexing. In burst operation on the DRAM interface, these bits also select the row address bits to be used for comparison. For details, refer to section 6.6.2, Address Multiplexing.
0	MXC0	0	R/W	
				000: 8-bit shift
				<ul style="list-style-type: none"> <li>When 8-bit access space is designated: Row address bits A23 to A8 used for comparison</li> <li>When 16-bit access space is designated: Row address bits A23 to A9 used for comparison</li> </ul>
				001: 9-bit shift
				<ul style="list-style-type: none"> <li>When 8-bit access space is designated: Row address bits A23 to A9 used for comparison</li> <li>When 16-bit access space is designated: Row address bits A23 to A10 used for comparison</li> </ul>
				010: 10-bit shift
				<ul style="list-style-type: none"> <li>When 8-bit access space is designated: Row address bits A23 to A10 used for comparison</li> <li>When 16-bit access space is designated: Row address bits A23 to A11 used for comparison</li> </ul>
				011: 11-bit shift
				<ul style="list-style-type: none"> <li>When 8-bit access space is designated: Row address bits A23 to A11 used for comparison</li> <li>When 16-bit access space is designated: Row address bits A23 to A12 used for comparison</li> </ul>
				1xx: Setting prohibited

---

Legend:

×: Don't care



**Figure 6.4  $\overline{\text{RAS}}$  Signal Assertion Timing  
(2-State Column Address Output Cycle, Full Access)**

DRACCR is used to set the DRAM interface bus specifications.

Bit	Bit Name	Initial Value	R/W	Description
7	DRMI	0	R/W	<p>Idle Cycle Insertion</p> <p>An idle cycle can be inserted after a DRAM access cycle when a continuous normal space access cycle follows a DRAM access cycle. Idle cycle insertion conditions, setting of number of states, etc., comply with settings of bits ICIS2, ICIS1, ICIS0, and IDLC in BCR register</p> <p>0: Idle cycle not inserted 1: Idle cycle inserted</p>
6	—	0	R/W	<p>Reserved</p> <p>Though this bit can be read from or written to, the write value should always be 0.</p>
5	TPC1	0	R/W	<p>Precharge State Control</p> <p>These bits select the number of states in the RAS precharge cycle in normal access and refreshing.</p> <p>00: 1-state RAS precharge cycle 01: 2-state RAS precharge cycle 10: 3-state RAS precharge cycle 11: 4-state RAS precharge cycle</p>
4	TPC0	0	R/W	
3, 2	—	All 0	R/W	<p>Reserved</p> <p>Though these bits can be read from or written to, the write value should always be 0.</p>
1	RCD1	0	R/W	<p>RAS-CAS Wait Control</p> <p>These bits select a wait cycle to be inserted between the RAS assert cycle and <math>\overline{\text{CAS}}</math> assert cycle. A 1- to 4-state wait cycle can be inserted.</p> <p>00: Wait cycle not inserted 01: 1-state wait cycle inserted 10: 2-state wait cycle inserted 11: 3-state wait cycle inserted</p>
0	RCD0	0	R/W	



Bit	Bit Name	Initial Value	R/W	Description
15	CMF	0	R/(W)*	<p>Compare Match Flag</p> <p>Status flag that indicates a match between the values of RTCNT and RTCOR.</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When 0 is written to CMF after reading CMF = 1 while the RFSHE bit is cleared to 0</li> <li>When CBR refreshing is executed while the RFSHE bit is set to 1</li> </ul> <p>[Setting condition]</p> <p>When RTCOR = RTCNT</p>
14	CMIE	0	R/W	<p>Compare Match Interrupt Enable</p> <p>Enables or disables interrupt requests (CMI) by the CMF flag when the CMF flag is set to 1.</p> <p>This bit is valid when refresh control is not performed. When the refresh control is performed, this bit is always cleared to 0 and cannot be modified.</p> <p>0: Interrupt request by CMF flag disabled 1: Interrupt request by CMF flag enabled</p>
13	RCW1	0	R/W	<p>CAS-RAS Wait Control</p> <p>These bits select the number of wait cycles to be inserted between the <math>\overline{\text{CAS}}</math> assert cycle and <math>\overline{\text{RAS}}</math> assert cycle in a DRAM refresh cycle.</p> <p>00: Wait state not inserted 01: 1 wait state inserted 10: 2 wait states inserted 11: 3 wait states inserted</p>
12	RCW0	0	R/W	
11	—	0	R/W	<p>Reserved</p> <p>Though this bit can be read from or written to, the write value should always be 0.</p>

Note: \* Only 0 can be written, to clear the flag.

9	RTCK1	0	R/W	<p>These bits select the clock to be used to increment the refresh counter. When the input clock is selected with bits RTCK2 to RTCK0, the refresh counter begins counting up.</p> <p>000: Count operation halted</p> <p>001: Count on <math>\phi/2</math></p> <p>010: Count on <math>\phi/8</math></p> <p>011: Count on <math>\phi/32</math></p> <p>100: Count on <math>\phi/128</math></p> <p>101: Count on <math>\phi/512</math></p> <p>110: Count on <math>\phi/2048</math></p> <p>111: Count on <math>\phi/4096</math></p>	
8	RTCK0	0	R/W		
<hr/>					
7	RFSHE	0	R/W		<p>Refresh Control</p> <p>Refresh control can be performed. When refresh control is not performed, the refresh timer can be used as an interval timer.</p> <p>0: Refresh control is not performed</p> <p>1: Refresh control is performed</p>
<hr/>					
6	CBRM	0	R/W		<p>CBR Refresh Control Mode</p> <p>Selects CBR refreshing performed in parallel with other external accesses, or execution of CBR refreshing alone.</p> <p>0: External access during CAS-before-RAS refreshing is enabled</p> <p>1: External access during CAS-before-RAS refreshing is disabled</p>
<hr/>					
5	RLW1	0	R/W		<p>Refresh Cycle Wait Control</p> <p>These bits select the number of wait states to be inserted in a DRAM interface CAS-before-RAS refresh cycle. This setting applies to all areas designated as DRAM space.</p> <p>00: No wait state inserted</p> <p>01: 1 wait state inserted</p> <p>10: 2 wait states inserted</p> <p>11: 3 wait states inserted</p>
4	RLW0	0	R/W		

If this bit is set to 1, DRAM self-refresh mode is selected when a transition is made to the software standby state. This bit is valid when the RFSHE bit is set to 1, enabling refresh operations. It is cleared after recovery from software standby mode.

0: Self-refreshing is disabled

1: Self-refreshing is enabled

---

2	TPCS2	0	R/W	Self-Refresh Precharge Cycle Control
1	TPCS1	0	R/W	These bits select the number of states in the precharge cycle immediately after self-refreshing.  The number of states in the precharge cycle immediately after self-refreshing are added to the number of states set by bits TPC1 and TPC0 in DRACCR.  000: [TPC set value] states 001: [TPC set value + 1] states 010: [TPC set value + 2] states 011: [TPC set value + 3] states 100: [TPC set value + 4] states 101: [TPC set value + 5] states 110: [TPC set value + 6] states 111: [TPC set value + 7] states
0	TPCS0	0	R/W	

---

RTCNT is an 8-bit readable/writable up-counter. RTCNT counts up using the internal clock selected by bits RTCK2 to RTCK0 in REFCR.

When RTCNT matches RTCOR (compare match), the CMF flag in REFCR is set to 1 and RTCNT is cleared to H'00. If the RFSHE bit in REFCR is set to 1 at this time, a refresh cycle is started. If the RFSHE bit is cleared to 0 and the CMIE bit in REFCR is set to 1, a compare match interrupt (CMI) is generated.

RTCNT is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

### **6.3.12 Refresh Time Constant Register (RTCOR)**

RTCOR is an 8-bit readable/writable register that sets the period for compare match operations with RTCNT.

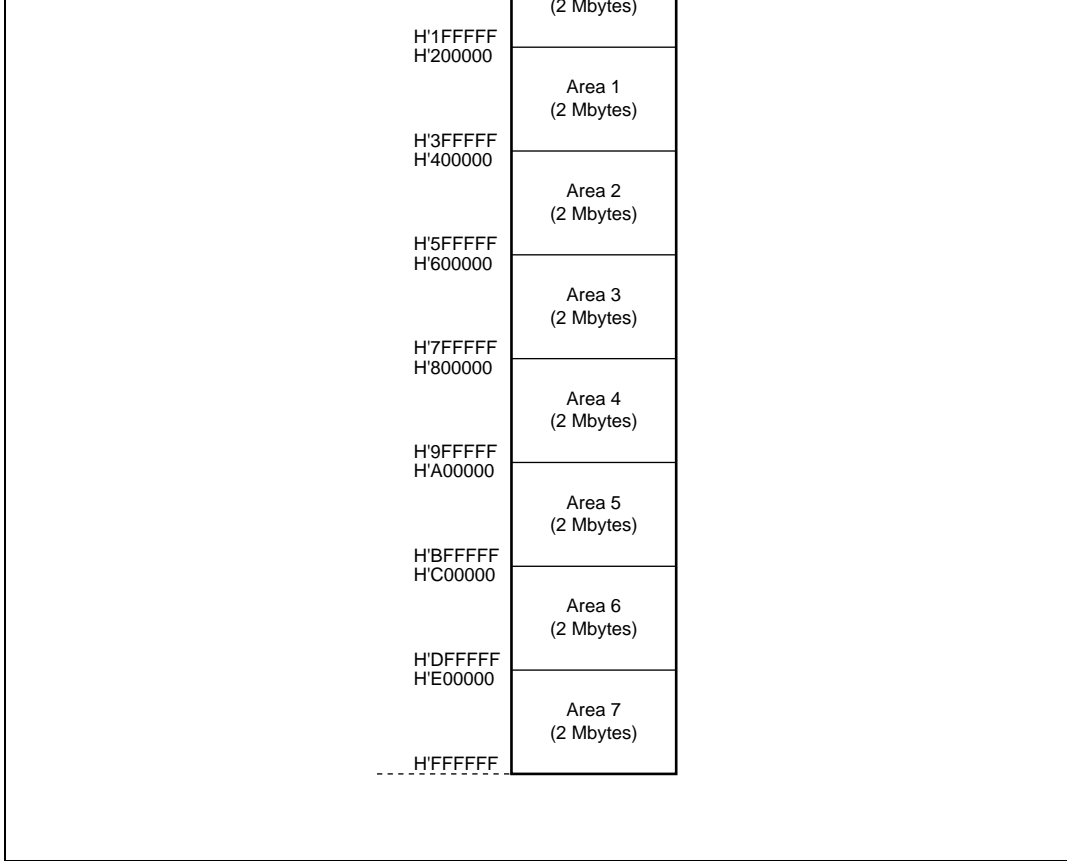
The values of RTCOR and RTCNT are constantly compared, and if they match, the CMF flag in REFCR is set to 1 and RTCNT is cleared to H'00.

RTCOR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

## **6.4 Operation**

### **6.4.1 Area Division**

The bus controller divides the 16-Mbyte address space into eight areas, 0 to 7, in 2-Mbyte units, and performs bus control for external address space in area units. Chip select signals ( $\overline{CS0}$  to  $\overline{CS7}$ ) can be output for each area. Figure 6.5 shows an outline of the memory map.



**Figure 6.5 Area Divisions**

The external space bus specifications consist of five elements: bus width, number of access states, number of program wait states, read strobe timing, and chip select ( $\overline{CS}$ ) assertion period extension states. The bus width and number of access states for on-chip memory and internal I/O registers are fixed, and are not affected by the bus controller.

**Bus Width:** A bus width of 8 or 16 bits can be selected with ABWCR. An area for which an 8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is selected functions as a 16-bit access space. If all areas are designated as 8-bit access space, 8-bit bus mode is set; if any area is designated as 16-bit access space, 16-bit bus mode is set.

**Number of Access States:** Two or three access states can be selected with ASTCR. An area for which 2-state access is selected functions as a 2-state access space, and an area for which 3-state access is selected functions as a 3-state access space. With the DRAM interface and burst ROM interface, the number of access states may be determined without regard to the setting of ASTCR.

When 2-state access space is designated, wait insertion is disabled. When 3-state access space is designated, it is possible to insert program waits by means of the WTCRA and WTCRB, and external waits by means of the  $\overline{WAIT}$  pin.

**Number of Program Wait States:** When 3-state access space is designated by ASTCR, the number of program wait states to be inserted automatically is selected with WTCRA and WTCRB. From 0 to 7 program wait states can be selected. Table 6.2 shows the bus specifications (bus width, and number of access states and program wait states) for each basic bus interface area.

					Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn2	Wn1	Wn0	Bus Width	Access States	Program Wait States
0	0	—	—	—	16	2	0
	1	0	0	0		3	0
				1			1
				1		0	2
				1			3
		1	0	0			4
				1			5
			1	0	6		
			1		7		
1	0	—	—	—	8	2	0
	1	0	0	0		3	0
				1			1
				1		0	2
				1			3
		1	0	0			4
				1			5
			1	0	6		
			1		7		

(n = 0 to 7)

**Read Strobe Timing:** RDNCR can be used to select either of two negation timings (at the end of the read cycle or one half-state before the end of the read cycle) for the read strobe ( $\overline{RD}$ ) used in the basic bus interface space.

**Chip Select ( $\overline{CS}$ ) Assertion Period Extension States:** Some external I/O devices require a setup time and hold time between address and  $\overline{CS}$  signals and strobe signals such as  $\overline{RD}$ ,  $\overline{HWR}$ , and  $\overline{LWR}$ . CSACR can be used to insert states in which only the  $\overline{CS}$ , AS, and address signals are asserted before and after a basic bus space access cycle.

The memory interfaces in this LSI comprise a basic bus interface that allows direct connection of ROM, SRAM, and so on a synchronous DRAM interface that allows direct connection of synchronous DRAM; and a burst ROM interface that allows direct connection of burst ROM. The interface can be selected independently for each area.

An area for which the basic bus interface is designated functions as normal space, an area for which the DRAM interface is designated functions as DRAM space, and an area for which the burst ROM interface is designated functions as burst ROM space.

The initial state of each area is basic bus interface, 3-state access space. The initial bus width is selected according to the operating mode.

**Area 0:** Area 0 includes on-chip ROM in expanded mode with on-chip ROM enabled and the space excluding on-chip ROM is external space, and in expanded mode with on-chip ROM disabled, all of area 0 is external space.

When area 0 external space is accessed, the  $\overline{CS0}$  signal can be output.

Either basic bus interface or burst ROM interface can be selected for area 0.

**Area 1:** In externally expanded mode, all of area 1 is external space.

When area 1 external space is accessed, the  $\overline{CS1}$  signal can be output.

Either basic bus interface or burst ROM interface can be selected for area 1.

**Areas 2 to 5:** In externally expanded mode, areas 2 to 5 are all external space.

When area 2 to 5 external space is accessed, signals  $\overline{CS2}$  to  $\overline{CS5}$  can be output.

Basic bus interface, or DRAM interface can be selected for areas 2 to 5. With the DRAM interface, signals  $\overline{CS2}$  and  $\overline{CS3}$  are used as  $\overline{RAS}$  signals.

If areas 2 to 5 are designated as continuous DRAM space, large-capacity (e.g. 64-Mbit) DRAM can be connected. In this case, the  $\overline{CS2}$  signal is used as the  $\overline{RAS}$  signal for the continuous DRAM space.

**Area 6:** In externally expanded mode, all of area 6 is external space.

When area 6 external space is accessed, the  $\overline{CS6}$  signal can be output.



**Area 7:** Area 7 includes the on-chip RAM and internal I/O registers. In externally expanded mode, the space excluding the on-chip RAM and internal I/O registers is external space. The on-chip RAM is enabled when the RAME bit is set to 1 in the system control register (SYSCR); when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding addresses are in external space.

When area 7 external space is accessed, the  $\overline{CS7}$  signal can be output.

Only the basic bus interface can be used for the area 7 memory interface.

#### 6.4.4 Chip Select Signals

This LSI can output chip select signals ( $\overline{CS0}$  to  $\overline{CS7}$ ) for areas 0 to 7. The signal outputs low when the corresponding external space area is accessed. Figure 6.6 shows an example of  $\overline{CS0}$  to  $\overline{CS7}$  signals output timing.

Enabling or disabling of  $\overline{CS0}$  to  $\overline{CS7}$  signals output is set by the data direction register (DDR) bit for the port corresponding to the  $\overline{CS0}$  to  $\overline{CS7}$  pins.

In expanded mode with on-chip ROM disabled, the  $\overline{CS0}$  pin is placed in the output state after a reset. Pins  $\overline{CS1}$  to  $\overline{CS7}$  are placed in the input state after a reset and so the corresponding DDR bits should be set to 1 when outputting signals  $\overline{CS1}$  to  $\overline{CS7}$ .

In expanded mode with on-chip ROM enabled, pins  $\overline{CS0}$  to  $\overline{CS7}$  are all placed in the input state after a reset and so the corresponding DDR bits should be set to 1 when outputting signals  $\overline{CS0}$  to  $\overline{CS7}$ .

When areas 2 to 5 are designated as DRAM space, outputs  $\overline{CS2}$  is used as  $\overline{RAS}$  signals.

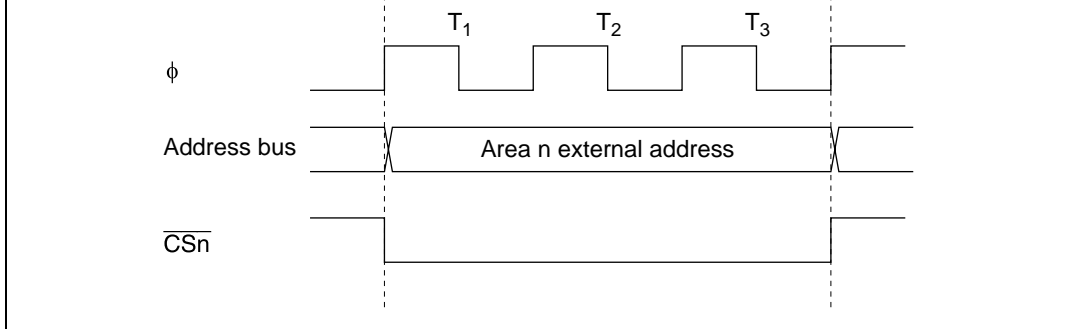


Figure 6.6  $\overline{CSn}$  Signal Output Timing (n = 0 to 7)

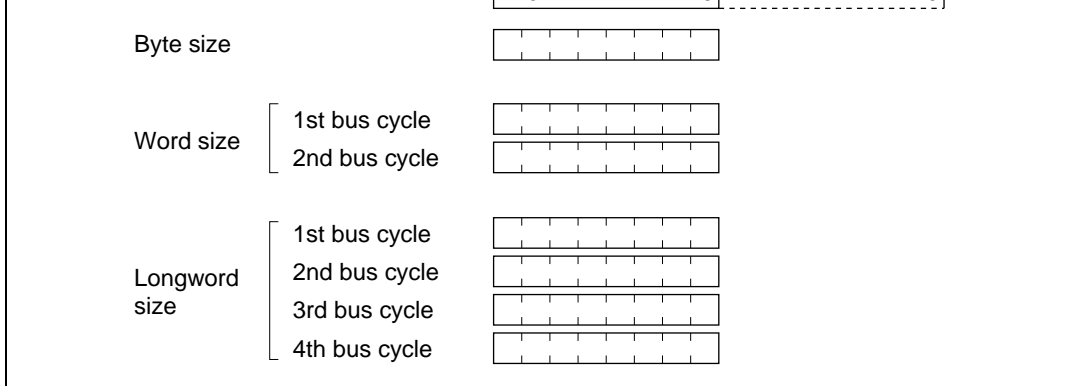
## 6.5 Basic Bus Interface

The basic bus interface enables direct connection of ROM, SRAM, and so on.

### 6.5.1 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The bus controller has a data alignment function, and when accessing external space, controls whether the upper data bus (D15 to D8) or lower data bus (D7 to D0) is used according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space) and the data size.

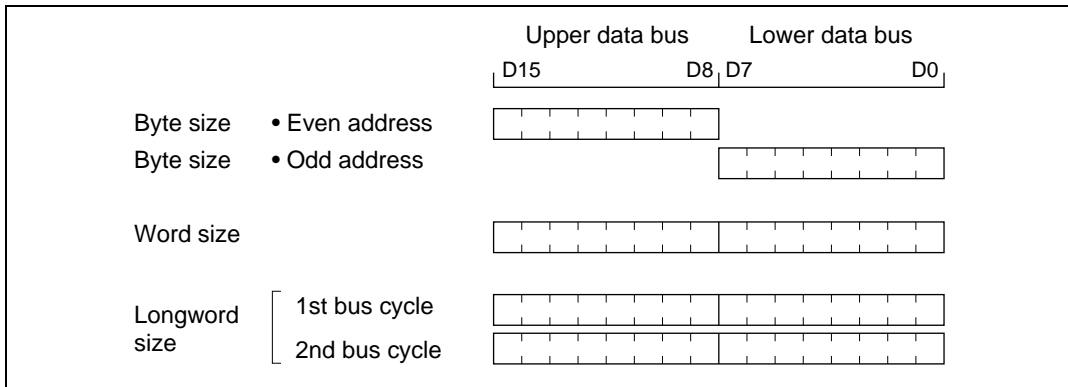
**8-Bit Access Space:** Figure 6.7 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word access is performed as two byte accesses, and a longword access, as four byte accesses.



**Figure 6.7 Access Sizes and Data Alignment Control (8-Bit Access Space)**

**16-Bit Access Space:** Figure 6.8 illustrates data alignment control for the 16-bit access space. With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to D0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword access is executed as two word accesses.

In byte access, whether the upper or lower data bus is used is determined by whether the address is even or odd. The upper data bus is used for an even address, and the lower data bus for an odd address.



**Figure 6.8 Access Sizes and Data Alignment Control (16-bit Access Space)**

Table 6.3 shows the data buses used and valid strobes for the access spaces.

In a read, the  $\overline{RD}$  signal is valid for both the upper and the lower half of the data bus. In a write, the  $\overline{HWR}$  signal is valid for the upper half of the data bus, and the  $\overline{LWR}$  signal for the lower half.

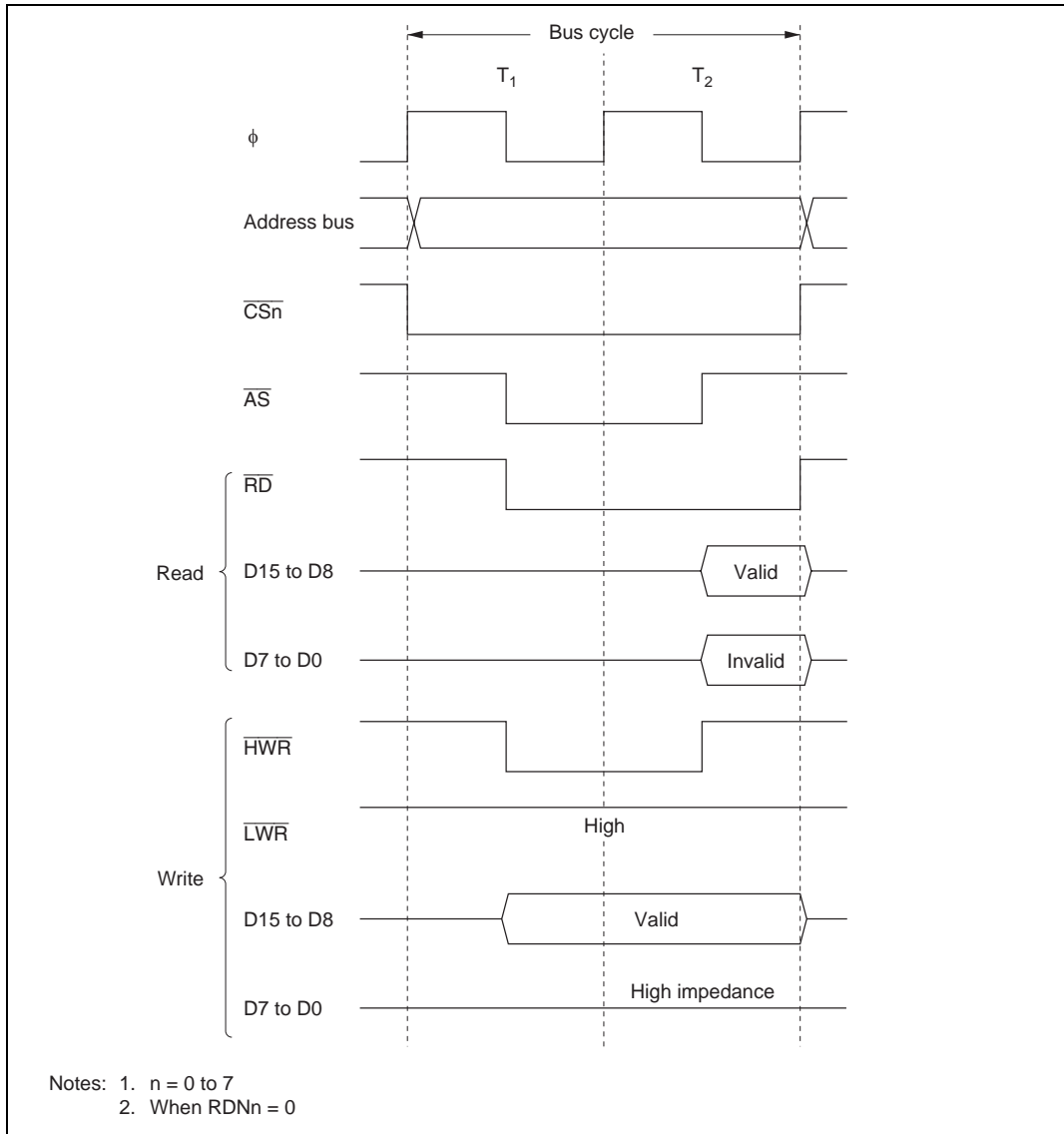
**Table 6.3 Data Buses Used and Valid Strobes**

Area	Access Size	Read/Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)	Lower Data Bus (D7 to D0)
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$		Hi-Z
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
	Write	Even	$\overline{HWR}$	Valid	Hi-Z	
		Odd	$\overline{LWR}$	Hi-Z	Valid	
Word	Read	—	$\overline{RD}$	Valid	Valid	
		Write	—	$\overline{HWR}, \overline{LWR}$	Valid	Valid

Notes: Hi-Z: High-impedance state

Invalid: Input state; input value is ignored.

**8-Bit, 2-State Access Space:** Figure 6.9 shows the bus timing for an 8-bit, 2-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used. The  $\overline{\text{LWR}}$  pin is always fixed high. Wait states can be inserted.



**Figure 6.9 Bus Timing for 8-Bit, 2-State Access Space**

LWR pin is always fixed high. Wait states can be inserted.

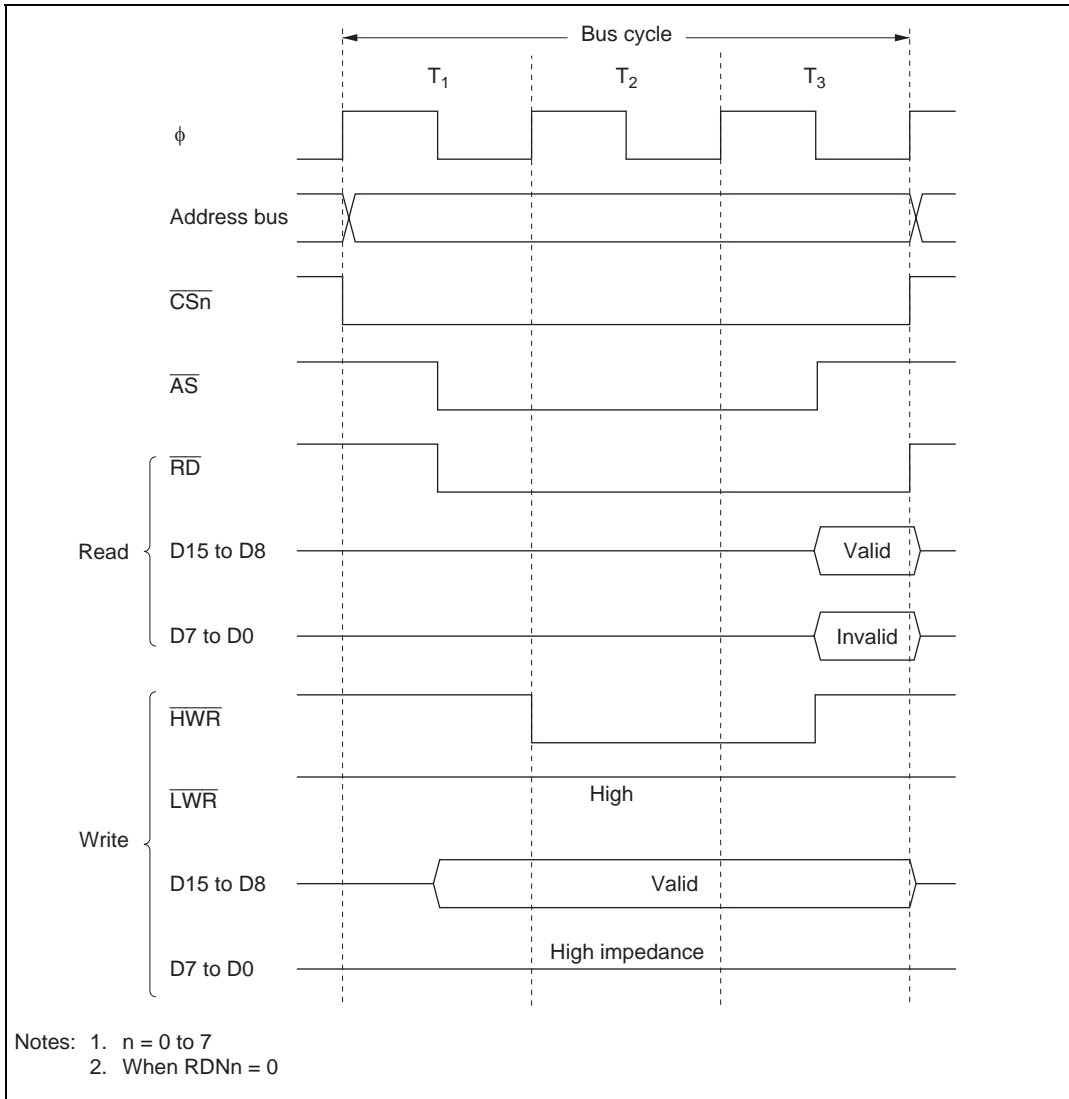
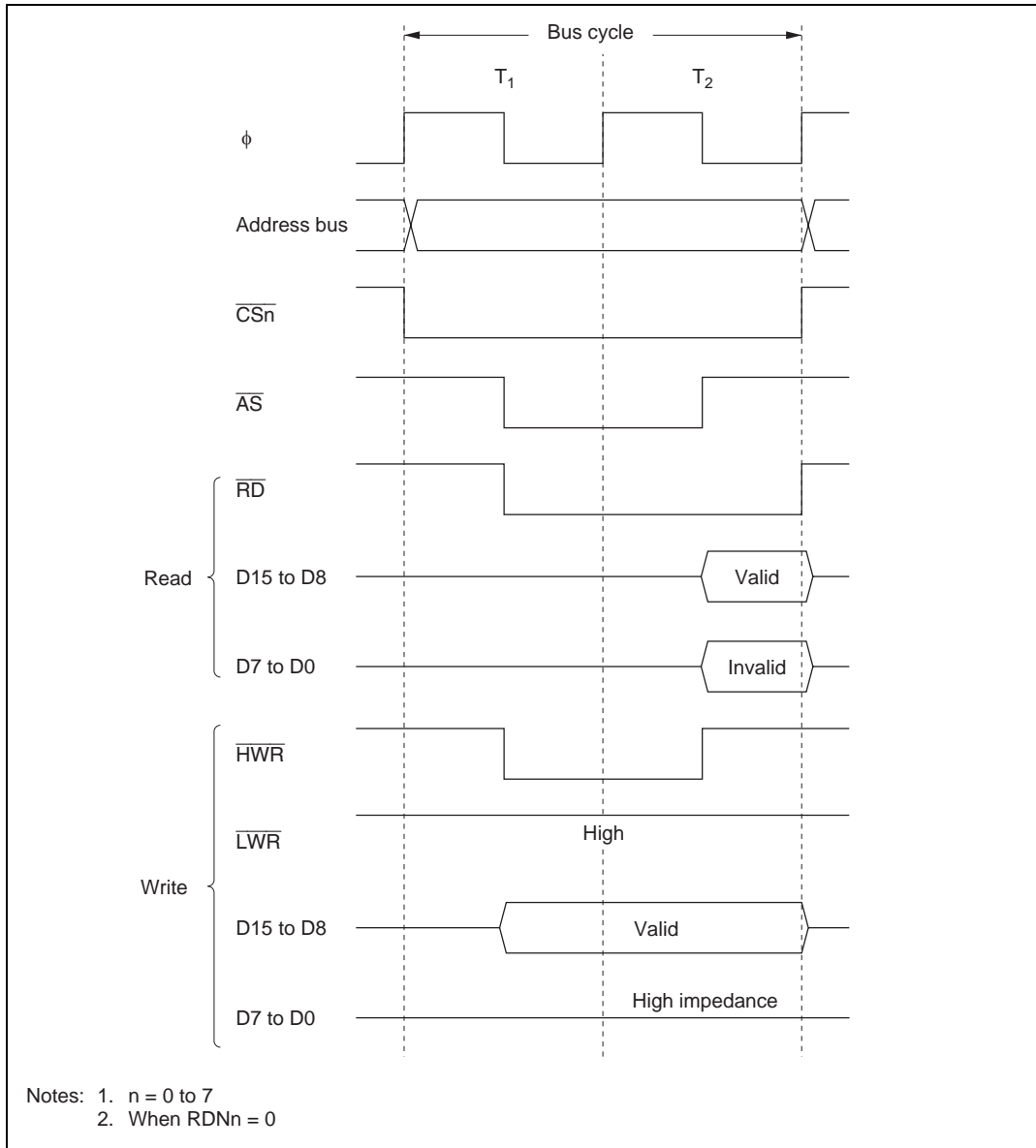
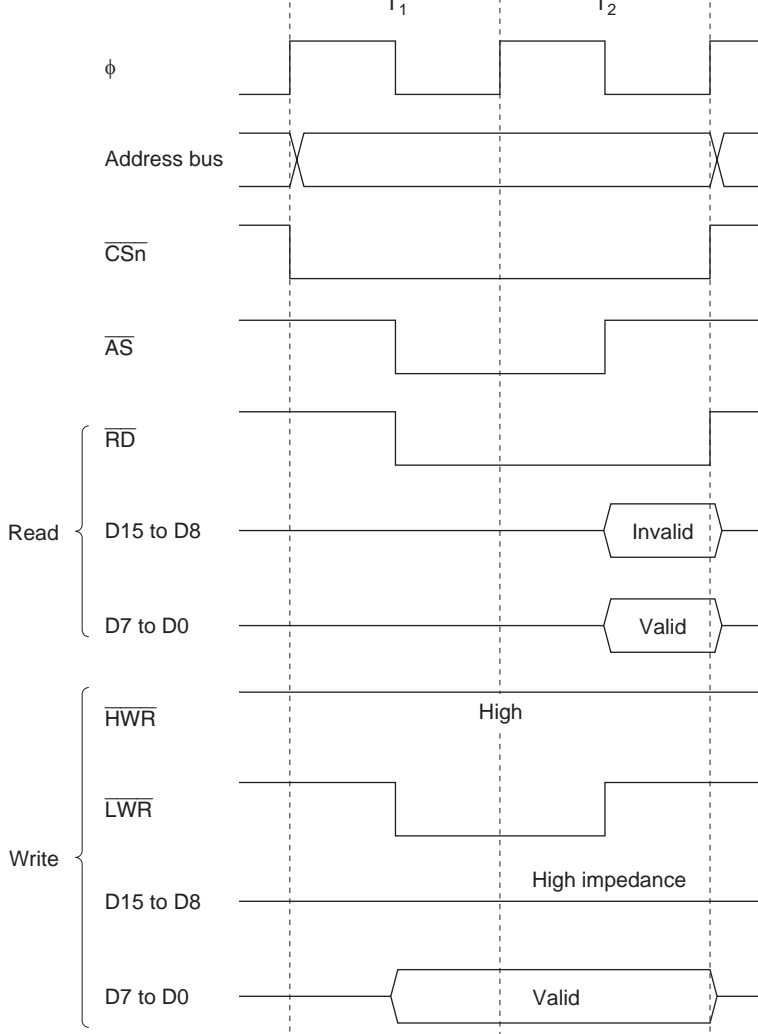


Figure 6.10 Bus Timing for 8-Bit, 3-State Access Space

for odd addresses, and the lower half (D7 to D0) for even addresses. Wait states cannot be inserted.



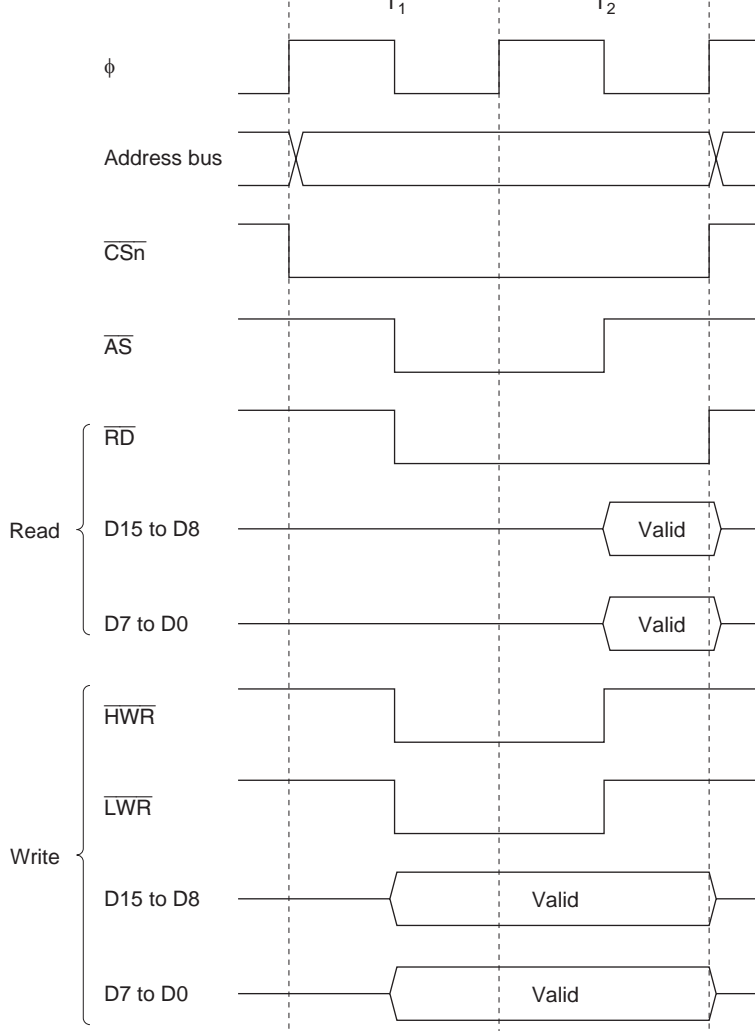
**Figure 6.11 Bus Timing for 16-Bit, 2-State Access Space (Even Address Byte Access)**



- Notes: 1.  $n = 0$  to 7  
 2. When  $RDNn = 0$

**Figure 6.12 Bus Timing for 16-Bit, 2-State Access Space (Odd Address Byte Access)**

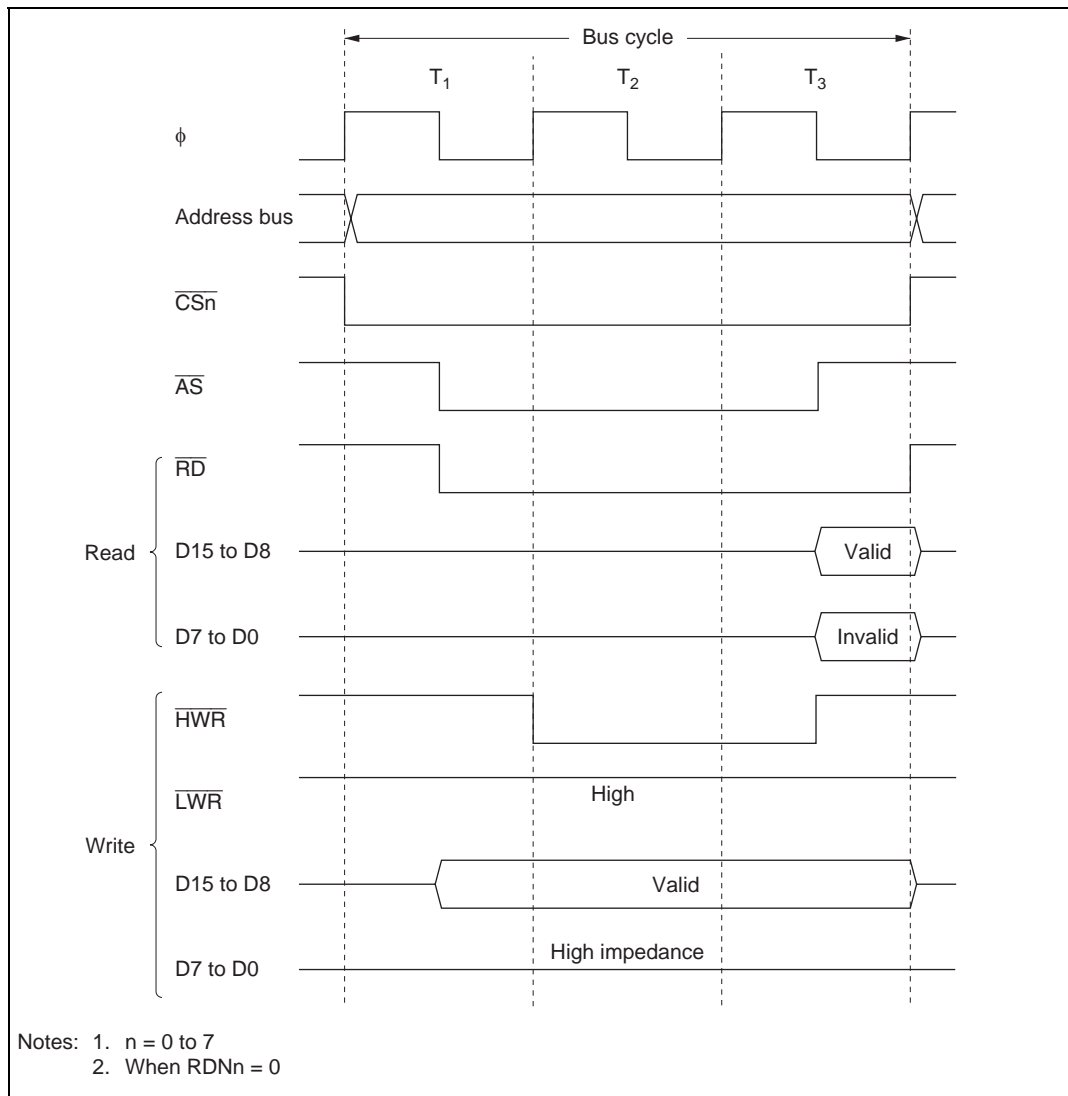




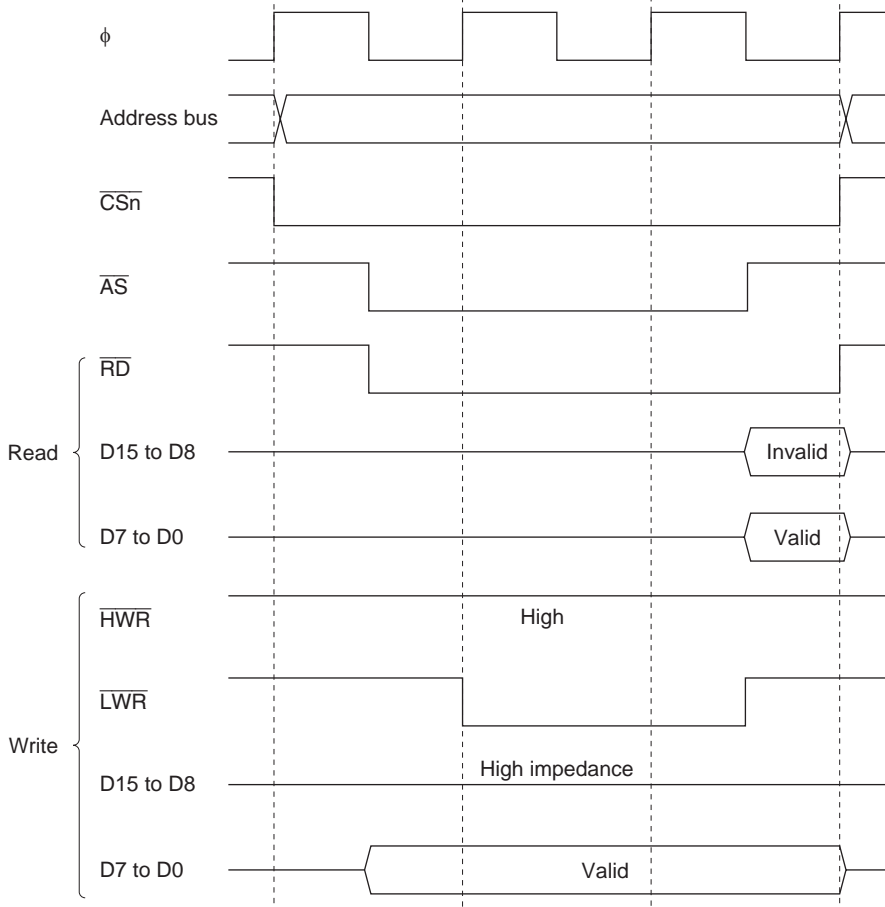
Notes: 1. n = 0 to 7  
 2. When RDNn = 0

**Figure 6.13 Bus Timing for 16-Bit, 2-State Access Space (Word Access)**

for the even address, and the lower half (D7 to D0) for the odd address. Wait states can be inserted.



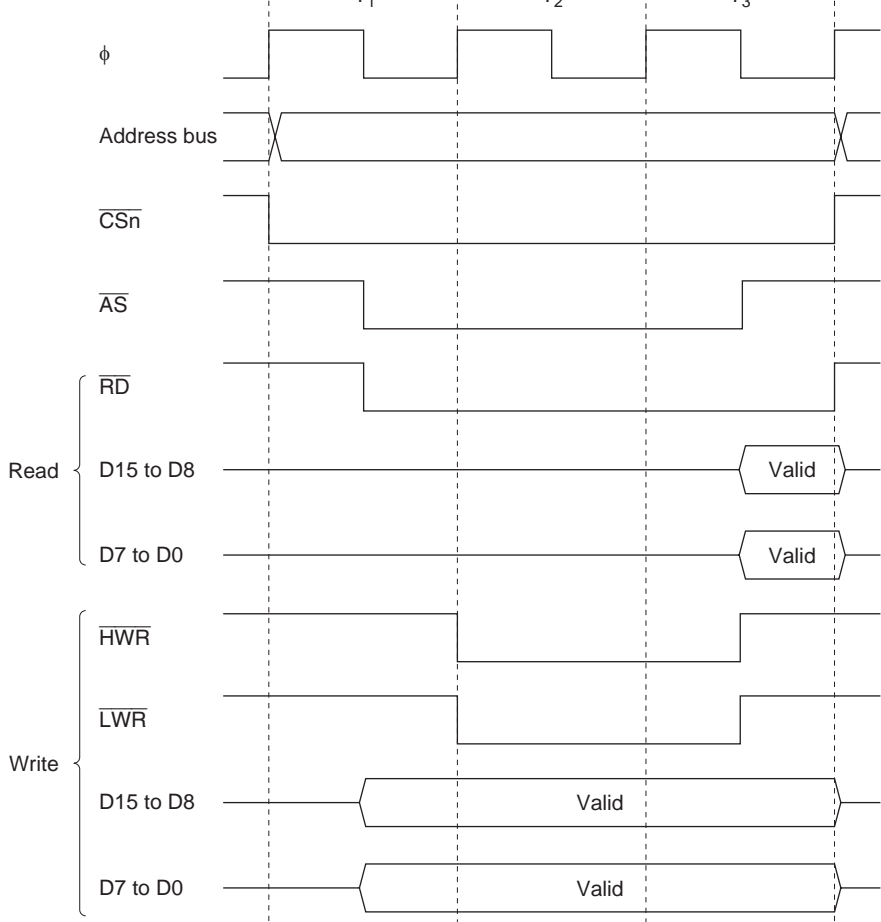
**Figure 6.14 Bus Timing for 16-Bit, 3-State Access Space (Even Address Byte Access)**



Notes: 1.  $n = 0$  to 7  
 2. When  $RDNn = 0$

**Figure 6.15 Bus Timing for 16-Bit, 3-State Access Space (Odd Address Byte Access)**





- Notes: 1.  $n = 0$  to 7  
 2. When  $RDNn = 0$

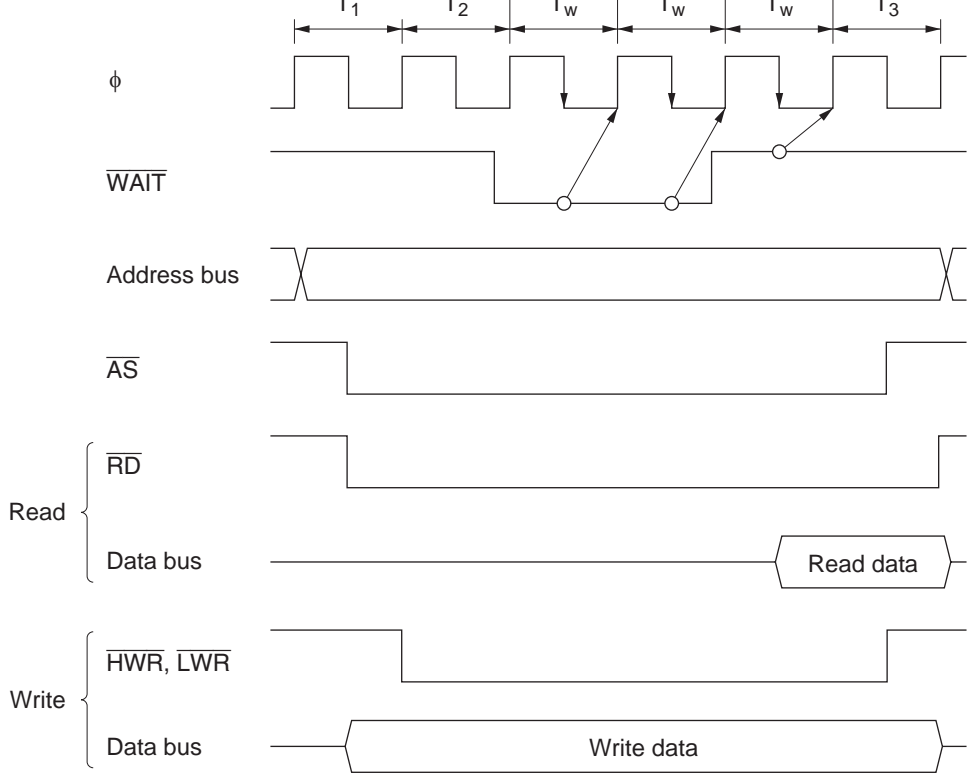
**Figure 6.16 Bus Timing for 16-Bit, 3-State Access Space (Word Access)**

When accessing external space, this LSI can extend the bus cycle by inserting one or more wait states ( $T_w$ ). There are two ways of inserting wait states: program wait insertion and pin wait insertion using the  $\overline{\text{WAIT}}$  pin.

**Program Wait Insertion:** From 0 to 7 wait states can be inserted automatically between the  $T_2$  state and  $T_3$  state on an individual area basis in 3-state access space, according to the settings in WTCRA and WTCRB.

**Pin Wait Insertion:** Setting the WAITE bit to 1 in BCR enables wait input by means of the  $\overline{\text{WAIT}}$  pin. When external space is accessed in this state, a program wait is first inserted in accordance with the settings in WTCRA and WTCRB. If the  $\overline{\text{WAIT}}$  pin is low at the falling edge of  $\phi$  in the last  $T_2$  or  $T_w$  state, another  $T_w$  state is inserted. If the  $\overline{\text{WAIT}}$  pin is held low,  $T_w$  states are inserted until it goes high. This is useful when inserting seven or more  $T_w$  states, or when changing the number of  $T_w$  states to be inserted for different external devices. The WAITE bit setting applies to all areas. Figure 6.17 shows an example of wait state insertion timing.

The settings after a reset are: 3-state access, insertion of 7 program wait states, and  $\overline{\text{WAIT}}$  input disabled.



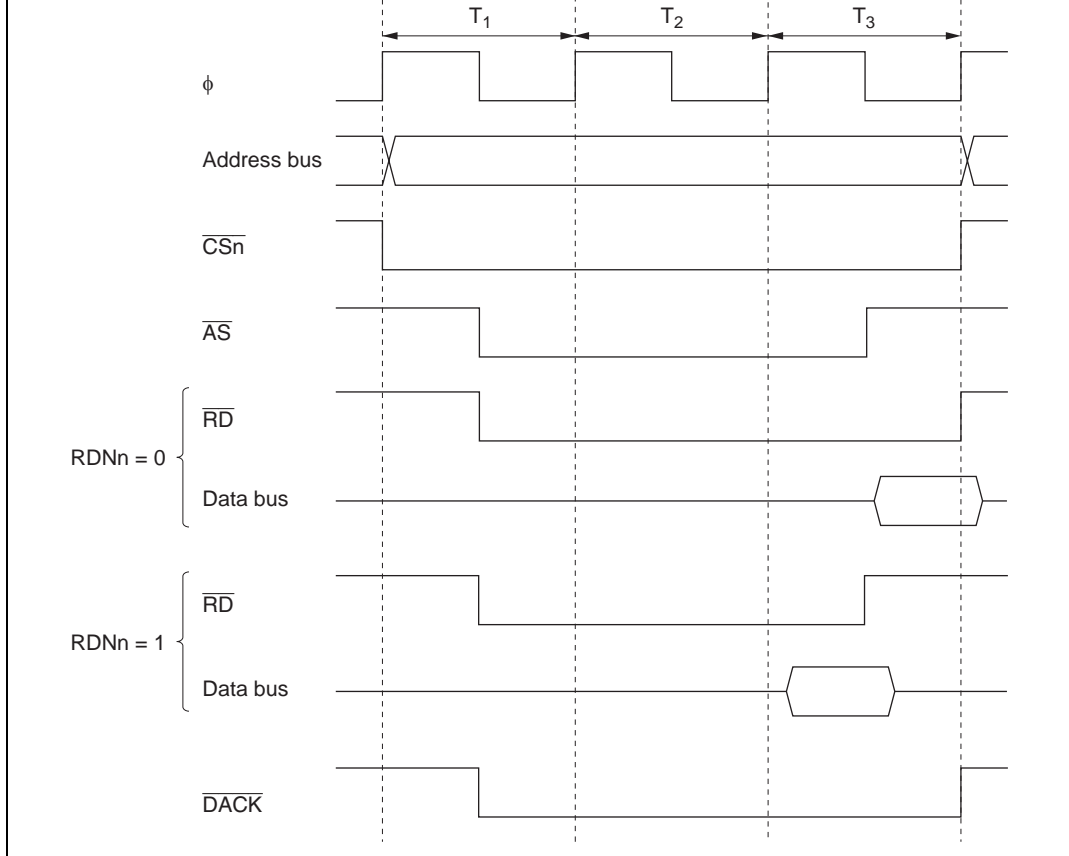
- Notes: 1. Downward arrows indicate the timing of  $\overline{\text{WAIT}}$  pin sampling.  
 2. When  $\text{RDN} = 0$

**Figure 6.17 Example of Wait State Insertion Timing**

### 6.5.5 Read Strobe ( $\overline{\text{RD}}$ ) Timing

The read strobe ( $\overline{\text{RD}}$ ) timing can be changed for individual areas by setting bits  $\text{RDN7}$  to  $\text{RDN0}$  to 1 in  $\text{RDNCR}$ . Figure 6.18 shows an example of the timing when the read strobe timing is changed in basic bus 3-state access space.

When the DMAC is used in single address mode, note that if the  $\overline{\text{RD}}$  timing is changed by setting  $\text{RDNn}$  to 1, the  $\overline{\text{RD}}$  timing will change relative to the rise of  $\overline{\text{DACK}}$ .

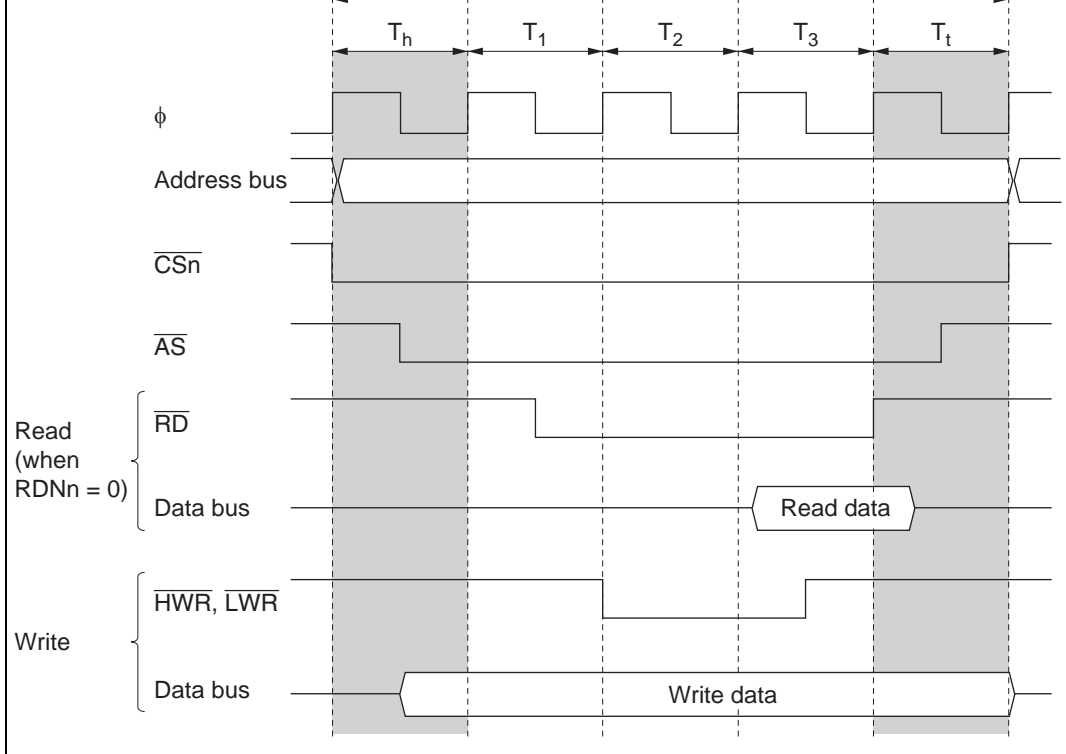


**Figure 6.18 Example of Read Strobe Timing**

### 6.5.6 Extension of Chip Select ( $\overline{CS}$ ) Assertion Period

Some external I/O devices require a setup time and hold time between address and  $\overline{CS}$  signals and strobe signals such as  $\overline{RD}$ ,  $\overline{HWR}$ , and  $\overline{LWR}$ . Settings can be made in the CSACR register to insert states in which only the  $\overline{CS}$ ,  $\overline{AS}$ , and address signals are asserted before and after a basic bus space access cycle. Extension of the  $\overline{CS}$  assertion period can be set for individual areas. With the  $\overline{CS}$  assertion extension period in write access, the data setup and hold times are less stringent since the write data is output to the data bus.

Figure 6.19 shows an example of the timing when the  $\overline{CS}$  assertion period is extended in basic bus 3-state access space.



**Figure 6.19 Example of Timing when Chip Select Assertion Period is Extended**

Both extension state  $T_h$  inserted before the basic bus cycle and extension state  $T_t$  inserted after the basic bus cycle, or only one of these, can be specified for individual areas. Insertion or non-insertion can be specified for the  $T_h$  state with the upper 8 bits ( $CSXH7$  to  $CSXH0$ ) in the  $CSACR$  register, and for the  $T_t$  state with the lower 8 bits ( $CSXT7$  to  $CSXT0$ ).



In this LSI, external space areas 2 to 5 can be designated as DRAM space, and DRAM interfacing performed. The DRAM interface allows DRAM to be directly connected to this LSI. A DRAM space of 2, 4, or 8 Mbytes can be set by means of bits RMTS2 to RMTS0 in DRAMCR. Burst operation is also possible, using fast page mode.

### 6.6.1 Setting DRAM Space

Areas 2 to 5 are designated as DRAM space by setting bits RMTS2 to RMTS0 in DRAMCR. The relation between the settings of bits RMTS2 to RMTS0 and DRAM space is shown in table 6.4. Possible DRAM space settings are: one area (area 2), two areas (areas 2 and 3), and continuous area (areas 2 to 5).

**Table 6.4 Relation between Settings of Bits RMTS2 to RMTS0 and DRAM Space**

RMTS2	RMTS1	RMTS0	Area 5	Area 4	Area 3	Area 2
0	0	1	Normal space	Normal space	Normal space	DRAM space
		0	Normal space	Normal space	DRAM space	DRAM space
	1	Reserved (setting prohibited)				
1	0	0	Reserved (setting prohibited)			
		1	Reserved (setting prohibited)			
	1	0	Reserved (setting prohibited)			
		1	Continuous DRAM space	Continuous DRAM space	Continuous DRAM space	Continuous DRAM space

With continuous DRAM space,  $\overline{\text{RAS2}}$  is valid. The bus specifications (bus width, number of wait states, etc.) for continuous DRAM space conform to the settings for area 2.

### 6.6.2 Address Multiplexing

With DRAM space, the row address and column address are multiplexed. In address multiplexing, the size of the shift of the row address is selected with bits MXC2 to MXC0 in DRAMCR. Table 6.5 shows the relation between the settings of MXC2 to MXC0 and the shift size.

The MXC2 bit should be cleared to 0 when the DRAM interface is used.

				A23 to A16	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
	MXC2	MXC1	MXC0	Shift Size																	
Row address	0	0	0	8 bits	A23	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
			1	9 bits	A23	A15	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9
		1	0	10 bits	A23	A15	A14	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10
			1	11 bits	A23	A15	A14	A13	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
	1	×	×	Reserved (setting prohibited)																	
Column address	0	×	×	—	A23	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
	1	×	×	Reserved (setting prohibited)																	

Legend:

×: Don't care.

### 6.6.3 Data Bus

If a bit in ABWCR corresponding to an area designated as DRAM space is set to 1, that area is designated as 8-bit DRAM space; if the bit is cleared to 0, the area is designated as 16-bit DRAM space. In 16-bit DRAM space, ×16-bit configuration DRAM can be connected directly.

In 8-bit DRAM space the upper half of the data bus, D15 to D8, is enabled, while in 16-bit DRAM space both the upper and lower halves of the data bus, D15 to D0, are enabled.

Access sizes and data alignment are the same as for the basic bus interface: see section 6.5.1, Data Size and Data Alignment.

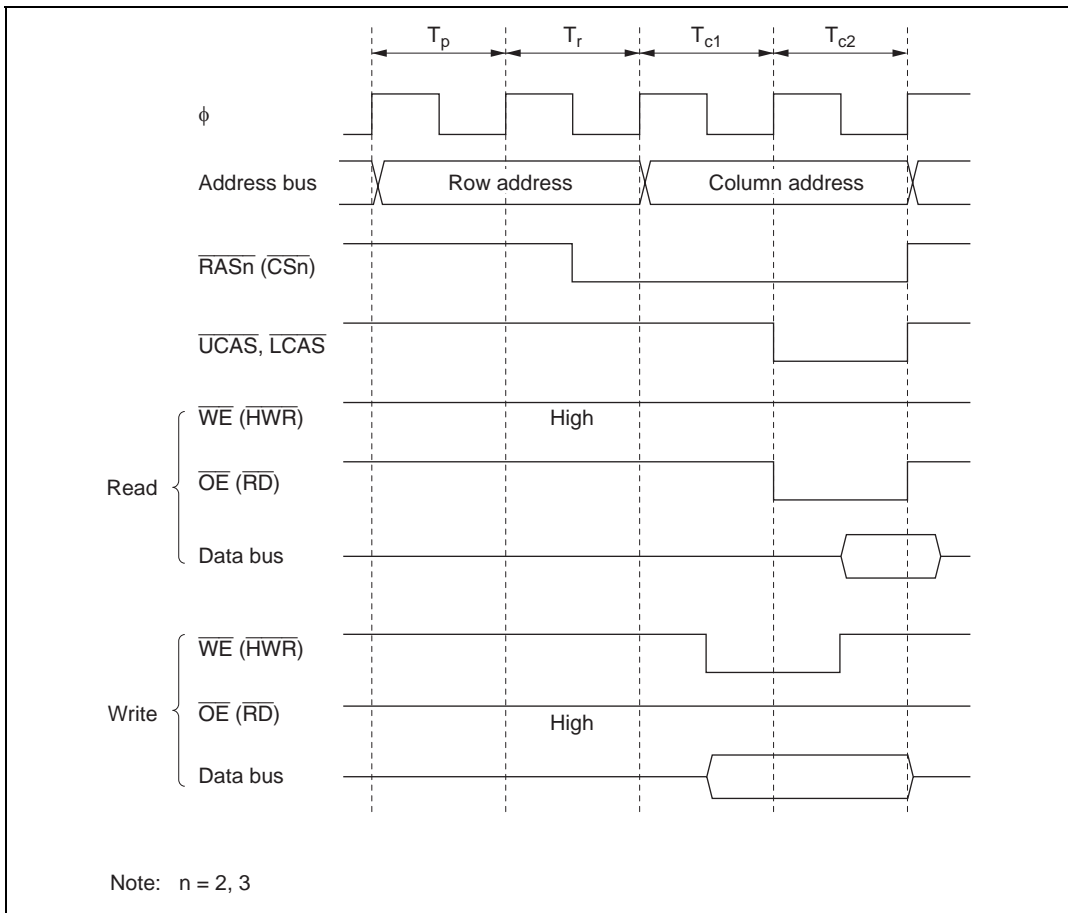
Table 6.6 shows the pins used for DRAM interfacing and their functions. Since the  $\overline{CS2}$ ,  $\overline{CS5}$  pins are in the input state after a reset, set the corresponding DDR to 1 when  $\overline{RAS2}$ ,  $\overline{RAS5}$  signals are output.

**Table 6.6 DRAM Interface Pins**

Pin	With DRAM Setting	Name	I/O	Function
$\overline{HWR}$	$\overline{WE}$	Write enable	Output	Write enable for DRAM space access
$\overline{CS2}$	$\overline{RAS2}$	Row address strobe 2/ row address strobe	Output	Row address strobe when area 2 is designated as DRAM space or row address strobe when areas 2 to 5 are designated as continuous DRAM space
$\overline{CS3}$	$\overline{RAS3}$	Row address strobe 3	Output	Row address strobe when area 3 is designated as DRAM space
$\overline{UCAS}$	$\overline{UCAS}$	Upper column address strobe	Output	Upper column address strobe for 16-bit DRAM space access or column address strobe for 8-bit DRAM space access
$\overline{LCAS}$	$\overline{LCAS}$	Lower column address strobe	Output	Lower column address strobe signal for 16-bit DRAM space access
$\overline{RD}$ , $\overline{OE}$	$\overline{OE}$	Output enable	Output	Output enable signal for DRAM space access
$\overline{WAIT}$	$\overline{WAIT}$	Wait	Input	Wait request signal
A15 to A0	A15 to A0	Address pins	Output	Row address/column address multiplexed output
D15 to D0	D15 to D0	Data pins	I/O	Data input/output pins

Figure 6.20 shows the basic access timing for DRAM space.

The four states of the basic timing consist of one  $T_p$  (precharge cycle) state, one  $T_r$  (row address output cycle) state, and the  $T_{c1}$  and two  $T_{c2}$  (column address output cycle) states.

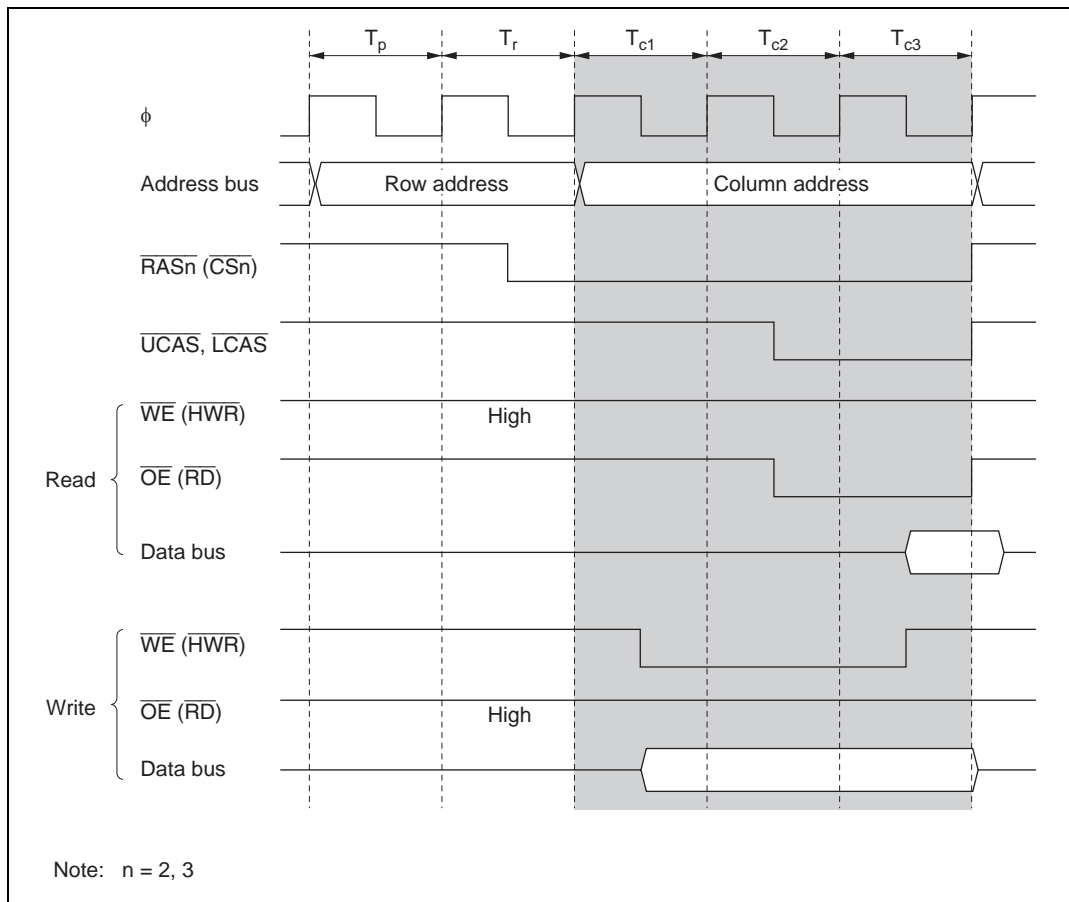


**Figure 6.20 DRAM Basic Access Timing (RAST = 0, CAST = 0)**

When DRAM space is accessed, the  $\overline{RD}$  signal is output as the  $\overline{OE}$  signal for DRAM. When connecting DRAM provided with an EDO page mode, the  $\overline{OE}$  signal should be connected to the ( $\overline{OE}$ ) pin of the DRAM. Setting the OEE bit to 1 in DRAMCR enables the  $\overline{OE}$  signal for DRAM space to be output from a dedicated  $\overline{OE}$  pin. In this case, the  $\overline{OE}$  signal for DRAM space is output

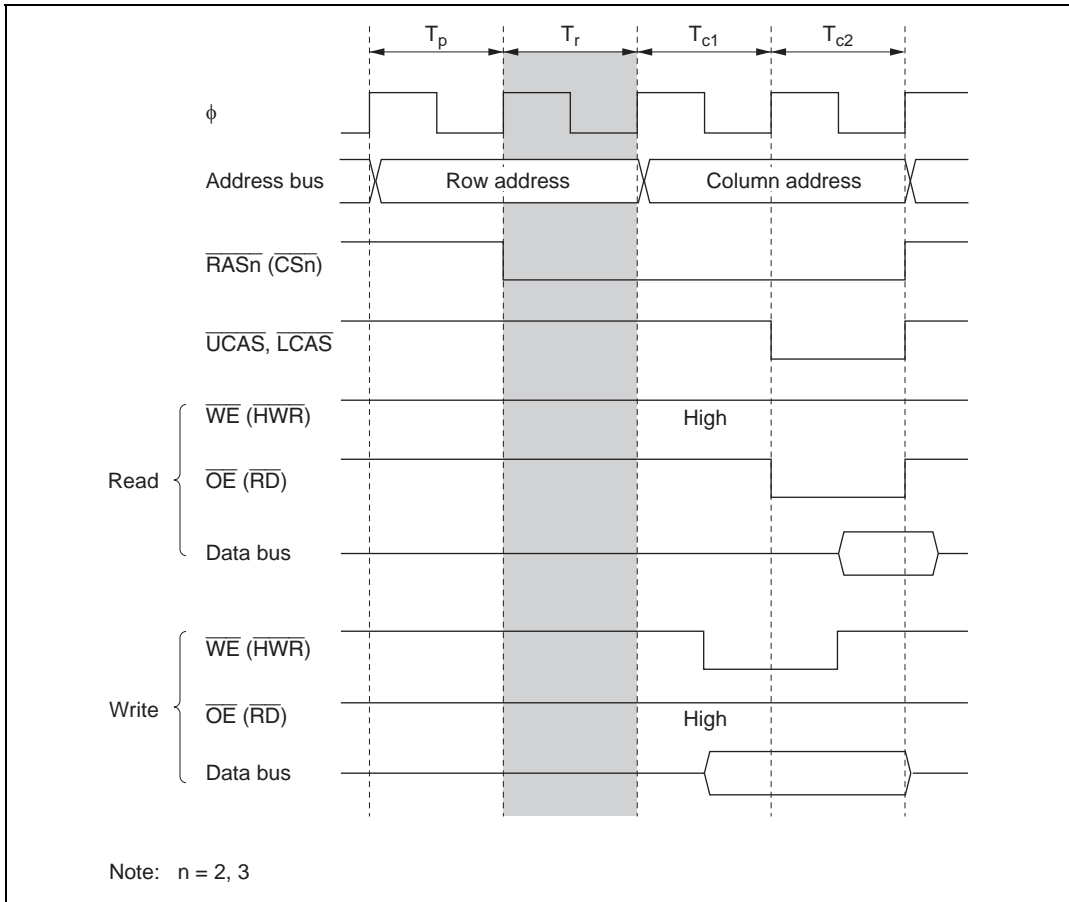
## 6.6.6 Column Address Output Cycle Control

The column address output cycle can be changed from 2 states to 3 states by setting the CAST bit to 1 in DRAMCR. Use the setting that gives the optimum specification values ( $\overline{\text{CAS}}$  pulse width, etc.) according to the DRAM connected and the operating frequency of this LSI. Figure 6.21 shows an example of the timing when a 3-state column address output cycle is selected.



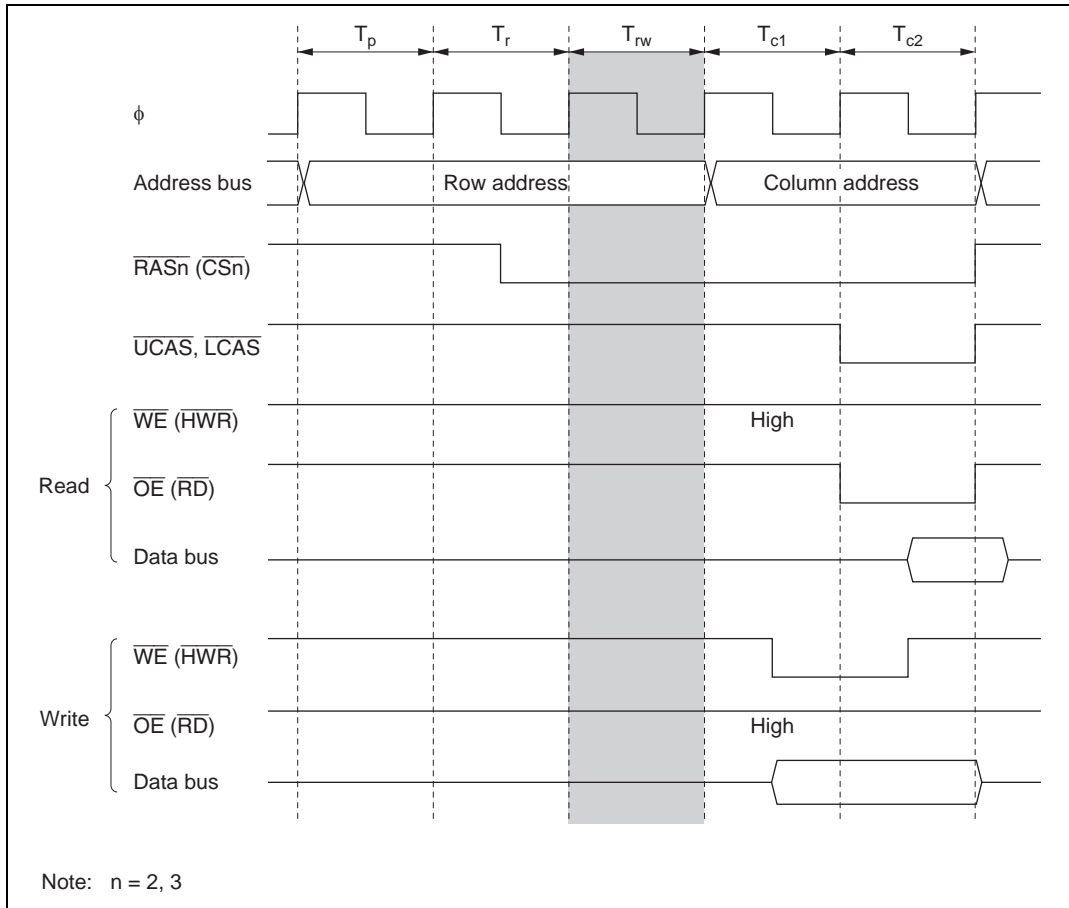
**Figure 6.21 Example of Access Timing with 3-State Column Address Output Cycle (RAST = 0)**

If the RAST bit is set to 1 in DRAMCR, the RAS signal goes low from the beginning of the  $T_r$  state, and the row address hold time and DRAM read access time are changed relative to the fall of the  $\overline{\text{RAS}}$  signal. Use the optimum setting according to the DRAM connected and the operating frequency of this LSI. Figure 6.22 shows an example of the timing when the  $\overline{\text{RAS}}$  signal goes low from the beginning of the  $T_r$  state.



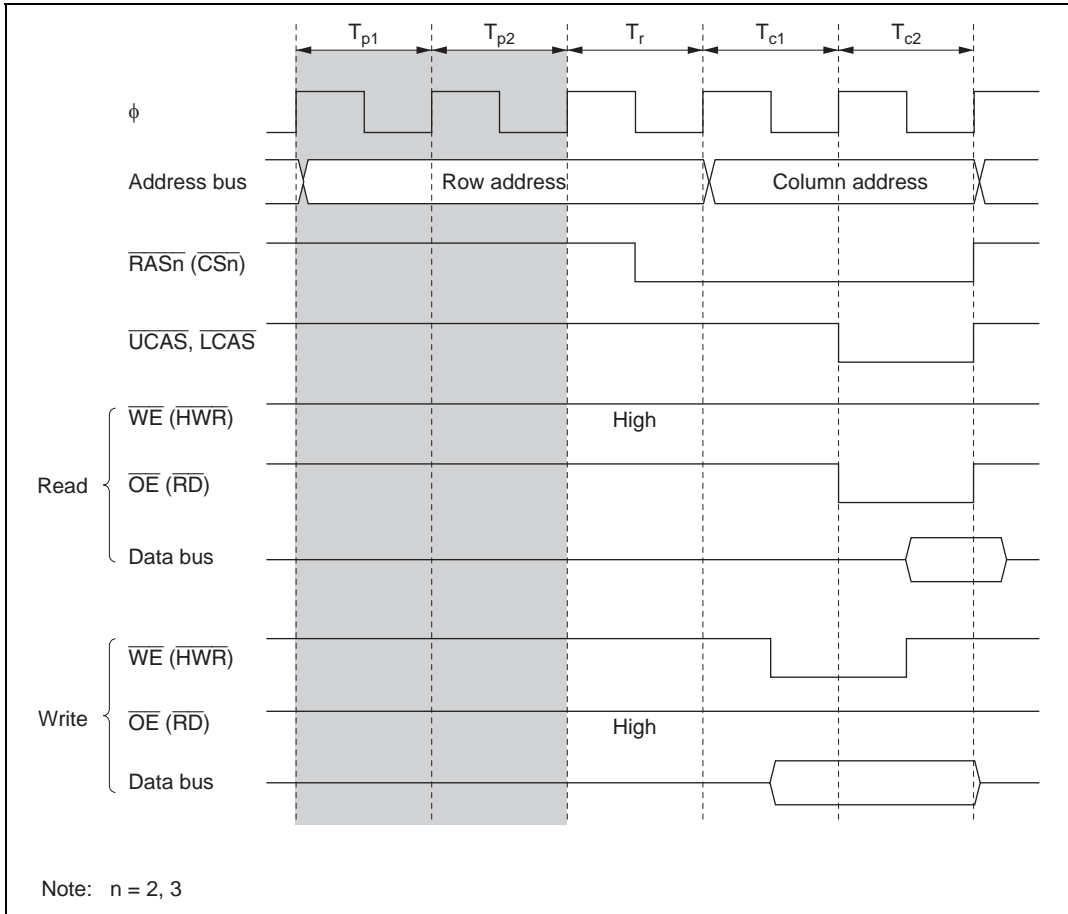
**Figure 6.22 Example of Access Timing when  $\overline{\text{RAS}}$  Signal Goes Low from Beginning of  $T_r$  State (CAST = 0)**

to be inserted between the  $T_r$  cycle, in which the RAS signal goes low, and the  $T_{c1}$  cycle, in which the column address is output. Use the setting that gives the optimum row address signal hold time relative to the falling edge of the  $\overline{\text{RAS}}$  signal according to the DRAM connected and the operating frequency of this LSI. Figure 6.23 shows an example of the timing when one  $T_{rw}$  state is set.



**Figure 6.23 Example of Timing with One Row Address Output Maintenance State (RAST = 0, CAST = 0)**

When DRAM is accessed, a  $\overline{\text{RAS}}$  precharge time must be secured. With this LSI, one  $T_p$  state is always inserted when DRAM space is accessed. From one to four  $T_p$  states can be selected by setting bits TPC1 and TPC0 in DRACCR. Set the optimum number of  $T_p$  cycles according to the DRAM connected and the operating frequency of this LSI. Figure 6.24 shows the timing when two  $T_p$  states are inserted. The setting of bits TPC1 and TPC0 is also valid for  $T_p$  states in refresh cycles.



**Figure 6.24 Example of Timing with Two-State Precharge Cycle  
( $\text{RAST} = 0, \text{CAST} = 0$ )**



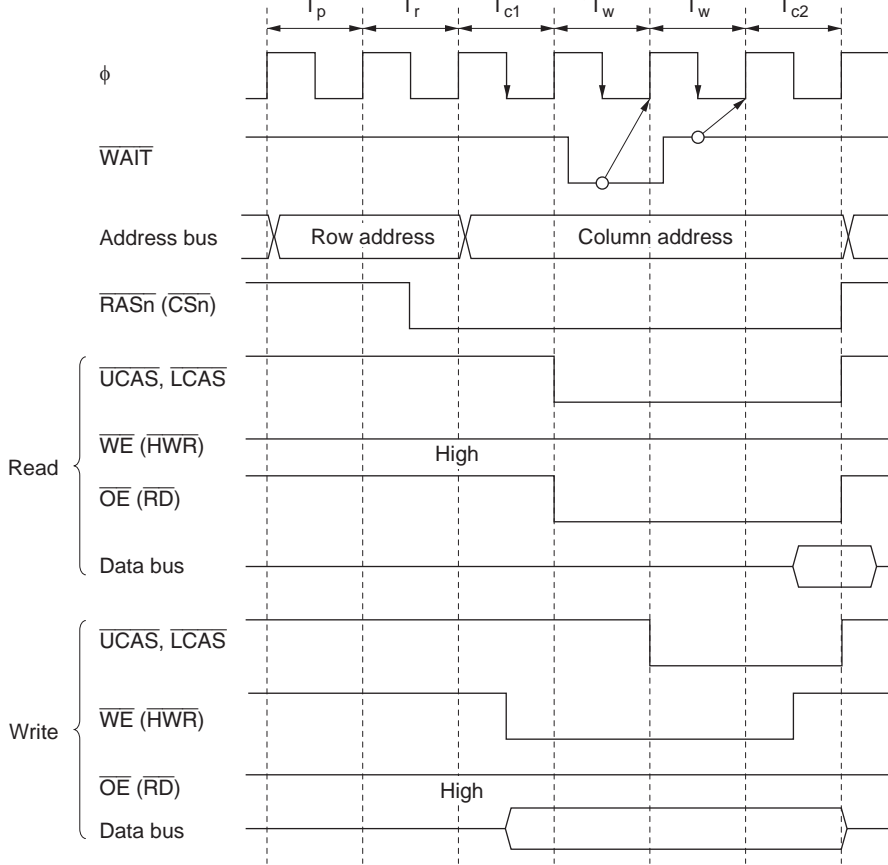
There are two ways of inserting wait states in a DRAM access cycle: program wait insertion and pin wait insertion using the  $\overline{\text{WAIT}}$  pin.

Wait states are inserted to extend the  $\overline{\text{CAS}}$  assertion period in a read access to DRAM space, and to extend the write data setup time relative to the falling edge of  $\overline{\text{CAS}}$  in a write access.

**Program Wait Insertion:** When the bit in ASTCR corresponding to an area designated as DRAM space is set to 1, from 0 to 7 wait states can be inserted automatically between the  $T_{c1}$  state and  $T_{c2}$  state, according to the WTCR setting.

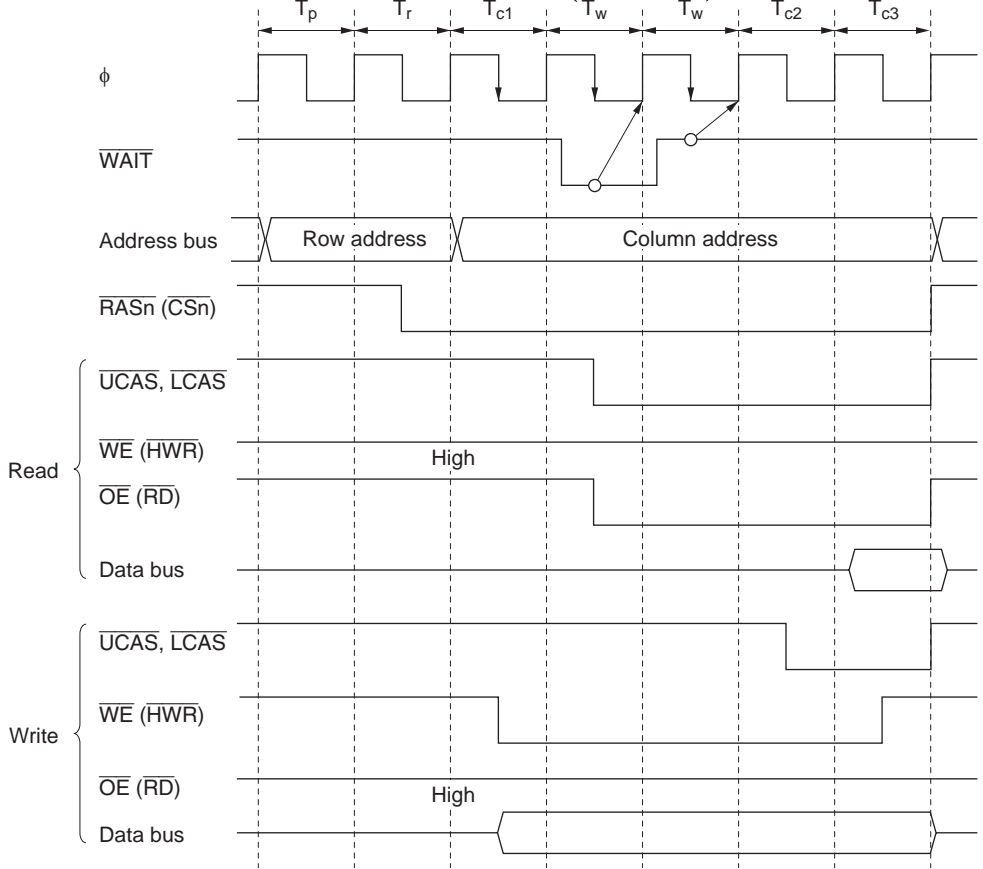
**Pin Wait Insertion:** When the WAITE bit in BCR is set to 1 and the ASTCR bit is set to 1, wait input by means of the  $\overline{\text{WAIT}}$  pin is enabled. When DRAM space is accessed in this state, a program wait ( $T_w$ ) is first inserted. If the  $\overline{\text{WAIT}}$  pin is low at the falling edge of  $\phi$  in the last  $T_{c1}$  or  $T_w$  state, another  $T_w$  state is inserted. If the  $\overline{\text{WAIT}}$  pin is held low,  $T_w$  states are inserted until it goes high.

Figures 6.25 and 6.26 show examples of wait cycle insertion timing in the case of 2-state and 3-state column address output cycles.



Notes: Downward arrows indicate the timing of  $\overline{\text{WAIT}}$  pin sampling.  
 n = 2, 3

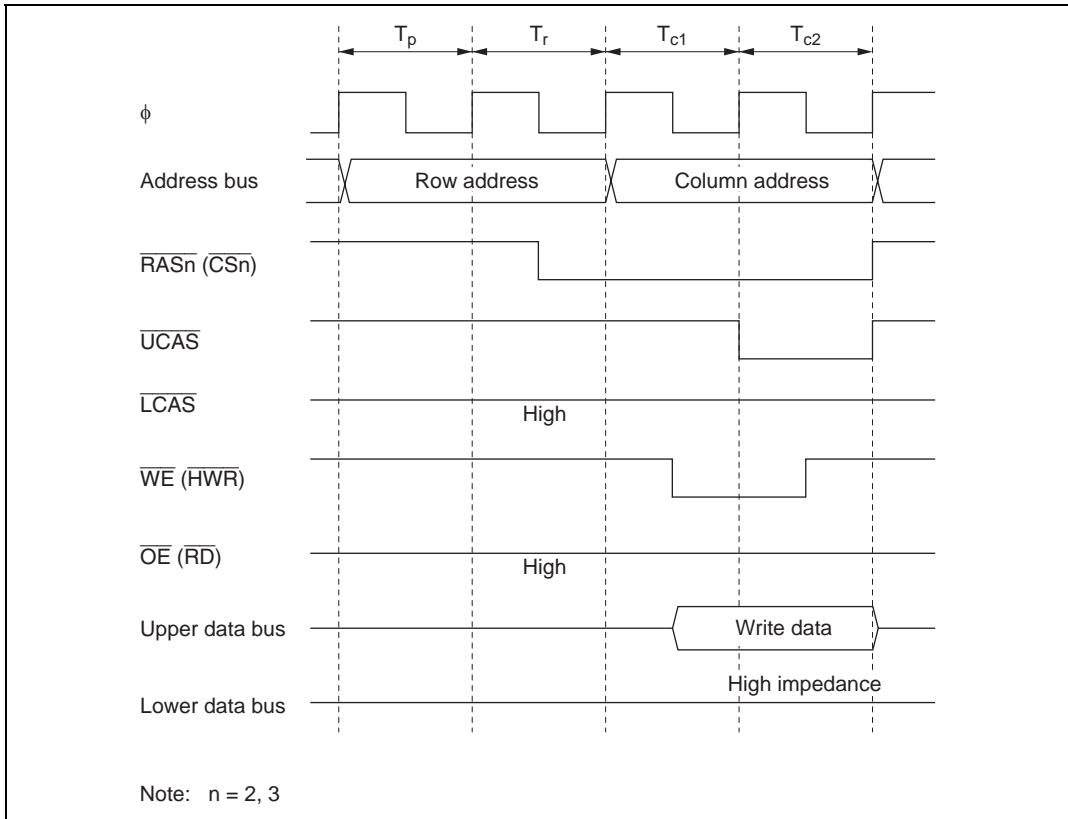
**Figure 6.25 Example of Wait State Insertion Timing  
 (2-State Column Address Output)**



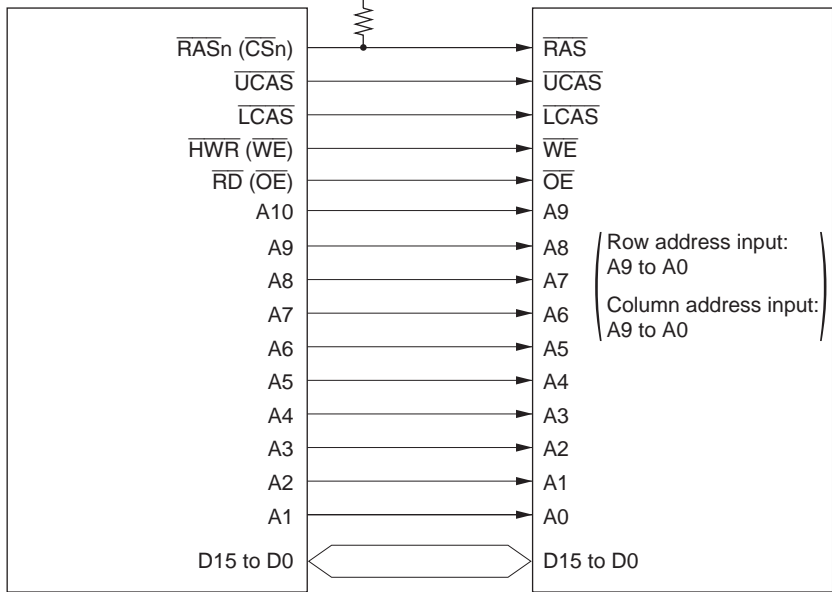
Notes: Downward arrows indicate the timing of  $\overline{\text{WAIT}}$  pin sampling.  
 $n = 2, 3$

**Figure 6.26 Example of Wait State Insertion Timing  
 (3-State Column Address Output)**

When DRAM with a  $\times 16$ -bit configuration is connected, the 2-CAS access method is used for the control signals needed for byte access. Figure 6.27 shows the control timing for 2-CAS access, and figure 6.28 shows an example of 2-CAS DRAM connection.



**Figure 6.27 2-CAS Control Timing**  
**(Upper Byte Write Access: RAST = 0, CAST = 0)**

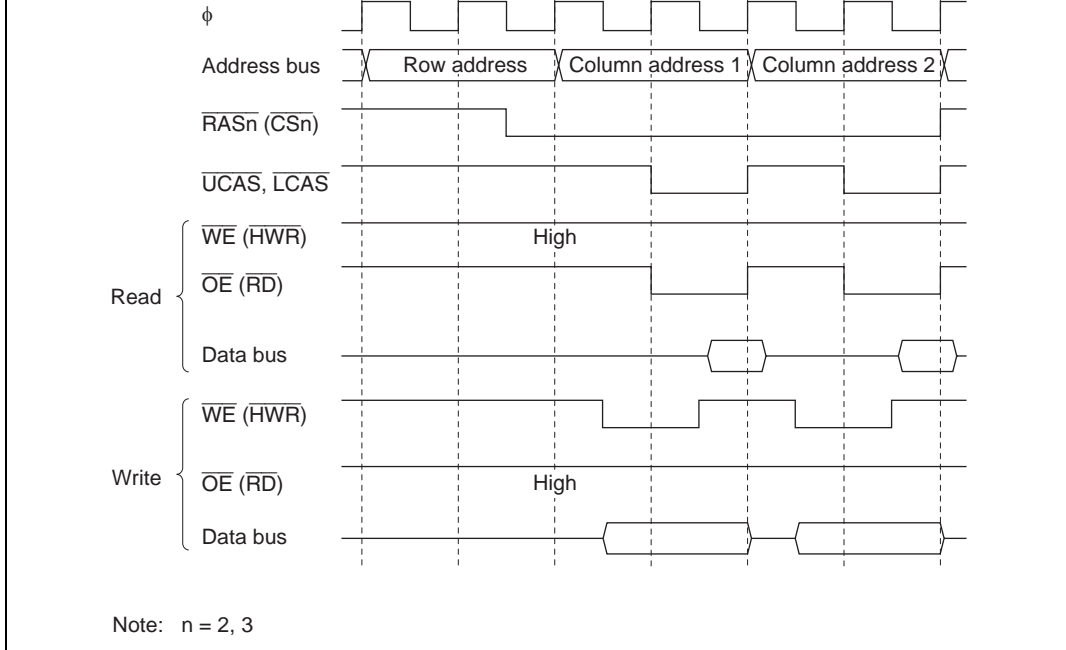


**Figure 6.28 Example of 2-CAS DRAM Connection**

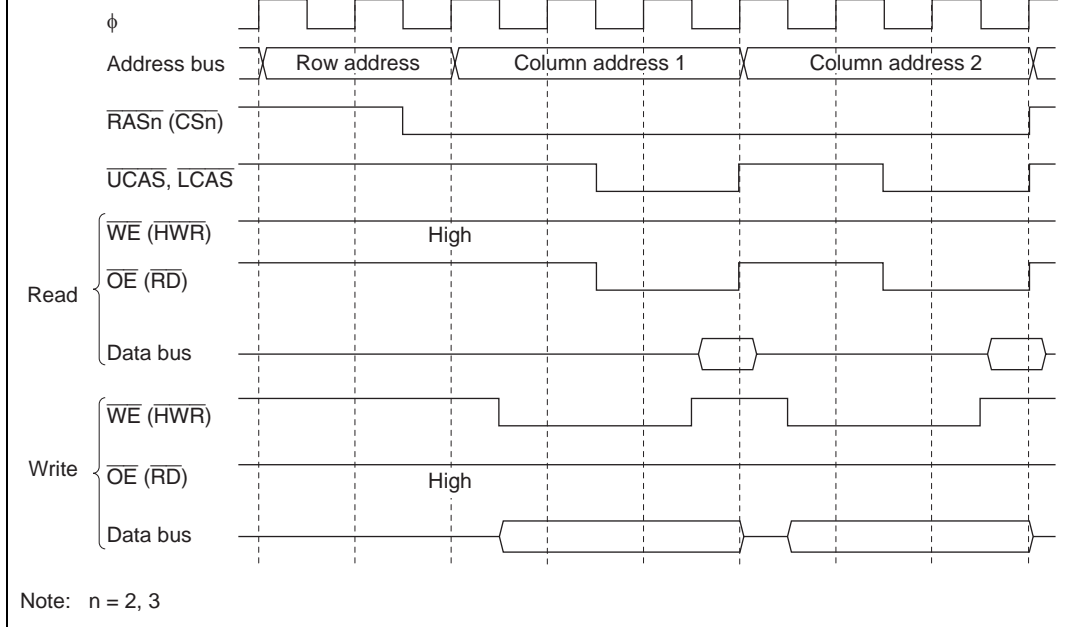
### 6.6.11 Burst Operation

With DRAM, in addition to full access (normal access) in which data is accessed by outputting a row address for each access, a fast page mode is also provided which can be used when making consecutive accesses to the same row address. This mode enables fast (burst) access of data by simply changing the column address after the row address has been output. Burst access can be selected by setting the BE bit to 1 in DRAMCR.

**Burst Access (Fast Page Mode):** Figures 6.29 and 6.30 show the operation timing for burst access. When there are consecutive access cycles for DRAM space, the  $\overline{CAS}$  signal and column address output cycles (two states) continue as long as the row address is the same for consecutive access cycles. The row address used for the comparison is set with bits MXC2 to MXC0 in DRAMCR.



**Figure 6.29 Operation Timing in Fast Page Mode  
(RAST = 0, CAST = 0)**



**Figure 6.30 Operation Timing in Fast Page Mode  
(RAST = 0, CAST = 1)**

The bus cycle can also be extended in burst access by inserting wait states. The wait state insertion method and timing are the same as for full access. For details see section 6.6.9, Wait Control.

**RAS Down Mode and RAS Up Mode:** Even when burst operation is selected, it may happen that access to DRAM space is not continuous, but is interrupted by access to another space. In this case, if the  $\overline{RAS}$  signal is held low during the access to the other space, burst operation can be resumed when the same row address in DRAM space is accessed again.

- **RAS Down Mode**

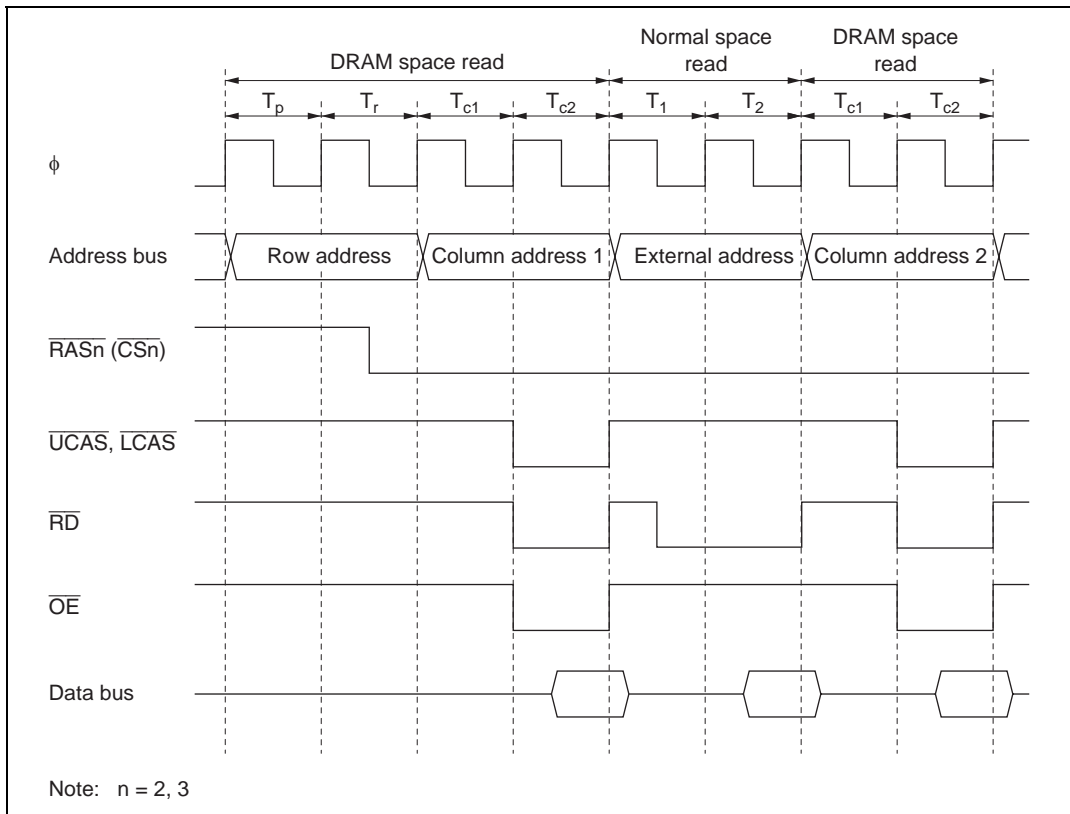
To select RAS down mode, set both the RCDM bit and the BE bit to 1 in DRAMCR. If access to DRAM space is interrupted and another space is accessed, the  $\overline{RAS}$  signal is held low during the access to the other space, and burst access is performed when the row address of the next DRAM space access is the same as the row address of the previous DRAM space access. Figure 6.31 shows an example of the timing in RAS down mode.

Note, however, that the  $\overline{RAS}$  signal will go high if:

- a refresh operation is initiated in the RAS down state
- self-refreshing is performed

— the RCDM bit or BE bit is cleared to 0

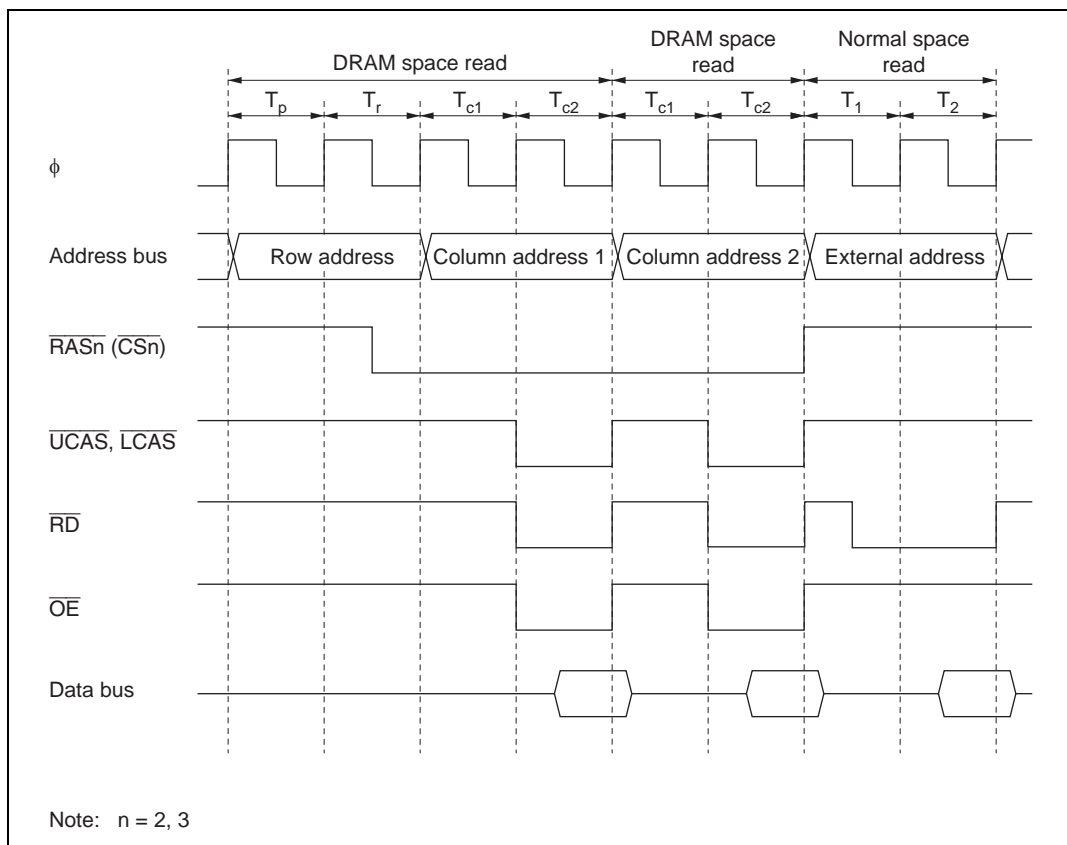
If a transition is made to the all-module-clocks-stopped mode in the  $\overline{\text{RAS}}$  down state, the clock will stop with  $\overline{\text{RAS}}$  low. To enter the all-module-clocks-stopped mode with  $\overline{\text{RAS}}$  high, the RCDM bit must be cleared to 0 before executing the SLEEP instruction.



**Figure 6.31 Example of Operation Timing in RAS Down Mode  
( $\text{RAST} = 0, \text{CAST} = 0$ )**



space is interrupted and another space is accessed, the RAS signal goes high again. Burst operation is only performed if DRAM space is continuous. Figure 6.32 shows an example of the timing in RAS up mode.



**Figure 6.32 Example of Operation Timing in RAS Up Mode  
(RAST = 0, CAST = 0)**

This LSI is provided with a DRAM refresh control function. CAS-before-RAS (CBR) refreshing is used. In addition, self-refreshing can be executed when the chip enters the software standby state.

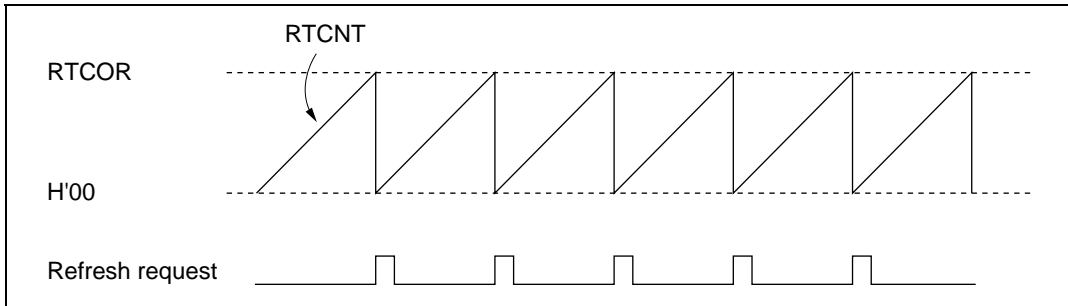
Refresh control is enabled when any area is designated as DRAM space in accordance with the setting of bits RMTS2 to RMTS0 in DRAMCR.

**CAS-before-RAS (CBR) Refreshing:** To select CBR refreshing, set the RFSHE bit to 1 in REFCR.

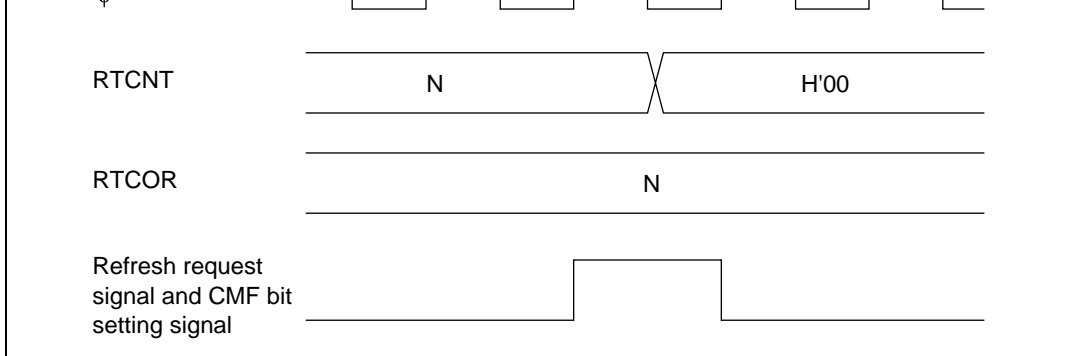
With CBR refreshing, RTCNT counts up using the input clock selected by bits RTCK2 to RTCK0 in REFCR, and when the count matches the value set in RTCOR (compare match), refresh control is performed. At the same time, RTCNT is reset and starts counting up again from H'00. Refreshing is thus repeated at fixed intervals determined by RTCOR and bits RTCK2 to RTCK0. Set a value in RTCOR and bits RTCK2 to RTCK0 that will meet the refreshing interval specification for the DRAM used.

When bits RTCK2 to RTCK0 in REFCR are set, RTCNT starts counting up. RTCNT and RTCOR settings should therefore be completed before setting bits RTCK2 to RTCK0. RTCNT operation is shown in figure 6.33, compare match timing in figure 6.34, and CBR refresh timing in figure 6.35.

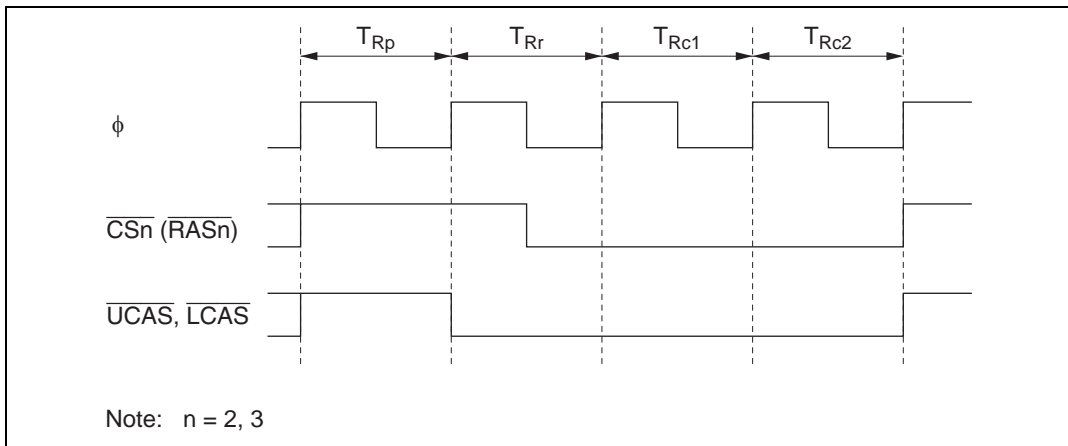
When the CBRM bit in REFCR is cleared to 0, access to external space other than DRAM space is performed in parallel during the CBR refresh period.



**Figure 6.33 RTCNT Operation**



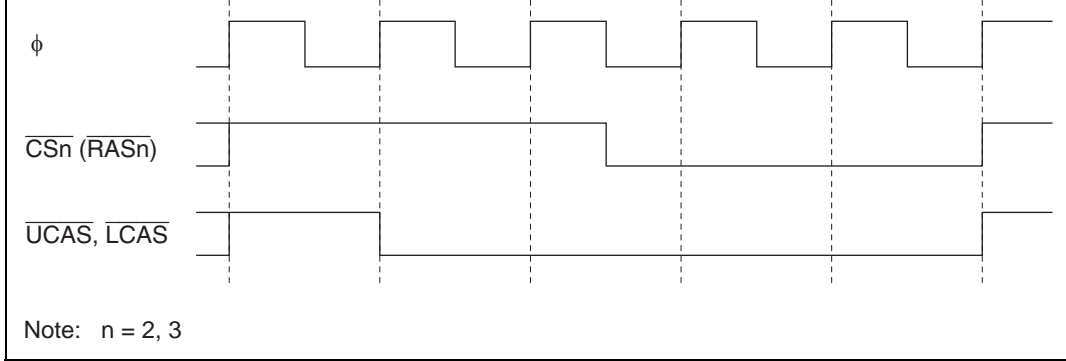
**Figure 6.34 Compare Match Timing**



**Figure 6.35 CBR Refresh Timing**

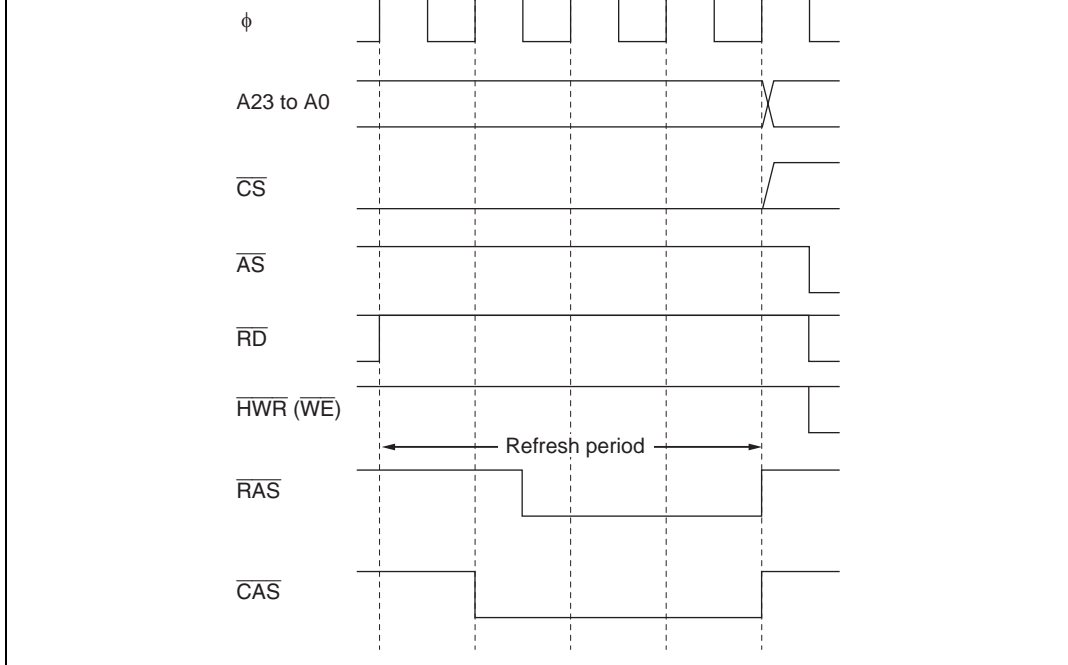
A setting can be made in bits RCW1 and RCW0 in REFCR to delay  $\overline{RAS}$  signal output by one to three cycles. Use bits RLW1 and RLW0 in REFCR to adjust the width of the  $\overline{RAS}$  signal. The settings of bits RCW1, RCW0, RLW1, and RLW0 are valid only in refresh operations.

Figure 6.36 shows the timing when bits RCW1 and RCW0 are set.



**Figure 6.36 CBR Refresh Timing**  
**(RCW1 = 0, RCW0 = 1, RLW1 = 0, RLW0 = 0)**

Depending on the DRAM used, modification of the  $\overline{WE}$  signal may not be permitted during the refresh period. In this case, the CBRM bit in REFCR should be set to 1. The bus controller will then insert refresh cycles in appropriate breaks between bus cycles. Figure 6.37 shows an example of the timing when the CBRM bit is set to 1. In this case the  $\overline{CS}$  signal is not controlled, and retains its value prior to the start of the refresh period.



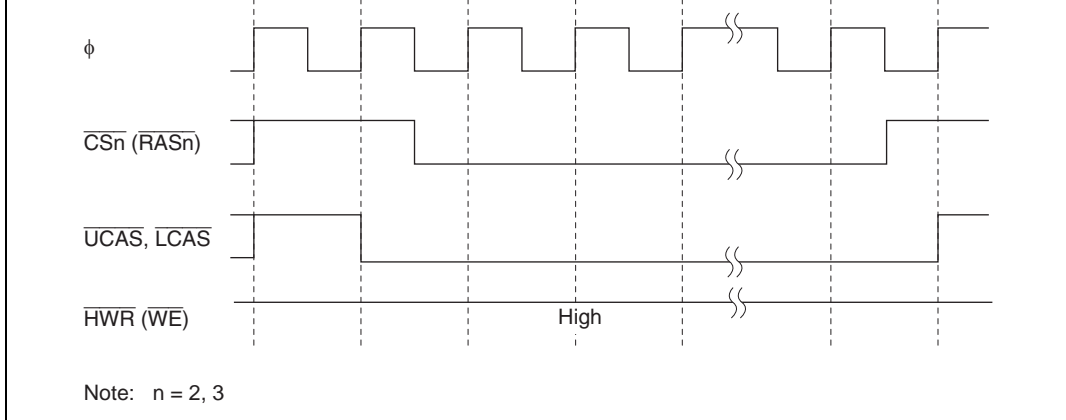
**Figure 6.37 Example of CBR Refresh Timing (CBRM = 1)**

**Self-Refreshing:** A self-refresh mode (battery backup mode) is provided for DRAM as a kind of standby mode. In this mode, refresh timing and refresh addresses are generated within the DRAM.

To select self-refreshing, set the RFSHE bit and SLFRF bit to 1 in REFCR. When a SLEEP instruction is executed to enter software standby mode, the  $\overline{CAS}$  and  $\overline{RAS}$  signals are output and DRAM enters self-refresh mode, as shown in figure 6.38.

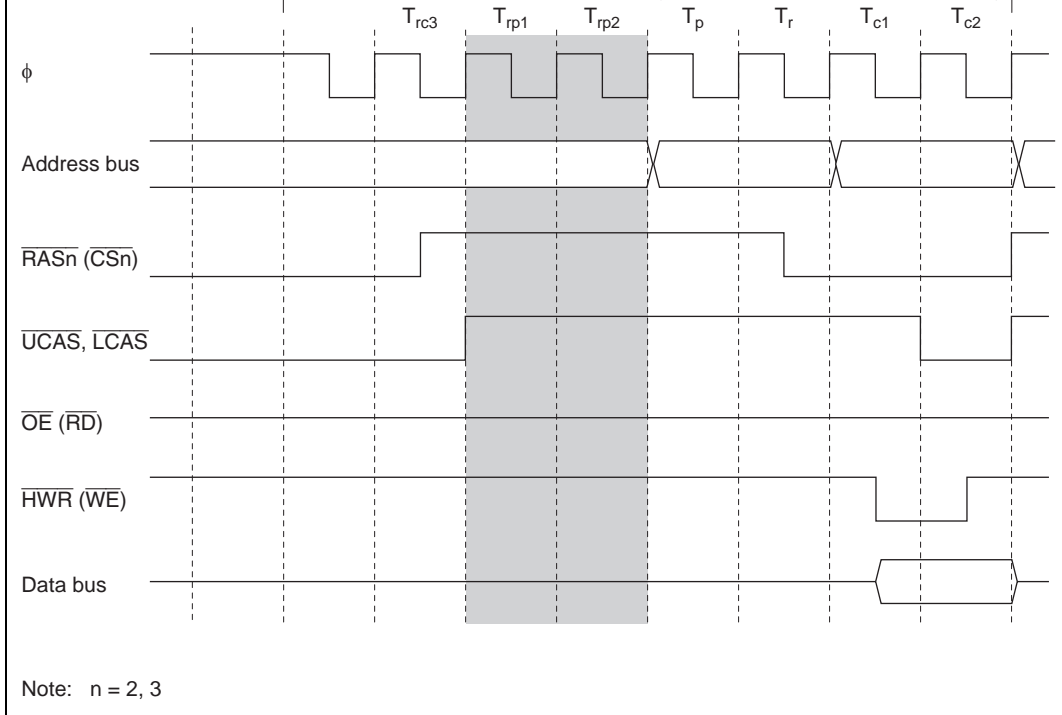
When software standby mode is exited, the SLFRF bit is cleared to 0 and self-refresh mode is exited automatically. If a CBR refresh request occurs when making a transition to software standby mode, CBR refreshing is executed, and then self-refresh mode is entered.

When using self-refresh mode, the OPE bit must not be cleared to 0 in the SBYCR register.



**Figure 6.38 Self-Refresh Timing**

In some DRAMs provided with a self-refresh mode, the  $\overline{RAS}$  signal precharge time immediately after self-refreshing is longer than the normal precharge time. A setting can be made in bits TPCS2 to TPCS0 in REFCR to make the precharge time immediately after self-refreshing from 1 to 7 states longer than the normal precharge time. In this case, too, normal precharging is performed according to the setting of bits TPC1 and TPC0 in DRACCR, and therefore a setting should be made to give the optimum post-self-refresh precharge time, including this time. Figure 6.39 shows an example of the timing when the precharge time immediately after self-refreshing is extended by 2 states.



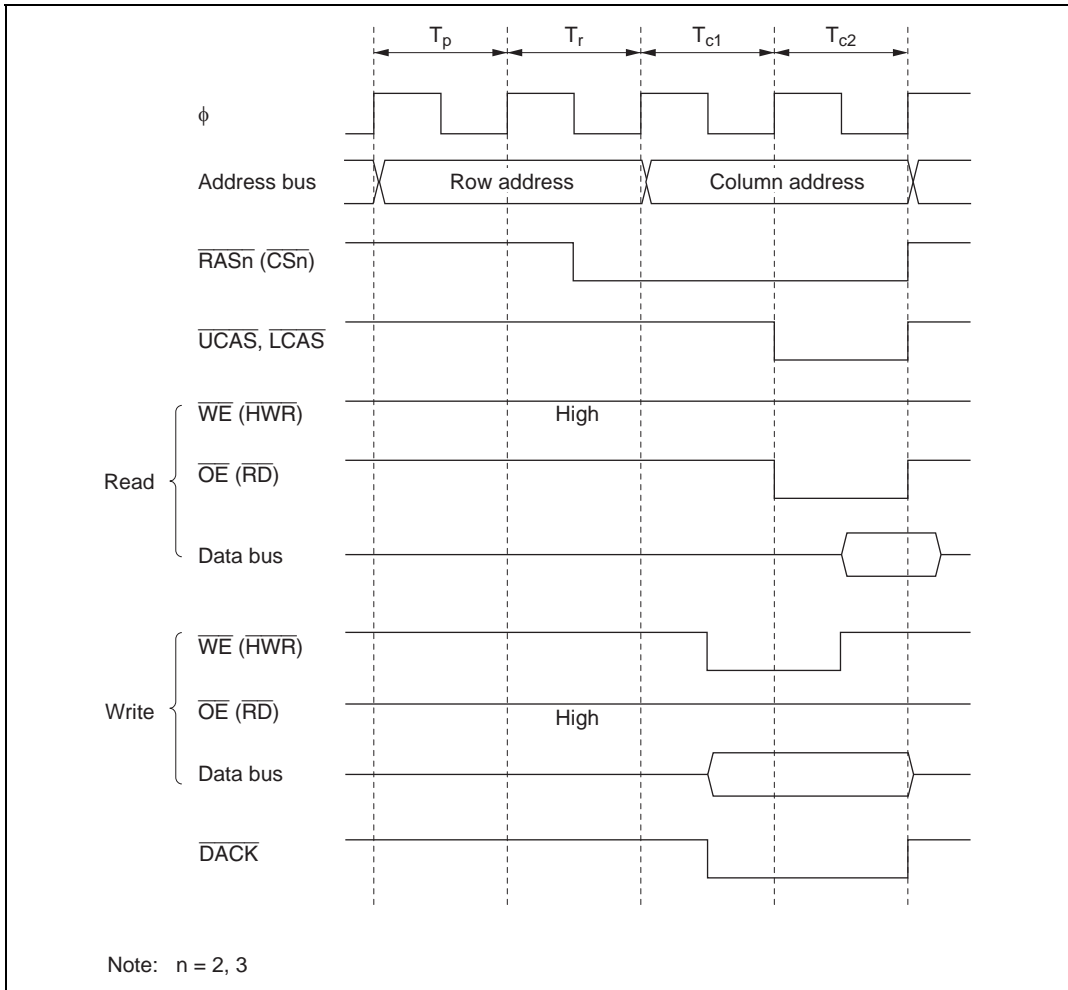
**Figure 6.39 Example of Timing when Precharge Time after Self-Refreshing Is Extended by 2 States**

**Refreshing and All-Module-Clocks-Stopped Mode:** In this LSI, if the ACSE bit is set to 1 in MSTPCRH, and then a SLEEP instruction is executed with the setting for all peripheral module clocks to be stopped (MSTPCR = H'FFFF, EXMSTPCR = H'FFFF) or for operation of the 8-bit timer module alone (MSTPCR = H'FFFE, EXMSTPCR = H'FFFF), and a transition is made to the sleep state, the all-module-clocks-stopped mode is entered, in which the bus controller and I/O port clocks are also stopped. As the bus controller clock is also stopped in this mode, CBR refreshing is not executed. If DRAM is connected externally and DRAM data is to be retained in sleep mode, the ACSE bit must be cleared to 0 in MSTPCRH.

### 6.6.13 DMAC Single Address Transfer Mode and DRAM Interface

When burst mode is selected on the DRAM interface, the  $\overline{\text{DACK}}$  output timing can be selected with the DDS bit in DRAMCR. When DRAM space is accessed in DMAC single address mode at the same time, these bits select whether or not burst access is to be performed.

Figure 6.40 shows the  $\overline{\text{DACK}}$  output timing for the DRAM interface when  $\text{DDS} = 1$ .



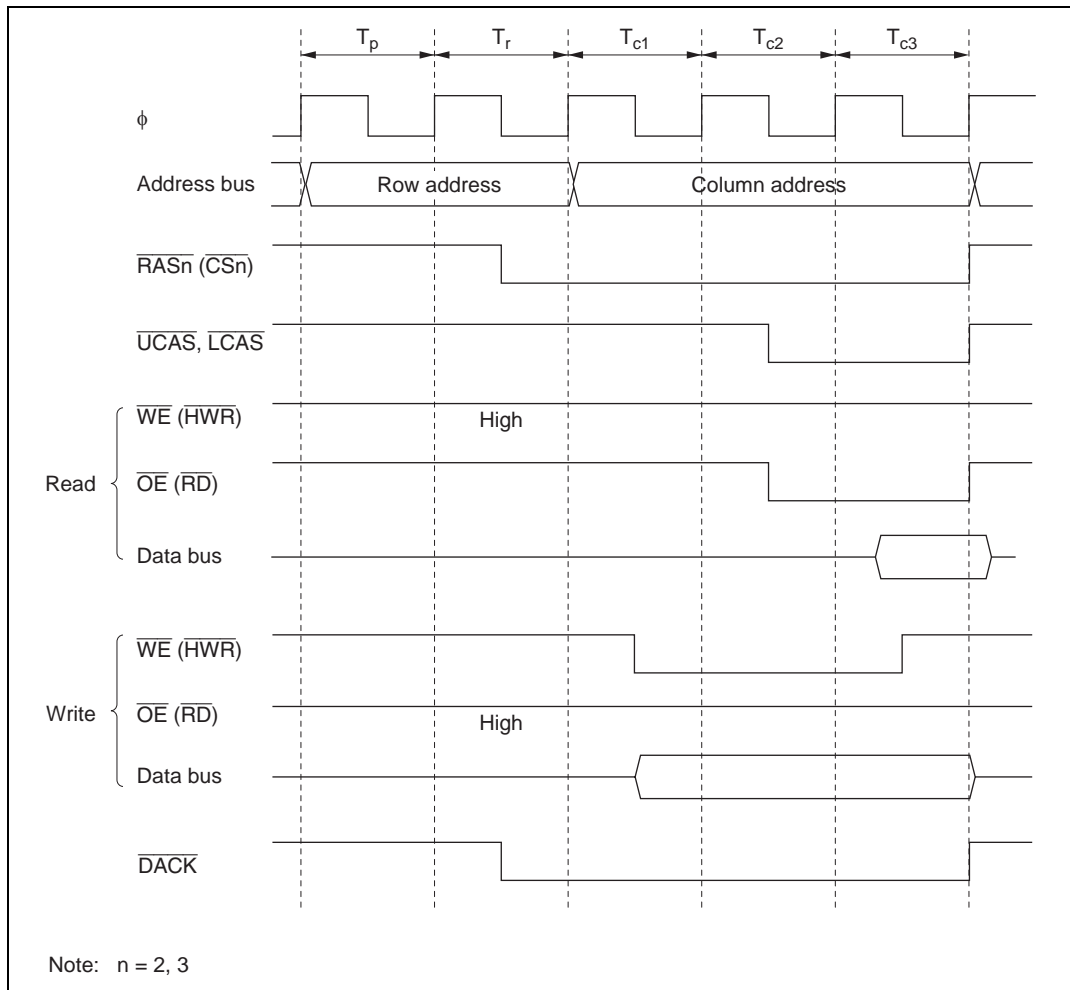
**Figure 6.40 Example of  $\overline{\text{DACK}}$  Output Timing when  $\text{DDS} = 1$  ( $\text{RAST} = 0$ ,  $\text{CAST} = 0$ )**



low from the  $T_r$  state.

In modes other than DMAC single address transfer mode, burst access can be used when accessing DRAM space.

Figure 6.41 shows the  $\overline{\text{DACK}}$  output timing for the DRAM interface when  $\text{DDS} = 0$ .



**Figure 6.41 Example of  $\overline{\text{DACK}}$  Output Timing when  $\text{DDS} = 0$  ( $\text{RAST} = 0$ ,  $\text{CAST} = 1$ )**

In this LSI, external address space areas 0 and 1 can be designated as burst ROM space, and burst ROM interfacing performed. The burst ROM space enables ROM with burst access capability to be accessed at high speed.

Areas 1 and 0 can be designated as burst ROM space by means of bits BSRM1 and BSRM0 in BROMCR. Continuous burst accesses of 4, 8, 16, or 32 words can be performed, according to the setting of the BSWD11 and BSWD10 bits in BROMCR. From 1 to 8 states can be selected for burst access.

Settings can be made independently for area 0 and area 1.

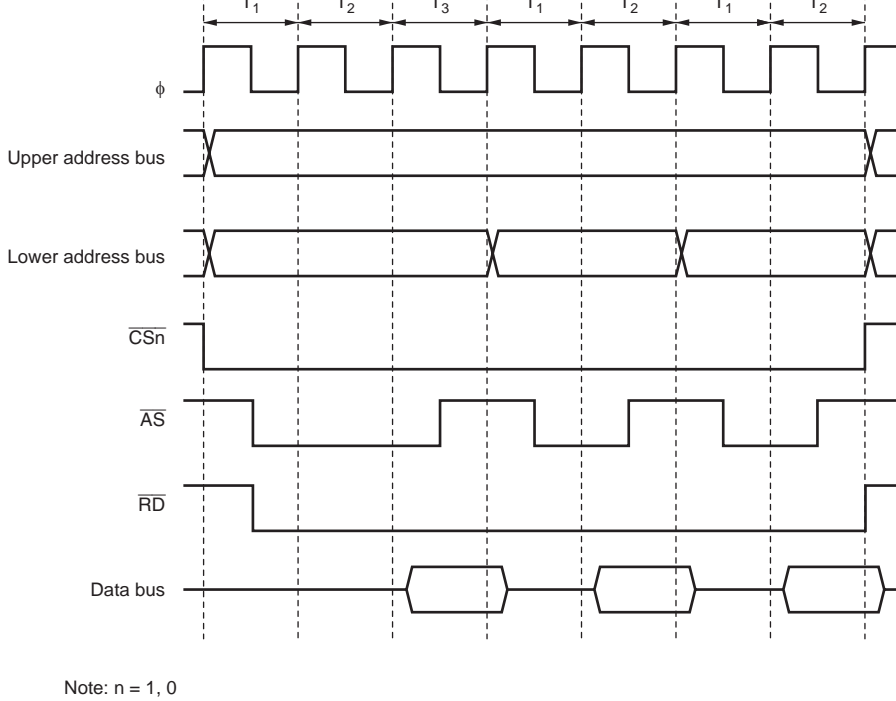
In burst ROM space, burst access covers only CPU read accesses.

### **6.7.1 Basic Timing**

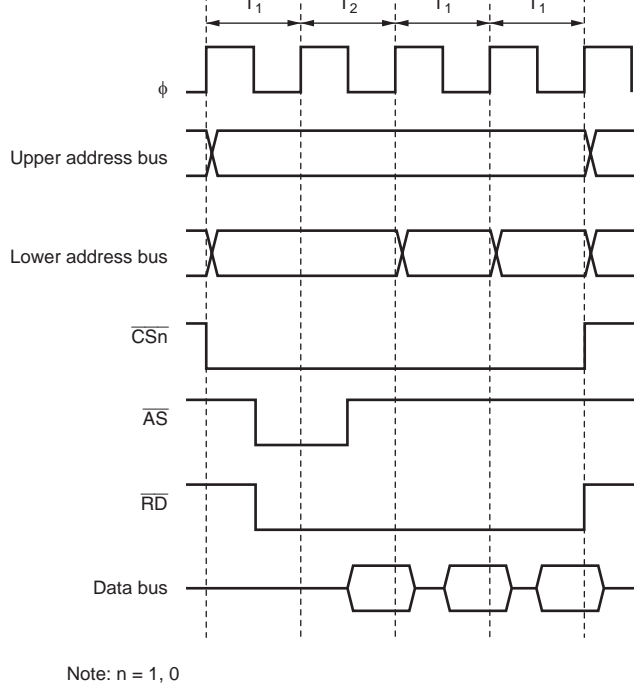
The number of access states in the initial cycle (full access) on the burst ROM interface is determined by the basic bus interface settings in ASTCR, ABWCR, WTCRA, WTCRB, and CSACRH. When area 0 or area 1 is designated as burst ROM space, the settings in RDNCR and CSACRL are ignored.

From 1 to 8 states can be selected for the burst cycle, according to the settings of bits BSTS02 to BSTS00 and BSTS12 to BSTS10 in BROMCR. Wait states cannot be inserted. Burst access of up to 32 words is performed, according to the settings of bits BSTS01, BSTS00, BSTS11, and BSTS10 in BROMCR.

The basic access timing for burst ROM space is shown in figures 6.42 and 6.43.



**Figure 6.42 Example of Burst ROM Access Timing**  
**(AST<sub>n</sub> = 1, 2-State Burst Cycle)**



**Figure 6.43 Example of Burst ROM Access Timing  
(ASTn = 0, 1-State Burst Cycle)**

## 6.7.2 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion using the  $\overline{WAIT}$  pin can be used in the initial cycle (full access) on the burst ROM interface. See section 6.5.4, Wait Control. Wait states cannot be inserted in a burst cycle.

## 6.7.3 Write Access

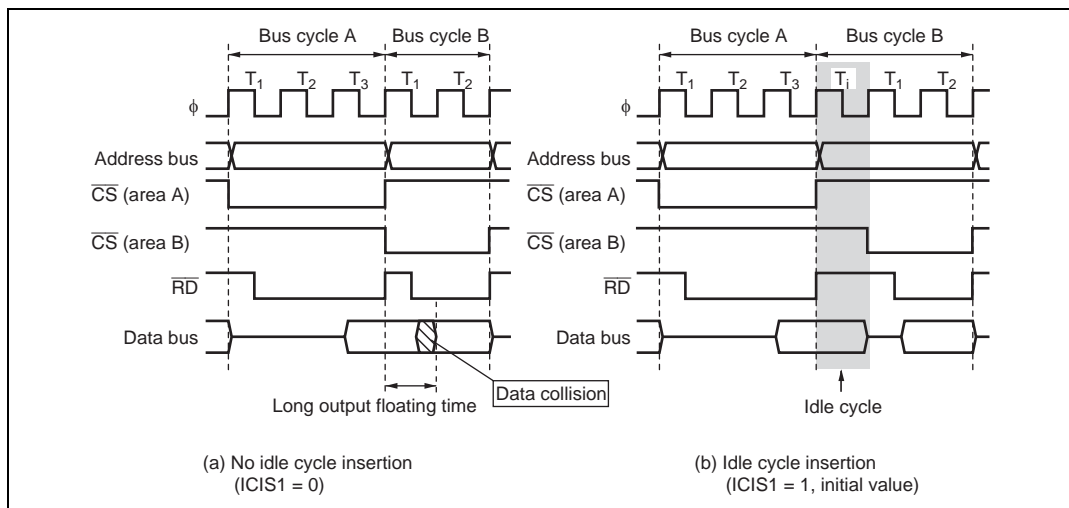
When a write access to burst ROM space is executed, burst access is interrupted at that point and the write access is executed in line with the basic bus interface settings. Write accesses are not performed in burst mode even though burst ROM space is designated.

## 6.8.1 Operation

When this LSI accesses external address space, it can insert an idle cycle ( $T_i$ ) between bus cycles in the following three cases: (1) when read accesses in different areas occur consecutively, (2) when a write cycle occurs immediately after a read cycle, and (3) when a read cycle occurs immediately after a write cycle. Insertion of a 1-state or 2-state idle cycle can be selected with the IDLC bit in BCR. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, etc., with a long output floating time, and high-speed memory, I/O interfaces, and so on.

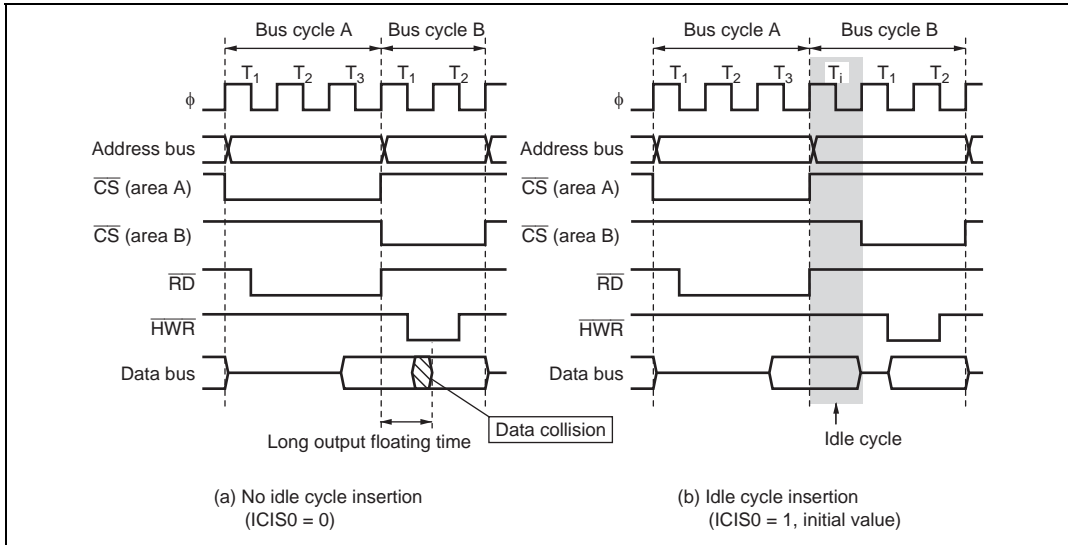
**Consecutive Reads in Different Areas:** If consecutive reads in different areas occur while the ICIS1 bit is set to 1 in BCR, an idle cycle is inserted at the start of the second read cycle.

Figure 6.44 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a read cycle for SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data collision is prevented.



**Figure 6.44** Example of Idle Cycle Operation  
(Consecutive Reads in Different Areas)

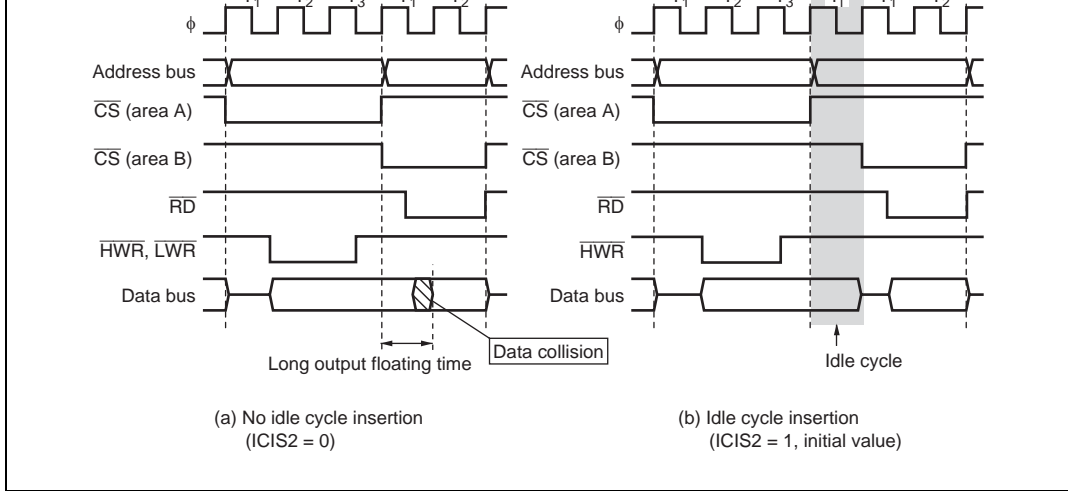
Figure 6.45 shows an example of the operation in this case. In this example, bus cycle A is a read cycle for ROM with a long output floating time, and bus cycle B is a CPU write cycle. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.



**Figure 6.45 Example of Idle Cycle Operation (Write after Read)**

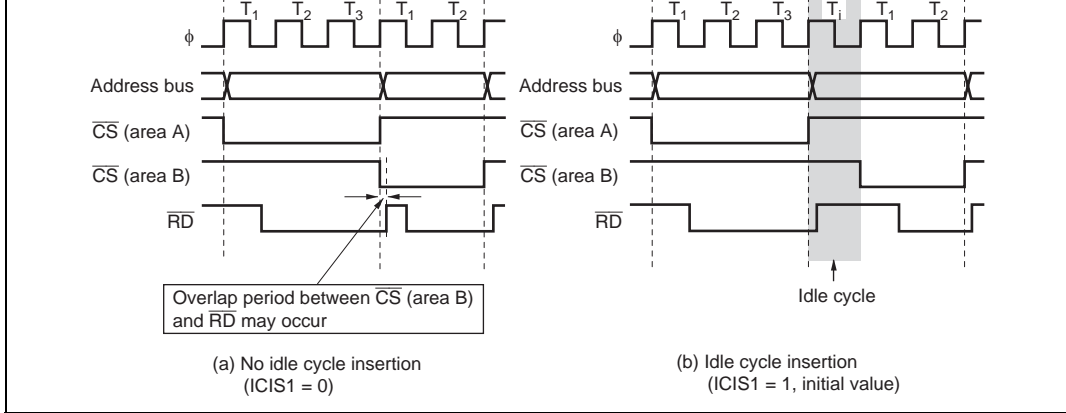
**Read after Write:** If an external read occurs after an external write while the ICIS2 bit is set to 1 in BCR, an idle cycle is inserted at the start of the read cycle.

Figure 6.46 shows an example of the operation in this case. In this example, bus cycle A is a CPU write cycle and bus cycle B is a read cycle from an external device. In (a), an idle cycle is not inserted, and a collision occurs in bus cycle B between the CPU write data and read data from an external device. In (b), an idle cycle is inserted, and a data collision is prevented.



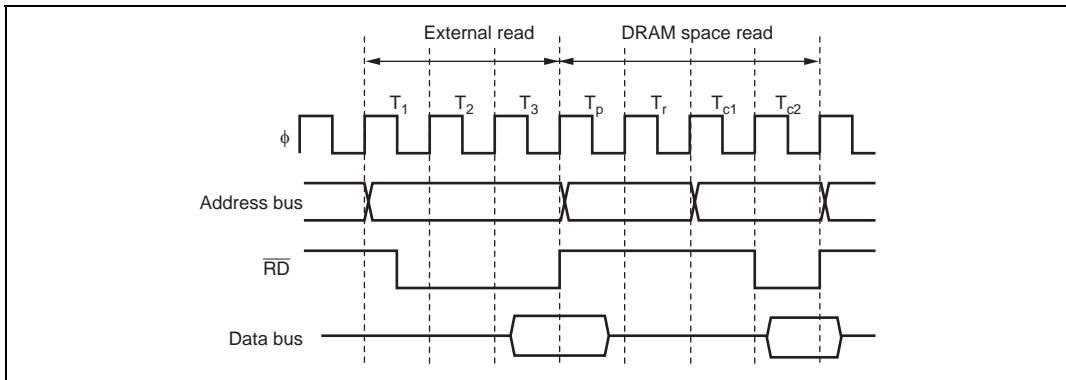
**Figure 6.46 Example of Idle Cycle Operation (Read after Write)**

**Relationship between Chip Select ( $\overline{CS}$ ) Signal and Read ( $\overline{RD}$ ) Signal:** Depending on the system's load conditions, the  $\overline{RD}$  signal may lag behind the  $\overline{CS}$  signal. An example is shown in figure 6.47. In this case, with the setting for no idle cycle insertion (a), there may be a period of overlap between the bus cycle A  $\overline{RD}$  signal and the bus cycle B  $\overline{CS}$  signal. Setting idle cycle insertion, as in (b), however, will prevent any overlap between the  $\overline{RD}$  and  $\overline{CS}$  signals. In the initial state after reset release, idle cycle insertion (b) is set.



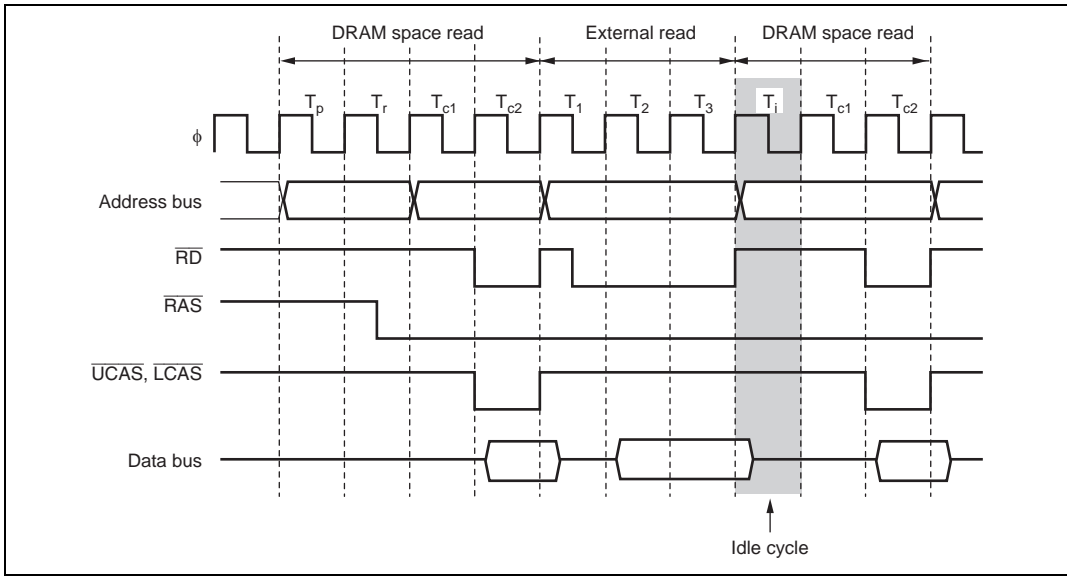
**Figure 6.47 Relationship between Chip Select ( $\overline{CS}$ ) and Read ( $\overline{RD}$ )**

**Idle Cycle in Case of DRAM Space Access after Normal Space Access:** In a DRAM space access following a normal space access, the settings of bits ICIS2, ICIS1, ICIS0, and IDLC in BCR are valid. However, in the case of consecutive reads in different areas, for example, if the second read is a full access to DRAM space, only a  $T_p$  cycle is inserted, and a  $T_i$  cycle is not. The timing in this case is shown in figure 6.48.

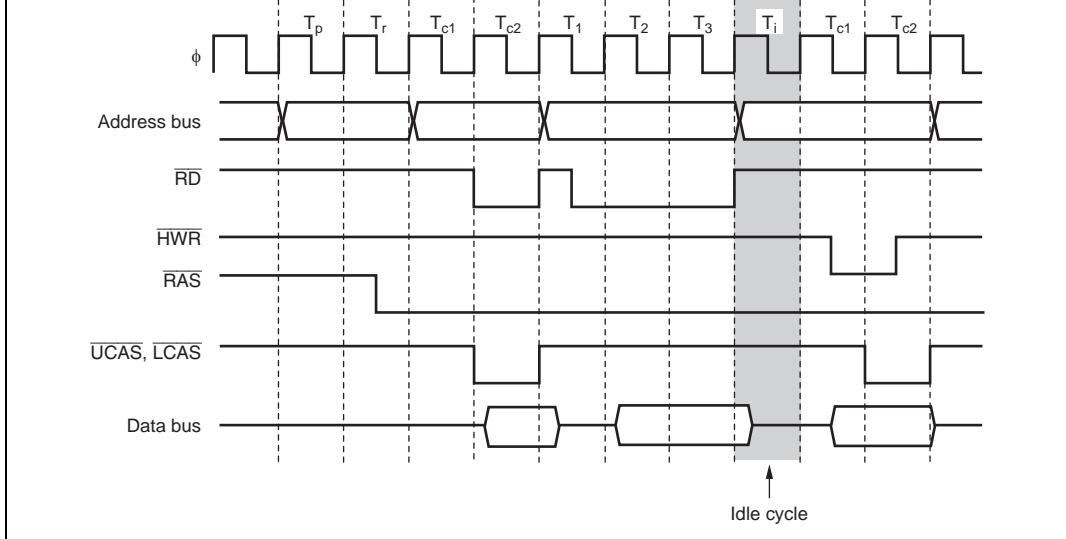


**Figure 6.48 Example of DRAM Full Access after External Read (CAST = 0)**





**Figure 6.49 Example of Idle Cycle Operation in RAS Down Mode (Consecutive Reads in Different Areas) (IDLIC = 0, RAST = 0, CAST = 0)**



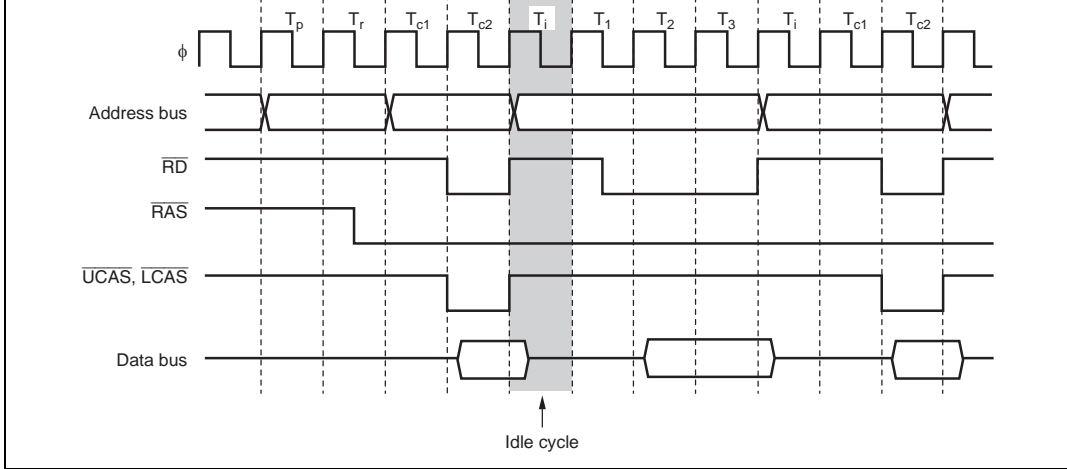
**Figure 6.50 Example of Idle Cycle Operation in RAS Down Mode (Write after Read) (IDLC = 0, RAST = 0, CAST = 0)**

**Idle Cycle in Case of Normal Space Access after DRAM Space Access:**

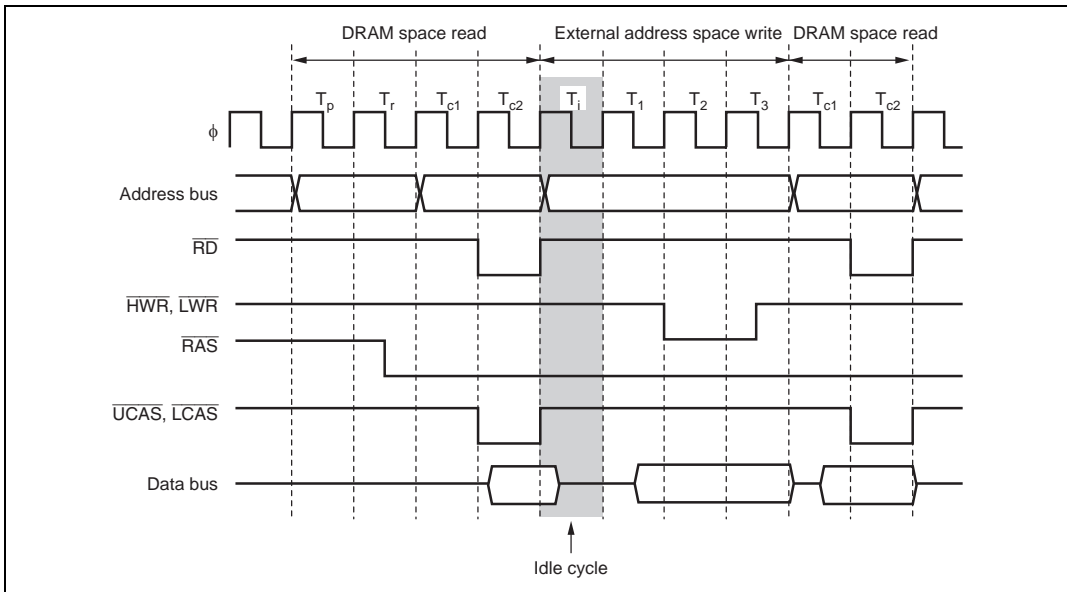
- Normal space access after DRAM space read access
 

While the DRMI bit is cleared to 0 in DRACCR, idle cycle insertion after DRAM space access is disabled. Idle cycle insertion after DRAM space access can be enabled by setting the DRMI bit to 1. The conditions and number of states of the idle cycle to be inserted are in accordance with the settings of bits ICIS1, ICIS0, and IDLC in BCR are valid. Figures 6.51 and 6.52 show examples of idle cycle operation when the DRMI bit is set to 1.

When the DRMI bit is cleared to 0, an idle cycle is not inserted after DRAM space access even if bits ICIS1 and ICIS0 are set to 1.

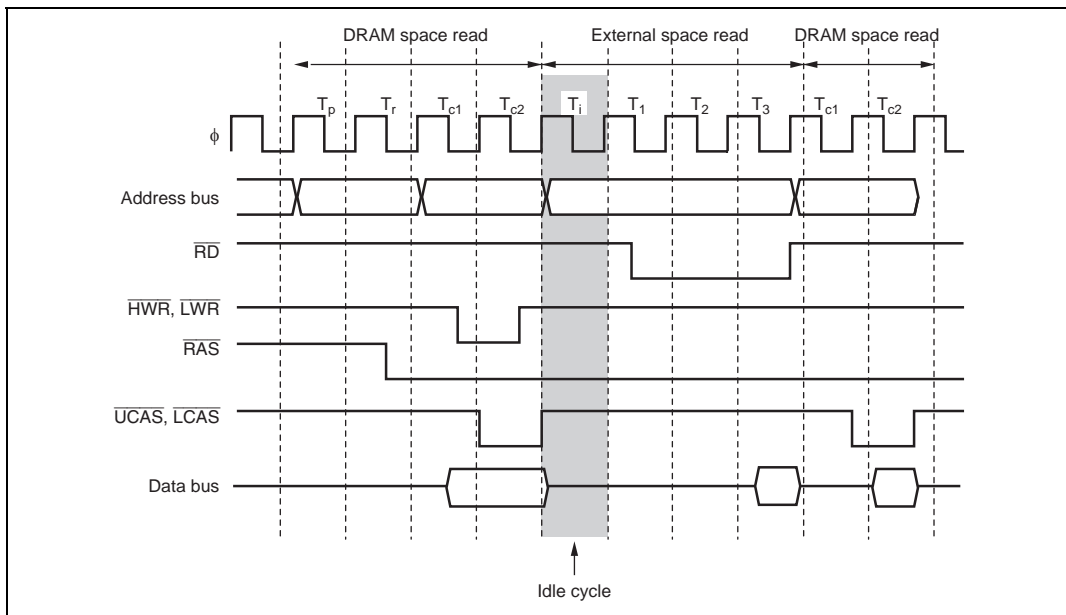


**Figure 6.51 Example of Idle Cycle Operation after DRAM Access (Consecutive Reads in Different Areas) (IDLC = 0, RAST = 0, CAST = 0)**



**Figure 6.52 Example of Idle Cycle Operation after DRAM Access (Write after Read) (IDLC = 0, RAST = 0, CAST = 0)**

space write access, idle cycle is inserted in the first read cycle. The number of states of the idle cycle to be inserted is in accordance with the setting of the IDLC bit. It does not depend on the DRMI bit in DRACCR. Figure 6.53 shows an example of idle cycle operation when the ICIS2 bit is set to 1.



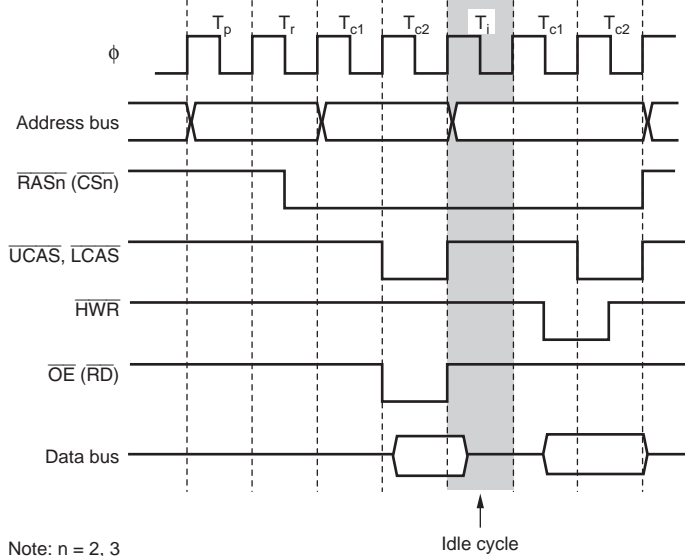
**Figure 6.53 Example of Idle Cycle Operation after DRAM Write Access (IDLC = 0, ICIS1 = 0, RAST = 0, CAST = 0)**

**Table 6.7 Idle Cycles in Mixed Accesses to Normal Space and DRAM**

Previous Access	Next Access	ICIS2	ICIS1	ICIS0	DRMI	IDLC	Idle cycle
Normal space read	Normal space read (different area)	—	0	—	—	—	Disabled
		—	1	—	—	0	1 state inserted
						1	2 states inserted
	DRAM/ space read	—	0	—	—	—	Disabled
		—	1	—	—	0	1 state inserted
						1	2 states inserted
	Normal space write	—	—	0	—	—	Disabled
		—	—	1	—	0	1 state inserted
						1	2 states inserted
	DRAM/ space write	—	—	0	—	—	Disabled
		—	—	1	—	0	1 state inserted
						1	2 states inserted
DRAM/ space read	Normal space read	—	0	—	—	—	Disabled
		—	1	—	0	—	Disabled
					1	0	1 state inserted
	DRAM/ space read	—	0	—	—	—	Disabled
		—	1	—	0	—	Disabled
					1	0	1 state inserted
	Normal space write	—	—	0	—	—	Disabled
		—	—	1	0	—	Disabled
					1	0	1 state inserted
	DRAM/ space write	—	—	0	—	—	Disabled
		—	—	1	0	—	Disabled
					1	0	1 state inserted
					1	2 states inserted	

		1	—	—	—	0	1 state inserted
						1	2 states inserted
	DRAM/ space read	0	—	—	—	—	Disabled
		1	—	—	—	0	1 state inserted
						1	2 states inserted
DRAM/ space write	Normal space read	0	—	—	—	—	Disabled
		1	—	—	—	0	1 state inserted
						1	2 states inserted
	DRAM/ space read	0	—	—	—	—	Disabled
		1	—	—	—	0	1 state inserted
						1	2 states inserted

Setting the DRMI bit in DRACCR to 1 enables an idle cycle to be inserted in the case of consecutive read and write operations in DRAM/ space burst access. Figures 6.54 shows an example of the timing for idle cycle insertion in the case of consecutive read and write accesses to DRAM/continuous synchronous DRAM space.



**Figure 6.54 Example of Timing for Idle Cycle Insertion in Case of Consecutive Read and Write Accesses to DRAM Space in RAS Down Mode**

**Table 6.8 Pin States in Idle Cycle**

<b>Pins</b>	<b>Pin State</b>
A23 to A0	Contents of following bus cycle
D15 to D0	High impedance
$\overline{\text{CSn}}$ (n = 7 to 0)	High <sup>*1*2</sup>
$\overline{\text{UCAS}}, \overline{\text{LCAS}}$	High <sup>*2</sup>
$\overline{\text{AS}}$	High
$\overline{\text{RD}}$	High
$\overline{\text{OE}}$	High
$\overline{\text{HWR}}, \overline{\text{LWR}}$	High
$\overline{\text{DAcKn}}$ (n = 1, 0)	High

Notes: 1. Remains low in DRAM space RAS down mode.

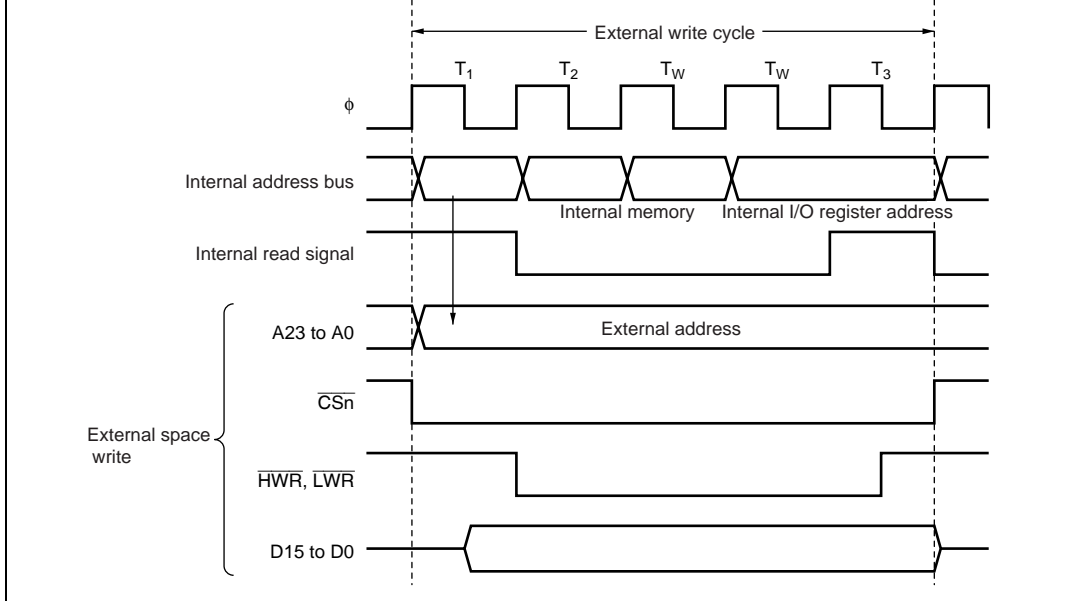
2. Remains low in a DRAM space refresh cycle.

## 6.9 Write Data Buffer Function

This LSI has a write data buffer function for the external data bus. Using the write data buffer function enables external writes and DMA single address mode transfers to be executed in parallel with internal accesses. The write data buffer function is made available by setting the WDBE bit to 1 in BCR.

Figure 6.55 shows an example of the timing when the write data buffer function is used. When this function is used, if an external address space write or DMA single address mode transfer continues for two states or longer, and there is an internal access next, an external write only is executed in the first state, but from the next state onward an internal access (on-chip memory or internal I/O register read/write) is executed in parallel with the external address space write rather than waiting until it ends.





**Figure 6.55 Example of Timing when Write Data Buffer Function is Used**

This LSI can release the external bus in response to a bus request from an external device. In the external bus released state, internal bus masters continue to operate as long as there is no external access. If any of the following requests are issued in the external bus released state, the  $\overline{\text{BREQ}}$  signal can be driven low to output a bus request externally.

- When an internal bus master wants to perform an external access
- When a refresh request is generated
- When a SLEEP instruction is executed to place the chip in software standby mode or all-module-clocks-stopped mode

### 6.10.1 Operation

In externally expanded mode, the bus can be released to an external device by setting the BRLE bit to 1 in BCR. Driving the  $\overline{\text{BREQ}}$  pin low issues an external bus request to this LSI. When the  $\overline{\text{BREQ}}$  pin is sampled, at the prescribed timing the  $\overline{\text{BACK}}$  pin is driven low, and the address bus, data bus, and bus control signals are placed in the high-impedance state, establishing the external bus released state.

In the external bus released state, internal bus masters can perform accesses using the internal bus. When an internal bus master wants to make an external access, it temporarily defers initiation of the bus cycle, and waits for the bus request from the external bus master to be canceled. If a refresh request is generated in the external bus released state, or if a SLEEP instruction is executed to place the chip in software standby mode or all-module-clocks-stopped mode, refresh control and software standby or all-module-clocks-stopped control is deferred until the bus request from the external bus master is canceled.

If the BREQOE bit is set to 1 in BCR, the  $\overline{\text{BREQO}}$  pin can be driven low when any of the following requests are issued, to request cancellation of the bus request externally.

- When an internal bus master wants to perform an external access
- When a refresh request is generated
- When a SLEEP instruction is executed to place the chip in software standby mode or all-module-clocks-stopped mode

When the  $\overline{\text{BREQ}}$  pin is driven high, the  $\overline{\text{BACK}}$  pin is driven high at the prescribed timing and the external bus released state is terminated.

If an external bus release request and external access occur simultaneously, the order of priority is as follows:

If a refresh request and external bus release request occur simultaneously, the order of priority is as follows:

(High) Refresh > External bus release (Low)

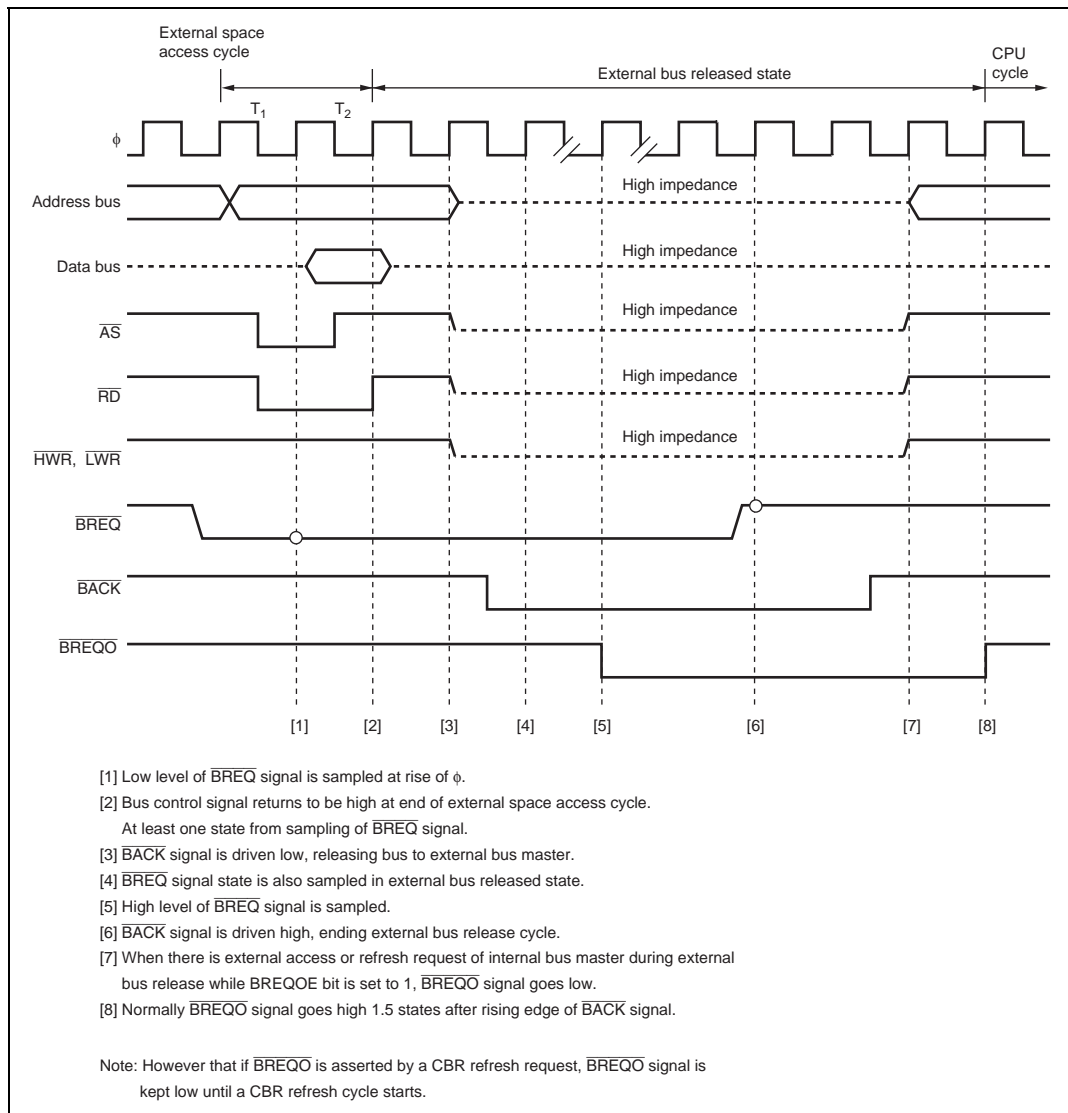
### 6.10.2 Pin States in External Bus Released State

Table 6.9 shows pin states in the external bus released state.

**Table 6.9 Pin States in Bus Released State**

<b>Pins</b>	<b>Pin State</b>
A23 to A0	High impedance
D15 to D0	High impedance
$\overline{CSn}$ (n = 7 to 0)	High impedance
$\overline{UCAS}$ , $\overline{LCAS}$	High impedance
$\overline{AS}$	High impedance
$\overline{RD}$	High impedance
$\overline{OE}$	High impedance
$\overline{HWR}$ , $\overline{LWR}$	High impedance
$\overline{DACKn}$ (n = 1, 0)	High

Figure 6.56 shows the timing for transition to the bus released state.



**Figure 6.56 Bus Released State Transition Timing**

This LSI has a bus arbiter that arbitrates bus master operations (bus arbitration).

There are three bus masters—the CPU, DTC, and DMAC—which perform read/write operations when they have possession of the bus. Each bus master requests the bus by means of a bus request signal. The bus arbiter determines priorities at the prescribed timing, and permits use of the bus by means of a bus request acknowledge signal. The selected bus master then takes possession of the bus and begins its operation.

### 6.11.1 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested, sends a bus request acknowledge signal to the bus master. If there are bus requests from more than one bus master, the bus request acknowledge signal is sent to the one with the highest priority. When a bus master receives the bus request acknowledge signal, it takes possession of the bus until that signal is canceled.

The order of priority of the bus masters is as follows:

(High) DMAC > DTC > CPU (Low)

An internal bus access by internal bus masters and external bus release, a refresh when the CBRM bit is 0 can be executed in parallel.

If an external bus release request, a refresh request, and an external access by an internal bus master occur simultaneously, the order of priority is as follows:

(High) Refresh > External bus release (Low)

(High) External bus release > External access by internal bus master (Low)

As a refresh when the CBRM bit in REFPCR is cleared to 0 and an external access other than to DRAM space by an internal bus master can be executed simultaneously, there is no relative order of priority for these two operations.

Even if a bus request is received from a bus master with a higher priority than that of the bus master that has acquired the bus and is currently operating, the bus is not necessarily transferred immediately. There are specific timings at which each bus master can relinquish the bus.

**CPU:** The CPU is the lowest-priority bus master, and if a bus request is received from the DTC, or DMAC, the bus arbiter transfers the bus to the bus master that issued the request. The timing for transfer of the bus is as follows:

- The bus is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the component operations.
- With bit manipulation instructions such as BSET and BCLR, the sequence of operations is: data read (read), relevant bit manipulation operation (modify), write-back (write). The bus is not transferred during this read-modify-write cycle, which is executed as a series of bus cycles.
- If the CPU is in sleep mode, the bus is transferred immediately.

**DTC:** The DTC sends the bus arbiter a request for the bus when an activation request is generated.

The DTC can release the bus after a vector read, a register information read (3 states), a single data transfer, or a register information write (3 states). It does not release the bus during a register information read (3 states), a single data transfer, or a register information write (3 states).

**DMAC:** The DMAC sends the bus arbiter a request for the bus when an activation request is generated.

In the case of an external request in short address mode or normal mode, and in cycle steal mode, the DMAC releases the bus after a single transfer.

In block transfer mode, it releases the bus after transfer of one block, and in burst mode, after completion of the transfer. However, in the event of external bus release request, which have a higher priority than the DMAC, the bus may be transferred to the bus master even if block or burst transfer is in progress.

**External Bus Release:** When the  $\overline{\text{BREQ}}$  pin goes low and an external bus release request is issued while the BRLE bit is set to 1 in BCR, a bus request is sent to the bus arbiter.

External bus release can be performed on completion of an external bus cycle.

In a reset, this LSI, including the bus controller, enters the reset state immediately, and any executing bus cycle is aborted.

## 6.13 Usage Notes

### 6.13.1 External Bus Release Function and All-Module-Clocks-Stopped Mode

In this LSI, if the ACSE bit is set to 1 in MSTPCR, and then a SLEEP instruction is executed with the setting for all peripheral module clocks to be stopped (MSTPCR = H'FFFF, EXMSTPCR = H'FFFF) or for operation of the 8-bit timer module alone (MSTPCR = H'FFFE, EXMSTPCR = H'FFFF), and a transition is made to the sleep state, the all-module-clocks-stopped mode is entered in which the clock is also stopped for the bus controller and I/O ports. In this state, the external bus release function is halted. To use the external bus release function in sleep mode, the ACSE bit in MSTPCR must be cleared to 0. Conversely, if a SLEEP instruction to place the chip in all-module-clocks-stopped mode is executed in the external bus released state, the transition to all-module-clocks-stopped mode is deferred and performed until after the bus is recovered.

### 6.13.2 External Bus Release Function and Software Standby

In this LSI, internal bus master operation does not stop even while the bus is released, as long as the program is running in on-chip ROM, etc., and no external access occurs. If a SLEEP instruction to place the chip in software standby mode is executed while the external bus is released, the transition to software standby mode is deferred and performed after the bus is recovered.

Also, since clock oscillation halts in software standby mode, if  $\overline{\text{BREQ}}$  goes low in this mode, indicating an external bus release request, the request cannot be answered until the chip has recovered from the software standby state.

### 6.13.3 External Bus Release Function and CBR Refreshing

CBR refreshing cannot be executed while the external bus is released. Setting the BREQOE bit to 1 in BCR beforehand enables the  $\overline{\text{BREQO}}$  signal to be output when a CBR refresh request is issued.

When the BREQOE bit is set to 1 and the BREQO signal is output, BREQO may go low before the  $\overline{\text{BACK}}$  signal.

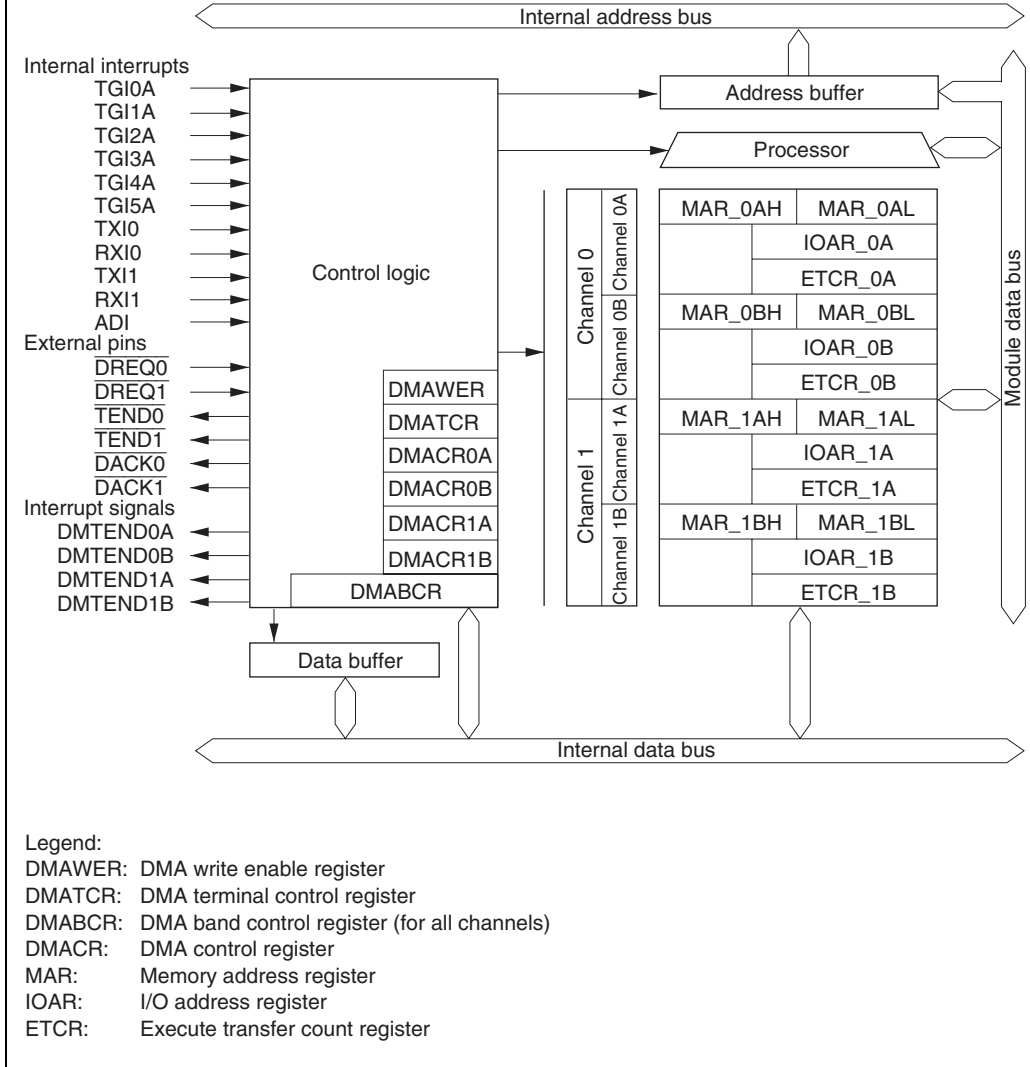
This will occur if the next external access request or CBR refresh request occurs while internal bus arbitration is in progress after the chip samples a low level of  $\overline{\text{BREQ}}$ .



This LSI has a built-in DMA controller (DMAC) which can carry out data transfer on up to 4 channels.

## 7.1 Features

- Choice of short address mode or full address mode
  - Short address mode
    - Maximum of 4 channels can be used
    - Dual address mode or single address mode can be selected
    - In dual address mode, one of the two addresses, transfer source and transfer destination, is specified as 24 bits and the other as 16 bits
    - In single address mode, transfer source or transfer destination address only is specified as 24 bits
    - In single address mode, transfer can be performed in one bus cycle
    - Choice of sequential mode, idle mode, or repeat mode for dual address mode and single address mode
  - Full address mode
    - Maximum of 2 channels can be used
    - Transfer source and transfer destination addresses as specified as 24 bits
    - Choice of normal mode or block transfer mode
- 16-Mbyte address space can be specified directly
- Byte or word can be set as the transfer unit
- Activation sources: internal interrupt, external request, auto-request (depending on transfer mode)
  - Six 16-bit timer-pulse unit (TPU) compare match/input capture interrupts
  - Serial communication interface (SCI\_0, SCI\_1) transmission complete interrupt, reception complete interrupt
  - A/D converter conversion end interrupt
  - External request
  - Auto-request
- Module stop mode can be set



**Figure 7.1 Block Diagram of DMAC**

Table 7.1 summarizes the pins of the interrupt controller.

**Table 7.1 Pin Configuration**

Channel	Pin Name	Symbol	I/O	Function
0	DMA request 0	$\overline{\text{DREQ0}}$	Input	Channel 0 external request
	DMA transfer acknowledge 0	$\overline{\text{DACK0}}$	Output	Channel 0 single address transfer acknowledge
	DMA transfer end 0	$\overline{\text{TEND0}}$	Output	Channel 0 transfer end
1	DMA request 1	$\overline{\text{DREQ1}}$	Input	Channel 1 external request
	DMA transfer acknowledge 1	$\overline{\text{DACK1}}$	Output	Channel 1 single address transfer acknowledge
	DMA transfer end 1	$\overline{\text{TEND1}}$	Output	Channel 1 transfer end

### 7.3 Register Descriptions

- Memory address register\_0AH (MAR\_0AH)
- Memory address register\_0AL (MAR\_0AL)
- I/O address register\_0A (IOAR\_0A)
- Transfer count register\_0A (ECTR\_0A)
- Memory address register\_0BH (MAR\_0BH)
- Memory address register\_0BL (MAR\_0BL)
- I/O address register\_0B (IOAR\_0B)
- Transfer count register\_0B (ECTR\_0B)
- Memory address register\_1AH (MAR\_1AH)
- Memory address register\_1AL (MAR\_1AL)
- I/O address register\_1A (IOAR\_1A)
- Transfer count register\_1A (ETCR\_1B)
- Memory address register\_1BH (MAR\_1BH)
- Memory address register\_1BL (MAR\_1BL)
- I/O address register\_1B (IOAR\_1B)
- Transfer count register\_1B (ETCR\_1B)
- DMA control register\_0A (DMACR\_0A)
- DMA control register\_0B (DMACR\_0B)
- DMA control register\_1A (DMACR\_1A)

- DMA band control register L (DMABCRL)
- DMA write enable register (DMAWER)
- DMA terminal control register (DMATCR)

The functions of MAR, IOAR, ETCR, DMACR, and DMABCR differ according to the transfer mode (short address mode or full address mode). The transfer mode can be selected by means of the FAE1 and FAE0 bits in DMABCRH. The register configurations for short address mode and full address mode of channel 0 are shown in table 7.2.

**Table 7.2 Short Address Mode and Full Address Mode (Channel 0)**

FAE0	Description																					
0	Short address mode specified (channels 0A and 0B operate independently)																					
Channel 0A	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 100px;">MAR_0AH</td> <td>MAR_0AL</td> <td>← Specifies transfer source/transfer destination address</td> </tr> <tr> <td></td> <td>IOAR_0A</td> <td>← Specifies transfer destination/transfer source address</td> </tr> <tr> <td></td> <td>ETCR_0A</td> <td>← Specifies number of transfers</td> </tr> <tr> <td></td> <td>DMACR_0A</td> <td>← Specifies transfer size, mode, activation source</td> </tr> </table>	MAR_0AH	MAR_0AL	← Specifies transfer source/transfer destination address		IOAR_0A	← Specifies transfer destination/transfer source address		ETCR_0A	← Specifies number of transfers		DMACR_0A	← Specifies transfer size, mode, activation source									
MAR_0AH	MAR_0AL	← Specifies transfer source/transfer destination address																				
	IOAR_0A	← Specifies transfer destination/transfer source address																				
	ETCR_0A	← Specifies number of transfers																				
	DMACR_0A	← Specifies transfer size, mode, activation source																				
Channel 0B	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 100px;">MAR_0BH</td> <td>MAR_0BL</td> <td>← Specifies transfer source/transfer destination address</td> </tr> <tr> <td></td> <td>IOAR_0B</td> <td>← Specifies transfer destination/transfer source address</td> </tr> <tr> <td></td> <td>ETCR_0B</td> <td>← Specifies number of transfers</td> </tr> <tr> <td></td> <td>DMACR_0B</td> <td>← Specifies transfer size, mode, activation source</td> </tr> </table>	MAR_0BH	MAR_0BL	← Specifies transfer source/transfer destination address		IOAR_0B	← Specifies transfer destination/transfer source address		ETCR_0B	← Specifies number of transfers		DMACR_0B	← Specifies transfer size, mode, activation source									
MAR_0BH	MAR_0BL	← Specifies transfer source/transfer destination address																				
	IOAR_0B	← Specifies transfer destination/transfer source address																				
	ETCR_0B	← Specifies number of transfers																				
	DMACR_0B	← Specifies transfer size, mode, activation source																				
1	Full address mode specified (channels 0A and 0B operate in combination as channel 0)																					
Channel 0	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 100px;">MAR_0AH</td> <td>MAR_0AL</td> <td>← Specifies transfer source address</td> </tr> <tr> <td>MAR_0BH</td> <td>MAR_0BL</td> <td>← Specifies transfer destination address</td> </tr> <tr> <td></td> <td>IOAR_0A</td> <td>← Not used</td> </tr> <tr> <td></td> <td>IOAR_0B</td> <td>← Not used</td> </tr> <tr> <td></td> <td>ETCR_0A</td> <td>← Specifies number of transfers</td> </tr> <tr> <td></td> <td>ETCR_0B</td> <td>← Specifies number of transfers (used in block transfer mode only)</td> </tr> <tr> <td></td> <td>DMACR_0A</td> <td>DMACR_0B ← Specifies transfer size, mode, activation source, etc.</td> </tr> </table>	MAR_0AH	MAR_0AL	← Specifies transfer source address	MAR_0BH	MAR_0BL	← Specifies transfer destination address		IOAR_0A	← Not used		IOAR_0B	← Not used		ETCR_0A	← Specifies number of transfers		ETCR_0B	← Specifies number of transfers (used in block transfer mode only)		DMACR_0A	DMACR_0B ← Specifies transfer size, mode, activation source, etc.
MAR_0AH	MAR_0AL	← Specifies transfer source address																				
MAR_0BH	MAR_0BL	← Specifies transfer destination address																				
	IOAR_0A	← Not used																				
	IOAR_0B	← Not used																				
	ETCR_0A	← Specifies number of transfers																				
	ETCR_0B	← Specifies number of transfers (used in block transfer mode only)																				
	DMACR_0A	DMACR_0B ← Specifies transfer size, mode, activation source, etc.																				

MAR is a 32-bit readable/writable register that specifies the source address (transfer source address) or destination address (transfer destination address). MAR consists of two 16-bit registers MARH and MARL. The upper 8 bits of MARH are reserved: they are always read as 0, and cannot be modified.

The DMA has four MAR registers: MAR\_0A in channel 0 (channel 0A), MAR\_0B in channel 0 (channel 0B), MAR\_1A in channel 1 (channel 1A), and MAR\_1B in channel 1 (channel 1B).

MAR is not initialized by a reset or in standby mode.

**Short Address Mode:** In short address mode, MARA and MARB operate independently. Whether MAR functions as the source address register or as the destination address register can be selected by means of the DTDIR bit in DMACR.

MAR is incremented or decremented each time a byte or word transfer is executed, so that the address specified by MAR is constantly updated.

**Full Address Mode:** In full address mode, MARA functions as the source address register, and MARB as the destination address register.

MAR is incremented or decremented each time a byte or word transfer is executed, so that the source or destination address is constantly updated.

### 7.3.2 I/O Address Registers (IOARA and IOARB)

IOAR is a 16-bit readable/writable register that specifies the lower 16 bits of the source address (transfer source address) or destination address (transfer destination address). The upper 8 bits of the transfer address are automatically set to H'FF.

The DMA has four IOAR registers: IOAR\_0A in channel 0 (channel 0A), IOAR\_0B in channel 0 (channel 0B), IOAR\_1A in channel 1 (channel 1A), and IOAR\_1B in channel 1 (channel 1B).

Whether IOAR functions as the source address register or as the destination address register can be selected by means of the DTDIR bit in DMACR.

IOAR is not incremented or decremented each time a data transfer is executed, so the address specified by IOAR is fixed.

IOAR is not initialized by a reset or in standby mode.

IOAR can be used in short address mode but not in full address mode.

ETCR is a 16-bit readable/writable register that specifies the number of transfers.

The DMA has four ETCR registers: ETCR\_0A in channel 0 (channel 0A), ETCR\_0B in channel 0 (channel 0B), ETCR\_1A in channel 1 (channel 1A), and ETCR\_1B in channel 1 (channel 1B).

ETCR is not initialized by a reset or in standby mode.

**Short Address Mode:** The function of ETCR in sequential mode and idle mode differs from that in repeat mode.

In sequential mode and idle mode, ETCR functions as a 16-bit transfer counter. ETCR is decremented by 1 each time a transfer is performed, and when the count reaches H'00, the DTE bit in DMABCRL is cleared, and transfer ends.

In repeat mode, ETCRL functions as an 8-bit transfer counter and ETCRH functions as a transfer count holding register. ETCRL is decremented by 1 each time a transfer is performed, and when the count reaches H'00, ETCRL is loaded with the value in ETCRH. At this point, MAR is automatically restored to the value it had when the count was started. The DTE bit in DMABCRL is not cleared, and so transfers can be performed repeatedly until the DTE bit is cleared by the user.

**Full Address Mode:** The function of ETCR in normal mode differs from that in block transfer mode.

In normal mode, ETCRA functions as a 16-bit transfer counter. ETCRA is decremented by 1 each time a data transfer is performed, and transfer ends when the count reaches H'0000. ETCRB is not used in normal mode.

In block transfer mode, ETCRAL functions as an 8-bit block size counter and ETCRAH functions as a block size holding register. ETCRAL is decremented by 1 each time a 1-byte or 1-word transfer is performed, and when the count reaches H'00, ETCRAL is loaded with the value in ETCRAH. So by setting the block size in ETCRAH and ETCRAL, it is possible to repeatedly transfer blocks consisting of any desired number of bytes or words.

In block transfer mode, ETCRB functions as a 16-bit block transfer counter. ETCRB is decremented by 1 each time a block is transferred, and transfer ends when the count reaches H'0000.

DMACR controls the operation of each DMAC channel.

The DMA has four DMACR registers: DMACR\_0A in channel 0 (channel 0A), DMACR\_0B in channel 0 (channel 0B), DMACR\_1A in channel 1 (channel 1A), and DMACR\_1B in channel 1 (channel 1B).

In short address mode, channels A and B operate independently, and in full address mode, channels A and B operate together. The bit functions in the DMACR registers differ according to the transfer mode.

### Short Address Mode:

- DMACR\_0A, DMACR\_0B, DMACR\_1A, and DMARC\_1B

Bit	Bit Name	Initial Value	R/W	Description
7	DTSZ	0	R/W	Data Transfer Size Selects the size of data to be transferred at one time. 0: Byte-size transfer 1: Word-size transfer
6	DTID	0	R/W	Data Transfer Increment/Decrement Selects incrementing or decrementing of MAR after every data transfer in sequential mode or repeat mode. In idle mode, MAR is neither incremented nor decremented. 0: MAR is incremented after a data transfer <ul style="list-style-type: none"> <li>• When DTSZ = 0, MAR is incremented by 1</li> <li>• When DTSZ = 1, MAR is incremented by 2</li> </ul> 1: MAR is decremented after a data transfer <ul style="list-style-type: none"> <li>• When DTSZ = 0, MAR is decremented by 1</li> <li>• When DTSZ = 1, MAR is decremented by 2</li> </ul>
5	RPE	0	R/W	Repeat Enable Used in combination with the DTIE bit in DMABCR to select the mode (sequential, idle, or repeat) in which transfer is to be performed. <ul style="list-style-type: none"> <li>• When DTIE = 0 (no transfer end interrupt)</li> </ul> 0: Transfer in sequential mode 1: Transfer in repeat mode <ul style="list-style-type: none"> <li>• When DTIE = 1 (with transfer end interrupt)</li> </ul> 0: Transfer in sequential mode 1: Transfer in idle mode

Used in combination with the SAE bit in DMABCR to specify the data transfer direction (source or destination). The function of this bit is therefore different in dual address mode and single address mode.

- When SAE = 0
    - 0: Transfer with MAR as source address and IOAR as destination address
    - 1: Transfer with IOAR as source address and MAR as destination address
  - When SAE = 1
    - 0: Transfer with MAR as source address and  $\overline{\text{DACK}}$  pin as write strobe
    - 1: Transfer with  $\overline{\text{DACK}}$  pin as read strobe and MAR as destination address
-



2	DTF2	0	R/W	These bits select the data transfer factor
1	DTF1	0	R/W	(activation source). There are some
0	DTF0	0	R/W	differences in activation sources for channel A and channel B.
				<ul style="list-style-type: none"> <li>• Channel A</li> </ul>
				0000: Setting prohibited
				0001: Activated by A/D converter conversion end interrupt
				0010: Setting prohibited
				0011: Setting prohibited
				0100: Activated by SCI channel 0 transmission complete interrupt
				0101: Activated by SCI channel 0 reception complete interrupt
				0110: Activated by SCI channel 1 transmission complete interrupt
				0111: Activated by SCI channel 1 reception complete interrupt
				1000: Activated by TPU channel 0 compare match/input capture A interrupt
				1001: Activated by TPU channel 1 compare match/input capture A interrupt
				1010: Activated by TPU channel 2 compare match/input capture A interrupt
				1011: Activated by TPU channel 3 compare match/input capture A interrupt
				1100: Activated by TPU channel 4 compare match/input capture A interrupt
				1101: Activated by TPU channel 5 compare match/input capture A interrupt
				1110: Setting prohibited
				1111: Setting prohibited

2	DTF2	0	R/W	0000: Setting prohibited
1	DTF1	0	R/W	0001: Activated by A/D converter conversion end interrupt
0	DTF0	0	R/W	0010: Activated by $\overline{\text{DREQ}}$ pin falling edge input (detected as a low level in the first transfer after transfer is enabled)
				0011: Activated by $\overline{\text{DREQ}}$ pin low-level input
				0100: Activated by SCI channel 0 transmission complete interrupt
				0101: Activated by SCI channel 0 reception complete interrupt
				0110: Activated by SCI channel 1 transmission complete interrupt
				0111: Activated by SCI channel 1 reception complete interrupt
				1000: Activated by TPU channel 0 compare match/input capture A interrupt
				1001: Activated by TPU channel 1 compare match/input capture A interrupt
				1010: Activated by TPU channel 2 compare match/input capture A interrupt
				1011: Activated by TPU channel 3 compare match/input capture A interrupt
				1100: Activated by TPU channel 4 compare match/input capture A interrupt
				1101: Activated by TPU channel 5 compare match/input capture A interrupt
				1110: Setting prohibited
				1111: Setting prohibited

The same factor can be selected for more than one channel. In this case, activation starts with the highest-priority channel according to the relative channel priorities. For relative channel priorities, see section 7.5.12, Multi-Channel Operation.

---

- DMACKR\_0A and DMACKR\_1A

Bit	Bit Name	Initial Value	R/W	Description
15	DTSZ	0	R/W	Data Transfer Size Selects the size of data to be transferred at one time. 0: Byte-size transfer 1: Word-size transfer
14	SAID	0	R/W	Source Address Increment/Decrement
13	SAIDE	0	R/W	Source Address Increment/Decrement Enable These bits specify whether source address register MARA is to be incremented, decremented, or left unchanged, when data transfer is performed. 00: MARA is fixed 01: MARA is incremented after a data transfer <ul style="list-style-type: none"> <li>• When DTSZ = 0, MARA is incremented by 1</li> <li>• When DTSZ = 1, MARA is incremented by 2</li> </ul> 10: MARA is fixed 11: MARA is decremented after a data transfer <ul style="list-style-type: none"> <li>• When DTSZ = 0, MARA is decremented by 1</li> <li>• When DTSZ = 1, MARA is decremented by 2</li> </ul>
12	BLKDIR	0	R/W	Block Direction
11	BLKE	0	R/W	Block Enable These bits specify whether normal mode or block transfer mode is to be used for data transfer. If block transfer mode is specified, the BLKDIR bit specifies whether the source side or the destination side is to be the block area. ×0: Transfer in normal mode 01: Transfer in block transfer mode (destination side is block area) 11: Transfer in block transfer mode (source side is block area)

to  
8

Though these bits can be read from or written to,  
the write value should always be 0.

Legend:

×: Don't care

- DMACR\_0B and DMACR\_1B

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved Though this bit can be read from or written to, the write value should always be 0.
6	DAID	0	R/W	Destination Address Increment/Decrement
5	DAIDE	0	R/W	Destination Address Increment/Decrement Enable These bits specify whether destination address register MARB is to be incremented, decremented, or left unchanged, when data transfer is performed. 00: MARB is fixed 01: MARB is incremented after a data transfer <ul style="list-style-type: none"><li>• When DTSZ = 0, MARB is incremented by 1</li><li>• When DTSZ = 1, MARB is incremented by 2</li></ul> 10: MARB is fixed 11: MARB is decremented after a data transfer <ul style="list-style-type: none"><li>• When DTSZ = 0, MARB is decremented by 1</li><li>• When DTSZ = 1, MARB is decremented by 2</li></ul>
4	—	0	R/W	Reserved Though this bit can be read from or written to, the write value should always be 0.
3	DTF3	0	R/W	Data Transfer Factor 3 to 0
2	DTF2	0	R/W	These bits select the data transfer factor
1	DTF1	0	R/W	(activation source). The factors that can be
0	DTF0	0	R/W	specified differ between normal mode and block transfer mode.

2	DTF2	0	R/W	0000: Setting prohibited
1	DTF1	0	R/W	0001: Setting prohibited
0	DTF0	0	R/W	0010: Activated by $\overline{\text{DREQ}}$ pin falling edge input (for the first transfer after data transfer is enabled, activated by $\overline{\text{DREQ}}$ pin low-level input)
				0011: Activated by $\overline{\text{DREQ}}$ pin low-level input
				010x: Setting prohibited
				0110: Auto-request (cycle steal)
				0111: Auto-request (burst)
				1xxx: Setting prohibited
				• Block Transfer Mode
				0000: Setting prohibited
				0001: Activated by A/D converter conversion end interrupt
				0010: Activated by $\overline{\text{DREQ}}$ pin falling edge input (for the first transfer after data transfer is enabled, activated by $\overline{\text{DREQ}}$ pin low-level input)
				0011: Activated by $\overline{\text{DREQ}}$ pin low-level input
				0100: Activated by SCI channel 0 transmission complete interrupt
				0101: Activated by SCI channel 0 reception complete interrupt
				0110: Activated by SCI channel 1 transmission complete interrupt
				0111: Activated by SCI channel 1 reception complete interrupt
				1000: Activated by TPU channel 0 compare match/input capture A interrupt
				1001: Activated by TPU channel 1 compare match/input capture A interrupt
				1010: Activated by TPU channel 2 compare match/input capture A interrupt
				1011: Activated by TPU channel 3 compare match/input capture A interrupt
				1100: Activated by TPU channel 4 compare match/input capture A interrupt
				1101: Activated by TPU channel 5 compare match/input capture A interrupt
				1110: Setting prohibited
				1111: Setting prohibited

---

2	DTF2	0	R/W	one channel. In this case, activation starts with the highest-priority channel according to the relative channel priorities. For relative channel priorities, see section 7.5.12, Multi-Channel Operation.
1	DTF1	0	R/W	
0	DTF0	0	R/W	

Legend:

×: Don't care

### 7.3.5 DMA Band Control Registers H and L (DMABCRH and DMABCRL)

DMABCR controls the operation of each DMAC channel. The bit functions in the DMACR registers differ according to the transfer mode.

#### Short Address Mode:

- DMABCRH

Bit	Bit Name	Initial Value	R/W	Description
15	FAE1	0	R/W	<p>Full Address Enable 1</p> <p>Specifies whether channel 1 is to be used in short address mode or full address mode. In short address mode, channels 1A and 1B can be used as independent channels.</p> <p>0: Short address mode 1: Full address mode</p>
14	FAE0	0	R/W	<p>Full Address Enable 0</p> <p>Specifies whether channel 0 is to be used in short address mode or full address mode. In short address mode, channels 0A and 0B can be used as independent channels.</p> <p>0: Short address mode 1: Full address mode</p>
13	SAE1	0	R/W	<p>Single Address Enable 1</p> <p>Specifies whether channel 1B is to be used for transfer in dual address mode or single address mode. This bit is invalid in full address mode.</p> <p>0: Dual address mode 1: Single address mode</p>

Specifies whether channel 0B is to be used for transfer in dual address mode or single address mode. This bit is invalid in full address mode.

0: Dual address mode

1: Single address mode

---

11	DTA1B	0	R/W	Data Transfer Acknowledge 1B
10	DTA1A	0	R/W	Data Transfer Acknowledge 1A
9	DTA0B	0	R/W	Data Transfer Acknowledge 0B
8	DTA0A	0	R/W	Data Transfer Acknowledge 0A

These bits enable or disable clearing when DMA transfer is performed for the internal interrupt source selected by the DTF3 to DTF0 bits in DMACR.

If the DTA bit is set to 1 when DTE = 1, the internal interrupt source is cleared automatically by DMA transfer. When DTE = 1 and DTA = 1, the internal interrupt source does not issue an interrupt request to the CPU or DTC.

If the DTA bit is cleared to 0 when DTE = 1, the internal interrupt source is not cleared when a transfer is performed, and can issue an interrupt request to the CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer.

When DTE = 0, the internal interrupt source issues an interrupt request to the CPU or DTC regardless of the DTA bit setting.

---

Bit	Bit Name	Initial Value	R/W	Description
7	DTE1B	0	R/W	Data Transfer Enable 1B
6	DTE1A	0	R/W	Data Transfer Enable 1A
5	DTE0B	0	R/W	Data Transfer Enable 0B
4	DTE0A	0	R/W	Data Transfer Enable 0A

If the DTIE bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.

When DTE = 0, data transfer is enabled and the DMAC ignores the activation source selected by the DTF3 to DTF0 bits in DMACR.

When DTE = 1, data transfer is enabled and the DMAC waits for a request by the activation source selected by the DTF3 to DTF0 bits in DMACR. When a request is issued by the activation source, DMA transfer is executed.

[Clearing conditions]

- When initialization is performed
- When the specified number of transfers have been completed in a transfer mode other than repeat mode
- When 0 is written to the DTE bit to forcibly suspend the transfer, or for a similar reason

[Setting condition]

When 1 is written to the DTE bit after reading DTE = 0



2	DTIE1A	0	R/W	Data Transfer End Interrupt Enable 1A
1	DTIE0B	0	R/W	Data Transfer End Interrupt Enable 0B
0	DTIE0A	0	R/W	Data Transfer End Interrupt Enable 0A

These bits enable or disable an interrupt to the CPU or DTC when transfer ends. If the DTIE bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.

A transfer end interrupt can be canceled either by clearing the DTIE bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE bit to 1.

---

- DMABCRH

Bit	Bit Name	Initial Value	R/W	Description
15	F AE1	0	R/W	<p>Full Address Enable 1</p> <p>Specifies whether channel 1 is to be used in short address mode or full address mode.</p> <p>In full address mode, channels 1A and 1B are used together as channel 1.</p> <p>0: Short address mode</p> <p>1: Full address mode</p>
14	F AE0	0	R/W	<p>Full Address Enable 0</p> <p>Specifies whether channel 0 is to be used in short address mode or full address mode.</p> <p>In full address mode, channels 0A and 0B are used together as channel 0.</p> <p>0: Short address mode</p> <p>1: Full address mode</p>
13, 12	—	All 0	R/W R/W	<p>Reserved</p> <p>Though these bits can be read from or written to, the write value should always be 0.</p>

These bits enable or disable clearing when DMA transfer is performed for the internal interrupt source selected by the DTF3 to DTF0 bits in DMACR of channel 1.

If the DTA1 bit is set to 1 when DTE1 = 1, the internal interrupt source is cleared automatically by DMA transfer. When DTE1 = 1 and DTA1 = 1, the internal interrupt source does not issue an interrupt request to the CPU or DTC.

If the DTA1 bit is cleared to 0 when DTE1 = 1, the internal interrupt source is not cleared when a transfer is performed, and can issue an interrupt request to the CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer.

When DTE1 = 0, the internal interrupt source issues an interrupt request to the CPU or DTC regardless of the DTA1 bit setting.

The state of the DTME1 bit does not affect the above operations.

---

10	—	0	R/W	Reserved
----	---	---	-----	----------

---

Though this bit can be read from or written to, the write value should always be 0.

---

These bits enable or disable clearing when DMA transfer is performed for the internal interrupt source selected by the DTF3 to DTF0 bits in DMACR of channel 0.

If the DTA0 bit is set to 1 when DTE0 = 1, the internal interrupt source is cleared automatically by DMA transfer. When DTE0 = 1 and DTA0 = 1, the internal interrupt source does not issue an interrupt request to the CPU or DTC.

If the DTA0 bit is cleared to 0 when DTE0 = 1, the internal interrupt source is not cleared when a transfer is performed, and can issue an interrupt request to the CPU or DTC in parallel. In this case, the interrupt source should be cleared by the CPU or DTC transfer.

When DTE0 = 0, the internal interrupt source issues an interrupt request to the CPU or DTC regardless of the DTA0 bit setting.

The state of the DTME0 bit does not affect the above operations.

---

8	—	0	R/W	Reserved
---	---	---	-----	----------

---

Though this bit can be read from or written to, the write value should always be 0.

---

Bit	Bit Name	Initial Value	R/W	Description
7	DTME1	0	R/W	<p>Data Transfer Master Enable 1</p> <p>Together with the DTE1 bit, this bit controls enabling or disabling of data transfer on channel 1. When both the DTME1 bit and DTE1 bit are set to 1, transfer is enabled for channel 1.</p> <p>If channel 1 is in the middle of a burst mode transfer when an NMI interrupt is generated, the DTME1 bit is cleared, the transfer is interrupted, and bus mastership passes to the CPU. When the DTME1 bit is subsequently set to 1 again, the interrupted transfer is resumed. In block transfer mode, however, the DTME1 bit is not cleared by an NMI interrupt, and transfer is not interrupted.</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When initialization is performed</li> <li>• When NMI is input in burst mode</li> <li>• When 0 is written to the DTME1 bit</li> </ul> <p>[Setting condition]</p> <p>When 1 is written to DTME1 after reading DTME1 = 0</p>

Enables or disables DMA transfer for the activation source selected by the DTF3 to DTF0 bits in DMACR of channel 1.

When DTE1 = 0, data transfer is disabled and the activation source is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. If the DTIE1 bit is set to 1 when DTE1 = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU.

When DTE1 = 1 and DTME1 = 1, data transfer is enabled and the DMAC waits for a request by the activation source. When a request is issued by the activation source, DMA transfer is executed.

[Clearing conditions]

- When initialization is performed
- When the specified number of transfers have been completed
- When 0 is written to the DTE1 bit to forcibly suspend the transfer, or for a similar reason

[Setting condition]

When 1 is written to the DTE1 bit after reading DTE1 = 0

---

Together with the DTE0 bit, this bit controls enabling or disabling of data transfer on channel 0. When both the DTME0 bit and DTE0 bit are set to 1, transfer is enabled for channel 0.

If channel 0 is in the middle of a burst mode transfer when an NMI interrupt is generated, the DTME0 bit is cleared, the transfer is interrupted, and bus mastership passes to the CPU. When the DTME0 bit is subsequently set to 1 again, the interrupted transfer is resumed. In block transfer mode, however, the DTME0 bit is not cleared by an NMI interrupt, and transfer is not interrupted.

[Clearing conditions]

- When initialization is performed
- When NMI is input in burst mode
- When 0 is written to the DTME0 bit

[Setting condition]

When 1 is written to DTME0 after reading  
DTME0 = 0

---

Enables or disables DMA transfer for the activation source selected by the DTF3 to DTF0 bits in DMACR of channel 0.

When DTE0 = 0, data transfer is disabled and the activation source is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. If the DTIE0 bit is set to 1 when DTE0 = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU.

When DTE0 = 1 and DTME0 = 1, data transfer is enabled and the DMAC waits for a request by the activation source. When a request is issued by the activation source, DMA transfer is executed.

[Clearing conditions]

- When initialization is performed
- When the specified number of transfers have been completed
- When 0 is written to the DTE0 bit to forcibly suspend the transfer, or for a similar reason

[Setting condition]

When 1 is written to the DTE0 bit after reading DTE0 = 0

---

3	DTIE1B	0	R/W	Data Transfer Interrupt Enable 1B
---	--------	---	-----	-----------------------------------

Enables or disables an interrupt to the CPU or DTC when transfer on channel 1 is interrupted. When DTME1 is cleared to 0 while this bit is set to 1, the DMAC regards this as indicating a break in the transfer, and issues a transfer break interrupt request to the CPU or DTC.

A transfer break interrupt can be canceled either by clearing the DTIE1B bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the DTME1 bit to 1.

---



Enables or disables an interrupt to the CPU or DTC when transfer ends. When DTE1 is cleared to 0 while this bit is set to 1, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.

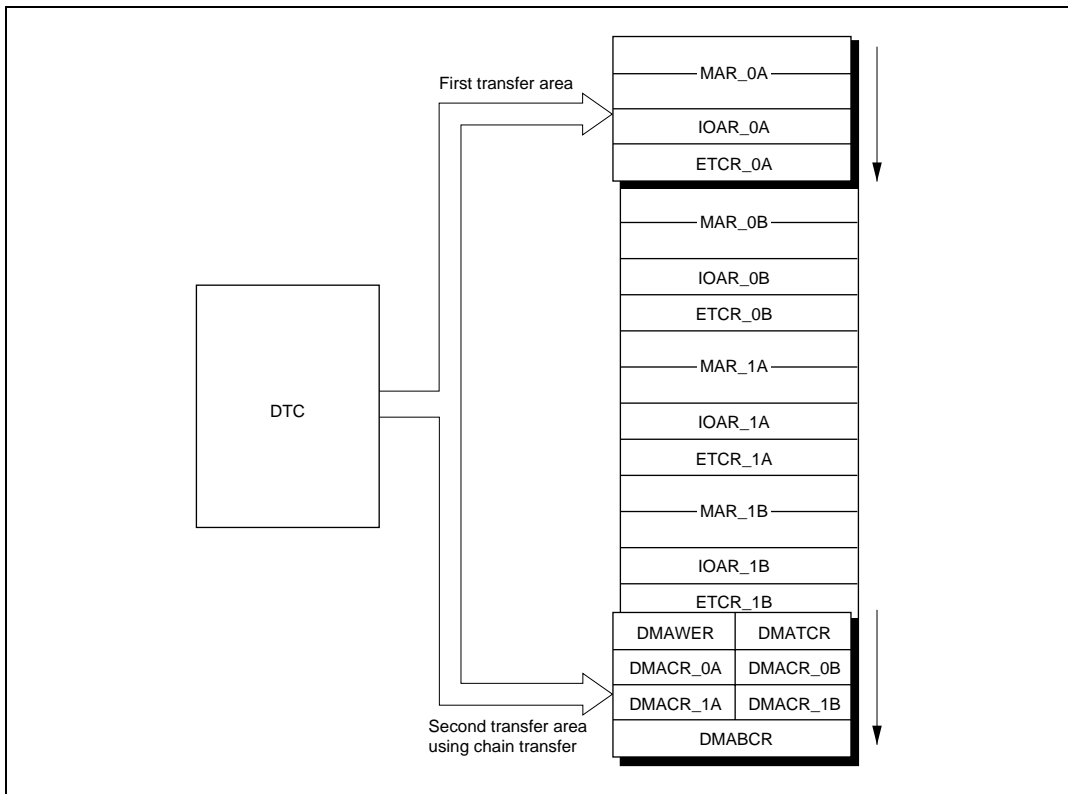
A transfer end interrupt can be canceled either by clearing the DTIE1A bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE1 bit to 1.

1	DTIE0B	0	R/W	<p>Data Transfer Interrupt Enable 0B</p> <p>Enables or disables an interrupt to the CPU or DTC when transfer on channel 1 is interrupted. When DTME0 is cleared to 0 while this bit is set to 1, the DMAC regards this as indicating a break in the transfer, and issues a transfer break interrupt request to the CPU or DTC.</p> <p>A transfer break interrupt can be canceled either by clearing the DTIE0B bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the DTME0 bit to 1.</p>
0	DTIE0A	0	R/W	<p>Data Transfer End Interrupt Enable 0A</p> <p>Enables or disables an interrupt to the CPU or DTC when transfer ends. When DTE0 is cleared to 0 while this bit is set to 1, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.</p> <p>A transfer end interrupt can be canceled either by clearing the DTIE0A bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE0 bit to 1.</p>

The DMAC can activate the DTC with a transfer end interrupt, rewrite the channel on which the transfer ended using a DTC chain transfer, and then reactivate the DTC. DMAWER applies restrictions for changing all bits of DMACR, and specific bits for DMATCR and DMABCR for the specific channel, to prevent inadvertent rewriting of registers other than those for the channel concerned. The restrictions applied by DMAWER are valid for the DTC.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	R	Reserved These bits are always read as 0 and cannot be modified.
3	WE1B	0	R/W	Write Enable 1B Enables or disables writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR. 0: Writes are disabled 1: Writes are enabled
2	WE1A	0	R/W	Write Enable 1A Enables or disables writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR. 0: Writes are disabled 1: Writes are enabled
1	WE0B	0	R/W	Write Enable 0B Enables or disables writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR. 0: Writes are disabled 1: Writes are enabled
0	WE0A	0	R/W	Write Enable 0A Enables or disables writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR. 0: Writes are disabled 1: Writes are enabled

Figure 7.2 shows the transfer areas for activating the DTC with a channel 0A transfer end interrupt request, and reactivating channel 0A. The address register and count register areas are set again during the first DTC transfer, then the control register area is set again during the second DTC



**Figure 7.2 Areas for Register Re-Setting by DTC (Channel 0A)**

Writes by the DTC to bits 15 to 12 (FAE and SAE) in DMABCR are invalid regardless of the DMAWER settings. These bits should be changed, if necessary, by CPU processing.

In writes by the DTC to bits 7 to 4 (DTE) in DMABCR, 1 can be written without first reading 0. To reactivate a channel set to full address mode, write 1 to both Write Enable A and Write Enable B for the channel to be reactivated.

MAR, IOAR, and ETCR can always be written to regardless of the DMAWER settings. When modifying these registers, the channel to be modified should be halted.

DMATCR controls enabling or disabling of output from the DMAC transfer end pin. A port can be set for output automatically, and a transfer end signal output, by setting the appropriate bit.

In short address mode, the  $\overline{\text{TEND}}$  pin is only available for channel B. The transfer end signal indicates the transfer cycle in which the transfer counter has become 0 regardless of the transfer source. Note however that the transfer end signal exceptionally indicates the transfer cycle in which the block counter has become 0 in block transfer mode.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	TEE1	0	R/W	Transfer End Enable 1 Enables or disables transfer end pin 1 ( $\overline{\text{TEND1}}$ ) output. 0: $\overline{\text{TEND1}}$ pin output disabled 1: $\overline{\text{TEND1}}$ pin output enabled
4	TEE0	0	R/W	Transfer End Enable 0 Enables or disables transfer end pin 0 ( $\overline{\text{TEND0}}$ ) output. 0: $\overline{\text{TEND0}}$ pin output disabled 1: $\overline{\text{TEND0}}$ pin output enabled
3 to 0	—	0	—	Reserved These bits are always read as 0 and cannot be modified.

DMAC activation sources consist of internal interrupt requests, external requests, and auto-requests. The DMAC activation sources that can be specified depend on the transfer mode and channel, as shown in table 7.3.

**Table 7.3 DMAC Activation Sources**

Activation Source		Short Address Mode		Full Address Mode	
		Channels 0A and 1A	Channels 0B and 1B	Normal Mode	Block Transfer Mode
Internal interrupts	ADI	O	O	X	O
	TXI0	O	O	X	O
	RXI0	O	O	X	O
	TXI1	O	O	X	O
	RXI1	O	O	X	O
	TGI0A	O	O	X	O
	TGI1A	O	O	X	O
	TGI2A	O	O	X	O
	TGI3A	O	O	X	O
	TGI4A	O	O	X	O
	TGI5A	O	O	X	O
External requests	$\overline{\text{DREQ}}$ pin falling edge input	X	O	O	O
	$\overline{\text{DREQ}}$ pin low-level input	X	O	O	O
Auto-request		X	X	O	X

Legend:

O: Can be specified

X: Cannot be specified

An interrupt request selected as a DMAC activation source can also simultaneously generate an interrupt request for the CPU or DTC. For details, see section 5, Interrupt Controller.

With activation by an internal interrupt request, the DMAC accepts the interrupt request independently of the interrupt controller. Consequently, interrupt controller priority settings are irrelevant.

If the DMAC is activated by a CPU interrupt source or an interrupt request that is not used as a DTC activation source ( $DTA = 1$ ), the interrupt request flag is cleared automatically by the DMA transfer. With ADI, TXI and RXI interrupts, however, the interrupt source flag is not cleared unless the relevant register is accessed in a DMA transfer. If the same interrupt is used as an activation source for more than one channel, the interrupt request flag is cleared when the highest-priority channel is activated. Transfer requests for other channels are held pending in the DMAC, and activation is carried out in order of priority.

When  $DTE = 0$  after completion of a transfer, an interrupt request from the selected activation source is not sent to the DMAC, regardless of the DTA bit setting. In this case, the relevant interrupt request is sent to the CPU or DTC.

In case of overlap with a CPU interrupt source or DTC activation source ( $DTA = 0$ ), the interrupt request flag is not cleared by the DMAC.

#### **7.4.2 Activation by External Request**

If an external request ( $\overline{DREQ}$  pin) is specified as a DMAC activation source, the relevant port should be set to input mode in advance\*. Level sensing or edge sensing can be used for external requests.

External request operation in normal mode of short address mode or full address mode is described below.

When edge sensing is selected, a byte or word is transferred each time a high-to-low transition is detected on the  $\overline{DREQ}$  pin. The next data transfer may not be performed if the next edge is input before data transfer is completed.

When level sensing is selected, the DMAC stands by for a transfer request while the  $\overline{DREQ}$  pin is held high. While the  $\overline{DREQ}$  pin is held low, transfers continue in succession, with the bus being released each time a byte or word is transferred. If the  $\overline{DREQ}$  pin goes high in the middle of a transfer, the transfer is interrupted and the DMAC stands by for a transfer request.

### 7.4.3 Activation by Auto-Request

Auto-request is activated by register setting only, and transfer continues to the end. With auto-request activation, cycle steal mode or burst mode can be selected.

In cycle steal mode, the DMAC releases the bus to another bus master each time a byte or word is transferred. DMA and CPU cycles are usually repeated alternately. In burst mode, the DMAC keeps possession of the bus until the end of the transfer so that transfer is performed continuously.

## 7.5.1 Transfer Modes

Table 7.4 lists the DMAC transfer modes.

**Table 7.4 DMAC Transfer Modes**

<b>Transfer Mode</b>	<b>Transfer Source</b>	<b>Remarks</b>
Short address mode	Dual address mode <ul style="list-style-type: none"><li>• 1-byte or 1-word transfer for a single transfer request</li><li>• Specifies the transfer destination/source address and performs transfer in 2 bus cycles</li></ul> (1) Sequential mode <ul style="list-style-type: none"><li>• Memory address incremented or decremented by 1 or 2</li><li>• Number of transfers: 1 to 65,536</li></ul> (2) Idle mode <ul style="list-style-type: none"><li>• Memory address fixed</li><li>• Number of transfers: 1 to 65,536</li></ul> (3) Repeat mode <ul style="list-style-type: none"><li>• 1-byte or 1-word transfer for a single transfer request</li><li>• Memory address incremented or decremented by 1 or 2</li><li>• Continues transfer after sending number of transfers (1 to 256) and restoring the initial value</li></ul>	<ul style="list-style-type: none"><li>• TPU channel 0 to 5 compare match/input capture A interrupt</li><li>• SCI transmission complete interrupt</li><li>• SCI reception complete interrupt</li><li>• A/D converter conversion end interrupt</li><li>• External request</li></ul> <ul style="list-style-type: none"><li>• Up to 4 channels can operate independently</li><li>• External request applies to channel B only</li><li>• Single address mode applies to channel B only</li></ul>

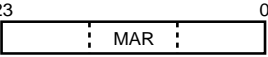
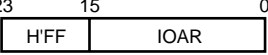
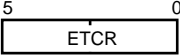


address mode	<ul style="list-style-type: none"> <li>• 1-byte or 1-word transfer for a single transfer request</li> <li>• 1-bus cycle transfer by means of <math>\overline{DACK}</math> pin instead of using address for specifying I/O</li> <li>• Sequential mode, idle mode, or repeat mode can be specified</li> </ul>	<ul style="list-style-type: none"> <li>• compare match/input capture A interrupt</li> <li>• SCI transmission complete interrupt</li> <li>• SCI reception complete interrupt</li> <li>• A/D converter conversion end interrupt</li> <li>• External request</li> </ul>	<ul style="list-style-type: none"> <li>• operate independently</li> <li>• External request applies to channel B only</li> <li>• Single address mode applies to channel B only</li> </ul>
Full address mode	Normal mode	<ul style="list-style-type: none"> <li>• Auto-request</li> </ul>	<ul style="list-style-type: none"> <li>• Max. 2-channel operation, combining channels A and B</li> </ul>
	(1) Auto-request	<ul style="list-style-type: none"> <li>• Transfer request is internally held</li> <li>• Number of transfers (1 to 65,536) is continuously sent</li> <li>• Burst/cycle steal transfer can be selected</li> </ul>	
	(2) External request	<ul style="list-style-type: none"> <li>• External request</li> </ul>	
	Block transfer mode	<ul style="list-style-type: none"> <li>• TPU channel 0 to 5</li> <li>• compare match/input capture A interrupt</li> <li>• SCI transmission complete interrupt</li> <li>• SCI reception complete interrupt</li> <li>• A/D converter conversion end interrupt</li> <li>• External request</li> </ul>	
	<ul style="list-style-type: none"> <li>• Transfer of 1-block, size selected for a single transfer request</li> <li>• Number of transfers: 1 to 65,536</li> <li>• Source or destination can be selected as block area</li> <li>• Block size: 1 to 256 bytes or word</li> </ul>		

Sequential mode can be specified by clearing the RPE bit in DMACR to 0. In sequential mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCR. One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR.

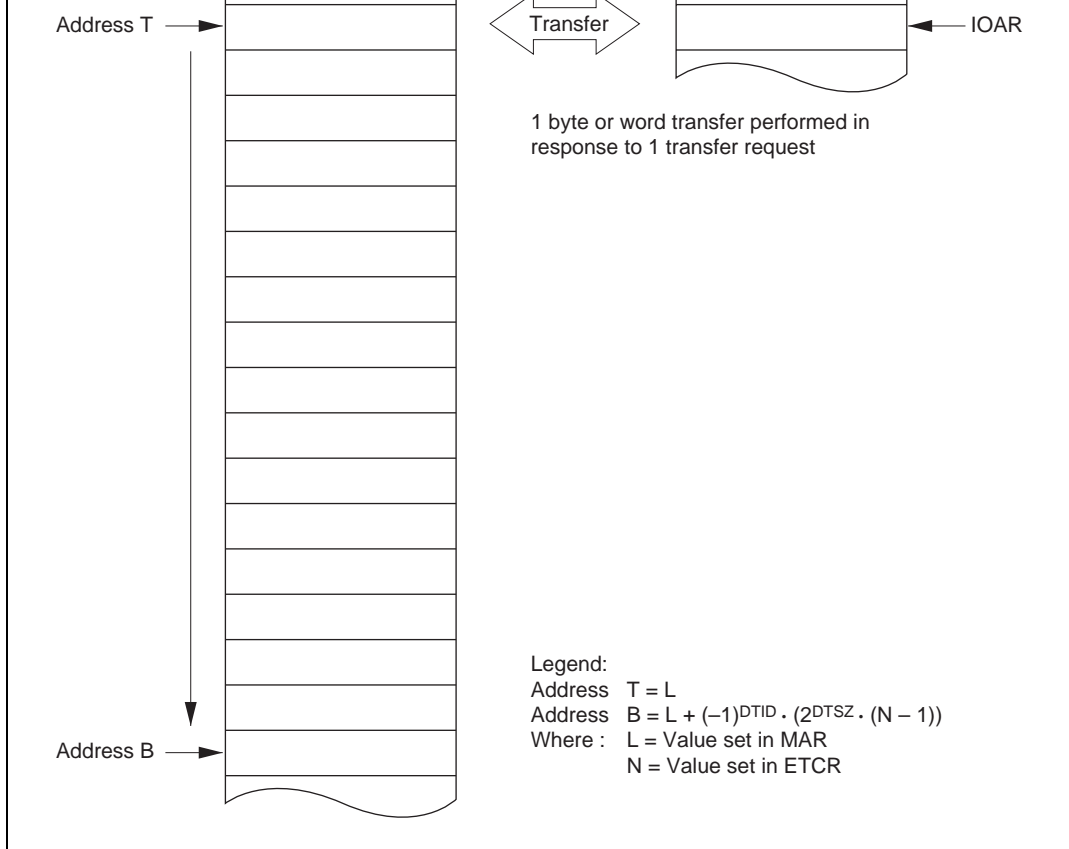
Table 7.5 summarizes register functions in sequential mode.

**Table 7.5 Register Functions in Sequential Mode**

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/decrypted every transfer
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Transfer counter		Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is incremented or decremented by 1 or 2 each time a byte or word is transferred. IOAR specifies the lower 16 bits of the other address. The 8 bits above IOAR have a value of H'FF.

Figure 7.3 illustrates operation in sequential mode.

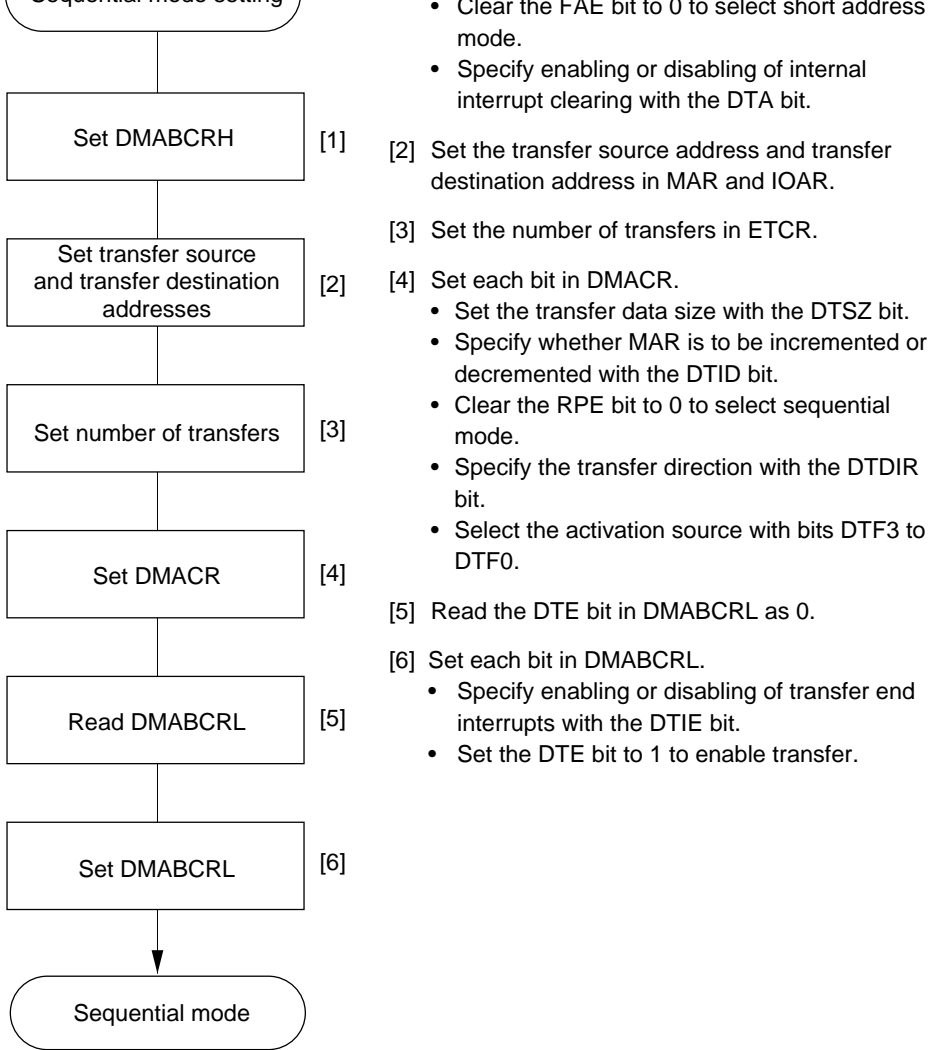


**Figure 7.3 Operation in Sequential Mode**

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a data transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and data transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC. The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can only be specified for channel B.

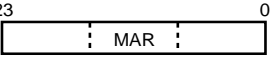
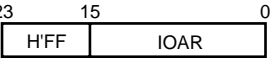

Figure 7.4 shows an example of the setting procedure for sequential mode.



**Figure 7.4 Example of Sequential Mode Setting Procedure**

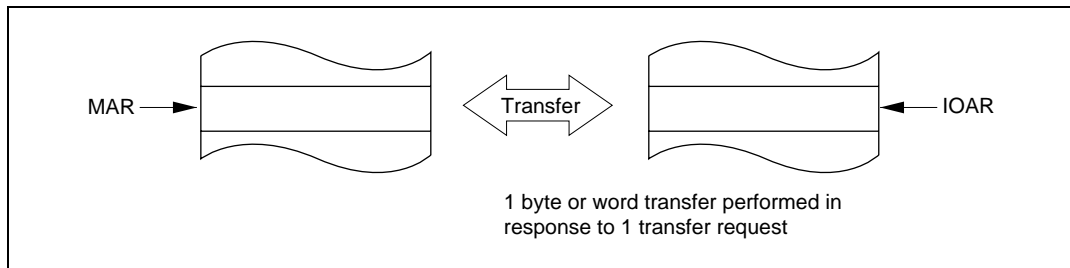
Idle mode can be specified by setting the RPE bit in DMACR and DTIE bit in DMABCRL to 1. In idle mode, one byte or word is transferred in response to a single transfer request, and this is executed the number of times specified in ETCR. One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR. Table 7.6 summarizes register functions in idle mode.

**Table 7.6 Register Functions in Idle Mode**

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Fixed
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Transfer counter		Number of transfers	Decremented every transfer; transfer ends when count reaches H'0000

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is neither incremented nor decremented by a data transfer. IOAR specifies the lower 16 bits of the other address. The upper 8 bits of IOAR have a value of H'FF.

Figure 7.5 illustrates operation in idle mode.

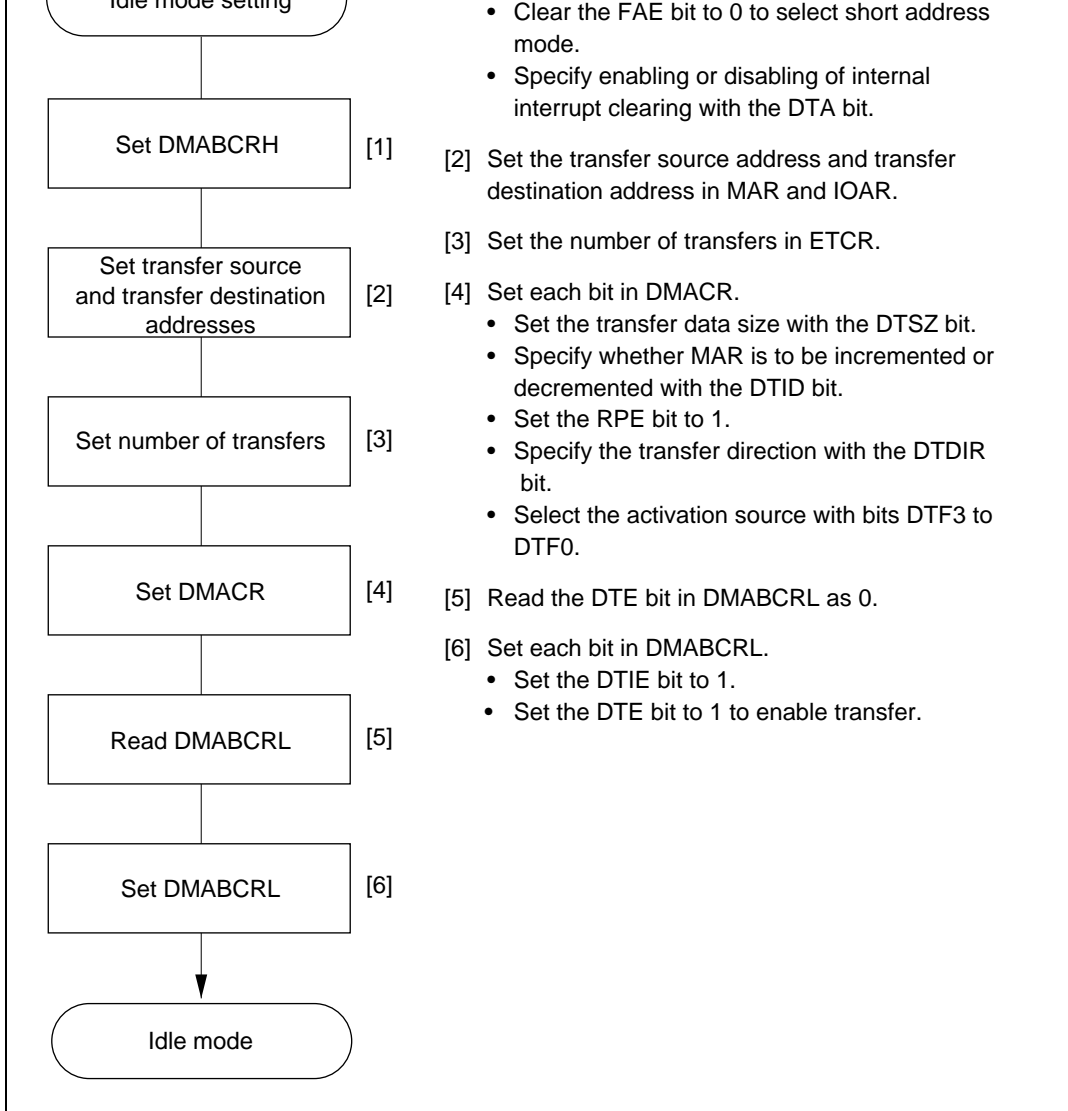


**Figure 7.5 Operation in Idle Mode**

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and data transfer

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can only be specified for channel B.

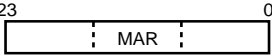
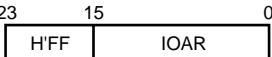


Figure 7.6 shows an example of the setting procedure for idle mode.



**Figure 7.6 Example of Idle Mode Setting Procedure**

Repeat mode can be specified by setting the RPE bit in DMACR to 1, and clearing the DTIE bit in DMABCRL to 0. In repeat mode, MAR is updated after each byte or word transfer in response to a single transfer request, and this is executed the number of times specified in ETCRL. On completion of the specified number of transfers, MAR and ETCRL are automatically restored to their original settings and operation continues. One address is specified by MAR, and the other by IOAR. The transfer direction can be specified by the DTDIR bit in DMACR. Table 7.7 summarizes register functions in repeat mode.

**Table 7.7 Register Functions in Repeat Mode**

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/decrypted every transfer. Initial setting is restored when value reaches H'0000
	Destination address register	Source address register	Start address of transfer source or transfer destination	Fixed
	Holds number of transfers		Number of transfers	Fixed
	Transfer counter		Number of transfers	Decremented every transfer. Loaded with ETCRH value when count reaches H'00

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is incremented or decremented by 1 or 2 each time a byte or word is transferred. IOAR specifies the lower 16 bits of the other address. The upper 8 bits of IOAR have a value of H'FF. The number of transfers is specified as 8 bits by ETCRH and ETCRL. The maximum number of transfers, when H'00 is set in both ETCRH and ETCRL, is 256.

In repeat mode, ETCRL functions as the transfer counter, and ETCRH is used to hold the number of transfers. ETCRL is decremented by 1 each time a data transfer is executed, and when its value reaches H'00, it is loaded with the value in ETCRH. At the same time, the value set in MAR is

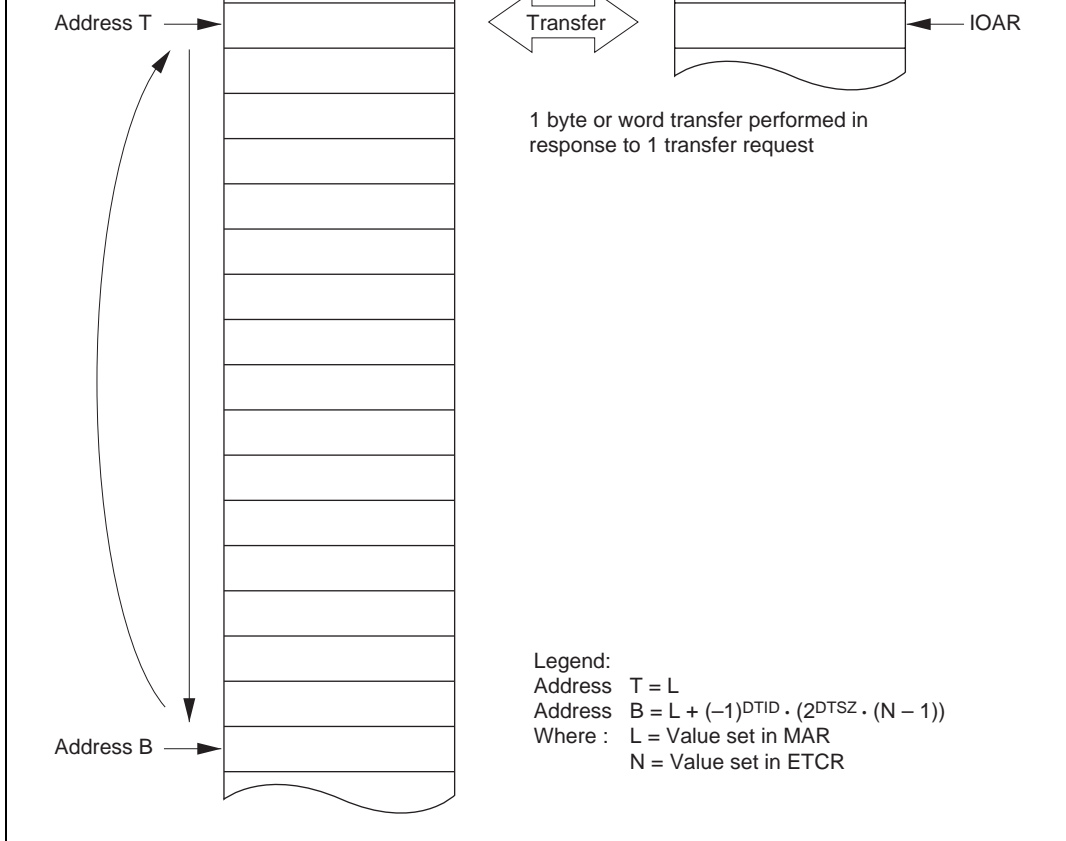


$$\text{MAR} = \text{MAR} - (-1)^{\text{DTID}} \cdot 2^{\text{DTSZ}} \cdot \text{ETCRH}$$

The same value should be set in ETCRH and ETCRL.

In repeat mode, operation continues until the DTE bit in DMABCRL is cleared. To end the transfer operation, therefore, the DTE bit should be cleared to 0. A transfer end interrupt request is not sent to the CPU or DTC. By setting the DTE bit to 1 again after it has been cleared, the operation can be restarted from the transfer after that terminated when the DTE bit was cleared.

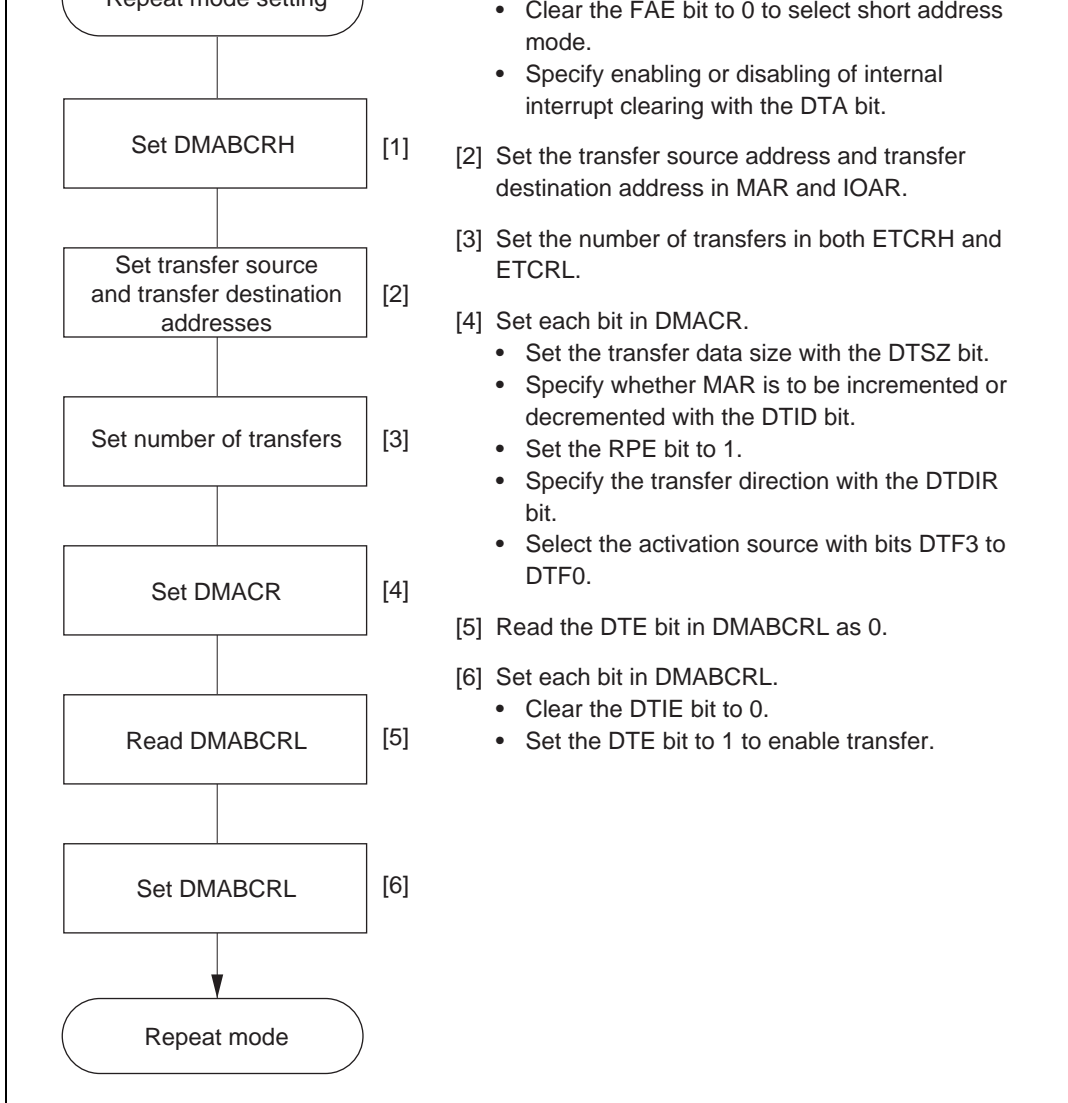
Figure 7.7 illustrates operation in repeat mode.



**Figure 7.7 Operation in Repeat mode**

Transfer requests (activation sources) consist of A/D converter conversion end interrupts, external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 5 compare match/input capture A interrupts. External requests can only be specified for channel B.

Figure 7.8 shows an example of the setting procedure for repeat mode.

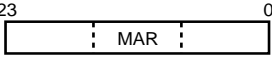
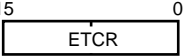


**Figure 7.8 Example of Repeat Mode Setting Procedure**

Single address mode can only be specified by B. This mode will be specified by setting the SAE bit in DMABCRH to 1 in short address mode.

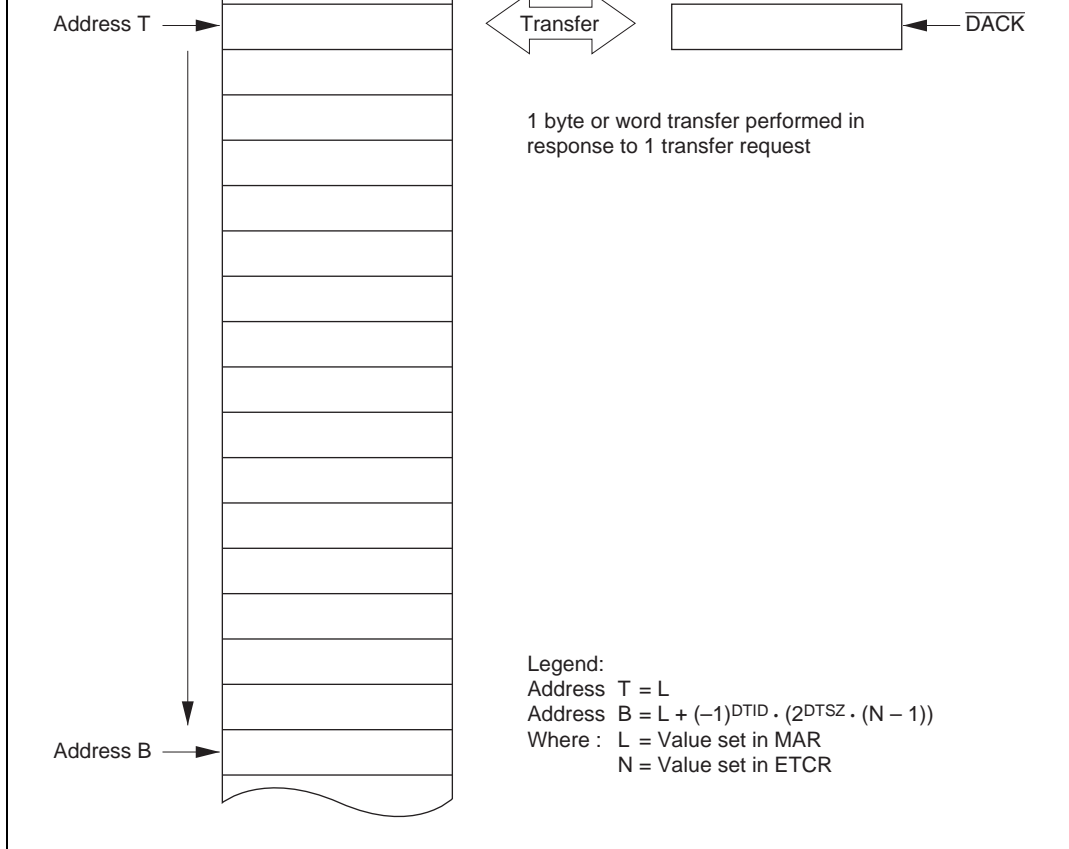
One address is specified by MAR, and the other is set automatically to the data transfer acknowledge pin ( $\overline{DACK}$ ). The transfer direction can be specified by the DTDIR bit in DMACR. Table 7.8 summarizes register functions in single address mode.

**Table 7.8 Register Functions in Single Address Mode**

Register	Function		Initial Setting	Operation
	DTDIR = 0	DTDIR = 1		
	Source address register	Destination address register	Start address of transfer destination or transfer source	See sections 7.5.2, Sequential Mode, 7.5.3, Idle Mode, and 7.5.4, Repeat Mode.
$\overline{DACK}$ pin	Write strobe	Read strobe	(Set automatically by SAE bit; IOAR is invalid)	Strobe for external device
	Transfer counter		Number of transfers	See sections 7.5.2, Sequential Mode, 7.5.3, Idle Mode, and 7.5.4, Repeat Mode.

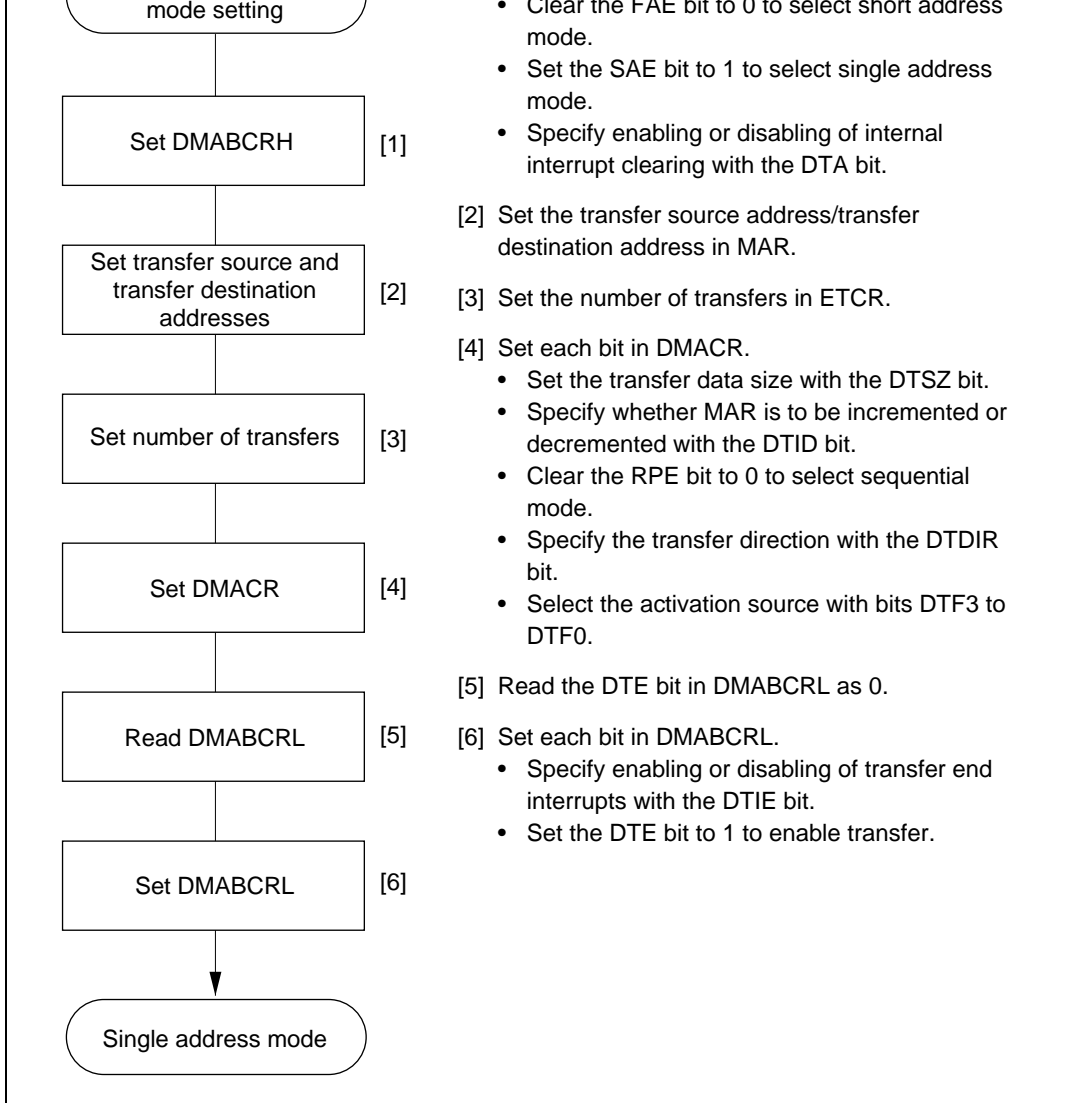
MAR specifies the start address of the transfer source or transfer destination as 24 bits. IOAR is invalid; in its place the strobe for external devices ( $\overline{DACK}$ ) is output.

Figure 7.9 illustrates operation in single address mode (when sequential mode is specified).



**Figure 7.9 Operation in Single Address Mode (When Sequential Mode is Specified)**




Figure 7.10 shows an example of the setting procedure for single address mode (when sequential mode is specified).



**Figure 7.10 Example of Single Address Mode Setting Procedure (When Sequential Mode is Specified)**

In normal mode, transfer is performed with channels A and B used in combination. Normal mode can be specified by setting the FAE bit in DMABCRH to 1 and clearing the BLKE bit in DMACRA to 0. In normal mode, MAR is updated after data transfer of a byte or word in response to a single transfer request, and this is executed the number of times specified in ETCRA. The transfer source is specified by MARA, and the transfer destination by MARB. Table 7.9 summarizes register functions in normal mode.

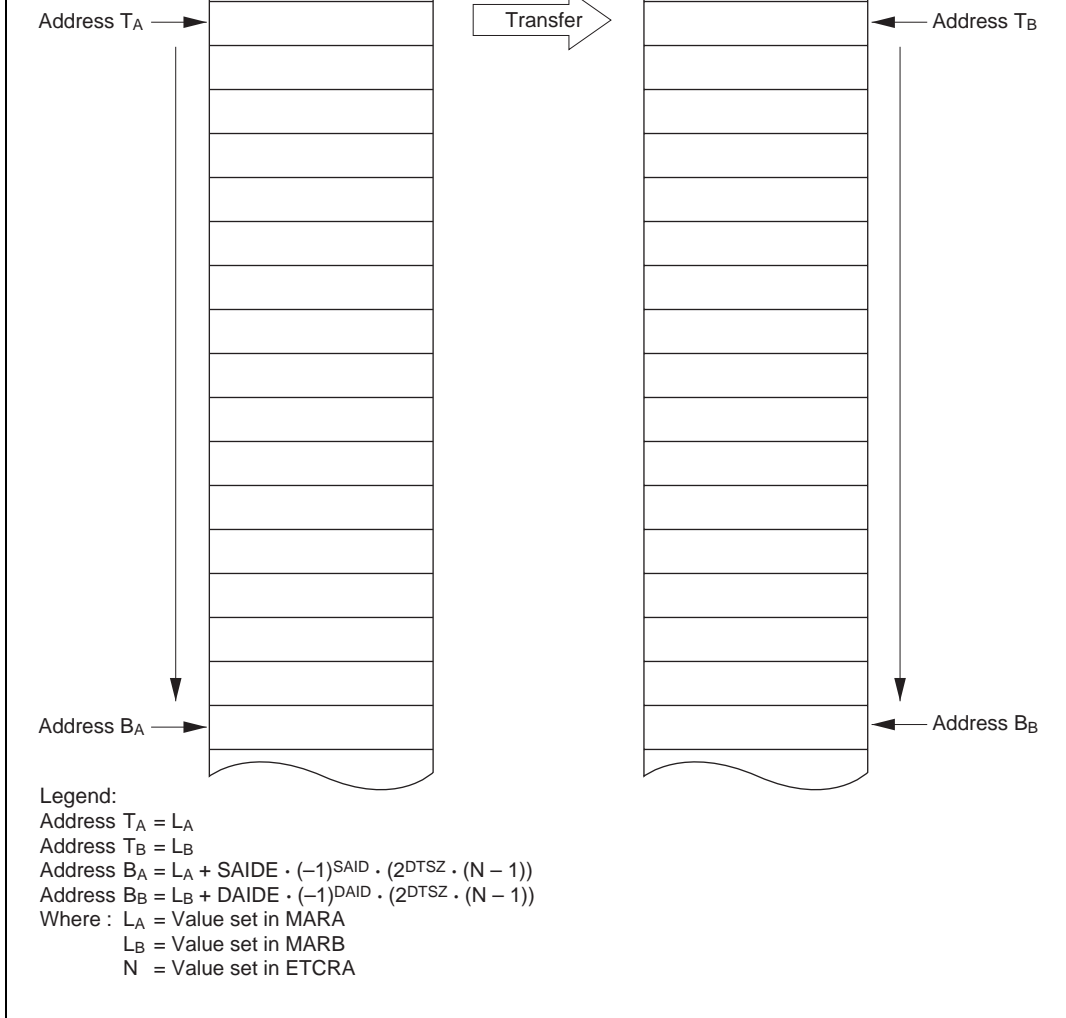
**Table 7.9 Register Functions in Normal Mode**

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/decremented every transfer, or fixed
	Destination address register	Start address of transfer destination	Incremented/decremented every transfer, or fixed
	Transfer counter	Number of transfers	Decrement every transfer; transfer ends when count reaches H'0000

MARA and MARB specify the start addresses of the transfer source and transfer destination, respectively, as 24 bits. MAR can be incremented or decremented by 1 or 2 each time a byte or word is transferred, or can be fixed. Incrementing, decrementing, or holding a fixed value can be set separately for MARA and MARB.

The number of transfers is specified by ETCRA as 16 bits. ETCRA is decremented by 1 each time a transfer is performed, and when its value reaches H'0000 the DTE bit in DMABCRL is cleared and transfer ends. If the DTIE bit in DMABCRL is set to 1 at this time, an interrupt request is sent to the CPU or DTC. The maximum number of transfers, when H'0000 is set in ETCRA, is 65,536.

Figure 7.11 illustrates operation in normal mode.

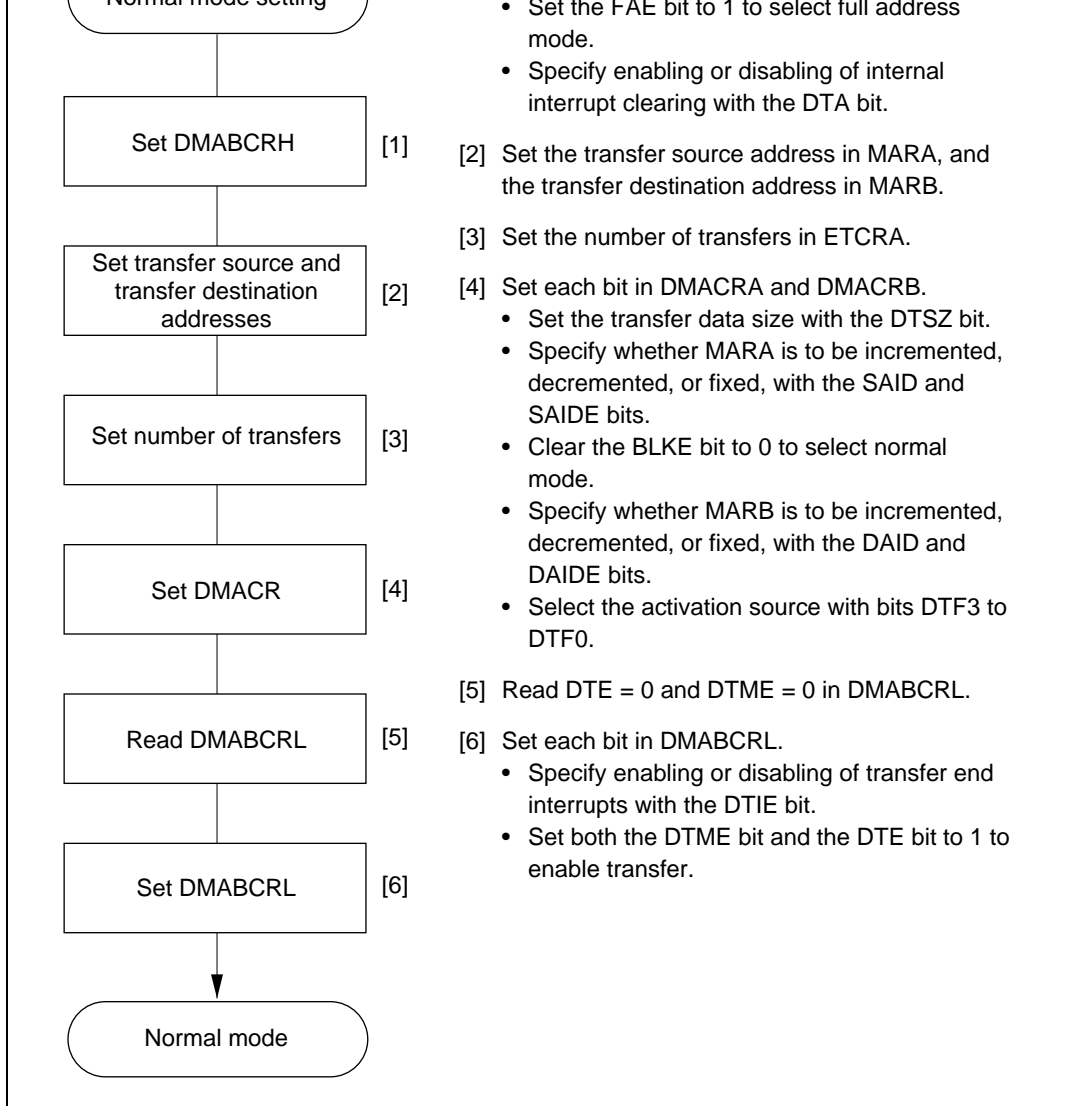


**Figure 7.11 Operation in Normal Mode**

Transfer requests (activation sources) are external requests and auto-requests. With auto-request, the DMAC is only activated by register setting, and the specified number of transfers are performed automatically. With auto-request, cycle steal mode or burst mode can be selected. In cycle steal mode, the bus is released to another bus master each time a transfer is performed. In burst mode, the bus is held continuously until transfer ends.

Figure 7.12 shows an example of the setting procedure for normal mode.





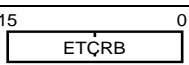




**Figure 7.12 Example of Normal Mode Setting Procedure**

In block transfer mode, data transfer is performed with channels A and B used in combination. Block transfer mode can be specified by setting the FAE bit in DMABCRH and the BLKE bit in DMACRA to 1. In block transfer mode, a data transfer of the specified block size is carried out in response to a single transfer request, and this is executed for the number of times specified in ETCRB. The transfer source is specified by MARA, and the transfer destination by MARB. Either the transfer source or the transfer destination can be selected as a block area (an area composed of a number of bytes or words). Table 7.10 summarizes register functions in block transfer mode.

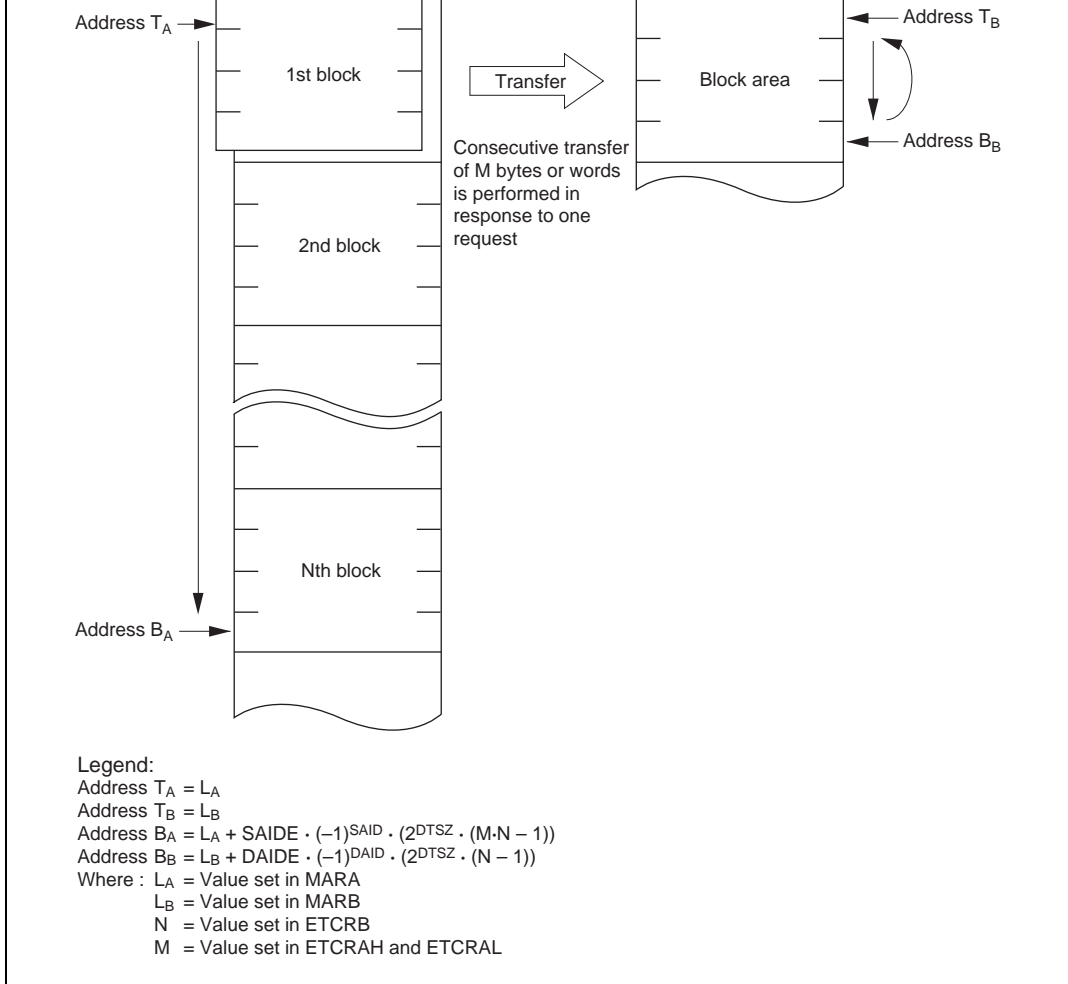
**Table 7.10 Register Functions in Block Transfer Mode**

Register	Function	Initial Setting	Operation
	Source address register	Start address of transfer source	Incremented/decremented every transfer, or fixed
	Destination address register	Start address of transfer destination	Incremented/decremented every transfer, or fixed
	Holds block size	Block size	Fixed
	Block size counter	Block size	Decrement every transfer; ETCRH value copied when count reaches H'00
	Block transfer counter	Number of block transfers	Decrement every block transfer; transfer ends when count reaches H'0000

MARA and MARB specify the start addresses of the transfer source and transfer destination, respectively, as 24 bits. MAR can be incremented or decremented by 1 or 2 each time a byte or word is transferred, or can be fixed. Incrementing, decrementing, or holding a fixed value can be set separately for MARA and MARB. Whether a block is to be designated for MARA or for MARB is specified by the BLKDIR bit in DMACRA.

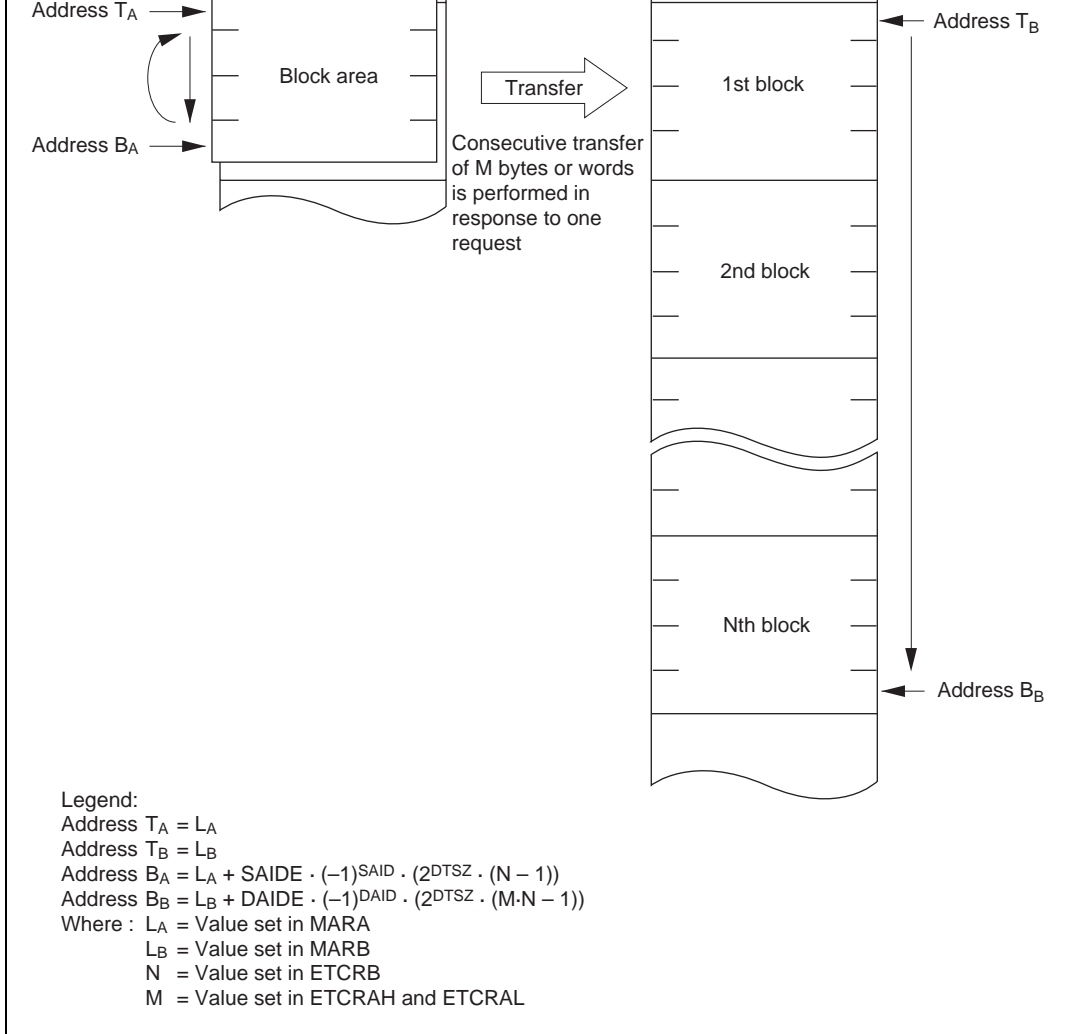
To specify the number of transfers, if M is the size of one block (where M = 1 to 256) and N transfers are to be performed (where N = 1 to 65,536), M is set in both ETCRAH and ETCRAL, and N in ETCRB.

Figure 7.13 illustrates operation in block transfer mode when MARB is designated as a block area.



**Figure 7.13 Operation in Block Transfer Mode (BLKDIR = 0)**

Figure 7.14 illustrates operation in block transfer mode when MARA is designated as a block area.



**Figure 7.14 Operation in Block Transfer Mode (BLKDIR = 1)**

ETCRAL is decremented by 1 each time a byte or word transfer is performed. In response to a single transfer request, burst transfer is performed until the value in ETCRAL reaches H'00. ETCRAL is then loaded with the value in ETCRAH. At this time, the value in the MAR register for which a block designation has been given by the BLKDIR bit in DMACRA is restored in accordance with the DTSZ, SAID/DAID, and SAIDE/DAIDE bits in DMACR.

this point, an interrupt request is sent to the CPU or DTC.

Figure 7.15 shows the operation flow in block transfer mode.

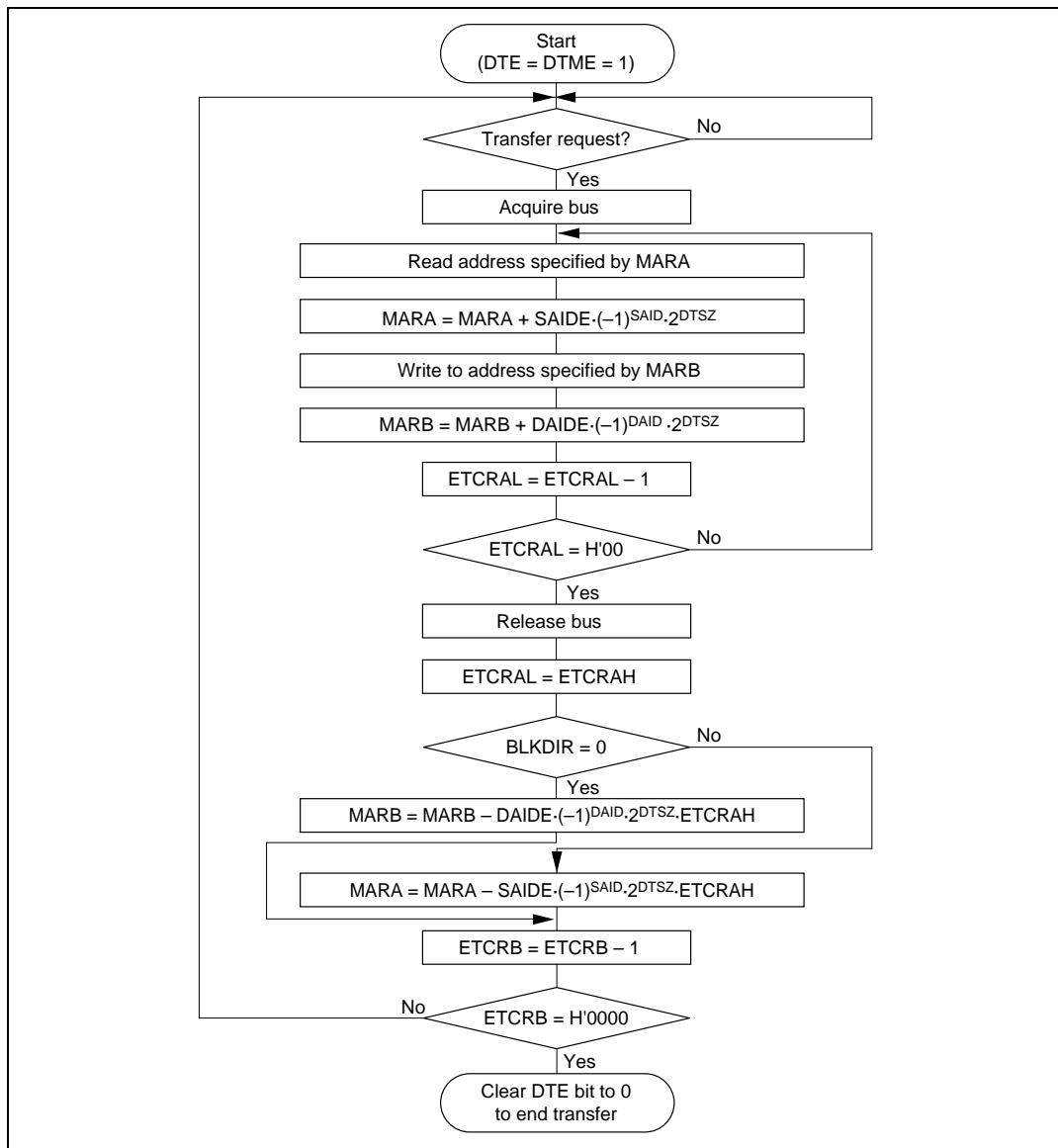
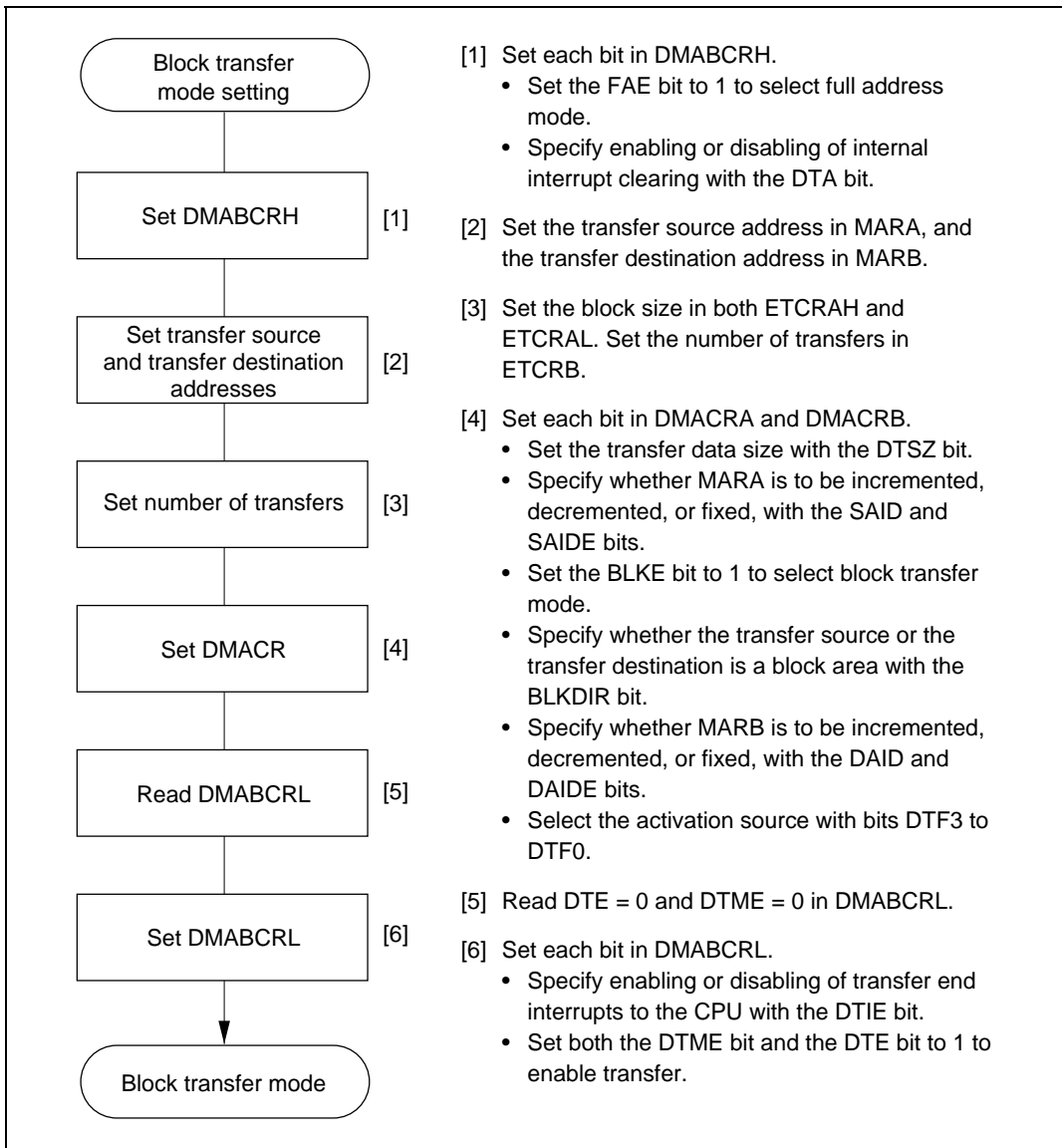


Figure 7.15 Operation Flow in Block Transfer Mode

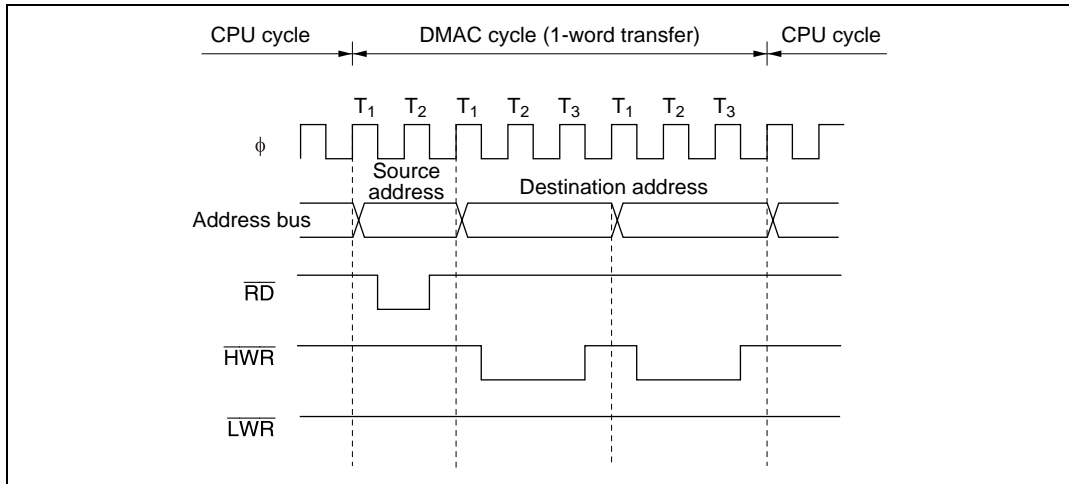
Figure 7.16 shows an example of the setting procedure for block transfer mode.



**Figure 7.16 Example of Block Transfer Mode Setting Procedure**

An example of the basic DMAC timing is shown in figure 7.17. In this example, word-size transfer is performed from 16-bit, 2-state access space to 8-bit, 3-state access space. When the bus is transferred from the CPU to the DMAC, a source address read and destination address write are performed. The bus is not released in response to another bus request, etc., between these read and write operations. As like CPU cycles, DMA cycles conform to the bus controller settings.

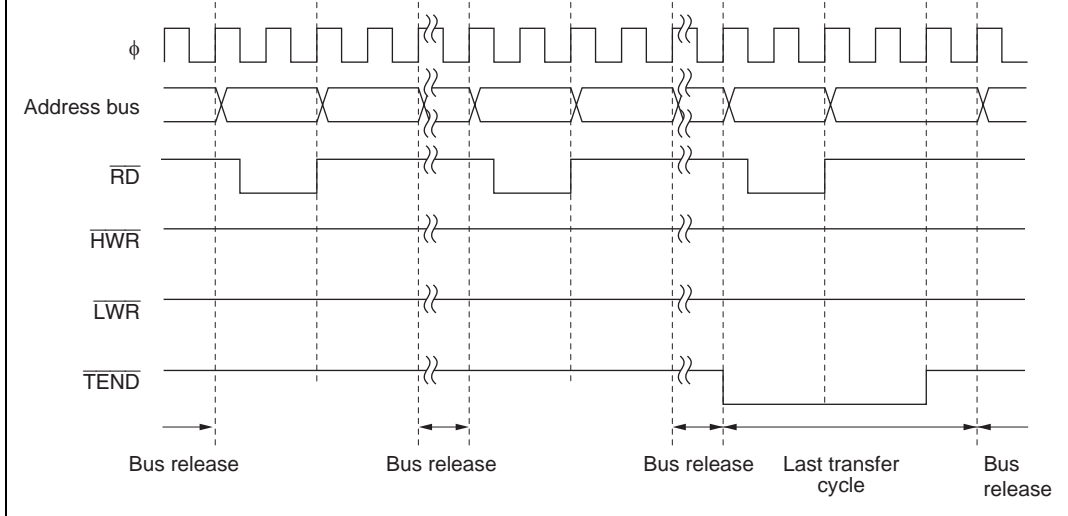
The address is not output to the external address bus in an access to on-chip memory or an internal I/O register.



**Figure 7.17 Example of DMA Transfer Bus Timing**

### 7.5.9 DMA Transfer (Dual Address Mode) Bus Cycles

**Short Address Mode:** Figure 7.18 shows a transfer example in which  $\overline{TEND}$  output is enabled and byte-size short address mode transfer (sequential/idle/repeat mode) is performed from external 8-bit, 2-state access space to internal I/O space.



**Figure 7.18 Example of Short Address Mode Transfer**

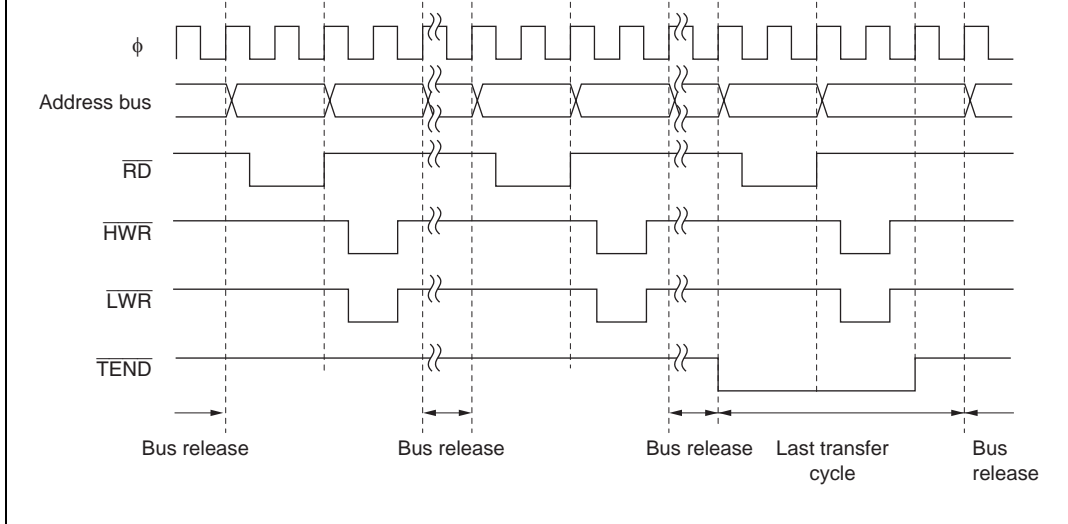
A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

In repeat mode, when  $\overline{\text{TEND}}$  output is enabled,  $\overline{\text{TEND}}$  output goes low in the transfer end cycle.

**Full Address Mode (Cycle Steal Mode):** Figure 7.19 shows a transfer example in which  $\overline{\text{TEND}}$  output is enabled and word-size full address mode transfer (cycle steal mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.



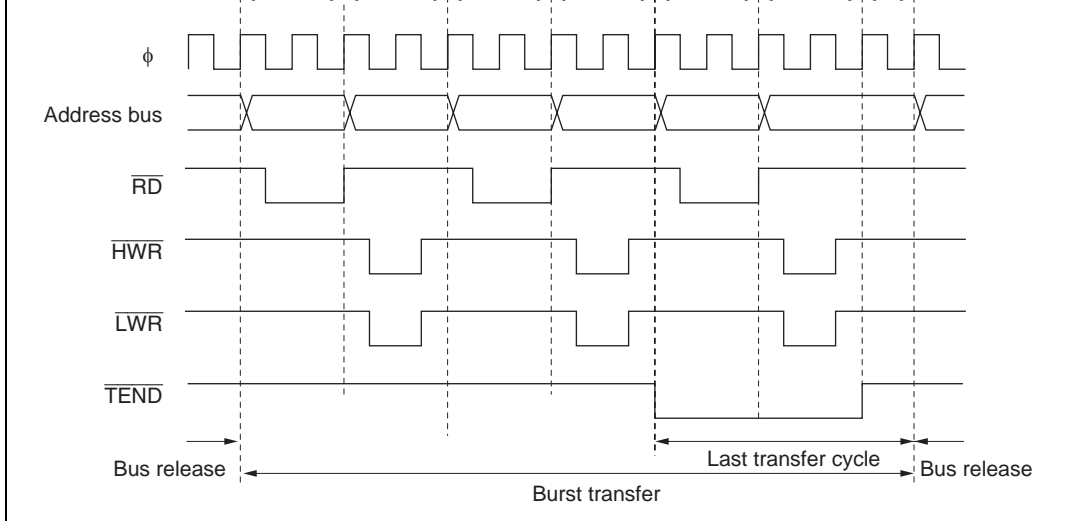


**Figure 7.19 Example of Full Address Mode Transfer (Cycle Steal)**

A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one bus cycle is executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

**Full Address Mode (Burst Mode):** Figure 7.20 shows a transfer example in which  $\overline{TEND}$  output is enabled and word-size full address mode transfer (burst mode) is performed from external 16-bit, 2-state access space to external 16-bit, 2-state access space.



**Figure 7.20 Example of Full Address Mode Transfer (Burst Mode)**

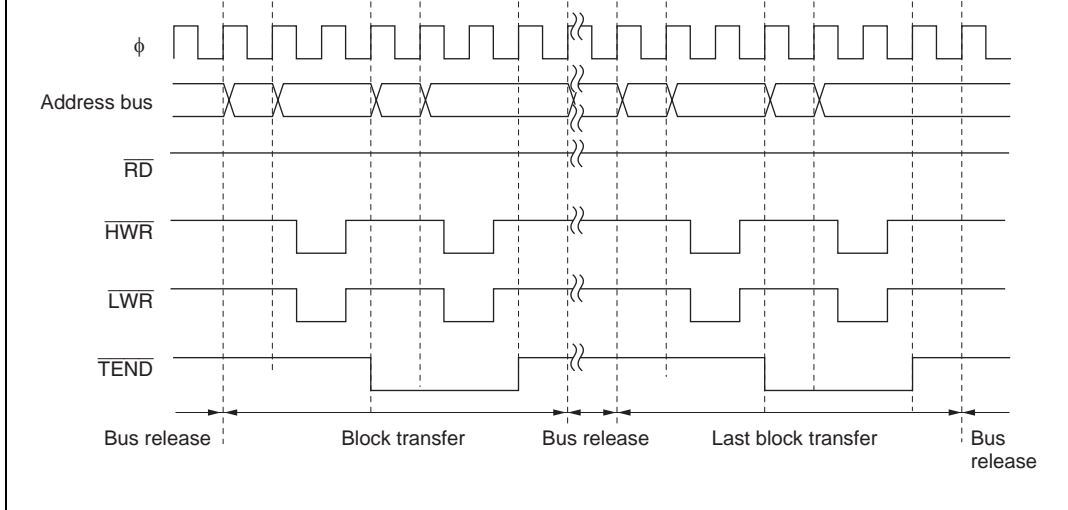
In burst mode, one-byte or one-word transfers are executed consecutively until transfer ends.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

If a request from another higher-priority channel is generated after burst transfer starts, that channel has to wait until the burst transfer ends.

If an NMI interrupt is generated while a channel designated for burst transfer is in the transfer enabled state, the DTME bit in DMABCRL is cleared and the channel is placed in the transfer disabled state. If burst transfer has already been activated inside the DMAC, the bus is released on completion of a one-byte or one-word transfer within the burst transfer, and burst transfer is suspended. If the last transfer cycle of the burst transfer has already been activated inside the DMAC, execution continues to the end of the transfer even if the DTME bit is cleared.

**Full Address Mode (Block Transfer Mode):** Figure 7.21 shows a transfer example in which TEND output is enabled and word-size full address mode transfer (block transfer mode) is performed from internal 16-bit, 1-state access space to external 16-bit, 2-state access space.



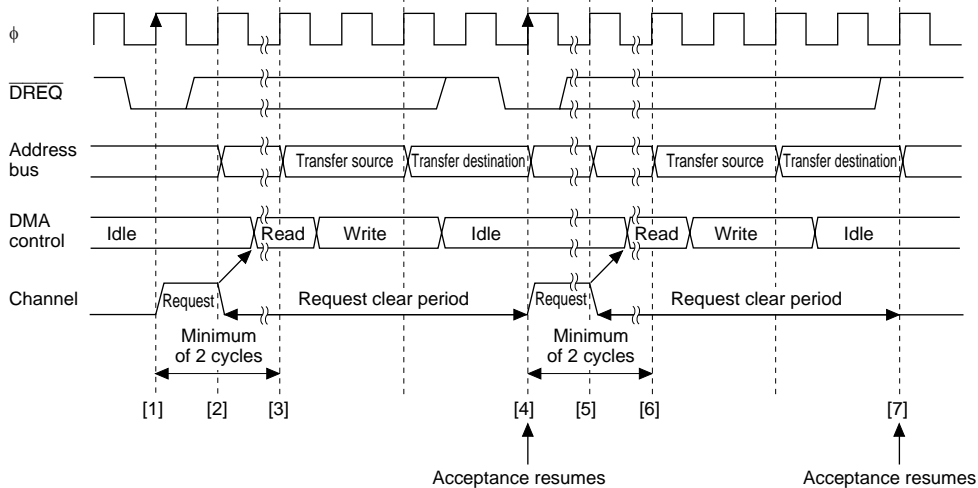
**Figure 7.21 Example of Full Address Mode Transfer (Block Transfer Mode)**

A one-block transfer is performed for a single transfer request, and after the transfer the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle of each block (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle. Even if an NMI interrupt is generated during data transfer, block transfer operation is not affected until data transfer for one block has ended.

**DREQ Pin Falling Edge Activation Timing:** Set the DTA bit in DMABCRH to 1 for the channel for which the DREQ pin is selected.

Figure 7.22 shows an example of normal mode transfer activated by the DREQ pin falling edge.



- [1] Acceptance after transfer enabling; the  $\overline{\text{DREQ}}$  pin low level is sampled on the rising edge of  $\phi$ , and the request is held.
- [2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
- [3] [6] Start of DMA cycle;  $\overline{\text{DREQ}}$  pin high level sampling on the rising edge of  $\phi$  starts.
- [4] [7] When the  $\overline{\text{DREQ}}$  pin high level has been sampled, acceptance is resumed after the write cycle is completed.  
(As in [1], the  $\overline{\text{DREQ}}$  pin low level is sampled on the rising edge of  $\phi$ , and the request is held.)

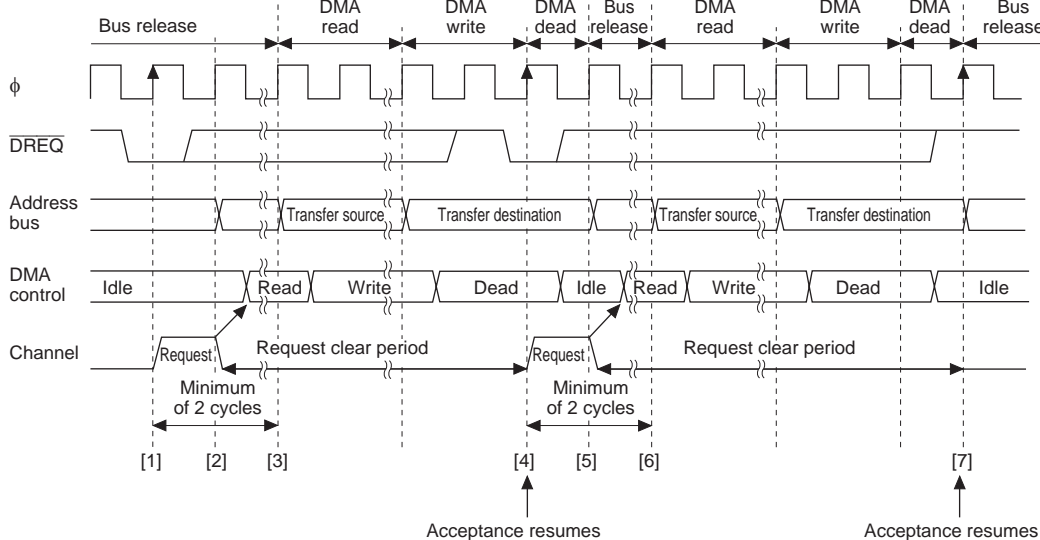
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

**Figure 7.22 Example of  $\overline{\text{DREQ}}$  Pin Falling Edge Activated Normal Mode Transfer**

$\overline{\text{DREQ}}$  pin sampling is performed every cycle, with the rising edge of the next  $\phi$  cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the  $\overline{\text{DREQ}}$  pin low level is sampled while acceptance by means of the  $\overline{\text{DREQ}}$  pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and  $\overline{\text{DREQ}}$  pin high level sampling for edge detection is started. If  $\overline{\text{DREQ}}$  pin high level sampling has been completed by the time the DMA write cycle ends, acceptance resumes after the end of the write cycle,  $\overline{\text{DREQ}}$  pin low level sampling is performed again, and this operation is repeated until the transfer ends.

Figure 7.23 shows an example of block transfer mode transfer activated by the  $\overline{\text{DREQ}}$  pin falling edge.



- [1] Acceptance after transfer enabling; the  $\overline{\text{DREQ}}$  pin low level is sampled on the rising edge of  $\phi$ , and the request is held.
- [2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
- [3] [6] Start of DMA cycle;  $\overline{\text{DREQ}}$  pin high level sampling on the rising edge of  $\phi$  starts.
- [4] [7] When the  $\overline{\text{DREQ}}$  pin high level has been sampled, acceptance is resumed after the dead cycle is completed.  
(As in [1], the  $\overline{\text{DREQ}}$  pin low level is sampled on the rising edge of  $\phi$ , and the request is held.)

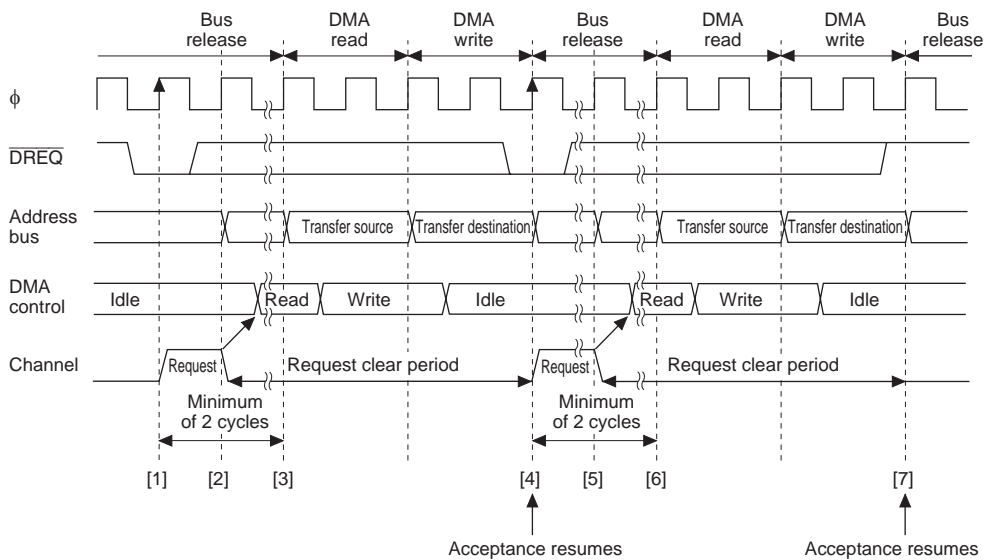
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

**Figure 7.23 Example of  $\overline{\text{DREQ}}$  Pin Falling Edge Activated Block Transfer Mode Transfer**

$\overline{\text{DREQ}}$  pin sampling is performed every cycle, with the rising edge of the next  $\phi$  cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the  $\overline{\text{DREQ}}$  pin low level is sampled while acceptance by means of the  $\overline{\text{DREQ}}$  pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and  $\overline{\text{DREQ}}$  pin high level sampling for edge detection is started. If  $\overline{\text{DREQ}}$  pin high level sampling has been completed by the time the DMA dead cycle ends, acceptance resumes after the end of the dead cycle,  $\overline{\text{DREQ}}$  pin low level sampling is performed again, and this operation is repeated until the transfer ends.

**$\overline{\text{DREQ}}$  Pin Low Level Activation Timing (Normal Mode):** Set the DTA bit in DMABCRH to 1 for the channel for which the  $\overline{\text{DREQ}}$  pin is selected.



- [1] Acceptance after transfer enabling; the  $\overline{\text{DREQ}}$  pin low level is sampled on the rising edge of  $\phi$ , and the request is held.
- [2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
- [3] [6] The DMA cycle is started.
- [4] [7] Acceptance is resumed after the write cycle is completed.  
(As in [1], the  $\overline{\text{DREQ}}$  pin low level is sampled on the rising edge of  $\phi$ , and the request is held.)

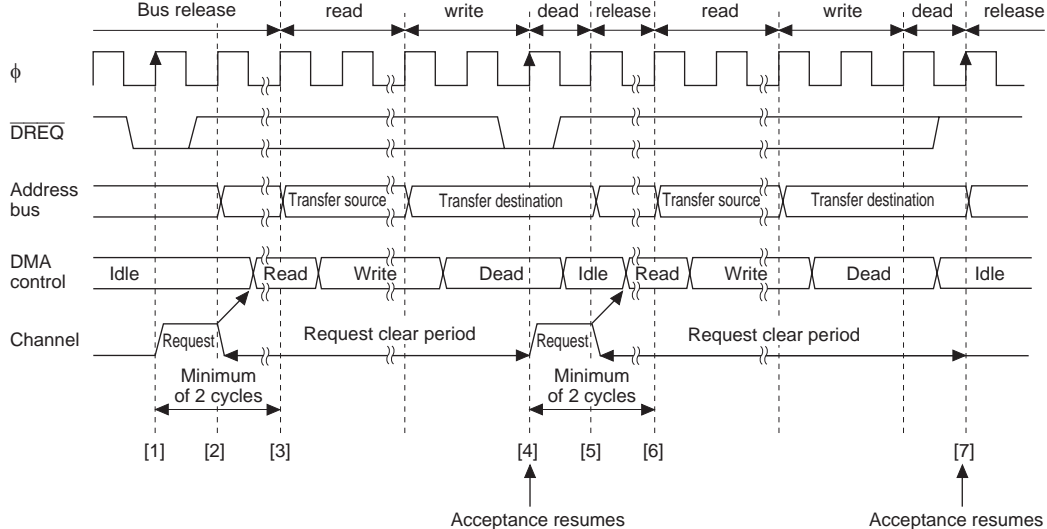
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

**Figure 7.24 Example of  $\overline{\text{DREQ}}$  Pin Low Level Activated Normal Mode Transfer**

$\overline{\text{DREQ}}$  pin sampling is performed every cycle, with the rising edge of the next  $\phi$  cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the  $\overline{\text{DREQ}}$  pin low level is sampled while acceptance by means of the  $\overline{\text{DREQ}}$  pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the write cycle, acceptance resumes,  $\overline{\text{DREQ}}$  pin low level sampling is performed again, and this operation is repeated until the transfer ends.

Figure 7.25 shows an example of block transfer mode transfer activated by  $\overline{\text{DREQ}}$  pin low level.



- [1] Acceptance after transfer enabling; the  $\overline{\text{DREQ}}$  pin low level is sampled on the rising edge of  $\phi$ , and the request is held.
- [2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
- [3] [6] The DMA cycle is started.
- [4] [7] Acceptance is resumed after the dead cycle is completed.  
(As in [1], the  $\overline{\text{DREQ}}$  pin low level is sampled on the rising edge of  $\phi$ , and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

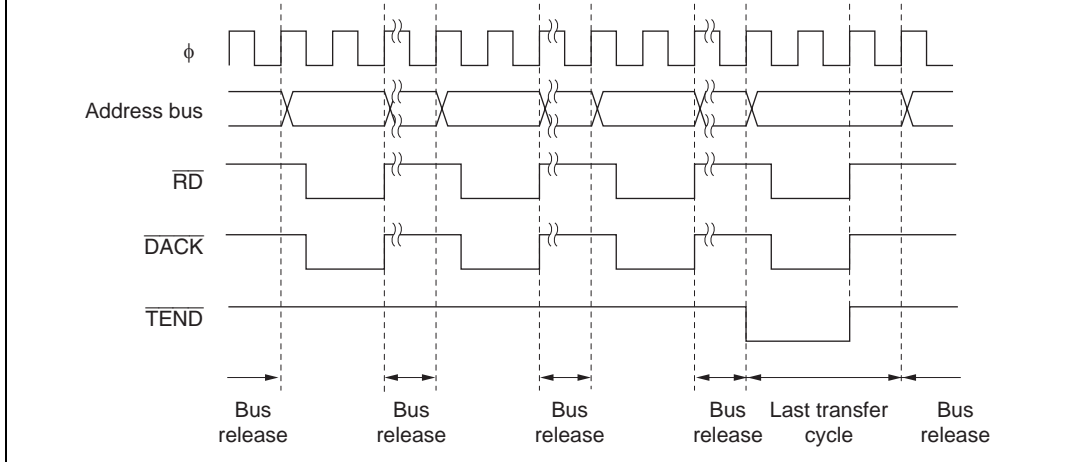
**Figure 7.25 Example of  $\overline{\text{DREQ}}$  Pin Low Level Activated Block Transfer Mode Transfer**

$\overline{\text{DREQ}}$  pin sampling is performed every cycle, with the rising edge of the next  $\phi$  cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the  $\overline{\text{DREQ}}$  pin low level is sampled while acceptance by means of the  $\overline{\text{DREQ}}$  pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared. After the end of the dead cycle, acceptance resumes,  $\overline{\text{DREQ}}$  pin low level sampling is performed again, and this operation is repeated until the transfer ends.

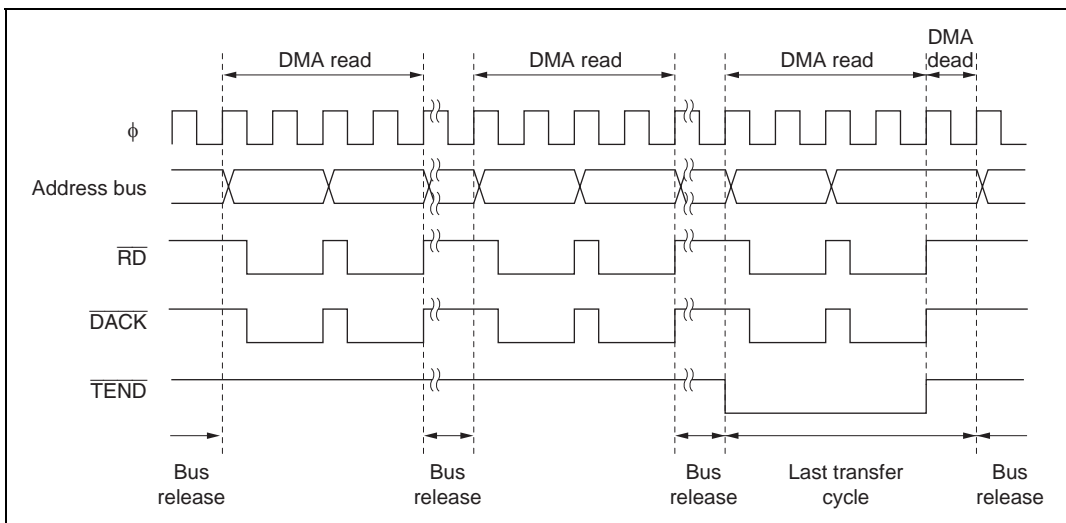
### 7.5.10 DMA Transfer (Single Address Mode) Bus Cycles

**Single Address Mode (Read):** Figure 7.26 shows a transfer example in which  $\overline{\text{TEND}}$  output is enabled and byte-size single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.



**Figure 7.26 Example of Single Address Mode Transfer (Byte Read)**

Figure 7.27 shows a transfer example in which  $\overline{\text{TEND}}$  output is enabled and word-size single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

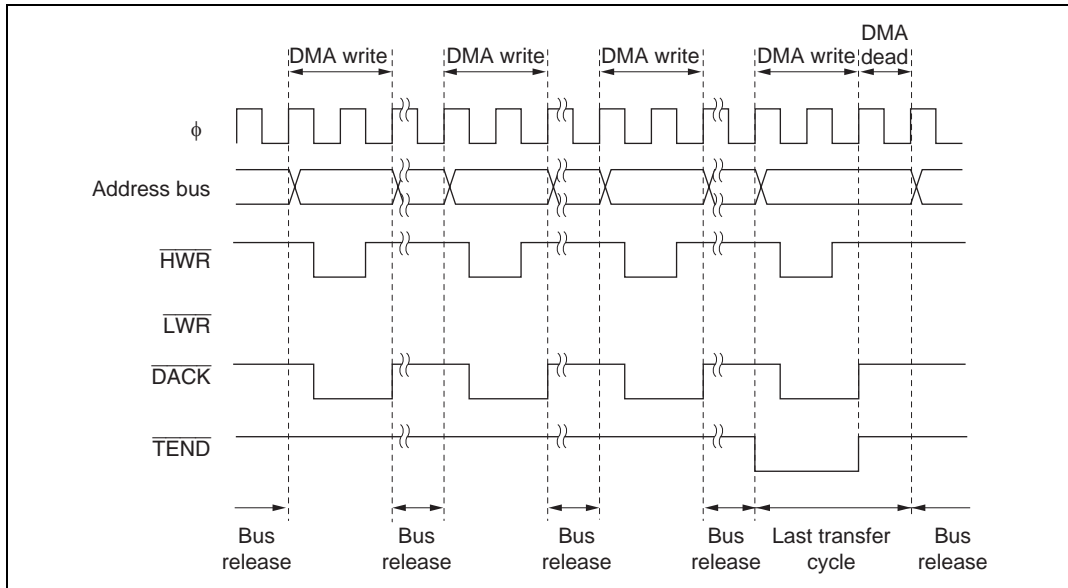


**Figure 7.27 Example of Single Address Mode (Word Read) Transfer**

A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

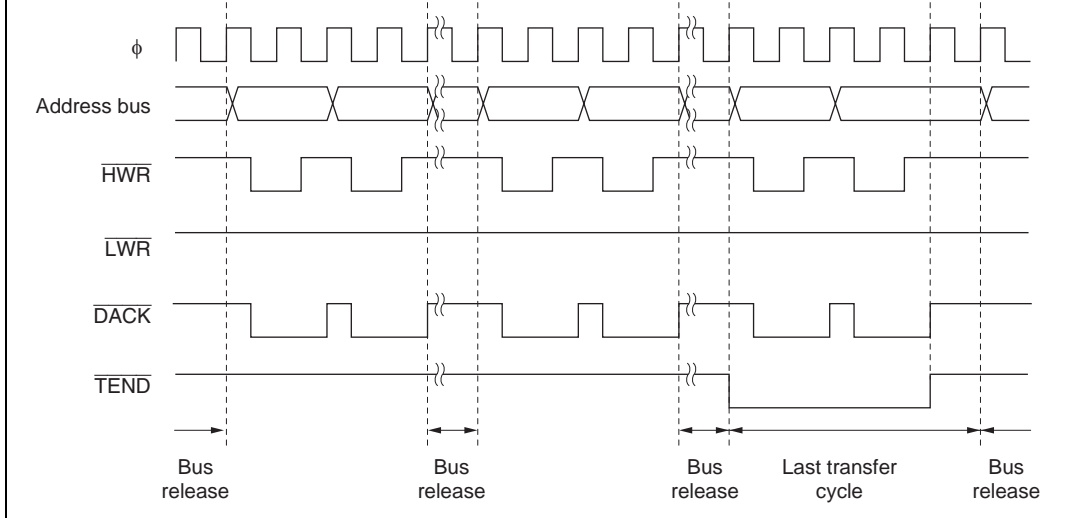


**Single Address Mode (Write):** Figure 7.28 shows a transfer example in which  $\overline{\text{TEND}}$  output is enabled and byte-size single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.



**Figure 7.28 Example of Single Address Mode Transfer (Byte Write)**

Figure 7.29 shows a transfer example in which  $\overline{\text{TEND}}$  output is enabled and word-size single address mode transfer (write) is performed from an external device to external 8-bit, 2-state access space.



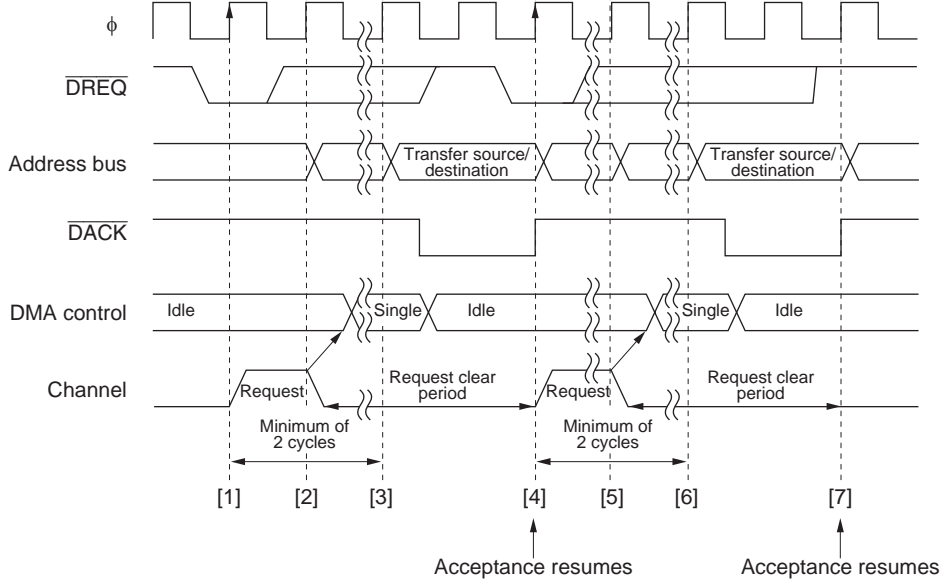
**Figure 7.29 Example of Single Address Mode Transfer (Word Write)**

A byte or word transfer is performed for a single transfer request, and after the transfer, the bus is released. While the bus is released, one or more bus cycles are executed by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

**DREQ Pin Falling Edge Activation Timing:** Set the DTA bit in DMABCRH to 1 for the channel for which the DREQ pin is selected.

Figure 7.30 shows an example of single address mode transfer activated by the DREQ pin falling edge.



- [1] Acceptance after transfer enabling; the  $\overline{\text{DREQ}}$  pin low level is sampled on the rising edge of  $\phi$ , and the request is held.
- [2] [5] The request is cleared at the next bus break, and activation is started in the DMAC.
- [3] [6] Start of DMA cycle;  $\overline{\text{DREQ}}$  pin high level sampling on the rising edge of  $\phi$  starts.
- [4] [7] When the  $\overline{\text{DREQ}}$  pin high level has been sampled, acceptance is resumed after the single cycle is completed. (As in [1], the  $\overline{\text{DREQ}}$  pin low level is sampled on the rising edge of  $\phi$ , and the request is held.)

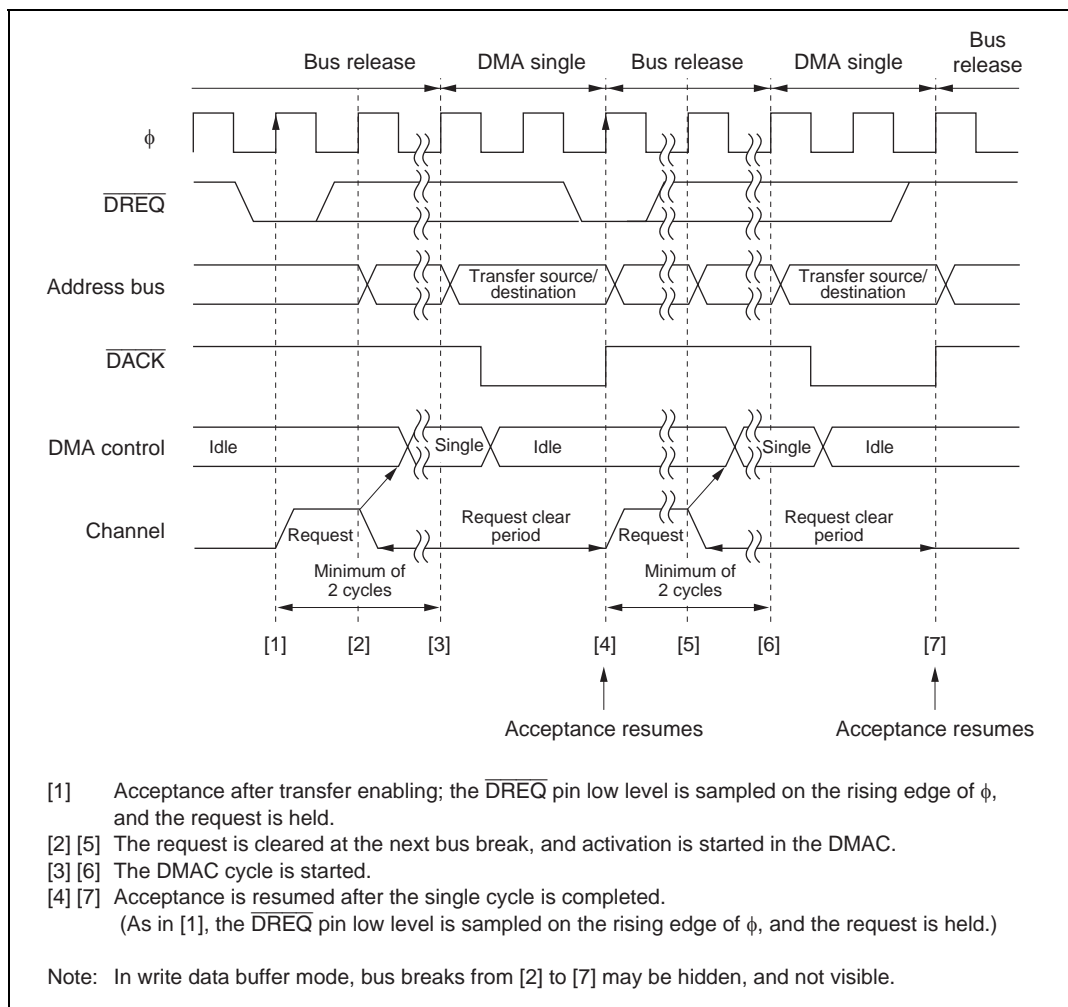
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

**Figure 7.30 Example of  $\overline{\text{DREQ}}$  Pin Falling Edge Activated Single Address Mode Transfer**

$\overline{\text{DREQ}}$  pin sampling is performed every cycle, with the rising edge of the next  $\phi$  cycle after the end of the DMABCR write cycle for setting the transfer enabled state as the starting point.

When the  $\overline{\text{DREQ}}$  pin low level is sampled while acceptance by means of the  $\overline{\text{DREQ}}$  pin is possible, the request is held in the DMAC. Then, when activation is initiated in the DMAC, the request is cleared, and  $\overline{\text{DREQ}}$  pin high level sampling for edge detection is started. If  $\overline{\text{DREQ}}$  pin high level sampling has been completed by the time the DMA single cycle ends, acceptance resumes after the end of the single cycle,  $\overline{\text{DREQ}}$  pin low level sampling is performed again, and this operation is repeated until the transfer ends.

Figure 7.31 shows an example of single address mode transfer activated by the  $\overline{\text{DREQ}}$  pin low level.



**Figure 7.31 Example of  $\overline{\text{DREQ}}$  Pin Low Level Activated Single Address Mode Transfer**

$\overline{\text{DREQ}}$  pin sampling is performed every cycle, with the rising edge of the next  $\phi$  cycle after the end of the DMACR write cycle for setting the transfer enabled state as the starting point.

request is cleared. After the end of the single cycle, acceptance resumes, DREQ pin low level sampling is performed again, and this operation is repeated until the transfer ends.

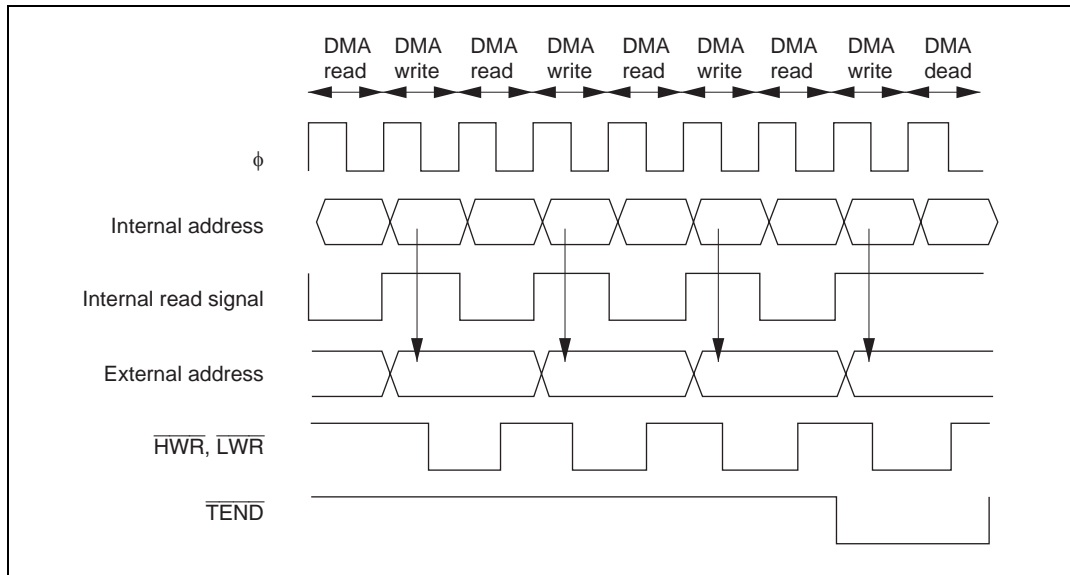
### 7.5.11 Write Data Buffer Function

DMAC internal-to-external dual address transfers and single address transfers can be executed at high speed using the write data buffer function, enabling system throughput to be improved.

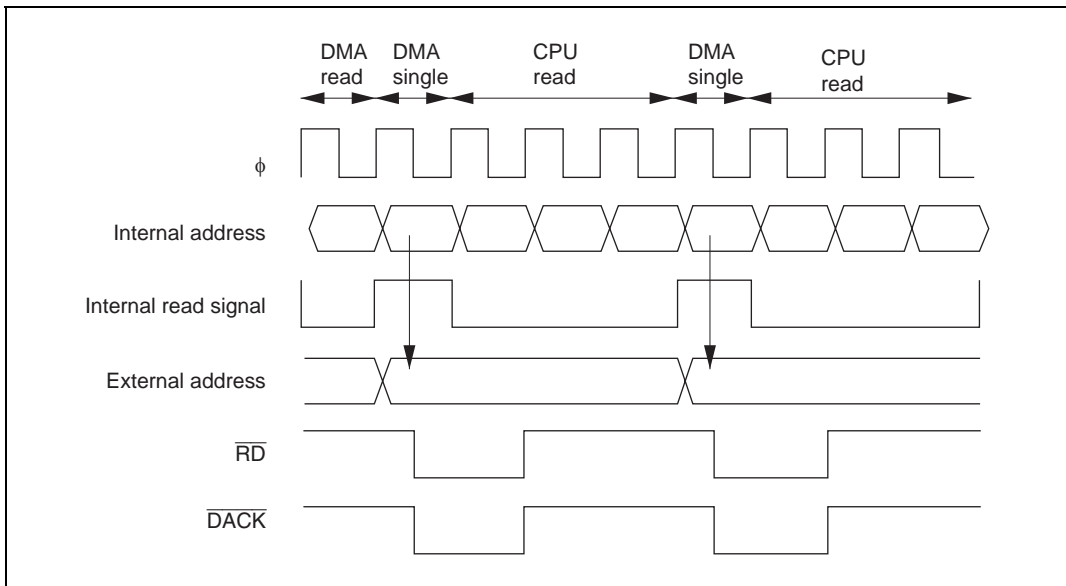
When the WDBE bit of BCR in the bus controller is set to 1, enabling the write data buffer function, external write cycles in dual address transfers or single address transfers are executed in parallel with internal accesses (on-chip memory or internal I/O registers). Internal accesses are independent of the bus master, and DMAC dead cycles are regarded as internal accesses.

A low level can always be output from the  $\overline{\text{TEND}}$  pin if the bus cycle in which a low level is to be output from the  $\overline{\text{TEND}}$  pin is an external bus cycle. However, a low level is not output from the  $\overline{\text{TEND}}$  pin if the bus cycle in which a low level is to be output from the  $\overline{\text{TEND}}$  pin is an internal bus cycle, and an external write cycle is executed in parallel with this cycle.

Figure 7.32 shows an example of dual address transfer using the write data buffer function. In this example, burst mode transfer from on-chip RAM to external memory is performed.



**Figure 7.32 Example of Dual Address Transfer Using Write Data Buffer Function**



**Figure 7.33 Example of Single Address Transfer Using Write Data Buffer Function**

When the write data buffer function is activated, the DMAC recognizes that the bus cycle concerned has ended, and starts the next operation. Therefore,  $\overline{\text{DREQ}}$  pin sampling is started one state after the start of the DMA write cycle or single address transfer.

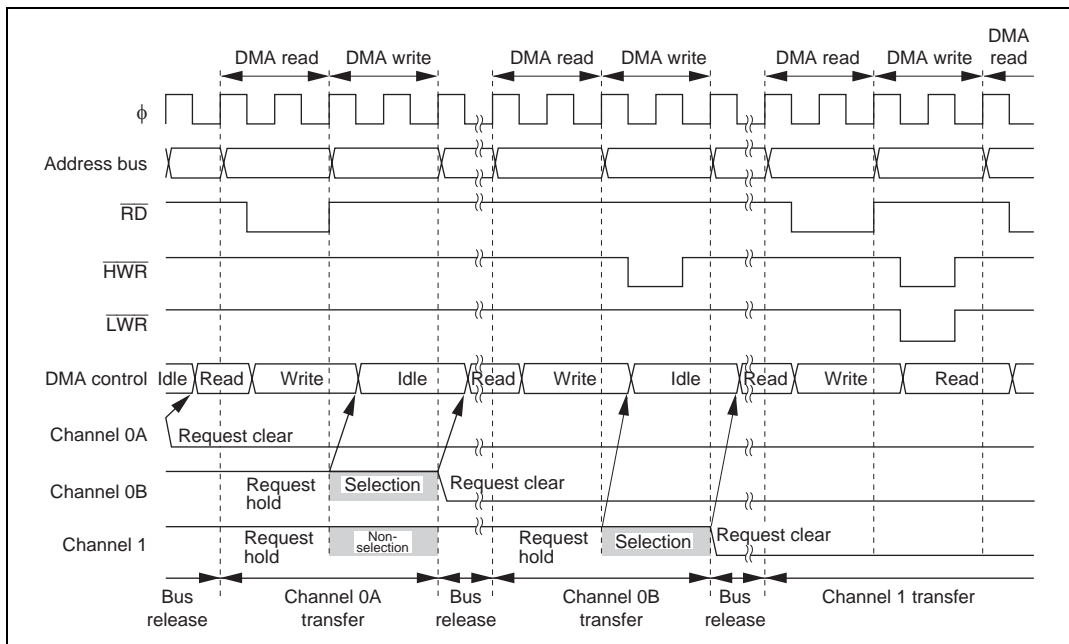
### 7.5.12 Multi-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1, and channel A > channel B. Table 7.11 summarizes the priority order for DMAC channels.

**Table 7.11 DMAC Channel Priority Order**

Short Address Mode	Full Address Mode	Priority
Channel 0A	Channel 0	High
Channel 0B		
Channel 1A	Channel 1	Low
Channel 1B		

highest-priority channel from among those issuing a request according to the priority order shown in table 7.11. During burst transfer, or when one block is being transferred in block transfer, the channel will not be changed until the end of the transfer. Figure 7.34 shows a transfer example in which transfer requests are issued simultaneously for channels 0A, 0B, and 1.



**Figure 7.34 Example of Multi-Channel Transfer**

### 7.5.13 Relation between DMAC and External Bus Requests and Refresh Cycles

When the DMAC accesses external space, conflict with a refresh cycle or external bus release cycle may arise. In this case, the bus controller will suspend the transfer and insert a refresh cycle or external bus release cycle, in accordance with the external bus priority order, even if the DMAC is executing a burst transfer or block transfer. (An external access by the DTC or CPU, which has a lower priority than the DMAC, is not executed until the DMAC releases the external bus.)

When the DMAC transfer mode is dual address mode, the DMAC releases the external bus after an external write cycle. The external read cycle and external write cycle are inseparable, and so the bus cannot be released between these two cycles.

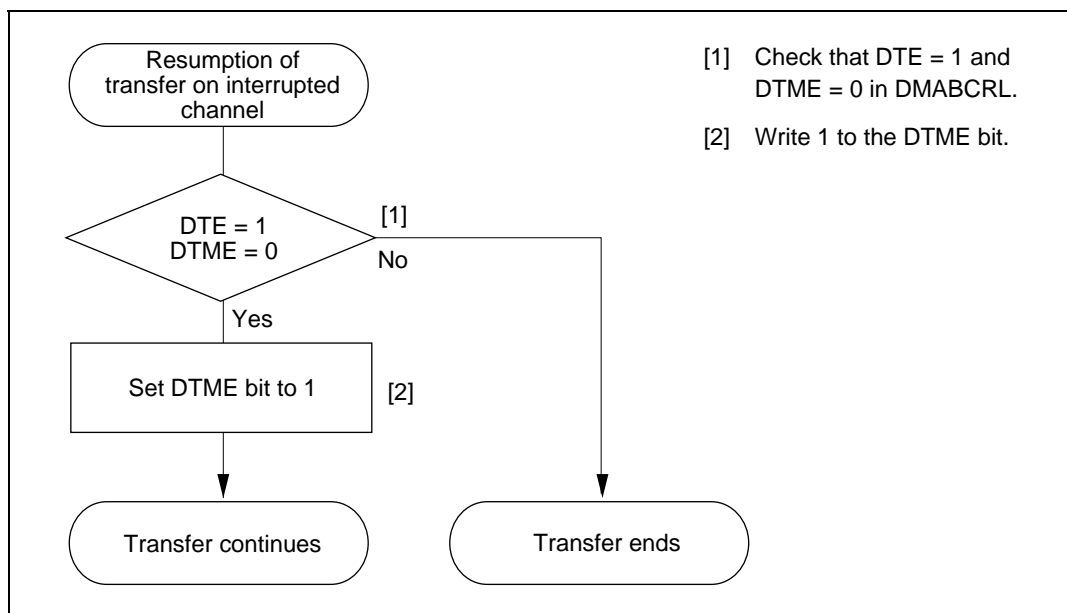
## 7.5.14 DMAC and NMI Interrupts

When an NMI interrupt is requested, burst mode transfer in full address mode is interrupted. An NMI interrupt does not affect the operation of the DMAC in other modes.

In full address mode, transfer is enabled for a channel when both the DTE bit and DTME bit are set to 1. With burst mode setting, the DTME bit is cleared when an NMI interrupt is requested.

If the DTME bit is cleared during burst mode transfer, the DMAC discontinues transfer on completion of the 1-byte or 1-word transfer in progress, then releases the bus, which passes to the CPU.

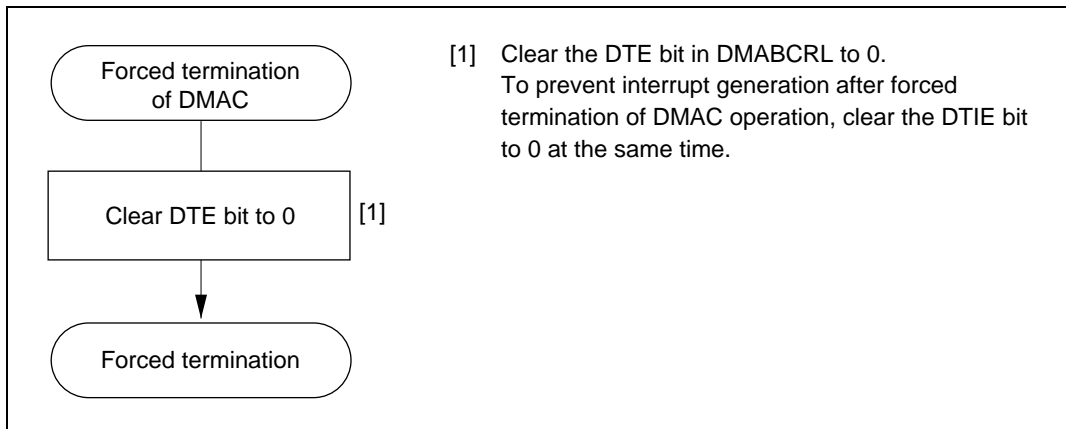
The channel on which transfer was interrupted can be restarted by setting the DTME bit to 1 again. Figure 7.35 shows the procedure for continuing transfer when it has been interrupted by an NMI interrupt on a channel designated for burst mode transfer.



**Figure 7.35 Example of Procedure for Continuing Transfer on Channel Interrupted by NMI Interrupt**

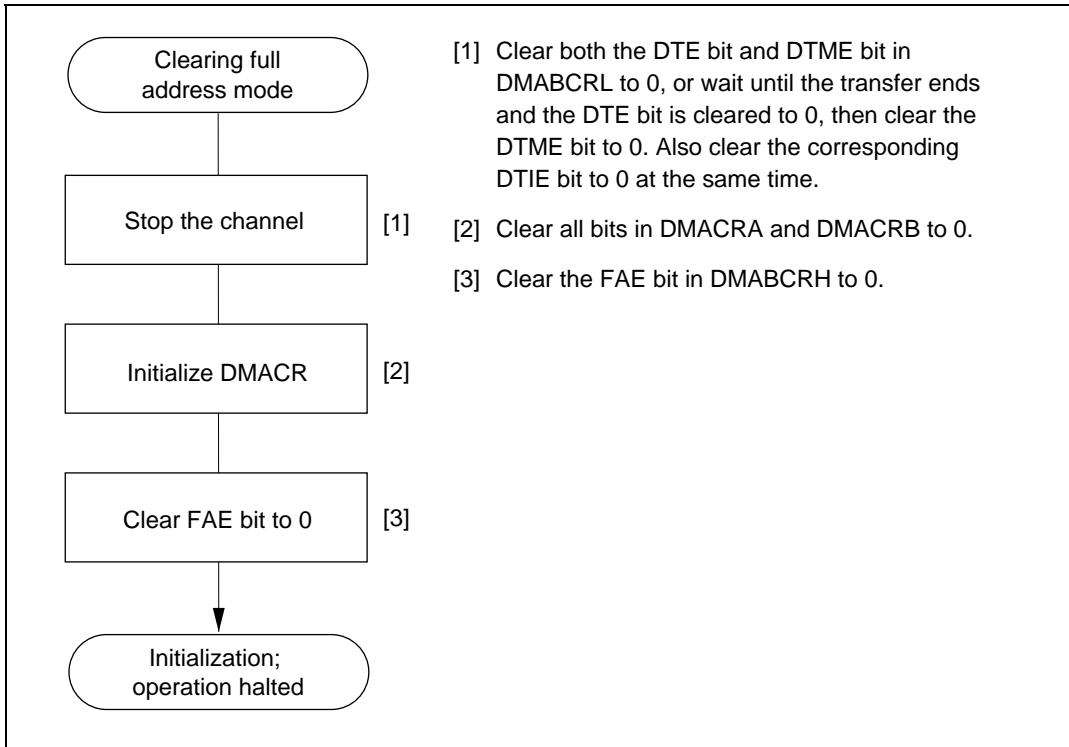


If the DTE bit in DMABCRL is cleared to 0 for the channel currently operating, the DMAC stops on completion of the 1-byte or 1-word transfer in progress. DMAC operation resumes when the DTE bit is set to 1 again. In full address mode, the same applies to the DTME bit in DMABCRL. Figure 7.36 shows the procedure for forcibly terminating DMAC operation by software.



**Figure 7.36 Example of Procedure for Forcibly Terminating DMAC Operation**

Figure 7.37 shows the procedure for releasing and initializing a channel designated for full address mode. After full address mode has been cleared, the channel can be set to another transfer mode using the appropriate setting procedure.



**Figure 7.37 Example of Procedure for Clearing Full Address Mode**

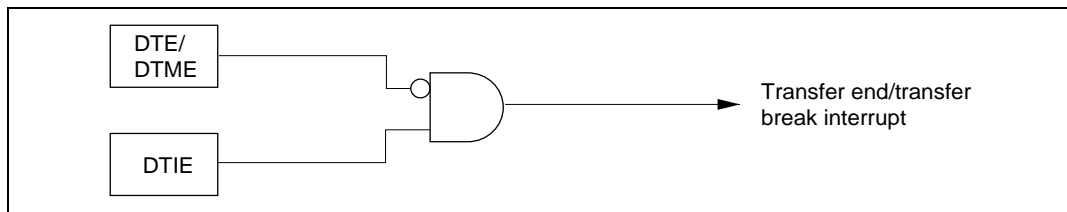
The sources of interrupts generated by the DMAC are transfer end and transfer break. Table 7.12 shows the interrupt sources and their priority order.

**Table 7.12 Interrupt Sources and Priority Order**

Interrupt Name	Interrupt Source		Interrupt Priority Order
	Short Address Mode	Full Address Mode	
DMTEND0A	Interrupt due to end of transfer on channel 0A	Interrupt due to end of transfer on channel 0	<div style="text-align: center;">           High            ↑                     Low         </div>
DMTEND0B	Interrupt due to end of transfer on channel 0B	Interrupt due to break in transfer on channel 0	
DMTEND1A	Interrupt due to end of transfer on channel 1A	Interrupt due to end of transfer on channel 1	
DMTEND1B	Interrupt due to end of transfer on channel 1B	Interrupt due to break in transfer on channel 1	

Enabling or disabling of each interrupt source is set by means of the DTIE bit in DMABCRL for the corresponding channel in DMABCRL, and interrupts from each source are sent to the interrupt controller independently. The priority of transfer end interrupts on each channel is decided by the interrupt controller, as shown in table 7.12.

Figure 7.38 shows a block diagram of a transfer end/transfer break interrupt. An interrupt is always generated when the DTIE bit is set to 1 while the DTE bit in DMABCRL is cleared to 0.



**Figure 7.38 Block Diagram of Transfer End/Transfer Break Interrupt**

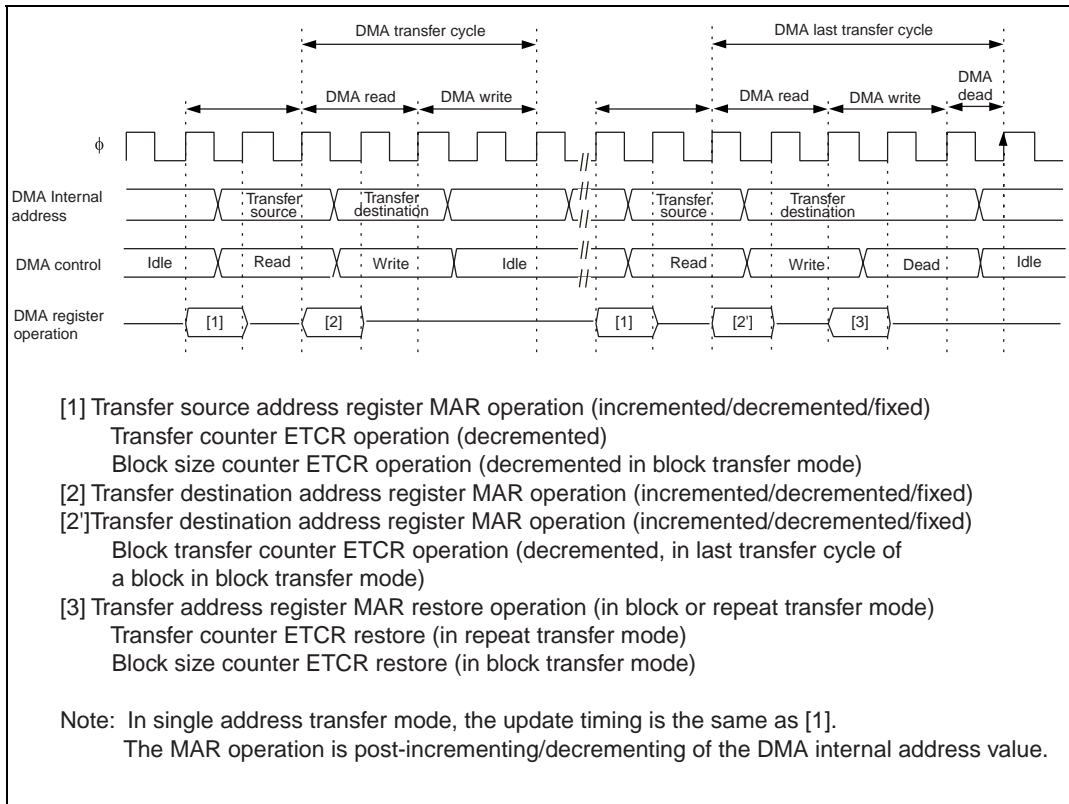
In full address mode, a transfer break interrupt is generated when the DTME bit is cleared to 0 while the DTIEB bit is set to 1. In both short address mode and full address mode, DMABCRL should be set so as to prevent the occurrence of a combination that constitutes a condition for interrupt generation during setting.

## 7.7.1 DMAC Register Access during Operation

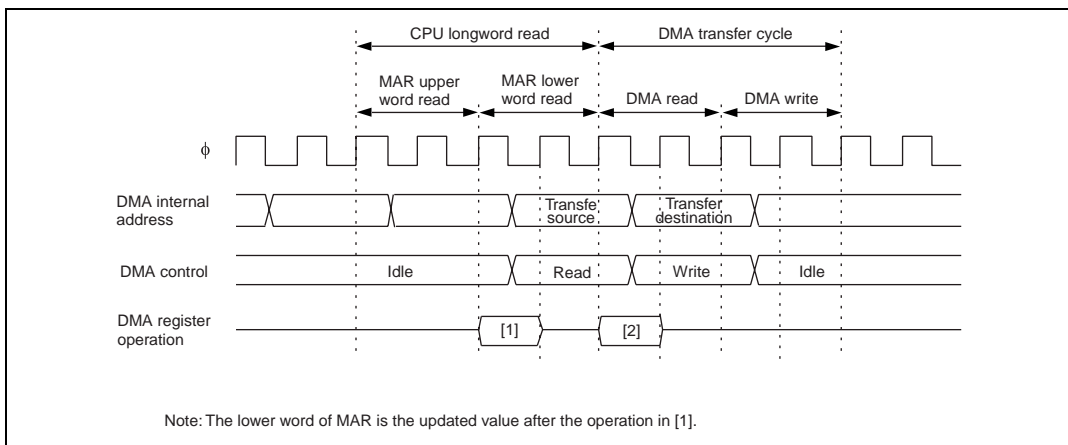
Except for forced termination of the DMAC, the operating (including transfer waiting state) channel setting should not be changed. The operating channel setting should only be changed when transfer is disabled. Also, DMAC registers should not be written to in a DMA transfer.

DMAC register reads during operation (including the transfer waiting state) are described below.

- DMAC control starts one cycle before the bus cycle, with output of the internal address. Consequently, MAR is updated in the bus cycle before DMA transfer. Figure 7.39 shows an example of the update timing for DMAC registers in dual address transfer mode.



**Figure 7.39 DMAC Register Update Timing**



**Figure 7.40 Contention between DMAC Register Update and CPU Read**

### 7.7.2 Module Stop

When the MSTP13 bit in MSTPCRH is set to 1, the DMAC clock stops, and the module stop state is entered. However, 1 cannot be written to the MSTP13 bit if any of the DMAC channels is enabled. This setting should therefore be made when DMAC operation is stopped.

When the DMAC clock stops, DMAC register accesses can no longer be made. Since the following DMAC register settings are valid even in the module stop state, they should be invalidated, if necessary, before a module stop.

- Transfer end/break interrupt (DTE = 0 and DTIE = 1)
- $\overline{\text{TEND}}$  pin enable (TEE = 1)
- $\overline{\text{DACK}}$  pin enable (FAE = 0 and SAE = 1)

### 7.7.3 Write Data Buffer Function

When the WDBE bit of BCR in the bus controller is set to 1, enabling the write data buffer function, external write cycles in dual address transfers or single address transfers are executed in parallel with internal accesses (on-chip memory or internal I/O registers).

external access by means of the write data buffer function, the external access may not be performed normally. Registers that control external accesses should only be manipulated when external reads, etc., are used with DMAC operation disabled, and the operation is not performed in parallel with external access.

- Write data buffer function and DMAC operation timing

The DMAC can start its next operation during external access using the write data buffer function. Consequently, the  $\overline{\text{DREQ}}$  pin sampling timing,  $\overline{\text{TEND}}$  output timing, etc., are different from the case in which the write data buffer function is disabled. Also, internal bus cycles may be hidden, and not visible.

#### 7.7.4 $\overline{\text{TEND}}$ Output

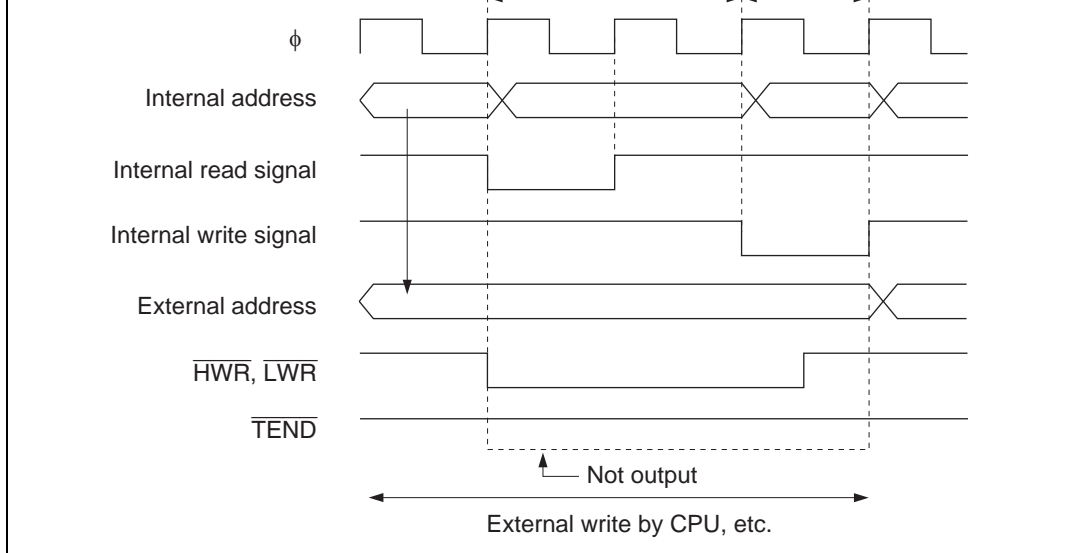
If the last transfer cycle is for an internal address, note that even if low-level output at the  $\overline{\text{TEND}}$  pin has been set, a low level may not be output at the  $\overline{\text{TEND}}$  pin under the following external bus conditions since the last transfer cycle (internal bus cycle) and the external bus cycle are executed in parallel.

1. Write cycle with write buffer mode enabled
2. DMAC single address cycle for a different channel with write buffer mode enabled
3. Bus release cycle
4. CBR refresh cycle

Figure 7.41 shows an example in which a low level is not output from the  $\overline{\text{TEND}}$  pin in case 2 above.

If the last transfer cycle is an external address cycle, a low level is output at the  $\overline{\text{TEND}}$  pin in synchronization with the bus cycle.

However, if the last transfer cycle and a CBR refresh occur simultaneously, note that although the CBR refresh and the last transfer cycle may be executed consecutively,  $\overline{\text{TEND}}$  may also go low in this case for the refresh cycle.



**Figure 7.41 Example in which Low Level is Not Output at  $\overline{\text{TEND}}$  Pin**

### 7.7.5 Activation by Falling Edge on $\overline{\text{DREQ}}$ Pin

$\overline{\text{DREQ}}$  pin falling edge detection is performed in synchronization with DMAC internal operations. The operation is as follows:

- [1] Activation request wait state: Waits for detection of a low level on the  $\overline{\text{DREQ}}$  pin, and switches to [2].
- [2] Transfer wait state: Waits for DMAC data transfer to become possible, and switches to [3].
- [3] Activation request disabled state: Waits for detection of a high level on the  $\overline{\text{DREQ}}$  pin, and switches to [1].

After DMAC transfer is enabled, a transition is made to [1]. Thus, initial activation after transfer is enabled is performed on detection of a low level.

### 7.7.6 Activation Source Acceptance

At the start of activation source acceptance, a low level is detected in both  $\overline{\text{DREQ}}$  pin falling edge sensing and low level sensing. Similarly, in the case of an internal interrupt, the interrupt request is detected. Therefore, a request is accepted from an internal interrupt or  $\overline{\text{DREQ}}$  pin low level that occurs before write to DMABCRL to enable transfer.

### **7.7.7 Internal Interrupt after End of Transfer**

When the DTE bit in DMABCRL is cleared to 0 at the end of a transfer or by a forcible termination, the selected internal interrupt request will be sent to the CPU or DTC even if the DTA bit in DMABCRH is set to 1.

Also, if internal DMAC activation has already been initiated when operation is forcibly terminated, the transfer is executed but flag clearing is not performed for the selected internal interrupt even if the DTA bit is set to 1.

An internal interrupt request following the end of transfer or a forcible termination should be handled by the CPU as necessary.

### **7.7.8 Channel Re-Setting**

To reactivate a number of channels when multiple channels are enabled, use exclusive handling of transfer end interrupts, and perform DMABCR control bit operations exclusively.

Note, in particular, that in cases where multiple interrupts are generated between reading and writing of DMABCR, and a DMABCR operation is performed during new interrupt handling, the DMABCR write data in the original interrupt handling routine will be incorrect, and the write may invalidate the results of the operations by the multiple interrupts. Ensure that overlapping DMABCR operations are not performed by multiple interrupts, and that there is no separation between read and write operations by the use of a bit-manipulation instruction.

Also, when the DTE and DTME bits are cleared by the DMAC or are written with 0, they must first be read while cleared to 0 before the CPU can write 1 to them.



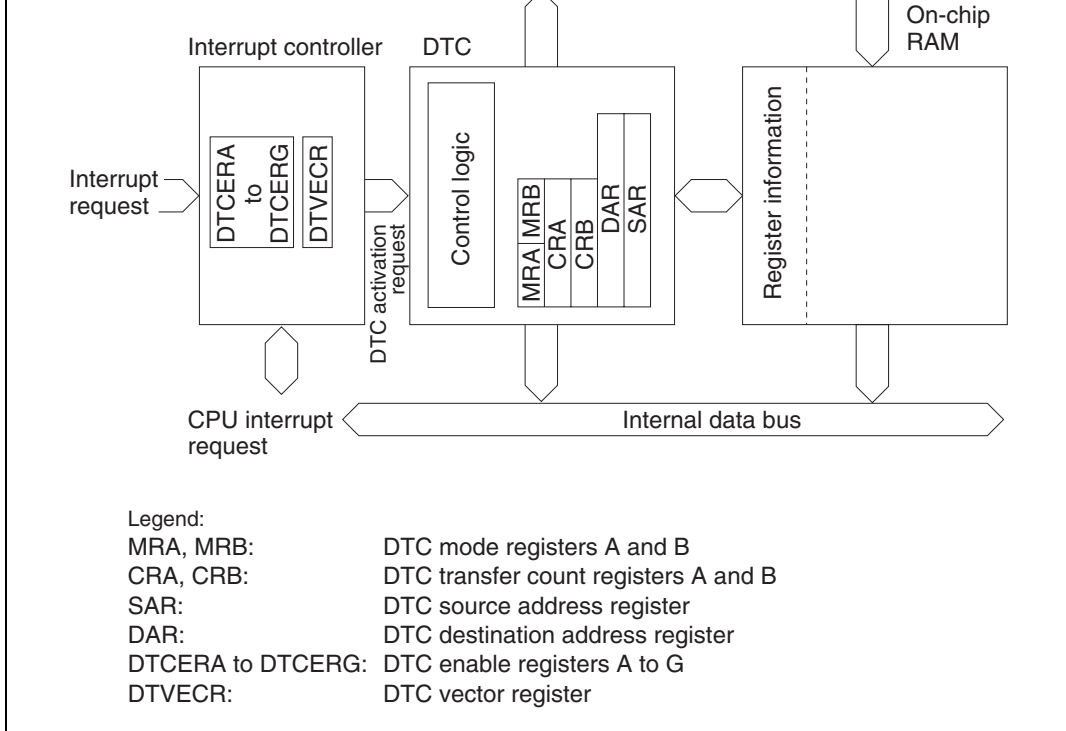
This LSI includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

Figure 8.1 shows a block diagram of the DTC.

## 8.1 Features

- Transfer possible over any number of channels
- Three transfer modes
  - Normal mode
    - One operation transfers one byte or one word of data.
    - Memory address is incremented or decremented by 1 or 2.
    - From 1 to 65,536 transfers can be specified.
  - Repeat mode
    - One operation transfers one byte or one word of data.
    - Memory address is incremented or decremented by 1 or 2.
    - Once the specified number of transfers (1 to 256) has ended, the initial state is restored, and transfer is repeated.
  - Block transfer mode
    - One operation transfers one block of data.
    - The block size is 1 to 256 bytes or words.
    - From 1 to 65,536 transfers can be specified.
    - Either the transfer source or the transfer destination is designated as a block area.
- One activation source can trigger a number of data transfers (chain transfer)
- Direct specification of 16-Mbyte address space possible
- Activation by software is possible
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
- Module stop mode can be set

The DTC's register information is stored in the on-chip RAM. When the DTC is used, the RAME bit in SYSCR must be set to 1. A 32-bit bus connects the DTC to the on-chip RAM (1 kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.



**Figure 8.1 Block Diagram of DTC**

DTC has the following registers.

- DTC mode register A (MRA)
- DTC mode register B (MRB)
- DTC source address register (SAR)
- DTC destination address register (DAR)
- DTC transfer count register A (CRA)
- DTC transfer count register B (CRB)

These six registers cannot be directly accessed from the CPU. When activated, the DTC reads a set of register information that is stored in an on-chip RAM to the corresponding DTC registers and transfers data. After the data transfer, it writes a set of updated register information back to the RAM.

- DTC enable registers A to G (DTCERA to DTCERG)
- DTC vector register (DTVECR)

### 8.2.1 DTC Mode Register A (MRA)

MRA selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	SM1	Undefined	—	Source Address Mode 1 and 0
6	SM0	Undefined	—	These bits specify an SAR operation after a data transfer. 0x: SAR is fixed 10: SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: SAR is decremented after a transfer (by -1 when Sz = 0; by -2 when Sz = 1)
5	DM1	Undefined	—	Destination Address Mode 1 and 0
4	DM0	Undefined	—	These bits specify a DAR operation after a data transfer. 0x: DAR is fixed 10: DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1) 11: DAR is decremented after a transfer (by -1 when Sz = 0; by -2 when Sz = 1)

2	MD0	Undefined	—	<p>These bits specify the DTC transfer mode.</p> <p>00: Normal mode</p> <p>01: Repeat mode</p> <p>10: Block transfer mode</p> <p>11: Setting prohibited</p>
1	DTS	Undefined	—	<p>DTC Transfer Mode Select</p> <p>Specifies whether the source side or the destination side is set to be a repeat area or block area, in repeat mode or block transfer mode.</p> <p>0: Destination side is repeat area or block area</p> <p>1: Source side is repeat area or block area</p>
0	Sz	Undefined	—	<p>DTC Data Transfer Size</p> <p>Specifies the size of data to be transferred.</p> <p>0: Byte-size transfer</p> <p>1: Word-size transfer</p>

Legend:

x: Don't care

MRB selects the DTC operating mode.

Bit	Bit Name	Initial Value	R/W	Description
7	CHNE	Undefined	—	<b>DTC Chain Transfer Enable</b>  When this bit is set to 1, a chain transfer will be performed. For details, refer to section 8.5.4, Chain Transfer.  In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the activation source flag, and clearing of DTCER is not performed.
6	DISEL	Undefined	—	<b>DTC Interrupt Select</b>  When this bit is set to 1, a CPU interrupt request is generated every time after a data transfer ends. When this bit is set to 0, a CPU interrupt request is generated at the time when the specified number of data transfer ends.
5	CHNS	Undefined	—	<b>DTC Chain Transfer Select</b>  Specifies the chain transfer condition. 0: Chain transfer every time 1: Chain transfer only when transfer counter = 0
4 to 0	—	Undefined	—	<b>Reserved</b>  These bits have no effect on DTC operation, and should always be written with 0.

### 8.2.3 DTC Source Address Register (SAR)

SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

### 8.2.4 DTC Destination Address Register (DAR)

DAR is a 24-bit register that designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

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

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

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

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

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

This register is not used in normal mode or repeat mode.

DTCE7 which is comprised of seven registers, DTCE7A to DTCE7G, is a register that specifies DTC activation interrupt sources. The correspondence between interrupt sources and DTCE bits is shown in table 8.2. For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. If all interrupts are masked, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

Bit	Bit Name	Initial Value	R/W	Description
7	DTCE7	0	R/W	DTC Activation Enable
6	DTCE6	0	R/W	Setting this bit to 1 specifies a relevant interrupt source to a DTC activation source.
5	DTCE5	0	R/W	
4	DTCE4	0	R/W	[Clearing conditions]
3	DTCE3	0	R/W	
2	DTCE2	0	R/W	
1	DTCE1	0	R/W	<ul style="list-style-type: none"> <li>When the DISEL bit is 1 and the data transfer has ended</li> <li>When the specified number of transfers have ended</li> </ul>
0	DTCE0	0	R/W	

These bits are not automatically cleared when the DISEL bit is 0 and the specified number of transfers have not ended

- When 0 is written to the DTCE bit after reading DTCE = 1

DTVECR enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

Bit	Bit Name	Initial Value	R/W	Description
7	SWDTE	0	R/W	<p>DTC Software Activation Enable</p> <p>Setting this bit to 1 activates DTC. Only 1 can be written to this bit.</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When the DISEL bit is 0 and the specified number of transfers have not ended</li> <li>• When 0 is written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU.</li> </ul> <p>When the DISEL bit is 1 and data transfer has ended or when the specified number of transfers have ended, this bit will not be cleared.</p>
6	DTVEC6	0	R/W	DTC Software Activation Vectors 6 to 0
5	DTVEC5	0	R/W	<p>These bits specify a vector number for DTC software activation.</p> <p>The vector address is expressed as H'0400 + (vector number × 2). For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420.</p> <p>When the bit SWDTE is 0, these bits can be written.</p>
4	DTVEC4	0	R/W	
3	DTVEC3	0	R/W	
2	DTVEC2	0	R/W	
1	DTVEC1	0	R/W	
0	DTVEC0	0	R/W	



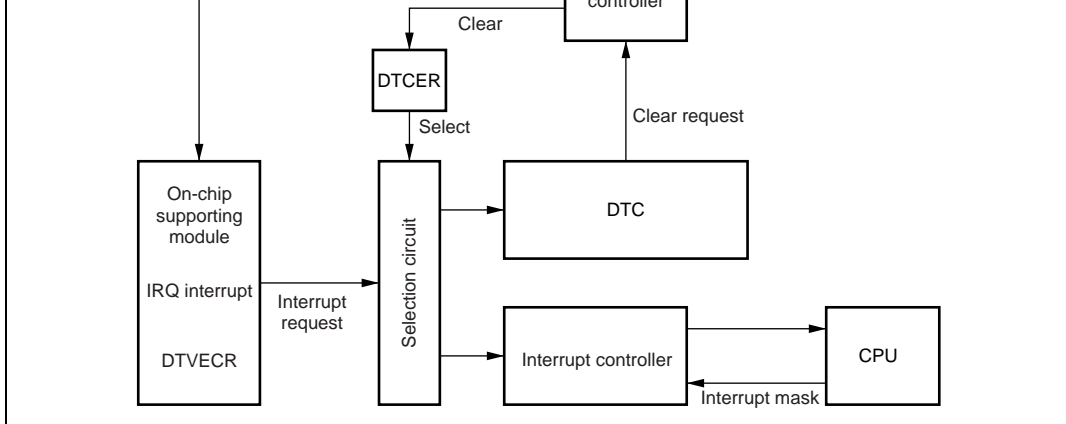
The DTC operates when activated by an interrupt or by a write to DTVECR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source or corresponding DTCER bit is cleared. The activation source flag, in the case of RXI0, for example, is the RDRF flag of SCI\_0.

When an interrupt has been designated a DTC activation source, existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities.

Table 8.1 shows the relationship between the activation sources and DTCER clearing, and figure 8.2 shows a block diagram of activation source control. For details see section 5, Interrupt Controller.

**Table 8.1 Relationship between Activation Sources and DTCER Clearing**

<b>Activation Source</b>	<b>DISEL = 0 and Specified Number of Transfers Has Not Ended</b>	<b>DISEL = 1 or Specified Number of Transfers Has Ended</b>
Activation by software	SWDTE bit is cleared to 0	<ul style="list-style-type: none"> <li>• SWDTE bit remains set to 1</li> <li>• Interrupt request to CPU</li> </ul>
Activation by an interrupt	<ul style="list-style-type: none"> <li>• Corresponding DTCER bit remains set to 1.</li> <li>• Activation source flag is cleared to 0.</li> </ul>	<ul style="list-style-type: none"> <li>• Corresponding DTCER bit is cleared to 0.</li> <li>• Activation source flag remains set to 1.</li> <li>• Interrupt that became the activation source is requested to the CPU.</li> </ul>



**Figure 8.2 Block Diagram of DTC Activation Source Control**

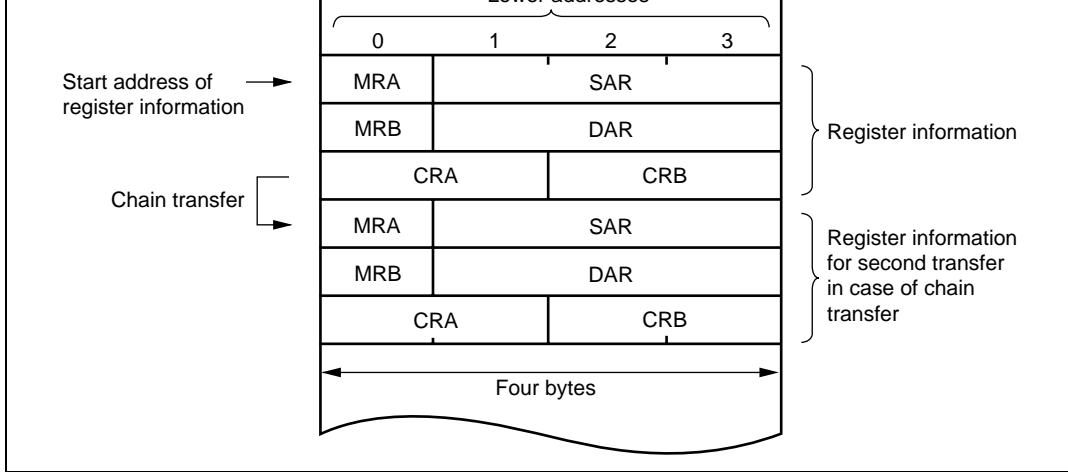
## 8.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM (addresses: H'FFBC00 to H'FFBFFF). Register information should be located at the address that is multiple of four within the range. Locating the register information in address space is shown in figure 8.3. Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information. In the case of chain transfer, register information should be located in consecutive areas as shown in figure 8.3 and the register information start address should be located at the corresponding vector address to the activation source. Figure 8.4 shows the correspondence between the DTC vector address and register information. The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address.

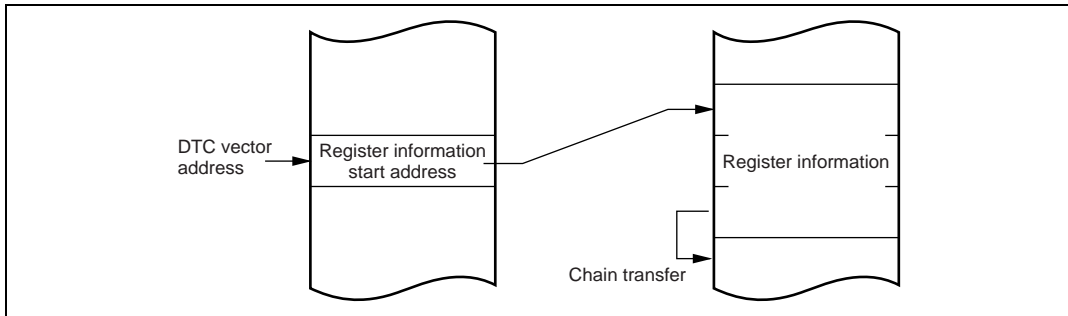
When the DTC is activated by software, the vector address is obtained from:  $H'0400 + (DTVECR[6:0] \times 2)$ . For example, if DTVECR is H'10, the vector address is H'0420.

The configuration of the vector address is the same in both normal\* and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the register information start address.

Note: \* Not available in this LSI.



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



**Figure 8.4 Correspondence between DTC Vector Address and Register Information**

Origin of Activation Source	Activation Source	Vector Number	DTC Vector Address	DTCE*	Priority
Software	Write to DTVECR	DTVECR	H'0400 + (DTVECR — [6:0] × 2)		High
External pin	IRQ0	16	H'0420	DTCEA7	↑
	IRQ1	17	H'0422	DTCEA6	
	IRQ2	18	H'0424	DTCEA5	
	IRQ3	19	H'0426	DTCEA4	
	IRQ4	20	H'0428	DTCEA3	
	IRQ5	21	H'042A	DTCEA2	
	IRQ6	22	H'042C	DTCEA1	
	IRQ7	23	H'042E	DTCEA0	
A/D	ADI	38	H'044C	DTCEC6	
TPU_0	TGI0A	40	H'0450	DTCEC5	
	TGI0B	41	H'0452	DTCEC4	
	TGI0C	42	H'0454	DTCEC3	
	TGI0D	43	H'0456	DTCEC2	
TPU_1	TGI1A	48	H'0460	DTCEC1	
	TGI1B	49	H'0462	DTCEC0	
TPU_2	TGI2A	52	H'0468	DTCED7	
	TGI2B	53	H'046A	DTCED6	
TPU_3	TGI3A	56	H'0470	DTCED5	
	TGI3B	57	H'0472	DTCED4	
	TGI3C	58	H'0474	DTCED3	
	TGI3D	59	H'0476	DTCED2	
TPU_4	TGI4A	64	H'0480	DTCED1	
	TGI4B	65	H'0482	DTCED0	
TPU_5	TGI5A	68	H'0488	DTCEE7	
	TGI5B	69	H'048A	DTCEE6	Low

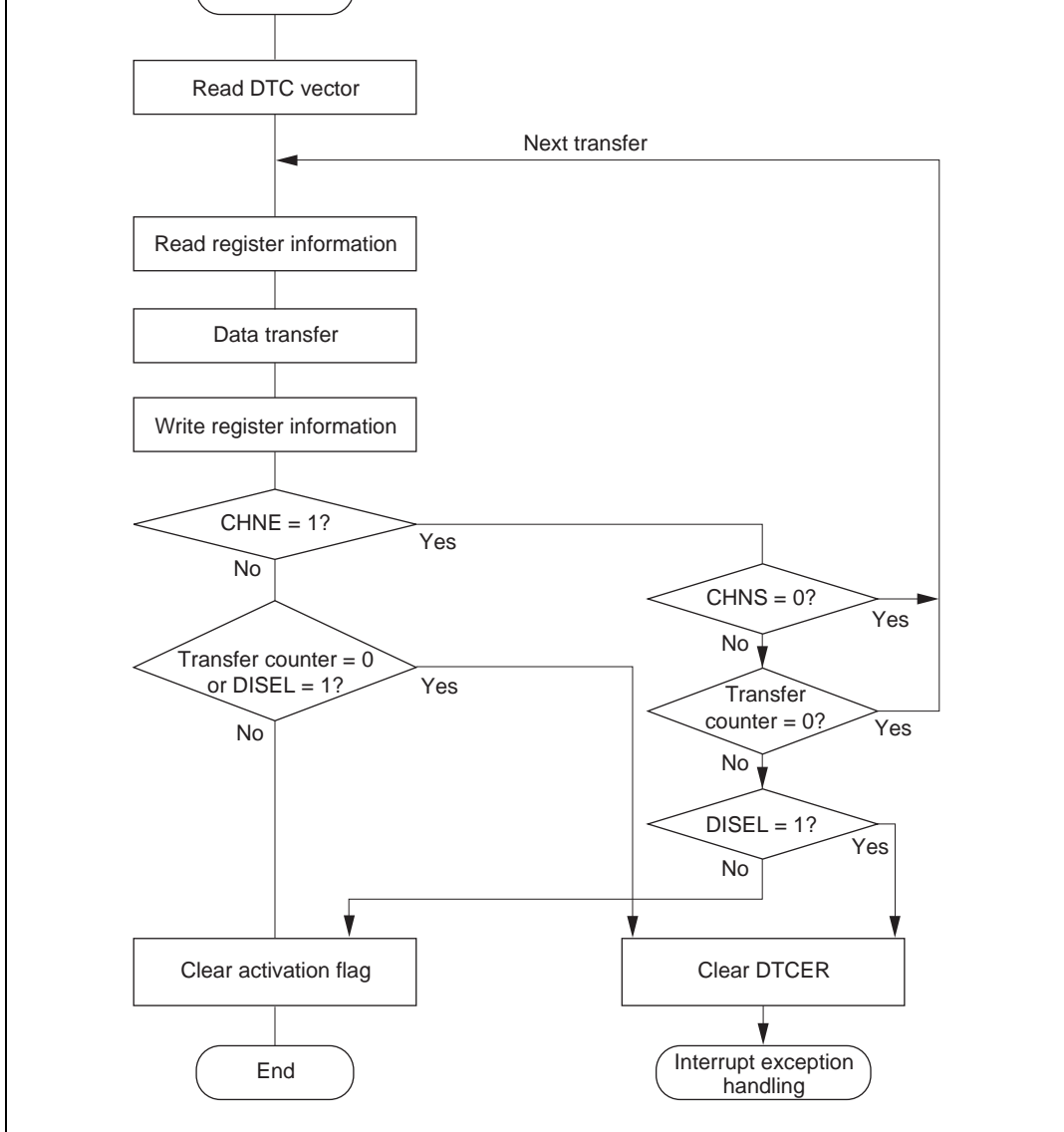
Source	Source	Vector Number	Vector Address	DTCE	Priority
TMR_0	CMIA0	72	H'0490	DTCEE3	High ↑ Low
	CMIB0	73	H'0492	DTCEE2	
TMR_1	CMIA1	76	H'0498	DTCEE1	
	CMIB1	77	H'049A	DTCEE0	
DMAC	DMTEND0A	80	H'04A0	DTCEF7	
	DMTEND0B	81	H'04A2	DTCEF6	
	DMTEND1A	82	H'04A4	DTCEF5	
	DMTEND1B	83	H'04A6	DTCEF4	
SCI_0	RX10	89	H'04B2	DTCEF3	
	TX10	90	H'04B4	DTCEF2	
SCI_1	RX11	93	H'04BA	DTCEF1	
	TX11	94	H'04BC	DTCEF0	
SCI_2	RX12	97	H'04C2	DTCEG7	
	TX12	98	H'04C4	DTCEG6	
SCI_3	RX13	101	H'04CA	DTCEF5	
	TX13	102	H'04CC	DTCEF4	
SCI_4	RX14	105	H'04D2	DTCEG3	
	TX14	106	H'04D4	DTCEG2	Low

Note: \* DTCE bits with no corresponding interrupt are reserved, and should be written with 0. When clearing the software standby state or all-module-clocks-stop mode with an interrupt, write 0 to the corresponding DTCE bit.

The DTC stores register information in the on-chip RAM. When activated, the DTC reads register information that is already stored in the on-chip RAM and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to the on-chip RAM. Pre-storage of register information in the on-chip RAM makes it possible to transfer data over any required number of channels. There are three transfer modes: normal mode, repeat mode, and block transfer mode. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation (chain transfer). A setting can also be made to have chain transfer performed only when the transfer counter value is 0. This enables DTC re-setting to be performed by the DTC itself.

The 24-bit SAR designates the DTC transfer source address and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed.

Figure 8.5 shows a flowchart of DTC operation, and table 8.3 summarizes the chain transfer conditions (combinations for performing the second and third transfers are omitted).



**Figure 8.5 Flowchart of DTC Operation**

1st Transfer				2nd Transfer				DTC Transfer
CHNE	CHNS	DISEL	CR	CHNE	CHNS	DISEL	CR	
0	—	0	Not 0	—	—	—	—	Ends at 1st transfer
0	—	0	0	—	—	—	—	Ends at 1st transfer
0	—	1	—	—	—	—	—	Interrupt request to CPU
1	0	—	—	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0	Ends at 2nd transfer
				0	—	1	—	Interrupt request to CPU
1	1	0	Not 0	—	—	—	—	Ends at 1st transfer
1	1	—	0	0	—	0	Not 0	Ends at 2nd transfer
				0	—	0	0	Ends at 2nd transfer
				0	—	1	—	Interrupt request to CPU
1	1	1	Not 0	—	—	—	—	Ends at 1st transfer Interrupt request to CPU

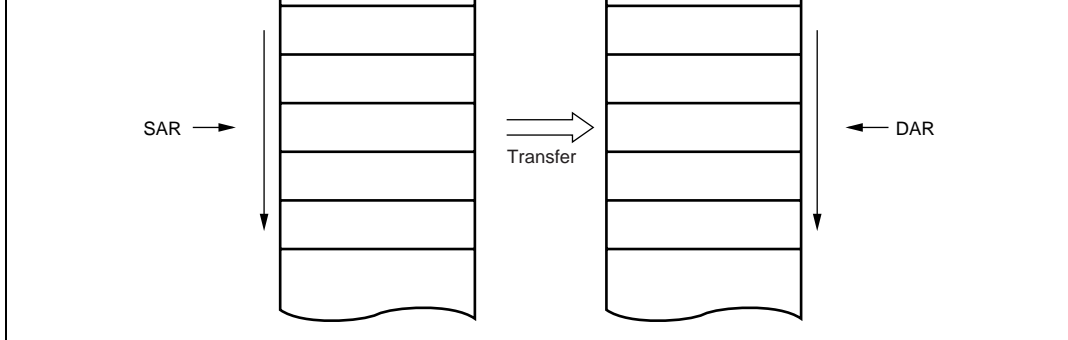
### 8.5.1 Normal Mode

In normal mode, one operation transfers one byte or one word of data. Table 8.4 lists the register function in normal mode. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has ended, a CPU interrupt can be requested.

**Table 8.4 Register Function in Normal Mode**

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register A	CRA	Designates transfer count
DTC transfer count register B	CRB	Not used





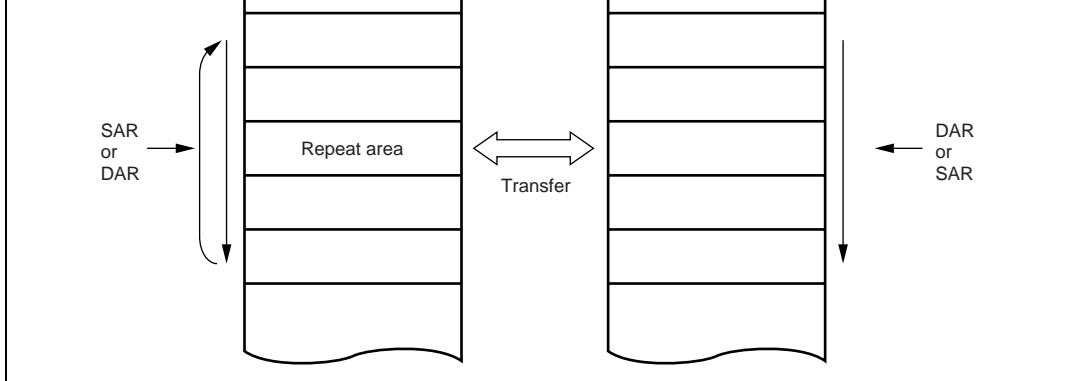
**Figure 8.6 Memory Mapping in Normal Mode**

### 8.5.2 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data. Table 8.5 lists the register function in repeat mode. From 1 to 256 transfers can be specified. Once the specified number of transfers has ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when DISEL = 0.

**Table 8.5 Register Function in Repeat Mode**

<b>Name</b>	<b>Abbreviation</b>	<b>Function</b>
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Designates transfer count
DTC transfer count register B	CRB	Not used



**Figure 8.7 Memory Mapping in Repeat Mode**

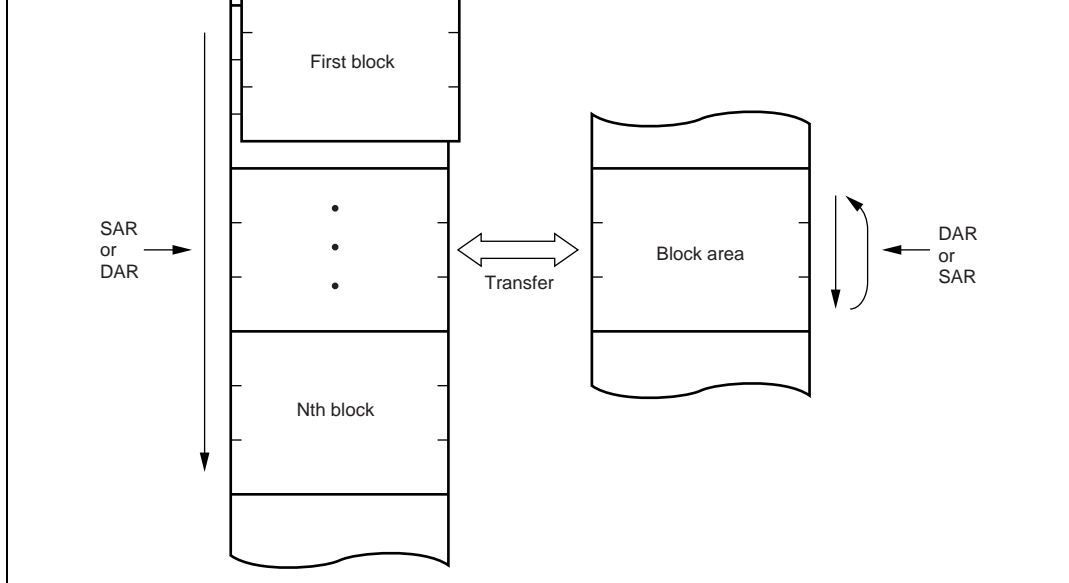
### 8.5.3 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area. Table 8.6 lists the register function in block transfer mode.

The block size is 1 to 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed. From 1 to 65,536 transfers can be specified. Once the specified number of transfers has ended, a CPU interrupt is requested.

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

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Designates block size count
DTC transfer count register B	CRB	Designates transfer count



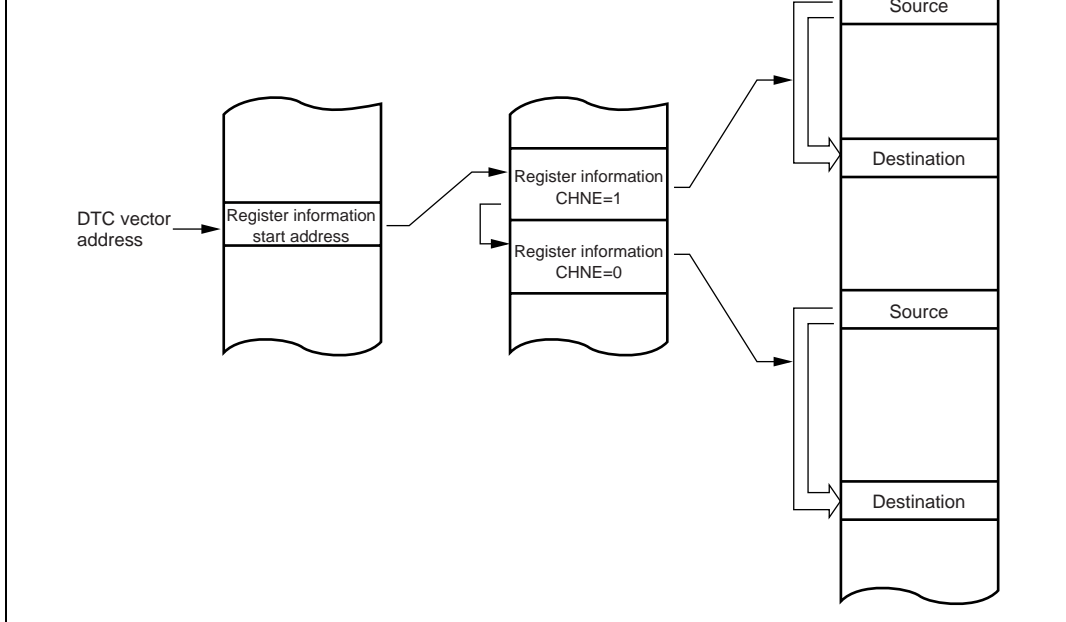
**Figure 8.8 Memory Mapping in Block Transfer Mode**

### 8.5.4 Chain Transfer

Setting the CHNE bit to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 8.9 shows the operation of chain transfer. When activated, the DTC reads the register information start address stored at the vector address, and then reads the first register information at that start address. The CHNE bit in MRB is checked after the end of data transfer, if the value is 1, the next register information, which is located consecutively, is read and transfer is performed. This operation is repeated until the end of data transfer of register information with CHNE = 0. It is also possible, by setting both the CHNE bit and CHNS bit to 1, to specify execution of chain transfer only when the transfer counter value is 0.

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



**Figure 8.9 Operation of Chain Transfer**

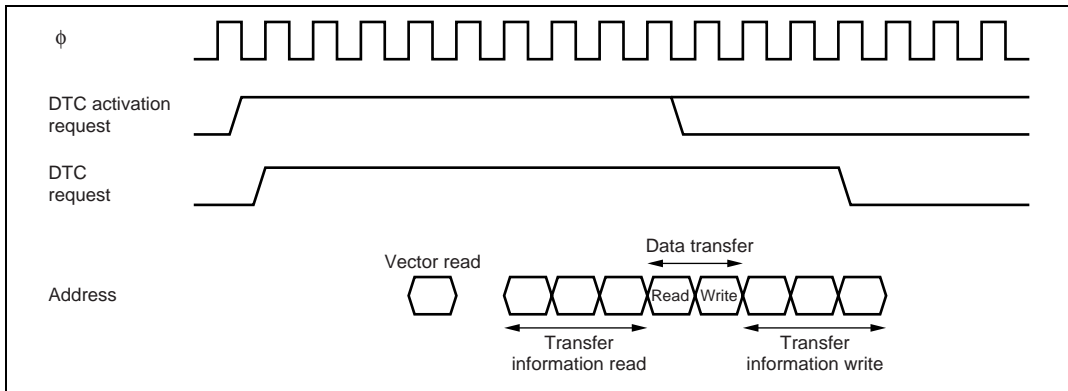
### 8.5.5 Interrupts

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

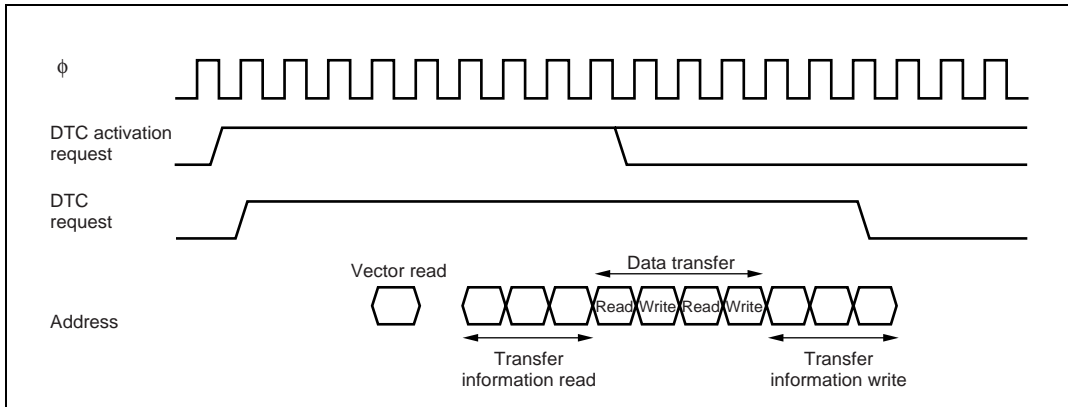
In the case of activation by software, a software activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers has ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine should clear the SWDTE bit to 0.

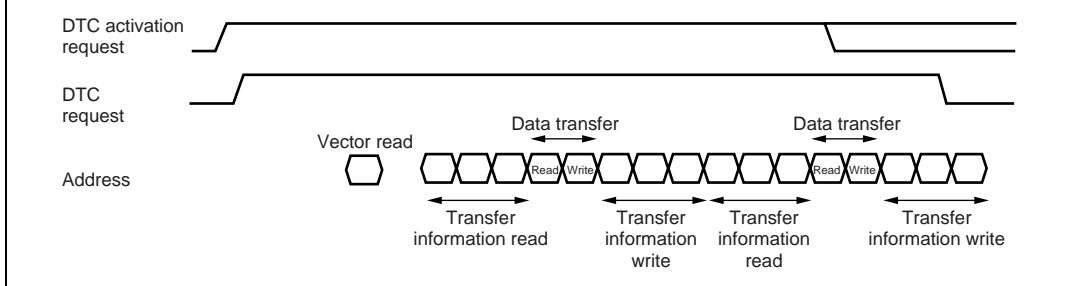
When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.



**Figure 8.10 DTC Operation Timing (Example in Normal Mode or Repeat Mode)**



**Figure 8.11 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)**



**Figure 8.12 DTC Operation Timing (Example of Chain Transfer)**

### 8.5.7 Number of DTC Execution States

Table 8.7 lists execution status for a single DTC data transfer, and table 8.7 shows the number of states required for each execution status.

**Table 8.7 DTC Execution Status**

Mode	Vector Read	Register Information		Data Read	Data Write	Internal Operations
	I	J	Read/Write	K	L	M
Normal	1	6		1	1	3
Repeat	1	6		1	1	3
Block transfer	1	6		N	N	3

Legend:

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

Object to be Accessed		On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices			
Bus width		32	16	8	16	8	16		
Access states		1	1	2	2	2	3	2	3
Execution status	Vector read $S_i$	—	1	—	—	4	6+2m	2	3+m
	Register information read/write $S_j$	1	—	—	—	—	—	—	—
	Byte data read $S_k$	1	1	2	2	2	3+m	2	3+m
	Word data read $S_k$	1	1	4	2	4	6+2m	2	3+m
	Byte data write $S_l$	1	1	2	2	2	3+m	2	3+m
	Word data write $S_l$	1	1	4	2	4	6+2m	2	3+m
Internal operation $S_m$		1							

The number of execution states is calculated from the formula below. Note that  $\Sigma$  means the sum of all transfers activated by one activation event (the number in which the CHNE bit is set to 1, plus 1).

$$\text{Number of execution states} = I \cdot S_i + \Sigma (J \cdot S_j + K \cdot S_k + L \cdot S_l) + M \cdot S_m$$

For example, when the DTC vector address table is located in on-chip ROM, normal mode is set, and data is transferred from the on-chip ROM to an internal I/O register, the time required for the DTC operation is 13 states. The time from activation to the end of the data write is 10 states.

### 8.6.1 Activation by Interrupt

The procedure for using the DTC with interrupt activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Set the corresponding bit in DTCECR to 1.
4. Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.
5. After the end of one data transfer, or after the specified number of data transfers have ended, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

### 8.6.2 Activation by Software

The procedure for using the DTC with software activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Check that the SWDTE bit is 0.
4. Write 1 to SWDTE bit and the vector number to DTVECR.
5. Check the vector number written to DTVECR.
6. After the end of one data transfer, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data transfers have ended, the SWDTE bit is held at 1 and a CPU interrupt is requested.



### 8.7.1 Normal Mode

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

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

### 8.7.2 Chain Transfer

An example of DTC chain transfer is shown in which pulse output is performed using the PPG. Chain transfer can be used to perform pulse output data transfer and PPG output trigger cycle updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the chain transfer, and normal mode transfer to the TPU's TGR in the second half. This is because clearing of the activation source and interrupt generation at the end of the specified number of transfers are restricted to the second half of the chain transfer (transfer when  $CHNE = 0$ ).

1. Perform settings for transfer to the PPG's NDR. Set MRA to source address incrementing ( $SM1 = 1$ ,  $SM0 = 0$ ), fixed destination address ( $DM1 = DM0 = 0$ ), repeat mode ( $MD1 = 0$ ,  $MD0 = 1$ ), and word size ( $Sz = 1$ ). Set the source side as a repeat area ( $DTS = 1$ ). Set MRB to chain mode ( $CHNE = 1$ ,  $DISEL = 0$ ). Set the data table start address in SAR, the NDRH address in DAR, and the data table size in CRAH and CRAL. CRB can be set to any value.

= 0), and word size (Sz = 1). Set the data table start address in SAR, the TGRA address in DAR, and the data table size in CRA. CRB can be set to any value.

3. Locate the TPU transfer register information consecutively after the NDR transfer register information.
4. Set the start address of the NDR transfer register information to the DTC vector address.
5. Set the bit corresponding to TGIA in DTCER to 1.
6. Set TGRA as an output compare register (output disabled) with TIOR, and enable the TGIA interrupt with TIER.
7. Set the initial output value in PODR, and the next output value in NDR. Set bits in DDR and NDER for which output is to be performed to 1. Using PCR, select the TPU compare match to be used as the output trigger.
8. Set the CST bit in TSTR to 1, and start the TCNT count operation.
9. Each time a TGRA compare match occurs, the next output value is transferred to NDR and the set value of the next output trigger period is transferred to TGRA. The activation source TGFA flag is cleared.
10. When the specified number of transfers are completed (the TPU transfer CRA value is 0), the TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

### 8.7.3 Chain Transfer when Counter = 0

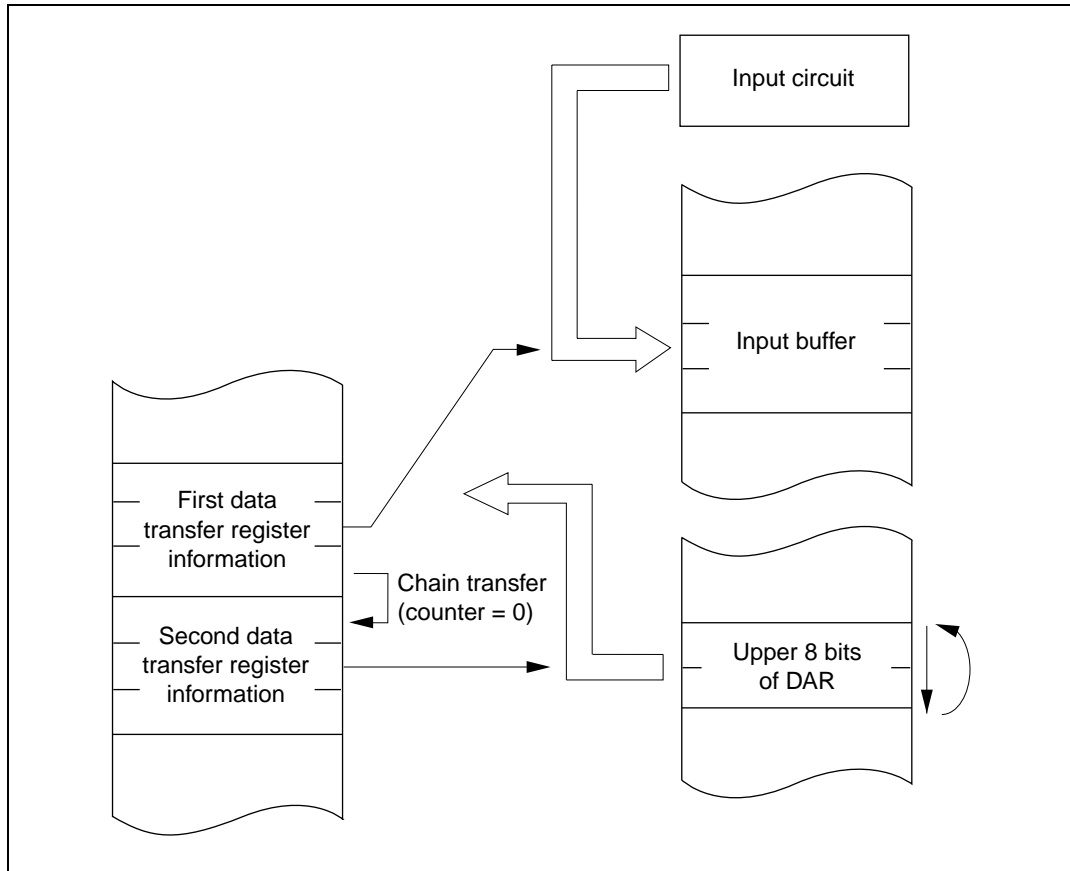
By executing a second data transfer, and performing re-setting of the first data transfer, only when the counter value is 0, it is possible to perform 256 or more repeat transfers.

An example is shown in which a 128-kbyte input buffer is configured. The input buffer is assumed to have been set to start at lower address H'0000. Figure 8.13 shows the chain transfer when the counter value is 0.

1. For the first transfer, set the normal mode for input data. Set fixed transfer source address (G/A, etc.), CRA = H'0000 (65,536 times), and CHNE = 1, CHNS = 1, and DISEL = 0.
2. Prepare the upper 8-bit addresses of the start addresses for each of the 65,536 transfer start addresses for the first data transfer in a separate area (in ROM, etc.). For example, if the input buffer comprises H'200000 to H'21FFFF, prepare H'21 and H'20.
3. For the second transfer, set repeat mode (with the source side as the repeat area) for re-setting the transfer destination address for the first data transfer. Use the upper 8 bits of DAR in the first register information area as the transfer destination. Set CHNE = DISEL = 0. If the above input buffer is specified as H'200000 to H'21FFFF, set the transfer counter to 2.

the transfer source address for the first data transfer to H'21. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.

- Next, execute the first data transfer the 65,536 times specified for the first data transfer by means of interrupts. When the transfer counter for the first data transfer reaches 0, the second data transfer is started. Set the upper 8 bits of the transfer source address for the first data transfer to H'20. The lower 16 bits of the transfer destination address of the first data transfer and the transfer counter are H'0000.
- Steps 4 and 5 are repeated endlessly. As repeat mode is specified for the second data transfer, an interrupt request is not sent to the CPU.



**Figure 8.13 Chain Transfer when Counter = 0**

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

1. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.
2. Set the start address of the register information at the DTC vector address (H'04C0).
3. Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.
4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.
5. Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.
6. If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.
7. After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform other wrap-up processing.

### **8.8.1 Module Stop Mode Setting**

DTC operation can be disabled or enabled using the module stop control register. The initial setting is for DTC operation to be enabled. Register access is disabled by setting module stop mode. Module stop mode cannot be set while the DTC is activated. For details, refer to section 23, Power-Down Modes.

### **8.8.2 On-Chip RAM**

The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

### **8.8.3 DTCE Bit Setting**

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are disabled, multiple activation sources can be set at one time (only at the initial setting) by writing data after executing a dummy read on the relevant register.

### **8.8.4 DMAC Transfer End Interrupt**

When DTC transfer is activated by a DMAC transfer end interrupt, regardless of the transfer counter and DISEL bit, the DMAC's DTE bit is not subject to DTC control, and the write data has priority. Consequently, an interrupt request may not be sent to the CPU when the DTC transfer counter reaches 0.

### **8.8.5 Chain Transfer**

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

Therefore, when the DTC is activated by an interrupt or activation source, if a read/write of the relevant register is not included in the last chained data transfer, the interrupt or activation source will be retained.



Table 9.1 summarizes the port functions. The pins of each port also have other functions such as input/output or external interrupt input pins of on-chip peripheral modules. Each I/O port includes a data direction register (DDR) that controls input/output, a data register (DR) that stores output data, and a port register (PORT) used to read the pin states. The input-only ports do not have a DR or DDR register.

Ports A to E have a built-in pull-up MOS function and a pull-up MOS control register (PCR) to control the on/off state of MOS input pull-up.

Ports 3 and A include an open-drain control register (ODR) that controls the on/off state of the output buffer PMOS.

Ports 1 to 3, 5 (P50 to P53), and 8 (P81, P83, P85) can drive a single TTL load and 30 pF capacitive load. Ports A to G can drive a single TTL load and 50 pF capacitive load.

All of the I/O ports can drive a Darlington transistor when outputting data.

Ports 1 and 2 are Schmitt-triggered inputs. Ports 4, 5, and A (PA4, PA5, PA6, PA7) are Schmitt-triggered inputs when used as IRQ inputs.

Port	Description	Mode 1*	Mode 2*	Mode 4	EXPE = 1	EXPE = 0	Input/Output Type
Port 1	General I/O port also functioning as PPG outputs, TPU I/Os, and DMAC I/Os	P17/PO15/TIOCB2/TCLKD	P16/PO14/TIOCA2	P15/PO13/TIOCB1/TCLKC/ $\overline{DACK1}$			Schmitt-triggered input
		P14/PO12/TIOCA1/ $\overline{DACK0}$		P13/PO11/TIOCD0/TCLKB/ $\overline{TEND1}$			
		P12/PO10/TIOCC0/TCLKA/ $\overline{TEND0}$		P11/PO9/TIOCB0/ $\overline{DREQ1}$			
		P10/PO8/TIOCA0/ $\overline{DREQ0}$					
Port 2	General I/O port also functioning as PPG outputs, TPU I/Os, and TMR I/Os	P27/PO7/TIOCB5	P26/PO6/TIOCA5	P25/PO5/TIOCB4/TMO1			Schmitt-triggered input
		P24/PO4/TIOCA4/RxD4/TMO0		P23/PO3/TIOCD3/TxD4/TMC11			
		P22/PO2/TIOCC3/TMCIO		P21/PO1/TIOCB3/TMRI1			
		P20/PO0/TIOCA3/TMIR0					
Port 3	General I/O port also functioning as SCI I/Os, I <sup>2</sup> C I/Os, and bus control I/Os	P35/SCK1/SCL0/ $\overline{OE}$			P35/SCK1/SCL0/ $\overline{OE}$	P35/SCK1/SCL0	Open-drain output capability
		P34/SCK0/SCK4/SDA0					
		P33/RxD1/SCL1					
		P32/RxD0/IrRxD/SDA1					
		P31/TxD1					
		P30/TxD0/IrTxD					
Port 4	General I/O port also functioning as A/D converter analog inputs and D/A converter analog outputs	P47/AN7/ $\overline{IRQ7}$	P46/AN6/ $\overline{IRQ6}$	P45/AN5/ $\overline{IRQ5}$			Schmitt-triggered input when used as IRQ input
		P44/AN4/ $\overline{IRQ4}$		P43/AN3/ $\overline{IRQ3}$			
		P42/AN2/ $\overline{IRQ2}$		P41/AN1/ $\overline{IRQ1}$			
		P40/AN0/ $\overline{IRQ0}$					



Port	Description	Mode 1	Mode 2	Exp. 1	Exp. 2	Type
Port 5	General I/O port also functioning as interrupt inputs, A/D converter inputs, and SCI I/Os	P53/ $\overline{ADTRG}/\overline{IRQ3}$ P52/ $\overline{SCK2}/\overline{IRQ2}$ P51/ $\overline{RxD2}/\overline{IRQ1}$ P50/ $\overline{TxD2}/\overline{IRQ0}$				Schmitt-triggered input when used as IRQ input
Port 8	General I/O port also functioning as SCI I/Os	P85/ $\overline{SCK3}$ P83/ $\overline{RxD3}$ P81/ $\overline{TxD3}$				
Port 9	Dedicated input port also functioning as A/D converter analog inputs and D/A converter analog outputs	P95/ $\overline{AN13}/\overline{DA3}$ P94/ $\overline{AN12}/\overline{DA2}$				
Port A	General I/O port also functioning as address outputs interrupt inputs, and bus control I/Os	PA7/ $\overline{A23}/\overline{CS7}/\overline{IRQ7}$ PA6/ $\overline{A22}/\overline{IRQ6}$ PA5/ $\overline{A21}/\overline{IRQ5}$ A20/ $\overline{IRQ4}$ A19 A18 A17 A16	PA7/ $\overline{A23}/\overline{CS7}/\overline{IRQ7}$ PA6/ $\overline{A22}/\overline{IRQ6}$ PA5/ $\overline{A21}/\overline{IRQ5}$ PA4/ $\overline{A20}/\overline{IRQ4}$	PA7/ $\overline{A23}/\overline{CS7}/\overline{IRQ7}$ PA6/ $\overline{A22}/\overline{IRQ6}$ PA5/ $\overline{A21}/\overline{IRQ5}$ PA4/ $\overline{A20}/\overline{IRQ4}$	PA7/ $\overline{IRQ7}$ PA6/ $\overline{IRQ6}$ PA5/ $\overline{IRQ5}$ PA4/ $\overline{IRQ4}$ PA3 PA2 PA1 PA0	Only PA4 to PA7 are Schmitt-triggered input when used as IRQ input. Built-in MOS input pull-up Open-drain output capability
Port B	General I/O port also functioning as address outputs	A15 A14 A13 A12 A11 A10 A9 A8	PB7/A15 PB6/A14 PB5/A13 PB4/A12 PB3/A11 PB2/A10 PB1/A9 PB0/A8	PB7/A15 PB6/A14 PB5/A13 PB4/A12 PB3/A11 PB2/A10 PB1/A9 PB0/A8	PB7 PB6 PB5 PB4 PB3 PB2 PB1 PB0	Built-in MOS input pull-up

Port	Description	Mode 1	Mode 2	EX2	EX3	Type
Port C	General I/O port also functioning as address outputs	A7	PC7/A7	PC7/A7	PC7	Built-in MOS input pull-up
		A6	PC6/A6	PC6/A6	PC6	
		A5	PC5/A5	PC5/A5	PC5	
		A4	PC4/A4	PC4/A4	PC4	
		A3	PC3/A3	PC3/A3	PC3	
		A2	PC2/A2	PC2/A2	PC2	
		A1	PC1/A1	PC1/A1	PC1	
		A0	PC0/A0	PC0/A0	PC0	
Port D	General I/O port also functioning as data I/Os	D15		D15	PD7	Built-in MOS input pull-up
		D14		D14	PD6	
		D13		D13	PD5	
		D12		D12	PD4	
		D11		D11	PD3	
		D10		D10	PD2	
		D9		D9	PD1	
		D8		D8	PD0	
Port E	General I/O port also functioning as data I/Os	PE7/D7		PE7/D7	PE7	Built-in MOS input pull-up
		PE6/D6		PE6/D6	PE6	
		PE5/D5		PE5/D5	PE5	
		PE4/D4		PE4/D4	PE4	
		PE3/D3		PE3/D3	PE3	
		PE2/D2		PE2/D2	PE2	
		PE1/D1		PE1/D1	PE1	
		PE0/D0		PE0/D0	PE0	
Port F	General I/O port also functioning as interrupt inputs and bus control I/Os	PF7/ $\phi$		PF7/ $\phi$	PF7/ $\phi$	
		PF6/ $\overline{AS}$		PF6/ $\overline{AS}$	PF6	
		$\overline{RD}$		$\overline{RD}$	PF5	
		$\overline{HWR}$		$\overline{HWR}$	PF4	
		PF3/ $\overline{LWR}$		PF3/ $\overline{LWR}$	PF3	
		PF2/ $\overline{CS6/LCAS}$		PF2/ $\overline{CS6/LCAS}$	PF2	
		PF1/ $\overline{CS5/UCAS}$		PF1/ $\overline{CS5/UCAS}$	PF1	
		PF0/ $\overline{WAIT/OE}$		PF0/ $\overline{WAIT/OE}$	PF0	

Port	Description	Mode 1	Mode 2	Mode 3	Type
Port G	General I/O port also functioning as bus control I/Os	PG6/ $\overline{\text{BREQ}}$		PG6/ $\overline{\text{BREQ}}$	PG6
		PG5/ $\overline{\text{BACK}}$		PG5/ $\overline{\text{BACK}}$	PG5
		PG4/ $\overline{\text{CS4}}/\overline{\text{BREQO}}$		PG4/ $\overline{\text{CS4}}/\overline{\text{BREQO}}$	PG4
		PG3/ $\overline{\text{CS3}}/\overline{\text{RAS3}}$		PG3/ $\overline{\text{CS3}}/\overline{\text{RAS3}}$	PG3
		PG2/ $\overline{\text{CS2}}/\overline{\text{RAS2}}$		PG2/ $\overline{\text{CS2}}/\overline{\text{RAS2}}$	PG2
		PG1/ $\overline{\text{CS1}}$ PG0/ $\overline{\text{CS0}}$		PG1/ $\overline{\text{CS1}}$ PG0/ $\overline{\text{CS0}}$	PG1 PG0

Note: \* Only modes 1 and 2 are available in the ROMless version.

## 9.1 Port 1

Port 1 is an 8-bit I/O port that also has other functions. The port 1 has the following registers.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 register (PORT1)

### 9.1.1 Port 1 Data Direction Register (P1DDR)

The individual bits of P1DDR specify input or output for the pins of port 1.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	

P1DR stores output data for the port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	P16DR	0	R/W	
5	P15DR	0	R/W	
4	P14DR	0	R/W	
3	P13DR	0	R/W	
2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

### 9.1.3 Port 1 Register (PORT1)

PORT1 shows the pin states.

PORT1 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	—*	R	If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read. If a port 1 read is performed while P1DDR bits are cleared to 0, the pin states are read.
6	P16	—*	R	
5	P15	—*	R	
4	P14	—*	R	
3	P13	—*	R	
2	P12	—*	R	
1	P11	—*	R	
0	P10	—*	R	

Note: \* Determined by the states of pins P17 to P10.

Port 1 pins also function as the pins for PPG outputs, TPU I/Os, and DMAC outputs. The correspondence between the register specification and the pin functions is shown below.

- P17/PO15/TIOCB2/TCLKD

The pin function is switched as shown below according to the combination of the TPU channel 2 settings (by bits MD3 to MD0 in TMDR\_2, bits IOB3 to IOB0 in TIOR\_2, and bits CCLR1 and CCLR0 in TCR\_2), bits TPSC2 to TPSC0 in TCR\_0 and TCR\_5, bit NDER15 in NDERH, and bit P17DDR.

TPU channel 2 settings	(1) in table below	(2) in table below		
P17DDR	—	0	1	1
NDER15	—	—	0	1
Pin function	TIOCB2 output	P17 input	P17 output	PO15 output
		TIOCB2 input* <sup>1</sup>		
	TCLKD input* <sup>2</sup>			

Notes: 1. TIOCB2 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 = 1.

2. TCLKD input when the setting for either TCR\_0 or TCR\_5 is TPSC2 to TPSC0 = B'111.  
TCLKD input when channels 2 and 4 are set to phase counting mode.

TPU channel 2 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

x: Don't care

2 settings (by bits MD3 to MD0 in TMDR\_2, bits IOA3 to IOA0 in TIOR\_2, and bits CCLR1 and CCLR0 in TCR\_2), bit NDER14 in NDERH, and bit P16DDR.

TPU channel 2 settings	(1) in table below	(2) in table below		
P16DDR	—	0	1	1
NDER14	—	—	0	1
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output
		TIOCA2 input*1		

Note: 1. TIOCA2 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 = 1.

TPU channel 1 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

Legend:

x: Don't care

Note: 2. TIOCB2 output disabled.

1 settings (by bits MD3 to MD0 in TMDR\_1, bits IOB3 to IOB0 in TIOR\_1, and bits CCLR1 and CCLR0 in TCR\_1), bits TPSC2 to TPSC0 in TCR\_0, TCR\_2, TCR\_4, and TCR\_5, bit NDER13 in NDERH, bit SAE1 in DMA BCRH and bit P15DDR.

SAE1	0			1	1
TPU channel 2 settings	(1) in table below	(2) in table below			—
P15DDR	—	0	1	1	—
NDER13	—	—	0	1	—
Pin function	TIOCB1 output	P15 input	P15 output	PO13 output	$\overline{\text{DACK1}}$ output
		TIOCB1 input*1			
	TCLKC input*2				

- Notes: 1. TIOCB1 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 to IOB0 = B'10xx.  
 2. TCLKC input when the setting for either TCR\_0 or TCR\_2 is TPSC2 to TPSC0 = B'111, or when the setting for either TCR\_4 or TCR\_5 is TPSC2 to TPSC0 = B'101.  
 TCLKC input when phase counting mode is set for channels 2 and 4.

TPU channel 1 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

x: Don't care

1 settings (by bits MD3 to MD0 in TMDR\_1, bits IOA3 to IOA0 in TIOR\_1, and bits CCLR1 and CCLR0 in TCR\_1), bit NDER12 in NDERH, bit SAE0 in DMABCRH and bit P14DDR.

SAE0	0		1	1	
TPU channel 1 settings	(1) in table below	(2) in table below			—
P14DDR	—	0	1	1	—
NDER12	—	—	0	1	—
Pin function	TIOCB1 output	P14 input	P14 output	PO12 output	$\overline{\text{DACK0}}$ output
		TIOCA1 input*1			

Note: 1. TIOCA1 input when MD3 to MD0 = B'0000, and B'01xx and IOA3 to IOA0 = B'10xx.

TPU channel 1 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

Legend:

x: Don't care

Note: 2. TIOCB1 output disabled.



0 settings (by bits MD3 to MD0 in TMDR\_0, bits IOD3 to IOD0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR\_0), bits TPSC2 to TPSC0 in TCR\_0 to TCR\_2, bit NDER11 in NDERH, bit TEE1 in DMATCR of DMAC and bit P13DDR.

TEE1	0				1
TPU channel 0 settings	(1) in table below	(2) in table below		—	—
P13DDR	—	0	1	1	—
NDER11	—	—	0	1	—
Pin function	TIOCD0 output	P13 input	P13 output	PO11 output	$\overline{\text{TEND1}}$ output
		TIOCD0 input* <sup>1</sup>			
TCLKB input* <sup>2</sup>					

Notes: 1. TIOCD0 input when MD3 to MD0 = B'0000 and IOD3 to IOD0 = B'10xx.

2. TCLKB input when the setting for any of TCR\_0 to TCR\_2 is TPSC2 to TPSC0 = B'101.  
TCLKB input when phase counting mode is set for channels 1 and 5.

TPU channel 0 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

x: Don't care

0 settings (by bits MD3 to MD0 in TMDR\_0, bits IOC3 to IOC0 in TIORL\_0, and bits CCLR2 to CCLR0 in TCR\_0), bits TPSC2 to TPSC0 in TCR\_0 to TCR\_5, bit NDER10 in NDERH, bit TEE0 in DMATCR of DMAC and bit P12DDR.

TEE0	0				1
TPU channel 2 settings	(1) in table below	(2) in table below			—
P12DDR	—	0	1	1	—
NDER10	—	—	0	1	—
Pin function	TIOCO0 output	P12 input	P12 output	PO10 output	$\overline{\text{TEND0}}$ output
		TIOCC0 input <sup>*1</sup>			
TCLKA input <sup>*2</sup>					

Notes: 1. TIOCC0 input when MD3 to MD0 = B'0000 and IOC3 to IOC0 = B'10xx.

2. TCLKA input when the setting for any of TCR\_0 to TCR\_5 is TPSC2 to TPSC0 = B'100.  
TCLKA input when phase counting mode is set for channels 1 and 5.

TPU channel 0 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM <sup>*3</sup> mode 1 output	PWM mode 2 output	—

Legend:

x: Don't care

Note: 3. TIOCD0 output disabled.

Output disabled and settings (2) effective when BFA = 1 or BFB = 1 in TMDR\_0.

0 settings (by bits MD3 to MD0 in TMDR\_0, bits IOB3 to IOB0 in TIORH\_0, and bits CCLR2 to CCLR0 in TCR\_0), bit NDER9 in NDERH, and bit P11DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P11DDR	—	0	1	1
NDER9	—	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output
		TIOCB0 input*		
	$\overline{\text{DREQ1}}$ input			

Note: \* TIOCB0 input when MD3 to MD0 = B'0000 and IOB3 to IOB0 = B'10xx.

TPU channel 0 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

x: Don't care

0 settings (by bits MD3 to MD0 in TMDR\_0, bits IOA3 to IOA0 in TIORH\_0, and bits CCLR2 to CCLR0 in TCR\_0), bit NDER8 in NDERH, and bit P10DDR.

TPU channel 0 settings	(1) in table below	(2) in table below		
P10DDR	—	0	1	1
NDER8	—	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output
		TIOCA0 input* <sup>1</sup>		
	DREQ0 input			

Note: 1. TIOCA0 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10xx.

TPU channel 0 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2, CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM* <sup>2</sup> mode 1 output	PWM mode 2 output	—

Legend:

x: Don't care

Note: 2. TIOCB0 output disabled.

Port 2 is an 8-bit I/O port that also has other functions. The port 2 has the following registers.

- Port 2 data direction register (P2DDR)
- Port 2 data register (P2DR)
- Port 2 register (PORT2)

### 9.2.1 Port 2 Data Direction Register (P2DDR)

The individual bits of P2DDR specify input or output for the pins of port 2.

P2DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

P2DR stores output data for the port 2 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	P26DR	0	R/W	
5	P25DR	0	R/W	
4	P24DR	0	R/W	
3	P23DR	0	R/W	
2	P22DR	0	R/W	
1	P21DR	0	R/W	
0	P20DR	0	R/W	

### 9.2.3 Port 2 Register (PORT2)

PORT2 shows the pin states.

PORT2 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P27	—*	R	If a port 2 read is performed while P2DDR bits are set to 1, the P2DR values are read. If a port 2 read is performed while P2DDR bits are cleared to 0, the pin states are read.
6	P26	—*	R	
5	P25	—*	R	
4	P24	—*	R	
3	P23	—*	R	
2	P22	—*	R	
1	P21	—*	R	
0	P20	—*	R	

Note: \* Determined by the states of pins P27 to P20.

Port 2 pins also function as PPG outputs, TPU I/Os, and TMR I/Os. The correspondence between the register specification and the pin functions is shown below.

- P27/PO7/TIOCB5

The pin function is switched as shown below according to the combination of the TPU channel 5 settings (by bits MD3 to MD0 in TMDR\_5, bits IOB3 to IOB0 in TIOR\_5, and bits CCLR1 and CCLR0 in TCR\_5), bit NDER7 in NDERL, and bit P27DDR.

TPU channel 5 settings	(1) in table below	(2) in table below		
		P27DDR	—	0
NDER7	—	—	0	1
Pin function	TIOCB5 output	P27 input	P27 output	PO7 output
		TIOCB5 input*		

Note: \* TIOCB5 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 = 1.

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000 to B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

x: Don't care

5 settings (by bits MD3 to MD0 in TMDR\_5, bits IOA3 to IOA0 in TIOR\_5, and bits CCLR1 and CCLR0 in TCR\_5), bit NDER6 in NDERL, and bit P26DDR.

TPU channel 5 settings	(1) in table below	(2) in table below		
P26DDR	—	0	1	1
NDER6	—	—	0	1
Pin function	TIOCA5 output	P26 input	P26 output	PO6 output
		TIOCA5 input* <sup>1</sup>		

Note: 1. TIOCA5 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 = 1.

TPU channel 5 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000 to B'00xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM* <sup>2</sup> mode 1 output	PWM mode 2 output	—

Legend:

x: Don't care

Note: 2. TIOCB5 output disabled.



4 settings (by bits MD3 to MD0 in TMDR\_4, bits IOB3 to IOB0 in TIOR\_4, and bits CCLR1 and CCLR0 in TCR\_4), bit NDER5 in NDERL, bit P25DDR, and bits OS3 to USO in TCSRI of TMR.

TPU channel 4 settings	(1) in table below	(2) in table below			
		OS3 to OS0	—	All 0	
P25DDR	—	0	1	1	—
NDER5	—	—	0	1	—
Pin function	TIOCB4 output	P25 input	P25 output	PO5 output	TMO1 output
		TIOCB4 input*			

Note: \* TIOCB4 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 to IOB0 = B'10xx.

TPU channel 4 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000 to B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

x: Don't care

4 settings (by bits MD3 to MD0 in TMDR\_4, bits IOA3 to IOA0 in TIOR\_4, and bits CCLR1 and CCLR0 in TCR4), bit NDER4 in NDERL, bit RE in SCI\_4, bit P24DDR, and bit OS3 to OS0 in TCSRO of TMR.

RE	0					1
TPU channel 4 settings	(1) in table below	(2) in table below				—
OS3 to OS0	All 0				Not all 0	—
P24DDR	—	0	1	1	—	—
NDER4	—	—	0	1	—	—
Pin function	TIOCA4 output	P24 input	P24 output	PO4 output	TMO0 output	RXD4 input
		TIOCA4 input*1				

Note: 1. TIOCA4 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 to IOA0 = B'10xx.

TPU channel 4 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000 to B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM*2 mode 1 output	PWM mode 2 output	—

Legend:

x: Don't care

Note: 2. TIOCB4 output disabled.

3 settings (by bits MD3 to MD0 in TMDR\_3, bits IOD3 to IOD0 in TIORL\_3, and bits CCLR2 to CCLR0 in TCR\_3), bit NDER3 in NDERL, bit TE in SCR of SCI\_4, and bit P23DDR.

TPU channel 3 settings	(1) in table below	(2) in table below			
TE	0				1
P23DDR	—	0	1	1	—
NDER3	—	—	0	1	—
Pin function	TIOCD3 output	P23 input	P23 output	PO3 output	TxD4 output
		TIOCD3 input <sup>*1</sup>			
TMCI1 input <sup>*2</sup>					

Notes: 1. TIOCD3 input when MD3 to MD0 = B'0000 and IOD3 to IOD0 = B'10xx.

2. When used as the TMR external clock input pin, the external clock is selected by the CKS2 to CKS0 bits in TCR1.

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0001 to B'0011		B'0010	B'0011		
IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

x: Don't care

3 settings (by bits MD3 to MD0 in TMDR\_3, bits IOC3 to IOC0 in TIORL\_3, and bits CCLR2 to CCLR0 in TCR\_3), bit NDER2 in NDERL, and bit P22DDR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P22DDR	—	0	1	1
NDER2	—	—	0	1
Pin function	TIOCC3 output	P22 input	P22 output	PO2 output
		TIOCC3 input* <sup>1</sup>		
	TMC10 input* <sup>2</sup>			

Notes: 1. TIOCC3 input when MD3 to MD0 = B'0000 and IOC3 to IOC0 = B'10xx.

2. When used as the TMR external clock input pin, the external clock is selected by the CKS2 to CKS0 bits in TCR\_1.

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM* <sup>3</sup> mode 1 output	PWM mode 2 output	—

Legend:

x: Don't care

Note: 3. TIOCD3 output disabled.

Output disabled and settings (2) effective when BFA = 1 or BFB = 1 in TMDR\_3.

3 settings (by bits MD3 to MD0 in TMDR\_3, bits IOB3 to IOB0 in TIORH\_3, and bits CCLR2 to CCLR0 in TCR\_3), bit NDER1 in NDERL, and bit P21DDR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P21DDR	—	0	1	1
NDER1	—	—	0	1
Pin function	TIOCB3 output	P21 input	P21 output	PO1 output
		TIOCB3 input*1		
	TMRI1 input*2			

Notes: 1. TIOCB3 input when MD3 to MD0 = B'0000 and IOB3 to IOB0 = B'10xx.

2. When used as the TMR counter reset pin, set both the CCLR1 and CCLR0 bits in TCR\_1 to 1.

TPU channel 3 settings	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

Legend:

x: Don't care

3 settings (by bits MD3 to MD0 in TMDR\_3, bits IOA3 to IOA0 in TIORH\_3, and bits CCLR2 to CCLR0 in TCR\_3), bit NDER0 in NDERL, and bit P20DDR.

TPU channel 3 settings	(1) in table below	(2) in table below		
P20DDR	—	0	1	1
NDER0	—	—	0	1
Pin function	TIOCA3 output	P20 input	P20 output	PO0 output
		TIOCA0 input* <sup>1</sup>		
	TMRI0 input* <sup>2</sup>			

Notes: 1. TIOCA3 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10xx.

2. When used as the TMR counter reset pin, set both the CCLR1 and CCLR0 bits in TCR\_1 to 1.

TPU channel 3 settings	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00	Other than B'xx00	
CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM* <sup>3</sup> mode 1 output	PWM mode 2 output	—

Legend:

x: Don't care

Note: 3. TIOCB3 output disabled.

Port 3 is a 6-bit I/O port that also has other functions. The port 3 has the following registers.

- Port 3 data direction register (P3DDR)
- Port 3 data register (P3DR)
- Port 3 register (PORT3)
- Port 3 open drain control register (P3ODR)
- Port function control register 2(PFCR2)

### 9.3.1 Port 3 Data Direction Register (P3DDR)

The individual bits of P3DDR specify input or output for the pins of port 3.

P3DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	P35DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
4	P34DDR	0	W	
3	P33DDR	0	W	
2	P32DDR	0	W	
1	P31DDR	0	W	
0	P30DDR	0	W	

P3DR stores output data for the port 3 pins.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	P35DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
4	P34DR	0	R/W	
3	P33DR	0	R/W	
2	P32DR	0	R/W	
1	P31DR	0	R/W	
0	P30DR	0	R/W	

### 9.3.3 Port 3 Register (PORT3)

PORT3 shows the pin states.

PORT3 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	P35	—*	R	If a port 3 read is performed while P3DDR bits are set to 1, the P3DR values are read. If a port 1 read is performed while P3DDR bits are cleared to 0, the pin states are read.
4	P34	—*	R	
3	P33	—*	R	
2	P32	—*	R	
1	P31	—*	R	
0	P30	—*	R	

Note: \* Determined by the states of pins P35 to P30.



P3ODR controls the output status for each port 3 pin.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	P35ODR	0	R/W	Setting a P3ODR bit to 1 makes the corresponding port 3 pin an NMOS open-drain output pin, while clearing the bit to 0 makes the pin a CMOS output pin.
4	P34ODR	0	R/W	
3	P33ODR	0	R/W	
2	P32ODR	0	R/W	
1	P31ODR	0	R/W	
0	P30ODR	0	R/W	

PFCR2 controls the I/O port.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	ASOE	1	R/W	$\overline{AS}$ Output Enable Selects to enable or disable the $\overline{AS}$ output pin. 0: PF6 is designated as I/O port 1: PF6 is designated as $\overline{AS}$ output pin
2	LWROE	1	R/W	$\overline{LWR}$ Output Enable Selects to enable or disable the $\overline{LWR}$ output pin. 0: PF3 is designated as I/O port 1: PF3 is designated as $\overline{LWR}$ output pin
1	OES	1	R/W	$\overline{OE}$ Output Select Selects the $\overline{OE}$ output pin port when the OEE bit is set to 1 in DRAMCR (enabling $\overline{OE}$ output). 0: P35 is designated as $\overline{OE}$ output pin 1: PH3 is designated as $\overline{OE}$ output pin
0	—	0	—	Reserved This bit is always read as 0 and cannot be modified.

Port 3 pins also function as the pins for SCI I/Os, I<sup>2</sup>C output, and a bus control signal output. The correspondence between the register specification and the pin functions is shown below.

- P35/SCK1/SCL0/( $\overline{\text{OE}}$ )

The pin function is switched as shown below according to the combination of the ICE bit in ICCRA of I<sup>2</sup>C\_0, C/ $\overline{\text{A}}$  bit in SMR of SCI\_1, bits CKE0 and CKE1 in SCR, bits OEE in DRAMCR, bit OES in PFCR2, and bit P35DDR.

Modes 1, 2, 4, 7 (EXPE = 1)

OEE	0						1						
OES	—						1						0
ICE	—					1	0					1	—
CKE1	0				1	—	0				1	—	—
C/ $\overline{\text{A}}$	0			1	—	—	0			1	—	—	—
CKE0	0		1	—	—	—	0		1	—	—	—	—
P35DDR	0	1	—	—	—	—	0	1	—	—	—	—	—
Pin function	P35 input	P35 output* <sup>1</sup>	SCK1 output* <sup>1</sup>	SCK1 output* <sup>1</sup>	SCK1 input	SCL0 I/O* <sup>2</sup>	P35 input	P35 output* <sup>1</sup>	SCK1 output* <sup>1</sup>	SCK1 output* <sup>1</sup>	SCK1 input	SCL0 I/O* <sup>2</sup>	$\overline{\text{OE}}$ output

Mode 7 (EXPE = 0)

OEE	—					
OES	—					
ICE	0					1
CKE1	0				1	—
C/ $\overline{\text{A}}$	0			1	—	—
CKE0	0		1	—	—	—
P35DDR	0	1	—	—	—	—
Pin function	P35 input	P35 output* <sup>1</sup>	SCK1 output* <sup>1</sup>	SCK1 output* <sup>1</sup>	SCK1 input	SCL0 I/O* <sup>2</sup>

- Notes: 1. NMOS open-drain output when P35ODR = 1.  
 2. NMOS open-drain output regardless of P35ODR

of I<sup>2</sup>C\_0, bit C/A in SMR, bits CKE0 and CKE1 in SCR, and bit P34DDR.

ICE	0				1	
CKE1	0			1		—
C/A	0		1		—	—
CKE0	0		1		—	—
P34DDR	0	1	—	—	—	—
Pin function	P34 input	P34 output <sup>*1</sup>	SCK0/SCK4 output <sup>*1*3</sup>	SCK0/SCK4 output <sup>*1*3</sup>	SCK0/SCK4 input	SDA0 I/O <sup>*2</sup>

- Notes: 1. NMOS open-drain output when P34ODR = 1.  
 2. NMOS open-drain output regardless of P34ODR  
 3. Simultaneous output of SCK0 and SCK4 cannot be set.

- P33/RxD1/SCL1

The pin function is switched as shown below according to the combination of bit ICE in ICCRA of I<sup>2</sup>C\_0, bit RE in SCR of SCI\_1 and bit P33DDR.

ICE	0			1	
RE	0		1		—
P33DDR	0	1	—	—	
Pin function	P33 input	P33 output <sup>*1</sup>	RxD1 input	SCL1 I/O <sup>*2</sup>	

- Notes: 1. NMOS open-drain output when P33ODR = 1.  
 2. NMOS open-drain output regardless of P33ODR

- P32/RxD0/IrRxD/SDA1

The pin function is switched as shown below according to the combination of bit ICE in ICCRA of I<sup>2</sup>C\_0, bit RE in SCR of SCI\_0 and bit P32DDR.

ICE	0			1	
RE	0		1		—
P32DDR	0	1	—	—	
Pin function	P32 input	P32 output <sup>*1</sup>	RxD0/IrRxD input	SDA1 I/O <sup>*2</sup>	

- Notes: 1. NMOS open-drain output when P32ODR = 1.  
 2. NMOS open-drain output regardless of P32ODR

SCI\_1 and bit P31DDR.

TE	0		1
P31DDR	0	1	—
Pin function	P31 input	P31 output*	TxD1 output*

Note: \* NMOS open-drain output when P31ODR = 1.

- P30/TxD0/IrTxD

The pin function is switched as shown below according to the combination of bit TE in SCR of SCI\_0 and bit P30DDR.

TE	0		1
P30DDR	0	1	—
Pin function	P30 input	P30 output*	TxD0/IrTxD output*

Note: \* NMOS open-drain output when P30ODR = 1.

Port 4 is an 8-bit input-only port. Port 4 has the following register.

- Port 4 register (PORT4)

#### 9.4.1 Port 4 Register (PORT4)

PORT4 is an 8-bit read-only register that shows port 4 pin states.

PORT4 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P47	—*	R	The pin states are always read from this register.
6	P46	—*	R	
5	P45	—*	R	
4	P44	—*	R	
3	P43	—*	R	
2	P42	—*	R	
1	P41	—*	R	
0	P40	—*	R	

Note: \* Determined by the states of pins P47 to P40.

Port 4 also functions as the pins for A/D converter analog input and D/A converter analog output. The correspondence between pins are as follows.

**P47/AN7/( $\overline{\text{IRQ7}}$ )**

Pin function	AN7 input
	$\overline{\text{IRQ7}}$ interrupt input*

Note: \*  $\overline{\text{IRQ7}}$  input when bit ITS7 in ITSr is 1.

**P46/AN6/DA0/( $\overline{\text{IRQ6}}$ )**

Pin function	AN6 input
	$\overline{\text{IRQ6}}$ interrupt input*

Note: \*  $\overline{\text{IRQ6}}$  input when bit ITS6 in ITSr is 1.

**P45/AN5/( $\overline{\text{IRQ5}}$ )**

Pin function	AN5 input
	$\overline{\text{IRQ5}}$ interrupt input*

Note: \*  $\overline{\text{IRQ5}}$  input when bit ITS5 in ITSr is 1.

**P44/AN4/( $\overline{\text{IRQ4}}$ )**

Pin function	AN4 input
	$\overline{\text{IRQ4}}$ interrupt input*

Note: \*  $\overline{\text{IRQ4}}$  input when bit ITS4 in ITSr is 1.

**P43/AN3/( $\overline{\text{IRQ3}}$ )**

Pin function	AN3 input
	$\overline{\text{IRQ3}}$ interrupt input*

Note: \*  $\overline{\text{IRQ3}}$  input when bit ITS3 in ITSr is 1.

Pin function	AN2 input
	$\overline{\text{IRQ2}}$ interrupt input*

Note: \*  $\overline{\text{IRQ2}}$  input when bit ITS2 in ITSR is 1.

### **P41/AN1/( $\overline{\text{IRQ1}}$ )**

Pin function	AN1 input
	$\overline{\text{IRQ1}}$ interrupt input*

Note: \*  $\overline{\text{IRQ1}}$  input when bit ITS1 in ITSR is 1.

### **P40/AN0/( $\overline{\text{IRQ0}}$ )**

Pin function	AN0 input
	$\overline{\text{IRQ0}}$ interrupt input*

Note: \*  $\overline{\text{IRQ0}}$  input when bit ITS0 in ITSR is 1.



Port 5 is a 4-bit I/O port. The port 5 has the following registers.

- Port 5 data direction register (P5DDR)
- Port 5 data register (P5DR)
- Port 5 register (PORT5)

### 9.5.1 Port 5 Data Direction Register (P5DDR)

The individual bits of P5DDR specify input or output for the pins of port 5.

P5DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	P53DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
2	P52DDR	0	W	
1	P51DDR	0	W	
0	P50DDR	0	W	

### 9.5.2 Port 5 Data Register (P5DR)

P5DR stores output data for the port 5 pins.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	P53DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
2	P52DR	0	R/W	
1	P51DR	0	R/W	
0	P50DR	0	R/W	

PORT5 shows the pin states. PORT5 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	R	Reserved Undefined values are read from these bits.
3	P53	—*	R	If bits P53 to P50 are read while P5DDR bits are set to 1, the P5DR values are read. If a port 5 read is performed while P5DDR bits are cleared to 0, the pin states are read.
2	P52	—*	R	
1	P51	—*	R	
0	P50	—*	R	

Note: \* Determined by the states of pins P53 to P50.

### 9.5.4 Pin Functions

Port 5 pins also function as the pins for SCI I/Os, A/D converter inputs, and interrupt inputs. The correspondence between the register specification and the pin functions is shown below.

- P53/ $\overline{\text{ADTRG}}$ / $\overline{\text{IRQ3}}$

The pin function is switched as shown below according to the combination of bits TRGS1 and TRGS0 in the A/D control register (ADCR), bit ITS3 in ITSR, and bit P53DDR.

P53DDR	0	1
Pin function	P53 input	P53 output
	$\overline{\text{ADTRG}}$ input <sup>*1</sup>	
	$\overline{\text{IRQ3}}$ interrupt input <sup>*2</sup>	

Notes: 1.  $\overline{\text{ADTRG}}$  input when TRGS1 = TRGS0 = 1.

2.  $\overline{\text{IRQ3}}$  input when ITS3 = 0.

SCI\_2, bits CKE0 and CKE1 in SCR, bit ITS2 in ITSR, and bit P52DDR.

CKE1	0			1	
$\overline{C/\overline{A}}$	0		1	—	
CKE0	0		1	—	—
P52DDR	0	1	—	—	—
Pin function	P52 input	P52 output	SCK2 output	SCK2 output	SCK2 input
	$\overline{\text{IRQ2}}$ interrupt input*				

Note: \*  $\overline{\text{IRQ2}}$  input when ITS2 = 0.

- P51/RxD2/ $\overline{\text{IRQ1}}$

The pin function is switched as shown below according to the combination of bit RE in SCR of SCI\_2, bit ITS1 in ITSR, and bit P51DDR.

RE	0		1
P51DDR	0	1	—
Pin function	P51 input	P51 output	RxD2 input
	$\overline{\text{IRQ1}}$ interrupt input*		

Note: \*  $\overline{\text{IRQ1}}$  input when ITS1 = 0.

- P50/TxD2/ $\overline{\text{IRQ0}}$

The pin function is switched as shown below according to the combination of bit TE in SCR of SCI\_2, bit ITS0 in ITSR, and bit P50DDR.

TE	0		1
P50DDR	0	1	—
Pin function	P50 input	P50 output	TxD2 input
	$\overline{\text{IRQ0}}$ interrupt input*		

Note: \*  $\overline{\text{IRQ0}}$  input when ITS0 = 0.

Port 8 is a 6-bit I/O port that also has other functions. The port 8 has the following registers.

- Port 8 data direction register (P8DDR)
- Port 8 data register (P8DR)
- Port 8 register (PORT8)

### 9.6.1 Port 8 Data Direction Register (P8DDR)

The individual bits of P8DDR specify input or output for the pins of port 8.

P8DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	P85DDR	0	W	When a pin function is specified to a general purpose I/O, setting bit 5, 3, or 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
4	—	0	—	
3	P83DDR	0	W	Bits 4, 2, and 0 are reserved.
2	—	0	—	
1	P81DDR	0	W	
0	—	0	—	

P8DR stores output data for the port 8 pins.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
5	P85DR	0	R/W	Bits 5, 3, and 1 store output data when the pin function is specified to a general purpose I/O.
4	—	0	—	
3	P83DR	0	R/W	Bits 4, 2, and 0 are reserved.
2	—	0	—	
1	P81DR	0	R/W	
0	—	0	—	

### 9.6.3 Port 8 Register (PORT8)

PORT8 shows the pin states.

PORT8 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	Undefined	—	Reserved These bits are reserved, if read they will return an undefined value.
5	P85	—*	R	If a port 8 read is performed while P8DDR bits are set to 1, the P8DR values are read. If a port 8 read is performed while P8DDR bits are cleared to 0, the pin states are read.
4	—	Undefined	R	
3	P83	—*	R	Bits 4, 2, and 0 are reserved.
2	—	Undefined	R	
1	P81	—*	R	
0	—	Undefined	R	

Note: \* Determined by the states of pins P85, P83 and P81.

Port 8 pins also function as interrupt inputs and SCI\_3 I/Os. The correspondence between the register specification and the pin functions is shown below.

- P85/SCK3

The pin function is switched as shown below according to the combination of bit  $C/\bar{A}$  in SMR in SCI\_3, bits CKE0 and CKE1 in SCR, and bit P85DDR.

CKE1	0			1	
C/ $\bar{A}$	0		1	—	
CKE0	0		1	—	—
P85DDR	0	1	—	—	—
Pin function	P85 input	P85 output	SCK3 output	SCK3 output	SCK3 input

- P83/RxD3

The pin function is switched as shown below according to the combination of bit RE in SCR of SCI\_3, and bit P83DDR.

RE	0		1
P83DDR	0	1	—
Pin function	P83 input	P83 output	RxD3 input

- P81/TxD3

The pin function is switched as shown below according to the combination of bit TE in SCR of SCI\_3, and bit P81DDR.

TE	0		1
P81DDR	0	1	—
Pin function	P81 input	P81 output	TxD3 output

Port 9 is a 2-bit input-only port. Port 9 has the following register.

- Port 9 register (PORT9)

### 9.7.1 Port 9 Register (PORT9)

PORT9 is an 8-bit read-only register that shows port 4 pin states.

PORT9 cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	Undefined	R	Reserved If read they will return an undefined value.
5	P95	—*	R	The pin states are always read when a port 9 read is performed.
4	P94	—*	R	
3	—	Undefined	R	Reserved
2	—	Undefined	R	If read they will return an undefined value.
1	—	Undefined	R	
0	—	Undefined	R	

Note: \* Determined by the states of pins P95 and P94.

### 9.7.2 Pin Functions

Port 9 also functions as the pins for A/D converter analog input and D/A converter analog output. The correspondence between pins are as follows.

#### P95/AN13/DA3

Pin function	AN13 input
	DA3 output

#### P94/AN12/DA2

Pin function	AN12 input
	DA2 output

Port A is an 8-bit I/O port that also has other functions. The port A has the following registers.

- Port A data direction register (PADDR)
- Port A data register (PADR)
- Port A register (PORTA)
- Port A MOS zcontrol register (PAPCR)
- Port A open-drain control register (PAODR)
- Port function control register 0 (PFCR0)
- Port function control register 1 (PFCR1)



The individual bits of PADDR specify input or output for the pins of port A. PADDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DDR	0	W	<ul style="list-style-type: none"> <li>Modes 1 and 2</li> </ul>
6	PA6DDR	0	W	Pins PA4 to PA0 are address outputs.
5	PA5DDR	0	W	<p>For pins PA6 and PA5, when the corresponding A22E and A21E bits are set to 1, setting a PADDR bit to 1 makes the corresponding port A pin an address output, while clearing the bit to 0 makes the pin an input port. Clearing A22E and A21E bits to 0 makes the corresponding port A pin an I/O port, and its function can be switched with PADDR. For pin PA7, when the A23E bit is set to 1, setting the PA7DDR bit to 1 makes the pin an address output, while clearing the bit to 0 makes the pin an input port. When the CS7E bit is set to 1 while the <math>\overline{CS7}</math> bit is cleared to 0, pin PA7 functions as the <math>\overline{CS7}</math> output pin when PA7DDR is set to 1, and as an input port when the bit is cleared to 0. When the CS7E bit is cleared to 0, pin PA7 is an I/O port, and its function can be switched with PA7DDR.</p> <ul style="list-style-type: none"> <li>Modes 4 and 7 (when EXPE = 1)</li> </ul> <p>For pins PA6 to PA0, when the corresponding A22E to A16E bits are set to 1, setting a PADDR bit to 1 makes the corresponding port A pin an address output, while clearing the bit to 0 makes the pin an input port. Clearing A22E to A21E bits to 0 makes the corresponding port A pin an I/O port, and its function can be switched with PADDR. For pin PA7, when the A23E bit is set to 1, setting the PA7DDR bit to 1 makes the pin an address output, while clearing the bit to 0 makes the pin an input port. When the CS7E bit is set to 1 while the A23E bit is cleared to 0, pin PA7 functions as the <math>\overline{CS7}</math> output pin when PA7DDR is set to 1, and as an input port when the bit is cleared to 0. When the CS7E bit is cleared to 0, pin PA7 is an I/O port, and its function can be switched with PA7DDR.</p> <ul style="list-style-type: none"> <li>Mode 7 (when EXPE = 0)</li> </ul> <p>Port A is an I/O port, and its pin functions can be switched with PADDR.</p>
4	PA4DDR	0	W	
3	PA3DDR	0	W	
2	PA2DDR	0	W	
1	PA1DDR	0	W	
0	PA0DDR	0	W	

PADDR stores output data for the port A pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PA6DR	0	R/W	
5	PA5DR	0	R/W	
4	PA4DR	0	R/W	
3	PA3DR	0	R/W	
2	PA2DR	0	R/W	
1	PA1DR	0	R/W	
0	PA0DR	0	R/W	

### 9.8.3 Port A Register (PORTA)

PORTA shows port A pin states.

PORTA cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7	—*	R	If a port A read is performed while PADDR bits are set to 1, the PADDR values are read. If a port A read is performed while PADDR bits are cleared to 0, the pin states are read.
6	PA6	—*	R	
5	PA5	—*	R	
4	PA4	—*	R	
3	PA3	—*	R	
2	PA2	—*	R	
1	PA1	—*	R	
0	PA0	—*	R	

Note: \* Determined by the states of pins PA7 to PA0.

PAPCR controls the MOS input pull-up function. Bits 7 to 5 are valid in modes 1 and 2 and all the bits are valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7PCR	0	R/W	When PADDR = 0 (input port), setting the corresponding bit to 1 turns on the MOS input pull-up for that pin.
6	PA6PCR	0	R/W	
5	PA5PCR	0	R/W	
4	PA4PCR	0	R/W	
3	PA3PCR	0	R/W	
2	PA2PCR	0	R/W	
1	PA1PCR	0	R/W	
0	PA0PCR	0	R/W	

### 9.8.5 Port A Open Drain Control Register (PAODR)

PAODR specifies an output type of port A.

Bit	Bit Name	Initial Value	R/W	Description
7	PA7ODR	0	R/W	When not specified for address output, setting the corresponding bit to 1 specifies a pin output type to NMOS open-drain output, while clearing this bit to 0 specifies that to CMOS output.
6	PA6ODR	0	R/W	
5	PA5ODR	0	R/W	
4	PA4ODR	0	R/W	
3	PA3ODR	0	R/W	
2	PA2ODR	0	R/W	
1	PA1ODR	0	R/W	
0	PA0ODR	0	R/W	

PFCR0 controls the I/O port.

Bit	Bit Name	Initial Value	R/W	Description
7	CS7E	1	R/W	$\overline{\text{CS7}}$ to $\overline{\text{CS0}}$ enable
6	CS6E	1	R/W	Enable/disable corresponding $\overline{\text{CSn}}$ output.
5	CS5E	1	R/W	0: Set as I/O port.
4	CS4E	1	R/W	1: Set as $\overline{\text{CSn}}$ output pin.
3	CS3E	1	R/W	(n = 7 to 0)
2	CS2E	1	R/W	
1	CS1E	1	R/W	
0	CS0E	1	R/W	

PFCR1 performs I/O port control. Bits 7 to 5 are valid in modes 1 and 2 and all the bits are valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	A23E	1	R/W	Address 23 Enable Enables or disables output for address output 23 (A23). 0: DR output when PA7DDR = 1 1: A23 output when PA7DDR = 1
6	A22E	1	R/W	Address 22 Enable Enables or disables output for address output 22 (A22). 0: DR output when PA6DDR = 1 1: A22 output when PA6DDR = 1
5	A21E	1	R/W	Address 21 Enable Enables or disables output for address output 21 (A21). 0: DR output when PA5DDR = 1 1: A21 output when PA5DDR = 1
4	A20E	1	R/W	Address 20 Enable Enables or disables output for address output 20 (A20). 0: DR output when PA4DDR = 1 1: A20 output when PA4DDR = 1
3	A19E	1	R/W	Address 19 Enable Enables or disables output for address output 19 (A19). 0: DR output when PA3DDR = 1 1: A19 output when PA3DDR = 1
2	A18E	1	R/W	Address 18 Enable Enables or disables output for address output 18 (A18). 0: DR output when PA2DDR = 1 1: A18 output when PA2DDR = 1

Enables or disables output for address output 17 (A17).

0: DR output when PA1DDR = 1

1: A17 output when PA1DDR = 1

0	A16E	1	R/W	Address 16 Enable
Enables or disables output for address output 16 (A16).				
0: DR output when PA0DDR = 1				
1: A16 output when PA0DDR = 1				

### 9.8.8 Pin Functions

Port A pins also function as the pins for address outputs and interrupt inputs. The correspondence between the register specification and the pin functions is shown below.

- PA7/A23/ $\overline{\text{CS7}}$ / $\overline{\text{IRQ7}}$

The pin function is switched as shown below according to the operating mode, bit EXPE, bits A23E and CS7E, bits ITS7 to ITS5 in ITSr, and bit PA7DDR.

Operating mode	1, 2, 4						7								
EXPE	—						0	1							
A23E	0			1			—	0				1			
CS7E	0		1		—		—	0		1		—			
PA7DDR	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
Pin function	PA7 input	PA7 output	PA7 input	CS7 output	PA7 output	Address output	PA7 input	PA7 output	PA7 input	PA7 output	PA7 input	CS7 output	PA7 input	Address output	
$\overline{\text{IRQ7}}$ interrupt input*															

Note: \*  $\overline{\text{IRQ7}}$  input when ITS7 = 0.

A22E and A21E, bits IS6 and ITS5 in ITSr, and bit PAnDDR.

Operating mode	1, 2, 4				7					
EXPE	—				0		1			
AxxE	0		1		—		0		1	
PAnDDR	0	1	0	1	0	1	0	1	0	1
Pin function	PAn input	PAn output	PAn input	Address output	PAn input	PAn output	PAn input	PAn output	PAn input	Address output
	$\overline{\text{IRQn}}$ interrupt input*									

n = 6 or 5      xx = 22 or 21

Note: \*  $\overline{\text{IRQn}}$  input when ITS<sub>n</sub> = 0.

- PA4/A20/ $\overline{\text{IRQ4}}$

The pin function is switched as shown below according to the operating mode, bit EXPE, bit A20E and bit PA4DDR.

Operating mode	1, 2	4				7					
EXPE	—	—				0		1			
A20E	—	0		1		—		0		1	
PA4DDR	—	0	1	0	1	0	1	0	1	0	1
Pin function	Address output	PA4 input	PA4 output	PA4 input	Address output	PA4 input	PA4 output	PA4 input	PA4 output	PA4 input	Address output
	$\overline{\text{IRQ4}}$ interrupt input*										

Note: \*  $\overline{\text{IRQ4}}$  input when ITS<sub>4</sub> = 0.

Operating mode	1, 2	4				7					
EXPE	—	—				0		1			
AxxE	—	0		1		—		0		1	
PAnDDR	—	0	1	0	1	0	1	0	1	0	1
Pin function	Address output	PAn input	PAn output	PAn input	Address output	PAn input	PAn output	PAn input	PAn output	PAn input	Address output

xx = 19 to 16      n = 3 to 0

### 9.8.9 Port A MOS Input Pull-Up States

Port A has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used by pins PA7 to PA5 in modes 1 and 2 and by all pins in modes 4 and 7. MOS input pull-up can be specified as on or off on a bit-by-bit basis.

Table 9.2 summarizes the MOS input pull-up states.

**Table 9.2 MOS Input Pull-Up States (Port A)**

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
4, 7	PA7 to PA0	Off	Off	On/Off
1, 2	PA7 to PA5			On/Off
	PA4 to PA0			Off

Legend:

Off: MOS input pull-up is always off.

On/Off: On when PADDR = 0 and PAPCR = 1; otherwise off.



Port B is an 8-bit I/O port that also has other functions. The port B has the following registers.

- Port B data direction register (PBDDR)
- Port B data register (PBDR)
- Port B register (PORTB)
- Port B MOS pull-up control register (PBPCR)

### 9.9.1 Port B Data Direction Register (PBDDR)

The individual bits of PBDDR specify input or output for the pins of port B.

PBDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	<ul style="list-style-type: none"><li>• Modes 1 and 2</li></ul>
6	PB6DDR	0	W	Port B pins are address outputs regardless of the PBDDR settings.
5	PB5DDR	0	W	
4	PB4DDR	0	W	<ul style="list-style-type: none"><li>• Modes 4 and 7 (when EXPE = 1)</li></ul>
3	PB3DDR	0	W	Setting a PBDDR bit to 1 makes the corresponding port B pin an address output, while clearing the bit to 0 makes the pin an input port.
2	PB2DDR	0	W	
1	PB1DDR	0	W	<ul style="list-style-type: none"><li>• Mode 7 (when EXPE = 0)</li></ul>
0	PB0DDR	0	W	Port B is an I/O port, and its pin functions can be switched with PBDDR.

PBDR is stores output data for the port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

### 9.9.3 Port B Register (PORTB)

PORTB shows port B pin states. PORTB cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7	—*	R	If this register is read is while PBDDR bits are set to 1, the PBDR values are read. If a port B read is performed while PBDDR bits are cleared to 0, the pin states are read.
6	PB6	—*	R	
5	PB5	—*	R	
4	PB4	—*	R	
3	PB3	—*	R	
2	PB2	—*	R	
1	PB1	—*	R	
0	PB0	—*	R	

Note: \* Determined by the states of pins PB7 to PB0.

PBPCR controls the on/off state of MOS input pull-up of port B. PBPCR is valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7PCR	0	R/W	When PBDDR = 0 (input port), setting the corresponding bit to 1 turns on the MOS input pull-up for that pin.
6	PB6PCR	0	R/W	
5	PB5PCR	0	R/W	
4	PB4PCR	0	R/W	
3	PB3PCR	0	R/W	
2	PB2PCR	0	R/W	
1	PB1PCR	0	R/W	
0	PB0PCR	0	R/W	

### 9.9.5 Pin Functions

Port B pins also function as the pins for address outputs. The correspondence between the register specification and the pin functions is shown below.

- PB7/A15, PB6/A14, PB5/A13, PB4/A12, PB3/A11, PB2/A10, PB1/A9, PB0/A8

The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PBDDR.

Operating mode	1, 2	4		7			
EXPE	—	—		0		1	
PBnDDR	—	0	1	0	1	0	1
Pin function	Address output	PBn input	Address output	PBn input	PBn output	PBn input	Address output

Port B has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 4 and 7. MOS input pull-up can be specified as on or off on a bit-by-bit basis.

In modes 4 and 7, when a PBDDR bit is cleared to 0, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

Table 9.3 summarizes the MOS input pull-up states.

**Table 9.3 MOS Input Pull-Up States (Port B)**

<b>Mode</b>	<b>Reset</b>	<b>Hardware Standby Mode</b>	<b>Software Standby Mode</b>	<b>In Other Operations</b>
1, 2	Off	Off	Off	Off
4, 7			On/Off	On/Off

Legend:

Off: MOS input pull-up is always off.

On/Off: On when PBDDR = 0 and PBPCR = 1; otherwise off.

Port C is an 8-bit I/O port that also has other functions. The port C has the following registers.

- Port C data direction register (PCDDR)
- Port C data register (PCDR)
- Port C register (PORTC)
- Port C MOS pull-up control register (PCPCR)

### 9.10.1 Port C Data Direction Register (PCDDR)

The individual bits of PCDDR specify input or output for the pins of port C.

PCDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DDR	0	W	<ul style="list-style-type: none"><li>• Modes 1 and 2</li></ul>
6	PC6DDR	0	W	Port C pins are address outputs regardless of the PCDDR settings.
5	PC5DDR	0	W	
4	PC4DDR	0	W	<ul style="list-style-type: none"><li>• Modes 4 and 7 (when EXPE = 1)</li></ul>
3	PC3DDR	0	W	Setting a PCDDR bit to 1 makes the corresponding port C pin an address output, while clearing the bit to 0 makes the pin an input port.
2	PC2DDR	0	W	
1	PC1DDR	0	W	<ul style="list-style-type: none"><li>• Mode 7 (when EXPE = 0)</li></ul>
0	PC0DDR	0	W	Port C is an I/O port, and its pin functions can be switched with PCDDR.

PCDR stores output data for the port C pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PC6DR	0	R/W	
5	PC5DR	0	R/W	
4	PC4DR	0	R/W	
3	PC3DR	0	R/W	
2	PC2DR	0	R/W	
1	PC1DR	0	R/W	
0	PC0DR	0	R/W	

### 9.10.3 Port C Register (PORTC)

PORTC is shows port C pin states.

PORTC cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7	—*	R	If a port C read is performed while PCDDR bits are set to 1, the PCDR values are read. If a port C read is performed while PCDDR bits are cleared to 0, the pin states are read.
6	PC6	—*	R	
5	PC5	—*	R	
4	PC4	—*	R	
3	PC3	—*	R	
2	PC2	—*	R	
1	PC1	—*	R	
0	PC0	—*	R	

Note: \* Determined by the states of pins PC7 to PC0.

PCPCR controls the on/off state of MOS input pull-up of port C. PCPCR is valid in modes 4 and 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PC7PCR	0	R/W	When PCDDR = 0 (input port), setting the corresponding bit to 1 turns on the MOS input pull-up for that pin.
6	PC6PCR	0	R/W	
5	PC5PCR	0	R/W	
4	PC4PCR	0	R/W	
3	PC3PCR	0	R/W	
2	PC2PCR	0	R/W	
1	PC1PCR	0	R/W	
0	PC0PCR	0	R/W	

### 9.10.5 Pin Functions

Port C pins also function as the pins for address outputs. The correspondence between the register specification and the pin functions is shown below.

- PC7/A7, PC6/A6, PC5/A5, PC4/A4, PC3/A3, PC2/A2, PC1/A1, PC0/A0

The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PCDDR.

Operating mode	1, 2	4		7			
EXPE	—	—		0		1	
PCnDDR	—	0	1	0	1	0	1
Pin function	Address output	PCn input	Address output	PCn input	PCn output	PCn input	Address output

Legend: n = 7 to 0

Port C has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 4 and 7. MOS input pull-up can be specified as on or off on a bit-by-bit basis.

In modes 4 and 7, when a PCDDR bit is cleared to 0, setting the corresponding PCPCR bit to 1 turns on the MOS input pull-up for that pin.

Table 9.4 summarizes the MOS input pull-up states.

**Table 9.4 MOS Input Pull-Up States (Port C)**

<b>Mode</b>	<b>Reset</b>	<b>Hardware Standby Mode</b>	<b>Software Standby Mode</b>	<b>In Other Operations</b>
1, 2	Off	Off	Off	Off
4, 7			On/Off	On/Off

Legend:

Off: MOS input pull-up is always off.

On/Off: On when PCDDR = 0 and PCPCR = 1; otherwise off.



Port D is an 8-bit I/O port that also has other functions. The port D has the following registers.

- Port D data direction register (PDDDR)
- Port D data register (PDDR)
- Port D register (PORTD)
- Port D MOS pull-up control register (PDPCR)

### 9.11.1 Port D Data Direction Register (PDDDR)

The individual bits of PDDDR specify input or output for the pins of port D.

PDDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DDR	0	W	• Modes 1, 2, 4, and 7 (when EXPE = 1)
6	PD6DDR	0	W	Port D is automatically designated for data input/output.
5	PD5DDR	0	W	
4	PD4DDR	0	W	• Mode 7 (when EXPE = 0)
3	PD3DDR	0	W	Port D is an I/O port, and its pin functions can be switched with PDDDR.
2	PD2DDR	0	W	
1	PD1DDR	0	W	
0	PD0DDR	0	W	

PDDR stores output data for the port D pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PD6DR	0	R/W	
5	PD5DR	0	R/W	
4	PD4DR	0	R/W	
3	PD3DR	0	R/W	
2	PD2DR	0	R/W	
1	PD1DR	0	R/W	
0	PD0DR	0	R/W	

### 9.11.3 Port D Register (PORTD)

PORTD shows port D pin states.

PORTD cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7	—*	R	If a port D read is performed while PDDDR bits are set to 1, the PDDR values are read. If a port D read is performed while PDDDR bits are cleared to 0, the pin states are read.
6	PD6	—*	R	
5	PD5	—*	R	
4	PD4	—*	R	
3	PD3	—*	R	
2	PD2	—*	R	
1	PD1	—*	R	
0	PD0	—*	R	

Note: \* Determined by the states of pins PD7 to PD0.

PDPCR controls on/off states of the input pull-up MOS of port D. PDPCR is valid in mode 7.

Bit	Bit Name	Initial Value	R/W	Description
7	PD7PCR	0	R/W	When PDDDR = 0 (input port), the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
6	PD6PCR	0	R/W	
5	PD5PCR	0	R/W	
4	PD4PCR	0	R/W	
3	PD3PCR	0	R/W	
2	PD2PCR	0	R/W	
1	PD1PCR	0	R/W	
0	PD0PCR	0	R/W	

### 9.11.5 Pin Functions

Port D pins also function as the pins for data I/Os. The correspondence between the register specification and the pin functions is shown below.

- PD7/D15, PD6/D14, PD5/D13, PD4/D12, PD3/D11, PD2/D10, PD1/D9, PD0/D8

The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PDDDR.

Operating mode	1, 2, 4	7		
EXPE	—	0		1
PDnDDR	—	0	1	—
Pin function	Data I/O	PDn input	PDn output	Data I/O

Legend: n = 7 to 0

Port D has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in mode 7. MOS input pull-up can be specified as on or off on a bit-by-bit basis.

In mode 7, when a PDDDR bit is cleared to 0, setting the corresponding PDPCR bit to 1 turns on the MOS input pull-up for that pin.

Table 9.5 summarizes the MOS input pull-up states.

**Table 9.5 MOS Input Pull-Up States (Port D)**

<b>Mode</b>	<b>Reset</b>	<b>Hardware Standby Mode</b>	<b>Software Standby Mode</b>	<b>In Other Operations</b>
1, 2, 4	Off	Off	Off	Off
7			On/Off	On/Off

Legend:

OFF: MOS input pull-up is always off.

On/Off: On when PDDDR = 0 and PDPCR = 1; otherwise off.

Port E is an 8-bit I/O port that also has other functions. The port E has the following registers.

- Port E data direction register (PEDDDR)
- Port E data register (PEDR)
- Port E register (PORTE)
- Port E MOS pull-up control register (PEPCR)

### 9.12.1 Port E Data Direction Register (PEDDDR)

The individual bits of PEDDDR specify input or output for the pins of port E.

PEDDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	<ul style="list-style-type: none"> <li>• Modes 1, 2, and 4</li> </ul>
6	PE6DDR	0	W	When 8-bit bus mode is selected, port E functions as an I/O port. The pin states can be changed with PEDDDR.
5	PE5DDR	0	W	
4	PE4DDR	0	W	When 16-bit bus mode is selected, port E is designated for data input/output.
3	PE3DDR	0	W	
2	PE2DDR	0	W	For details on 8-bit and 16-bit bus modes, see section 6, Bus Controller.
1	PE1DDR	0	W	
0	PE0DDR	0	W	<ul style="list-style-type: none"> <li>• Mode 7 (when EXPE = 1) When 8-bit bus mode is selected, port E functions as an I/O port. Setting a PEDDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port. When 16-bit bus mode is selected, port E is designated for data input/output.</li> <li>• Mode 7 (when EXPE = 0) Port E is an I/O port, and its pin functions can be switched with PEDDDR.</li> </ul>

PEDR stores output data for the port E pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PE6DR	0	R/W	
5	PE5DR	0	R/W	
4	PE4DR	0	R/W	
3	PE3DR	0	R/W	
2	PE2DR	0	R/W	
1	PE1DR	0	R/W	
0	PE0DR	0	R/W	

### 9.12.3 Port E Register (PORTE)

PORTE shows port E pin states.

PORTE cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7	—*	R	If a port E read is performed while PEDDR bits are set to 1, the PEDR values are read. If a port E read is performed while PEDDR bits are cleared to 0, the pin states are read.
6	PE6	—*	R	
5	PE5	—*	R	
4	PE4	—*	R	
3	PE3	—*	R	
2	PE2	—*	R	
1	PE1	—*	R	
0	PE0	—*	R	

Note: \* Determined by the states of pins PE7 to PE0.

PEPCR controls on/off states of the input pull-up MOS of port E. PEPCR is valid in 8-bit bus mode.

Bit	Bit Name	Initial Value	R/W	Description
7	PE7PCR	0	R/W	When PEDDR = 0 (input port), the input pull-up MOS of the input pin is on when the corresponding bit is set to 1.
6	PE6PCR	0	R/W	
5	PE5PCR	0	R/W	
4	PE4PCR	0	R/W	
3	PE3PCR	0	R/W	
2	PE2PCR	0	R/W	
1	PE1PCR	0	R/W	
0	PE0PCR	0	R/W	

### 9.12.5 Pin Functions

Port E pins also function as the pins for data I/Os. The correspondence between the register specification and the pin functions is shown below.

- PE7/D7, PE6/D6, PE5/D5, PE4/D4, PE3/D3, PE2/D2, PE1/D1, PE0/D0

The pin function is switched as shown below according to the operating mode, bus mode, bit EXPE, and bit PEDDR.

Operating mode	1, 2, 4			7				
	Bus mode	All areas 8-bit space		At least one area 16-bit space	—		All areas 8-bit space	
EXPE	—		—	0		1		1
PE <sub>n</sub> DDR	0	1	—	0	1	0	1	—
Pin function	PE <sub>n</sub> input	PE <sub>n</sub> output	Data I/O	PE <sub>n</sub> input	PE <sub>n</sub> output	PE <sub>n</sub> input	PE <sub>n</sub> output	Data I/O

Legend: n = 7 to 0

Port E has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in 8-bit bus mode. MOS input pull-up can be specified as on or off on a bit-by-bit basis. In 8-bit bus mode, when a PEDDR bit is cleared to 0, setting the corresponding PEPCR bit to 1 turns on the MOS input pull-up for that pin.

Table 9.6 summarizes the MOS input pull-up states.

**Table 9.6 MOS Input Pull-Up States (Port E)**

Mode		Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, and 4	8-bit bus	Off	Off	On/Off	On/Off
	16-bit bus			Off	Off

Legend:

Off: MOS input pull-up is always off.

On/Off: On when PEDDR = 0 and PEPCR = 1; otherwise off.

## 9.13 Port F

Port F is an 8-bit I/O port that also has other functions. The port F has the following registers. For details on the port function control register 0, refer to section 9.8.6, Port Function Control Register 0 (PFCR0), and for details on the port function control register 2, refer to section 9.3.5, Port Function Control Register 2 (PFCR2).

- Port F data direction register (PFDDR)
- Port F data register (PFDR)
- Port F register (PORTF)
- Port Function Control Register 0 (PFCR0)
- Port Function Control Register 2 (PFCR2)



The individual bits of PFDDR specify input or output for the pins of port F. PFDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DDR	1/0*	W	<ul style="list-style-type: none"> <li>Modes 1, 2, 4, and 7 (when EXPE = 1)</li> </ul>
6	PF6DDR	0	W	Pin PF7 functions as the $\phi$ output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0.
5	PF5DDR	0	W	
4	PF4DDR	0	W	Pin PF6 functions as the $\overline{AS}$ output pin when ASOE is set to 1. When ASOE is cleared to 0, pin PF6 is an I/O port and its function can be switched with PF6DDR.
3	PF3DDR	0	W	
2	PF2DDR	0	W	
1	PF1DDR	0	W	Pins PF5 and PF4 are automatically designated as bus control outputs ( $\overline{RD}$ and $\overline{HWR}$ ).
0	PF0DDR	0	W	<p>Pin PF3 functions as the <math>\overline{LWR}</math> output pin when LWROE is set to 1. When LWROE is cleared to 0, pin PF3 is an I/O port and its function can be switched with PF3DDR.</p> <p>Pins PF2 and PF1 function as bus control output pins (<math>\overline{LCAS}</math> and <math>\overline{UCAS}</math>) when the appropriate bus controller settings are made. When the <math>\overline{CS}</math> output enable bits (<math>\overline{CS6E}</math> and <math>\overline{CS5E}</math>) are set to 1, they function as <math>\overline{CS}</math> outputs. When the <math>\overline{CS}</math> output enable bits (<math>\overline{CS6E}</math> and <math>\overline{CS5E}</math>) are cleared to 0, pins PF2 and PF1 are I/O ports and their functions can be switched with the corresponding PFDDR bits.</p> <p>Pin PF0 functions as a bus control input pin (<math>\overline{WAIT}</math>) when the appropriate bus controller settings are made. Otherwise, this pin is an output port when the corresponding PFDDR bit is set to 1, and an input port when the bit is cleared to 0.</p> <ul style="list-style-type: none"> <li>Mode 7 (when EXPE = 0)</li> </ul> <p>Pin PF7 functions as the <math>\phi</math> output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0.</p> <p>Pins PF6 to PF0 are I/O ports, and their functions can be switched with PFDDR.</p>

Note: \* PF7DDR is initialized to 1 in modes 1, 2, and 4, and to 0 in mode 7.

PFDR stores output data for the port F pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7DR	0	R/W	Output data for a pin is stored when the pin function is specified to a general purpose I/O.
6	PF6DR	0	R/W	
5	PF5DR	0	R/W	
4	PF4DR	0	R/W	
3	PF3DR	0	R/W	
2	PF2DR	0	R/W	
1	PF1DR	0	R/W	
0	PF0DR	0	R/W	

### 9.13.3 Port F Register (PORTF)

PORTF shows port F pin states.

PORTF cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	PF7	—*	R	If a port F read is performed while PFDDR bits are set to 1, the PFDR values are read. If a port F read is performed while PFDDR bits are cleared to 0, the pin states are read.
6	PF6	—*	R	
5	PF5	—*	R	
4	PF4	—*	R	
3	PF3	—*	R	
2	PF2	—*	R	
1	PF1	—*	R	
0	PF0	—*	R	

Note: \* Determined by the states of pins PF7 to PF0.

Port F pins also function as the pins for external interrupt inputs, bus control signal I/Os, and system clock outputs ( $\phi$ ). The correspondence between the register specification and the pin functions is shown below.

- PF7/ $\phi$

The pin function is switched as shown below according to bit PF7DDR.

Operating mode	1, 2, 4, 7	
PF7DDR	0	1
Pin function	PF7 input	$\phi$ output

- PF6/ $\overline{AS}$

The pin function is switched as shown below according to the operating mode, bit EXPE, bit ASOE, and bit PF6DDR.

Operating mode	1, 2, 4			7				
EXPE	—			0		1		
ASOE	1	0		—		1	0	
PF6DDR	—	0	1	0	1	—	0	1
Pin function	$\overline{AS}$ output	PF6 input	PF6 output	PF6 input	PF6 output	$\overline{AS}$ output	PF6 input	PF6 output

- PF5/ $\overline{RD}$

The pin function is switched as shown below according to the operating mode, bit EXPE, and bit PF5DDR.

Operating mode	1, 2, 4		7		
EXPE	—		0		1
PF5DDR	—		0	1	—
Pin function	$\overline{RD}$ output		PF5 input	PF5 output	$\overline{RD}$ output

PF4DDR.

Operating mode	1, 2, 4	7		
EXPE	—	0		1
PF4DDR	—	0	1	—
Pin function	$\overline{\text{HWR}}$ output	PF4 input	PF4 output	$\overline{\text{HWR}}$ output

- PF3/ $\overline{\text{LWR}}$

The pin function is switched as shown below according to the operating mode, bit EXPE, bit LWROE, and bit PF3DDR.

Operating mode	1, 2, 4			7				
EXPE	—			0		1		
LWROD	1	0		—		1	0	
PF3DDR	—	0	1	0	1	—	0	1
Pin function	$\overline{\text{LWR}}$ output	PF3 input	PF3 output	PF3 input	PF3 output	$\overline{\text{LWR}}$ output	PF3 input	PF3 output

bit EXPE, bits RMTS2 to RMTS0 in DRAMCR, bits ABW5 to ABW2 in ABWCR, and bit PF2DDR.

Operating mode	1, 2, 4						3, 7					
EXPE	—						0		1			
Areas 2 to 5	Any DRAM space area is 16-bit bus space	All DRAM space areas are 8-bit bus space, or areas 2 to 5 are all normal space				—		Any DRAM space area is 16-bit bus space	All DRAM space areas are 8-bit bus space, or areas 2 to 5 are all normal space			
CS6E	—	1		0		—		—	1		0	
PF2DDR	—	0	1	0	1	0	1	—	0	1	0	1
Pin function	$\overline{\text{LCAS}}$ output	PF2 input	$\overline{\text{CS6}}$ output	PF2 input	PF2 output	PF2 input	PF2 output	$\overline{\text{LCAS}}$ output	PF2 input	$\overline{\text{CS6}}$ output	PF2 input	PF2 output

- PF1/ $\overline{\text{CS5}}$ / $\overline{\text{UCAS}}$

The pin function is switched as shown below according to the combination of the operating mode, bit EXPE, bits RMTS2 to RMTS0 in DRAMCR, and bit PF1DDR.

Operating mode	1, 2, 4						3, 7					
EXPE	—						0		1			
Areas 2 to 5	Any DRAM space area is 16-bit bus space	All DRAM space areas are 8-bit bus space, or areas 2 to 5 are all normal space				—		Any DRAM space area is 16-bit bus space	All DRAM space areas are 8-bit bus space, or areas 2 to 5 are all normal space			
CS6E	—	1		0		—		—	1		0	
PF1DDR	—	0	1	0	1	0	1	—	0	1	0	1
Pin function	$\overline{\text{UCAS}}$ output	PF1 input	$\overline{\text{CS5}}$ output	PF1 input	PF1 output	PF1 input	PF1 output	$\overline{\text{UCAS}}$ output	PF1 input	$\overline{\text{CS5}}$ output	PF1 input	PF1 output

WAITE, bit OEE in DRAMCR, bit OES in PFCR2, and bit PF0DDR.

Operating mode	1, 2, 4						7							
EXPE	—						0		1					
OEE	0		1				—		0		1			
OES	—		0		1		—		—		0		1	
WAITE	—		0		1		—		—		0		1	
PF0DDR	0	1	0	1	—	—	0	1	0	1	0	1	—	—
Pin function	PF0 input	PF0 output	PF0 input	PF0 output	WAIT input	OE output	PF0 input	PF0 output	PF0 input	PF0 output	PF0 input	PF0 output	WAIT input	OE output

Port G is a 7-bit I/O port that also has other functions. The port G has the following registers.

- Port G data direction register (PGDDR)
- Port G data register (PGDR)
- Port G register (PORTG)
- Port Function Control Register 0 (PFCR0)

### 9.14.1 Port G Data Direction Register (PGDDR)

The individual bits of PGDDR specify input or output for the pins of port G.

PGDDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved
6	PG6DDR	0	W	<ul style="list-style-type: none"> <li>• Modes 1, 2, 4, and 7 (when EXPE = 1) Pins PG6 and PG5 function as bus control input/output pins (<math>\overline{\text{BREQ}}</math> and <math>\overline{\text{BACK}}</math>) when the appropriate bus controller settings are made. Otherwise, these pins are I/O ports, and their functions can be switched with PGDDR. Pin PG4 functions as the bus control input/output pin (<math>\overline{\text{BREQO}}</math>) when the appropriate bus controller settings are made. Otherwise, when the CS7E bit is set to 1, pin PG4 functions as the CS7 output pin when PG4DDR is set to 1, and as an input port when the bit is cleared to 0. When the CS7E bit is cleared to 0, pin PG4 is an I/O port, and its function can be switched with PG4DDR. When the <math>\overline{\text{CS}}</math> output enable bits (CS3E to CS0E) are set to 1, pins PG3 to PG0 function as <math>\overline{\text{CS}}</math> output pins when the corresponding PGDDR bit is set to 1, and as input ports when the bit is cleared to 0. When CS3E to CS0E are cleared to 0, pins PG3 to PG0 are I/O ports, and their functions can be switched with PGDDR.</li> <li>• Mode 7 (when EXPE = 0) Pins PG6 to PG0 are I/O ports, and their functions can be switched with PGDDR.</li> </ul>
5	PG5DDR	0	W	
4	PG4DDR	0	W	
3	PG3DDR	0	W	
2	PG2DDR	0	W	
1	PG1DDR	0	W	
0	PG0DDR	1/0*	W	

Note: \* PG0DDR is initialized to 1 in modes 1 and 2, and to 0 in modes 4 and 7.

PGDR stores output data for the port G pins.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	—	Reserved This bit is always read as 0, and cannot be modified.
6	PG6DR	0	R/W	An output data for a pin is stored when the pin function is specified to a general purpose I/O.
5	PG5DR	0	R/W	
4	PG4DR	0	R/W	
3	PG3DR	0	R/W	
2	PG2DR	0	R/W	
1	PG1DR	0	R/W	
0	PG0DR	0	R/W	

### 9.14.3 Port G Register (PORTG)

PORTG shows port G pin states.

PORTG cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	—	Undefined	—	Reserved If this bit is read, it will return an undefined value.
6	PG6	—*	R	If a port G read is performed while PGDDR bits are set to 1, the PGDR values are read. If a port G read is performed while PGDDR bits are cleared to 0, the pin states are read.
5	PG5	—*	R	
4	PG4	—*	R	
3	PG3	—*	R	
2	PG2	—*	R	
1	PG1	—*	R	
0	PG0	—*	R	

Note: \* Determined by the states of pins PG6 to PG0.



Port G pins also function as the pins for bus control signal I/Os. The correspondence between the register specification and the pin functions is shown below.

- PG6/ $\overline{\text{BREQ}}$

The pin function is switched as shown below according to the operating mode, bit EXPE, bit BRLE, and bit PG6DDR.

Operating mode	1, 2, 4			7				
EXPE	—			0		1		
BRLE	0		1	—		0		1
PG6DDR	0	1	—	0	1	0	1	—
Pin function	PG6 input	PG6 output	$\overline{\text{BREQ}}$ input	PG6 input	PG6 output	PG6 input	PG6 output	$\overline{\text{BREQ}}$ input

- PG5/ $\overline{\text{BACK}}$

The pin function is switched as shown below according to the operating mode, bit EXPE, bit BRLE, and bit PG5DDR.

Operating mode	1, 2, 4			7				
EXPE	—			0		1		
BRLE	0		1	—		0		1
PG5DDR	0	1	—	0	1	0	1	—
Pin function	PG5 input	PG5 output	$\overline{\text{BACK}}$ output	PG5 input	PG5 output	PG5 input	PG5 output	$\overline{\text{BACK}}$ output

BRLE, bit BREQO, bit CS4E and bit PG4DDR.

Operating mode	1, 2, 4										7											
EXPE	-										0	1										
BRLE	0					1					-	0					1					
BREQOE	-					0					1	-					0					1
CS4E	0		1		0		1		-	-		0		1		0		1		1		
PG4DDR	0	1	0	1	0	1	0	1	-	0	1	0	1	0	1	0	1	0	1	-		
Pin function	PG4 input	PG4 output	PG4 input	CS4 output	PG4 input	PG4 output	PG4 input	CS4 output	BREQO output	PG4 input	PG4 output	PG4 input	PG4 output	PG4 input	CS4 output	PG4 input	PG4 output	PG4 input	CS4 output	BREQO output		

- $\overline{\text{PG3}}/\overline{\text{CS3}}/\overline{\text{RAS3}}$ ,  $\overline{\text{PG2}}/\overline{\text{CS2}}/\overline{\text{RAS2}}$

The pin function is switched as shown below according to the operating mode, bit EXPE, bit PGnDDR, bit CSnE, and bits RMTS2 to RMTS0.

Operating mode	1, 2, 4										7																			
EXPE	-										0	1																		
CSnE	0					1					-	0					1													
RMTS2 to RMTS0	-					Area n is in normal space					Area n is in DRAM space					-					Area n is in normal space					Area n is in DRAM space				
PGnDDR	0	1	0	1	0	1	0	1	-	0	1	0	1	0	1	0	1	0	1	-										
Pin function	PGn input	PGn output	PGn input	$\overline{\text{CSn}}$ output	$\overline{\text{RASn}}$ output	PGn input	PGn output	PGn input	PGn output	PGn input	PGn output	PGn input	PGn output	PGn input	$\overline{\text{CSn}}$ output	$\overline{\text{RASn}}$ output														

Legend: n = 2 or 3

PGnDDR, and bit CsnE.

Operating mode	1, 2, 4				7					
EXPE	—				0		1			
CSnE	0		1		—		0		1	
PGnDDR	0	1	0	1	0	1	0	1	0	1
Pin function	PGn input	PGn output	PGn input	$\overline{\text{CSn}}$ output	PGn input	PGn output	PGn input	PGn output	PGn input	$\overline{\text{CSn}}$ output

Legend: n = 1 or 0



This LSI has an on-chip 16-bit timer pulse unit (TPU) that comprises six 16-bit timer channels. The function list of the 16-bit timer unit and its block diagram are shown in table 10.1 and figure 10.1, respectively.

## 10.1 Features

- Maximum 16-pulse input/output
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output at compare match
  - Input capture function
  - Counter clear operation
  - Synchronous operations:
    - Multiple timer counters (TCNT) can be written to simultaneously
    - Simultaneous clearing by compare match and input capture possible
    - Register simultaneous input/output possible by counter synchronous operation
  - Maximum of 15-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channels 0 and 3
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
- Cascaded operation
- Fast access via internal 16-bit bus
- 26 interrupt sources
- Automatic transfer of register data
- Programmable pulse generator (PPG) output trigger can be generated
- A/D converter conversion start trigger can be generated
- Module stop mode can be set

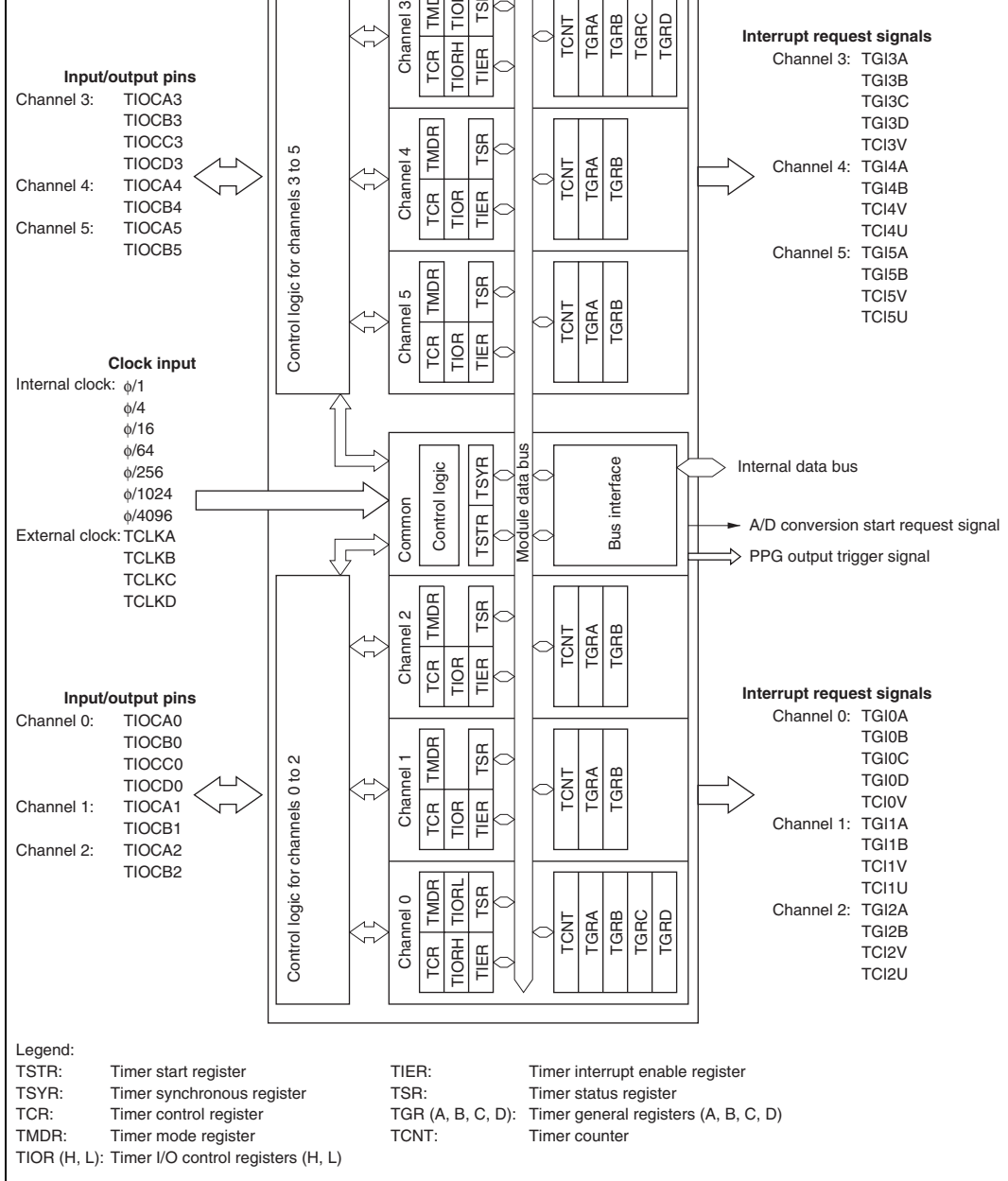
Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock	$\phi/1$	$\phi/1$	$\phi/1$	$\phi/1$	$\phi/1$	$\phi/1$
	$\phi/4$	$\phi/4$	$\phi/4$	$\phi/4$	$\phi/4$	$\phi/4$
	$\phi/16$	$\phi/16$	$\phi/16$	$\phi/16$	$\phi/16$	$\phi/16$
	$\phi/64$	$\phi/64$	$\phi/64$	$\phi/64$	$\phi/64$	$\phi/64$
	TCLKA	$\phi/256$	$\phi/1024$	$\phi/256$	$\phi/1024$	$\phi/256$
	TCLKB	TCLKA	TCLKA	$\phi/1024$	TCLKA	TCLKA
	TCLKC	TCLKB	TCLKB	$\phi/4096$	TCLKC	TCLKC
TCLKD		TCLKC	TCLKA		TCLKD	
General registers (TGR)	TGRA_0	TGRA_1	TGRA_2	TGRA_3	TGRA_4	TGRA_5
	TGRB_0	TGRB_1	TGRB_2	TGRB_3	TGRB_4	TGRB_5
General registers/ buffer registers	TGRC_0	—	—	TGRC_3	—	—
	TGRD_0			TGRD_3		
I/O pins	TIOCA0	TIOCA1	TIOCA2	TIOCA3	TIOCA4	TIOCA5
	TIOCB0	TIOCB1	TIOCB2	TIOCB3	TIOCB4	TIOCB5
	TIOCC0			TIOCC3		
	TIOCD0			TIOCD3		
Counter clear function	TGR	TGR	TGR	TGR	TGR	TGR
	compare match or input capture	compare match or input capture	compare match or input capture	compare match or input capture	compare match or input capture	compare match or input capture
Compare match output	0 output	0	0	0	0	0
	1 output	0	0	0	0	0
	Toggle output	0	0	0	0	0
Input capture function	0	0	0	0	0	0
Synchronous operation	0	0	0	0	0	0
PWM mode	0	0	0	0	0	0
Phase counting mode	—	0	0	—	0	0
Buffer operation	0	—	—	0	—	—

activation	compare match or input capture	compare match or input capture	compare match or input capture	compare match or input capture	compare match or input capture	compare match or input capture
DMAC activation	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
A/D converter trigger	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
PPG trigger	TGRA/ TGRB compare match or input capture	TGRA/ TGRB compare match or input capture	TGRA/ TGRB compare match or input capture	TGRA/ TGRB compare match or input capture	—	—
Interrupt sources	5 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 0A</li> <li>• Compare match or input capture 0B</li> <li>• Compare match or input capture 0C</li> <li>• Compare match or input capture 0D</li> <li>• Overflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 1A</li> <li>• Compare match or input capture 1B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 2A</li> <li>• Compare match or input capture 2B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	5 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 3A</li> <li>• Compare match or input capture 3B</li> <li>• Compare match or input capture 3C</li> <li>• Compare match or input capture 3D</li> <li>• Overflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 4A</li> <li>• Compare match or input capture 4B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 5A</li> <li>• Compare match or input capture 5B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>

Legend:

O: Possible

—: Not possible



**Figure 10.1 Block Diagram of TPU**



**Table 10.2 Pin Configuration**

<b>Channel</b>	<b>Symbol</b>	<b>I/O</b>	<b>Function</b>
All	TCLKA	Input	External clock A input pin (Channels 1 and 5 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (Channels 1 and 5 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (Channels 2 and 4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (Channels 2 and 4 phase counting mode B phase input)
0	TIOCA0	I/O	TGRA_0 input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TGRB_0 input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TGRC_0 input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TGRD_0 input capture input/output compare output/PWM output pin
1	TIOCA1	I/O	TGRA_1 input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TGRB_1 input capture input/output compare output/PWM output pin
2	TIOCA2	I/O	TGRA_2 input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TGRB_2 input capture input/output compare output/PWM output pin
3	TIOCA3	I/O	TGRA_3 input capture input/output compare output/PWM output pin
	TIOCB3	I/O	TGRB_3 input capture input/output compare output/PWM output pin
	TIOCC3	I/O	TGRC_3 input capture input/output compare output/PWM output pin
	TIOCD3	I/O	TGRD_3 input capture input/output compare output/PWM output pin
4	TIOCA4	I/O	TGRA_4 input capture input/output compare output/PWM output pin
	TIOCB4	I/O	TGRB_4 input capture input/output compare output/PWM output pin
5	TIOCA5	I/O	TGRA_5 input capture input/output compare output/PWM output pin
	TIOCB5	I/O	TGRB_5 input capture input/output compare output/PWM output pin

The TPU has the following registers in each channel.

- Timer control register\_0 (TCR\_0)
- Timer mode register\_0 (TMDR\_0)
- Timer I/O control register H\_0 (TIORH\_0)
- Timer I/O control register L\_0 (TIORL\_0)
- Timer interrupt enable register\_0 (TIER\_0)
- Timer status register\_0 (TSR\_0)
- Timer counter\_0 (TCNT\_0)
- Timer general register A\_0 (TGRA\_0)
- Timer general register B\_0 (TGRB\_0)
- Timer general register C\_0 (TGRC\_0)
- Timer general register D\_0 (TGRD\_0)
- Timer control register\_1 (TCR\_1)
- Timer mode register\_1 (TMDR\_1)
- Timer I/O control register \_1 (TIOR\_1)
- Timer interrupt enable register\_1 (TIER\_1)
- Timer status register\_1 (TSR\_1)
- Timer counter\_1 (TCNT\_1)
- Timer general register A\_1 (TGRA\_1)
- Timer general register B\_1 (TGRB\_1)
- Timer control register\_2 (TCR\_2)
- Timer mode register\_2 (TMDR\_2)
- Timer I/O control register\_2 (TIOR\_2)
- Timer interrupt enable register\_2 (TIER\_2)
- Timer status register\_2 (TSR\_2)
- Timer counter\_2 (TCNT\_2)
- Timer general register A\_2 (TGRA\_2)
- Timer general register B\_2 (TGRB\_2)
- Timer control register\_3 (TCR\_3)
- Timer mode register\_3 (TMDR\_3)
- Timer I/O control register H\_3 (TIORH\_3)
- Timer I/O control register L\_3 (TIORL\_3)

- Timer counter\_3 (TCNT\_3)
- Timer general register A\_3 (TGRA\_3)
- Timer general register B\_3 (TGRB\_3)
- Timer general register C\_3 (TGRC\_3)
- Timer general register D\_3 (TGRD\_3)
- Timer control register\_4 (TCR\_4)
- Timer mode register\_4 (TMDR\_4)
- Timer I/O control register\_4 (TIOR\_4)
- Timer interrupt enable register\_4 (TIER\_4)
- Timer status register\_4 (TSR\_4)
- Timer counter\_4 (TCNT\_4)
- Timer general register A\_4 (TGRA\_4)
- Timer general register B\_4 (TGRB\_4)
- Timer control register\_5 (TCR\_5)
- Timer mode register\_5 (TMDR\_5)
- Timer I/O control register\_5 (TIOR\_5)
- Timer interrupt enable register\_5 (TIER\_5)
- Timer status register\_5 (TSR\_5)
- Timer counter\_5 (TCNT\_5)
- Timer general register A\_5 (TGRA\_5)
- Timer general register B\_5 (TGRB\_5)

#### Common Registers

- Timer start register (TSTR)
- Timer synchronous register (TSYR)

The TCR registers control the TCNT operation for each channel. The TPU has a total of six TCR registers, one for each channel. TCR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial Value	R/W	Description
7	CCLR2	0	R/W	Counter Clear 2 to 0
6	CCLR1	0	R/W	These bits select the TCNT counter clearing source. See tables 10.3 and 10.4 for details.
5	CCLR0	0	R/W	
4	CKEG1	0	R/W	Clock Edge 1 and 0
3	CKEG0	0	R/W	<p>These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. <math>\phi/4</math> both edges = <math>\phi/2</math> rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is <math>\phi/4</math> or slower. This setting is ignored if the input clock is <math>\phi/1</math>, or when overflow/underflow of another channel is selected.</p> <p>00: Count at rising edge            01: Count at falling edge            1x: Count at both edges</p> <p>Legend: x: Don't care</p>
2	TPSC2	0	R/W	Time Prescaler 2 to 0
1	TPSC1	0	R/W	These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See tables 10.5 to 10.10 for details.
0	TPSC0	0	R/W	

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
			1	TCNT cleared by TGRB compare match/input capture
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture <sup>*2</sup>
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation <sup>*1</sup>

- Notes:
1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
  2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

**Table 10.4 CCLR2 to CCLR0 (Channels 1, 2, 4, and 5)**

Channel	Bit 7 Reserved <sup>*2</sup>	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation <sup>*1</sup>

- Notes:
1. Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.
  2. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
0	0	0	0	Internal clock: counts on $\phi/1$
			1	Internal clock: counts on $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	External clock: counts on TCLKD pin input

**Table 10.6 TPSC2 to TPSC0 (Channel 1)**

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on $\phi/1$
			1	Internal clock: counts on $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	Internal clock: counts on $\phi/256$
			1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
2	0	0	0	Internal clock: counts on $\phi/1$
			1	Internal clock: counts on $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKB pin input
		1	0	External clock: counts on TCLKC pin input
			1	Internal clock: counts on $\phi/1024$

Note: This setting is ignored when channel 2 is in phase counting mode.

**Table 10.8 TPSC2 to TPSC0 (Channel 3)**

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
3	0	0	0	Internal clock: counts on $\phi/1$
			1	Internal clock: counts on $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	1	0	0	External clock: counts on TCLKA pin input
			1	Internal clock: counts on $\phi/1024$
		1	0	Internal clock: counts on $\phi/256$
			1	Internal clock: counts on $\phi/4096$

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
4	0	0	0	Internal clock: counts on $\phi/1$
			1	Internal clock: counts on $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKC pin input
		1	0	Internal clock: counts on $\phi/1024$
			1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

**Table 10.10 TPSC2 to TPSC0 (Channel 5)**

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on $\phi/1$
			1	Internal clock: counts on $\phi/4$
		1	0	Internal clock: counts on $\phi/16$
			1	Internal clock: counts on $\phi/64$
	1	0	0	External clock: counts on TCLKA pin input
			1	External clock: counts on TCLKC pin input
		1	0	Internal clock: counts on $\phi/256$
			1	External clock: counts on TCLKD pin input

Note: This setting is ignored when channel 5 is in phase counting mode.



TMDR registers are used to set the operating mode for each channel. The TPU has six TMDR registers, one for each channel. TMDR register settings should be made only when TCNT operation is stopped.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
5	BFB	0	R/W	Buffer Operation B Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified. 0: TGRB operates normally 1: TGRB and TGRD used together for buffer operation
4	BFA	0	R/W	Buffer Operation A Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated. In channels 1, 2, 4, and 5, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified. 0: TGRA operates normally 1: TGRA and TGRC used together for buffer operation
3	MD3	0	R/W	Modes 3 to 0
2	MD2	0	R/W	These bits are used to set the timer operating mode.
1	MD1	0	R/W	
0	MD0	0	R/W	

Bit 3 MD3*1	Bit 2 MD2*2	Bit 1 MD1	Bit 0 MD0	Description
0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	x	x	x	—

Legend: x: Don't care

Notes: 1. MD3 is a reserved bit. The write value should always be 0.

2. Phase counting mode cannot be set for channels 0 and 3. In this case, 0 should always be written to MD2.

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

TIOR registers control the TGR registers. The TPU has eight TIOR registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. Care is required since TIOR is affected by the TMDR setting.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Bit	Bit Name	Initial Value	R/W	Description
7	IOB3	0	R/W	I/O Control B3 to B0
6	IOB2	0	R/W	Specify the function of TGRB. For details, see tables 10.12, 10.14, 10.15, 10.16, 10.18, and 10.19.
5	IOB1	0	R/W	
4	IOB0	0	R/W	
3	IOA3	0	R/W	
2	IOA2	0	R/W	Specify the function of TGRA. For details, see tables 10.20, 10.22, 10.23, 10.24, 10.26, and 10.27.
1	IOA1	0	R/W	
0	IOA0	0	R/W	

### TIORL\_0, TIORL\_3

Bit	Bit Name	Initial Value	R/W	Description
7	IOD3	0	R/W	I/O Control D3 to D0
6	IOD2	0	R/W	Specify the function of TGRD. For details, see tables 10.13, and 10.17.
5	IOD1	0	R/W	
4	IOD0	0	R/W	
3	IOC3	0	R/W	
2	IOC2	0	R/W	Specify the function of TGRC. For details, see tables 10.21, and 10.25
1	IOC1	0	R/W	
0	IOC0	0	R/W	

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_0 Function	TIOCB0 Pin Function	
0	0	0	0	Output compare register	Output disabled	
			1		Initial output is 0 output 0 output at compare match	
		1	0		Initial output is 0 output 1 output at compare match	
			1		Initial output is 0 output Toggle output at compare match	
	1	0	0		Output disabled	
			1		Initial output is 1 output 0 output at compare match	
		1	0		Initial output is 1 output 1 output at compare match	
			1		Initial output is 1 output Toggle output at compare match	
1	0	0	0	Input capture register	Capture input source is TIOCB0 pin Input capture at rising edge	
			1		Capture input source is TIOCB0 pin Input capture at falling edge	
		1	×		Capture input source is TIOCB0 pin Input capture at both edges	
	1	×	×		0	Capture input source is channel 1/count clock Input capture at TCNT_1 count- up/count-down*

Legend: ×: Don't care

Note: \* When bits TPSC2 to TPSC0 in TCR\_1 are set to B'000 and  $\phi/1$  is used as the TCNT\_1 count clock, this setting is invalid and input capture is not generated.

				Description		
Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	TGRD_0 Function	TIOCD0 Pin Function	
0	0	0	0	Output compare register <sup>*2</sup>	Output disabled	
			1		Initial output is 0 output 0 output at compare match	
		1	0	Initial output is 0 output 1 output at compare match		
			1	Initial output is 0 output Toggle output at compare match		
	1	0	0	Input capture register <sup>*2</sup>	Output disabled	
			1		Initial output is 1 output 0 output at compare match	
			0		Initial output is 1 output 1 output at compare match	
		1	×	0	Input capture register <sup>*2</sup>	Initial output is 1 output Toggle output at compare match
				1		Capture input source is TIOCD0 pin Input capture at rising edge
				×		Capture input source is TIOCD0 pin Input capture at falling edge
1	×	×	Input capture register <sup>*2</sup>	Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down <sup>*1</sup>		

Legend: ×: Don't care

- Notes:
1. When bits TPSC2 to TPSC0 in TCR\_1 are set to B'000 and  $\phi/1$  is used as the TCNT\_1 count clock, this setting is invalid and input capture is not generated.
  2. When the BFB bit in TMDR\_0 is set to 1 and TGRD\_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_1 Function	TIOCB1 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 output 0 output at compare match	
			0	Initial output is 1 output 1 output at compare match	
		1	0	Initial output is 1 output Toggle output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
1	0	0	0	Input capture register	Capture input source is TIOCB1 pin Input capture at rising edge
			1		Capture input source is TIOCB1 pin Input capture at falling edge
		1	×		Capture input source is TIOCB1 pin Input capture at both edges
			×		TGRC_0 compare match/input capture Input capture at generation of TGRC_0 compare match/input capture
	1	×	×		

Legend: ×: Don't care

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_2 Function	Description TIOCB2 Pin Function	
0	0	0	0	Output compare register	Output disabled	
			1		Initial output is 0 output 0 output at compare match	
		1	0		Initial output is 0 output 1 output at compare match	
			1		Initial output is 0 output Toggle output at compare match	
		1	0		0	Output disabled
					1	Initial output is 1 output 0 output at compare match
	1		0	Initial output is 1 output 1 output at compare match		
			1	Initial output is 1 output Toggle output at compare match		
	1	x	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
				1		Capture input source is TIOCB2 pin Input capture at falling edge
			1	x		Capture input source is TIOCB2 pin Input capture at both edges

Legend: x: Don't care

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_3 Function	TIOCB3 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 output 0 output at compare match	
			0	Initial output is 1 output 1 output at compare match	
		1	0	Initial output is 1 output Toggle output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
1	0	0	Input capture register	Capture input source is TIOCB3 pin Input capture at rising edge	
		1		Capture input source is TIOCB3 pin Input capture at falling edge	
		1		×	Capture input source is TIOCB3 pin Input capture at both edges
	1	×	×	Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down*	

Legend: ×: Don't care

Note: \* When bits TPSC2 to TPSC0 in TCR\_4 are set to B'000 and  $\phi/1$  is used as the TCNT\_4 count clock, this setting is invalid and input capture is not generated.



				Description		
Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	TGRD_3 Function	TIOCD3 Pin Function	
0	0	0	0	Output compare register <sup>*2</sup>	Output disabled	
			1		Initial output is 0 output 0 output at compare match	
		1	0		Initial output is 0 output 1 output at compare match	
			1		Initial output is 0 output Toggle output at compare match	
	1	0	0	Output disabled		
			1	Initial output is 1 output 0 output at compare match		
		1	0	Initial output is 1 output 1 output at compare match		
			1	Initial output is 1 output Toggle output at compare match		
1	0	0	0	Input capture register <sup>*2</sup>	Capture input source is TIOCD3 pin Input capture at rising edge	
			1		Capture input source is TIOCD3 pin Input capture at falling edge	
		1	×		Capture input source is TIOCD3 pin Input capture at both edges	
	1	×	×			Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down <sup>*1</sup>

Legend: ×: Don't care

- Notes:
1. When bits TPSC2 to TPSC0 in TCR\_4 are set to B'000 and  $\phi/1$  is used as the TCNT\_4 count clock, this setting is invalid and input capture is not generated.
  2. When the BFB bit in TMDR\_3 is set to 1 and TGRD\_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_4 Function	TIOCB4 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 output 0 output at compare match	
		1	0	Initial output is 1 output 1 output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
1	0	0	0	Input capture register	Capture input source is TIOCB4 pin Input capture at rising edge
			1		Capture input source is TIOCB4 pin Input capture at falling edge
		1	×		Capture input source is TIOCB4 pin Input capture at both edges
			1		×
	Input capture at generation of TGRC_3 compare match/input capture				

Legend: ×: Don't care

Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	TGRB_5 Function	Description TIOCB5 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 output 0 output at compare match	
		1	0	Initial output is 1 output 1 output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
1	x	0	0	Input capture register	Capture input source is TIOCB5 pin Input capture at rising edge
			1		Capture input source is TIOCB5 pin Input capture at falling edge
		1	x		Capture input source is TIOCB5 pin Input capture at both edges

Legend: x: Don't care

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_0 Function	TIOCA0 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 output 0 output at compare match	
		1	0	Initial output is 1 output 1 output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin Input capture at rising edge
			1		Capture input source is TIOCA0 pin Input capture at falling edge
		1	×		Capture input source is TIOCA0 pin Input capture at both edges
	1	×	×	×	Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down

Legend: ×: Don't care

				Description	
Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	TGRC_0 Function	TIOCC0 Pin Function
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 output 0 output at compare match	
			0	Initial output is 1 output 1 output at compare match	
		1	0	Initial output is 1 output Toggle output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
1	0	0	Input capture register*	Capture input source is TIOCC0 pin Input capture at rising edge	
		1		Capture input source is TIOCC0 pin Input capture at falling edge	
		1		Capture input source is TIOCC0 pin Input capture at both edges	
	1	×	×	Capture input source is channel 1/count clock Input capture at TCNT_1 count-up/count-down	

Legend: ×: Don't care

Note: \* When the BFA bit in TMDR\_0 is set to 1 and TGRC\_0 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_1 Function	TIOCA1 Pin Function		
0	0	0	0	Output compare register	Output disabled		
			1		Initial output is 0 output 0 output at compare match		
		1	0		Initial output is 0 output 1 output at compare match		
			1		Initial output is 0 output Toggle output at compare match		
	1	0	0		Output disabled		
			1		Initial output is 1 output 0 output at compare match		
		1	0		Initial output is 1 output 1 output at compare match		
			1		Initial output is 1 output Toggle output at compare match		
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin Input capture at rising edge		
			1		Capture input source is TIOCA1 pin Input capture at falling edge		
		1	×		Capture input source is TIOCA1 pin Input capture at both edges		
			×		Capture input source is TGRA_0 compare match/input capture Input capture at generation of channel 0/TGRA_0 compare match/input capture		
	1	×	×				
		1	×		×		

Legend: ×: Don't care

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_2 Function	TIOCA2 Pin Function	
0	0	0	0	Output compare register	Output disabled	
			1		Initial output is 0 output 0 output at compare match	
		1	0		Initial output is 0 output 1 output at compare match	
			1		Initial output is 0 output Toggle output at compare match	
		1	0		0	Output disabled
					1	Initial output is 1 output 0 output at compare match
	1		0	Initial output is 1 output 1 output at compare match		
			1	Initial output is 1 output Toggle output at compare match		
	1	x	0	0	Input capture register	Capture input source is TIOCA2 pin Input capture at rising edge
				1		Capture input source is TIOCA2 pin Input capture at falling edge
			1	x		Capture input source is TIOCA2 pin Input capture at both edges

Legend: x: Don't care

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_3 Function	TIOCA3 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 output 0 output at compare match	
		1	0	Initial output is 1 output 1 output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin Input capture at rising edge
			1		Capture input source is TIOCA3 pin Input capture at falling edge
		1	×		Capture input source is TIOCA3 pin Input capture at both edges
	1	×	×	×	Capture input source is channel 4/count clock Input capture at TCNT_4 count-up/count-down

Legend: ×: Don't care



Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	TGRC_3 Function	TIOCC3 Pin Function
0	0	0	0	Output compare register*	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 output 0 output at compare match	
			1	Initial output is 1 output 1 output at compare match	
		1	0	Initial output is 1 output Toggle output at compare match	
			1	0	Initial output is 1 output 0 output at compare match
				1	Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register*	Capture input source is TIOCC3 pin Input capture at rising edge
			1		Capture input source is TIOCC3 pin Input capture at falling edge
		1	×		Capture input source is TIOCC3 pin Input capture at both edges
			1		×

Legend: ×: Don't care

Note: \* When the BFA bit in TMDR\_3 is set to 1 and TGRC\_3 is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_4 Function	TIOCA4 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin Input capture at rising edge
			1		Capture input source is TIOCA4 pin Input capture at falling edge
		1	×		Capture input source is TIOCA4 pin Input capture at both edges
			1		×
	Input capture at generation of TGRA_3 compare match/input capture				

Legend: ×: Don't care

Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	TGRA_5 Function	TIOCA5 Pin Function
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0	Output disabled	
			1	Initial output is 1 output 0 output at compare match	
			0	Initial output is 1 output 1 output at compare match	
		1	0	Initial output is 1 output Toggle output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
			1	Initial output is 1 output Toggle output at compare match	
1	×	0	0	Input capture register	Input capture source is TIOCA5 pin Input capture at rising edge
			1		Input capture source is TIOCA5 pin Input capture at falling edge
		1	×		Input capture source is TIOCA5 pin Input capture at both edges
			×		Input capture source is TIOCA5 pin Input capture at both edges

Legend: ×: Don't care

TIER registers control enabling or disabling of interrupt requests for each channel. The TPU has six TIER registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TTGE	0	R/W	<p>A/D Conversion Start Request Enable</p> <p>Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match.</p> <p>0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled</p>
6	—	1	—	<p>Reserved</p> <p>This bit is always read as 1 and cannot be modified.</p>
5	TCIEU	0	R/W	<p>Underflow Interrupt Enable</p> <p>Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1, 2, 4, and 5.</p> <p>In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.</p> <p>0: Interrupt requests (TCIU) by TCFU disabled 1: Interrupt requests (TCIU) by TCFU enabled</p>
4	TCIEV	0	R/W	<p>Overflow Interrupt Enable</p> <p>Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1.</p> <p>0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled</p>
3	TGIED	0	R/W	<p>TGR Interrupt Enable D</p> <p>Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3.</p> <p>In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified.</p> <p>0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled</p>

Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.

0: Interrupt requests (TGIC) by TGFC bit disabled

1: Interrupt requests (TGIC) by TGFC bit enabled

---

1	TGIEB	0	R/W	TGR Interrupt Enable B
				Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1.
				0: Interrupt requests (TGIB) by TGFB bit disabled
				1: Interrupt requests (TGIB) by TGFB bit enabled

---

0	TGIEA	0	R/W	TGR Interrupt Enable A
				Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1.
				0: Interrupt requests (TGIA) by TGFA bit disabled
				1: Interrupt requests (TGIA) by TGFA bit enabled

---

TSR registers indicate the status of each channel. The TPU has six TSR registers, one for each channel.

Bit	Bit Name	Initial value	R/W	Description
7	TCFD	1	R	<p>Count Direction Flag</p> <p>Status flag that shows the direction in which TCNT counts in channels 1, 2, 4, and 5. In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified.</p> <p>0: TCNT counts down 1: TCNT counts up</p>
6	—	1	—	<p>Reserved</p> <p>This bit is always read as 1 and cannot be modified.</p>
5	TCFU	0	R/(W)*	<p>Underflow Flag</p> <p>Status flag that indicates that TCNT underflow has occurred when channels 1, 2, 4, and 5 are set to phase counting mode. In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.</p> <p>[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)</p> <p>[Clearing condition] When 0 is written to TCFU after reading TCFU = 1</p>
4	TCFV	0	R/(W)*	<p>Overflow Flag</p> <p>Status flag that indicates that TCNT overflow has occurred.</p> <p>[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)</p> <p>[Clearing condition] When 0 is written to TCFV after reading TCFV = 1</p>

3	TGFD	0	R/(W)*	<p>Input Capture/Output Compare Flag D</p> <p>Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0 and 3.</p> <p>In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>When TCNT = TGRD while TGRD is functioning as output compare register</li> <li>When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0</li> <li>When 0 is written to TGFD after reading TGFD = 1</li> </ul>
---	------	---	--------	--

2	TGFC	0	R/(W)*	<p>Input Capture/Output Compare Flag C</p> <p>Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0 and 3.</p> <p>In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>When TCNT = TGRC while TGRC is functioning as output compare register</li> <li>When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When DTC is activated by TGIC interrupt while DISEL bit of MRB in DTC is 0</li> <li>When 0 is written to TGFC after reading TGFC = 1</li> </ul>
---	------	---	--------	--

1	TGFB	0	R/(W)*	<p>Input Capture/Output Compare Flag B</p> <p>Status flag that indicates the occurrence of TGRB input capture or compare match.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When TCNT = TGRB while TGRB is functioning as output compare register</li> <li>• When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0</li> <li>• When 0 is written to TGFB after reading TGFB = 1</li> </ul>
0	TGFA	0	R/(W)*	<p>Input Capture/Output Compare Flag A</p> <p>Status flag that indicates the occurrence of TGRA input capture or compare match.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When TCNT = TGRA while TGRA is functioning as output compare register</li> <li>• When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0</li> <li>• When 0 is written to TGFA after reading TGFA = 1</li> <li>• When DMAC is activated by TGIA interrupt while DTE bit of DMABCR in DMAC is set to 1</li> </ul>

Note: \* Only 0 can be written, for flag clearing.



The TCNT registers are 16-bit readable/writable counters. The TPU has six TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset, or in hardware standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

### 10.3.7 Timer General Register (TGR)

The TGR registers are 16-bit readable/writable registers with a dual function as output compare and input capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operation as buffer registers. The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit. TGR buffer register combinations are TGRA–TGRC and TGRB–TGRD.

### 10.3.8 Timer Start Register (TSTR)

TSTR selects operation/stoppage for channels 0 to 5. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	All 0	—	Reserved The write value should always be 0.
5	CST5	0	R/W	Counter Start 5 to 0
4	CST4	0	R/W	These bits select operation or stoppage for TCNT. If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value. 0: TCNT_5 to TCNT_0 count operation is stopped 1: TCNT_5 to TCNT_0 performs count operation
3	CST3	0	R/W	
2	CST2	0	R/W	
1	CST1	0	R/W	
0	CST0	0	R/W	

TSYR selects independent operation or synchronous operation for the TCNT counters of channels 0 to 5. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

Bit	Bit Name	Initial value	R/W	Description
7, 6	—	—	R/W R/W	Reserved The write value should always be 0.
5	SYNC5	0	R/W	Timer Synchronization 5 to 0
4	SYNC4	0	R/W	These bits select whether operation is independent of or synchronized with other channels.
3	SYNC3	0	R/W	
2	SYNC2	0	R/W	When synchronous operation is selected, synchronous presetting of multiple channels, and synchronous clearing through counter clearing on another channel are possible.
1	SYNC1	0	R/W	
0	SYNC0	0	R/W	

To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR.

0: TCNT\_5 to TCNT\_0 operates independently (TCNT presetting /clearing is unrelated to other channels)

1: TCNT\_5 to TCNT\_0 performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is possible)

## 10.4.1 Basic Functions

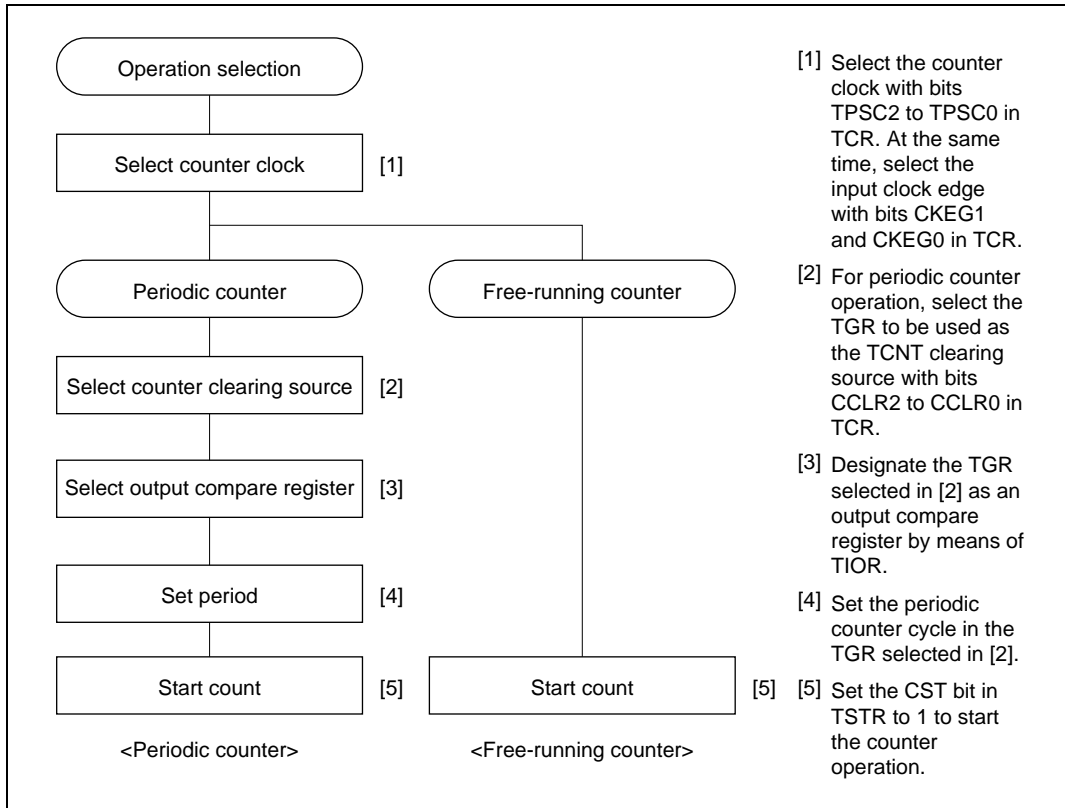
Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

**Counter Operation:** When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

### 1. Example of count operation setting procedure

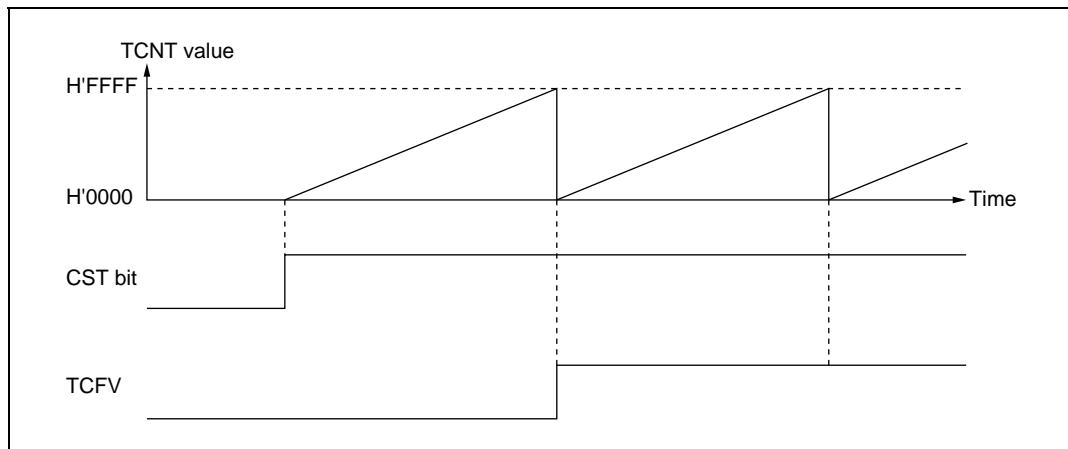
Figure 10.2 shows an example of the count operation setting procedure.



**Figure 10.2 Example of Counter Operation Setting Procedure**

counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (changes from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

Figure 10.3 illustrates free-running counter operation.

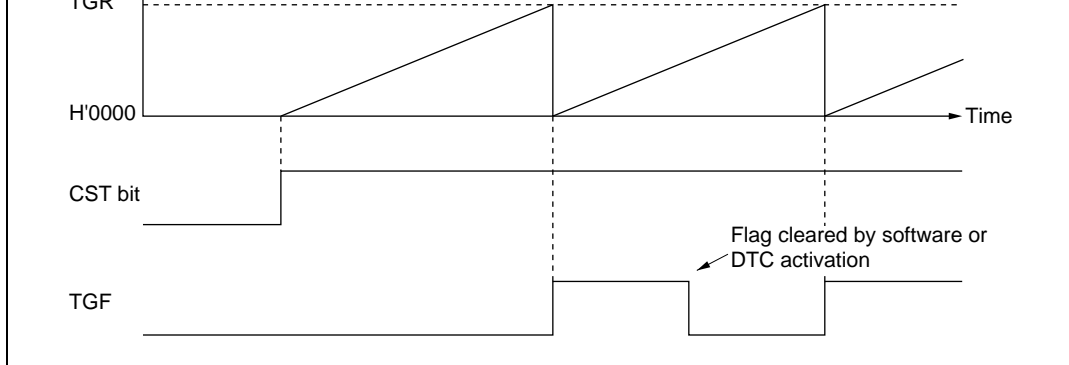


**Figure 10.3 Free-Running Counter Operation**

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

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

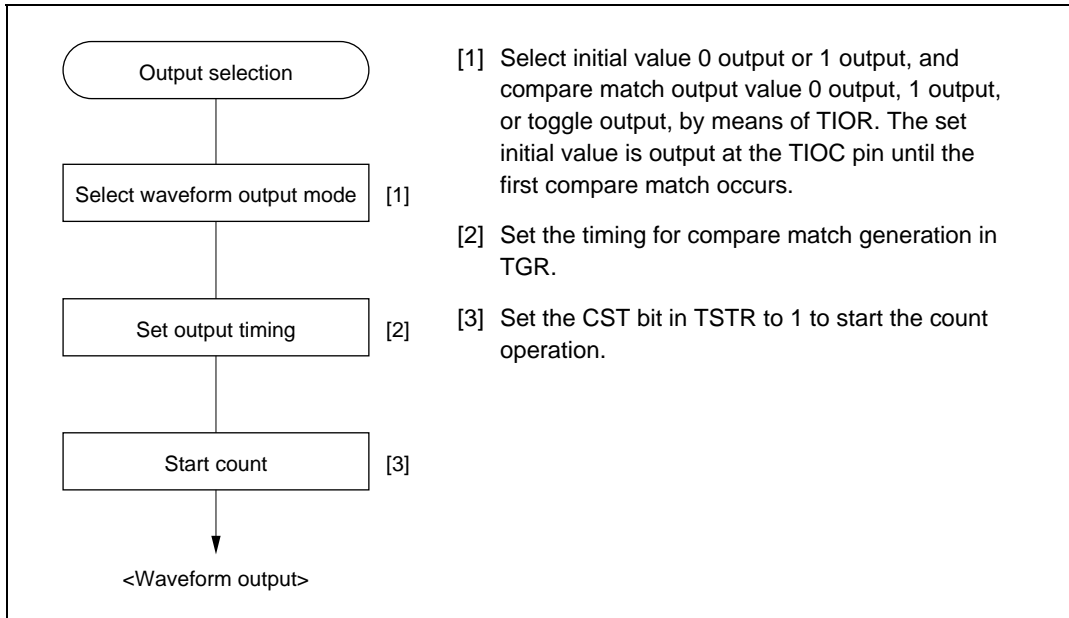
Figure 10.4 illustrates periodic counter operation.



**Figure 10.4 Periodic Counter Operation**

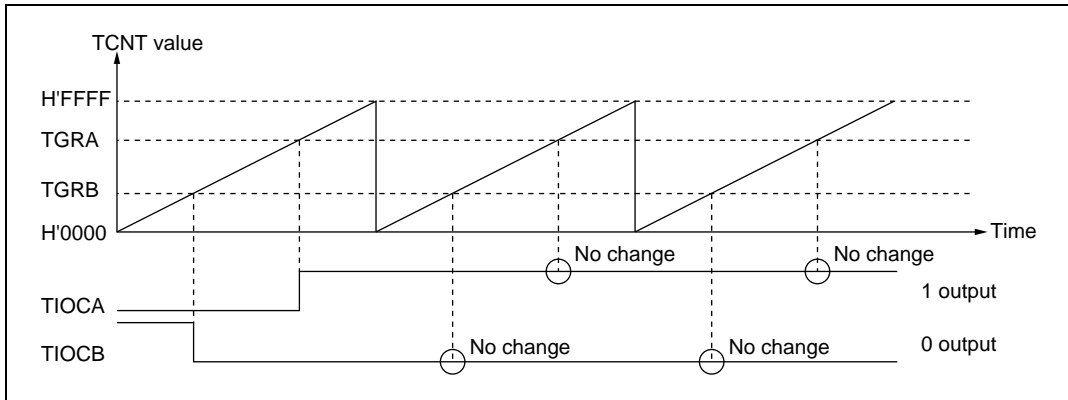
## 1. Example of setting procedure for waveform output by compare match

Figure 10.5 shows an example of the setting procedure for waveform output by a compare match.



**Figure 10.5 Example of Setting Procedure for Waveform Output by Compare Match**

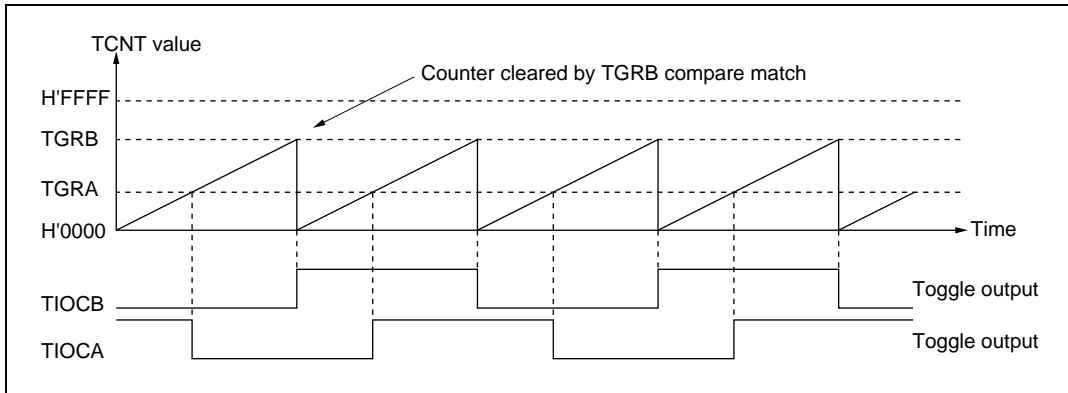
In this example, TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level match, the pin level does not change.



**Figure 10.6 Example of 0 Output/1 Output Operation**

Figure 10.7 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.



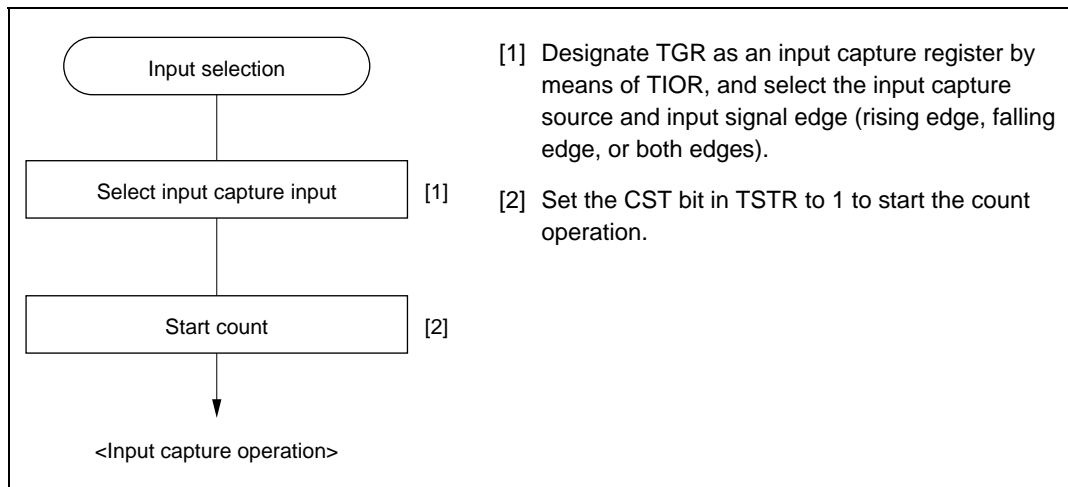
**Figure 10.7 Example of Toggle Output Operation**

Rising edge, falling edge, or both edges can be selected as the detection edge. For channels 0, 1, 3, and 4, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 3,  $\phi/1$  should not be selected as the counter input clock used for input capture input. Input capture will not be generated if  $\phi/1$  is selected.

1. Example of setting procedure for input capture operation

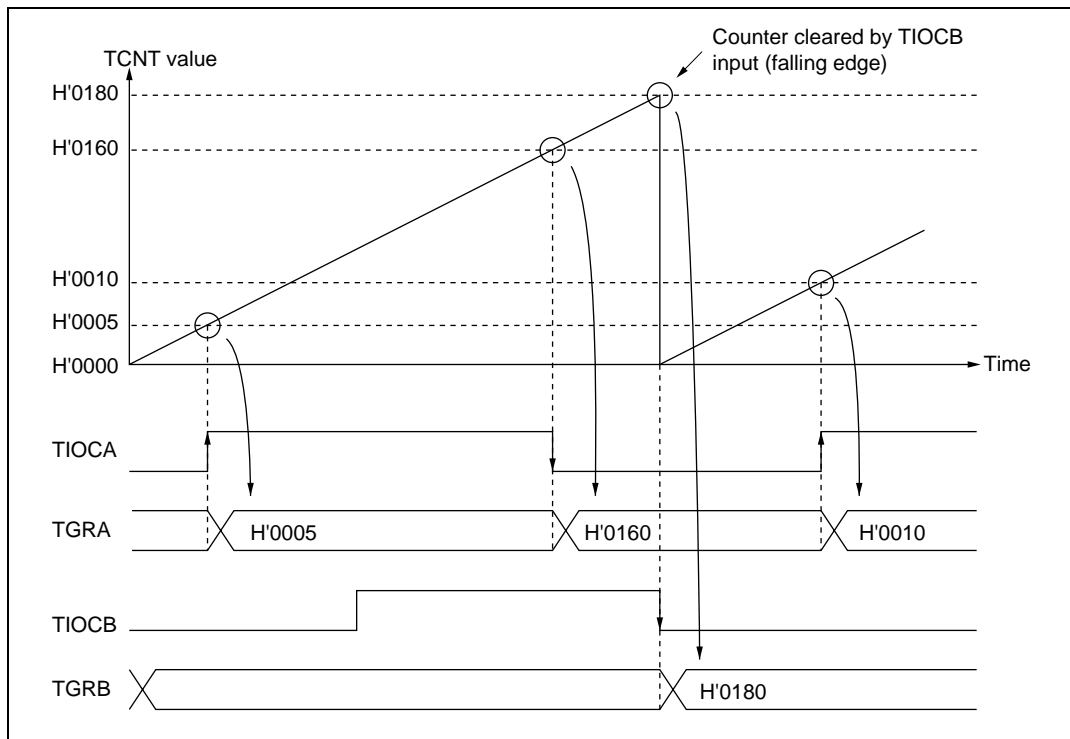
Figure 10.8 shows an example of the setting procedure for input capture operation.



**Figure 10.8 Example of Setting Procedure for Input Capture Operation**



In this example both rising and falling edges have been selected as the TIOCA pin input capture edge, falling edge has been selected as the TIOCB pin input capture edge, and counter clearing by TGRB input capture has been designated for TCNT.



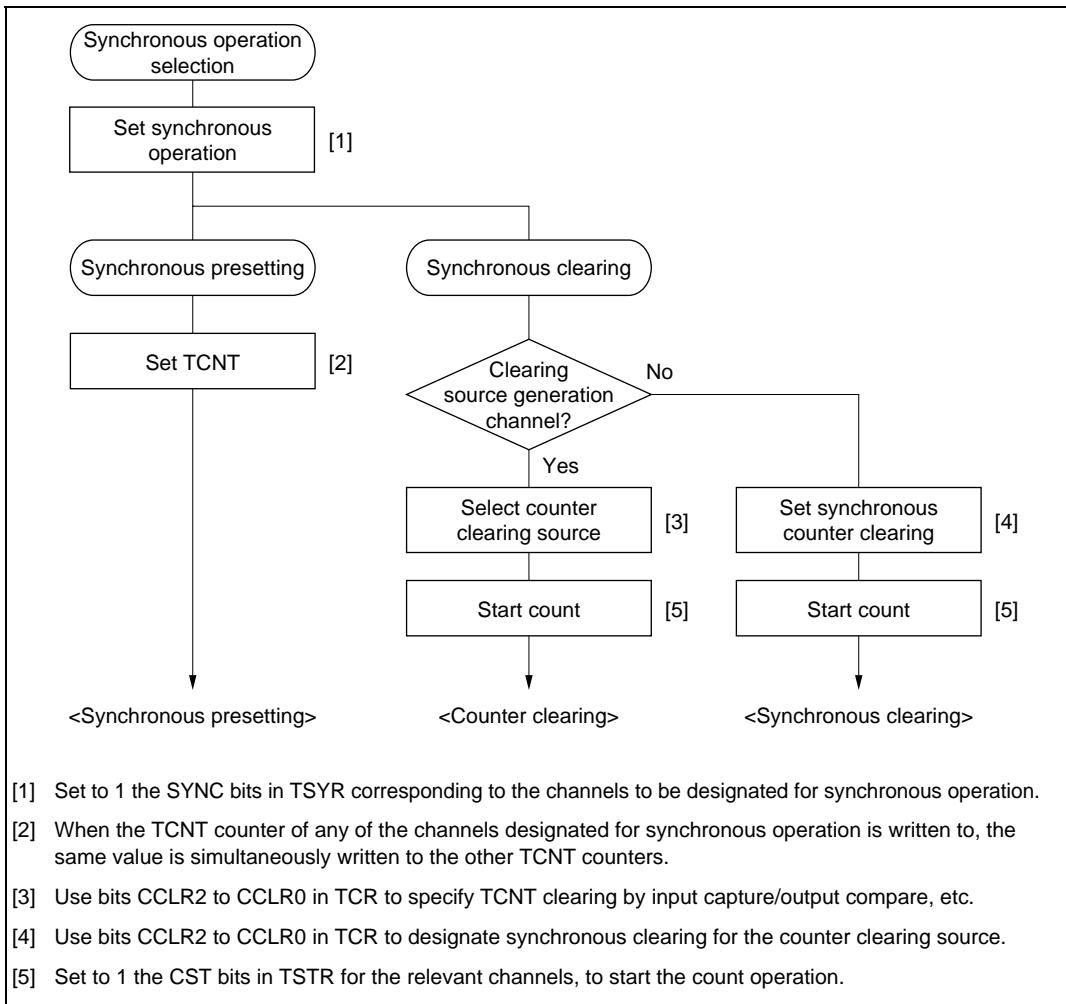
**Figure 10.9 Example of Input Capture Operation**

### 10.4.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be rewritten simultaneously (synchronous presetting). Also, multiple of TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 5 can all be designated for synchronous operation.



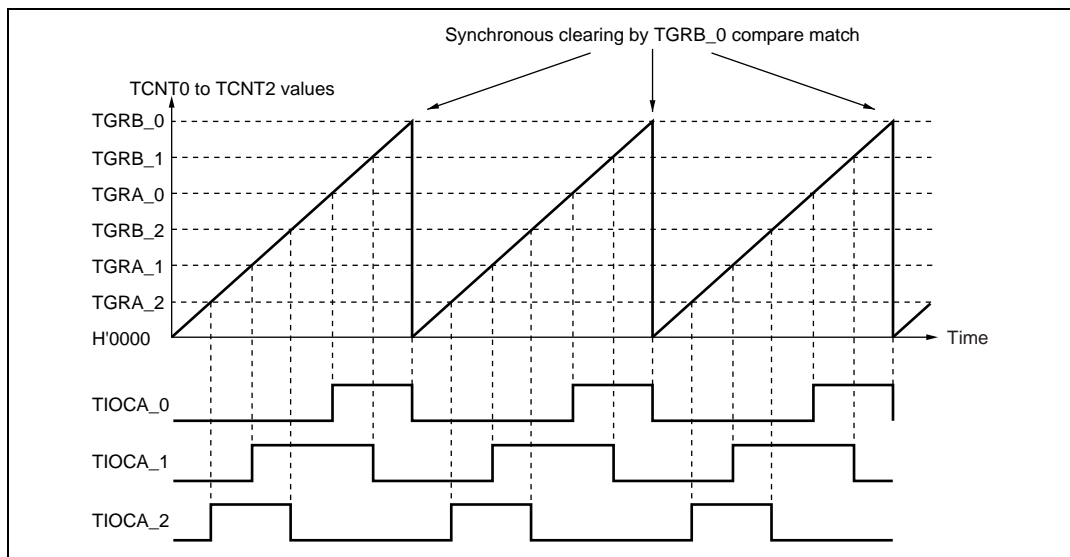
**Figure 10.10 Example of Synchronous Operation Setting Procedure**

**Example of Synchronous Operation:** Figure 10.11 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGRB\_0 compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

for channel 0 to 2 TCNT counters, and the data set in TGRB\_0 is used as the PWM cycle.

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



**Figure 10.11 Example of Synchronous Operation**

### 10.4.3 Buffer Operation

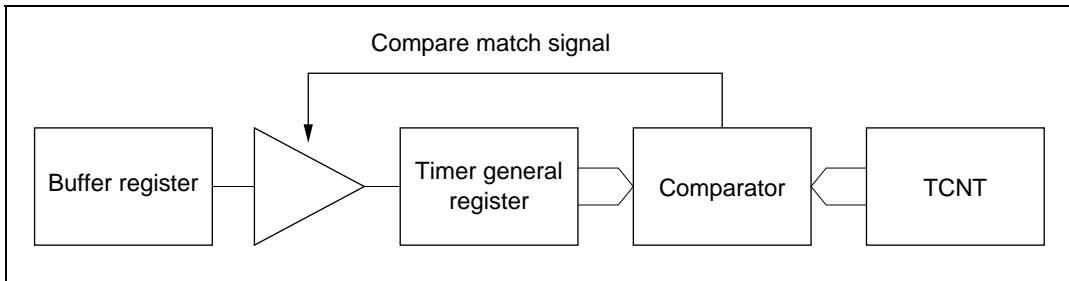
Buffer operation, provided for channels 0 and 3, enables TGRB and TGRD to be used as buffer registers.

Buffer operation differs depending on whether TGR has been designated as an input capture register or a compare match register.

Table 10.28 shows the register combinations used in buffer operation.

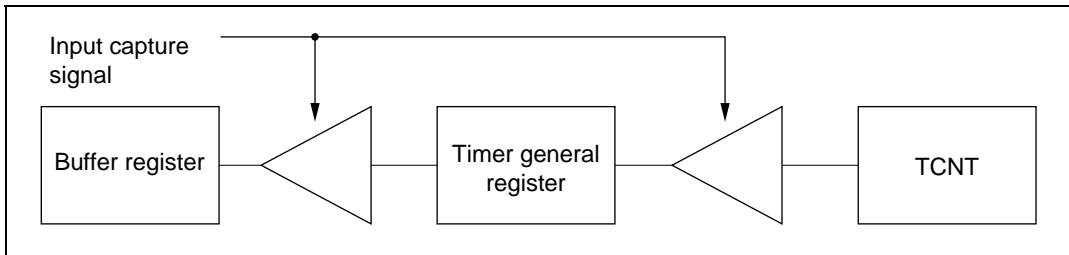
Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3

- When TGR is an output compare register  
When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.  
This operation is illustrated in figure 10.12.

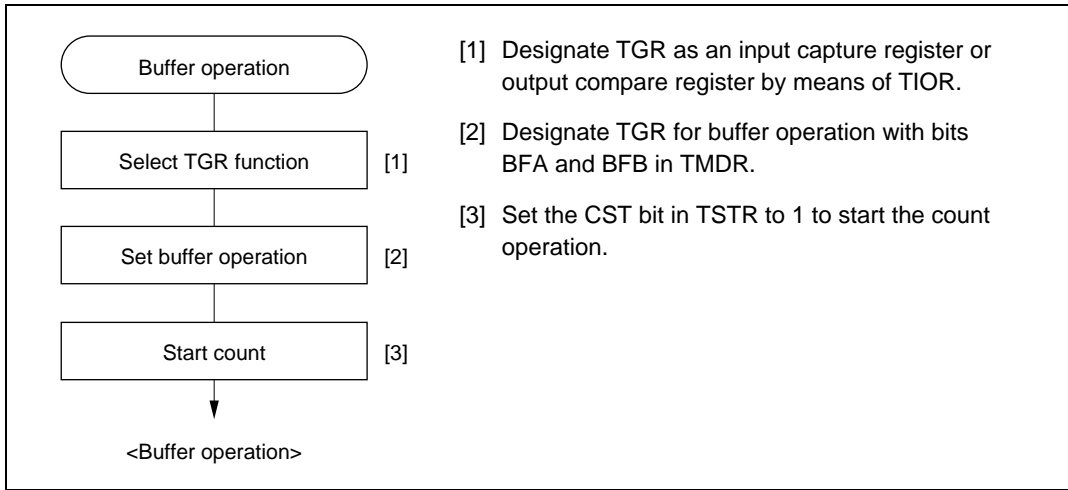


**Figure 10.12 Compare Match Buffer Operation**

- When TGR is an input capture register  
When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.  
This operation is illustrated in figure 10.13.



**Figure 10.13 Input Capture Buffer Operation**



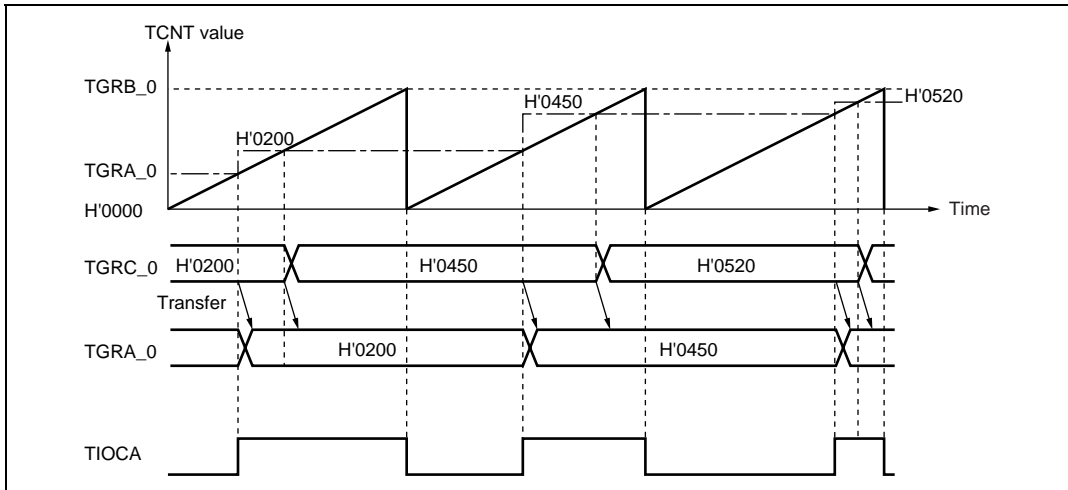
**Figure 10.14 Example of Buffer Operation Setting Procedure**

1. When TGR is an output compare register

Figure 10.15 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

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

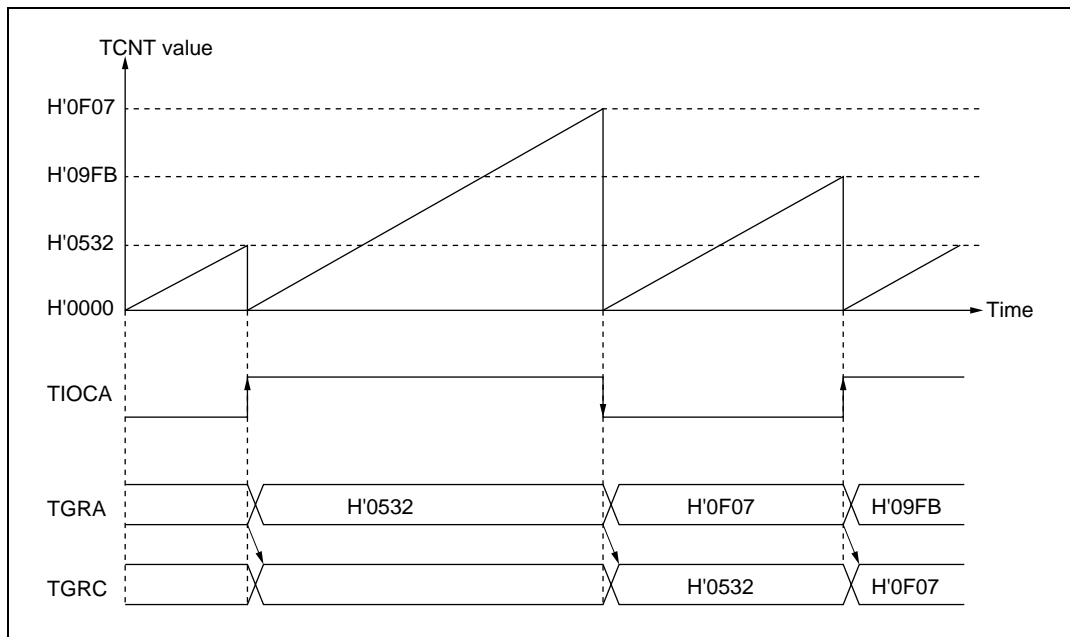


**Figure 10.15 Example of Buffer Operation (1)**

capture register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.



**Figure 10.16 Example of Buffer Operation (2)**

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 (channel 4) counter clock at overflow/underflow of TCNT\_2 (TCNT\_5) as set in bits TPSC2 to TPSC0 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase counting mode.

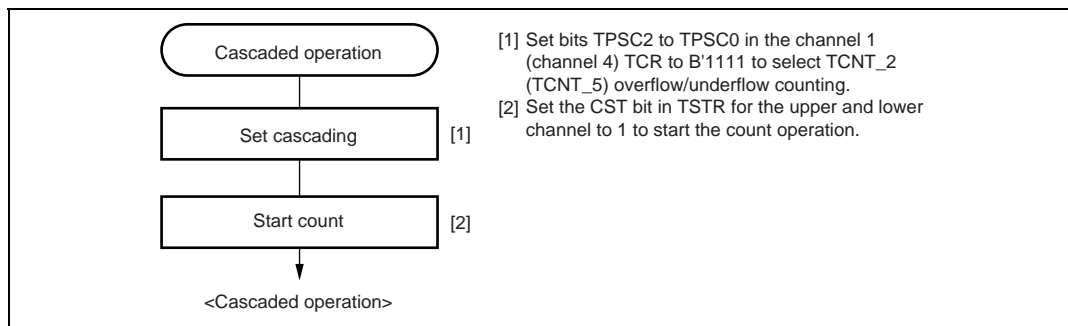
Table 10.29 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is invalid and the counter operates independently in phase counting mode.

**Table 10.29 Cascaded Combinations**

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT_1	TCNT_2
Channels 4 and 5	TCNT_4	TCNT_5

**Example of Cascaded Operation Setting Procedure:** Figure 10.17 shows an example of the setting procedure for cascaded operation.

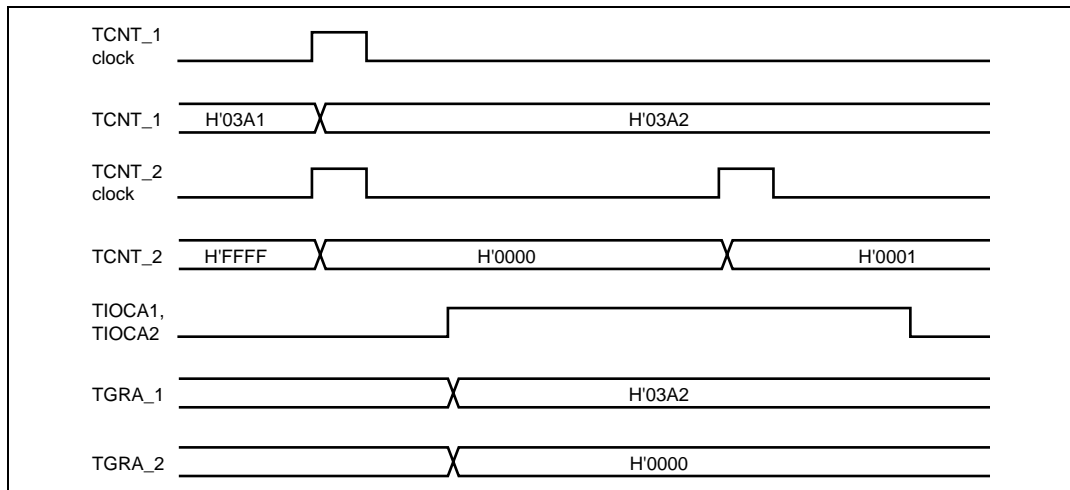


**Figure 10.17 Cascaded Operation Setting Procedure**



designated as input capture registers, and the TIOC pin rising edge has been selected.

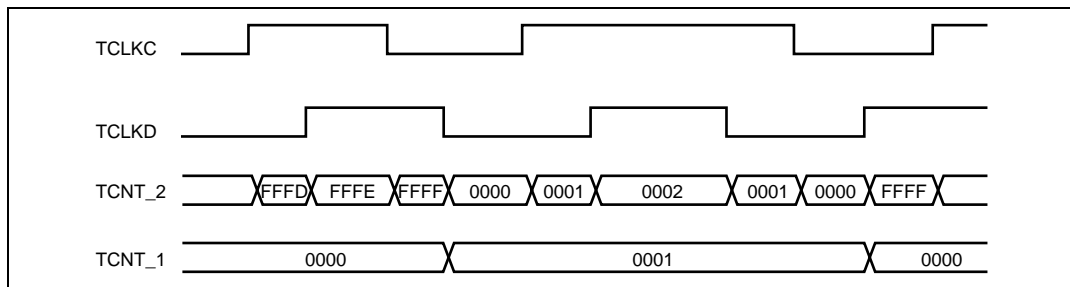
When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TGRA\_1, and the lower 16 bits to TGRA\_2.



**Figure 10.18 Example of Cascaded Operation (1)**

Figure 10.19 illustrates the operation when counting upon TCNT\_2 overflow/underflow has been set for TCNT\_1, and phase counting mode has been designated for channel 2.

TCNT\_1 is incremented by TCNT\_2 overflow and decremented by TCNT\_2 underflow.



**Figure 10.19 Example of Cascaded Operation (2)**

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR.

Settings of TGR registers can output a PWM waveform in the range of 0-% to 100-% duty.

Designating TGR compare match as the counter clearing source enables the cycle to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The outputs specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR are output from the TIOCA and TIOCC pins at compare matches A and C, respectively. The outputs specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at compare matches B and D, respectively. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

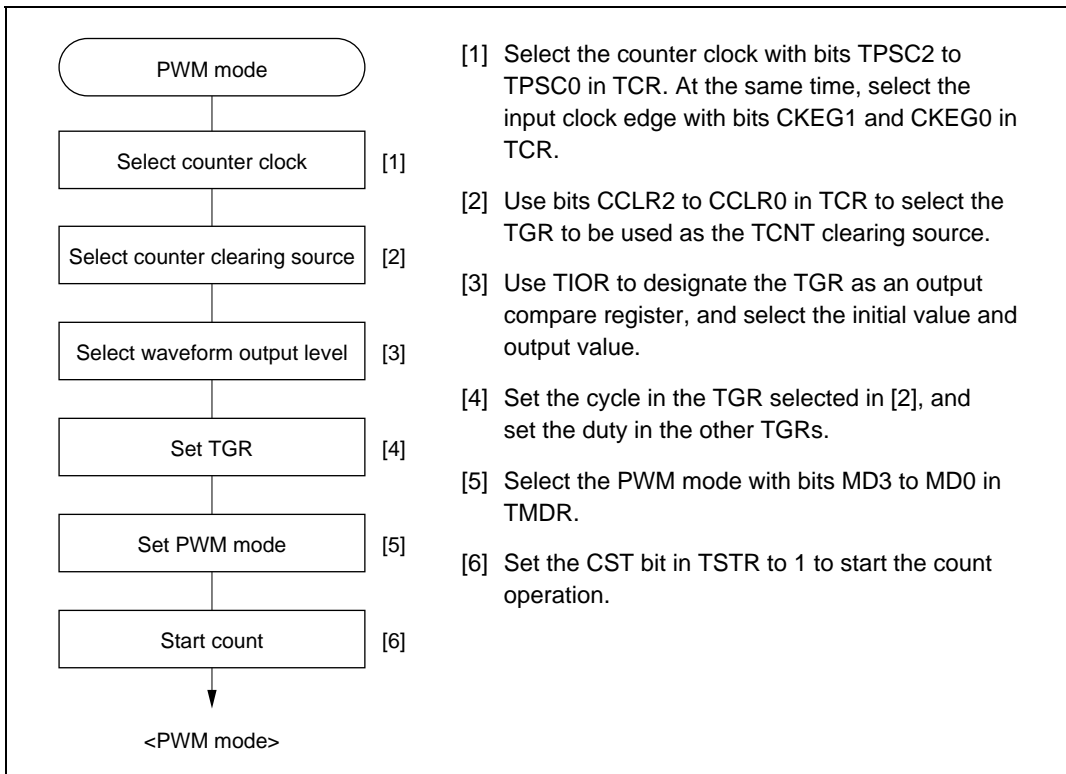
PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the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

In PWM mode 2, a maximum 15-phase PWM output is possible by combined use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 10.30.

Channel	Registers	PWM Mode 1	PWM Mode 2
0	TGRA_0	TIOCA0	TIOCA0
	TGRB_0		TIOCB0
	TGRC_0	TIOCC0	TIOCC0
	TGRD_0		TIOCD0
1	TGRA_1	TIOCA1	TIOCA1
	TGRB_1		TIOCB1
2	TGRA_2	TIOCA2	TIOCA2
	TGRB_2		TIOCB2
3	TGRA_3	TIOCA3	TIOCA3
	TGRB_3		TIOCB3
	TGRC_3	TIOCC3	TIOCC3
	TGRD_3		TIOCD3
4	TGRA_4	TIOCA4	TIOCA4
	TGRB_4		TIOCB4
5	TGRA_5	TIOCA5	TIOCA5
	TGRB_5		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the cycle is set.

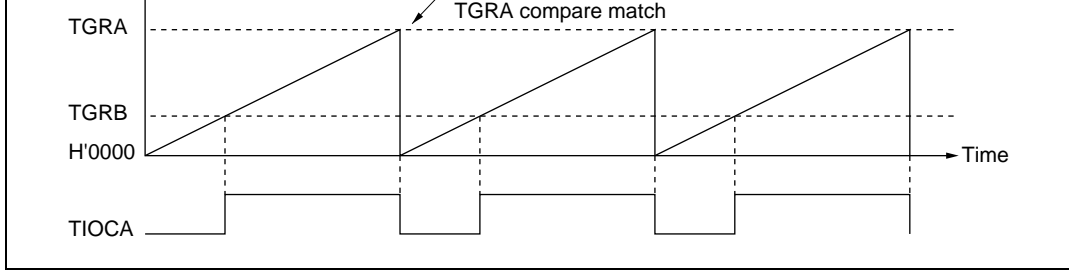


**Figure 10.20 Example of PWM Mode Setting Procedure**

**Examples of PWM Mode Operation:** Figure 10.21 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the cycle, and the values set in TGRB registers as the duty.

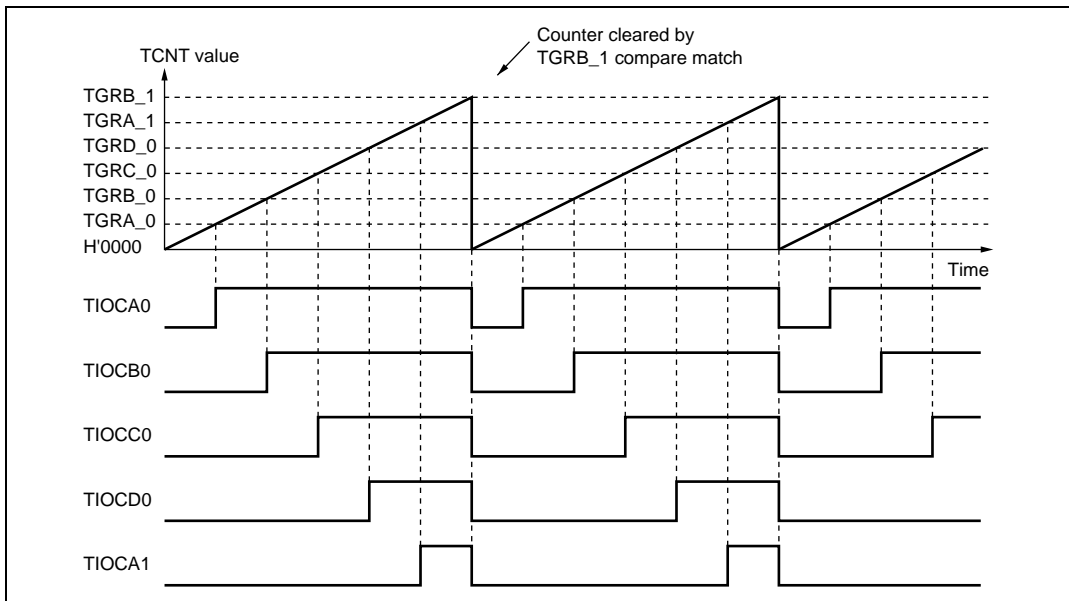


**Figure 10.21 Example of PWM Mode Operation (1)**

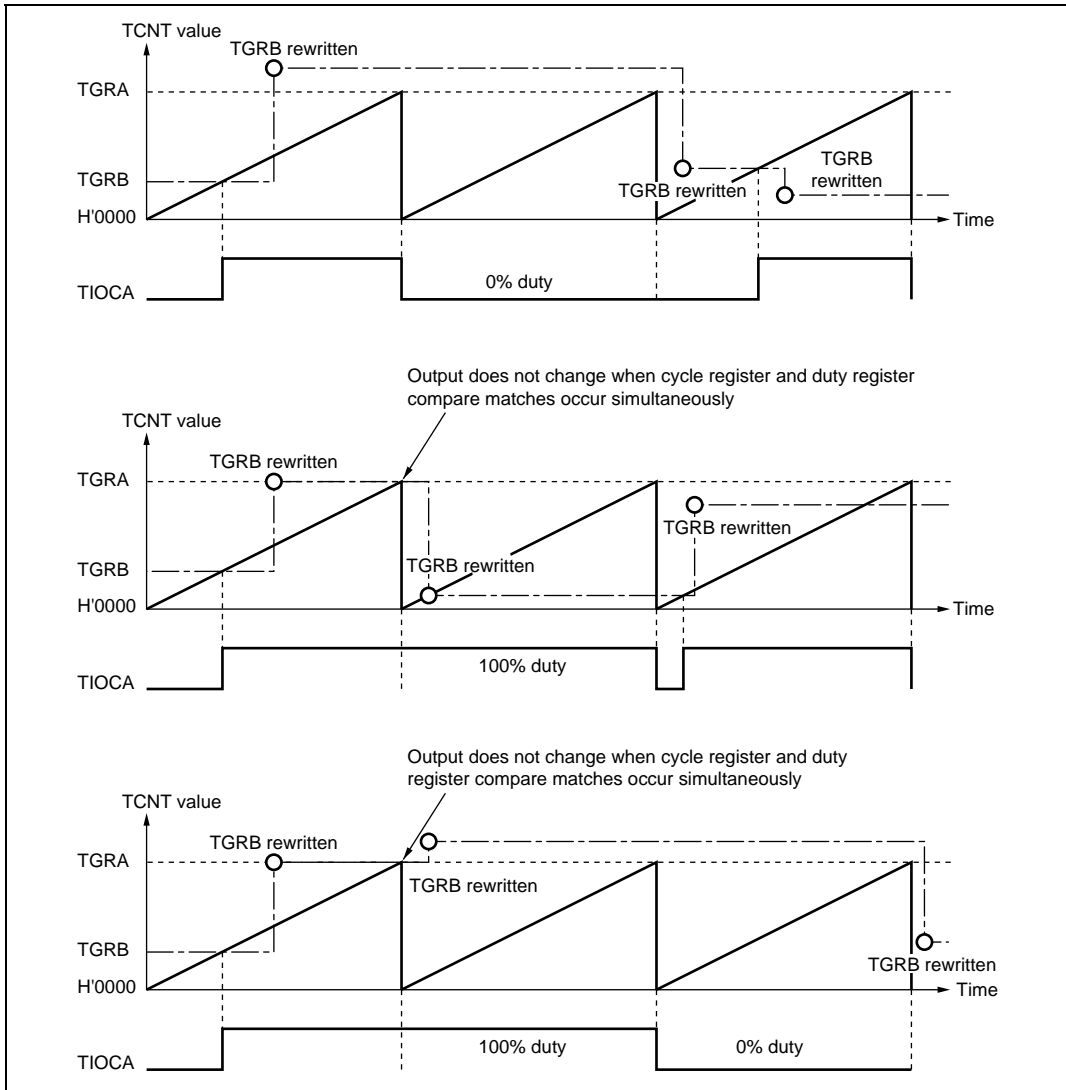
Figure 10.22 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGRB\_1 compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGRA\_0 to TGRD\_0, TGRA\_1), to output a 5-phase PWM waveform.

In this case, the value set in TGRB\_1 is used as the cycle, and the values set in the other TGRs as the duty.



**Figure 10.22 Example of PWM Mode Operation (2)**



**Figure 10.23 Example of PWM Mode Operation (3)**

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1, 2, 4, and 5.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

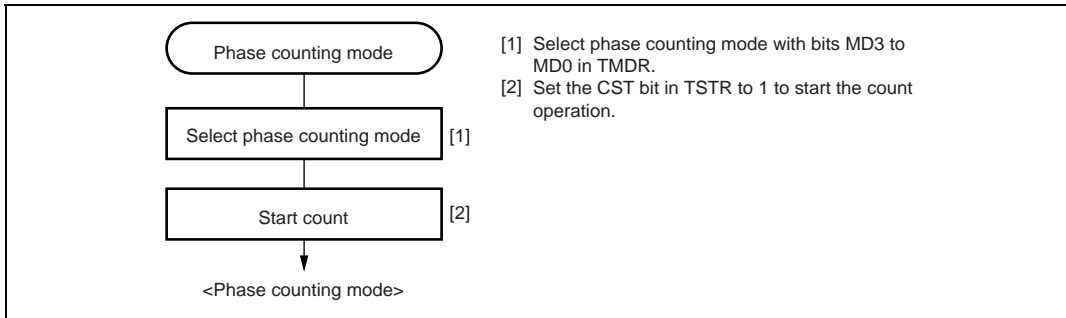
When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 10.31 shows the correspondence between external clock pins and channels.

**Table 10.31 Clock Input Pins in Phase Counting Mode**

<b>Channels</b>	<b>External Clock Pins</b>	
	<b>A-Phase</b>	<b>B-Phase</b>
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD



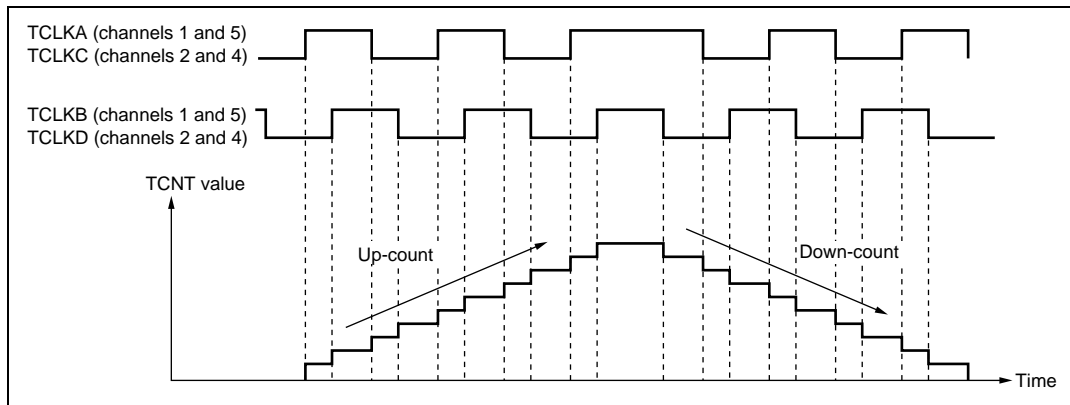
**Figure 10.24 Example of Phase Counting Mode Setting Procedure**



according to the count conditions.

### 1. Phase counting mode 1

Figure 10.25 shows an example of phase counting mode 1 operation, and table 10.32 summarizes the TCNT up/down-count conditions.



**Figure 10.25 Example of Phase Counting Mode 1 Operation**

**Table 10.32 Up/Down-Count Conditions in Phase Counting Mode 1**

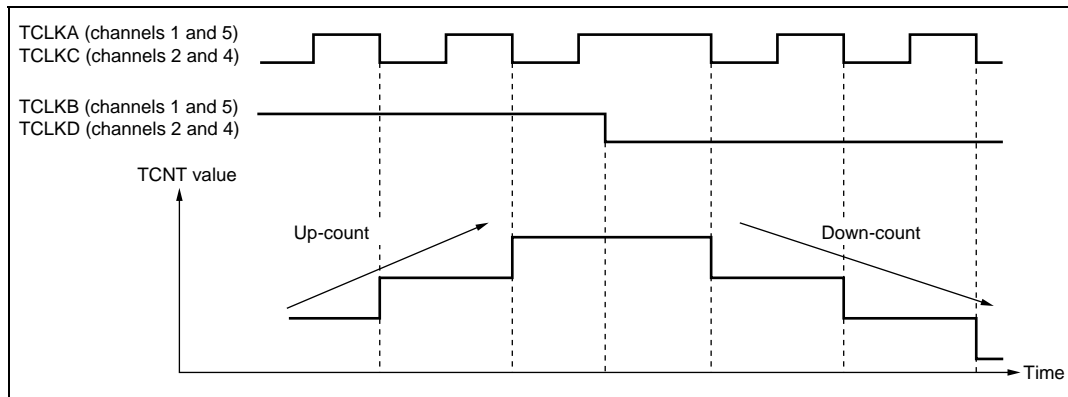
TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	┌	Up-count
Low level	┐	
┌	Low level	
┐	High level	
High level	┐	Down-count
Low level	┌	
┌	High level	
┐	Low level	

Legend:

┌ : Rising edge

┐ : Falling edge

summarizes the TCNT up/down-count conditions.



**Figure 10.26 Example of Phase Counting Mode 2 Operation**

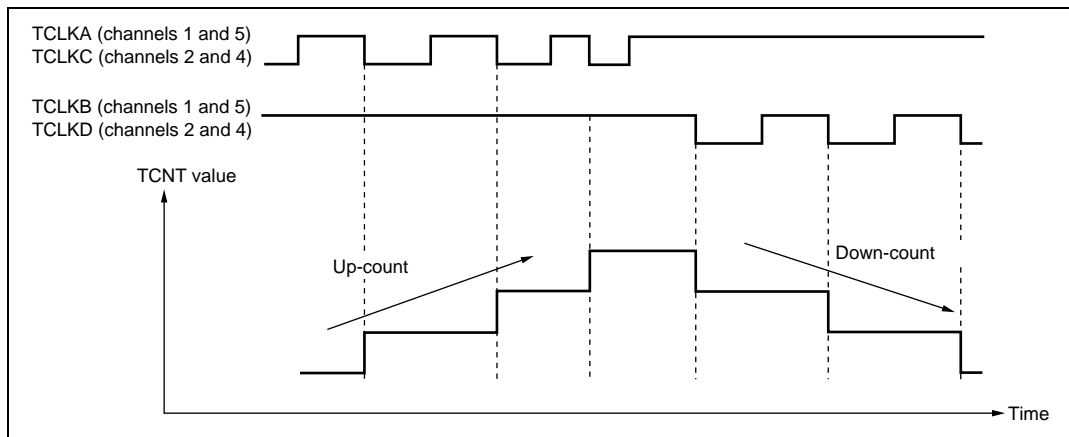
**Table 10.33 Up/Down-Count Conditions in Phase Counting Mode 2**

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	┌	Don't care
Low level	┐	Don't care
┌	Low level	Don't care
┐	High level	Up-count
High level	┐	Don't care
Low level	┌	Don't care
┌	High level	Don't care
┐	Low level	Down-count

Legend:

- ┌ : Rising edge
- ┐ : Falling edge

summarizes the TCNT up/down-count conditions.



**Figure 10.27 Example of Phase Counting Mode 3 Operation**

**Table 10.34 Up/Down-Count Conditions in Phase Counting Mode 3**

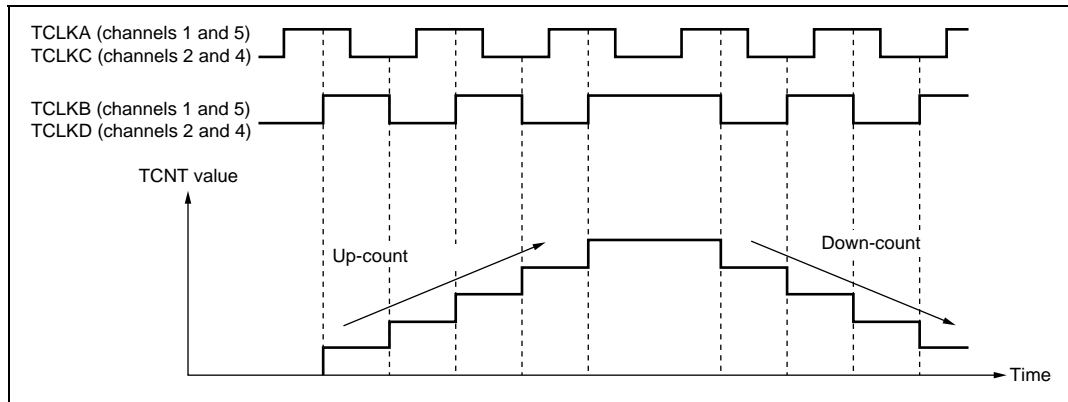
TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	$\uparrow$	Don't care
Low level	$\downarrow$	Don't care
$\uparrow$	Low level	Don't care
$\downarrow$	High level	Up-count
High level	$\downarrow$	Down-count
Low level	$\uparrow$	Don't care
$\uparrow$	High level	Don't care
$\downarrow$	Low level	Don't care

Legend:

$\uparrow$  : Rising edge

$\downarrow$  : Falling edge

summarizes the TCNT up/down-count conditions.



**Figure 10.28 Example of Phase Counting Mode 4 Operation**

**Table 10.35 Up/Down-Count Conditions in Phase Counting Mode 4**

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level Low level	┌┐	Up-count
┌┐ └┘	Low level High level	Don't care
High level Low level	└┘ ┌┐	Down-count
┌┐ └┘	High level Low level	Don't care

Legend:

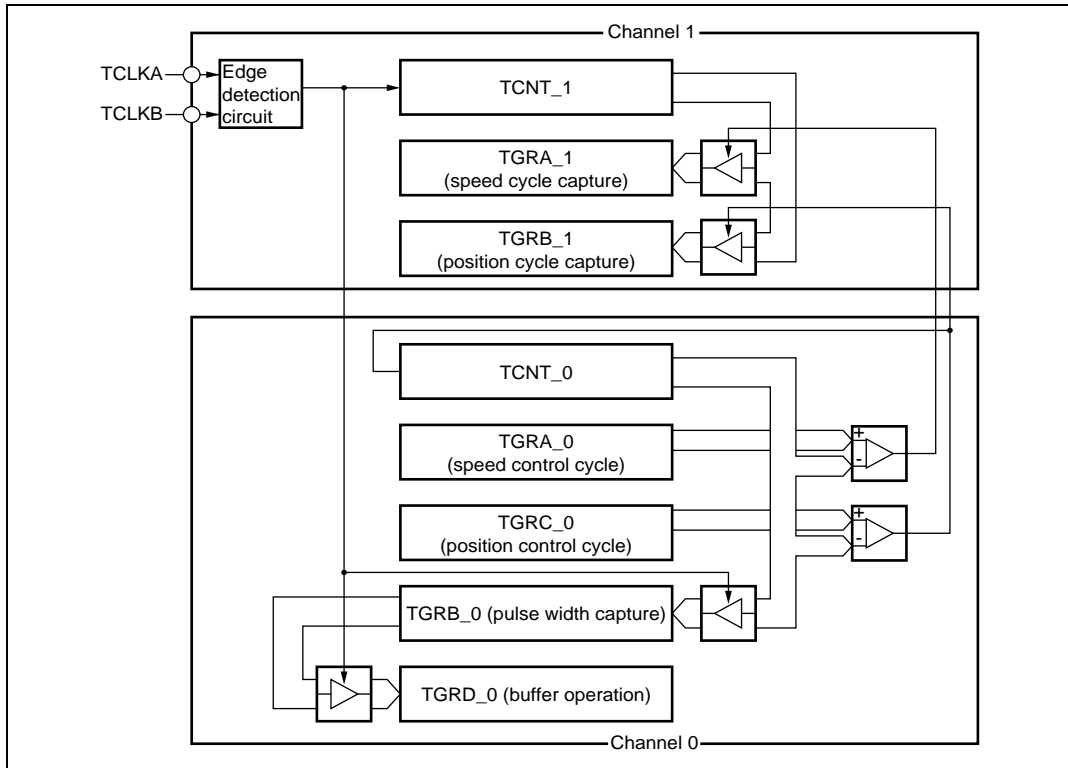
- ┌┐ : Rising edge
- └┘ : Falling edge

**Phase Counting Mode Application Example:** Figure 10.29 shows an example in which phase counting mode is designated for channel 1, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

Channel 0 operates with TCNT counter clearing by TGRC\_0 compare match; TGRA\_0 and TGRC\_0 are used for the compare match function, and are set with the speed control cycle and position control cycle. TGRB\_0 is used for input capture, with TGRB\_0 and TGRD\_0 operating in buffer mode. The channel 1 counter input clock is designated as the TGRB\_0 input capture source, and detection of the pulse width of 2-phase encoder 4-multiplication pulses is performed.

TGRA\_1 and TGRB\_1 for channel 1 are designated for input capture, channel 0 TGRA\_0 and TGRC\_0 compare matches are selected as the input capture source, and the up/down-counter values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.



**Figure 10.29 Phase Counting Mode Application Example**

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disable bit, allowing generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority order within a channel is fixed. For details, see section 5, Interrupt Controller.

Table 10.36 lists the TPU interrupt sources.

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0	Possible	Possible
	TGI0B	TGRB_0 input capture/compare match	TGFB_0	Possible	Not possible
	TGI0C	TGRC_0 input capture/compare match	TGFC_0	Possible	Not possible
	TGI0D	TGRD_0 input capture/compare match	TGFD_0	Possible	Not possible
	TGI0E	TCNT_0 overflow	TCFV_0	Not possible	Not possible
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	Possible	Possible
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	Possible	Not possible
	TCI1V	TCNT_1 overflow	TCFV_1	Not possible	Not possible
	TCI1U	TCNT_1 underflow	TCFU_1	Not possible	Not possible
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	Possible	Possible
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	Possible	Not possible
	TCI2V	TCNT_2 overflow	TCFV_2	Not possible	Not possible
	TCI2U	TCNT_2 underflow	TCFU_2	Not possible	Not possible
3	TGI3A	TGRA_3 input capture/compare match	TGFA_3	Possible	Possible
	TGI3B	TGRB_3 input capture/compare match	TGFB_3	Possible	Not possible
	TGI3C	TGRC_3 input capture/compare match	TGFC_3	Possible	Not possible
	TGI3D	TGRD_3 input capture/compare match	TGFD_3	Possible	Not possible
	TCI3V	TCNT_3 overflow	TCFV_3	Not possible	Not possible
4	TGI4A	TGRA_4 input capture/compare match	TGFA_4	Possible	Possible
	TGI4B	TGRB_4 input capture/compare match	TGFB_4	Possible	Not possible
	TCI4V	TCNT_4 overflow	TCFV_4	Not possible	Not possible
	TCI4U	TCNT_4 underflow	TCFU_4	Not possible	Not possible
5	TGI5A	TGRA_5 input capture/compare match	TGFA_5	Possible	Possible
	TGI5B	TGRB_5 input capture/compare match	TGFB_5	Possible	Not possible
	TCI5V	TCNT_5 overflow	TCFV_5	Not possible	Not possible
	TCI5U	TCNT_5 underflow	TCFU_5	Not possible	Not possible

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match interrupts, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

**Overflow Interrupt:** An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

**Underflow Interrupt:** An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channels 1, 2, 4, and 5.

## 10.6 DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 8, Data Transfer Controller (DTC).

A total of 16 TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

## 10.7 DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt for a channel. For details, see section 7, DMA Controller (DMAC).

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

## 10.8 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a channel.

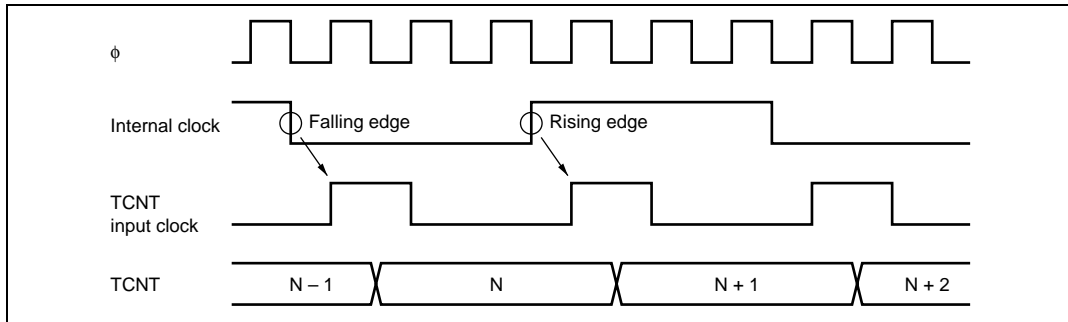
If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

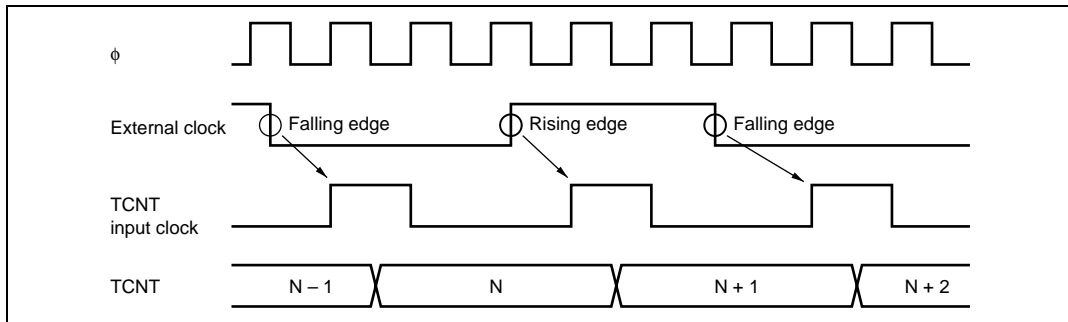


## 10.9.1 Input/Output Timing

**TCNT Count Timing:** Figure 10.30 shows TCNT count timing in internal clock operation, and figure 10.31 shows TCNT count timing in external clock operation.



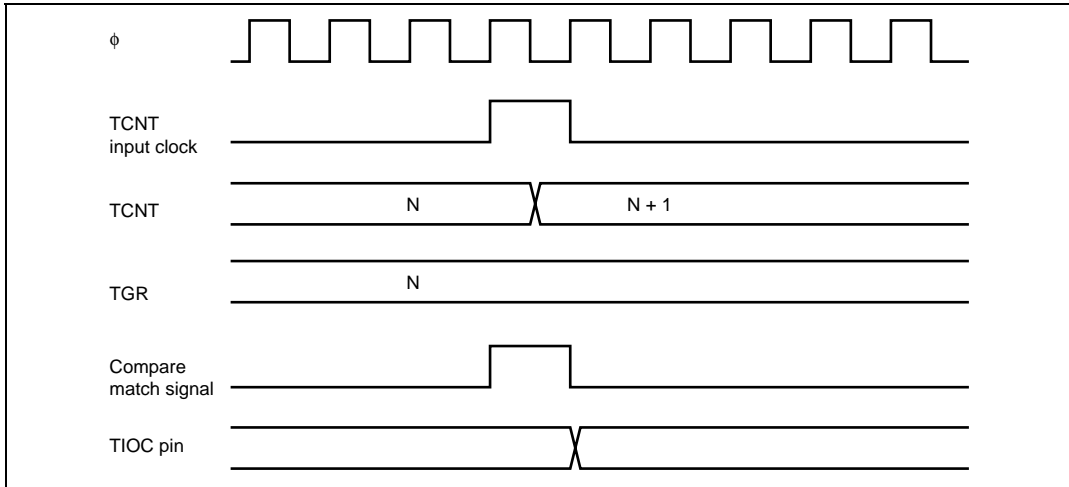
**Figure 10.30 Count Timing in Internal Clock Operation**



**Figure 10.31 Count Timing in External Clock Operation**

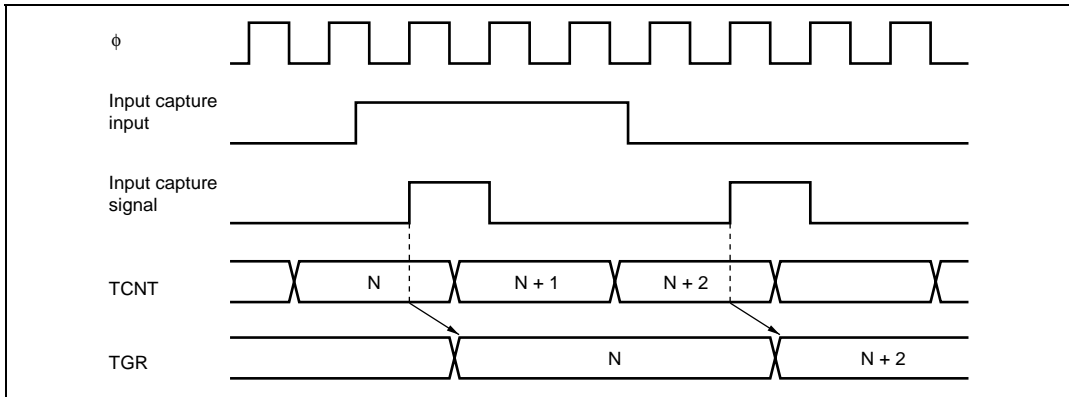
When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin. After a match between TCNT and TGR, the compare match signal is not generated until the (TIOC pin) TCNT input clock is generated.

Figure 10.32 shows output compare output timing.



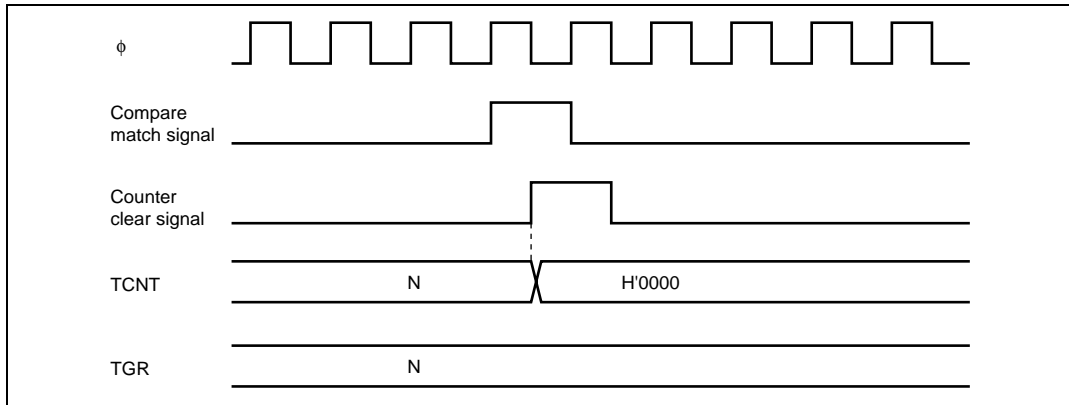
**Figure 10.32 Output Compare Output Timing**

**Input Capture Signal Timing:** Figure 10.33 shows input capture signal timing.

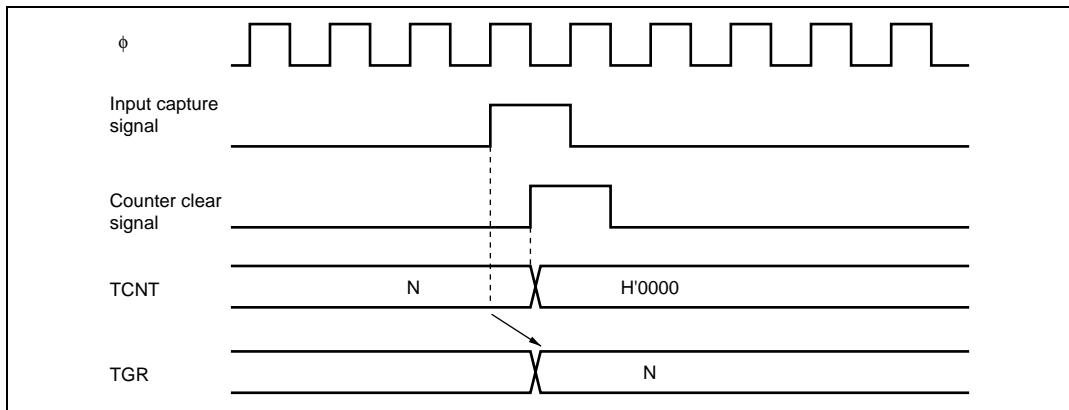


**Figure 10.33 Input Capture Input Signal Timing**

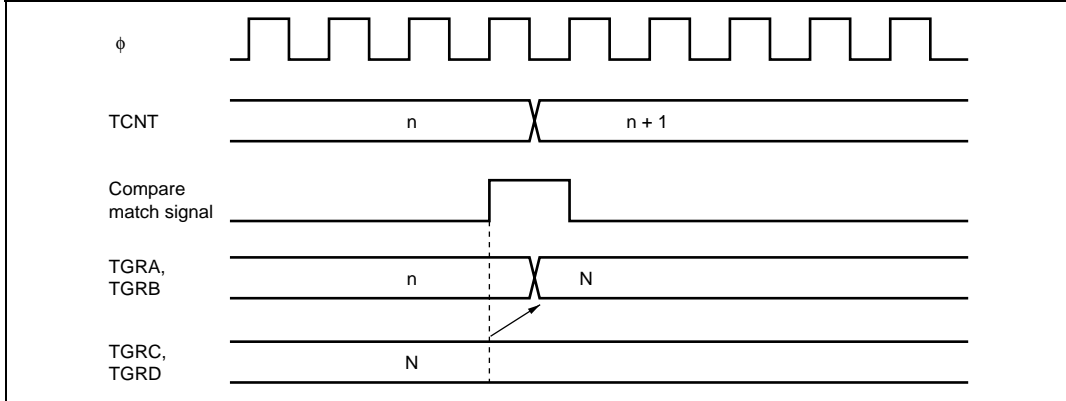
the timing when counter clearing by input capture occurrence is specified.



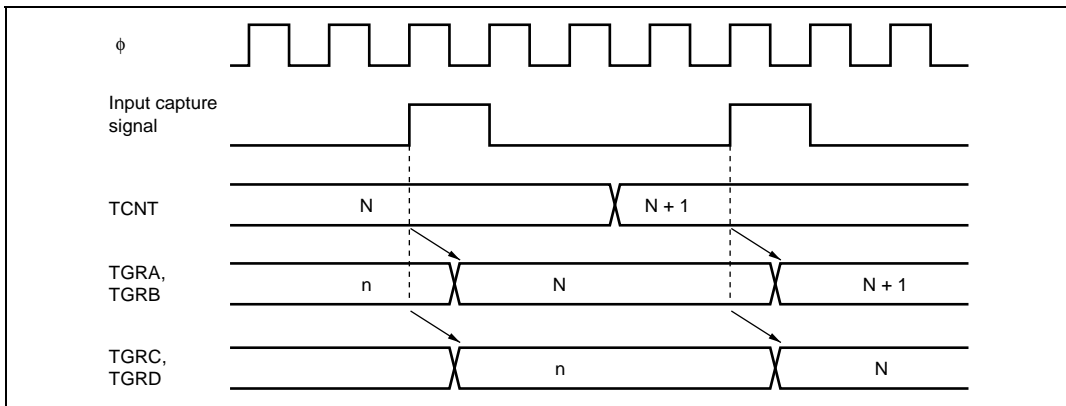
**Figure 10.34 Counter Clear Timing (Compare Match)**



**Figure 10.35 Counter Clear Timing (Input Capture)**

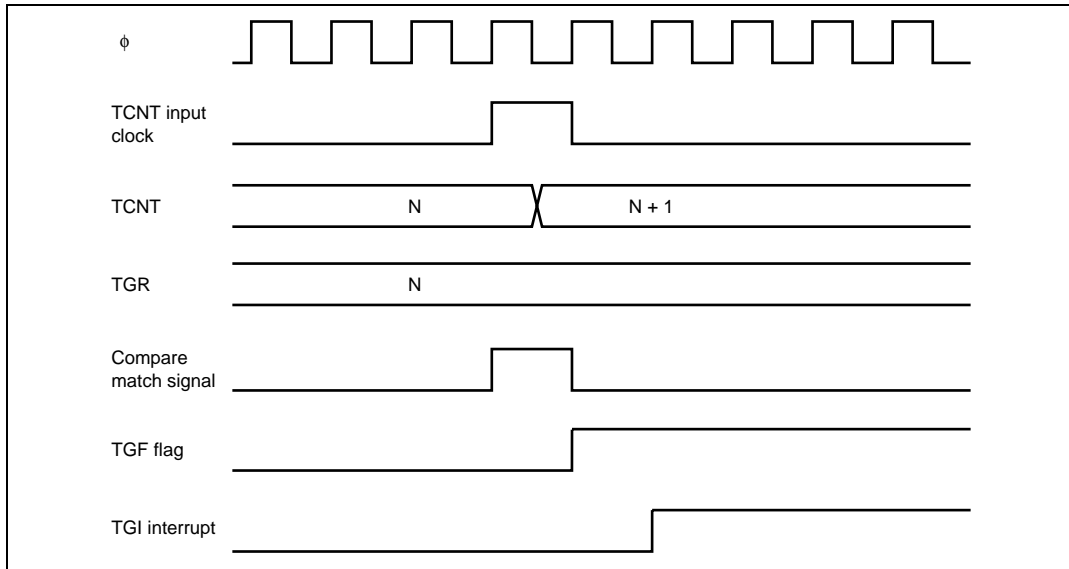


**Figure 10.36 Buffer Operation Timing (Compare Match)**

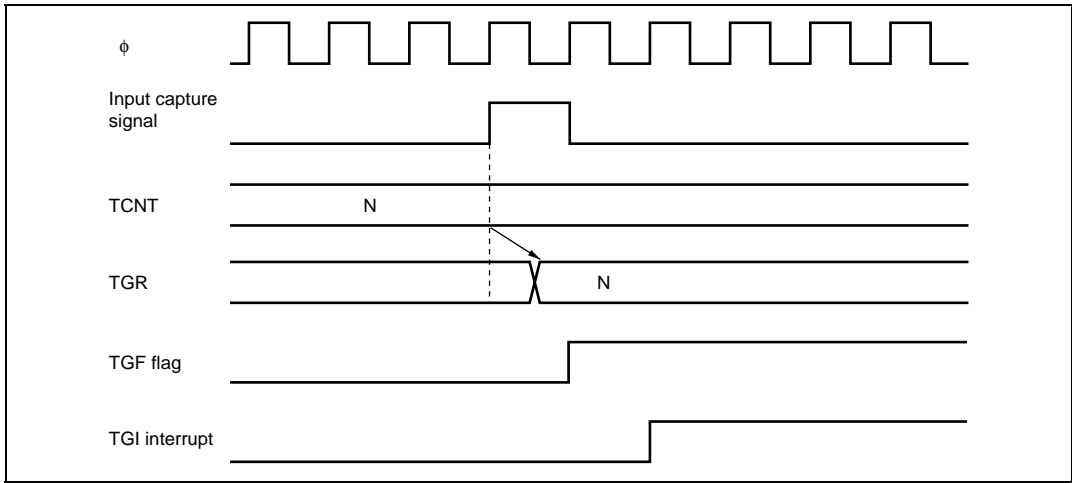


**Figure 10.37 Buffer Operation Timing (Input Capture)**

**TGF Flag Setting Timing in Case of Compare Match:** Figure 10.38 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and the TGI interrupt request signal timing.

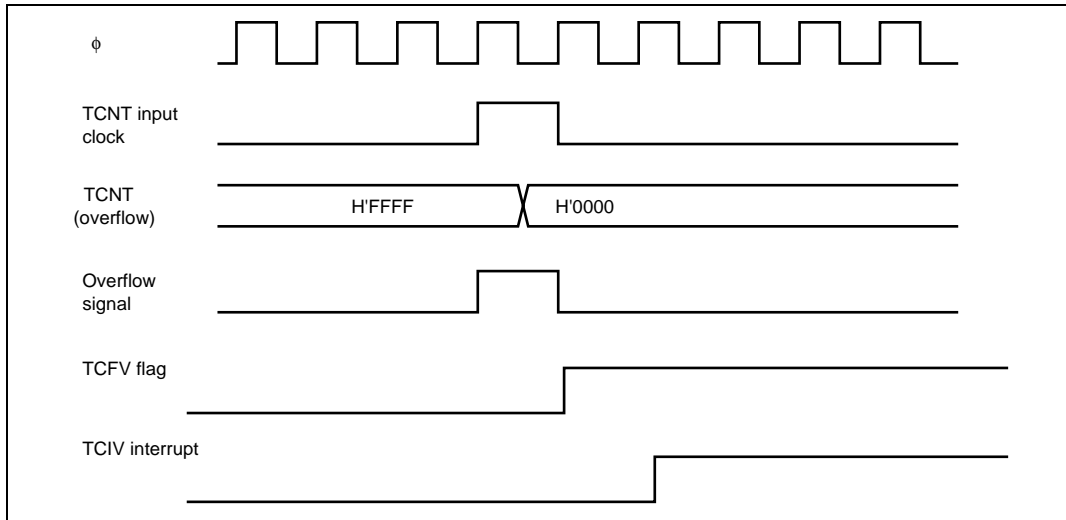


**Figure 10.38 TGI Interrupt Timing (Compare Match)**

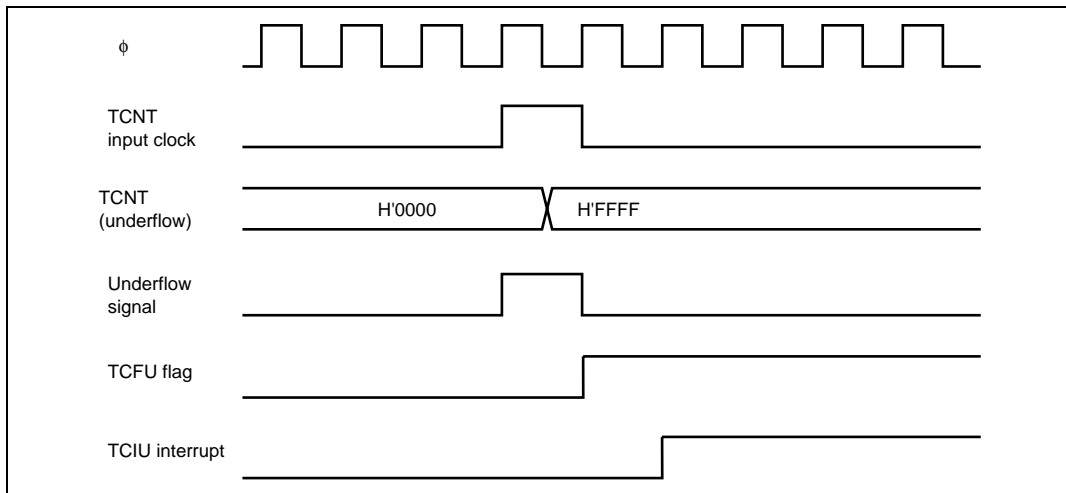


**Figure 10.39 TGI Interrupt Timing (Input Capture)**

Figure 10.41 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and the TCIU interrupt request signal timing.

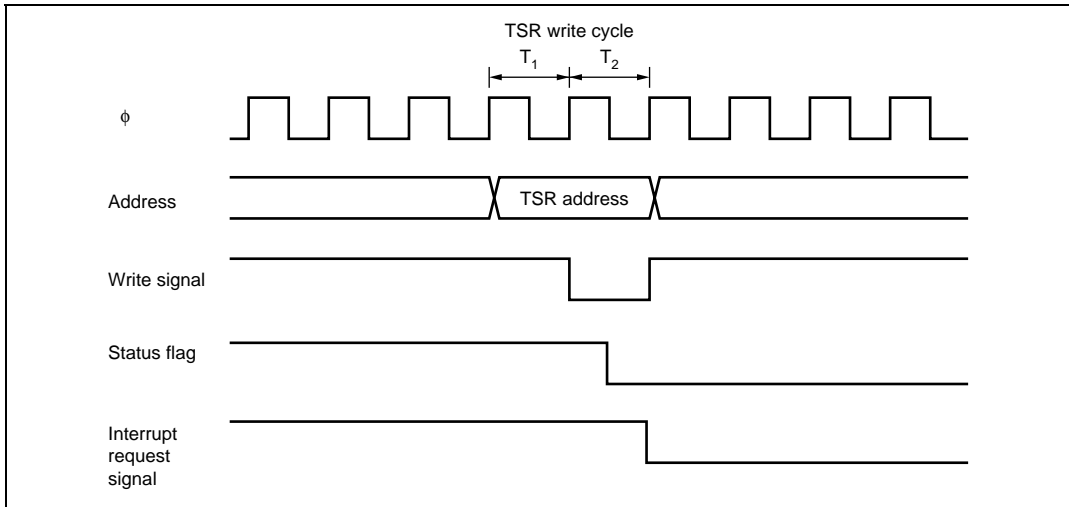


**Figure 10.40 TCIV Interrupt Setting Timing**

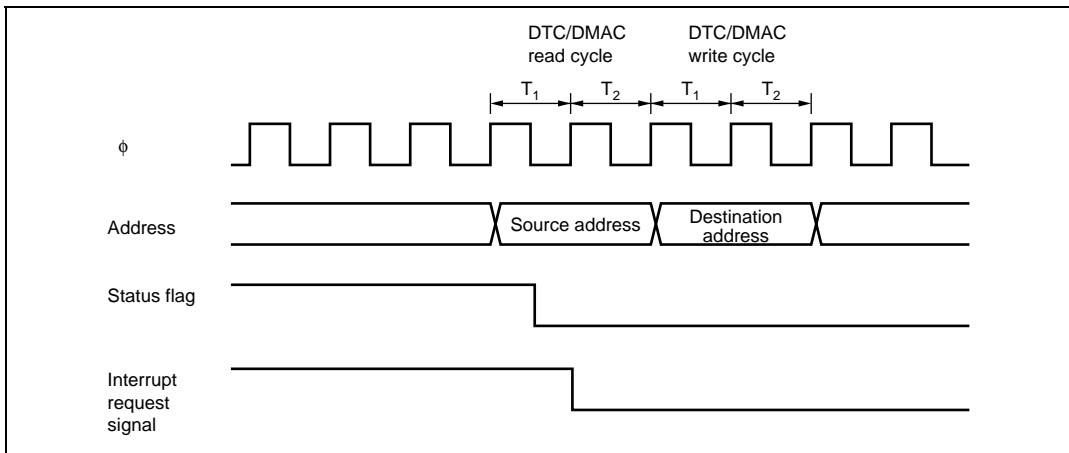


**Figure 10.41 TCIU Interrupt Setting Timing**

shows the timing for status flag clearing by the CPU, and figure 10.43 shows the timing for status flag clearing by the DTC or DMAC.



**Figure 10.42 Timing for Status Flag Clearing by CPU**



**Figure 10.43 Timing for Status Flag Clearing by DTC/DMAC\* Activation**



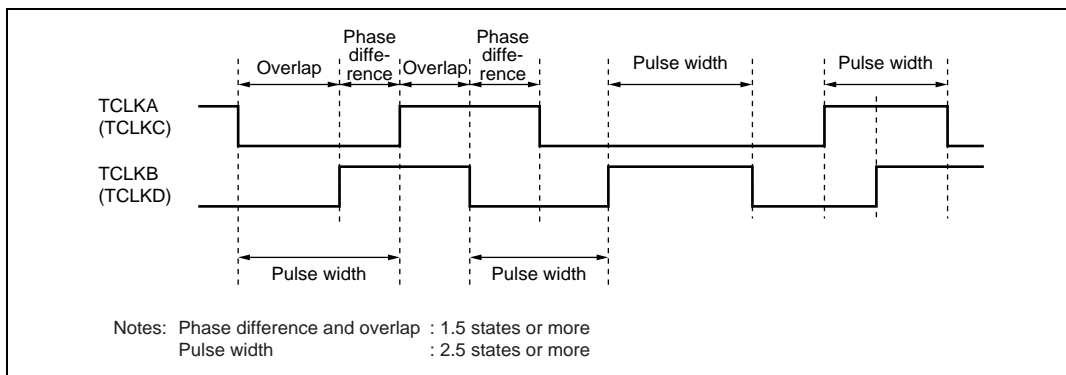
### 10.10.1 Module Stop Mode Setting

TPU operation can be disabled or enabled using the module stop control register. The initial setting is for TPU operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 23, Power-Down Modes.

### 10.10.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10.44 shows the input clock conditions in phase counting mode.



**Figure 10.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode**

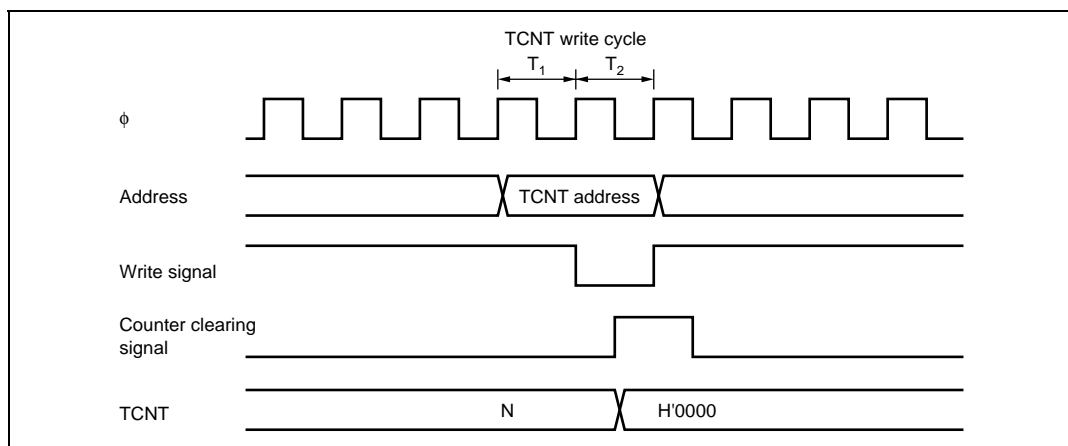
When counter clearing by compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{\phi}{(N + 1)}$$

Where f: Counter frequency  
 $\phi$ : Operating frequency  
 N: TGR set value

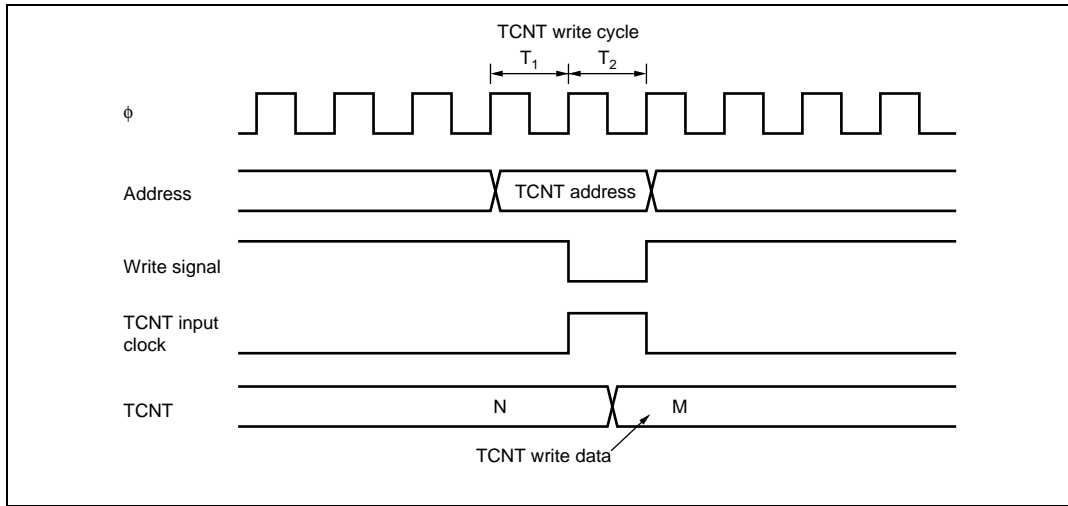
#### 10.10.4 Contention between TCNT Write and Clear Operations

If the counter clearing signal is generated in the  $T_2$  state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 10.45 shows the timing in this case.



**Figure 10.45 Contention between TCNT Write and Clear Operations**

If incrementing occurs in the  $T_2$  state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented. Figure 10.46 shows the timing in this case.

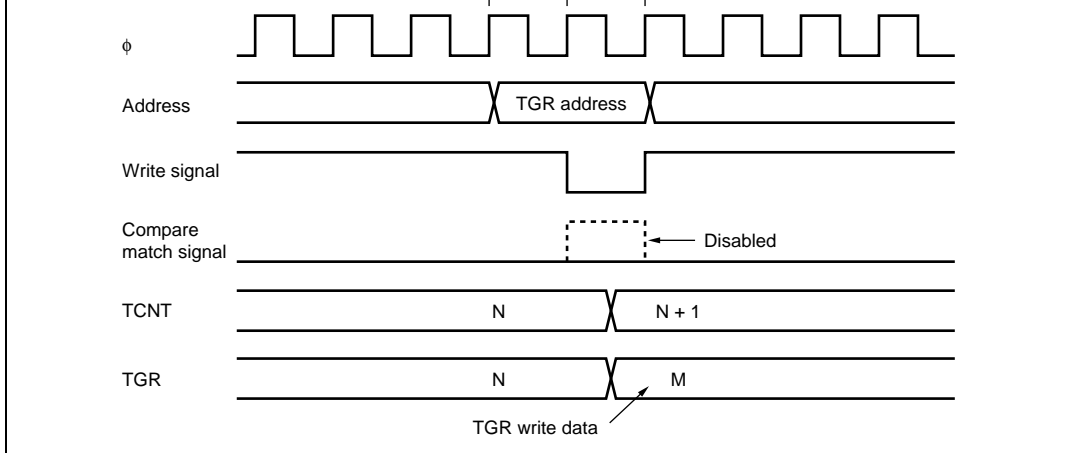


**Figure 10.46 Contention between TCNT Write and Increment Operations**

### 10.10.6 Contention between TGR Write and Compare Match

If a compare match occurs in the  $T_2$  state of a TGR write cycle, the TGR write takes precedence and the compare match signal is disabled. A compare match also does not occur when the same value as before is written.

Figure 10.47 shows the timing in this case.

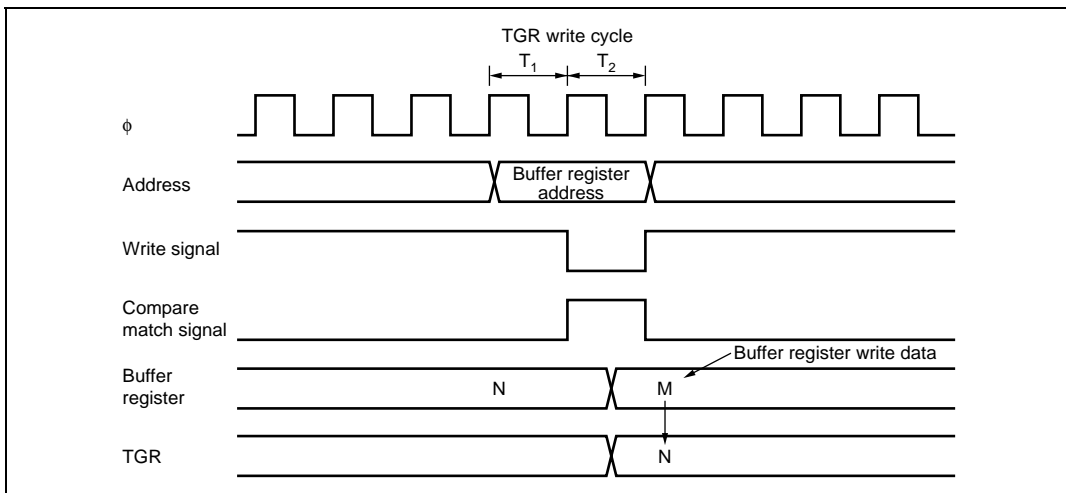


**Figure 10.47 Contention between TGR Write and Compare Match**

### 10.10.7 Contention between Buffer Register Write and Compare Match

If a compare match occurs in the  $T_2$  state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the data prior to the write.

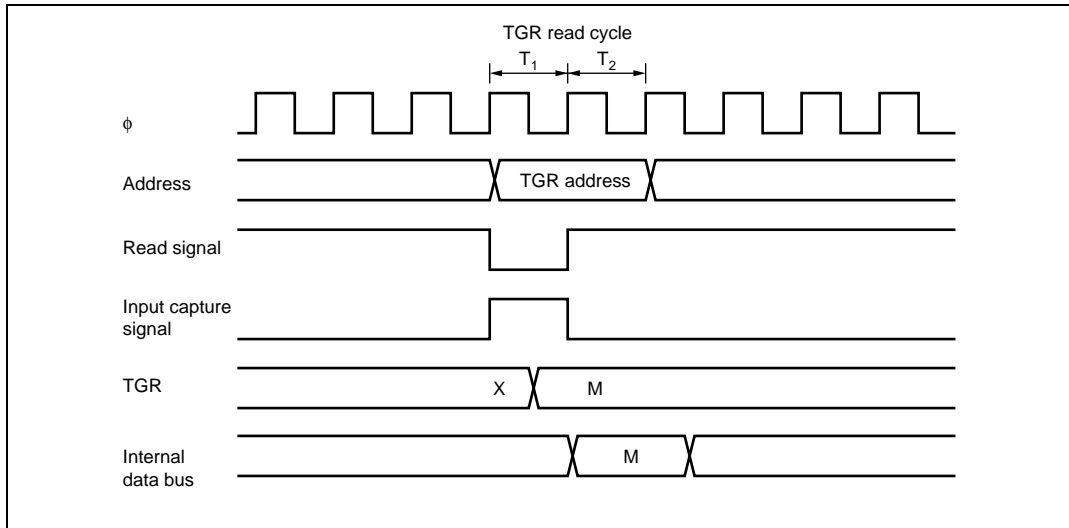
Figure 10.48 shows the timing in this case.



**Figure 10.48 Contention between Buffer Register Write and Compare Match**

If the input capture signal is generated in the  $T_1$  state of a TGR read cycle, the data that is read will be the data after input capture transfer.

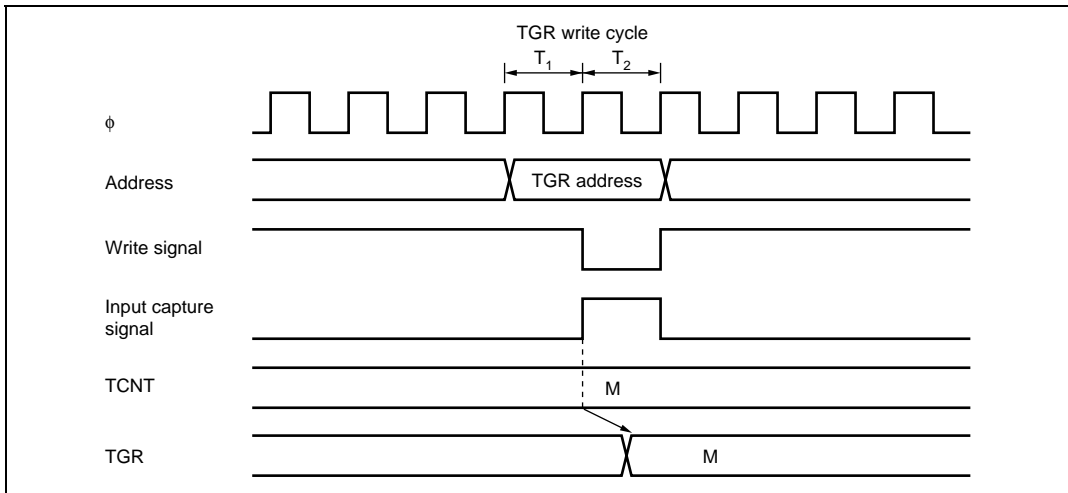
Figure 10.49 shows the timing in this case.



**Figure 10.49 Contention between TGR Read and Input Capture**

If the input capture signal is generated in the  $T_2$  state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 10.50 shows the timing in this case.

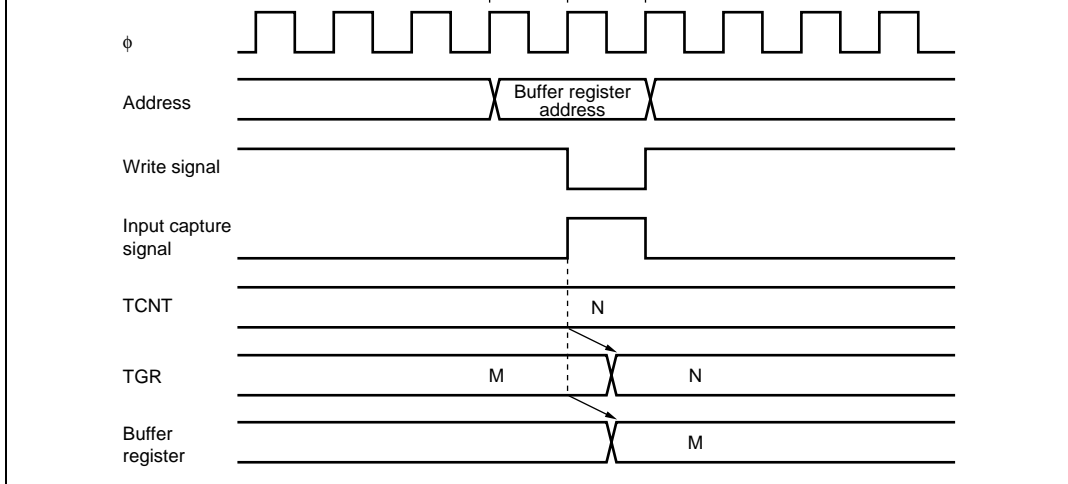


**Figure 10.50 Contention between TGR Write and Input Capture**

#### 10.10.10 Contention between Buffer Register Write and Input Capture

If the input capture signal is generated in the  $T_2$  state of a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 10.51 shows the timing in this case.

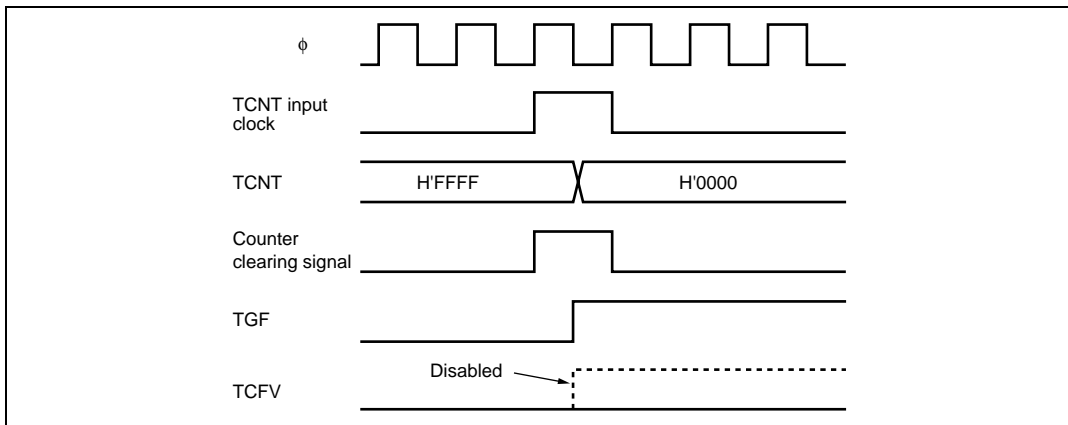


**Figure 10.51 Contention between Buffer Register Write and Input Capture**

### 10.10.11 Contention between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

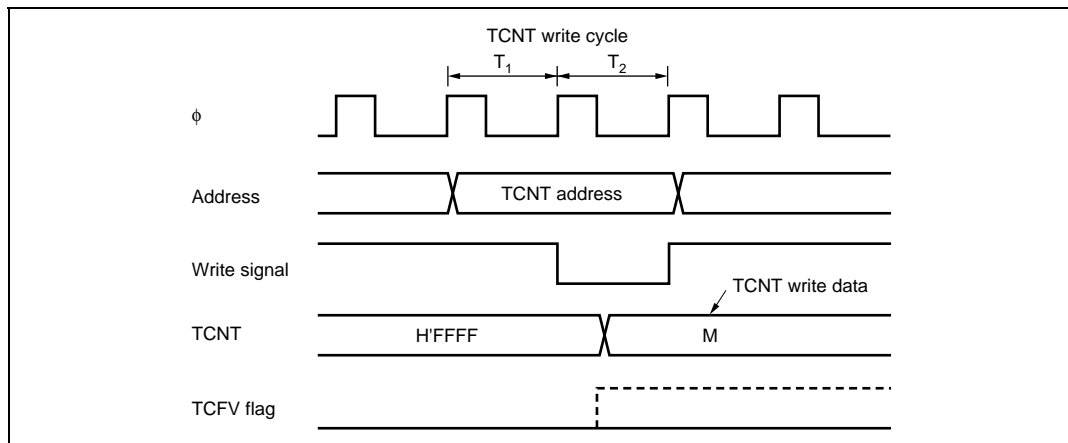
Figure 10.52 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.



**Figure 10.52 Contention between Overflow and Counter Clearing**

If there is an up-count or down-count in the  $T_2$  state of a TCNT write cycle, when overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10.53 shows the operation timing when there is contention between TCNT write and overflow.



**Figure 10.53 Contention between TCNT Write and Overflow**

### 10.10.13 Multiplexing of I/O Pins

In this LSI, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

### 10.10.14 Interrupts and Module Stop Mode

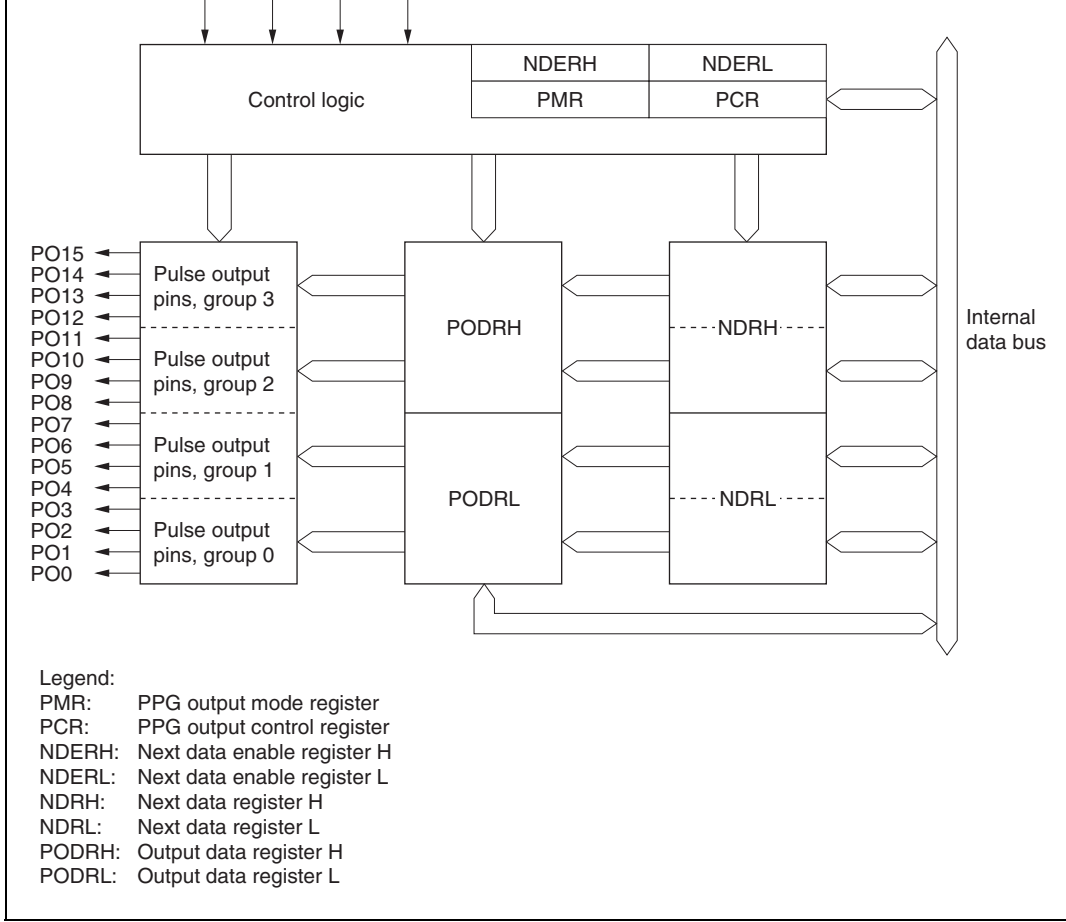
If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC or DTC activation source. Interrupts should therefore be disabled before entering module stop mode.



The programmable pulse generator (PPG) provides pulse outputs by using the 16-bit timer pulse unit (TPU) as a time base. The PPG pulse outputs are divided into 4-bit groups (groups 3 to 0) that can operate both simultaneously and independently. The block diagram of PPG is shown in figure 11.1

## 11.1 Features

- 16-bit output data
- Four output groups
- Selectable output trigger signals
- Non-overlap mode
- Can operate together with the data transfer controller (DTC) and the DMA controller (DMAC)
- Settable inverted output
- Module stop mode can be set



**Figure 11.1 Block Diagram of PPG**

Table 11.1 summarizes the I/O pins of the PPG.

**Table 11.1 Pin Configuration**

<b>Pin Name</b>	<b>I/O</b>	<b>Function</b>
PO15	Output	Group 3 pulse output
PO14	Output	
PO13	Output	
PO12	Output	
PO11	Output	Group 2 pulse output
PO10	Output	
PO9	Output	
PO8	Output	
PO7	Output	Group 1 pulse output
PO6	Output	
PO5	Output	
PO4	Output	
PO3	Output	Group 0 pulse output
PO2	Output	
PO1	Output	
PO0	Output	

### 11.3 Register Descriptions

The PPG has the following registers.

- Next data enable register H (NDERH)
- Next data enable register L (NDERL)
- Output data register H (PODRH)
- Output data register L (PODRL)
- Next data register H (NDRH)
- Next data register L (NDRL)
- PPG output control register (PCR)
- PPG output mode register (PMR)

NDERH, NDERL enable or disable pulse output on a bit-by-bit basis. For outputting pulse by the PPG, set the corresponding DDR to 1.

- NDREH

Bit	Bit Name	Initial Value	R/W	Description
7	NDER15	0	R/W	Next Data Enable 15 to 8
6	NDER14	0	R/W	When a bit is set to 1, the value in the corresponding NDRH bit is transferred to the PODRH bit by the selected output trigger. Values are not transferred from NDRH to PODRH for cleared bits.
5	NDER13	0	R/W	
4	NDER12	0	R/W	
3	NDER11	0	R/W	
2	NDER10	0	R/W	
1	NDER9	0	R/W	
0	NDER8	0	R/W	

- NDERL

Bit	Bit Name	Initial Value	R/W	Description
7	NDER7	0	R/W	Next Data Enable 7 to 0
6	NDER6	0	R/W	When a bit is set to 1, the value in the corresponding NDRL bit is transferred to the PODRL bit by the selected output trigger. Values are not transferred from NDRL to PODRL for cleared bits.
5	NDER5	0	R/W	
4	NDER4	0	R/W	
3	NDER3	0	R/W	
2	NDER2	0	R/W	
1	NDER1	0	R/W	
0	NDER0	0	R/W	

PODRH and PODRL store output data for use in pulse output. A bit that has been set for pulse output by NDER is read-only and cannot be modified.

- PODRH

Bit	Bit Name	Initial Value	R/W	Description
7	POD15	0	R/W	Output Data Register 15 to 8
6	POD14	0	R/W	For bits which have been set to pulse output by NDERH, the output trigger transfers NDRH values to this register during PPG operation. While NDERH is set to 1, the CPU cannot write to this register. While NDERH is cleared, the initial output value of the pulse can be set.
5	POD13	0	R/W	
4	POD12	0	R/W	
3	POD11	0	R/W	
2	POD10	0	R/W	
1	POD9	0	R/W	
0	POD8	0	R/W	

- PODRL

Bit	Bit Name	Initial Value	R/W	Description
7	POD7	0	R/W	Output Data Register 7 to 0
6	POD6	0	R/W	For bits which have been set to pulse output by NDERL, the output trigger transfers NDRL values to this register during PPG operation. While NDERL is set to 1, the CPU cannot write to this register. While NDERL is cleared, the initial output value of the pulse can be set.
5	POD5	0	R/W	
4	POD4	0	R/W	
3	POD3	0	R/W	
2	POD2	0	R/W	
1	POD1	0	R/W	
0	POD0	0	R/W	

NDRH, NDRL store the next data for pulse output. The NDR addresses differ depending on whether pulse output groups have the same output trigger or different output triggers.

- NDRH

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 8
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3	NDR11	0	R/W	
2	NDR10	0	R/W	
1	NDR9	0	R/W	
0	NDR8	0	R/W	

If pulse output groups 2 and 3 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR15	0	R/W	Next Data Register 15 to 12
6	NDR14	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
5	NDR13	0	R/W	
4	NDR12	0	R/W	
3 to 0	—	All 1	—	

to 4				These bits are always read as 1 and cannot be modified.
3	NDR11	0	R/W	Next Data Register 11 to 8
2	NDR10	0	R/W	The register contents are transferred to the corresponding PODRH bits by the output trigger specified with PCR.
1	NDR9	0	R/W	
0	NDR8	0	R/W	

- NDRL

If pulse output groups 0 and 1 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time, as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 0
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3	NDR3	0	R/W	
2	NDR2	0	R/W	
1	NDR1	0	R/W	
0	NDR0	0	R/W	

If pulse output groups 0 and 1 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses as shown below.

Bit	Bit Name	Initial Value	R/W	Description
7	NDR7	0	R/W	Next Data Register 7 to 4
6	NDR6	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
5	NDR5	0	R/W	
4	NDR4	0	R/W	
3 to 0	—	All 1	—	Reserved  These bits are always read as 1 and cannot be modified.

to 4				These bits are always read as 1 and cannot be modified.
3	NDR3	0	R/W	Next Data Register 3 to 0
2	NDR2	0	R/W	The register contents are transferred to the corresponding PODRL bits by the output trigger specified with PCR.
1	NDR1	0	R/W	
0	NDR0	0	R/W	

### 11.3.4 PPG Output Control Register (PCR)

PCR selects output trigger signals on a group-by-group basis. For details on output trigger selection, refer to section 11.3.5, PPG Output Mode Register (PMR).

Bit	Bit Name	Initial Value	R/W	Description
7	G3CMS1	1	R/W	Group 3 Compare Match Select 1 and 0
6	G3CMS0	1	R/W	Select output trigger of pulse output group 3. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
5	G2CMS1	1	R/W	Group 2 Compare Match Select 1 and 0
4	G2CMS0	1	R/W	Select output trigger of pulse output group 2. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
3	G1CMS1	1	R/W	Group 1 Compare Match Select 1 and 0
2	G1CMS0	1	R/W	Select output trigger of pulse output group 1. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3



0	G0CMS0	1	R/W	Select output trigger of pulse output group 0. 00: Compare match in TPU channel 0 01: Compare match in TPU channel 1 10: Compare match in TPU channel 2 11: Compare match in TPU channel 3
---	--------	---	-----	--

---

### 11.3.5 PPG Output Mode Register (PMR)

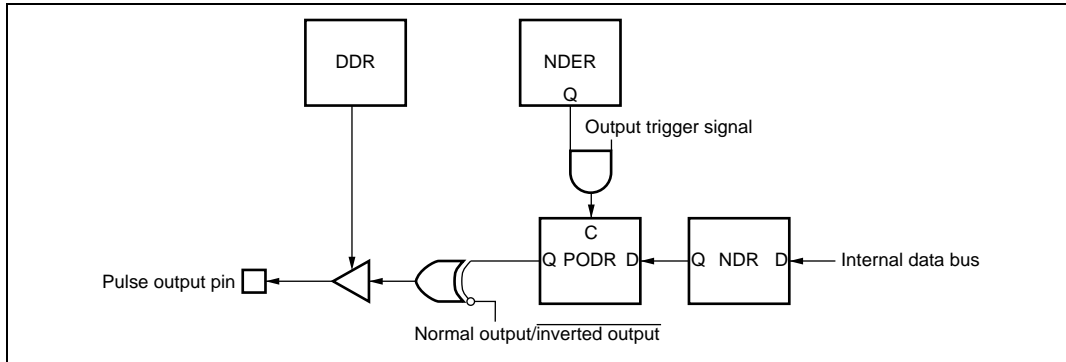
PMR selects the pulse output mode of the PPG for each group. If inverted output is selected, a low-level pulse is output when PODRH is 1 and a high-level pulse is output when PODRH is 0. If non-overlapping operation is selected, PPG updates its output values at compare match A or B of the TPU that becomes the output trigger. For details, refer to section 11.4.4, Non-Overlapping Pulse Output.

Bit	Bit Name	Initial Value	R/W	Description
7	G3INV	1	R/W	Group 3 Inversion Selects direct output or inverted output for pulse output group 3. 0: Inverted output 1: Direct output
6	G2INV	1	R/W	Group 2 Inversion Selects direct output or inverted output for pulse output group 2. 0: Inverted output 1: Direct output
5	G1INV	1	R/W	Group 1 Inversion Selects direct output or inverted output for pulse output group 1. 0: Inverted output 1: Direct output
4	G0INV	1	R/W	Group 0 Inversion Selects direct output or inverted output for pulse output group 0. 0: Inverted output 1: Direct output

---

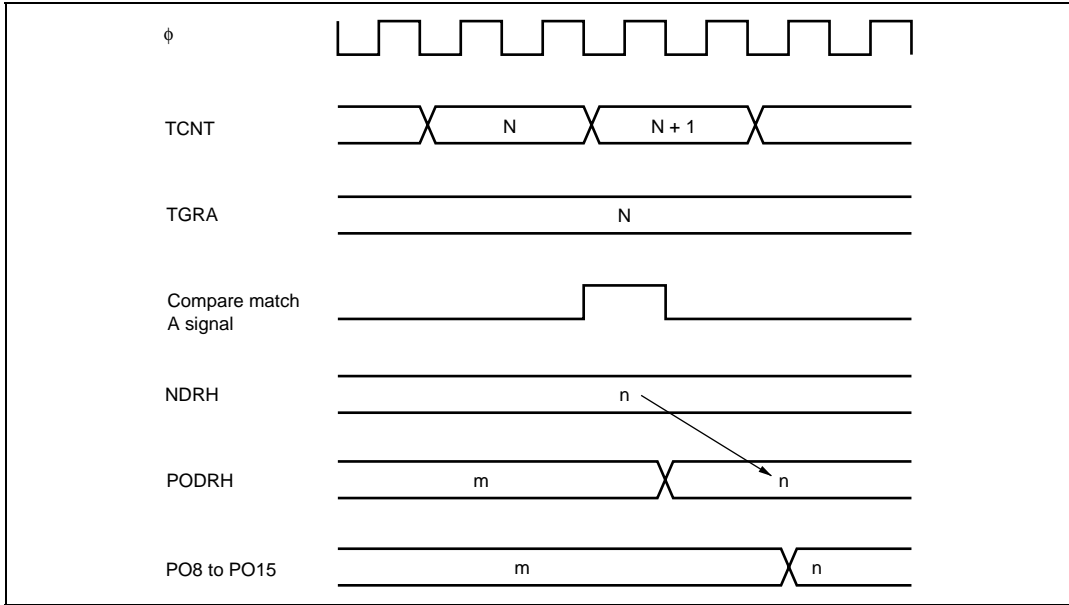
				<p>Selects normal or non-overlapping operation for pulse output group 3.</p> <p>0: Normal operation (output values updated at compare match A in the selected TPU channel)</p> <p>1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)</p>
2	G2NOV	0	R/W	<p>Group 2 Non-Overlap</p> <p>Selects normal or non-overlapping operation for pulse output group 2.</p> <p>0: Normal operation (output values updated at compare match A in the selected TPU channel)</p> <p>1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)</p>
1	G1NOV	0	R/W	<p>Group 1 Non-Overlap</p> <p>Selects normal or non-overlapping operation for pulse output group 1.</p> <p>0: Normal operation (output values updated at compare match A in the selected TPU channel)</p> <p>1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)</p>
0	G0NOV	0	R/W	<p>Group 0 Non-Overlap</p> <p>Selects normal or non-overlapping operation for pulse output group 0.</p> <p>0: Normal operation (output values updated at compare match A in the selected TPU channel)</p> <p>1: Non-overlapping operation (output values updated at compare match A or B in the selected TPU channel)</p>

Figure 11.2 shows an overview diagram of the PPG. PPG pulse output is enabled when the corresponding bits in P1DDR, P2DDR, and NDER are set to 1. An initial output value is determined by its corresponding PODR initial setting. When the compare match event specified by PCR occurs, the corresponding NDR bit contents are transferred to PODR to update the output values. Sequential output of data of up to 16 bits is possible by writing new output data to NDR before the next compare match.



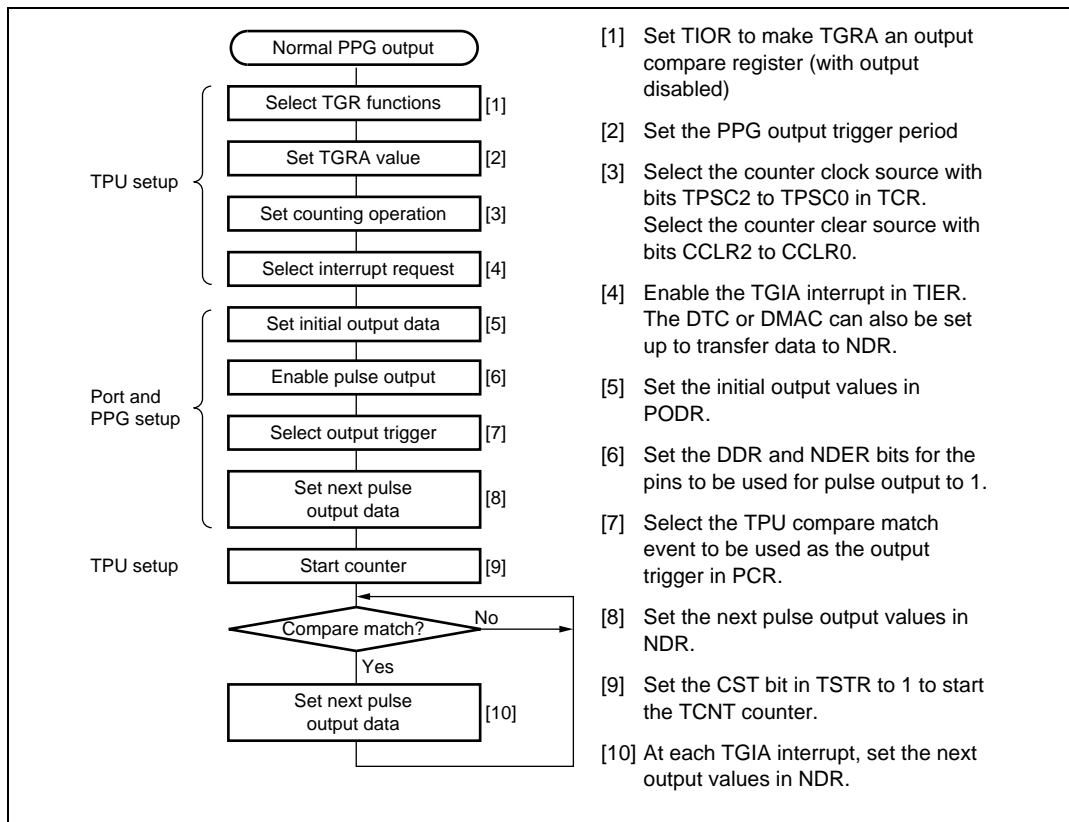
**Figure 11.2 Overview Diagram of PPG**

If pulse output is enabled, NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 11.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.



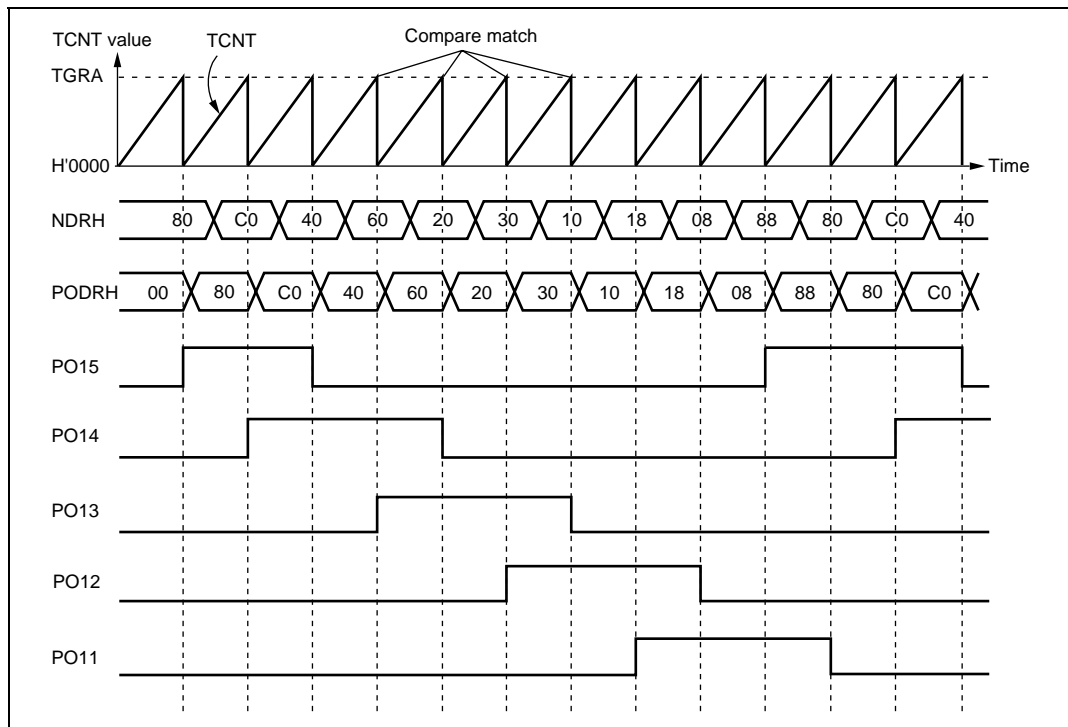
**Figure 11.3 Timing of Transfer and Output of NDR Contents (Example)**

Figure 11.4 shows a sample procedure for setting up normal pulse output.



**Figure 11.4 Setup Procedure for Normal Pulse Output (Example)**

Figure 11.5 shows an example in which pulse output is used for cyclic five-phase pulse output.



**Figure 11.5 Normal Pulse Output Example (Five-Phase Pulse Output)**

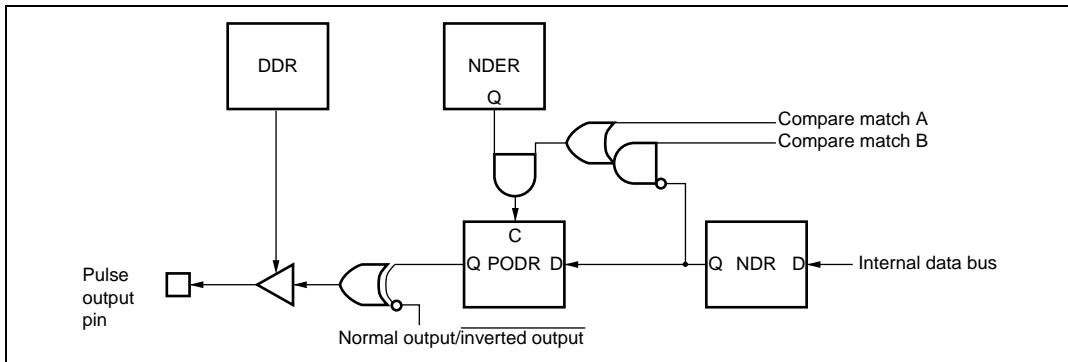
1. Set up TGRA in TPU which is used as the output trigger to be an output compare register. Set a cycle in TGRA so the counter will be cleared by compare match A. Set the TGIEA bit in TIER to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write H'F8 in P1DDR and NDRH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.
3. The timer counter in the TPU channel starts. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.
4. Five-phase pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88... at successive TGIA interrupts.

## 11.4.4 Non-Overlapping Pulse Output

During non-overlapping operation, transfer from NDR to PODR is performed as follows:

- NDR bits are always transferred to PODR bits at compare match A.
- At compare match B, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 11.6 illustrates the non-overlapping pulse output operation.



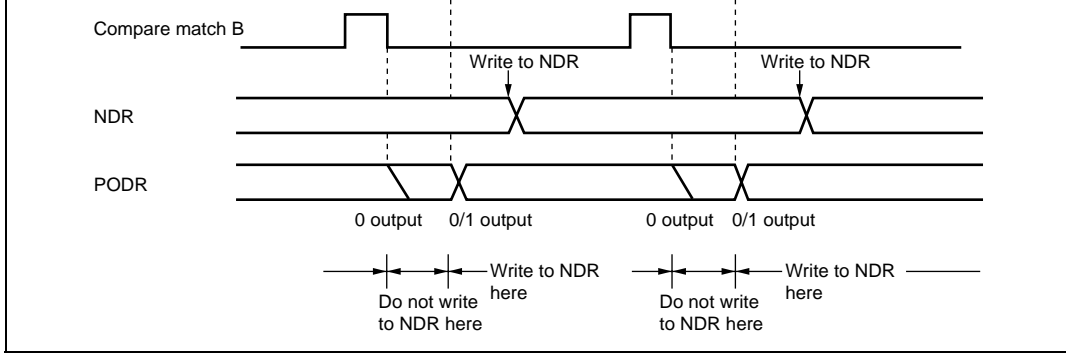
**Figure 11.6 Non-Overlapping Pulse Output**

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A.

The NDR contents should not be altered during the interval from compare match B to compare match A (the non-overlap margin).

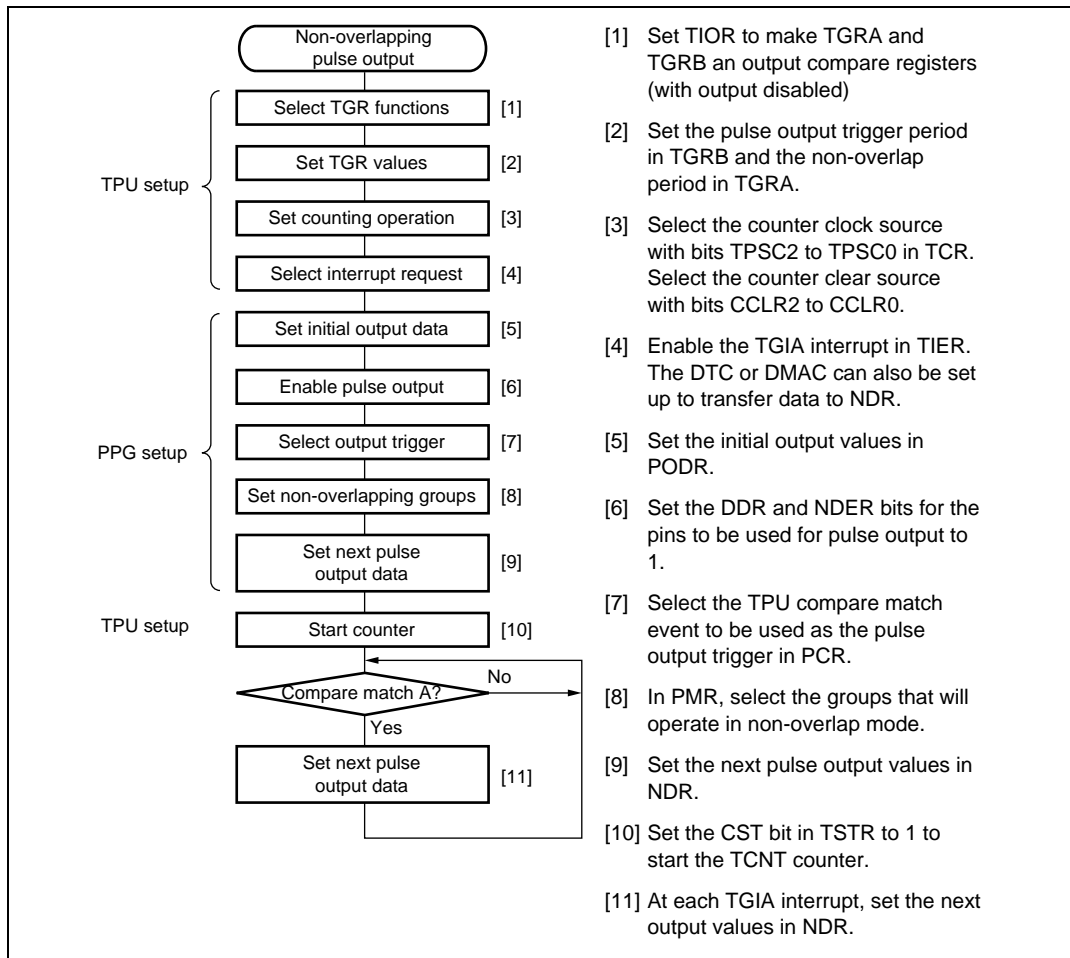
This can be accomplished by having the TGIA interrupt handling routine write the next data in NDR, or by having the TGIA interrupt activate the DTC or DMAC. Note, however, that the next data must be written before the next compare match B occurs.

Figure 11.7 shows the timing of this operation.



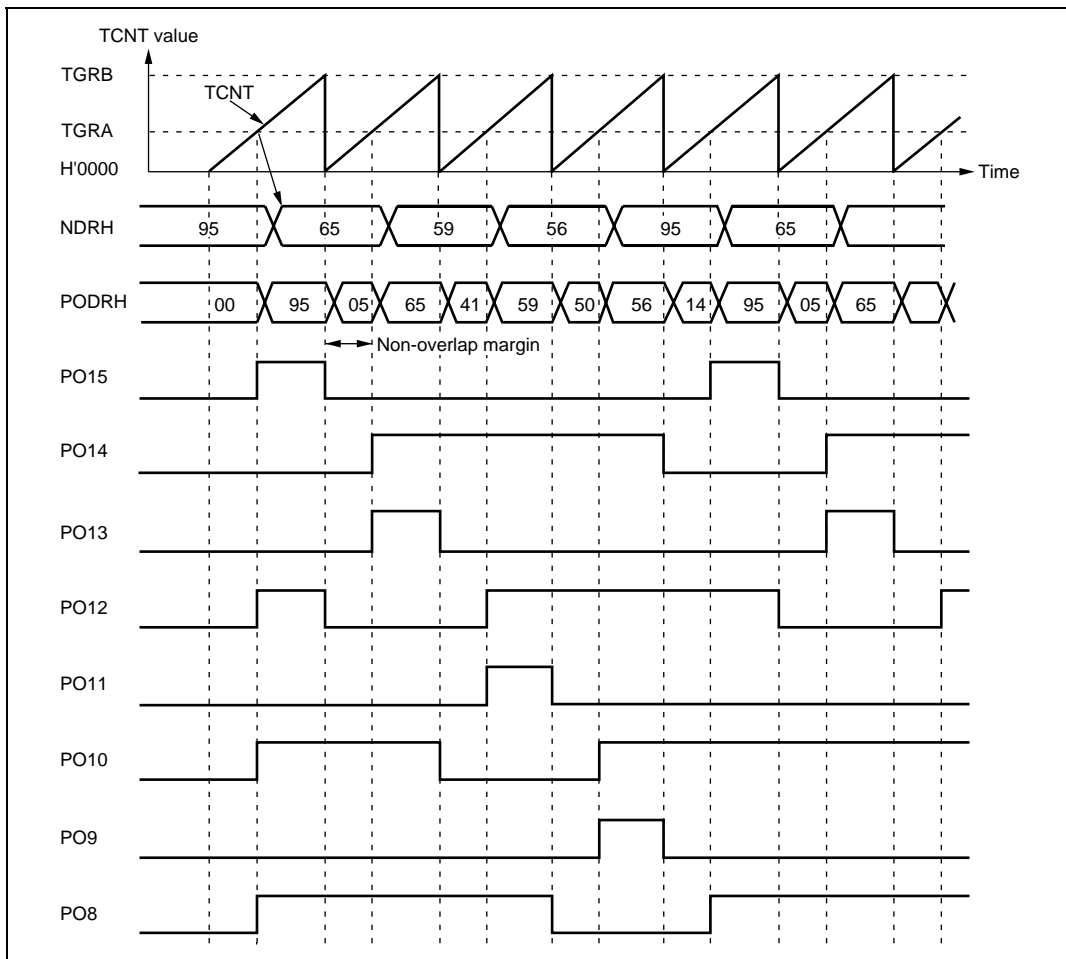
**Figure 11.7 Non-Overlapping Operation and NDR Write Timing**





**Figure 11.8 Setup Procedure for Non-Overlapping Pulse Output (Example)**

Figure 11.9 shows an example in which pulse output is used for four-phase complementary non-overlapping pulse output.



**Figure 11.9 Non-Overlapping Pulse Output Example (Four-Phase Complementary)**

1. Set up the TPU channel to be used as the output trigger channel so that TGRA and TGRB are output compare registers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the TGIEA bit in TIER to 1 to enable the TGIA interrupt.

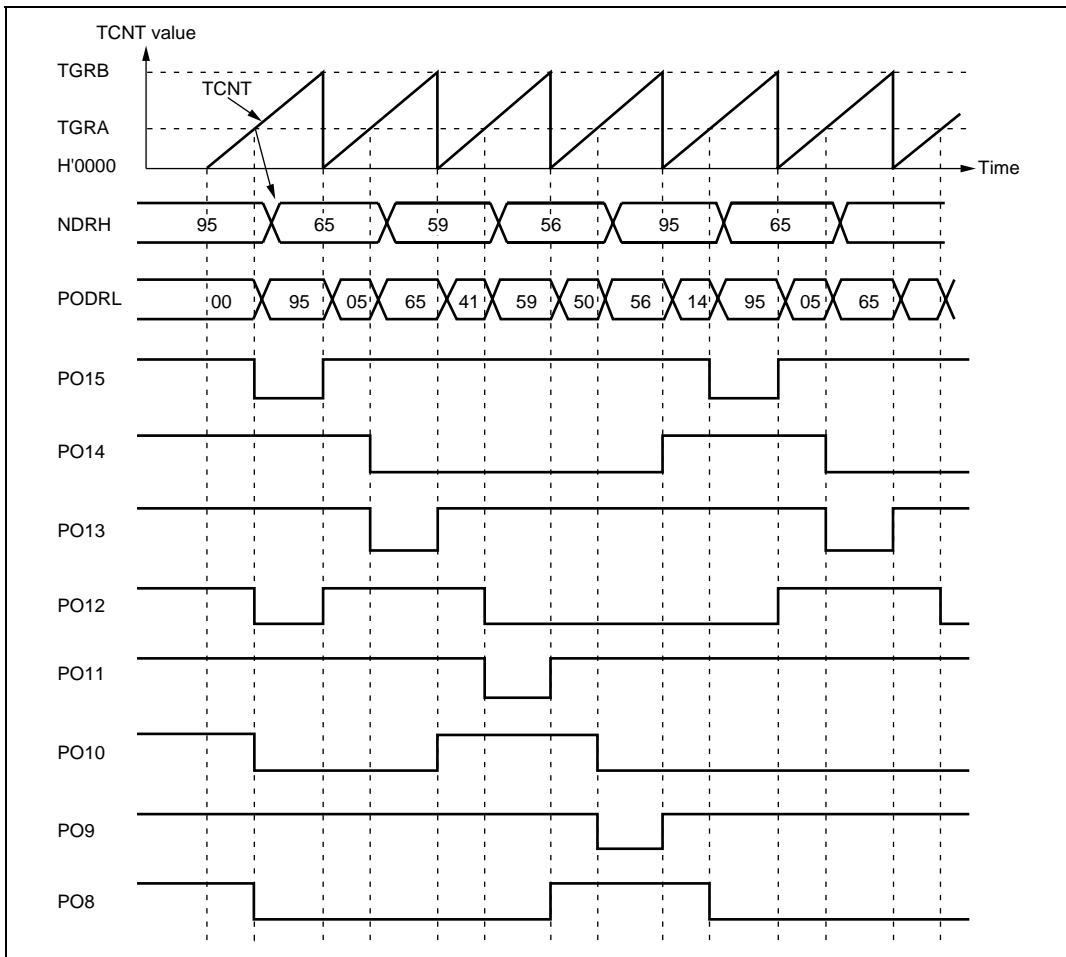
output trigger. Set the G3NOV and G2NOV bits in PMR to 1 to select non-overlapping output. Write output data H'95 in NDRH.

3. The timer counter in the TPU channel starts. When a compare match with TGRB occurs, outputs change from 1 to 0. When a compare match with TGRA occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interrupt handling routine writes the next output data (H'65) in NDRH.
4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing H'59, H'56, H'95... at successive TGIA interrupts.

If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

If the G3INV, G2INV, G1INV, and G0INV bits in PMR are cleared to 0, values that are the inverse of the PODR contents can be output.

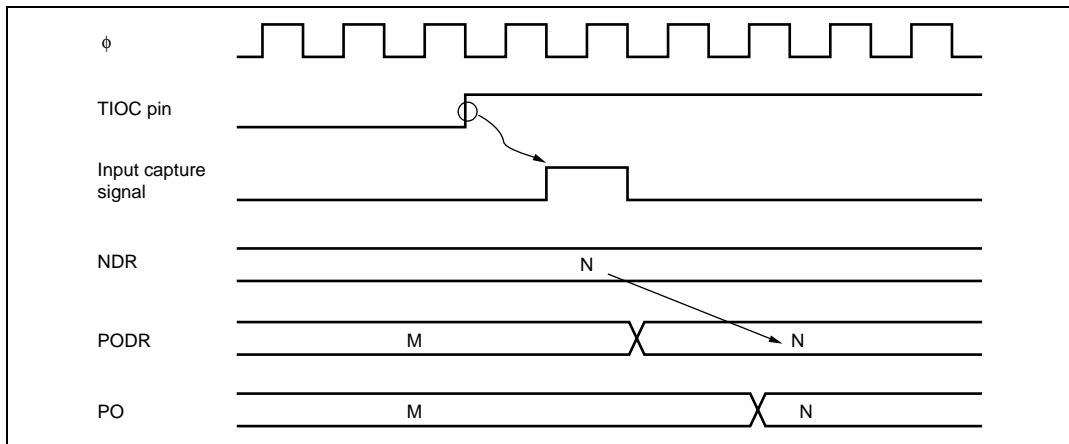
Figure 11.10 shows the outputs when G3INV and G2INV are cleared to 0, in addition to the settings of figure 11.9.



**Figure 11.10 Inverted Pulse Output (Example)**

Pulse output can be triggered by TPU input capture as well as by compare match. If TGRA functions as an input capture register in the TPU channel selected by PCR, pulse output will be triggered by the input capture signal.

Figure 11.11 shows the timing of this output.



**Figure 11.11 Pulse Output Triggered by Input Capture (Example)**

## 11.5 Usage Notes

### 11.5.1 Module Stop Mode Setting

PPG operation can be disabled or enabled using the module stop control register. The initial value is for PPG operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 22, Power-Down Modes.

### 11.5.2 Operation of Pulse Output Pins

Pins PO0 to PO15 are also used for other peripheral functions such as the TPU. When output by another peripheral function is enabled, the corresponding pins cannot be used for pulse output. Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of the usage of the pins.

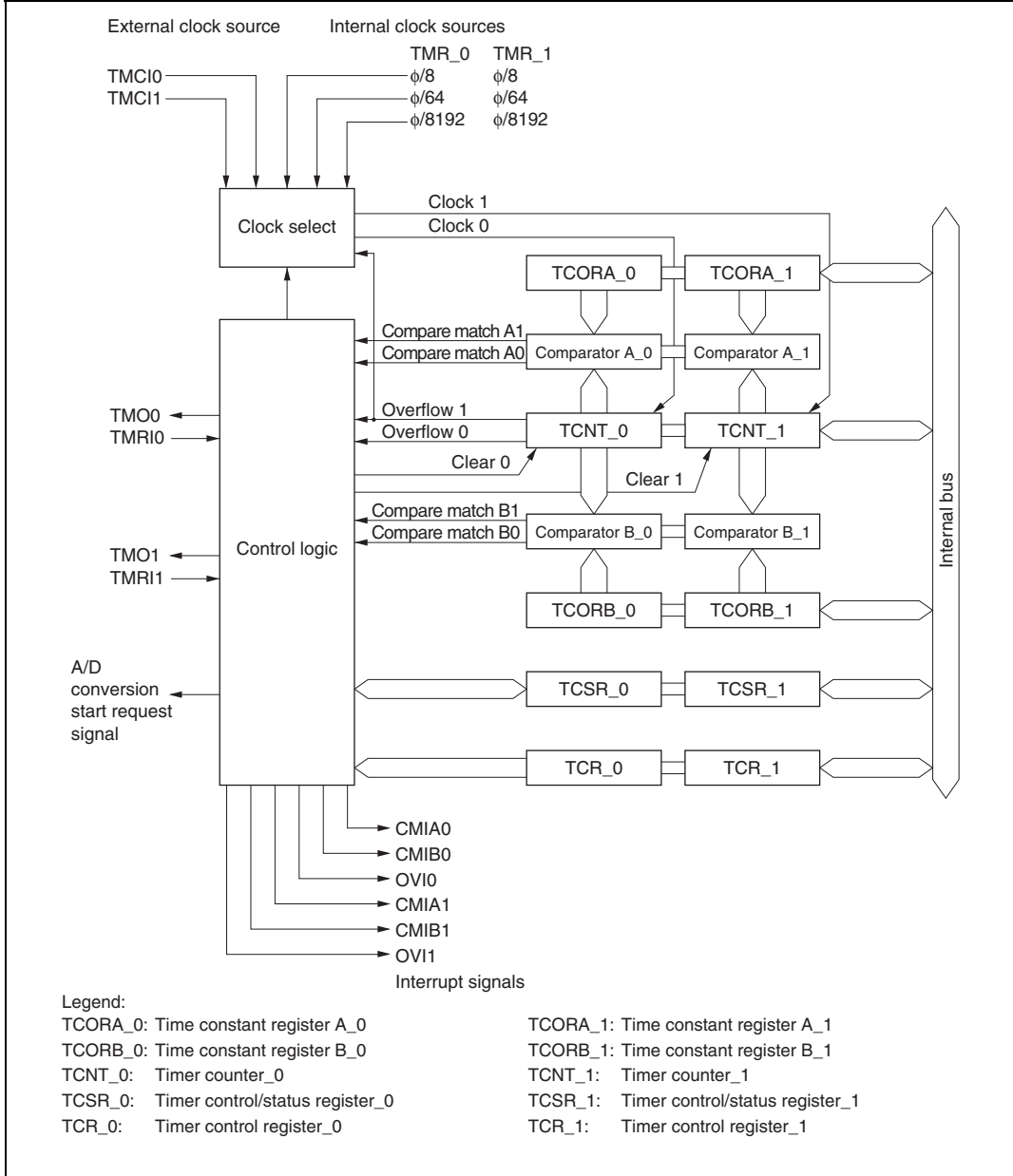
Pin functions should be changed only under conditions in which the output trigger event will not occur.



This LSI has an on-chip 8-bit timer module with two channels operating on the basis of an 8-bit counter. The 8-bit timer module can be used to count external events and be used as a multifunction timer in a variety of applications, such as generation of counter reset, interrupt requests, and pulse output with an arbitrary duty cycle using a compare-match signal with two registers.

## 12.1 Features

- Selection of four clock sources  
The counters can be driven by one of three internal clock signals ( $\phi/8$ ,  $\phi/64$ , or  $\phi/8192$ ) or an external clock input
- Selection of three ways to clear the counters  
The counters can be cleared on compare match A or B, or by an external reset signal
- Timer output control by a combination of two compare match signals  
The timer output signal in each channel is controlled by a combination of two independent compare match signals, enabling the timer to generate output waveforms with an arbitrary duty cycle or PWM output
- Provision for cascading of two channels (TMR\_0 and TMR\_1)  
Operation as a 16-bit timer is possible, using TMR\_0 for the upper 8 bits and TMR\_1 for the lower 8 bits (16-bit count mode)  
TMR\_1 can be used to count TMR\_0 compare matches (compare match count mode)
- Three independent interrupts  
Compare match A and B and overflow interrupts can be requested independently
- A/D converter conversion start trigger can be generated



**Figure 12.1 Block Diagram of 8-Bit Timer Module**



Table 12.1 summarizes the pins of the 8-bit timer module.

**Table 12.1 Pin Configuration**

Channel	Name	Symbol	I/O	Function
0	Timer output pin	TMO0	Output	Outputs at compare match
	Timer clock input pin	TMC10	Input	Inputs external clock for counter
	Timer reset input pin	TMRI0	Input	Inputs external reset to counter
1	Timer output pin	TMO1	Output	Outputs at compare match
	Timer clock input pin	TMC11	Input	Inputs external clock for counter
	Timer reset input pin	TMRI1	Input	Inputs external reset to counter

## 12.3 Register Descriptions

The 8-bit timer module has the following registers. For details on the module stop control register, refer to section 23.1.2, Module Stop Control Registers H, L (MSTPCRH, MSTPCRL).

- Timer counter\_0 (TCNT\_0)
- Time constant register A\_0 (TCORA\_0)
- Time constant register B\_0 (TCORB\_0)
- Timer control register\_0 (TCR\_0)
- Timer control/status register\_0 (TCSR\_0)
- Timer counter\_1 (TCNT\_1)
- Time constant register A\_1 (TCORA\_1)
- Time constant register B\_1 (TCORB\_1)
- Timer control register\_1 (TCR\_1)
- Timer control/status register\_1 (TCSR\_1)

### 12.3.1 Timer Counter (TCNT)

TCNT is 8-bit up-counter. TCNT\_0 and TCNT\_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction. Bits CKS2 to CKS0 in TCR are used to select a clock. TCNT can be cleared by an external reset input or by a compare match signal A or B. Which signal is to be used for clearing is selected by bits CCLR1 and CCLR0 in TCR. When TCNT overflows from H'FF to H'00, OVF in TCSR is set to 1. TCNT is initialized to H'00.

TCORA is 8-bit readable/writable register. TCORA\_0 and TCORA\_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag in TCSR is set to 1. Note, however, that comparison is disabled during the  $T_2$  state of a TCORA write cycle.

The timer output from the TMO pin can be freely controlled by this compare match signal (compare match A) and the settings of bits OS1 and OS0 in TCSR.

TCORA is initialized to H'FF.

### **12.3.3 Time Constant Register B (TCORB)**

TCORB is 8-bit readable/writable register. TCORB\_0 and TCORB\_1 comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding CMFB flag in TCSR is set to 1. Note, however, that comparison is disabled during the  $T_2$  state of a TCOBR write cycle.

The timer output from the TMO pin can be freely controlled by this compare match signal (compare match B) and the settings of bits OS3 and OS2 in TCSR.

TCORB is initialized to H'FF.

### **12.3.4 Timer Control Register (TCR)**

TCR selects the clock source and the time at which TCNT is cleared, and controls interrupts.

				<p>Selects whether CMFB interrupt requests (CMIB) are enabled or disabled when the CMFB flag in TCSR is set to 1.</p> <p>0: CMFB interrupt requests (CMIB) are disabled 1: CMFB interrupt requests (CMIB) are enabled</p>
6	CMIEA	0	R/W	<p>Compare Match Interrupt Enable A</p> <p>Selects whether CMFA interrupt requests (CMIA) are enabled or disabled when the CMFA flag in TCSR is set to 1.</p> <p>0: CMFA interrupt requests (CMIA) are disabled 1: CMFA interrupt requests (CMIA) are enabled</p>
5	OVIE	0	R/W	<p>Timer Overflow Interrupt Enable</p> <p>Selects whether OVF interrupt requests (OVI) are enabled or disabled when the OVF flag in TCSR is set to 1.</p> <p>0: OVF interrupt requests (OVI) are disabled 1: OVF interrupt requests (OVI) are enabled</p>
4	CCLR1	0	R/W	Counter Clear 1 and 0
3	CCLR0	0	R/W	<p>These bits select the method by which TCNT is cleared</p> <p>00: Clearing is disabled 01: Clear by compare match A 10: Clear by compare match B 11: Clear by rising edge of external reset input</p>
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	These bits select the clock input to TCNT and count condition. See table 12.2.
0	CKS0	0	R/W	

Channel	Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Description
TMR_0	0		0	Clock input disabled
			1	Internal clock, counted at falling edge of $\phi/8$
			0	Internal clock, counted at falling edge of $\phi/64$
			1	Internal clock, counted at falling edge of $\phi/8192$
	1	0	0	Count at TCNT_1 overflow signal*
TMR_1	0		0	Clock input disabled
			1	Internal clock, counted at falling edge of $\phi/8$
			0	Internal clock, counted at falling edge of $\phi/64$
			1	Internal clock, counted at falling edge of $\phi/8192$
	1	0	0	Count at TCNT_0 compare match A*
All	1	0	1	External clock, counted at rising edge
			0	External clock, counted at falling edge
			1	External clock, counted at both rising and falling edges

Note: \* If the count input of TMR\_0 is the TCNT\_1 overflow signal and that of TMR\_1 is the TCNT\_0 compare match signal, no incrementing clock is generated. Do not use this setting.

### 12.3.5 Timer Control/Status Register (TCSR)

TCSR displays status flags, and controls compare match output.

TCSR\_0

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare Match Flag B [Setting condition] Set when TCNT matches TCORB [Clearing conditions] <ul style="list-style-type: none"> <li>Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB</li> <li>When DTC is activated by CMIB interrupt while DISEL bit of MRB in DTC is 0</li> </ul>

				<p>[Setting condition]</p> <p>Set when TCNT matches TCORA</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA</li> <li>• When DTC is activated by CMIA interrupt while DISEL bit of MRB in DTC is 0</li> </ul>
5	OVF	0	R/(W)*	<p>Timer Overflow Flag</p> <p>[Setting condition]</p> <p>Set when TCNT overflows from H'FF to H'00</p> <p>[Clearing condition]</p> <p>Cleared by reading OVF when OVF = 1, then writing 0 to OVF</p>
4	ADTE	0	R/W	<p>A/D Trigger Enable</p> <p>Selects enabling or disabling of A/D converter start requests by compare match A.</p> <p>0: A/D converter start requests by compare match A are disabled</p> <p>1: A/D converter start requests by compare match A are enabled</p>
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	<p>These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs.</p> <p>00: No change when compare match B occurs</p> <p>01: 0 is output when compare match B occurs</p> <p>10: 1 is output when compare match B occurs</p> <p>11: Output is inverted when compare match B occurs (toggle output)</p>

0	OS0	0	R/W	These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs. 00: No change when compare match A occurs 01: 0 is output when compare match A occurs 10: 1 is output when compare match A occurs 11: Output is inverted when compare match A occurs (toggle output)
---	-----	---	-----	--

Note: \* Only 0 can be written to bits 7 to 5, to clear these flags.

### TCSR\_1

Bit	Bit Name	Initial Value	R/W	Description
7	CMFB	0	R/(W)*	Compare Match Flag B [Setting condition] Set when TCNT matches TCORB [Clearing conditions] <ul style="list-style-type: none"> <li>• Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB</li> <li>• When DTC is activated by CMIB interrupt while DISEL bit of MRB in DTC is 0</li> </ul>
6	CMFA	0	R/(W)*	Compare Match Flag A [Setting condition] Set when TCNT matches TCORA [Clearing conditions] <ul style="list-style-type: none"> <li>• Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA</li> <li>• When DTC is activated by CMIA interrupt while DISEL bit of MRB in DTC is 0</li> </ul>
5	OVF	0	R/(W)*	Timer Overflow Flag [Setting condition] Set when TCNT overflows from H'FF to H'00 [Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 to OVF

				This bit is always read as 1 and cannot be modified.
3	OS3	0	R/W	Output Select 3 and 2
2	OS2	0	R/W	These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs. 00: No change when compare match B occurs 01: 0 is output when compare match B occurs 10: 1 is output when compare match B occurs 11: Output is inverted when compare match B occurs (toggle output)
1	OS1	0	R/W	Output Select 1 and 0
0	OS0	0	R/W	These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs. 00: No change when compare match A occurs 01: 0 is output when compare match A occurs 10: 1 is output when compare match A occurs 11: Output is inverted when compare match A occurs (toggle output)

Note: \* Only 0 can be written to bits 7 to 5, to clear these flags.

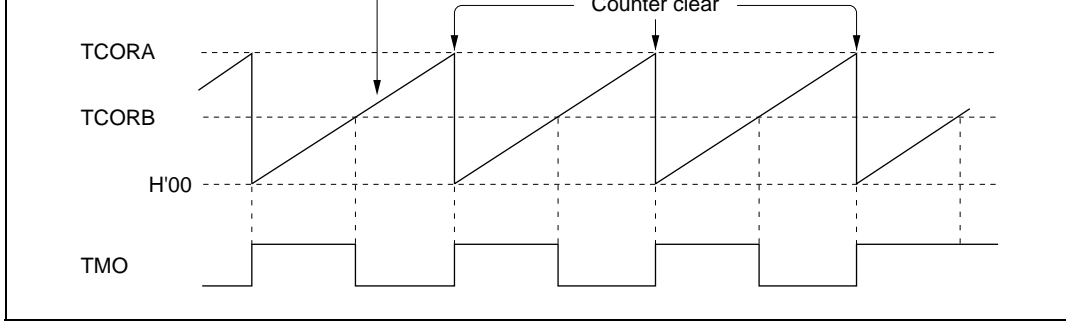
## 12.4 Operation

### 12.4.1 Pulse Output

Figure 12.2 shows an example that the 8-bit timer is used to generate a pulse output with a selected duty cycle. The control bits are set as follows:

- [1] In TCR, bit CCLR1 is cleared to 0 and bit CCLR0 is set to 1 so that the timer counter is cleared at a TCORA compare match.
- [2] In TCSR, bits OS3 to OS0 are set to B'0110, causing the output to change to 1 at a TCORA compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides output of pulses at a rate determined by TCORA with a pulse width determined by TCORB. No software intervention is required.

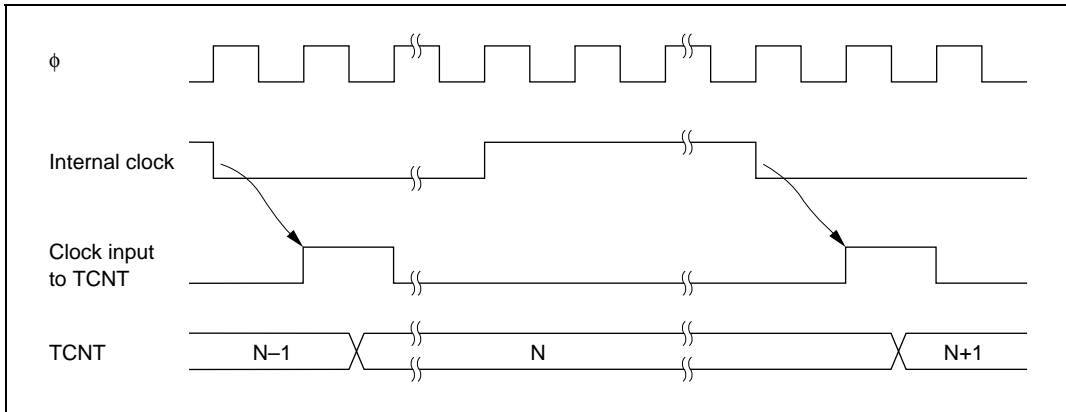


**Figure 12.2 Example of Pulse Output**

## 12.5 Operation Timing

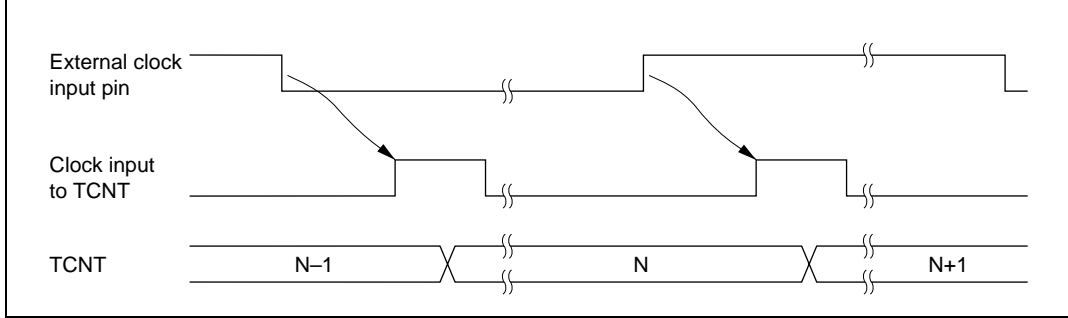
### 12.5.1 TCNT Incrementation Timing

Figure 12.3 shows the count timing for internal clock input. Figure 12.4 shows the count timing for external clock signal. Note that the external clock pulse width must be at least 1.5 states for incrementation at a single edge, and at least 2.5 states for incrementation at both edges. The counter will not increment correctly if the pulse width is less than these values.



**Figure 12.3 Count Timing for Internal Clock Input**

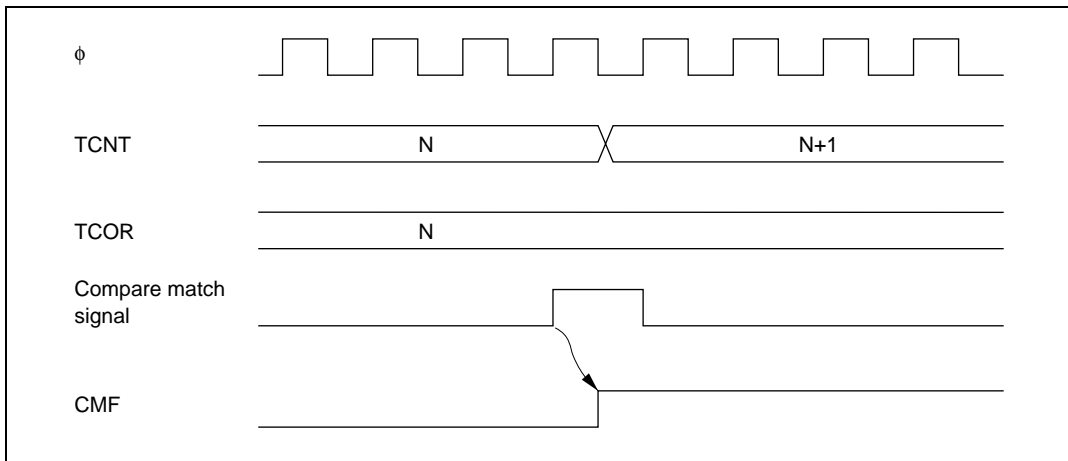




**Figure 12.4 Count Timing for External Clock Input**

### 12.5.2 Timing of CMFA and CMFB Setting when Compare-Match Occurs

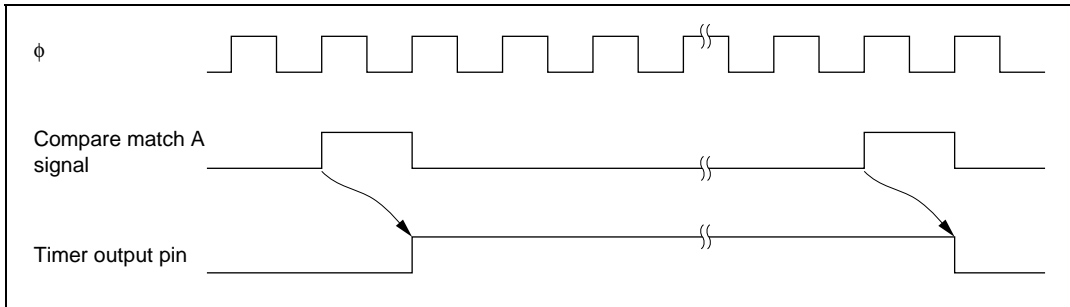
The CMFA and CMFB flags in TCSR are set to 1 by a compare match signal generated when the TCOR and TCNT values match. The compare match signal is generated at the last state in which the match is true, just before the timer counter is updated. Therefore, when TCOR and TCNT match, the compare match signal is not generated until the next incrementation clock input. Figure 12.5 shows this timing.



**Figure 12.5 Timing of CMF Setting**

When compare match A or B occurs, the timer output changes as specified by bits OS3 to OS0 in TCSR.

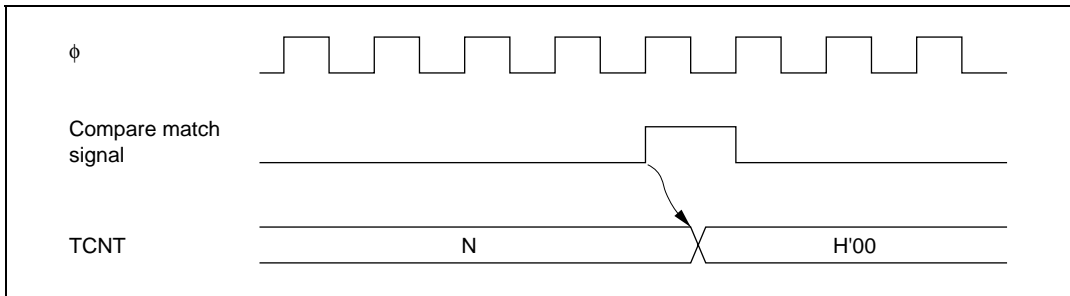
Figure 12.6 shows the timing when the output is set to toggle at compare match A.



**Figure 12.6 Timing of Timer Output**

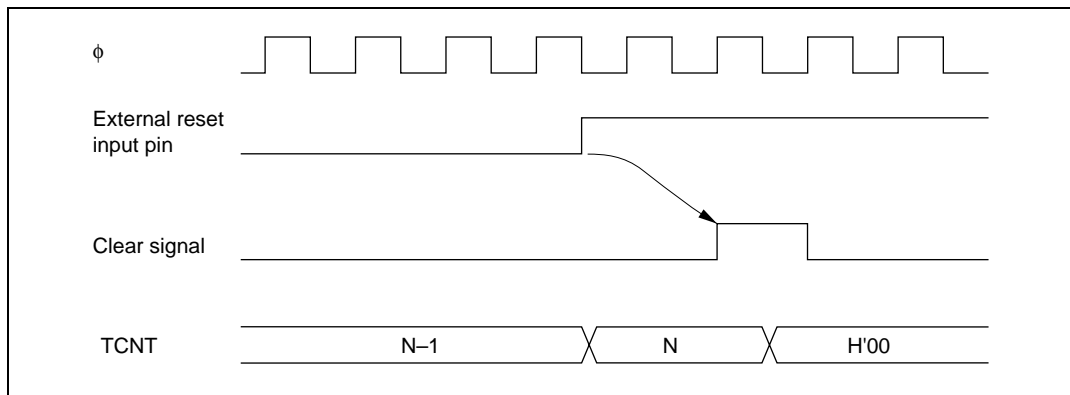
#### 12.5.4 Timing of Compare Match Clear

TCNT is cleared when compare match A or B occurs, depending on the setting of the CCLR1 and CCLR0 bits in TCR. Figure 12.7 shows the timing of this operation.



**Figure 12.7 Timing of Compare Match Clear**

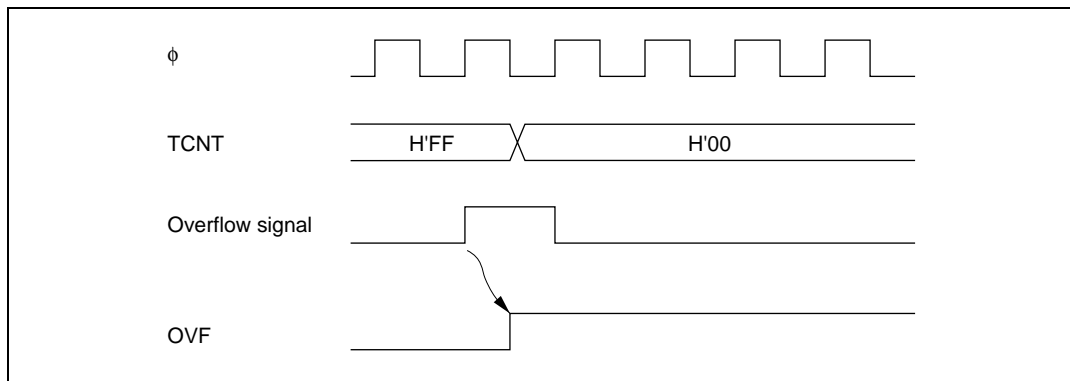
TCNT is cleared at the rising edge of an external reset input, depending on the settings of the CCLR1 and CCLR0 bits in TCR. The clear pulse width must be at least 1.5 states. Figure 12.8 shows the timing of this operation.



**Figure 12.8 Timing of Clearance by External Reset**

### 12.5.6 Timing of Overflow Flag (OVF) Setting

The OVF in TCSR is set to 1 when TCNT overflows (changes from H'FF to H'00). Figure 12.9 shows the timing of this operation.



**Figure 12.9 Timing of OVF Setting**

If bits CKS2 to CKS0 in either TCR\_0 or TCR\_1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of the 8-bit channel 0 could be counted by the timer of channel 1 (compare match count mode). In this case, the timer operates as below.

### 12.6.1 16-Bit Counter Mode

When bits CKS2 to CKS0 in TCR\_0 are set to B'100, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

[1] Setting of compare match flags

- The CMF flag in TCSR\_0 is set to 1 when a 16-bit compare match event occurs.
- The CMF flag in TCSR\_1 is set to 1 when a lower 8-bit compare match event occurs.

[2] Counter clear specification

- If the CCLR1 and CCLR0 bits in TCR\_0 have been set for counter clear at compare match, the 16-bit counters (TCNT\_0 and TCNT\_1 together) are cleared when a 16-bit compare match event occurs. The 16-bit counters (TCNT0 and TCNT1 together) are cleared even if counter clear by the TMRI0 pin has also been set.
- The settings of the CCLR1 and CCLR0 bits in TCR\_1 are ignored. The lower 8 bits cannot be cleared independently.

[3] Pin output

- Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR\_0 is in accordance with the 16-bit compare match conditions.
- Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR\_1 is in accordance with the lower 8-bit compare match conditions.

### 12.6.2 Compare Match Count Mode

When bits CKS2 to CKS0 in TCR\_1 are B'100, TCNT\_1 counts compare match A's for channel 0.

Channels 0 and 1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clear are in accordance with the settings for each channel.

## 12.7.1 Interrupt Sources and DTC Activation

There are three 8-bit timer interrupt sources: CMIA, CMIB, and OVI. Their relative priorities are shown in table 12.3. Each interrupt source is set as enabled or disabled by the corresponding interrupt enable bit in TCR or TCSR, and independent interrupt requests are sent for each to the interrupt controller. It is also possible to activate the DTC by means of CMIA and CMIB interrupts.

**Table 12.3 8-Bit Timer Interrupt Sources**

Name	Interrupt Source	Interrupt Flag	DTC Activation	Priority
CMIA0	TCORA_0 compare match	CMFA	Possible	High
CMIB0	TCORB_0 compare match	CMFB	Possible	↑ Low
OVI0	TCNT_0 overflow	OVF	Not possible	Low
CMIA1	TCORA_1 compare match	CMFA	Possible	High
CMIB1	TCORB_1 compare match	CMFB	Possible	↑ Low
OVI1	TCNT_1 overflow	OVF	Not possible	Low

## 12.7.2 A/D Converter Activation

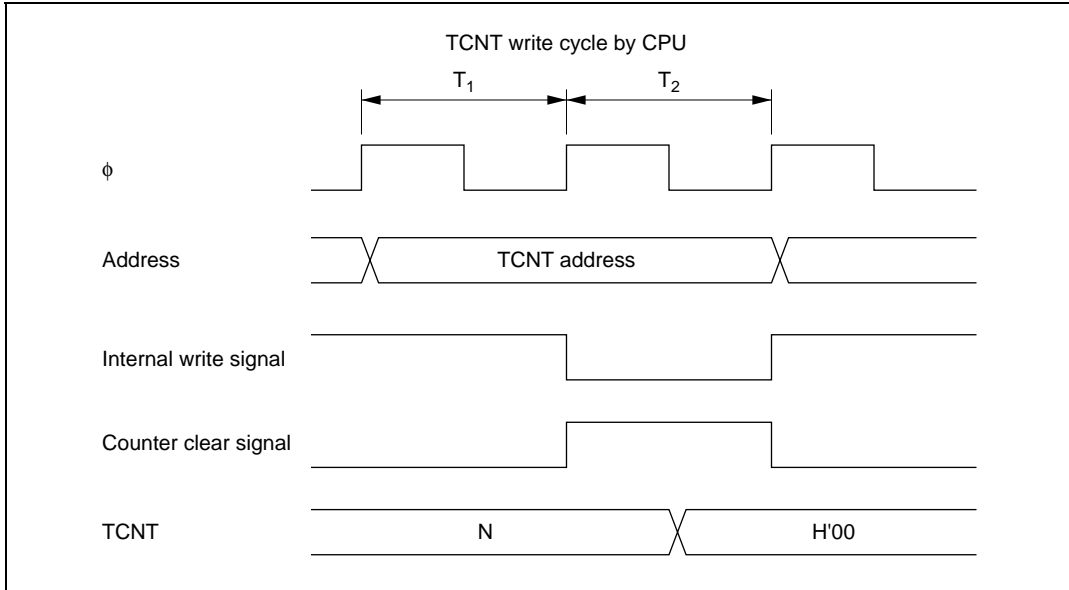
The A/D converter can be activated only by TMR\_0 compare match A.

If the ADTE bit in TCSR0 is set to 1 when the CMFA flag is set to 1 by the occurrence of TMR\_0 compare match A, a request to start A/D conversion is sent to the A/D converter. If the 8-bit timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

### 12.8.1 Contention between TCNT Write and Clear

If a timer counter clock pulse is generated during the  $T_2$  state of a TCNT write cycle, the clear takes priority, so that the counter is cleared and the write is not performed.

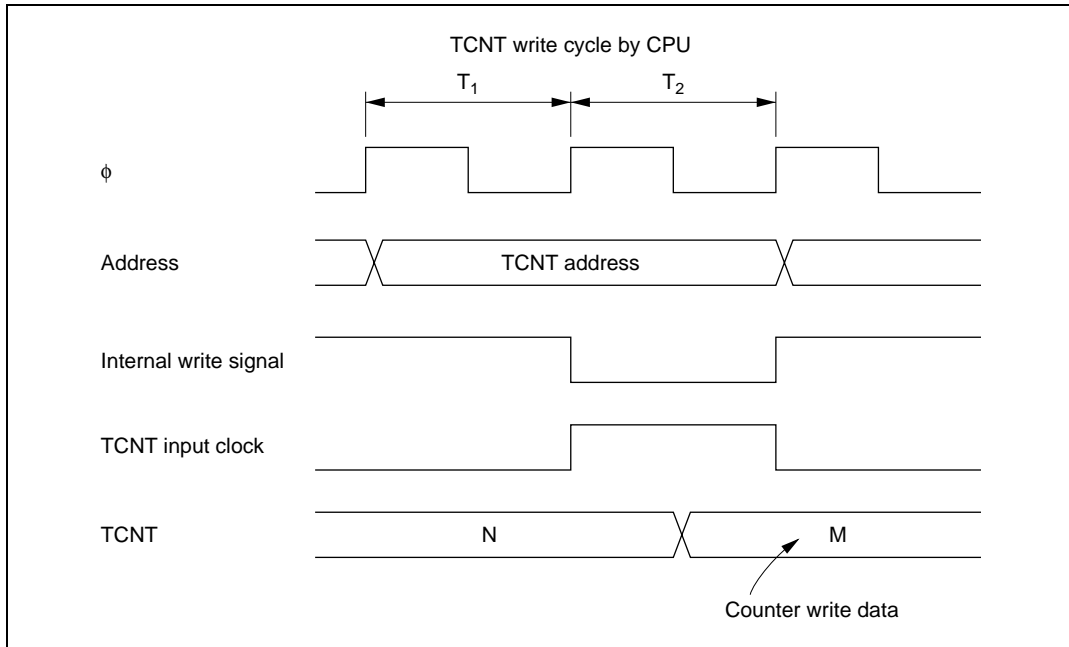
Figure 12.10 shows this operation.



**Figure 12.10 Contention between TCNT Write and Clear**

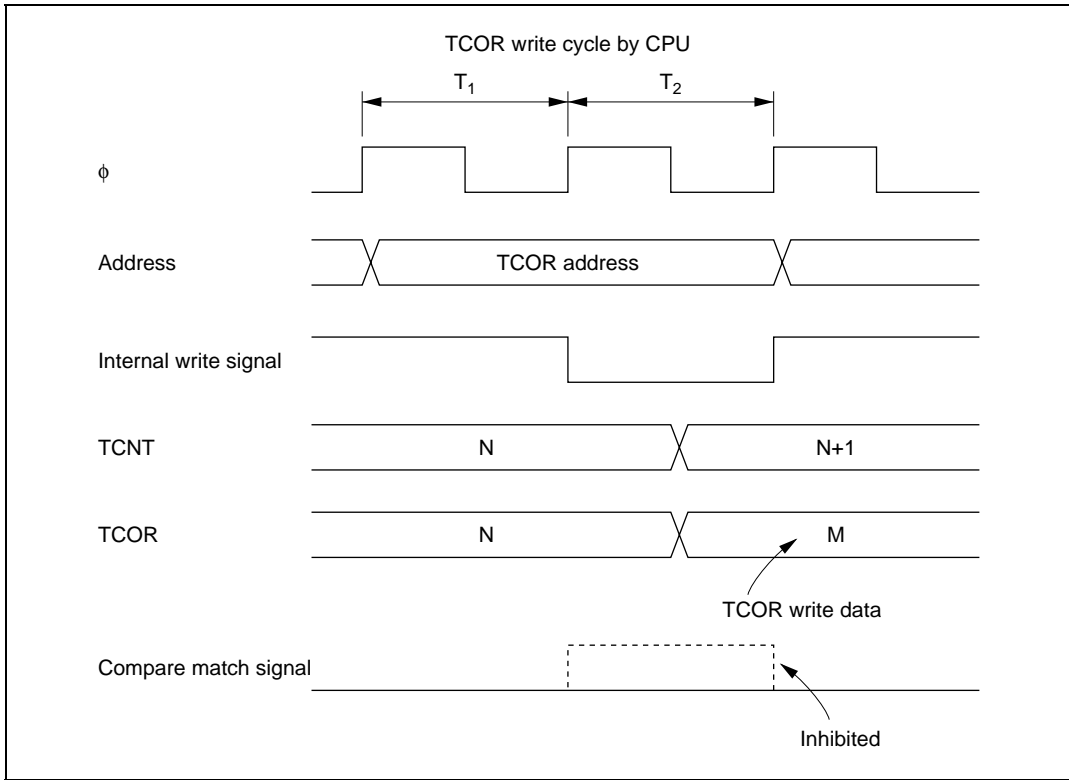
If a timer counter clock pulse is generated during the  $T_2$  state of a TCNT write cycle, the write takes priority and the counter is not incremented.

Figure 12.11 shows this operation.



**Figure 12.11 Contention between TCNT Write and Increment**

During the  $T_2$  state of a TCOR write cycle, the TCOR write has priority and the compare match signal is inhibited even if a compare match event occurs as shown in figure 12.12.



**Figure 12.12 Contention between TCOR Write and Compare Match**



If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output statuses set for compare match A and compare match B, as shown in table 12.4.

**Table 12.4 Timer Output Priorities**

<b>Output Setting</b>	<b>Priority</b>
Toggle output	High
1 output	▲ ↑
0 output	
No change	Low

### 12.8.5 Switching of Internal Clocks and TCNT Operation

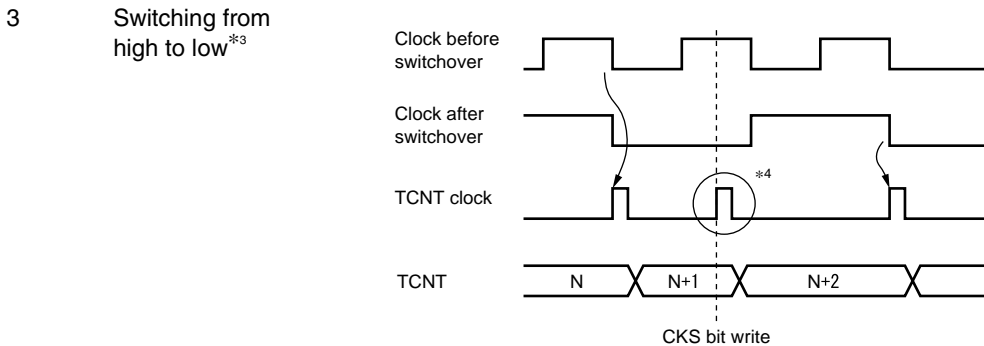
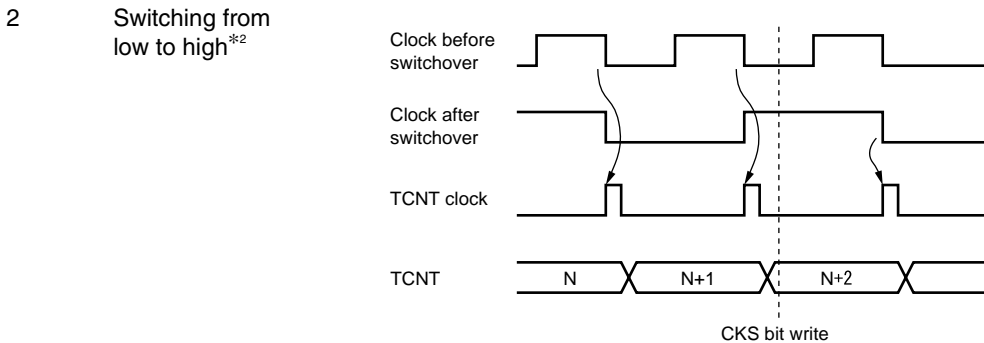
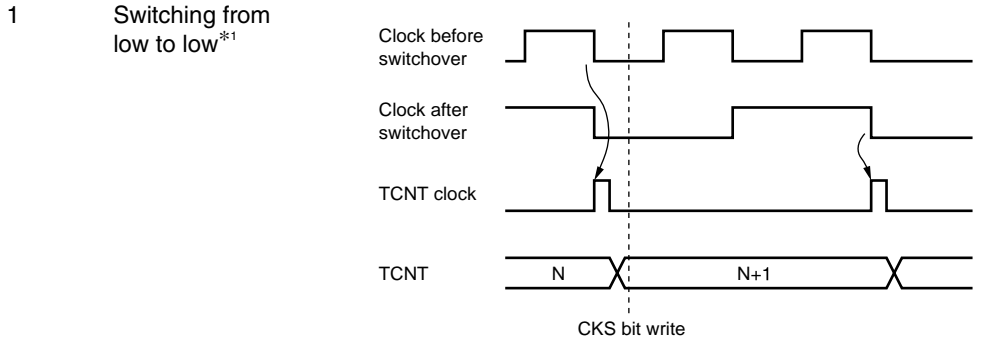
TCNT may increment erroneously when the internal clock is switched over. Table 12.5 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the falling edge of the internal clock pulse is detected. If clock switching causes a change from high to low level, as shown in case 3 in table 12.5, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge. This increments TCNT.

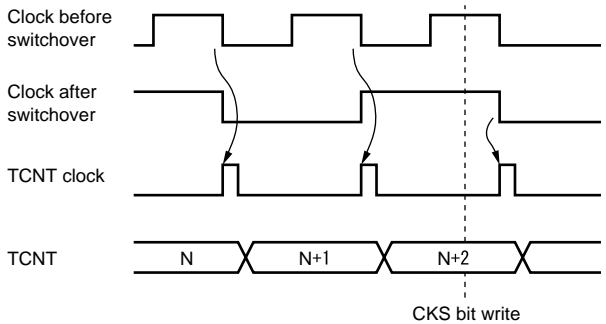
The erroneous incrementation can also happen when switching between internal and external clocks.

# Timing of Switchover by Means of CKS1 and CKS0 Bits

## No. TCNT Clock Operation



## 4 Switching from high to high



- Notes:
1. Includes switching from low to stop, and from stop to low.
  2. Includes switching from stop to high.
  3. Includes switching from high to stop.
  4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

### 12.8.6 Mode Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, input clocks for TCNT\_0 and TCNT\_1 are not generated, and the counter stops. Do not specify 16-bit counter and compare match count modes simultaneously.

### 12.8.7 Interrupts in Module Stop Mode

If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC and DMAC activation source. Interrupts should therefore be disabled before entering module stop mode.



The watchdog timer (WDT) is an 8-bit timer that outputs an overflow signal ( $\overline{\text{WDTOVF}}$ ) if a system crash prevents the CPU from writing to the timer counter, thus allowing it to overflow.

At the same time, the WDT can also generate an internal reset signal.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

The block diagram of the WDT is shown in figure 13.1.

## 13.1 Features

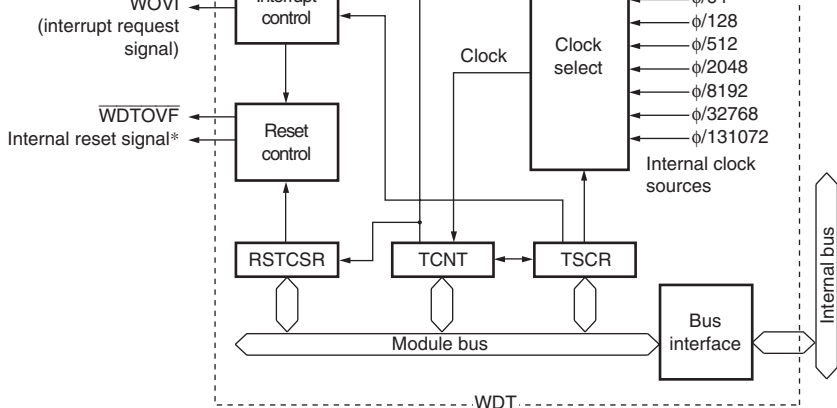
- Selectable from eight counter input clocks
- Switchable between watchdog timer mode and interval timer mode

In watchdog timer mode

- If the counter overflows, the WDT outputs  $\overline{\text{WDTOVF}}$ . It is possible to select whether or not the entire chip is reset at the same time.

In interval timer mode

- If the counter overflows, the WDT generates an interval timer interrupt (WOVI).



**Figure 13.1 Block Diagram of WDT**

## 13.2 Input/Output Pin

Table 13.1 describes the WDT output pin.

**Table 13.1 Pin Configuration**

Name	Symbol	I/O	Function
Watchdog timer overflow	WDTOVF	Output	Outputs counter overflow signal in watchdog timer mode

The WDT has the following three registers. To prevent accidental overwriting, TCSR, TCNT, and RSTCSR have to be written to in a method different from normal registers. For details, refer to section 13.6.1, Notes on Register Access.

- Timer counter (TCNT)
- Timer control/status register (TCSR)
- Reset control/status register (RSTCSR)

### 13.3.1 Timer Counter (TCNT)

TCNT is an 8-bit readable/writable up-counter. TCNT is initialized to H'00 when the TME bit in TCSR is cleared to 0.

### 13.3.2 Timer Control/Status Register (TCSR)

TCSR selects the clock source to be input to TCNT, and the timer mode.

Bit	Bit Name	Initial Value	R/W	Description
7	OVF	0	R/(W)*	<p>Overflow Flag</p> <p>Indicates that TCNT has overflowed in interval timer mode. Only a write of 0 is permitted, to clear the flag.</p> <p>[Setting condition]</p> <p>When TCNT overflows in interval timer mode (changes from H'FF to H'00)</p> <p>When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.</p> <p>[Clearing condition]</p> <p>Cleared by reading TCSR when OVF = 1, then writing 0 to OVF</p>

Selects whether the WDT is used as a watchdog timer or interval timer.

0: Interval timer mode

When TCNT overflows, an interval timer interrupt (WOVI) is requested.

1: Watchdog timer mode

When TCNT overflows, the  $\overline{\text{WDTOVF}}$  signal is output.

5	TME	0	R/W	Timer Enable When this bit is set to 1, TCNT starts counting. When this bit is cleared, TCNT stops counting and is initialized to H'00.
4, 3	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.
2	CKS2	0	R/W	Clock Select 2 to 0
1	CKS1	0	R/W	Selects the clock source to be input to TCNT. The overflow frequency for $\phi = 20$ MHz is enclosed in parentheses.
0	CKS0	0	R/W	000: Clock $\phi/2$ (frequency: 25.6 $\mu\text{s}$ ) 001: Clock $\phi/64$ (frequency: 819.2 $\mu\text{s}$ ) 010: Clock $\phi/128$ (frequency: 1.6 ms) 011: Clock $\phi/512$ (frequency: 6.6 ms) 100: Clock $\phi/2048$ (frequency: 26.2 ms) 101: Clock $\phi/8192$ (frequency: 104.9 ms) 110: Clock $\phi/32768$ (frequency: 419.4 ms) 111: Clock $\phi/131072$ (frequency: 1.68 s)

Note: \* Only a write of 0 is permitted, to clear the flag.



RSTCSR controls the generation of the internal reset signal when TCNT overflows, and selects the type of internal reset signal. RSTCSR is initialized to H'1F by a reset signal from the RES pin, but not by the WDT internal reset signal caused by overflows.

Bit	Bit Name	Initial Value	R/W	Description
7	WOVF	0	R/(W)*	<p>Watchdog Timer Overflow Flag</p> <p>This bit is set when TCNT overflows in watchdog timer mode. This bit cannot be set in interval timer mode, and only 0 can be written.</p> <p>[Setting condition]</p> <p>Set when TCNT overflows (changed from H'FF to H'00) in watchdog timer mode</p> <p>[Clearing condition]</p> <p>Cleared by reading RSTCSR when WOVF = 1, and then writing 0 to WOVF</p>
6	RSTE	0	R/W	<p>Reset Enable</p> <p>Specifies whether or not a reset signal is generated in the chip if TCNT overflows during watchdog timer operation.</p> <p>0: Reset signal is not generated even if TCNT overflows (Though this LSI is not reset, TCNT and TCSR in WDT are reset)</p> <p>1: Reset signal is generated if TCNT overflows</p>
5	—	0	R/W	<p>Reserved</p> <p>These bits can be read from or written to, but the operation is not affected.</p>
4 to 0	—	All 1	—	<p>Reserved</p> <p>These bits are always read as 1 and cannot be modified.</p>

Note: \* Only a write of 0 is permitted, to clear the flag.

### 13.4.1 Watchdog Timer Mode

To use the WDT as a watchdog timer mode, set the  $\overline{WT/IT}$  and TME bits in TCSR to 1.

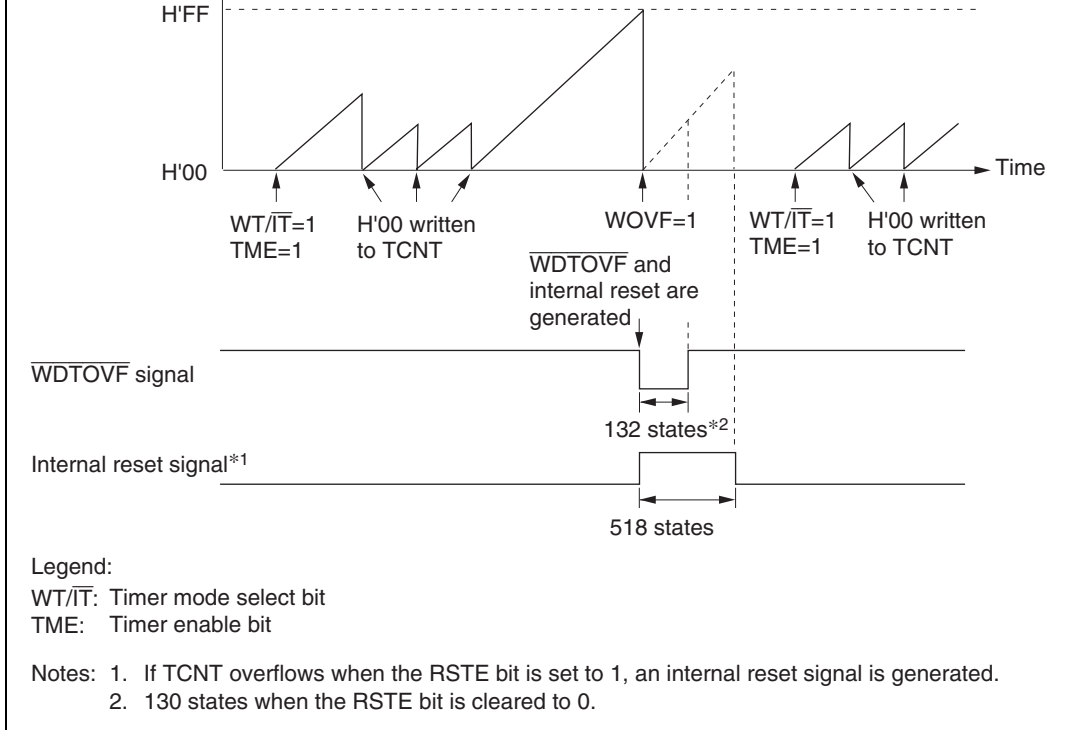
If TCNT overflows without being rewritten because of a system crash or other error, the  $\overline{WDTOVF}$  signal is output.

This ensures that TCNT does not overflow while the system is operating normally. Software must prevent TCNT overflows by rewriting the TCNT value (normally by writing H'00) before overflow occurs. This  $\overline{WDTOVF}$  signal can be used to reset the chip internally in watchdog timer mode.

If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, a signal that resets this LSI internally is generated at the same time as the  $\overline{WDTOVF}$  signal. If a reset caused by a signal input to the  $\overline{RES}$  pin occurs at the same time as a reset caused by a WDT overflow, the  $\overline{RES}$  pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.

The  $\overline{WDTOVF}$  signal is output for 132 states when RSTE = 1, and for 130 states when RSTE = 0. The internal reset signal is output for 518 states.

When TCNT overflows in watchdog timer mode, the WOVF bit in RSTCSR is set to 1. If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, an internal reset signal is generated to the entire chip.



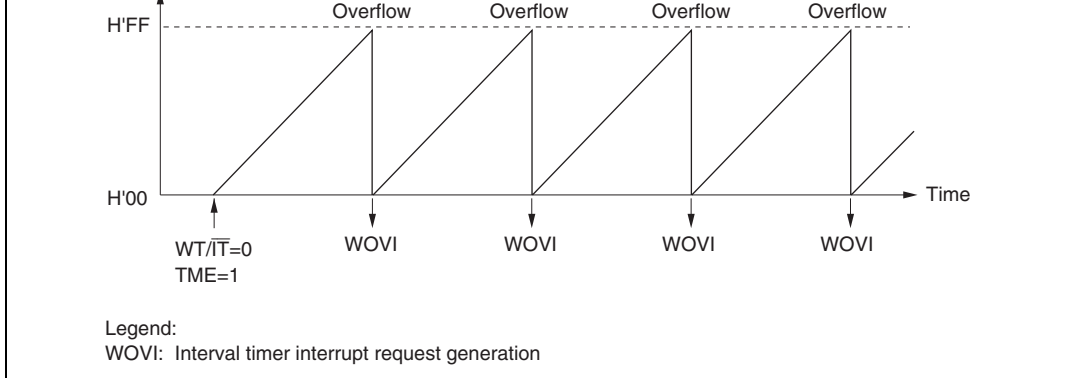
**Figure 13.2 Operation in Watchdog Timer Mode**

### 13.4.2 Interval Timer Mode

To use the WDT as an interval timer, set the  $WT/\overline{IT}$  bit to 0 and TME bit in TCSR to 1.

When the WDT is used as an interval timer, an interval timer interrupt (WOVI) is generated each time the TCNT overflows. Therefore, an interrupt can be generated at intervals.

When the TCNT overflows in interval timer mode, an interval timer interrupt (WOVI) is requested at the same time the OVF bit in the TCSR is set to 1.



**Figure 13.3 Operation in Interval Timer Mode**

## 13.5 Interrupts

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

**Table 13.2 WDT Interrupt Source**

Name	Interrupt Source	Interrupt Flag	DTC Activation
WOVI	TCNT overflow	OVF	Impossible

## 13.6 Usage Notes

### 13.6.1 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

#### Writing to TCNT, TCSR, and RSTCSR

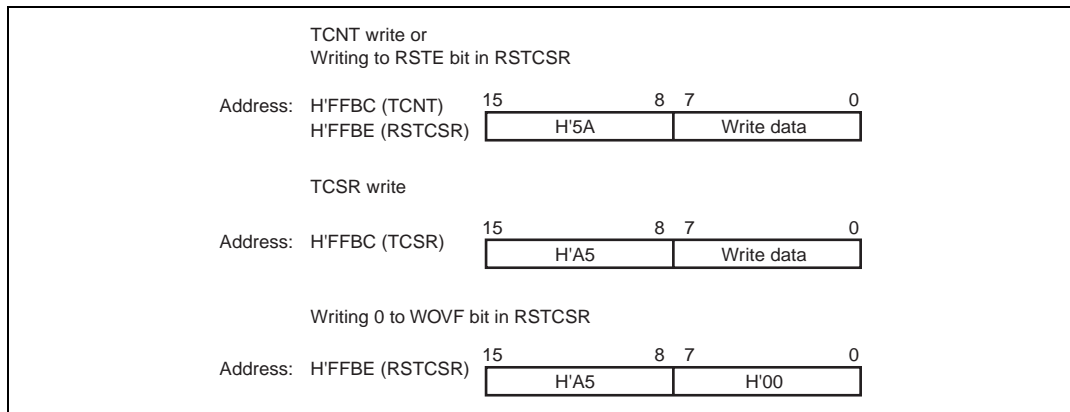
TCNT and TCSR must be written to by a word transfer instruction. They cannot be written to by a byte transfer instruction.

data to TCNT or TCSR according to the satisfied condition.

To write to RSTCSR, execute a word transfer instruction for address H'FFBE. A byte transfer instruction cannot perform writing to RSTCSR.

The method of writing 0 to the WOVF bit differs from that of writing to the RSTE bit. To write 0 to the WOVF bit, satisfy the lower condition shown in figure 13.4.

If satisfied, the transfer instruction clears the WOVF bit to 0, but has no effect on the RSTE bit. To write to the RSTE bit, satisfy the above condition shown in figure 13.4. If satisfied, the transfer instruction writes the value in bit 6 of the lower byte into the RSTE bit, but has no effect on the WOVF bit.

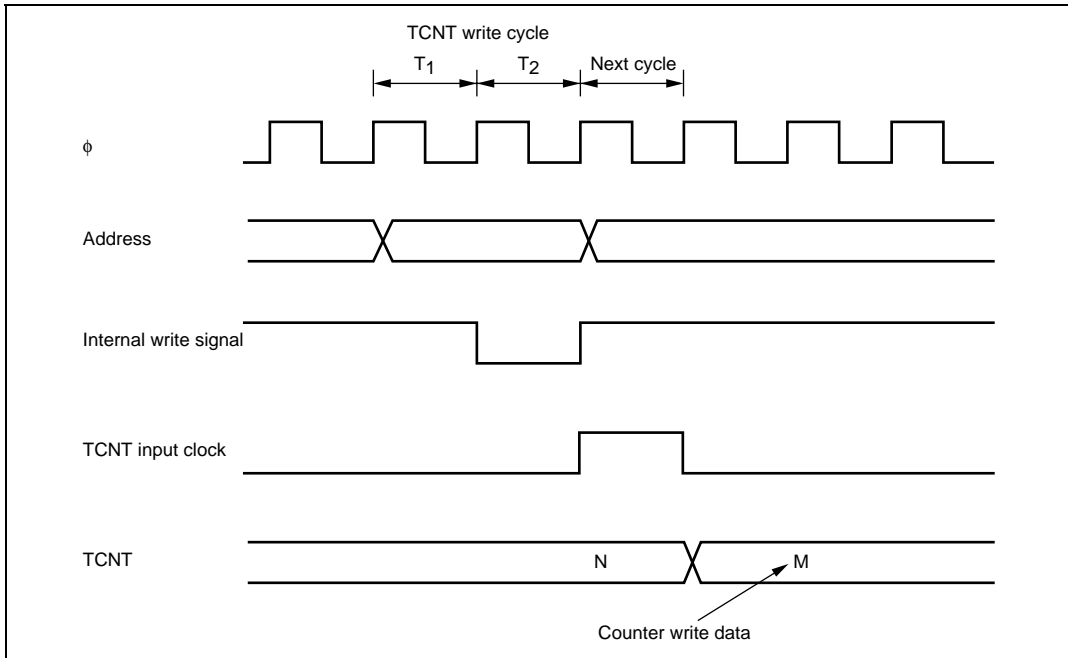


**Figure 13.4 Writing to TCNT, TCSR, and RSTCSR**

### Reading TCNT, TCSR, and RSTCSR

These registers are read in the same way as other registers. The read addresses are H'FFBC for TCSR, H'FFBD for TCNT, and H'FFBF for RSTCSR.

If a timer counter clock pulse is generated during the next cycle after the  $T_2$  state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 13.5 shows this operation.



**Figure 13.5 Contention between TCNT Write and Increment**

### 13.6.3 Changing Value of CKS2 to CKS0

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits CKS2 to CKS0.

### 13.6.4 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

This LSI is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer mode operation, but TCNT and TCSR of the WDT are reset.

TCNT, TCSR, and RSTCR cannot be written to while the  $\overline{\text{WDTOVF}}$  signal is low. Also note that a read of the WOVF flag is not recognized during this period. To clear the WOVF flag, therefore, read TCSR after the  $\overline{\text{WDTOVF}}$  signal goes high, then write 0 to the WOVF flag.

### 13.6.6 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the  $\overline{\text{WDTOVF}}$  output signal is input to the  $\overline{\text{RES}}$  pin, the chip will not be initialized correctly. Make sure that the  $\overline{\text{WDTOVF}}$  signal is not input logically to the  $\overline{\text{RES}}$  pin.

To reset the entire system by means of the  $\overline{\text{WDTOVF}}$  signal, use the circuit shown in figure 13.6.

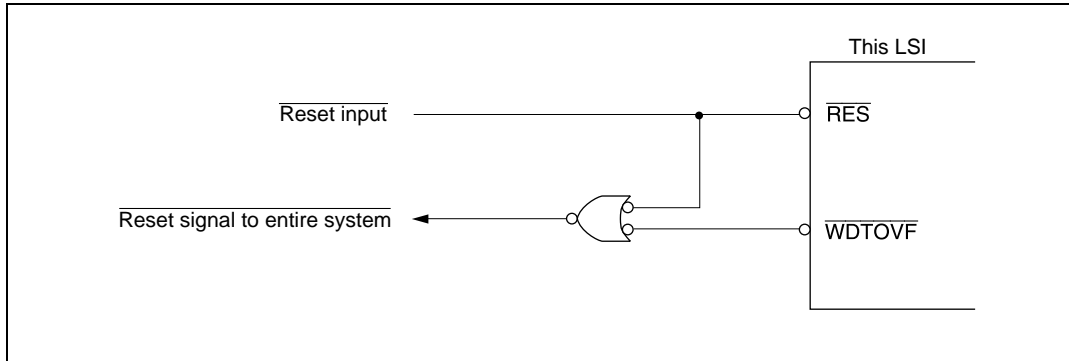


Figure 13.6 Circuit for System Reset by  $\overline{\text{WDTOVF}}$  Signal (Example)





This LSI has five independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A function is also provided for serial communication between processors (multiprocessor communication function) in asynchronous mode. The SCI also supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as an asynchronous serial communication interface extension function. One of the five SCI channels (SCI\_0) can generate an IrDA communication waveform conforming to IrDA specification version 1.0.

Figure 14.1 shows a block diagram of the SCI.

## 14.1 Features

- Choice of asynchronous or clocked synchronous serial communication mode
- Full-duplex communication capability  
The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously. Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data.
- On-chip baud rate generator allows any bit rate to be selected  
External clock can be selected as a transfer clock source (except for in Smart Card interface mode).
- Choice of LSB-first or MSB-first transfer (except in the case of asynchronous mode 7-bit data)
- Four interrupt sources  
Four interrupt sources — transmit-end, transmit-data-empty, receive-data-full, and receive error — that can issue requests. The transmit-data-empty interrupt and receive data full interrupts can activate the data transfer controller (DTC) or DMA controller (DMAC).
- Module stop mode can be set

### Asynchronous mode

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity: Even, odd, or none
- Receive error detection: Parity, overrun, and framing errors
- Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error

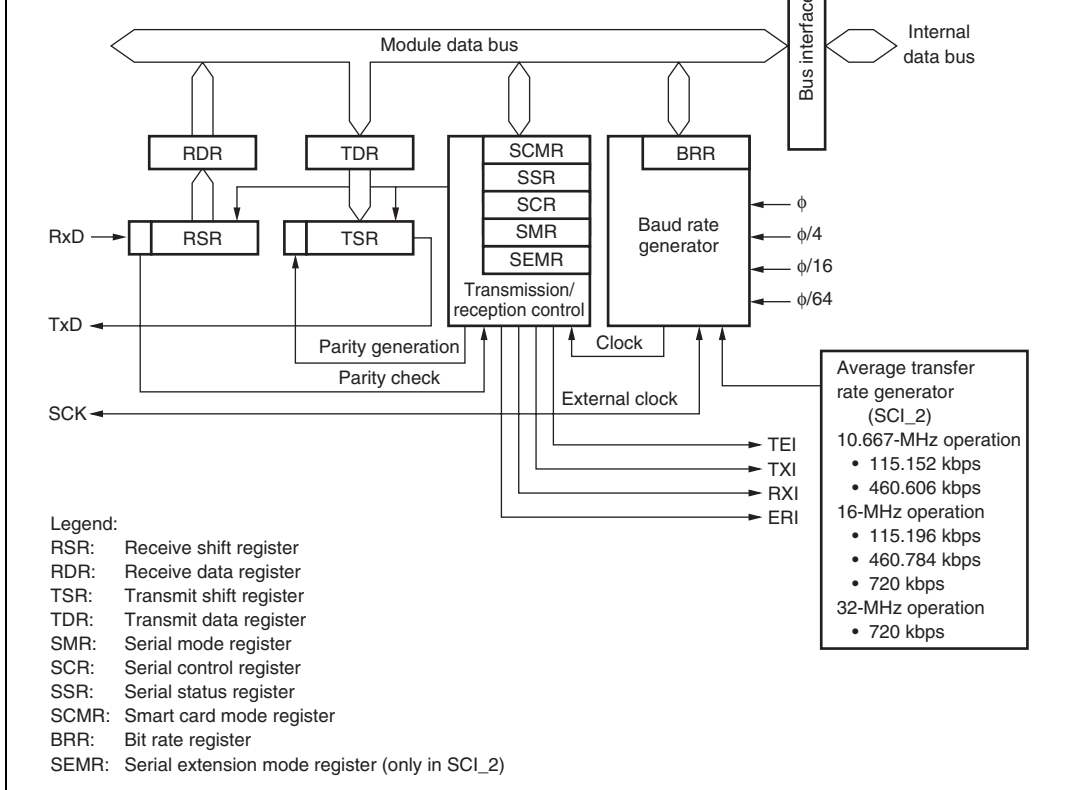
115.196, 460.784 or 720 kbps at 16-MHz operation  
720 kbps at 32-MHz operation

#### Clocked synchronous mode

- Data length: 8 bits
- Receive error detection: Overrun errors detected

#### Smart Card Interface

- Automatic transmission of error signal (parity error) in receive mode
- Error signal detection and automatic data retransmission in transmit mode
- Direct convention and inverse convention both supported



**Figure 14.1 Block Diagram of SCI**

Table 14.1 shows the serial pins for each SCI channel.

**Table 14.1 Pin Configuration**

<b>Channel</b>	<b>Pin Name*</b>	<b>I/O</b>	<b>Function</b>
0	SCK0	I/O	Channel 0 clock input/output
	RxD0/IrRxD	Input	Channel 0 receive data input (normal/IrDA)
	TxD0/IrTxD	Output	Channel 0 transmit data output (normal/IrDA)
1	SCK1	I/O	Channel 1 clock input/output
	RxD1	Input	Channel 1 receive data input
	TxD1	Output	Channel 1 transmit data output
2	SCK2	I/O	Channel 2 clock input/output
	RxD2	Input	Channel 2 receive data input
	TxD2	Output	Channel 2 transmit data output
3	SCK3	I/O	Channel 3 clock input/output
	RxD3	Input	Channel 3 receive data input
	TxD3	Output	Channel 3 transmit data output
4	SCK4	I/O	Channel 4 clock input/output
	RxD4	Input	Channel 4 receive data input
	TxD4	Output	Channel 4 transmit data output

Note: \* Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

The SCI has the following registers. The serial mode register (SMR), serial status register (SSR), and serial control register (SCR) are described separately for normal serial communication interface mode and Smart Card interface mode because their bit functions partially differ.

- Receive shift register\_0 (RSR\_0)
- Transmit shift register\_0 (TSR\_0)
- Receive data register\_0 (RDR\_0)
- Transmit data register\_0 (TDR\_0)
- Serial mode register\_0 (SMR\_0)
- Serial control register\_0 (SCR\_0)
- Serial status register\_0 (SSR\_0)
- Smart card mode register\_0 (SCMR\_0)
- Bit rate register\_0 (BRR\_0)
- IrDA control register\_0 (IrCR\_0)
- Receive shift register\_1 (RSR\_1)
- Transmit shift register\_1 (TSR\_1)
- Receive data register\_1 (RDR\_1)
- Transmit data register\_1 (TDR\_1)
- Serial mode register\_1 (SMR\_1)
- Serial control register\_1 (SCR\_1)
- Serial status register\_1 (SSR\_1)
- Smart card mode register\_1 (SCMR\_1)
- Bit rate register\_1 (BRR\_1)
- Receive shift register\_2 (RSR\_2)
- Transmit shift register\_2 (TSR\_2)
- Receive data register\_2 (RDR\_2)
- Transmit data register\_2 (TDR\_2)
- Serial mode register\_2 (SMR\_2)
- Serial control register\_2 (SCR\_2)
- Serial status register\_2 (SSR\_2)
- Smart card mode register\_2 (SCMR\_2)
- Bit rate register\_2 (BRR\_2)
- Serial extension mode register\_2 (SEMR\_2)
- Receive shift register\_3 (RSR\_3)

- Transmit data register\_3 (TDR\_3)
- Serial mode register\_3 (SMR\_3)
- Serial control register\_3 (SCR\_3)
- Serial status register\_3 (SSR\_3)
- Smart card mode register\_3 (SCMR\_3)
- Bit rate register\_3 (BRR\_3)
- Receive shift register\_4 (RSR\_4)
- Transmit shift register\_4 (TSR\_4)
- Receive data register\_4 (RDR\_4)
- Transmit data register\_4 (TDR\_4)
- Serial mode register\_4 (SMR\_4)
- Serial control register\_4 (SCR\_4)
- Serial status register\_4 (SSR\_4)
- Smart card mode register\_4 (SCMR\_4)
- Bit rate register\_4 (BRR\_4)

#### 14.3.1 Receive Shift Register (RSR)

RSR is a shift register used to receive serial data that is input to the RxD pin and convert it into parallel data. When one byte of data has been received, it is transferred to RDR automatically. RSR cannot be directly accessed by the CPU.

#### 14.3.2 Receive Data Register (RDR)

RDR is an 8-bit register that stores receive data. When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR where it is stored. After this, RSR is receive-enabled. Since RSR and RDR function as a double buffer in this way, enables continuous receive operations to be performed. After confirming that the RDRF bit in SSR is set to 1, read RDR for only once. RDR cannot be written to by the CPU.

#### 14.3.3 Transmit Data Register (TDR)

TDR is an 8-bit register that stores transmit data. When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has

achieve reliable serial transmission, write transmit data to TDR for only once after confirming that the TDRE bit in SSR is set to 1.

#### 14.3.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin starting. TSR cannot be directly accessed by the CPU.

#### 14.3.5 Serial Mode Register (SMR)

SMR is used to set the SCI's serial transfer format and select the on-chip baud rate generator clock source.

Some bit functions of SMR differ in normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	C/ $\bar{A}$	0	R/W	Communication Mode 0: Asynchronous mode 1: Clocked synchronous mode
6	CHR	0	R/W	Character Length (enabled only in asynchronous mode) 0: Selects 8 bits as the data length. 1: Selects 7 bits as the data length. LSB-first is fixed and the MSB (bit 7) of TDR is not transmitted in transmission. In clocked synchronous mode, a fixed data length of 8 bits is used.
5	PE	0	R/W	Parity Enable (enabled only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. For a multiprocessor format, parity bit addition and checking are not performed regardless of the PE bit setting.

asynchronous mode)

0: Selects even parity.

1: Selects odd parity.

---

3	STOP	0	R/W	<p>Stop Bit Length (enabled only in asynchronous mode)</p> <p>Selects the stop bit length in transmission.</p> <p>0: 1 stop bit</p> <p>1: 2 stop bits</p> <p>In reception, only the first stop bit is checked regardless of the STOP bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit character.</p>
2	MP	0	R/W	<p>Multiprocessor Mode (enabled only in asynchronous mode)</p> <p>When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and <math>O/\bar{E}</math> bit settings are invalid in multiprocessor mode.</p>
1	CKS1	0	R/W	<p>Clock Select 1 and 0:</p>
0	CKS0	0	R/W	<p>These bits select the clock source for the on-chip baud rate generator.</p> <p>00: <math>\phi</math> clock (<math>n = 0</math>)</p> <p>01: <math>\phi/4</math> clock (<math>n = 1</math>)</p> <p>10: <math>\phi/16</math> clock (<math>n = 2</math>)</p> <p>11: <math>\phi/64</math> clock (<math>n = 3</math>)</p> <p>For the relation between the bit rate register setting and the baud rate, see section 14.3.9, Bit Rate Register (BRR). <math>n</math> is the decimal display of the value of <math>n</math> in BRR (see section 14.3.9, Bit Rate Register (BRR)).</p>

---



Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	<p>GSM Mode</p> <p>When this bit is set to 1, the SCI operates in GSM mode. In GSM mode, the timing of the TEND setting is advanced by 11.0 etu (Elementary Time Unit: the time for transfer of 1 bit), and clock output control mode addition is performed. For details, refer to section 14.7.8, Clock Output Control.</p>
6	BLK	0	R/W	<p>When this bit is set to 1, the SCI operates in block transfer mode. For details on block transfer mode, refer to section 14.7.3, Block Transfer Mode.</p>
5	PE	0	R/W	<p>Parity Enable (enabled only in asynchronous mode)</p> <p>When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. In Smart Card interface mode, this bit must be set to 1.</p>
4	O $\bar{E}$	0	R/W	<p>Parity Mode (enabled only when the PE bit is 1 in asynchronous mode)</p> <p>0: Selects even parity. 1: Selects odd parity.</p> <p>For details on setting this bit in Smart Card interface mode, refer to section 14.7.2, Data Format (Except for Block Transfer Mode).</p>
3	BCP1	0	R/W	Basic Clock Pulse 1 and 0
2	BCP0	0	R/W	<p>These bits select the number of basic clock periods in a 1-bit transfer interval on the Smart Card interface.</p> <p>00: 32 clock (S = 32) 01: 64 clock (S = 64) 10: 372 clock (S = 372) 11: 256 clock (S = 256)</p> <p>For details, refer to section 14.7.4, Receive Data Sampling Timing and Reception Margin. S stands for the value of S in BRR (see section 14.3.9, Bit Rate Register (BRR)).</p>

0	CKS0	0	R/W	These bits select the clock source for the on-chip baud rate generator. 00: $\phi$ clock (n = 0) 01: $\phi/4$ clock (n = 1) 10: $\phi/16$ clock (n = 2) 11: $\phi/64$ clock (n = 3) For the relation between the bit rate register setting and the baud rate, see section 14.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 14.3.9, Bit Rate Register (BRR)).
---	------	---	-----	--

---

SCR performs enabling or disabling of SCI transfer operations and interrupt requests, and selection of the transfer/receive clock source. For details on interrupt requests, refer to section 14.9, SCI Interrupts. Some bit functions of SCR differ in normal serial communication interface mode and Smart Card interface mode.

Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	<p>Transmit Interrupt Enable</p> <p>When this bit is set to 1, TXI interrupt request is enabled.</p> <p>TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.</p>
6	RIE	0	R/W	<p>Receive Interrupt Enable</p> <p>When this bit is set to 1, RXI and ERI interrupt requests are enabled.</p> <p>RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or by clearing the RIE bit to 0.</p>
5	TE	0	R/W	<p>Transmit Enable</p> <p>When this bit is set to 1, transmission is enabled. In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0. SMR setting must be performed to decide the transfer format before setting the TE bit to 1.</p> <p>The TDRE flag in SSR is fixed at 1 if transmission is disabled by clearing this bit to 0.</p>

When this bit is set to 1, reception is enabled.

Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode. SMR setting must be performed to decide the transfer format before setting the RE bit to 1.

Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.

---

3	MPIE	0	R/W	<p>Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode)</p> <p>When this bit is set to 1, receive data in which the multiprocessor bit is 0 is skipped, and setting of the RDRF, FER, and ORER status flags in SSR is prohibited. On receiving data in which the multiprocessor bit is 1, this bit is automatically cleared and normal reception is resumed. For details, refer to section 14.5, Multiprocessor Communication Function.</p> <p>When receive data including MPB = 0 in SSR is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and ORER flags in SSR, is not performed. When receive data including MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting is enabled.</p>
2	TEIE	0	R/W	<p>Transmit End Interrupt Enable</p> <p>When this bit is set to 1, TEI interrupt request is enabled. TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or by clearing the TEIE bit to 0.</p>

---

0      CKE0      0

R/W      Selects the clock source and SCK pin function.

Asynchronous mode

00: On-chip baud rate generator  
SCK pin functions as I/O port

01: On-chip baud rate generator  
(Outputs a clock of the same frequency as the  
bit rate from the SCK pin.)

1x: External clock  
(Inputs a clock with a frequency 16 times the  
bit rate from the SCK pin.)

Clocked synchronous mode

0x: Internal clock (SCK pin functions as clock  
output)

1x: External clock (SCK pin functions as clock  
input)

---

Legend: x: Don't care

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	<p>Transmit Interrupt Enable</p> <p>When this bit is set to 1, TXI interrupt request is enabled. TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.</p>
6	RIE	0	R/W	<p>Receive Interrupt Enable</p> <p>When this bit is set to 1, RXI and ERI interrupt requests are enabled.</p> <p>RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or by clearing the RIE bit to 0.</p>
5	TE	0	R/W	<p>Transmit Enable</p> <p>When this bit is set to 1, transmission is enabled. In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0. SMR setting must be performed to decide the transfer format before setting the TE bit to 1.</p> <p>The TDRE flag in SSR is fixed at 1 if transmission is disabled by clearing this bit to 0.</p>
4	RE	0	R/W	<p>Receive Enable</p> <p>When this bit is set to 1, reception is enabled. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode. SMR setting must be performed to decide the transfer format before setting the RE bit to 1.</p> <p>Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.</p>
3	MPIE	0	R/W	<p>Multiprocessor Interrupt Enable (enabled only when the MP bit in SMR is 1 in asynchronous mode)</p> <p>Write 0 to this bit in Smart Card interface mode.</p>

Write 0 to this bit in Smart Card interface mode.

---

1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0		Enables or disables clock output from the SCK pin. The clock output can be dynamically switched in GSM mode. For details, refer to section 14.7.8, Clock Output Control.

When the GM bit in SMR is 0:

- 00: Output disabled (SCK pin can be used as an I/O port pin)
- 01: Clock output
- 1x: Reserved

When the GM bit in SMR is 1:

- 00: Output fixed low
- 01: Clock output
- 10: Output fixed high
- 11: Clock output

---

Legend: x: Don't care

SSR is a register containing status flags of the SCI and multiprocessor bits for transfer. 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER; they can only be cleared. Some bit functions of SSR differ in normal serial communication interface mode and Smart Card interface mode.

#### Normal Serial Communication Interface Mode (When SMIF in SCMR is 0)

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)*	<p>Transmit Data Register Empty</p> <p>Indicates whether TDR contains transmit data.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When the TE bit in SCR is 0</li> <li>• When data is transferred from TDR to TSR, and data writing to TDR is enabled.</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written to TDRE after reading TDRE = 1</li> <li>• When the DMAC or DTC is activated by a TXI interrupt request and transfers data to TDR</li> </ul>
6	RDRF	0	R/(W)*	<p>Receive Data Register Full</p> <p>Indicates that the received data is stored in RDR.</p> <p>[Setting condition]</p> <p>When serial reception ends normally and receive data is transferred from RSR to RDR</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written to RDRF after reading RDRF = 1</li> <li>• When the DMAC or DTC is activated by an RXI interrupt and transferred data from RDR</li> </ul> <p>The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0. Exercise care because if reception of the next data is completed while the RDRF flag is set to 1, an overrun error occurs and receive data will be lost.</p>



Indicates that an overrun error occurred while receiving and the reception has ended abnormally.

[Setting condition]

When the next serial reception is completed while RDRF = 1

The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

[Clearing condition]

When 0 is written to ORER after reading ORER = 1

The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

---

4	FER	0	R/(W)*	<p>Framing Error</p> <p>Indicates that a framing error occurred while receiving in asynchronous mode and the reception has ended abnormally.</p> <p>[Setting condition]</p> <p>When the stop bit is 0</p> <p>In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the FER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.</p> <p>[Clearing condition]</p> <p>When 0 is written to FER after reading FER = 1</p> <p>The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.</p>
---	-----	---	--------	--

---

Indicates that a parity error occurred while receiving in asynchronous mode and the reception has ended abnormally.

[Setting condition]

When a parity error is detected during reception

If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

[Clearing condition]

When 0 is written to PER after reading PER = 1  
The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

2	TEND	1	R	<p>Transmit End</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When the TE bit in SCR is 0</li> <li>• When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written to TDRE after reading TDRE = 1</li> <li>• When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR</li> </ul>
1	MPB	0	R	<p>Multiprocessor Bit</p> <p>MPB stores the multiprocessor bit in the receive data. When the RE bit in SCR is cleared to 0 its previous state is retained.</p>
0	MPBT	0	R/W	<p>Multiprocessor Bit Transfer</p> <p>MPBT sets the multiprocessor bit to be added to the transmit data.</p>

Note: \* Only 0 can be written, to clear the flag. Alternately, use the bit clear instruction to clear the flag.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	1	R/(W)* <sup>1</sup>	<p>Transmit Data Register Empty</p> <p>Indicates whether TDR contains transmit data.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>• When the TE bit in SCR is 0</li> <li>• When data is transferred from TDR to TSR, and data writing to TDR is enabled.</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written to TDRE after reading TDRE = 1</li> <li>• When the DMAC or DTC is activated by a TXI interrupt request and transfers data to TDR</li> </ul>
6	RDRF	0	R/(W)* <sup>1</sup>	<p>Receive Data Register Full</p> <p>Indicates that the received data is stored in RDR.</p> <p>[Setting condition]</p> <p>When serial reception ends normally and receive data is transferred from RSR to RDR</p> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>• When 0 is written to RDRF after reading RDRF = 1</li> <li>• When the DMAC or DTC is activated by an RXI interrupt and transferred data from RDR</li> </ul> <p>The RDRF flag is not affected and retains its previous value when the RE bit in SCR is cleared to 0. Exercise care because if reception of the next data is completed while the RDRF flag is set to 1, an overrun error occurs and receive data will be lost.</p>

Indicates that an overrun error occurred while receiving and the reception has ended abnormally.

[Setting condition]

When the next serial reception is completed while RDRF = 1

The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

[Clearing condition]

When 0 is written to ORER after reading ORER = 1

The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

---

4	ERS	0	R/(W)* <sup>1</sup>	Error Signal Status [Setting condition] When the low level of the error signal is sampled [Clearing condition] When 0 is written to ERS after reading ERS = 1
---	-----	---	---------------------	---

---

Indicates that a parity error occurred while receiving in asynchronous mode and the reception has ended abnormally.

[Setting condition]

When a parity error is detected during reception

If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

[Clearing condition]

When 0 is written to PER after reading PER = 1

The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

---

This bit is set to 1 when no error signal has been sent back from the receiving end and the next transmit data is ready to be transferred to TDR.

[Setting conditions]

- When the TE bit in SCR is 0 and the ERS bit is also 0
- If the ERS bit is 0 and the TDRE bit is 1 after the specified interval after transmission of 1-byte data

Timing to set this bit differs according to the register settings.

GM = 0, BLK = 0: 2.5 etu\*<sup>2</sup> after transmission

GM = 0, BLK = 1: 1.5 etu\*<sup>2</sup> after transmission

GM = 1, BLK = 0: 1.0 etu\*<sup>2</sup> after transmission

GM = 1, BLK = 1: 1.0 etu\*<sup>2</sup> after transmission

[Clearing conditions]

- When 0 is written to TEND after reading TEND = 1
- When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR

1	MPB	0	R	Multiprocessor Bit This bit is not used in Smart Card interface mode.
0	MPBT	0	R/W	Multiprocessor Bit Transfer Write 0 to this bit in Smart Card interface mode.

- Notes: 1. Only 0 can be written, to clear the flag. Alternately, use the bit clear instruction to clear the flag.
2. etu: Elementary Time Unit: (time for transfer of 1 bit)

SCMR selects Smart Card interface mode and its format.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 1	—	Reserved These bits are always read as 1.
3	SDIR	0	R/W	Smart Card Data Transfer Direction Selects the serial/parallel conversion format. 0: LSB-first in transfer 1: MSB-first in transfer The bit setting is valid only when the transfer data format is 8 bits. For 7-bit data, LSB-first is fixed.
2	SINV	0	R/W	Smart Card Data Invert Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the $O/\bar{E}$ bit in SMR. 0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.
1	—	1	—	Reserved This bit is always read as 1.
0	SMIF	0	R/W	Smart Card Interface Mode Select This bit is set to 1 to make the SCI operate in Smart Card interface mode. 0: Normal asynchronous mode or clocked synchronous mode 1: Smart card interface mode

BRR is an 8-bit register that adjusts the bit rate. As the SCI performs baud rate generator control independently for each channel, different bit rates can be set for each channel. Table 14.2 shows the relationships between the N setting in BRR and bit rate B for normal asynchronous mode, clocked synchronous mode, and Smart Card interface mode. The initial value of BRR is H'FF, and it can be read or written to by the CPU at all times.

**Table 14.2 Relationships between N Setting in BRR and Bit Rate B**

Mode	Bit Rate	Error
Asynchronous Mode	$B = \frac{\phi \times 10^6}{64 \times 2^{2n-1} \times (N + 1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \right\} \times 100$
Clocked Synchronous Mode	$B = \frac{\phi \times 10^6}{8 \times 2^{2n-1} \times (N + 1)}$	
Smart Card Interface Mode	$B = \frac{\phi \times 10^6}{S \times 2^{2n-1} \times (N + 1)}$	$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{B \times S \times 2^{2n-1} \times (N + 1)} - 1 \right\} \times 100$

Notes: B: Bit rate (bit/s)

N: BRR setting for baud rate generator ( $0 \leq N \leq 255$ )

$\phi$ : Operating frequency (MHz)

n and S: Determined by the SMR settings shown in the following tables.

SMR Setting			SMR Setting		
CKS1	CKS0	n	BCP1	BCP0	S
0	0	0	0	0	32
0	1	1	0	1	64
1	0	2	1	0	372
1	1	3	1	1	256

Table 14.3 shows sample N settings in BRR in normal asynchronous mode. Table 14.4 shows the maximum bit rate for each frequency in normal asynchronous mode. Table 14.6 shows sample N settings in BRR in clocked synchronous mode. Table 14.8 shows sample N settings in BRR in Smart Card interface mode. In Smart Card interface mode, S (the number of basic clock periods in a 1-bit transfer interval) can be selected. For details, refer to section 14.7.4, Receive Data Sampling Timing and Reception Margin. Tables 14.5 and 14.7 show the maximum bit rates with external clock input.



Operating Frequency  $\phi$  (MHz)

Bit Rate (bit/s)	8			9.8304			10			12		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	-0.26	2	177	-0.25	2	212	0.03
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	-1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	-1.73	0	19	-2.34
31250	0	7	0.00	0	9	-1.70	0	9	0.00	0	11	0.00
38400	—	—	—	0	7	0.00	0	7	-1.73	0	9	-2.34

Operating Frequency  $\phi$  (MHz)

Bit Rate (bit/s)	12.288			14			14.7456			16		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	-0.17	3	64	0.69	3	70	0.03
150	2	159	0.00	2	181	0.16	2	191	0.00	2	207	0.16
300	2	79	0.00	2	90	0.16	2	95	0.00	2	103	0.16
600	1	159	0.00	1	181	0.16	1	191	0.00	1	207	0.16
1200	1	79	0.00	1	90	0.16	1	95	0.00	1	103	0.16
2400	0	159	0.00	0	181	0.16	0	191	0.00	0	207	0.16
4800	0	79	0.00	0	90	0.16	0	95	0.00	0	103	0.16
9600	0	39	0.00	0	45	-0.93	0	47	0.00	0	51	0.16
19200	0	19	0.00	0	22	-0.93	0	23	0.00	0	25	0.16
31250	0	11	2.40	0	13	0.00	0	14	-1.70	0	15	0.00
38400	0	9	0.00	—	—	—	0	11	0.00	0	12	0.16

Bit Rate (bit/s)				Error (%)						Error (%)		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	75	0.48	3	79	-0.12	3	86	0.31	3	88	-0.25
150	2	223	0.00	2	233	0.16	2	255	0.00	3	64	0.16
300	2	111	0.00	2	116	0.16	2	127	0.00	2	129	0.16
600	1	223	0.00	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	111	0.00	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	223	0.00	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	111	0.00	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	55	0.00	0	58	-0.69	0	63	0.00	0	64	0.16
19200	0	27	0.00	0	28	1.02	0	31	0.00	0	32	-1.36
31250	0	16	1.20	0	17	0.00	0	19	-1.70	0	19	0.00
38400	0	13	0.00	0	14	-2.34	0	15	0.00	0	15	1.73

### Operating Frequency $\phi$ (MHz)

Bit Rate (bit/s)	25			30			33			34*		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	110	-0.02	3	132	0.13	3	145	0.33	3	150	-0.05
150	3	80	0.47	3	97	-0.35	3	106	0.39	3	110	-0.29
300	2	162	-0.15	2	194	0.16	2	214	-0.07	2	220	0.16
600	2	80	0.47	2	97	-0.35	2	106	0.39	2	110	-0.29
1200	1	162	-0.15	1	194	0.16	1	214	-0.07	1	220	0.16
2400	1	80	0.47	1	97	-0.35	1	106	0.39	1	110	-0.29
4800	0	162	-0.15	0	194	0.16	0	214	-0.07	0	220	0.16
9600	0	80	0.47	0	97	-0.35	0	106	0.39	0	110	-0.29
19200	0	40	-0.76	0	48	-0.35	0	53	-0.54	0	54	0.62
31250	0	24	0.00	0	29	0	0	32	0	0	33	0.00
38400	0	19	1.73	0	23	1.73	0	26	-0.54	0	27	-1.18

Note: \* Supported only by the H8S/2368 0.18  $\mu$ m F-ZTAT Group.

$\psi$ (MHz)	Maximum Bit Rate (bits)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	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
19.6608	614400	0	0
20	625000	0	0
25	781250	0	0
30	937500	0	0
33	1031250	0	0
34*	1062500	0	0

Note: \* Supported only by the H8S/2368 0.18  $\mu$ m F-ZTAT Group.

$\psi$ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits)
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
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
19.6608	4.9152	307200
20	5.0000	312500
25	6.2500	390625
30	7.5000	468750
33	8.2500	515625
34*	8.5000	531250

Note: \* Supported only by the H8S/2368 0.18  $\mu$ m F-ZTAT Group.

Bit Rate (bit/s)	8		10		16		20		25		30		33		34 <sup>*1</sup>	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																
250	3	124	—	—	3	249										
500	2	249	—	—	3	124	—	—			3	233				
1 k	2	124	—	—	2	249	—	—	3	97	3	116	3	128	3	128
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	2	212
5 k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	105
10 k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1	212
25 k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	84
50 k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	0	169
100 k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	84
250 k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	33
500 k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	16
1 M	0	1			0	3	0	4	—	—	—	—	—	—		
2.5 M			0	0*			0	1	—	—	0	2	—	—		
5 M							0	0*	—	—	—	—	—	—		

## Legend:

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

\*: Continuous transfer is not possible.

Note: 1. Supported only by the H8S/2368 0.18  $\mu$ m F-ZTAT Group.

$\phi$ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits)
8	1.3333	1333333.3
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
25	4.1667	4166666.7
30	5.0000	5000000.0
33	5.5000	5500000.0
34*	5.6667	5666666.7

Note: \* Supported only by the H8S/2368 0.18  $\mu$ m F-ZTAT Group.

		Operating Frequency $\phi$ (MHz)											
		10.00			10.7136			13.00			14.2848		
Bit Rate (bit/s)			Error (%)			Error (%)			Error (%)			Error (%)	
	n	N		n	N		n	N		n	N		
9600	0	1	30	0	1	25	0	1	8.99	0	1	0.00	

		Operating Frequency $\phi$ (MHz)											
		16.00			18.00			20.00			25.00		
Bit Rate (bit/s)			Error (%)			Error (%)			Error (%)			Error (%)	
	n	N		n	N		n	N		n	N		
9600	0	1	12.01	0	2	15.99	0	2	6.60	0	3	12.49	

		Operating Frequency $\phi$ (MHz)								
		30.00			33.00			34.00*		
Bit Rate (bit/s)			Error (%)			Error (%)			Error (%)	
	n	N		n	N		n	N		
9600	0	3	5.01	0	4	7.59	0	4	4.79	

Note: \* Supported only by the H8S/2368 0.18  $\mu$ m F-ZTAT Group.

$\phi$ (MHz)	Maximum Bit Rate (bit/s)	n	N
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0
20.00	26882	0	0
25.00	33602	0	0
30.00	40323	0	0
33.00	44355	0	0
34.00*	45699	0	0

Note: \* Supported only by the H8S/2368 0.18  $\mu$ m F-ZTAT Group.



IrCR selects the function of SCI\_0.

Bit	Bit Name	Initial Value	R/W	Description	
7	IrE	0	R/W	IrDA Enable Specifies normal SCI mode or IrDA mode for SCI_0 input/output. 0: Pins TxD0/IrTxD and RxD0/IrRxD function as TxD0 and RxD0 1: Pins TxD0/IrTxD and RxD0/IrRxD function as IrTxD and IrRxD	
6	IrCKS2	0	R/W	IrDA Clock Select 2 to 0	
5	IrCKS1	0	R/W	Specifies the high pulse width in IrTxD output pulse encoding when the IrDA function is enabled. 000: Pulse width = $B \times 3/16$ (3/16 of bit rate) 001: Pulse width = $\phi/2$ 010: Pulse width = $\phi/4$ 011: Pulse width = $\phi/8$ 100: Pulse width = $\phi/16$ 101: Pulse width = $\phi/32$ 110: Pulse width = $\phi/64$ 111: Pulse width = $\phi/128$	
4	IrCKS0	0	R/W		
3	—	All 0	—		Reserved
to					
0					These bits are always read as 0 and cannot be modified.

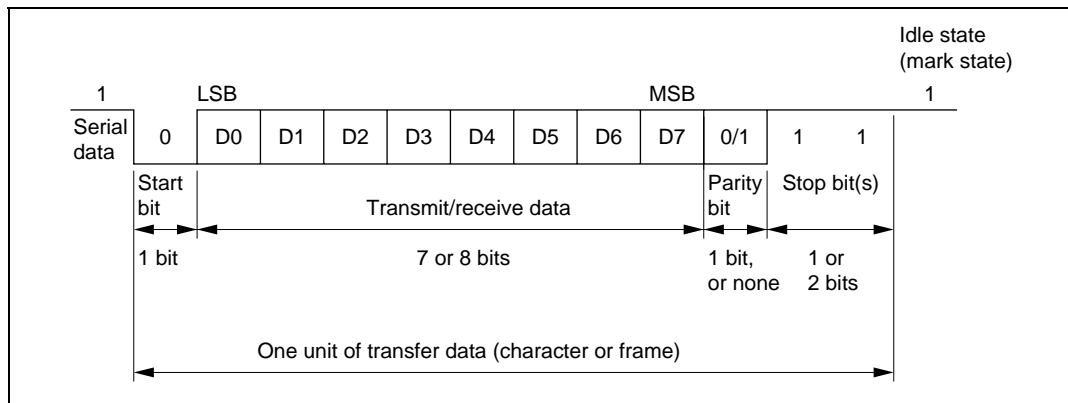
SEMR selects the clock source in asynchronous mode. The basic clock can be automatically set by selecting the average transfer rate.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved If these bits are read, an undefined value will be returned. They cannot be modified.
3	ABCS	0	R/W	Asynchronous basic clock selection (valid only in asynchronous mode) Selects the basic clock for 1-bit period in asynchronous mode. 0: Operates on a basic clock with a frequency of 16 times the transfer rate. 1: Operates on a basic clock with a frequency of 8 times the transfer rate.

1	ACS1	0	R/W	CKE1 = 1 in asynchronous mode)
0	ACS0	0	R/W	Selects the clock source for the average transfer rate.
				The basic clock can be automatically set by selecting the average transfer rate in spite of the value of ABCS.
				000: External clock input
				001: Selects 115.152 kbps which is the average transfer rate dedicated for $\phi = 10.667$ MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.)
				010: Selects 460.606 kbps which is the average transfer rate dedicated for $\phi = 10.667$ MHz. (Operates on a basic clock with a frequency of 8 times the transfer rate.)
				011: Selects 720 kbps which is the average transfer rate dedicated for $\phi = 32$ MHz. (Operates on a basic clock with a frequency of 16 times the transfer rate.)
				100: Reserved
				101: Selects 115.196 kbps which is the average transfer rate dedicated for $\phi = 16$ MHz (Operates on a basic clock with a frequency of 16 times the transfer rate.)
				110: Selects 460.784 kbps which is the average transfer rate dedicated for $\phi = 16$ MHz (Operates on a basic clock with a frequency of 16 times the transfer rate.)
				111: Selects 720 kbps which is the average transfer rate dedicated for $\phi = 16$ MHz (Operates on a basic clock with a frequency of 8 times the transfer rate.)
				Note that the average transfer rate does not correspond to the frequency other than 10.667, 16, or 32 MHz.

---

Figure 14.2 shows the general format for asynchronous serial communication. One frame consists of a start bit (low level), followed by transfer data, a parity bit, and finally stop bits (high level). In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. In asynchronous serial communication, the communication line is usually held in the mark state (high level). The SCI monitors the communication line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.



**Figure 14.2 Data Format in Asynchronous Communication  
(Example with 8-Bit Data, Parity, Two Stop Bits)**

#### 14.4.1 Data Transfer Format

Table 14.10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting. For details on the multiprocessor bit, refer to section 14.5, Multiprocessor Communication Function.

SMR Settings				Serial Transfer Format and Frame Length													
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12		
0	0	0	0	S	8-bit data								STOP				
0	0	0	1	S	8-bit data								STOP	STOP			
0	1	0	0	S	8-bit data								P	STOP			
0	1	0	1	S	8-bit data								P	STOP	STOP		
1	0	0	0	S	7-bit data							STOP					
1	0	0	1	S	7-bit data							STOP	STOP				
1	1	0	0	S	7-bit data							P	STOP				
1	1	0	1	S	7-bit data							P	STOP	STOP			
0	—	1	0	S	8-bit data								MPB	STOP			
0	—	1	1	S	8-bit data								MPB	STOP	STOP		
1	—	1	0	S	7-bit data							MPB	STOP				
1	—	1	1	S	7-bit data							MPB	STOP	STOP			

Legend:

S: Start bit

STOP: Stop bit

P: Parity bit

MPB: Multiprocessor bit

In asynchronous mode, the SCI operates on a clock with a frequency of 16 times the bit rate. In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched at the middle of each bit by sampling the data at the rising edge of the 8th pulse of the basic clock as shown in figure 14.3. Thus the reception margin in asynchronous mode is given by formula (1) below.

$$M = \left\{ \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right\} \times 100 [\%]$$

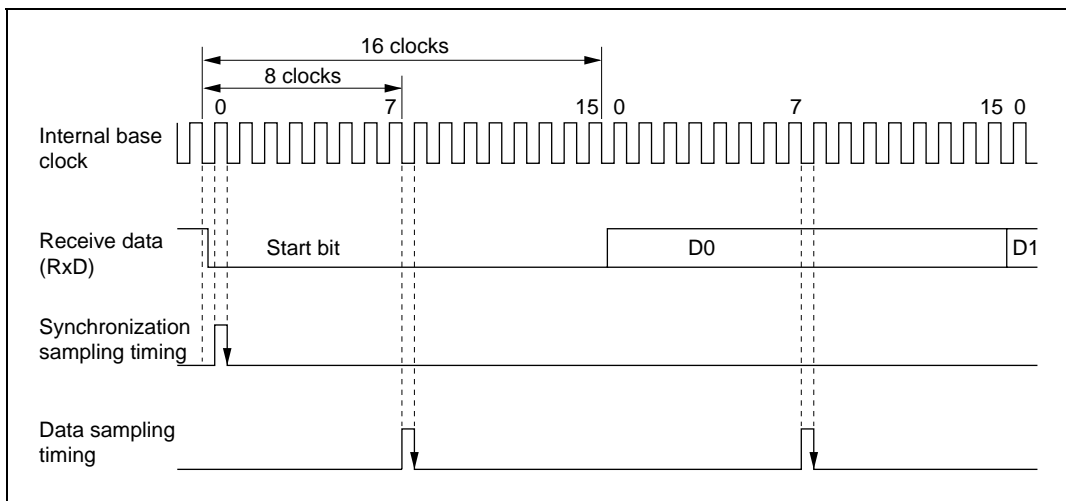
... Formula (1)

- Where
- M: Reception Margin
  - N: Ratio of bit rate to clock (N = 16)
  - D: Clock duty (D = 0.5 to 1.0)
  - L: Frame length (L = 9 to 12)
  - F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), a reception margin is given by formula below.

$$M = \{ 0.5 - 1/(2 \times 16) \} \times 100 [\%] = 46.875\%$$

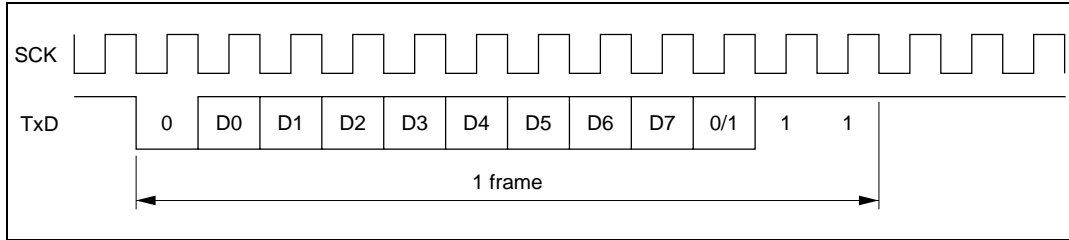
However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.



**Figure 14.3 Receive Data Sampling Timing in Asynchronous Mode**

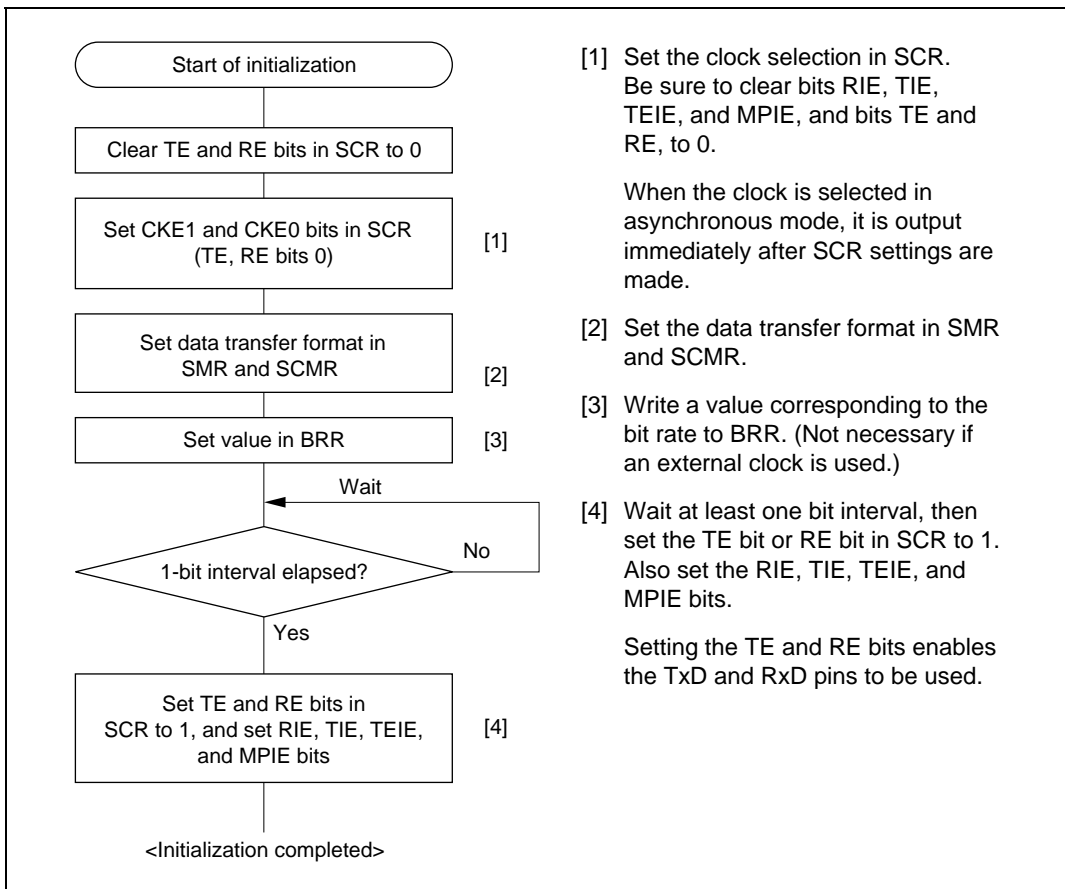
Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the  $C/\bar{A}$  bit in SMR and the CKE1 and CKE0 bits in SCR. When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 14.4.



**Figure 14.4 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)**

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as shown in figure 14.5. Do not write to SMR, SCMR, IrCR, or SEMR while the SCI is operating. This also applies to writing the same data as the current register contents. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not initialize the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR. When the external clock is used in asynchronous mode, the clock must be supplied even during initialization.

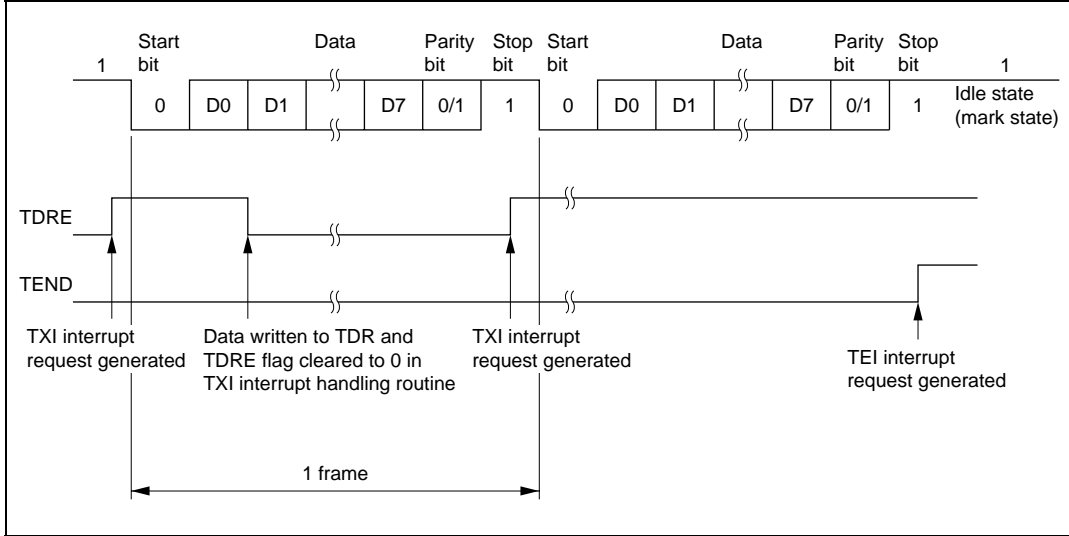


**Figure 14.5 Sample SCI Initialization Flowchart**

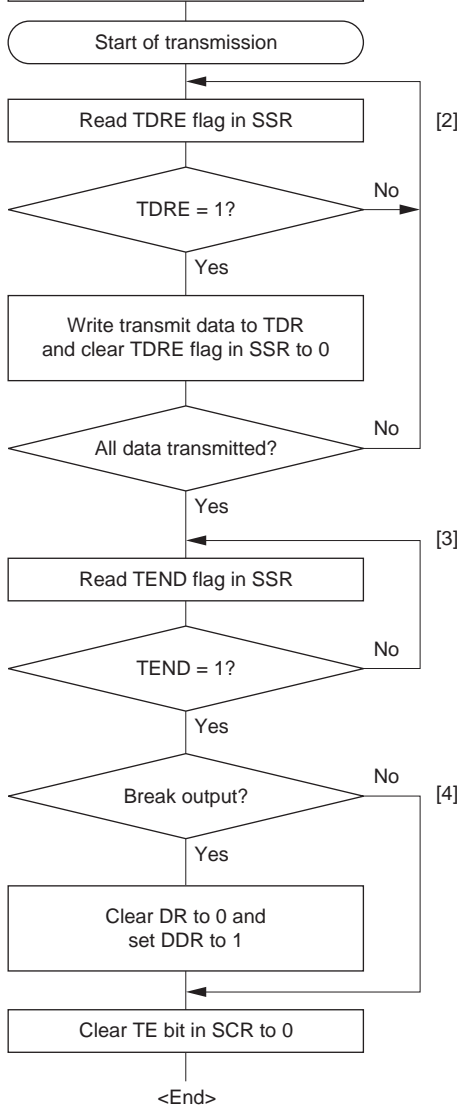


Figure 14.6 shows an example of the operation for transmission in asynchronous mode. In transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is cleared to 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit is set to 1 at this time, a transmit data empty interrupt request (TXI) is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. Data is sent from the TxD pin in the following order: start bit, transmit data, parity bit or multiprocessor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks the TDRE flag at the timing for sending the stop bit.
5. If the TDRE flag is 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.
6. If the TDRE flag is 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the “mark state” is entered in which 1 is output. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.



**Figure 14.6 Example of Operation in Transmission in Asynchronous Mode  
(Example with 8-Bit Data, Parity, One Stop Bit)**



The TXD pin is automatically designated as the transmit data output pin.  
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

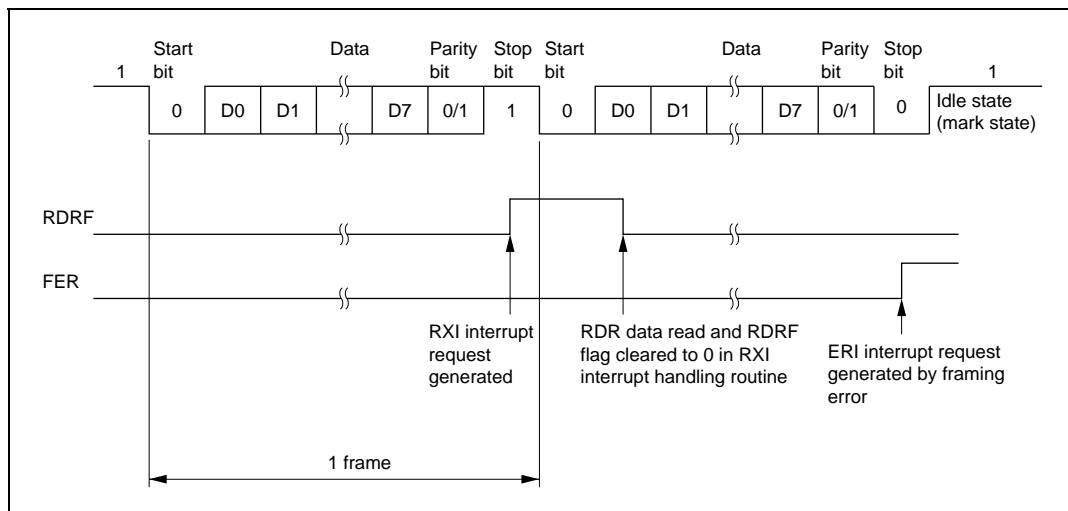
- [2] SCI status check and transmit data write:  
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.
- [3] Serial transmission continuation procedure:  
To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC\* or DTC is activated by a transmit-data-empty interrupt (TXI) request, and data is written to TDR.
- [4] Break output at the end of serial transmission:  
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Note: \* Not supported by the H8S/2366F.

**Figure 14.7 Sample Serial Transmission Flowchart**

Figure 14.8 shows an example of the operation for reception in asynchronous mode. In serial reception, the SCI operates as described below.

1. The SCI monitors the communication line, and if a start bit is detected, performs internal synchronization, receives receive data in RSR, and checks the parity bit and stop bit.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the ORE bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
4. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated.
5. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.



**Figure 14.8 Example of SCI Operation in Reception  
(Example with 8-Bit Data, Parity, One Stop Bit)**

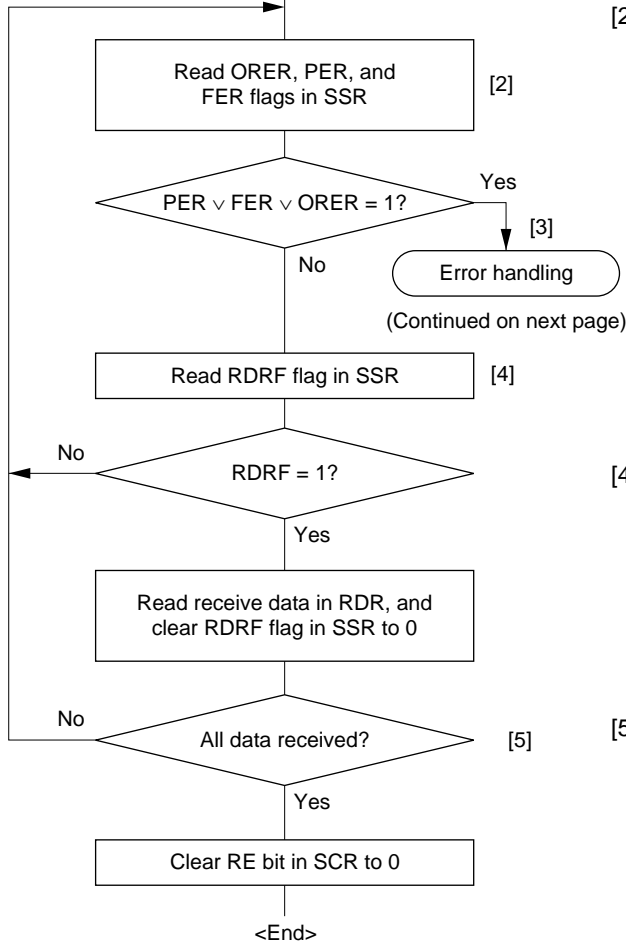
data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.9 shows a sample flowchart for serial data reception.

**Table 14.11 SSR Status Flags and Receive Data Handling**

SSR Status Flag				Receive Data	Receive Error Type
RDRF*	ORER	FER	PER		
1	1	0	0	Lost	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

Note: \* The RDRF flag retains its state before data reception.

The RxD pin is automatically designated as the receive data input pin.



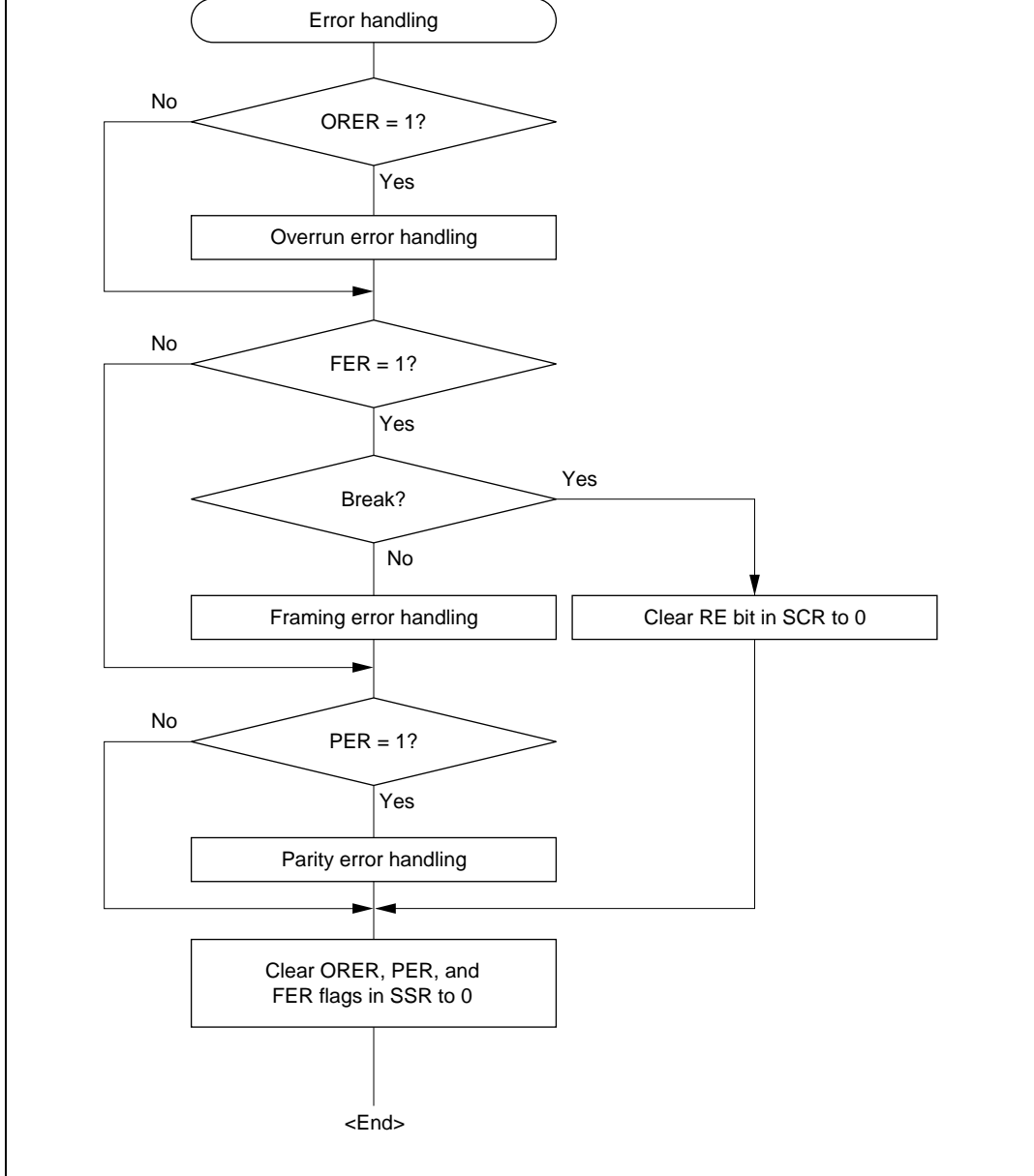
[2] [3] Receive error handling and break detection:  
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read :  
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:  
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when the DMAC\* or DTC is activated by an RXI interrupt and the RDR value is read.

Note: \* Not supported by the H8S/2366F.

Figure 14.9 Sample Serial Reception Data Flowchart (1)



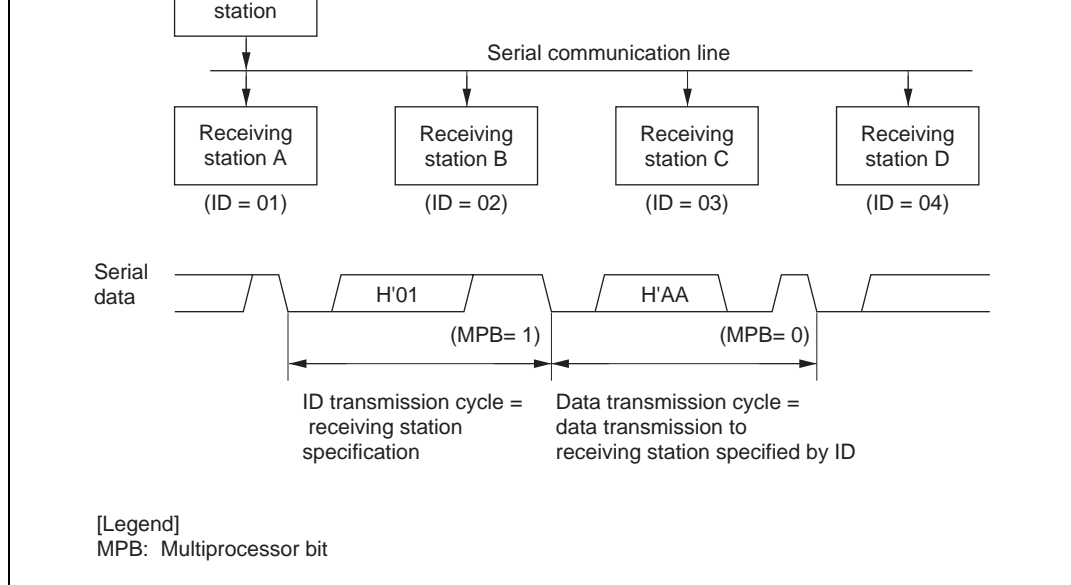
**Figure 14.9 Sample Serial Reception Data Flowchart (2)**

Use of the multiprocessor communication function enables data transfer to be performed among a number of processors sharing communication lines by means of asynchronous serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data. When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code. The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle to the specified receiving station. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle. If the multiprocessor bit is 1, the cycle is an ID transmission cycle, and if the multiprocessor bit is 0, the cycle is a data transmission cycle. Figure 14.10 shows an example of inter-processor communication using the multiprocessor format. The transmitting station first sends communication data with a 1 multiprocessor bit added to the ID code of the receiving station. It then sends transmit data as data with a 0 multiprocessor bit added. When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip data until data with a 1 multiprocessor bit is again received.

The SCI uses the MPIE bit in SCR to implement this function. When the MPIE bit is set to 1, transfer of receive data from RSR to RDR, error flag detection, and setting the SSR status flags, RDRF, FER, and ORER to 1 are inhibited until data with a 1 multiprocessor bit is received. On reception of receive character with a 1 multiprocessor bit, the MPBR bit in SSR is set to 1 and the MPIE bit is automatically cleared, thus normal reception is resumed. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt is generated.

When the multiprocessor format is selected, the parity bit setting is invalid. All other bit settings are the same as those in normal asynchronous mode. The clock used for multiprocessor communication is the same as that in normal asynchronous mode.

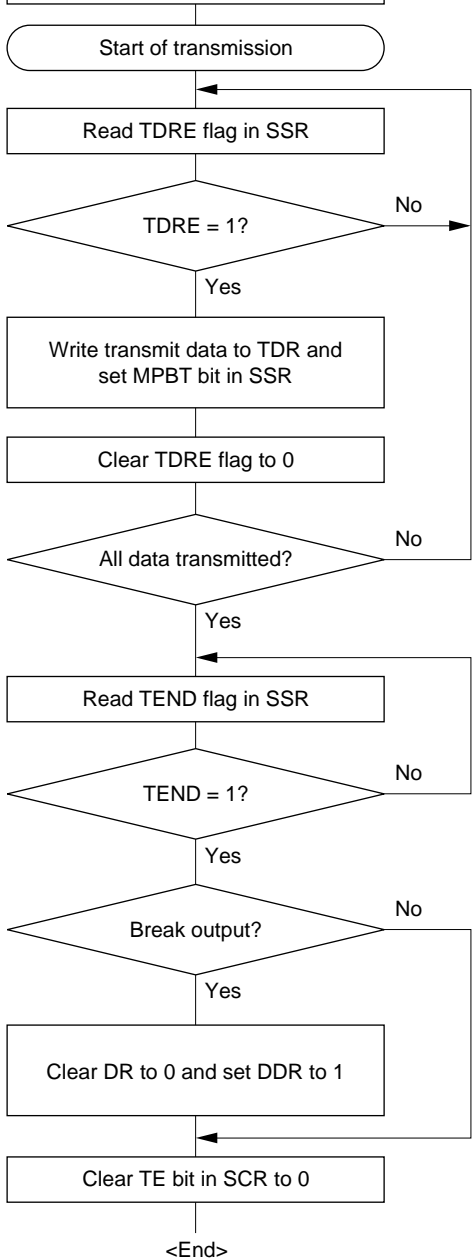




**Figure 14.10 Example of Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)**

### 14.5.1 Multiprocessor Serial Data Transmission

Figure 14.11 shows a sample flowchart for multiprocessor serial data transmission. For an ID transmission cycle, set the MPBT bit in SSR to 1 before transmission. For a data transmission cycle, clear the MPBT bit in SSR to 0 before transmission. All other SCI operations are the same as those in asynchronous mode.



The TXD pin is automatically designated as the transmit data output pin.

After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2]

[2] SCI status check and transmit data write:  
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3]

[3] Serial transmission continuation procedure:  
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC\* or DTC is activated by a transmit-data-empty interrupt (TXI) request, and data is written to TDR.

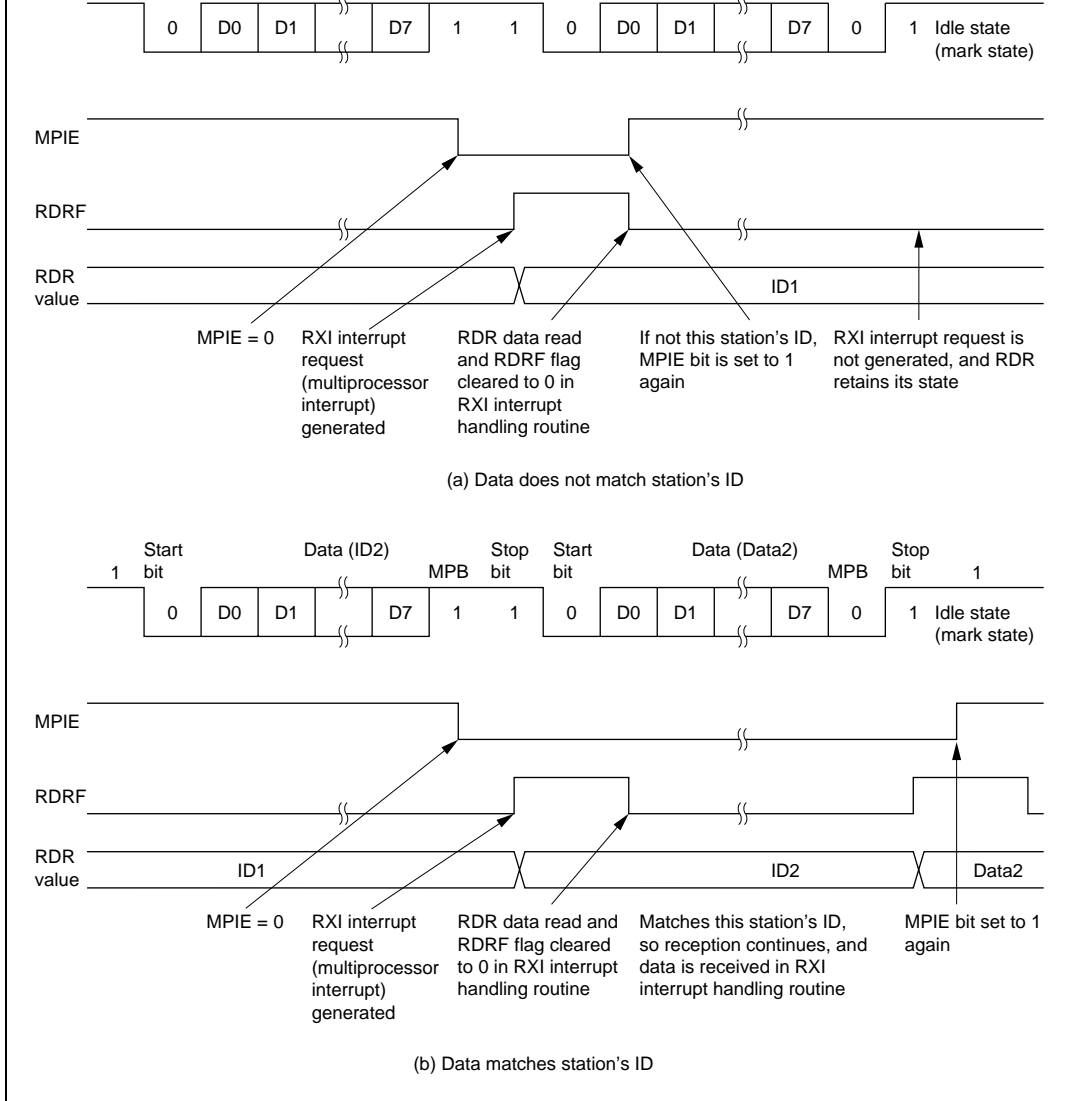
[4]

[4] Break output at the end of serial transmission:  
To output a break in serial transmission, set the port DDR to 1, clear DR to 0, then clear the TE bit in SCR to 0.

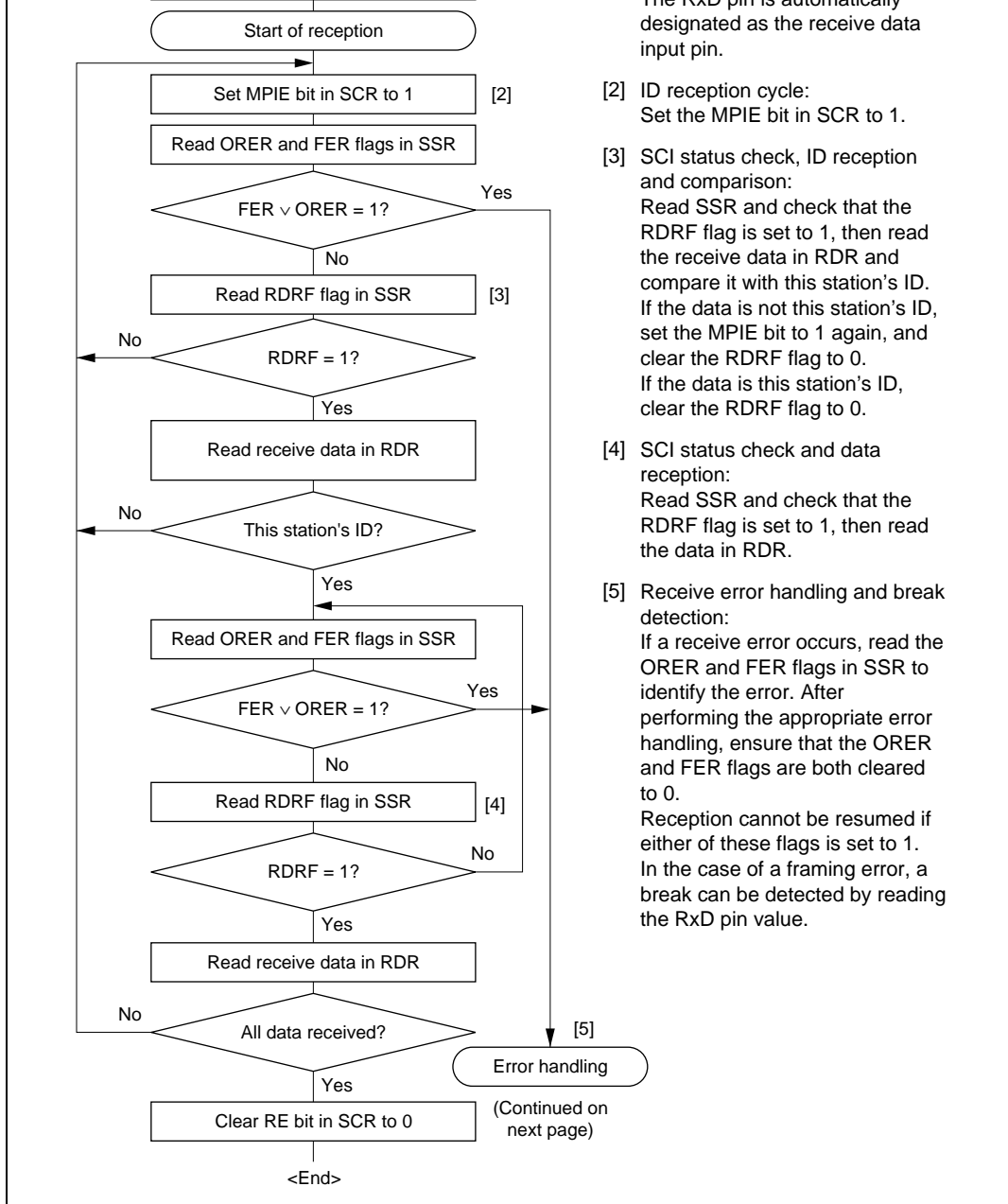
Note: \* Not supported by the H8S/2366F.

**Figure 14.11 Sample Multiprocessor Serial Transmission Flowchart**

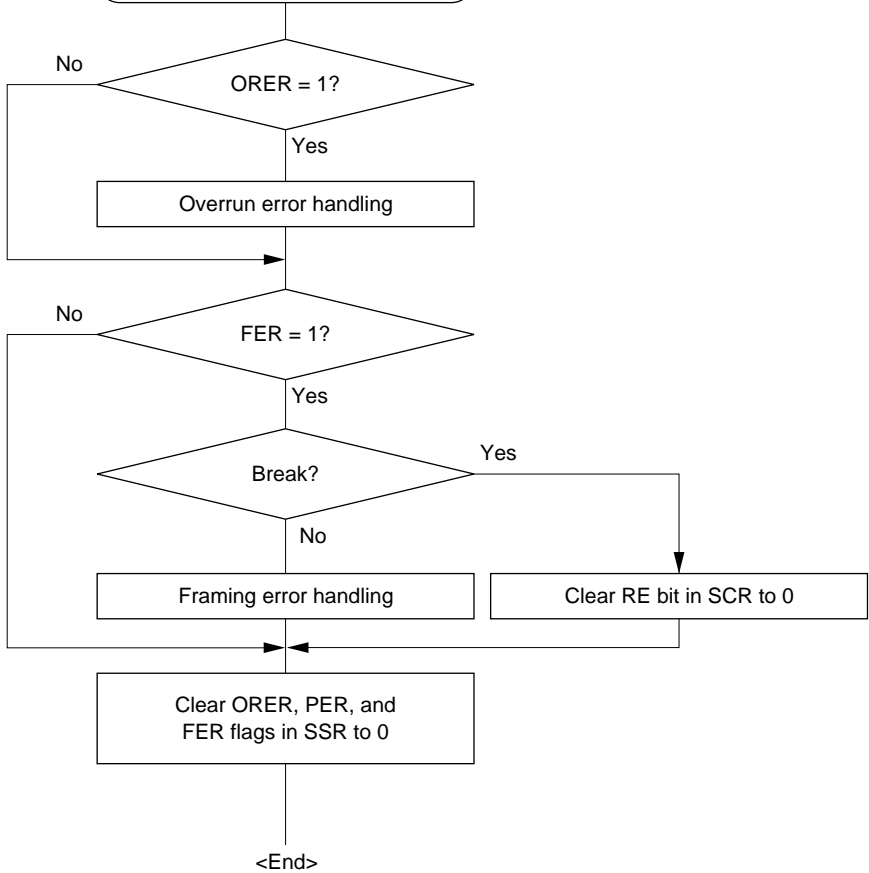
Figure 14.13 shows a sample flowchart for multiprocessor serial data reception. If the MPIE bit in SCR is set to 1, data is skipped until data with a 1 multiprocessor bit is received. On receiving data with a 1 multiprocessor bit, the receive data is transferred to RDR. An RXI interrupt request is generated at this time. All other SCI operations are the same as in asynchronous mode. Figure 14.12 shows an example of SCI operation for multiprocessor format reception.



**Figure 14.12 Example of SCI Operation in Reception  
(Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)**

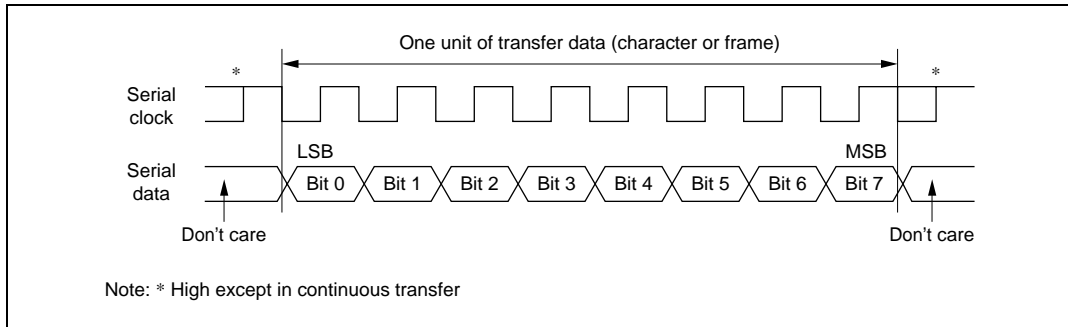


**Figure 14.13 Sample Multiprocessor Serial Reception Flowchart (1)**



**Figure 14.13 Sample Multiprocessor Serial Reception Flowchart (2)**

Figure 14.14 shows the general format for clocked synchronous communication. In clocked synchronous mode, data is transmitted or received in synchronization with clock pulses. One character of communication data consists of 8-bit data. In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. In clocked synchronous mode, the SCI receives data in synchronization with the rising edge of the serial clock. After 8-bit data is output, the transmission line holds the MSB state. In clocked synchronous mode, no parity or multiprocessor bit is added. Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

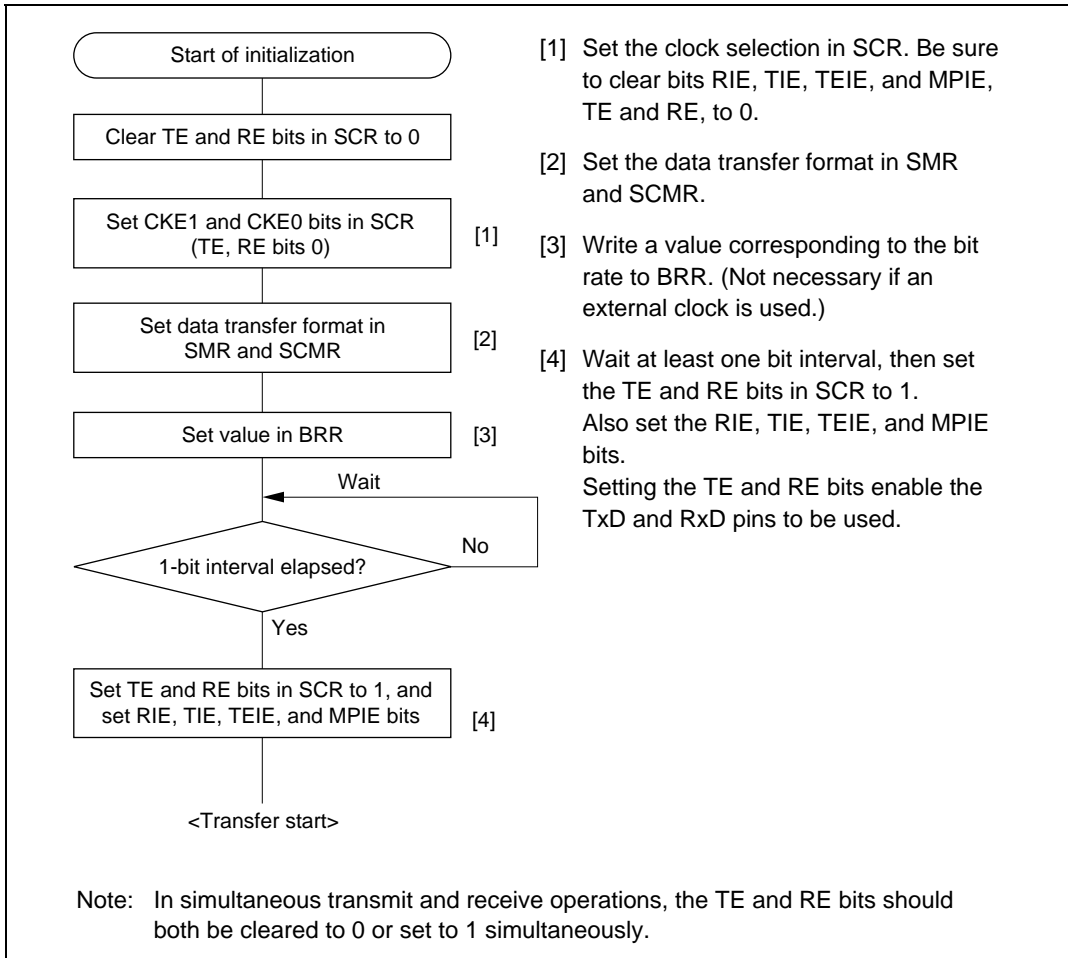


**Figure 14.14 Data Format in Clocked Synchronous Communication (For LSB-First)**

### 14.6.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCK pin can be selected, according to the setting of CKE1 and CKE0 bits in SCR. When the SCI is operated on an internal clock, the serial clock is output from the SCK pin. Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high.

Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described in a sample flowchart in figure 14.15. Do not write to SMR, SCMR, IrCR, or SEMR while the SCI is operating. This also applies to writing the same data as the current register contents. When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change. When the TE bit is cleared to 0, the TDRE flag is set to 1. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.



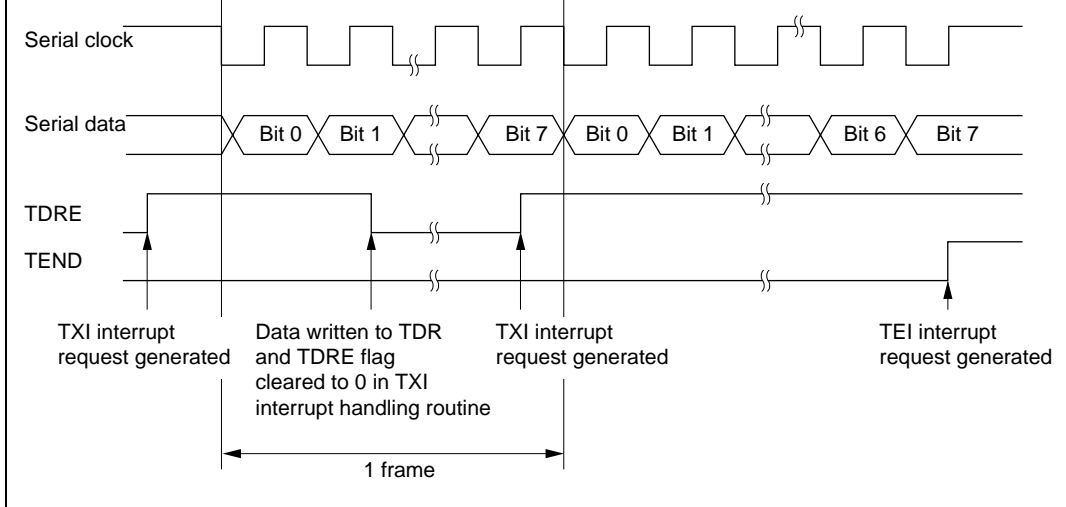
**Figure 14.15 Sample SCI Initialization Flowchart**



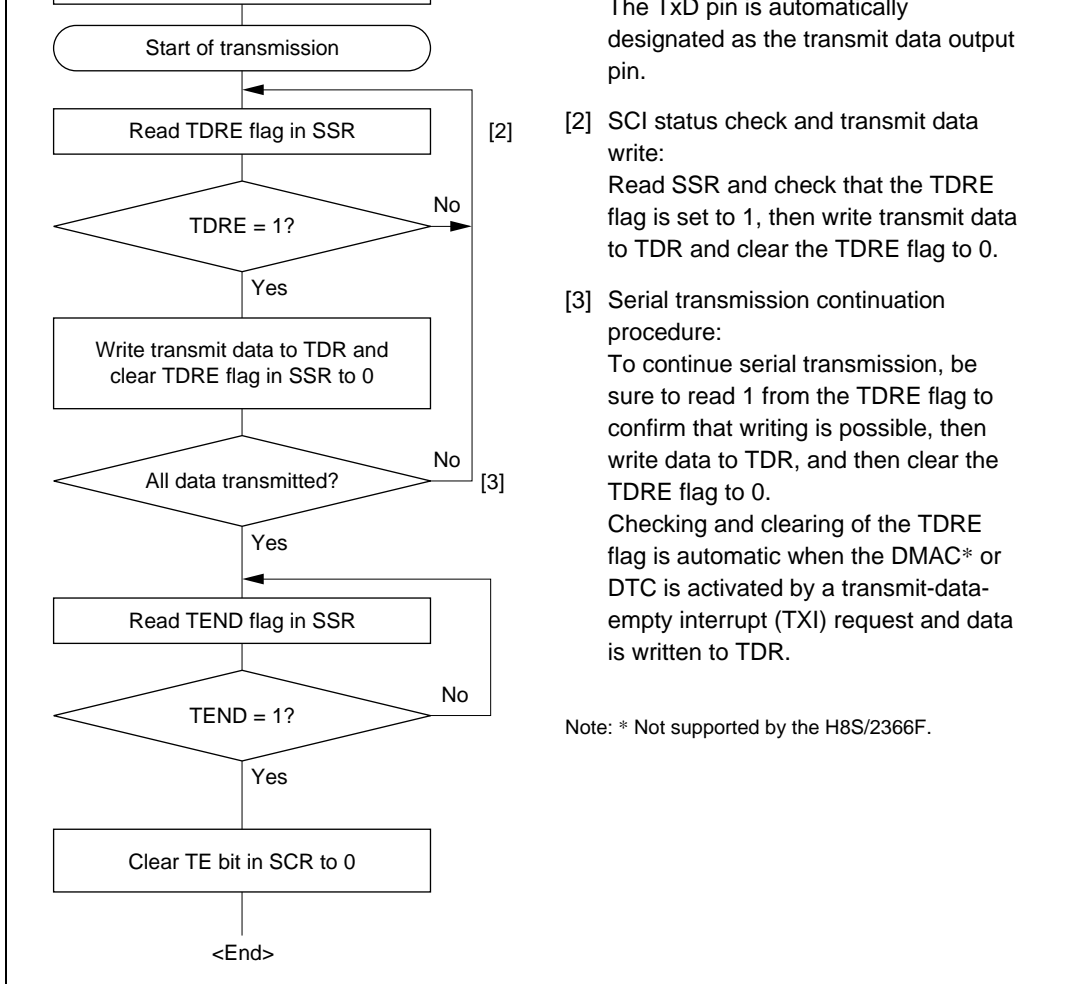
Figure 14.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI interrupt request is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the MSB.
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TxD pin maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 14.17 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.



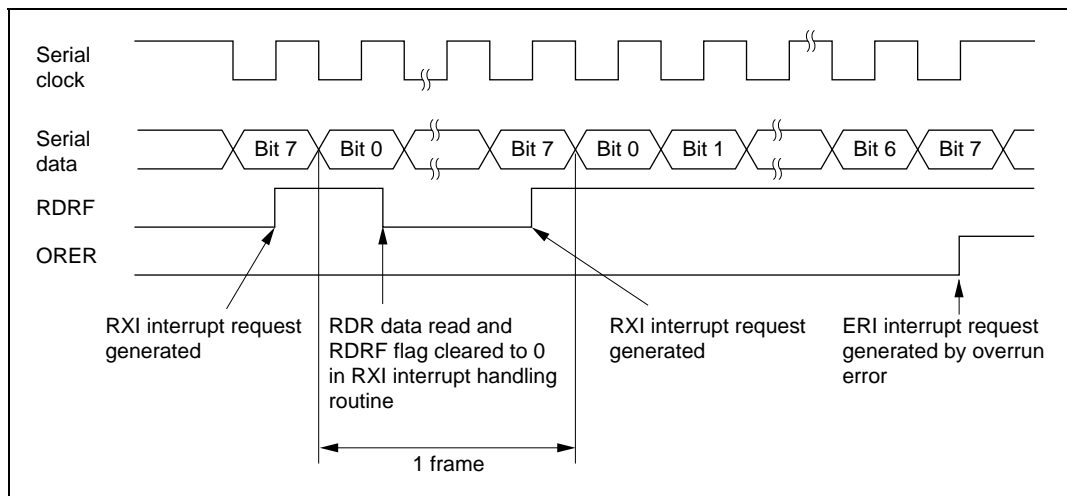
**Figure 14.16 Sample SCI Transmission Operation in Clocked Synchronous Mode**



**Figure 14.17 Sample Serial Transmission Flowchart**

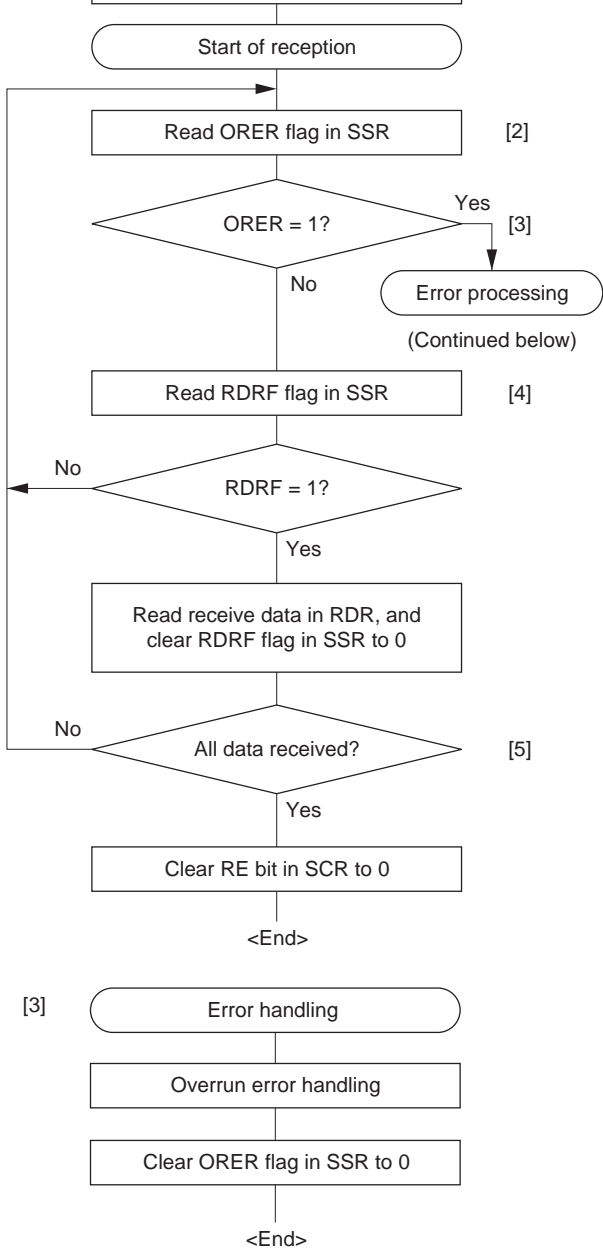
Figure 14.18 shows an example of SCI operation for reception in clocked synchronous mode. In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization in synchronization with a synchronization clock input or output, starts receiving data, and stores the received data in RSR.
2. If an overrun error (when reception of the next data is completed while the RDRF flag is still set to 1) occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is set to 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR. The RDRF flag remains to be set to 1.
3. If reception finishes successfully, the RDRF bit in SSR is set to 1, and receive data is transferred to RDR. If the RIE bit in SCR is set to 1 at this time, an RXI interrupt request is generated. Because the RXI interrupt routine reads the receive data transferred to RDR before reception of the next receive data has finished, continuous reception can be enabled.



**Figure 14.18 Example of SCI Operation in Reception**

Transfer cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 14.19 shows a sample flowchart for serial data reception.



The RxD pin is automatically designated as the receive data input pin.

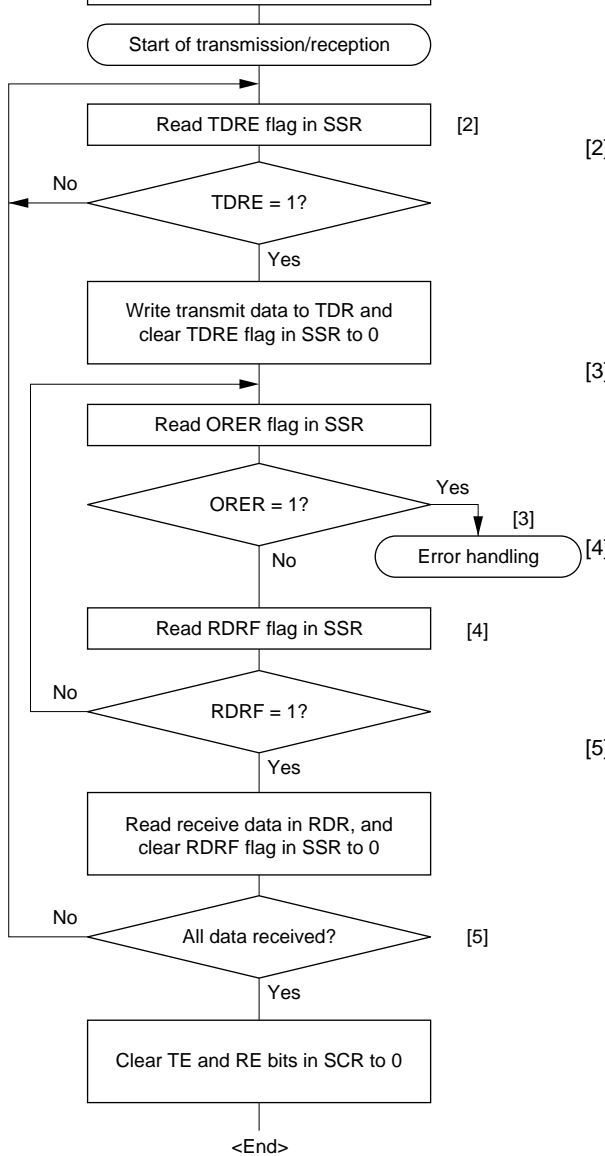
[2] [3] Receive error handling:  
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error handling, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:  
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:  
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. The RDRF flag is cleared automatically when the DMAC or DTC is activated by a receive-data-full interrupt (RXI) request and the RDR value is read.

**Figure 14.19 Sample Serial Reception Flowchart**

Figure 14.20 shows a sample flowchart for simultaneous serial transmit and receive operations. The following procedure should be used for simultaneous serial data transmit and receive operations after the SCI is initialized. To switch from transmit mode to simultaneous transmit and receive mode, after checking that the SCI has finished transmission and the TDRE and TEND flags are set to 1, clear TE to 0. Then simultaneously set TE and RE to 1 with a single instruction. To switch from receive mode to simultaneous transmit and receive mode, after checking that the SCI has finished reception, clear RE to 0. Then after checking that the RDRF and receive error flags (ORER, FER, and PER) are cleared to 0, simultaneously set TE and RE to 1 with a single instruction.



transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:  
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0. Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error handling:  
If a receive error occurs, read the ORER flag in SSR, and after performing the appropriate error handling, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:  
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:  
To continue serial transmission/reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DMAC or DTC is activated by a transmit-data-empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DMAC or DTC is activated by a receive-data-full interrupt (RXI) request and the RDR value is read.

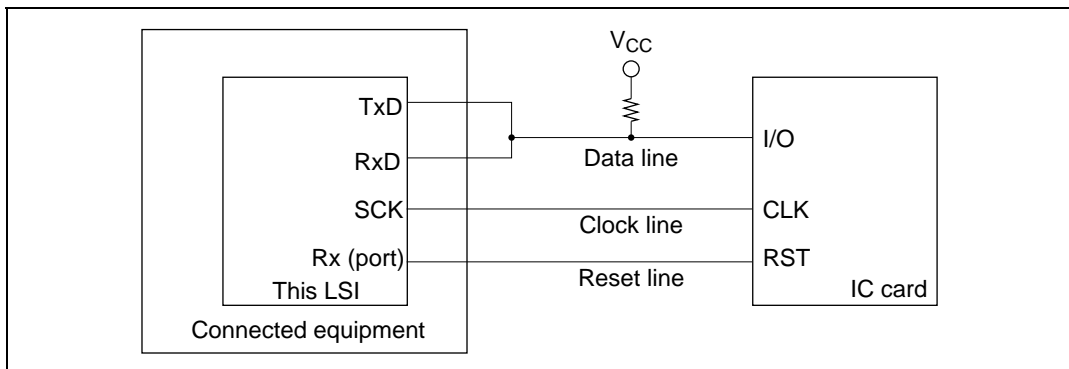
Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE and RE bits to 0, then set both these bits to 1 simultaneously.

**Figure 14.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations**

The SCI supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function. Switching between the normal serial communication interface and the Smart Card interface is carried out by means of a register setting.

### 14.7.1 Pin Connection Example

Figure 14.21 shows an example of connection with the Smart Card. In communication with an IC card, since both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should be connected with the LSI pin. The data transmission line should be pulled up to the  $V_{CC}$  power supply with a resistor. If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out. When the clock generated on the SCI is used by an IC card, the SCK pin output is input to the CLK pin of the IC card. This LSI port output is used as the reset signal.

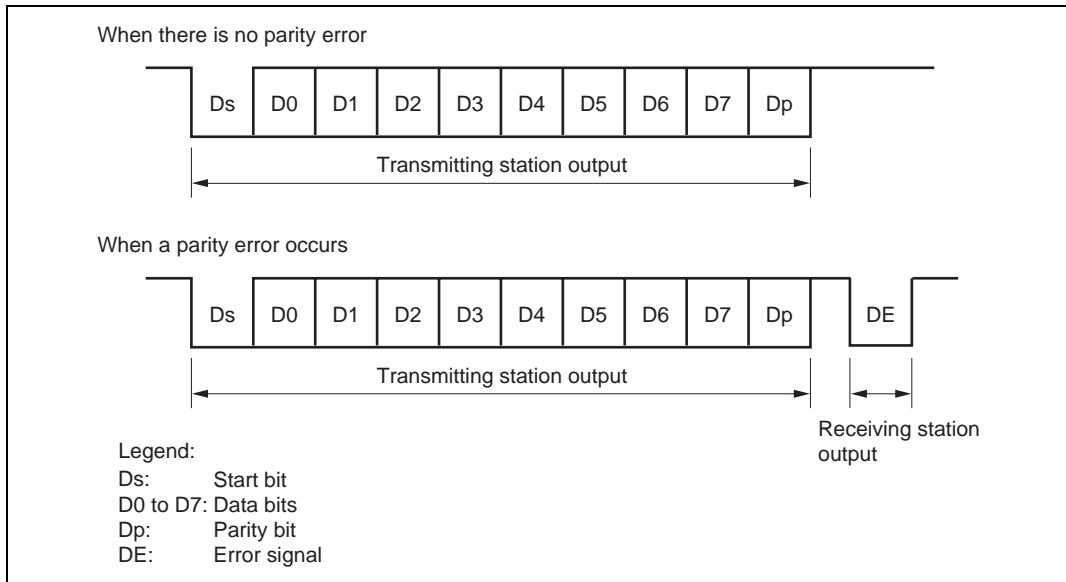


**Figure 14.21 Schematic Diagram of Smart Card Interface Pin Connections**



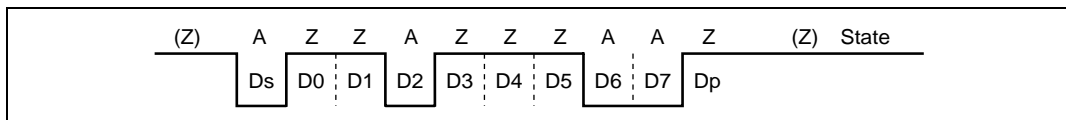
Figure 14.22 shows the transfer data format in Smart Card interface mode.

- One frame consists of 8-bit data plus a parity bit in asynchronous mode.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: time for transfer of 1 bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for one etu period, 10.5 etu after the start bit.
- If an error signal is sampled during transmission, the same data is retransmitted automatically after the elapse of 2 etu or longer.



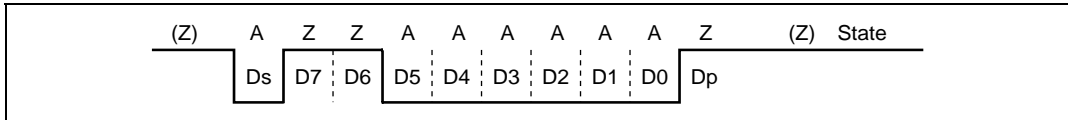
**Figure 14.22 Normal Smart Card Interface Data Format**

Data transfer with the types of IC cards (direct convention and inverse convention) are performed as described in the following.



**Figure 14.23 Direct Convention (SDIR = SINV =  $\overline{O/E} = 0$ )**

The start character data above is H'3B. For the direct convention type, clear the SDIR and SINV bits in SCMR to 0. According to the Smart Card regulations, clear the  $O/\bar{E}$  bit in SMR to 0 to select even parity mode.



**Figure 14.24 Inverse Convention (SDIR = SINV =  $O/\bar{E}$  = 1)**

With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data above is H'3F. For the inverse convention type, set the SDIR and SINV bits in SCMR to 1. According to the Smart Card regulations, even parity mode is the logic 0 level of the parity bit, and corresponds to state Z. In this LSI, the SINV bit inverts only data bits D7 to D0. Therefore, set the  $O/\bar{E}$  bit in SMR to 1 to invert the parity bit for both transmission and reception.

### 14.7.3 Block Transfer Mode

Operation in block transfer mode is the same as that in normal Smart Card interface, except for the following points.

- In reception, though the parity check is performed, no error signal is output even if an error is detected. However, the PER bit in SSR is set to 1 and must be cleared before receiving the parity bit of the next frame.
- In transmission, a guard time of at least 1 etu is left between the end of the parity bit and the start of the next frame.
- In transmission, because retransmission is not performed, the TEND flag is set to 1, 11.5 etu after transmission start.
- As with the normal Smart Card interface, the ERS flag indicates the error signal status, but since error signal transfer is not performed, this flag is always cleared to 0.

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

### 14.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator is used as transmit/receive clock in Smart Card interface. In Smart Card interface mode, the SCI operates on a basic clock with a frequency of 32, 64, 372, or 256 times the bit rate (fixed at 16 times in normal

shown in figure 14.25, by sampling data at the rising-edge of the 16th, 32nd, 186th, or 128th pulse of the basic clock, data can be latched at the middle of the bit. The reception margin is given by the following formula.

$$M = \left| \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100\%$$

Where M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, and 256)

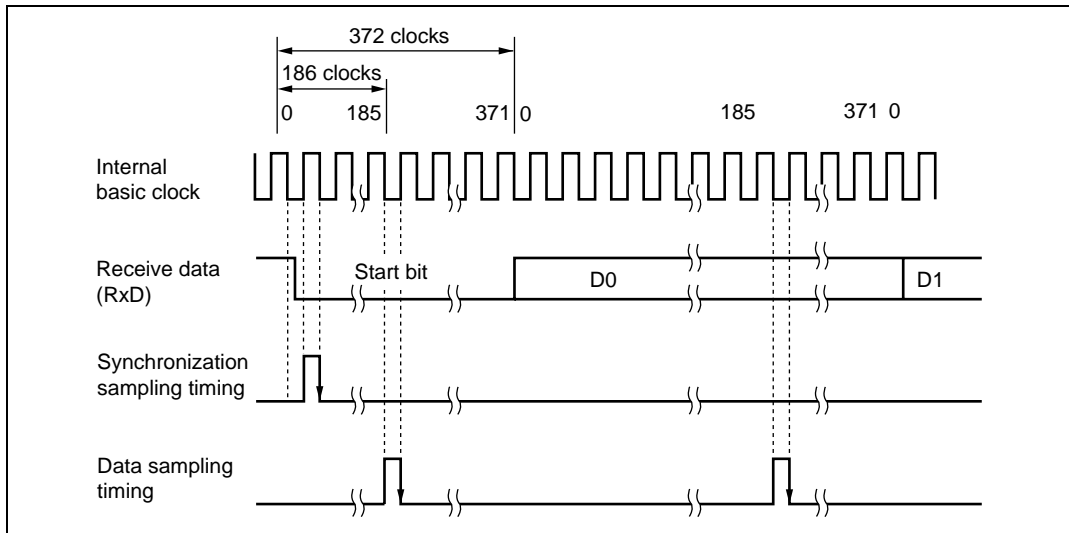
D: Clock duty (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5 and N = 372 in the above formula, the reception margin formula is as follows.

$$\begin{aligned} M &= (0.5 - 1/2 \times 372) \times 100\% \\ &= 49.866\% \end{aligned}$$



**Figure 14.25 Receive Data Sampling Timing in Smart Card Mode  
(Using Clock of 372 Times the Bit Rate)**

Before transmitting and receiving data, initialize the SCI as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

1. Clear the TE and RE bits in SCR to 0.
2. Clear the error flags ERS, PER, and ORER in SSR to 0.
3. Set the GM, BLK,  $O/\bar{E}$ , BCP1, BCP0, CKS1, and CKS0 bits in SMR. Set the PE bit to 1.
4. Set the SMIF, SDIR, and SINV bits in SCCR.  
When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports to SCI pins, and are placed in the high-impedance state.
5. Set the value corresponding to the bit rate in BRR.
6. Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0.  
If the CKE0 bit is set to 1, the clock is output from the SCK pin.
7. Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.

To switch from receive mode to transmit mode, after checking that the SCI has finished reception, initialize the SCI, and clear RE to 0 and set TE to 1. Whether SCI has finished reception can be checked with the RDRF, PER, or ORER flag. To switch from transmit mode to receive mode, after checking that the SCI has finished transmission, initialize the SCI, and clear TE to 0 and set RE to 1. Whether SCI has finished transmission can be checked with the TEND flag.

#### 14.7.6 Data Transmission (Except for Block Transfer Mode)

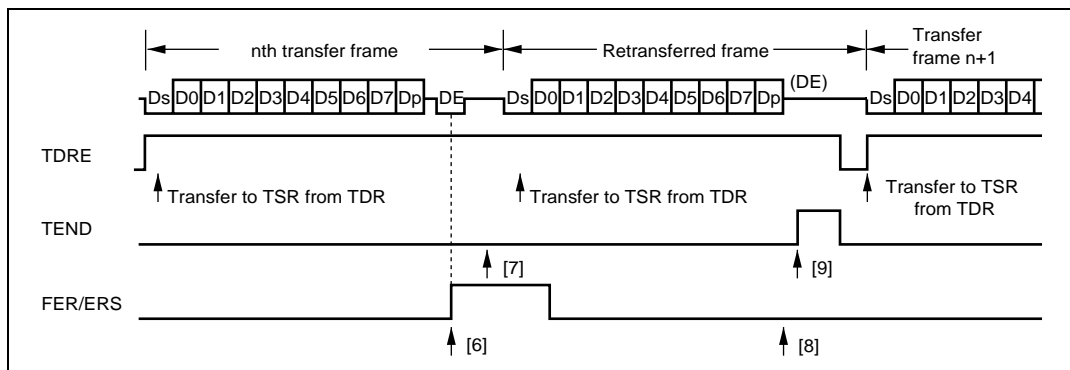
As data transmission in Smart Card interface mode involves error signal sampling and retransmission processing, the operations are different from those in normal serial communication interface mode (except for block transfer mode). Figure 14.26 illustrates the retransfer operation when the SCI is in transmit mode.

1. If an error signal is sampled from the receiving end after transmission of one frame is completed, the ERS bit in SSR is set to 1. If the RIE bit in SCR is set at this time, an ERI interrupt request is generated. The ERS bit in SSR should be cleared to 0 before the next parity bit is sampled.
2. The TEND bit in SSR is not set for a frame for which an error signal is received. Data is retransferred from TDR to TSR, and retransmitted automatically.
3. If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set.

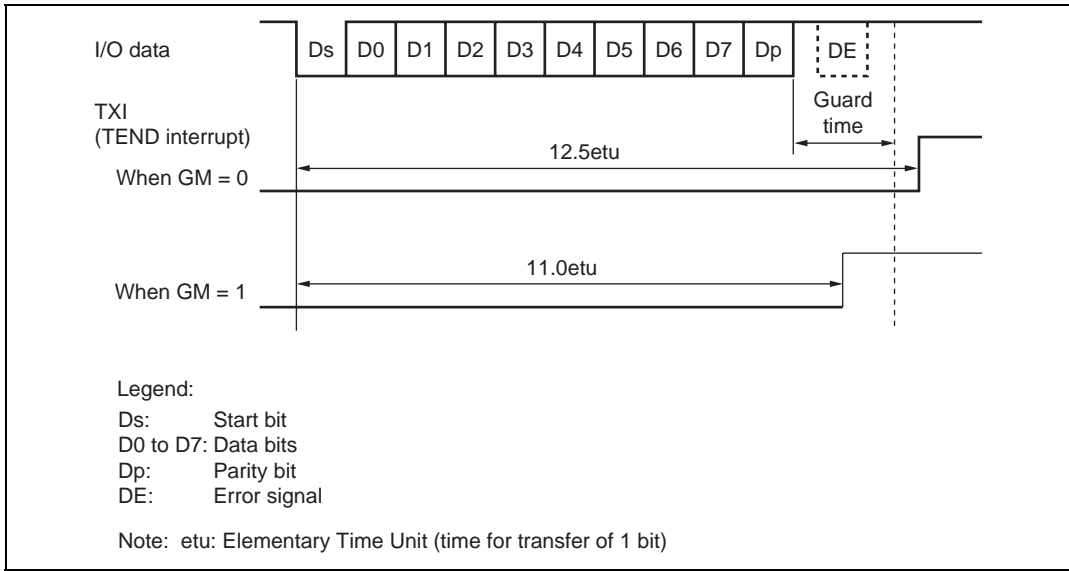
generated. Writing transmit data to TDR transfers the next transmit data.

Figure 14.28 shows a flowchart for transmission. The sequence of transmit operations can be performed automatically by specifying the DTC or DMAC to be activated with a TXI interrupt source. In a transmit operation, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR, and a TXI interrupt will be generated if the TIE bit in SCR has been set to 1. If the TXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data transfer is performed by the DTC or DMAC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

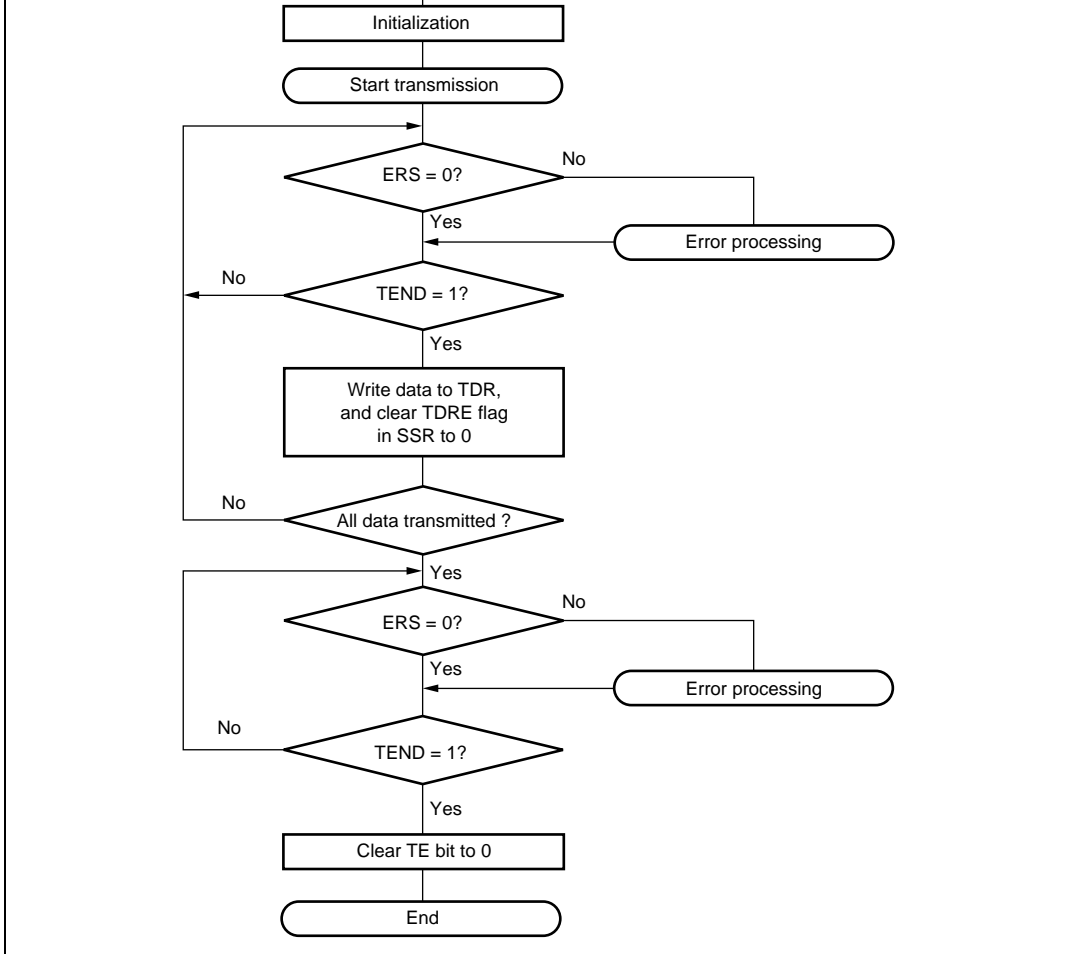
When performing transfer using the DTC or DMAC, it is essential to set and enable the DTC or DMAC before carrying out SCI setting. For details on the DTC or DMAC setting procedures, refer to section 8, Data Transfer Controller (DTC) or section 7, DMA Controller (DMAC).



**Figure 14.26 Retransfer Operation in SCI Transmit Mode**



**Figure 14.27 TEND Flag Generation Timing in Transmission Operation**



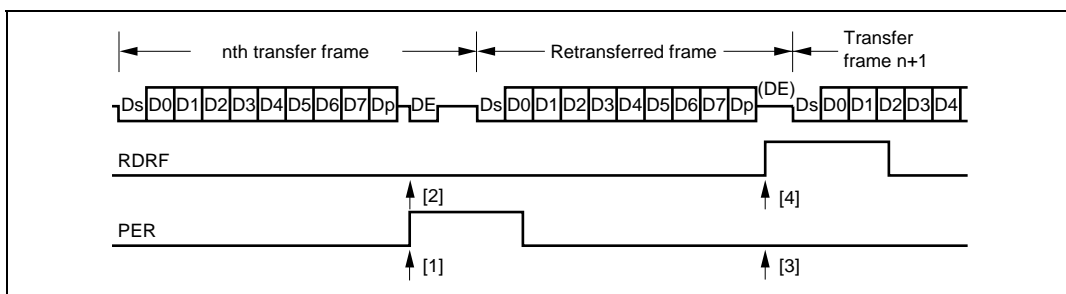
**Figure 14.28 Example of Transmission Processing Flow**

Data reception in Smart Card interface mode uses the same operation procedure as for normal serial communication interface mode. Figure 14.29 illustrates the retransfer operation when the SCI is in receive mode.

1. If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is set at this time, an ERI interrupt request is generated. The PER bit in SSR should be cleared to 0 before the next parity bit is sampled.
2. The RDRF bit in SSR is not set for a frame in which an error has occurred.
3. If no error is found when the received parity bit is checked, the PER bit in SSR is not set to 1.
4. The receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is set at this time, an RXI interrupt request is generated.

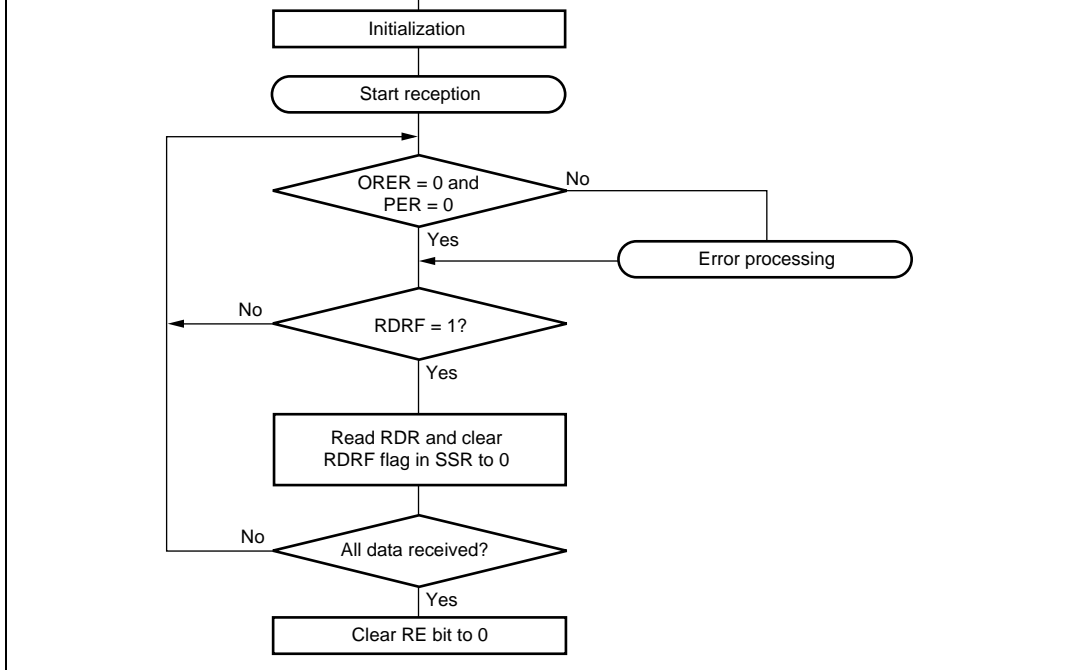
Figure 14.30 shows a flowchart for reception. The sequence of receive operations can be performed automatically by specifying the DTC or DMAC to be activated with an RXI interrupt source. In a receive operation, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the RXI request, and transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC. If an error occurs in receive mode and the ORER or PER flag is set to 1, a transfer error interrupt (ERI) request will be generated, and so the error flag must be cleared to 0. In the event of an error, the DTC or DMAC is not activated and receive data is skipped. Therefore, receive data is transferred for only the specified number of bytes in the event of an error. Even when a parity error occurs in receive mode and the PER flag is set to 1, the data that has been received is transferred to RDR and can be read from there.

Note: For details on receive operations in block transfer mode, refer to section 14.4, Operation in Asynchronous Mode.



**Figure 14.29 Retransfer Operation in SCI Receive Mode**



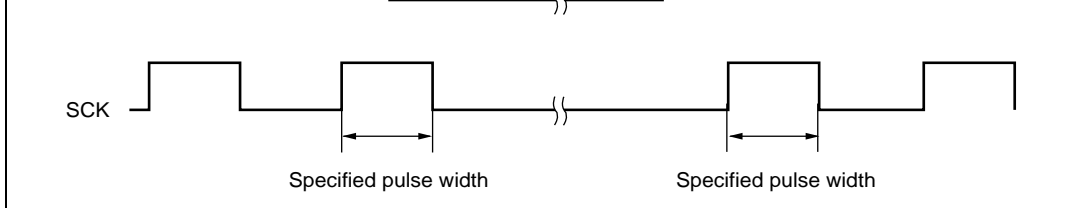


**Figure 14.30 Example of Reception Processing Flow**

### 14.7.8 Clock Output Control

When the GM bit in SMR is set to 1, the clock output level can be fixed with bits CKE1 and CKE0 in SCR. At this time, the minimum clock pulse width can be made the specified width.

Figure 14.31 shows the timing for fixing the clock output level. In this example, GM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.



**Figure 14.31 Timing for Fixing Clock Output Level**

When turning on the power or switching between Smart Card interface mode and software standby mode, the following procedures should be followed in order to maintain the clock duty.

**Powering On:** To secure the clock duty from power-on, the following switching procedure should be followed.

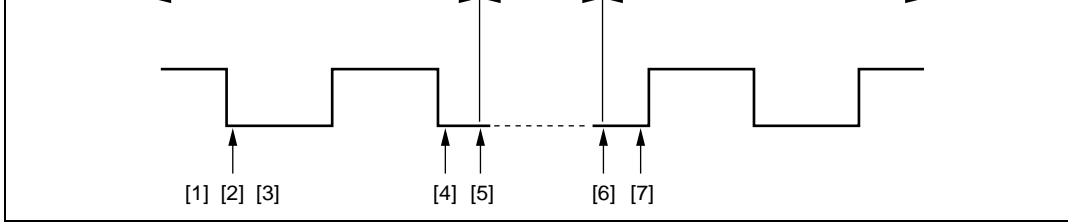
1. The initial state is port input and high impedance. Use a pull-up resistor or pull-down resistor to fix the potential.
2. Fix the SCK pin to the specified output level with the CKE1 bit in SCR.
3. Set SMR and SCMR, and switch to smart card mode operation.
4. Set the CKE0 bit in SCR to 1 to start clock output.

**When changing from smart card interface mode to software standby mode:**

1. Set the data register (DR) and data direction register (DDR) corresponding to the SCK pin to the value for the fixed output state in software standby mode.
2. Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transmit/receive operation. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.
3. Write 0 to the CKE0 bit in SCR to halt the clock.
4. Wait for one serial clock period.  
During this interval, clock output is fixed at the specified level, with the duty preserved.
5. Make the transition to the software standby state.

**When returning to smart card interface mode from software standby mode:**

1. Exit the software standby state.
2. Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started with the normal duty.

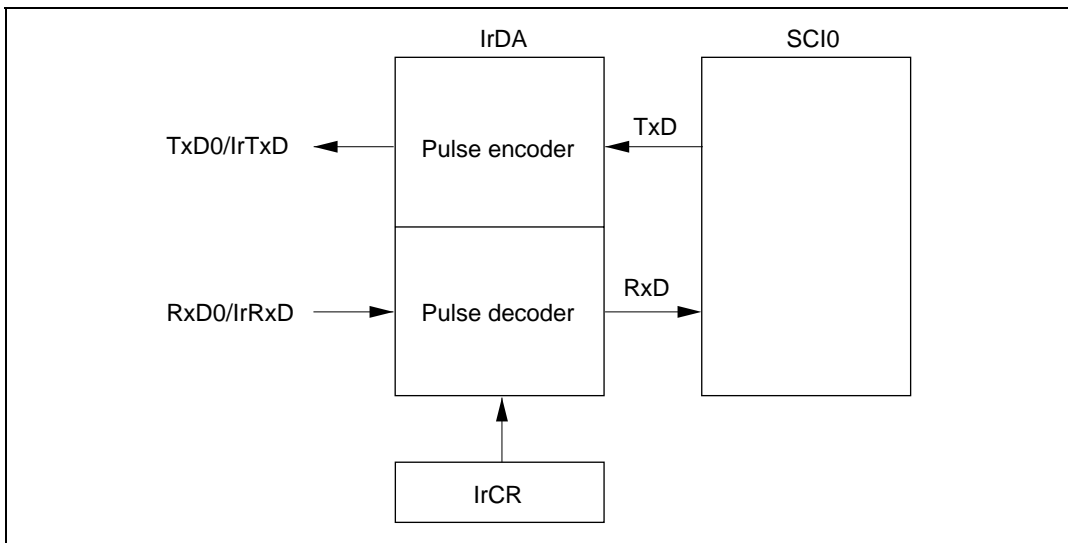


**Figure 14.32 Clock Halt and Restart Procedure**

When the IrDA function is enabled with bit IrE in IrCR, the SCI\_0 TxD0 and RxD0 signals are subjected to waveform encoding/decoding conforming to IrDA specification version 1.0 (IrTxD and IrRxD pins). By connecting these pins to an infrared transceiver/receiver, it is possible to implement infrared transmission/reception conforming to the IrDA specification version 1.0 system.

In the IrDA specification version 1.0 system, communication is started at a transfer rate of 9600 bps, and subsequently the transfer rate can be varied as necessary. As the IrDA interface in this LSI does not include a function for varying the transfer rate automatically, the transfer rate setting must be changed by software.

Figure 14.33 shows a block diagram of the IrDA function.



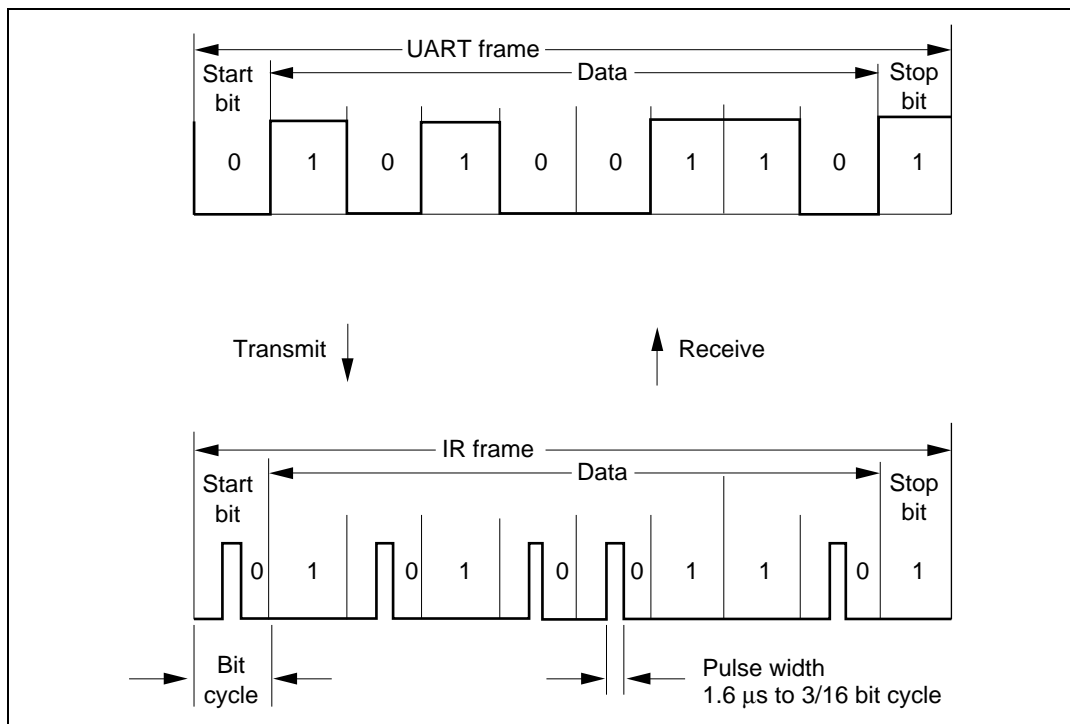
**Figure 14.33 Block Diagram of IrDA**

**Transmission:** In transmission, the output signal (UART frame) from the SCI is converted to an IR frame by the IrDA interface (see figure 14.34).

When the serial data is 0, a high pulse of 3/16 the bit rate (interval equivalent to the width of one bit) is output (initial value). The high-level pulse can be varied according to the setting of bits IrCKS2 to IrCKS0 in IrCR.

be set for a high pulse width with a minimum value of 1.41  $\mu\text{s}$ .

When the serial data is 1, no pulse is output.



**Figure 14.34 IrDA Transmit/Receive Operations**

**Reception:** In reception, IR frame data is converted to a UART frame by the IrDA interface, and input to the SCI.

When a high pulse is detected, 0 data is output, and if there is no pulse during a one-bit interval, 1 data is output. Note that a pulse shorter than the minimum pulse width of 1.41  $\mu\text{s}$  will be identified as a 0 signal.

**High Pulse Width Selection:** Table 14.12 shows possible settings for bits IrCKS2 to IrCKS0 (minimum pulse width), and operating frequencies of this LSI and bit rates, for making the pulse width shorter than 3/16 times the bit rate in transmission.

Operating Frequency φ (MHz)	Bit Rate (bps) (Above) / Bit Period × 3/10 (μs) (Below)					
	2400	9600	19200	38400	57600	115200
8	100	100	100	100	100	100
9.8304	100	100	100	100	100	100
10	100	100	100	100	100	100
12	101	101	101	101	101	101
12.288	101	101	101	101	101	101
14	101	101	101	101	101	101
14.7456	101	101	101	101	101	101
16	101	101	101	101	101	101
16.9344	101	101	101	101	101	101
17.2032	101	101	101	101	101	101
18	101	101	101	101	101	101
19.6608	101	101	101	101	101	101
20	101	101	101	101	101	101
25	110	110	110	110	110	—
30	110	110	110	110	110	—
33	110	110	110	110	110	—
34*	110	110	110	110	110	—

Legend:

—: A bit rate setting cannot be made on the SCI side.

Note: \* Supported only by the H8S/2368 0.18 μm F-ZTAT Group.

## 14.9.1 Interrupts in Normal Serial Communication Interface Mode

Table 14.13 shows the interrupt sources in normal serial communication interface mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled using the enable bits in SCR.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DTC or DMAC to perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt request can activate the DTC or DMAC to perform data transfer. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC.

A TEI interrupt is generated when the TEND flag is set to 1 while the TEIE bit is set to 1. If a TEI interrupt and a TXI interrupt are generated simultaneously, the TXI interrupt has priority for acceptance. However, note that if the TDRE and TEND flags are cleared simultaneously by the TXI interrupt routine, the SCI cannot branch to the TEI interrupt routine later.

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
0	ERI0	Receive Error	ORER, FER, PER	Not possible	Not possible	High ↑
	RXI0	Receive Data Full	RDRF	Possible	Possible	
	TXI0	Transmit Data Empty	TDRE	Possible	Possible	
	TEI0	Transmission End	TEND	Not possible	Not possible	
1	ERI1	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI1	Receive Data Full	RDRF	Possible	Possible	
	TXI1	Transmit Data Empty	TDRE	Possible	Possible	
	TEI1	Transmission End	TEND	Not possible	Not possible	
2	ERI2	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI2	Receive Data Full	RDRF	Possible	Not possible	
	TXI2	Transmit Data Empty	TDRE	Possible	Not possible	
	TEI2	Transmission End	TEND	Not possible	Not possible	
3	ERI3	Receive Error	ORER, FER, PER	Not possible	Not possible	
	RXI3	Receive Data Full	RDRF	Possible	Not possible	
	TXI3	Transmit Data Empty	TDRE	Possible	Not possible	
	TEI3	Transmission End	TEND	Not possible	Not possible	
4	ERI4	Receive Error	ORER, FER, PER	Not possible	Not possible	Low
	RXI4	Receive Data Full	RDRF	Possible	Not possible	
	TXI4	Transmit Data Empty	TDRE	Possible	Not possible	
	TEI4	Transmission End	TEND	Not possible	Not possible	



Table 14.14 shows the interrupt sources in Smart Card interface mode. The transmit end interrupt (TEI) request cannot be used in this mode.

**Table 14.14 Interrupt Sources**

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
0	ERI0	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	High ↑
	RX10	Receive Data Full	RDRF	Possible	Possible	
	TX10	Transmit Data Empty	TEND	Possible	Possible	
1	ERI1	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RX11	Receive Data Full	RDRF	Possible	Possible	
	TX11	Transmit Data Empty	TEND	Possible	Possible	
2	ERI2	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RX12	Receive Data Full	RDRF	Possible	Not possible	
	TX12	Transmit Data Empty	TEND	Possible	Not possible	
3	ERI3	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	
	RX13	Receive Data Full	RDRF	Possible	Not possible	
	TX13	Transmit Data Empty	TEND	Possible	Not possible	
4	ERI4	Receive Error, detection	ORER, PER, ERS	Not possible	Not possible	Low
	RX14	Receive Data Full	RDRF	Possible	Not possible	
	TX14	Transmit Data Empty	TEND	Possible	Not possible	

In Smart Card interface mode, as in normal serial communication interface mode, transfer can be carried out using the DTC or DMAC. In transmit operations, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR, and a TXI interrupt is generated. If the TXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data transfer is performed by the DTC or DMAC. In the event of an error, the SCI retransmits the same data automatically. During this period, the TEND flag remains cleared to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC will automatically transmit the specified number of bytes in the event of an error, including retransmission. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

refer to section 8, Data Transfer Controller (DTC) or section 7, DMA Controller (DMAC).

In receive operations, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC or DMAC activation source, the DTC or DMAC will be activated by the RXI request, and transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC or DMAC. If an error occurs, an error flag is set but the RDRF flag is not. Consequently, the DTC or DMAC is not activated, but instead, an ERI interrupt request is sent to the CPU. Therefore, the error flag should be cleared.

## **14.10 Usage Notes**

### **14.10.1 Module Stop Mode Setting**

SCI operation can be disabled or enabled using the module stop control register. The initial setting is for SCI operation to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 23, Power-Down Modes.

### **14.10.2 Break Detection and Processing**

When framing error detection is performed, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is set, and the PER flag may also be set. Note that, since the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

### **14.10.3 Mark State and Break Sending**

When TE is 0, the TxD pin is used as an I/O port whose direction (input or output) and level are determined by DR and DDR. This can be used to set the TxD pin to mark state or send a break during serial data transmission. To maintain the communication line at mark state until TE is set to 1, set both DDR and DR to 1. Since TE is cleared to 0 at this point, the TxD pin becomes an I/O port, and 1 is output from the TxD pin. To send a break during serial transmission, first set DDR to 1 and clear DR to 0, and then clear TE to 0. When TE is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission. Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

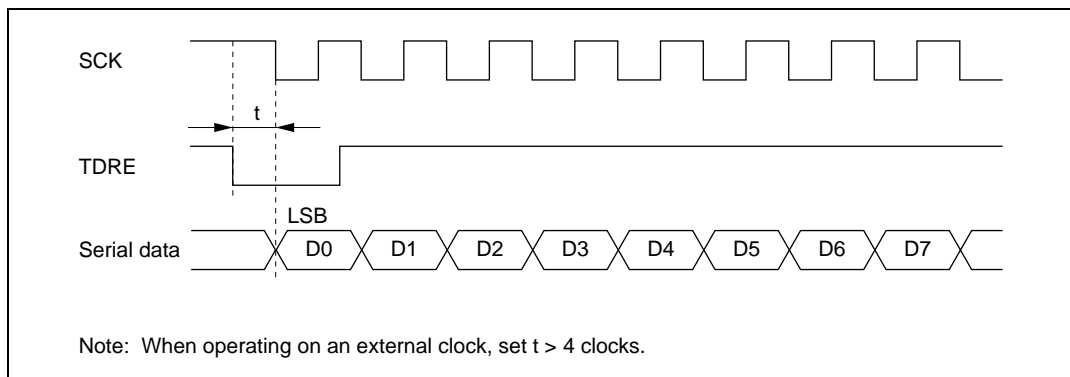
### 14.10.5 Relation between Writes to TDR and the TDRE Flag

The TDRE flag in SSR is a status flag that indicates that transmit data has been transferred from TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to 1.

Data can be written to TDR regardless of the state of the TDRE flag. However, if new data is written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost since it has not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is set to 1 before writing transmit data to TDR.

### 14.10.6 Restrictions on Use of DMAC or DTC

- When an external clock source is used as the serial clock, the transmit clock should not be input until at least five  $\phi$  clock cycles after TDR is updated by the DMAC or DTC. Abnormal operation may occur if the transmit clock is input within 4  $\phi$  clocks after TDR is updated. (figure 14.35)
- When RDR is read by the DMAC or DTC, be sure to set the activation source to the relevant SCI receive-data-full interrupt (RXI).



**Figure 14.35 Example of Synchronous Transmission Using DTC**

- Transmission

Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making a module stop mode or software standby mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode or software standby mode depend on the port settings, and become high-level output after the relevant mode is cleared. If a transition is made during transmission, the data being transmitted will be undefined.

When transmitting without changing the transmit mode after the relevant mode is cleared, transmission can be started by setting TE to 1 again, and performing the following sequence: SSR read → TDR write → TDRE clearance. To transmit with a different transmit mode after clearing the relevant mode, the procedure must be started again from initialization.

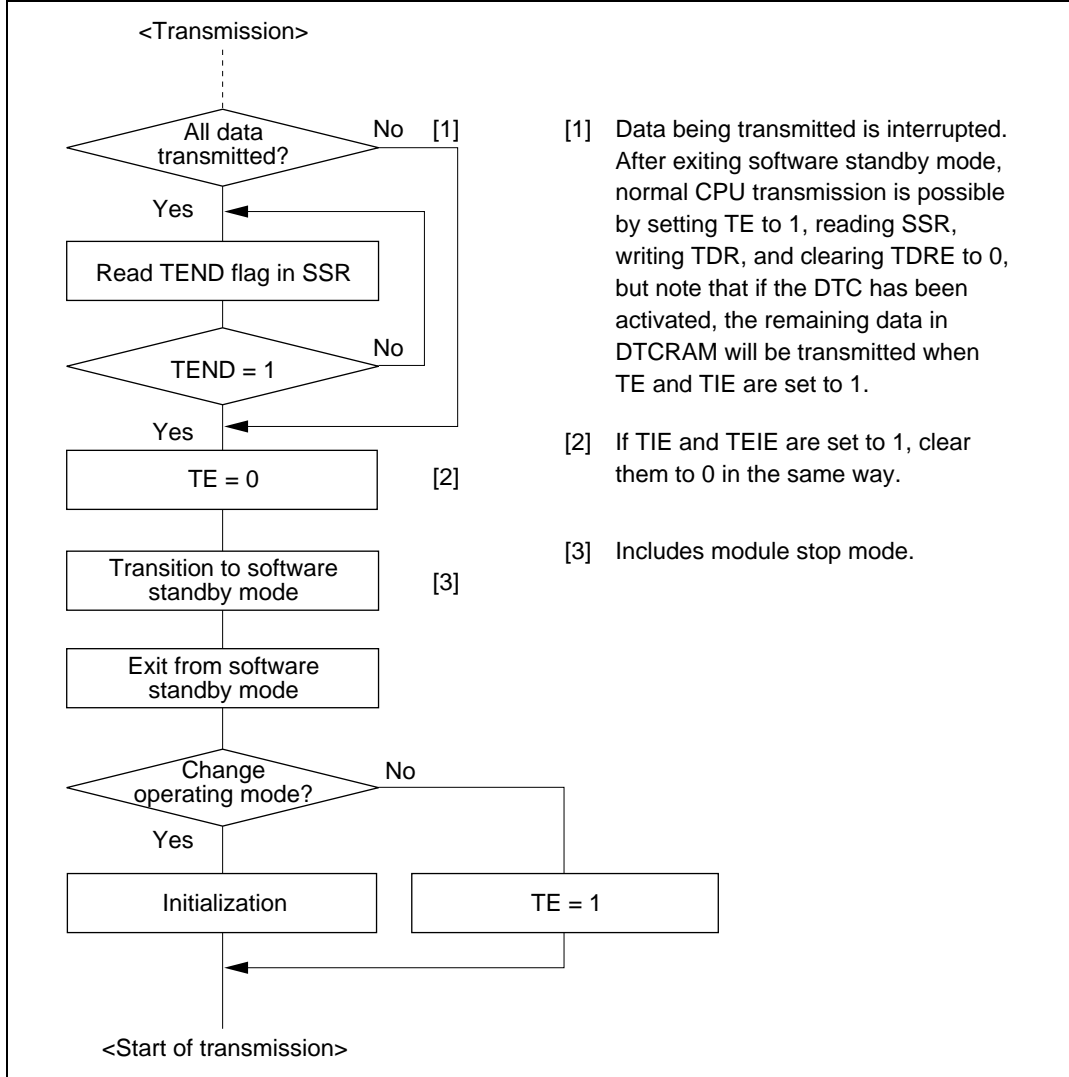
Figure 14.36 shows a sample flowchart for mode transition during transmission. Port pin states during mode transition are shown in figures 14.37 and 14.38.

Operation should also be stopped (by clearing TE, TIE, and TEIE to 0) before making a transition from transmission by DTC transfer to module stop mode or software standby mode transition. To perform transmission with the DTC after the relevant mode is cleared, setting TE and TIE to 1 will set the TXI flag and start DTC transmission.

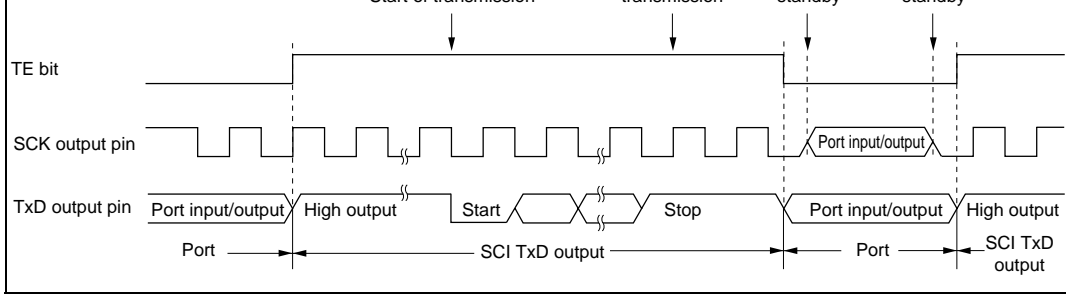
- Reception

Receive operation should be stopped (by clearing RE to 0) before making a module stop mode or software standby mode transition. RSR, RDR, and SSR are reset. If a transition is made during reception, the data being received will be invalid.

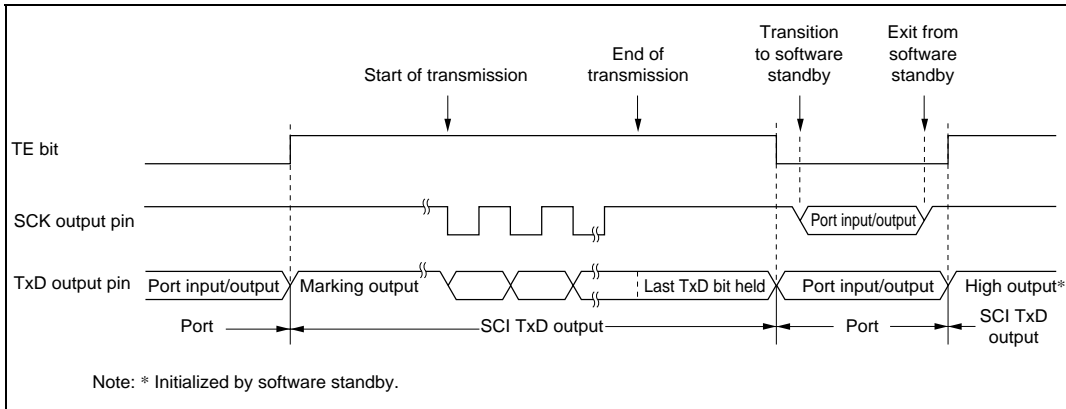
To continue receiving without changing the reception mode after the relevant mode is cleared, set RE to 1 before starting reception. To receive with a different receive mode, the procedure must be started again from initialization.



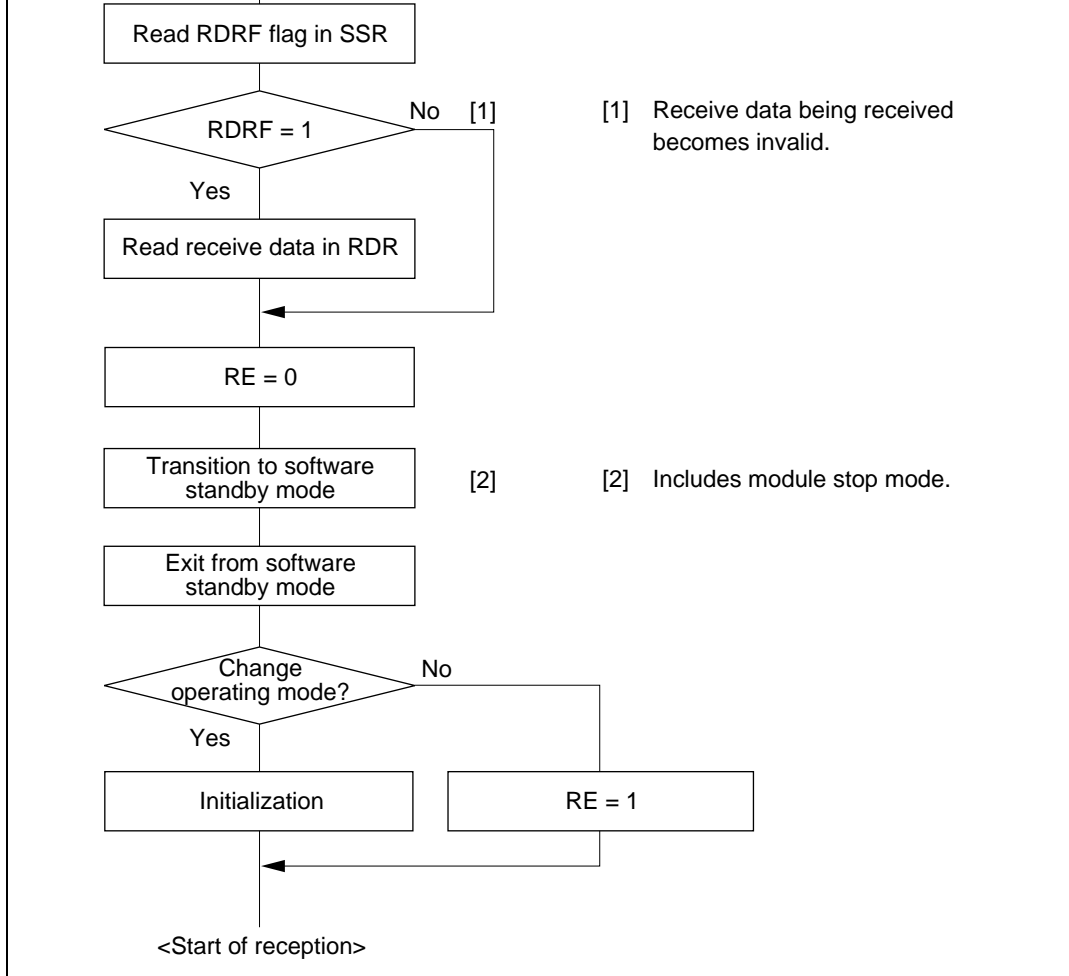
**Figure 14.36 Sample Flowchart for Mode Transition during Transmission**



**Figure 14.37 Port Pin States during Mode Transition  
(Internal Clock, Asynchronous Transmission)**



**Figure 14.38 Port Pin States during Mode Transition  
(Internal Clock, Synchronous Transmission)**



**Figure 14.39 Sample Flowchart for Mode Transition during Reception**





An I<sup>2</sup>C bus interface is an option. When using the optional functions, take notice of the following item:

For the masked ROM version, 'W' is added to the model name of the product that uses optional functions.

For example: HD6432365WTE

This LSI has a two-channel I<sup>2</sup>C bus interface,

The I<sup>2</sup>C bus interface conforms to and provides a subset of the NXP Semiconductors I<sup>2</sup>C bus (inter-IC bus) interface (Rev.03) standard and fast mode functions. The register configuration that controls the I<sup>2</sup>C bus differs partly from the NXP Semiconductors configuration, however.

Figure 15.1 shows a block diagram of the I<sup>2</sup>C bus interface2.

Figure 15.2 shows an example of I/O pin connections to external circuits.

## 15.1 Features

- Continuous transmission/reception

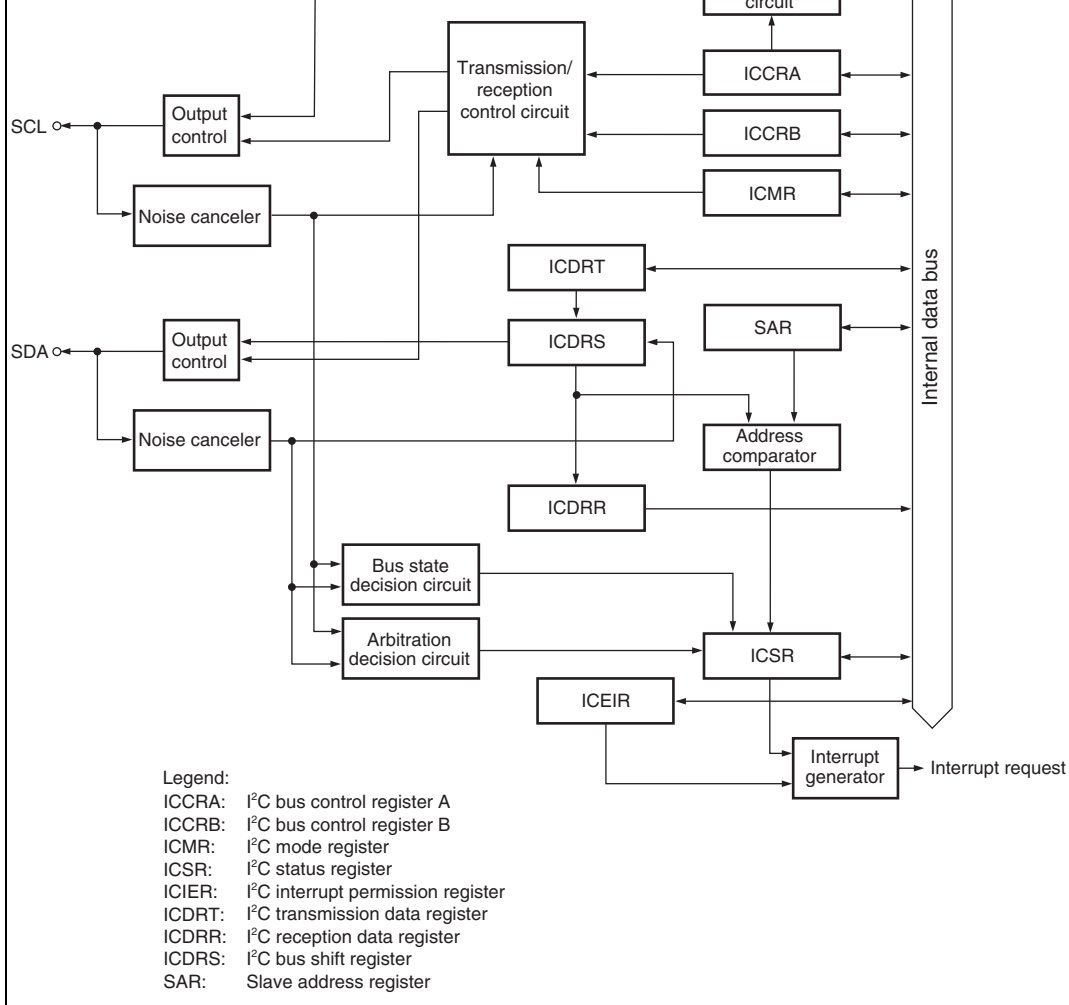
Since the shift register, transmit data register, and receive data register are independent from each other, the continuous transmission/reception can be performed.

- Start and stop conditions generated automatically in master mode
- Selection of acknowledge output levels when receiving
- Automatic loading of acknowledge bit when transmitting
- Bit synchronization/wait function

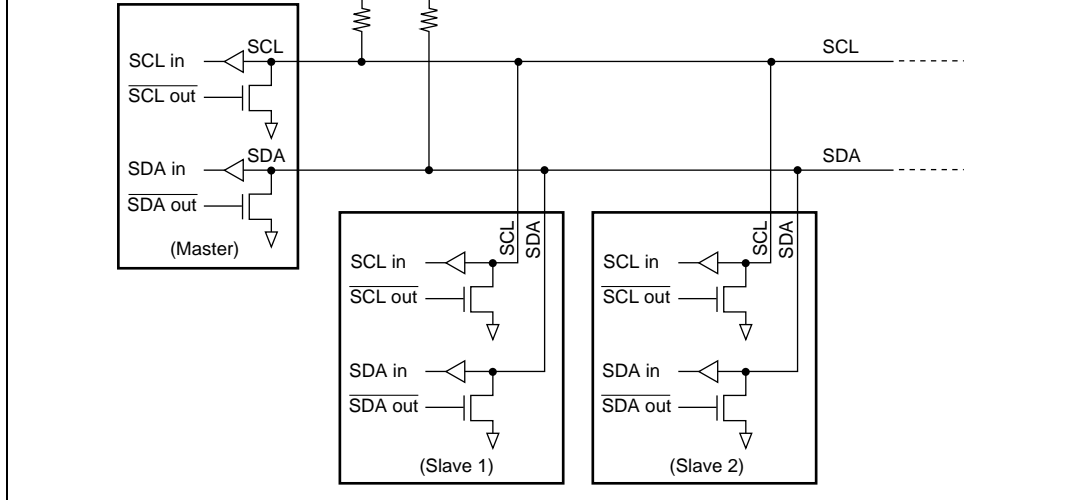
In master mode, the state of SCL is monitored per bit, and the timing is synchronized automatically

If transmission/reception is not yet possible, set the SCL to low until preparations are completed.

- Six interrupt sources
  - Transmit-data-empty (including slave-address match), transmit-end, receive-data-full (including slave-address match), arbitration lost, NACK detection, and stop condition detection
- Direct bus drive
  - Two pins, SCL and SDA pins function as NMOS open-drain outputs.



**Figure 15.1 Block Diagram of I<sup>2</sup>C Bus Interface2**



**Figure 15.2 External Circuit Connections of I/O Pins**

## 15.2 Input/Output Pins

Table 15.1 summarizes the input/output pins used by the I<sup>2</sup>C bus interface.

**Table 15.1 Pin Configuration**

Name	Abbreviation	I/O	Function
Serial clock	SCL0	I/O	IIC2_0 serial clock input/output
Serial data	SDA0	I/O	IIC2_0 serial data input/output
Serial clock	SCL1	I/O	IIC2_1 serial clock input/output
Serial data	SDA1	I/O	IIC2_1 serial data input/output

Note: The pin symbols are represented as SCL and SDA; channel numbers are omitted in this manual.

The I<sup>2</sup>C bus interface has the following registers.

- I<sup>2</sup>C bus control register A\_0 (ICCRA\_0)
- I<sup>2</sup>C bus control register B\_0 (ICCRB\_0)
- I<sup>2</sup>C bus mode register\_0 (ICMR\_0)
- I<sup>2</sup>C bus interrupt enable register\_0 (ICIER\_0)
- I<sup>2</sup>C bus status register\_0 (ICSR\_0)
- I<sup>2</sup>C bus slave address register\_0 (SAR\_0)
- I<sup>2</sup>C bus transmit data register\_0 (ICDRT\_0)
- I<sup>2</sup>C bus receive data register\_0 (ICDRR\_0)
- I<sup>2</sup>C bus shift register\_0 (ICDRS\_0)
- I<sup>2</sup>C bus control register A\_1 (ICCRA\_1)
- I<sup>2</sup>C bus control register B\_1 (ICCRB\_1)
- I<sup>2</sup>C bus mode register\_1 (ICMR\_1)
- I<sup>2</sup>C bus interrupt enable register\_1 (ICIER\_1)
- I<sup>2</sup>C bus status register\_1 (ICSR\_1)
- I<sup>2</sup>C bus slave address register\_1 (SAR\_1)
- I<sup>2</sup>C bus transmit data register\_1 (ICDRT\_1)
- I<sup>2</sup>C bus receive data register\_1 (ICDRR\_1)
- I<sup>2</sup>C bus shift register\_1 (ICDRS\_1)

ICCRA is an 8-bit readable/writable register that enables or disables the I<sup>2</sup>C bus interface, controls transmission or reception, and selects master or slave mode, transmission or reception, and transfer clock frequency in master mode.

Bit	Bit Name	Initial Value	R/W	Description
7	ICE	0	R/W	I <sup>2</sup> C Bus Interface Enable 0: This module is halted. 1: This bit is enabled for transfer operations. (SCL and SDA pins are bus drive state.)
6	RCVD	0	R/W	Reception Disable This bit enables or disables the next operation when TRS is 0 and ICDRR is read. 0: Enables next reception 1: Disables next reception
5	MST	0	R/W	Master/Slave Select
4	TRS	0	R/W	Transmit/Receive Select When arbitration is lost in master mode, MST and TRS are both reset by hardware, causing a transition to slave receive mode. Modification of the TRS bit should be made between transfer frames. In addition, TRS is set to 1 automatically in slave receive mode when the seventh bit of the start condition matches the slave address set in SAR and the eighth bit is set to 1. Operating modes are described below according to MST and TRS combination. 00: Slave receive mode 01: Slave transmit mode 10: Master receive mode 11: Master transmit mode
3	CKS3	0	R/W	Transfer clock select 3 to 0
2	CKS2	0	R/W	In master mode, these bits should be set according to the necessary transfer rate. In slave mode, they are used to secure the data setup time in transmit mode. The data setup time is 10 t <sub>cy</sub> when CKS3 is cleared to 0; 20 t <sub>cy</sub> when CKS3 is set to 1.
1	CKS1	0	R/W	
0	CKS0	0	R/W	

				Transfer Rate							
CKS3	CKS2	CKS1	CKS0	Clock	$\phi =$ 8MHz	$\phi =$ 10MHz	$\phi =$ 20MHz	$\phi =$ 25 MHz	$\phi =$ 33 MHz	$\phi =$ 34 MHz <sup>*1</sup>	
0 <sup>*3</sup>	0 <sup>*3</sup>	0	0	$\phi/28$	286 kHz	357 kHz	714 kHz <sup>*2</sup>	893 kHz <sup>*2</sup>	1179 kHz <sup>*2</sup>	1214 kHz <sup>*2</sup>	
			1	$\phi/40$	200 kHz	250 kHz	500 kHz <sup>*2</sup>	625 kHz <sup>*2</sup>	825 kHz <sup>*2</sup>	850 kHz <sup>*2</sup>	
		1	0	$\phi/48$	167 kHz	208 kHz	417 kHz <sup>*2</sup>	521 kHz <sup>*2</sup>	688 kHz <sup>*2</sup>	708 kHz <sup>*2</sup>	
			1	$\phi/64$	125 kHz	156 kHz	313 kHz	391 kHz <sup>*2</sup>	516 kHz <sup>*2</sup>	531 kHz <sup>*2</sup>	
	1	0	0	$\phi/168$	47.6 kHz	59.5 kHz	119 kHz	149 kHz	196 kHz	202 kHz	
			1	$\phi/100$	80.0 kHz	100 kHz	200 kHz	250 kHz	330 kHz	340 kHz	
		1	0	$\phi/112$	71.4 kHz	89.3 kHz	179 kHz	223 kHz	295 kHz	304 kHz	
			1	$\phi/128$	62.5 kHz	78.1 kHz	156 kHz	195 kHz	258 kHz	266 kHz	
1	0	0	0	$\phi/56$	143 kHz	179 kHz	357 kHz	446 kHz <sup>*2</sup>	589 kHz <sup>*2</sup>	607 kHz <sup>*2</sup>	
			1	$\phi/80$	100 kHz	125 kHz	250 kHz	313 kHz	413 kHz <sup>*2</sup>	425 kHz <sup>*2</sup>	
		1	0	$\phi/96$	83.3 kHz	104 kHz	208 kHz	260 kHz	344 kHz	354 kHz	
			1	$\phi/128$	62.5 kHz	78.1 kHz	156 kHz	195 kHz	258 kHz	266 kHz	
	1	0	0	$\phi/336$	23.8 kHz	29.8 kHz	59.5 kHz	74.4 kHz	98.2 kHz	101 kHz	
			1	$\phi/200$	40.0 kHz	50.0 kHz	100 kHz	125 kHz	165 kHz	170 kHz	
		1	0	$\phi/224$	35.7 kHz	44.6 kHz	89.3 kHz	112 kHz	147 kHz	152 kHz	
			1	$\phi/256$	31.3 kHz	39.1 kHz	78.1 kHz	97.7 kHz	129 kHz	133 kHz	

- Notes:
1. Supported only by the H8S/2368 0.18  $\mu\text{m}$  F-ZTAT Group.
  2. Does not conform to the I<sup>2</sup>C bus interface specification (standard mode: max. 100 kHz, fast mode: max. 400 kHz).
  3. If CKS3 and CKS2 are both cleared to 0 (7.5 tcyc bit synchronization) and the operating frequency is 20 MHz or greater, it may not be possible to maintain the prescribed transfer rate under certain load conditions. Therefore, a bit synchronization setting other than 7.5 tcyc should be used if the operating frequency exceeds 20 MHz.

ICCRB is an 8-bit readable/writable register that issues start/stop conditions, manipulates the SDA pin, monitors the SCL pin, and controls reset in I<sup>2</sup>C control.

Bit	Bit Name	Initial Value	R/W	Description
7	BBSY	0	R/W	<p>Bus Busy</p> <p>This bit enables to confirm whether the I<sup>2</sup>C bus is occupied or released and to issue start and stop conditions in master mode. This bit is set to 1 when the SDA level changes from high to low under the condition of SCL = high, assuming that the start condition has been issued. This bit is cleared to 0 when the SDA level changes from low to high under the condition of SCL = high, assuming that the stop condition has been issued. Write 1 to BBSY and 0 to SCP to issue a start condition. Follow this procedure when also re-transmitting a start condition. Write 0 to BBSY and 0 to SCP to issue a stop condition. To issue a start/stop condition, use the MOV instruction.</p>
6	SCP	1	W	<p>Start Condition/Stop Condition Prohibit</p> <p>The SCP bit controls the issue of start/stop conditions in master mode.</p> <p>To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored.</p>
5	SDAO	1	R/W	<p>Monitors the output level of SDA.</p> <p>0: When reading, SDA pin outputs low. 1: When reading, SDA pin outputs high.</p> <p>The write value should always be 1.</p>
4	—	1	R/W	<p>Reserved</p> <p>The write value should always be 1.</p>
3	SCLO	1	R	<p>This bit monitors SCL output level. When reading and SCLO is 1, SCL pin outputs high. When reading and SCLO is 0, SCL pin outputs low.</p>
2	—	1	—	<p>Reserved</p> <p>This bit is always read as 1.</p>
1	IICRST	0	R/W	<p>IIC control part reset</p> <p>This bit resets control parts except for I<sup>2</sup>C registers. If this bit is set to 1 when hang-up is occurred because of communication failure during I<sup>2</sup>C operation, I<sup>2</sup>C control part can be reset without setting ports and initializing registers.</p>

### 15.3.3 I<sup>2</sup>C Bus Mode Register (ICMR)

ICMR performs master mode wait control and selects the transfer bit count.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R/W	Reserved The write value should always be 0.
6	WAIT	0	R/W	Wait Insertion Bit This bit selects whether to insert a wait after data transfer except for the acknowledge bit. When WAIT is set to 1, after the fall of the clock for the final data bit, low period is extended for two transfer clocks. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted. The setting of this bit is invalid in slave mode.
5, 4	—	All 1	—	Reserved These bits are always read as 1.
3	BCWP	1	R/W	BC Write Protect This bit controls the BC2 to BC0 modifications. When modifying BC2 to BC0, this bit should be cleared to 0 and use the MOV instruction. 0: When writing, values of BC2 to BC0 are set. 1: When reading, 1 is always read. When writing, settings of BC2 to BC0 are invalid.



1	BC1	0	R/W	These bits specify the number of bits to be transferred next. When read, the remaining number of transfer bits is indicated. The data is transferred with one additional acknowledge bit. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL line is low. The value returns to 000 at the end of a data transfer, including the acknowledge bit. With the clock synchronous serial format, these bits should not be modified.
0	BC0	0	R/W	
				000: 9
				001: 2
				010: 3
				011: 4
				100: 5
				101: 6
				110: 7
				111: 8

### 15.3.4 I<sup>2</sup>C Bus Interrupt Enable Register (ICIER)

ICIER is an 8-bit readable/writable register that enables or disables interrupt sources and acknowledge bits, sets acknowledge bits to be transferred, and confirms acknowledge bits to be received.

Bit	Bit Name	Initial Value	R/W	Description
7	TIE	0	R/W	Transmit Interrupt Enable When the TDRE bit in ICSR is set to 1, this bit enables or disables the transmit data empty interrupt (TXI). 0: Transmit data empty interrupt request (TXI) is disabled. 1: Transmit data empty interrupt request (TXI) is enabled.
6	TEIE	0	R/W	Transmit End Interrupt Enable This bit enables or disables the transmit end interrupt (TEI) at the rising of the ninth clock while the TDRE bit in ICSR is 1. TEI can be canceled by clearing the TEND bit or the TEIE bit to 0. 0: Transmit end interrupt request (TEI) is disabled. 1: Transmit end interrupt request (TEI) is enabled.

This bit enables or disables the receive data full interrupt request (RXI) when a received data is transferred from ICDRS to ICDRR and the RDRF bit in ICSR is set to 1. RXI can be canceled by clearing the RDRF or RIE bit to 0.  
 0: Receive data full interrupt request (RXI) is disabled.  
 1: Receive data full interrupt request (RXI) is enabled.

4	NAKIE	0	R/W	<p>NACK receive interrupt enable</p> <p>This bit enables or disables the NACK receive interrupt request (NAKI) when the NACKF and AL bits in ICSR are set to 1. NAKI can be canceled by clearing the NACKF, AL, or NAKIE bit to 0.</p> <p>0: NACK receive interrupt request (NAKI) is disabled.          1: NACK receive interrupt request (NAKI) is enabled.</p>
3	STIE	0	R/W	<p>Stop condition detection interrupt enable</p> <p>0: Stop condition detection interrupt request (STPI) is disabled.          1: Stop condition detection interrupt request (STPI) is enabled.</p>
2	ACKE	0	R/W	<p>Acknowledge Bit Judgement Select</p> <p>0: The value of the acknowledge bit is ignored, and continuous transfer is performed.          1: If the acknowledge bit is 1, continuous transfer is interrupted.</p>
1	ACKBR	0	R	<p>Receive acknowledge</p> <p>In transmit mode, this bit stores the acknowledge data that are returned by the receive device. This bit cannot be modified.</p> <p>0: Receive acknowledge = 0          1: Receive acknowledge = 1</p>
0	ACKBT	0	R/W	<p>Transmit acknowledge</p> <p>In receive mode, this bit specifies the bit to be sent at the acknowledge timing.</p> <p>0: 0 is sent at the acknowledge timing.          1: 1 is sent at the acknowledge timing.</p>

ICSR is an 8-bit readable/writable register that performs confirmation of interrupt request flags and status.

Bit	Bit Name	Initial Value	R/W	Description
7	TDRE	0	R/W	Transmit Data Register Empty [Setting conditions] <ul style="list-style-type: none"> <li>• When data is transferred from ICDRT to ICDRS and ICDRT becomes empty</li> <li>• When TRS is set</li> <li>• When a transition from receive mode to transmit mode is made in slave mode</li> </ul> [Clearing conditions] <ul style="list-style-type: none"> <li>• When 0 is written in TDRE after reading TDRE = 1</li> <li>• When data is written to ICDRT</li> </ul>
6	TEND	0	R/W	Transmit end [Setting condition] When the ninth clock of SCL is rose while the TDRE flag is 1 [Clearing conditions] <ul style="list-style-type: none"> <li>• When 0 is written in TEND after reading TEND = 1</li> <li>• When data is written to ICDRT</li> </ul>
5	RDRF	0	R/W	Receive Data Register Full [Setting condition] When a received data is transferred from ICDRS to ICDRR [Clearing conditions] <ul style="list-style-type: none"> <li>• When 0 is written in RDRF after reading RDRF = 1</li> <li>• When ICDRR is read</li> </ul>

[Setting condition]

When no acknowledge is detected from the receive device in transmission while the ACKE bit in ICIER is 1

[Clearing condition]

When 0 is written in NACKF after reading NACKF = 1

Note: When NACKF = 1 is detected, NACKF must be cleared to 0. Subsequent transmission is not made until NACKF is cleared to 0.

---

3	STOP	0	R/W	Stop condition detection flag
---	------	---	-----	-------------------------------

[Setting conditions]

- In master mode, when a stop condition is detected after frame transfer
- In slave mode, when a stop condition is detected after the general call address or the first byte slave address, next to detection of start condition, accords with the address set in SAR

[Clearing condition]

When 0 is written in STOP after reading STOP = 1

---

2	AL	0	R/W	Arbitration Lost Flag
---	----	---	-----	-----------------------

This flag indicates that arbitration was lost in master mode. When two or more master devices attempt to seize the bus at nearly the same time, if the I<sup>2</sup>C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master.

[Setting conditions]

- If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode
- When the SDA pin outputs high in master mode while a start condition is detected

[Clearing condition]

When 0 is written in AL/OVE after reading AL/OVE=1

---

In slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR.

[Setting conditions]

- When the slave address is detected in slave receive mode
- When the general call address is detected in slave receive mode.

[Clearing condition]

When 0 is written in AAS after reading AAS=1

---

0	ADZ	0	R/W	General Call Address Recognition Flag This bit is valid in slave receive mode. [Setting condition] When the general call address is detected in slave receive mode [Clearing conditions] When 0 is written in ADZ after reading ADZ=1
---	-----	---	-----	--

---

### 15.3.6 Slave Address Register (SAR)

SAR is an 8-bit readable/writable register that sets slave address. When the chip is in slave mode, if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	SVA6 to SVA0	0	R/W	Slave Address 6 to 0 These bits set a unique address in bits SVA6 to SVA0, differing from the addresses of other slave devices connected to the I <sup>2</sup> C bus.
0	—	0	R/W	Reserved Though this bit can be read from or written to, the write value should always be 0.

---

ICDRT is an 8-bit readable/writable register that stores the transmit data. When ICDRT detects the space in the I<sup>2</sup>C bus shift register (ICDRS), it transfers the transmit data which is written in ICDRT to ICDRS and starts transferring data. If the next transfer data is written to ICDRT during transferring data of ICDRS, continuous transfer is possible. The initial value of ICDRT is H'FF.

### **15.3.8 I<sup>2</sup>C Bus Receive Data Register (ICDRR)**

ICDRR is an 8-bit register that stores the receive data. When data of one byte is received, ICDRR transfers the received data from ICDRS to ICDRR and the next data can be received. ICDRR is a receive-only register, therefore the CPU cannot be written to this register. The initial value of ICDRR is H'FF.

### **15.3.9 I<sup>2</sup>C Bus Shift Register (ICDRS)**

ICDRS is a register that is used to transfer/receive data. In transmission, data is transferred from ICDRT to ICDRS and the data is sent from the SDA pin. In reception, data is transferred from ICDRS to ICDRR after data of one byte is received. This register cannot be read from the CPU.

## 15.4.1 I<sup>2</sup>C Bus Format

Figure 15.3 shows the I<sup>2</sup>C bus formats. Figure 15.4 shows the I<sup>2</sup>C bus timing. The first frame following a start condition always consists of 8 bits.

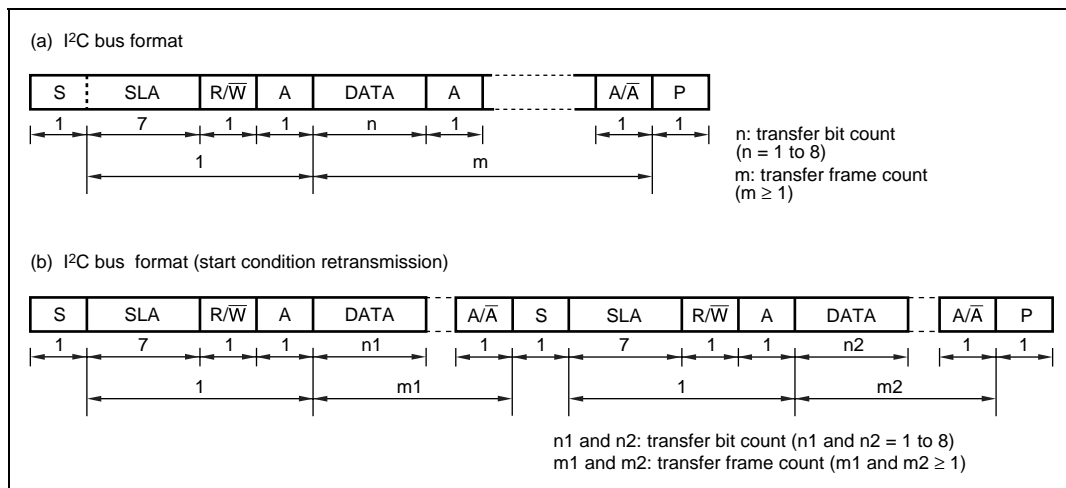


Figure 15.3 I<sup>2</sup>C Bus Formats

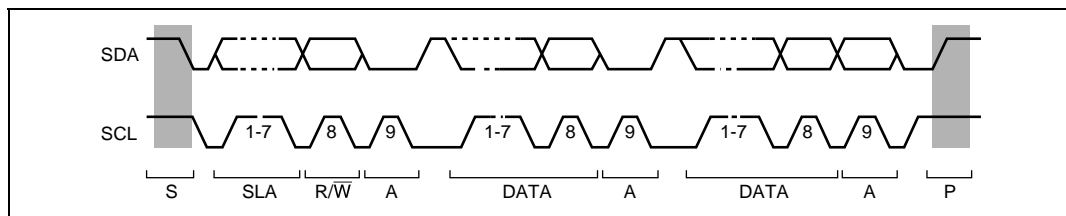


Figure 15.4 I<sup>2</sup>C Bus Timing

S: Start condition. The master device drives SDA from high to low while SCL is high.

SLA: Slave address

R/ $\overline{W}$ : Indicates the direction of data transfer: from the slave device to the master device when R/ $\overline{W}$  is 1, or from the master device to the slave device when R/ $\overline{W}$  is 0.

A: Acknowledge. The receiving device drives SDA to low.

DATA: Transferred data

P: Stop condition. The master device drives SDA from low to high while SCL is high.

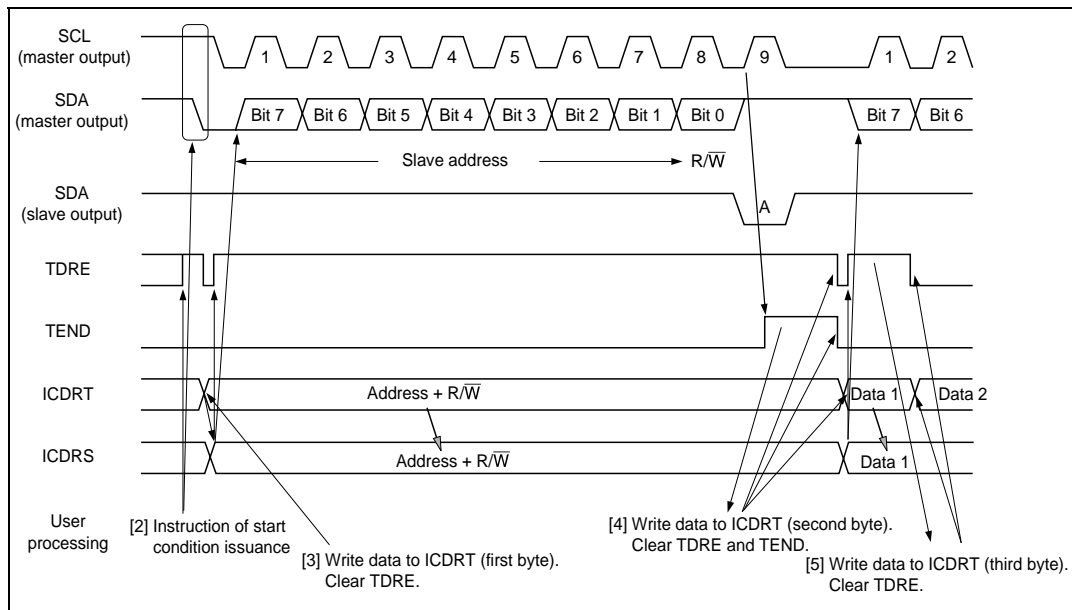
### 15.4.2 Master Transmit Operation

In I<sup>2</sup>C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For master transmit mode operation timing, refer to figures 15.5 and 15.6. The transmission procedure and operations in master transmit mode are described below.

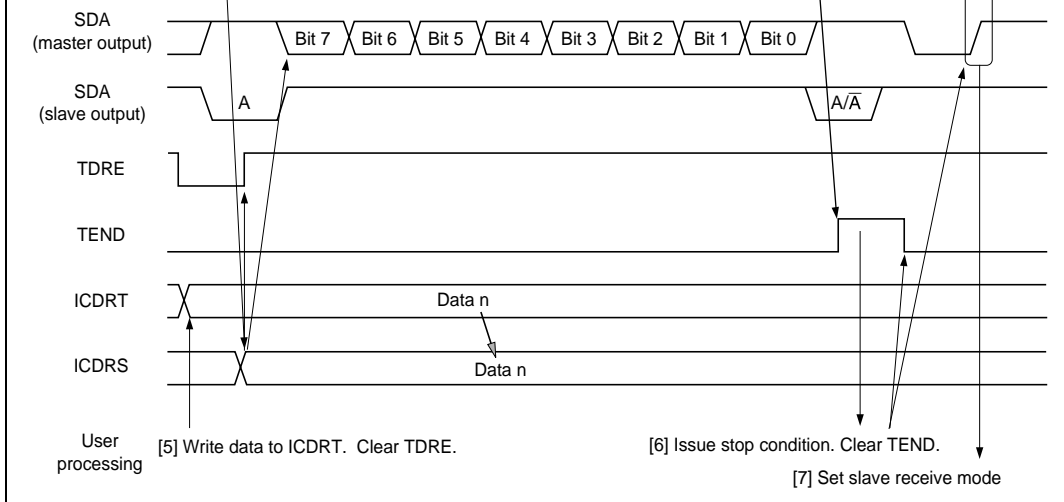
1. Set the ICE bit in ICCRA to 1. Set the WAIT bit in ICMR and the CKS3 to CKS0 bits in ICCR1 to 1. (Initial setting)
2. Read the BBSY flag in ICCRB to confirm that the bus is free. Set the MST and TRS bits in ICCRA to select master transmit mode. Then, write 1 to BBSY and 0 to SCP using MOV instruction. (Start condition issued) This generates the start condition.
3. After confirming that TDRE in ICSR has been set, write the transmit data (the first byte data show the slave address and R/ $\overline{W}$ ) to ICDRT. After this, when TDRE is cleared to 0, data is transferred from ICDRT to ICDRS. TDRE is set again.
4. When transmission of one byte data is completed while TDRE is 1, TEND in ICSR is set to 1 at the rise of the 9th transmit clock pulse. Read the ACKBR bit in ICIER, and confirm that the slave device has been selected. Then, write second byte data to ICDRT, and clear TDRE and TEND. When ACKBR is 1, the slave device has not been acknowledged, so issue the stop condition. To issue the stop condition, write 0 to BBSY and SCP using MOV instruction. SCL is fixed low until the transmit data is prepared or the stop condition is issued.
5. The transmit data after the second byte is written to ICDRT every time TDRE is set, thus clearing TDRE.
6. Write the number of bytes to be transmitted to ICDRT. Wait until TEND is set (the end of last byte data transmission) while TDRE is 1, or wait for NACK (NACKF in ICSR = 1) from the



7. When the STOP bit in ICSR is set to 1, the operation returns to the slave receive mode.



**Figure 15.5 Master Transmit Mode Operation Timing 1**



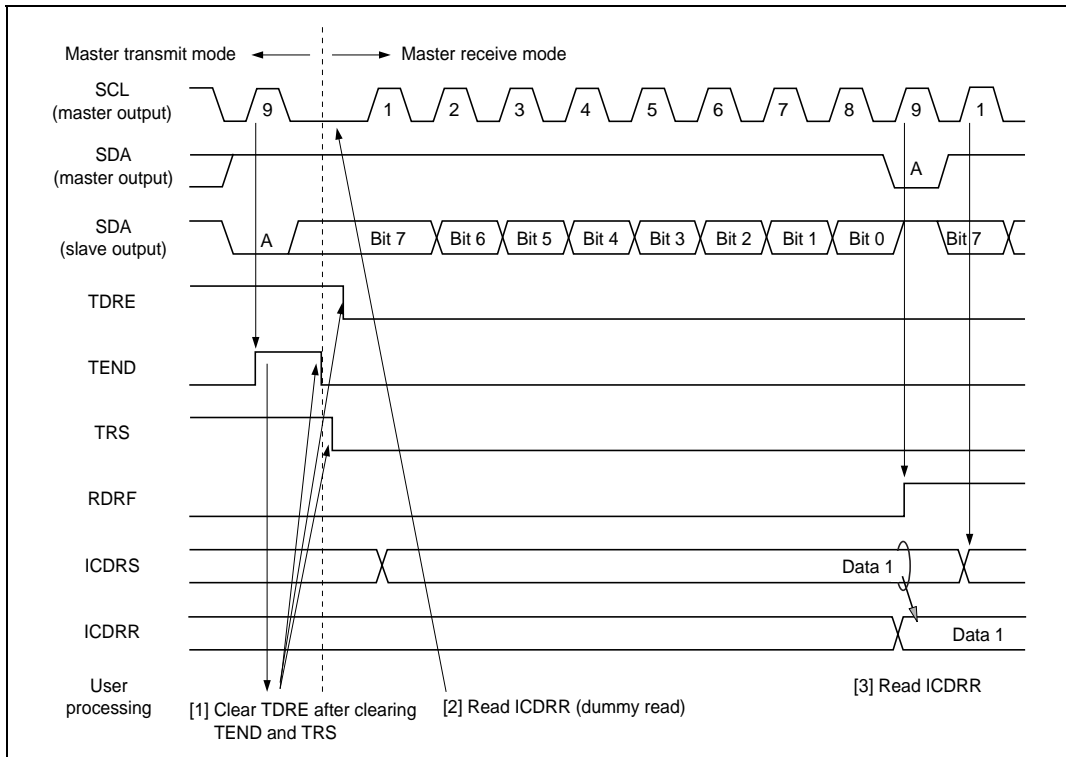
**Figure 15.6 Master Transmit Mode Operation Timing 2**

### 15.4.3 Master Receive Operation

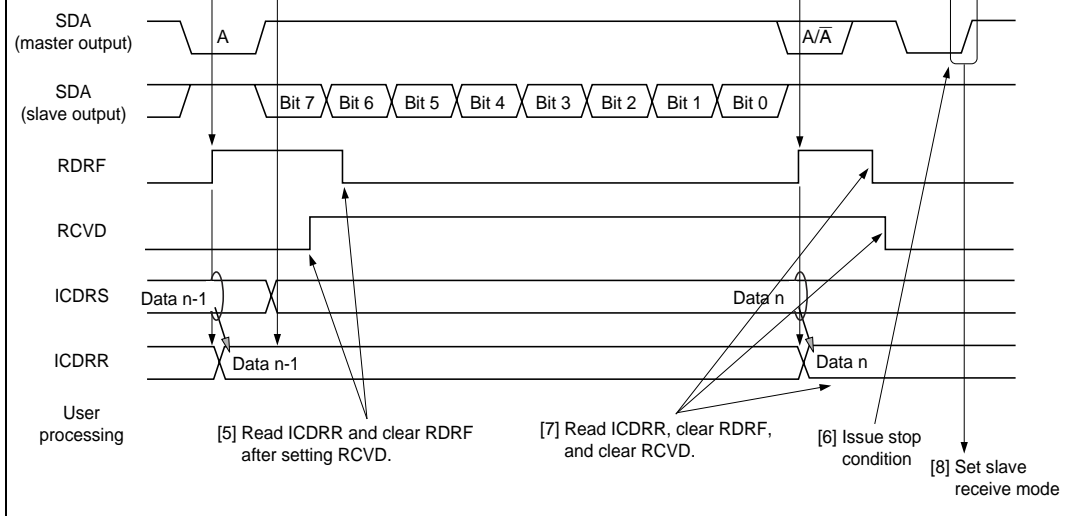
In master receive mode, the master device outputs the receive clock, receives data from the slave device, and returns an acknowledge signal. For master receive mode operation timing, refer to figures 15.7 and 15.8. The reception procedure and operations in master receive mode are shown below.

1. Clear the TEND bit in ICSR to 0, then clear the TRS bit in ICCRA to 0 to switch from master transmit mode to master receive mode. Then, clear the TDRE bit to 0.
2. When ICDRR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. The master device outputs the level specified by ACKBT in ICIER to SDA, at the 9th receive clock pulse.
3. After the reception of first frame data is completed, the RDRF bit in ICST is set to 1 at the rise of 9th receive clock pulse. At this time, the received data is read by reading ICDRR.
4. The continuous reception is performed by reading ICDRR and clearing RDRF to 0 every time RDRF is set. If 8th receive clock pulse falls after reading ICDRR by the other processing while RDRF is 1, SCL is fixed low until ICDRR is read.
5. If next frame is the last receive data, set the RCVD bit in ICCR1 to 1 before reading ICDRR. This enables the issuance of the stop condition after the next reception.
6. When the RDRF bit is set to 1 at rise of the 9th receive clock pulse, read ICDRR. Then, clear RCVD.

8. The operation returns to the slave receive mode.



**Figure 15.7 Master Receive Mode Operation Timing 1**



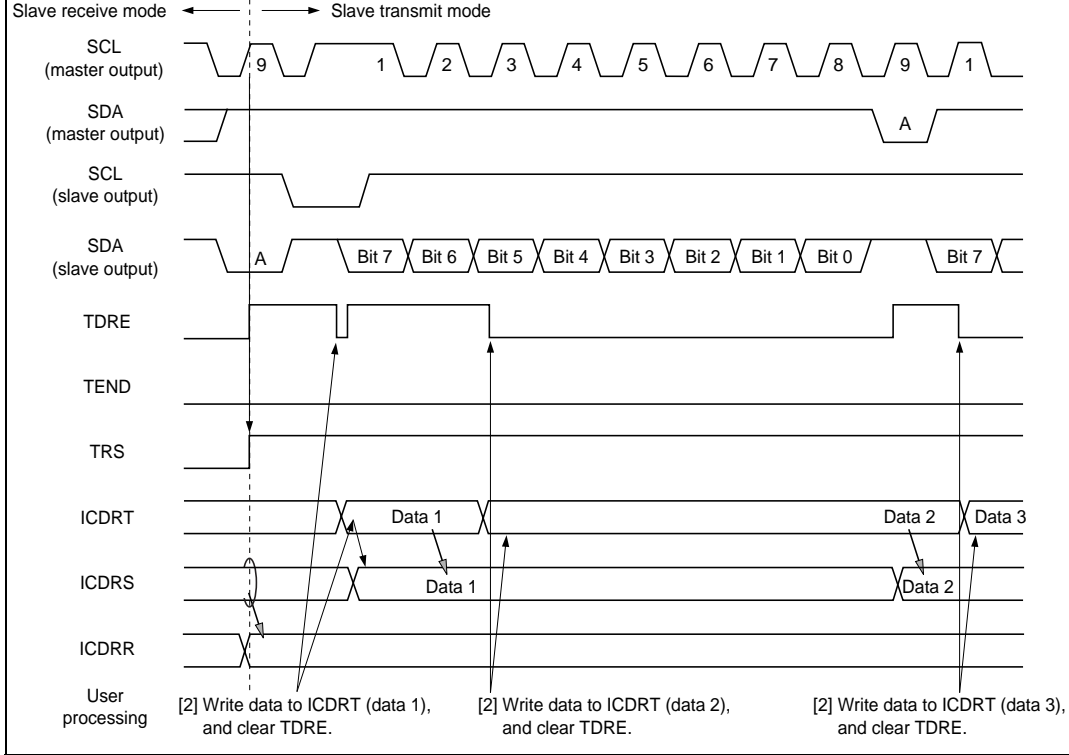
**Figure 15.8 Master Receive Mode Operation Timing 2**

### 15.4.4 Slave Transmit Operation

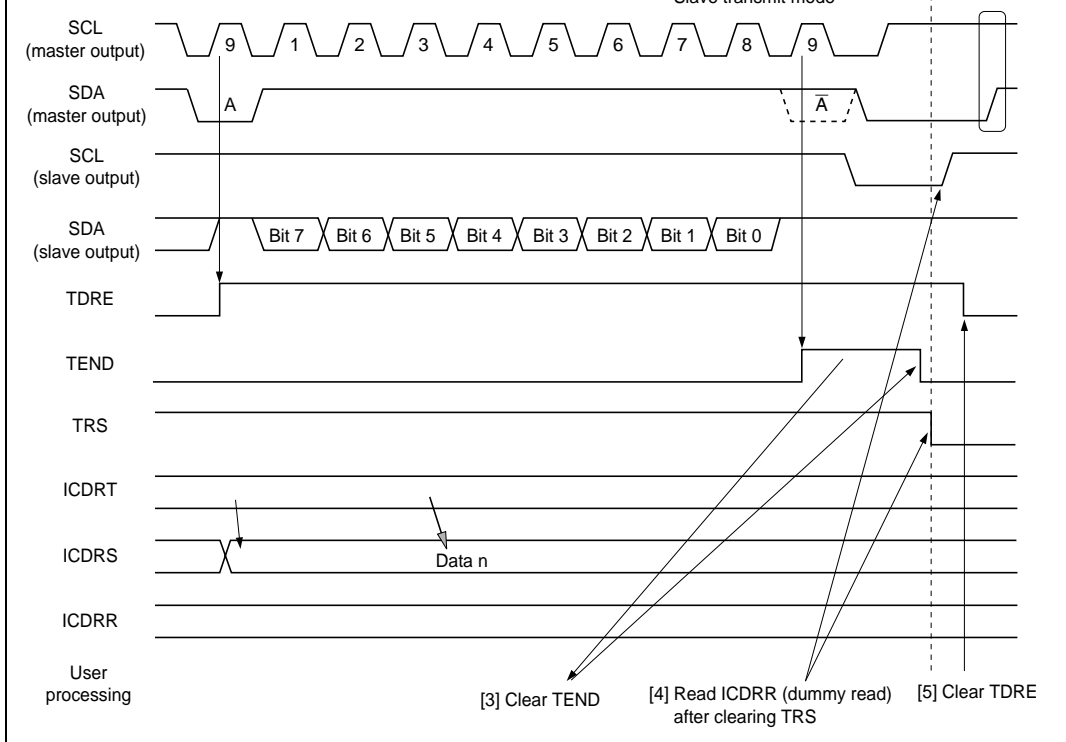
In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. For slave transmit mode operation timing, refer to figures 15.9 and 15.10.

The transmission procedure and operations in slave transmit mode are described below.

1. Set the ICE bit in ICCRA to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCRA to 1. (Initial setting) Set the MST and TRS bits in ICCRA to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At this time, if the 8th bit data ( $\overline{R/W}$ ) is 1, the TRS in ICCRA and TDRE in ICSR are set to 1, and the mode changes to slave transmit mode automatically. The continuous transmission is performed by clearing TDRE after writing transmit data to ICDRT every time TDRE is set.
3. If TDRE is set after writing last transmit data to ICDRT, wait until TEND in ICSR is set to 1, with TDRE = 1. When TEND is set, clear TEND.
4. Clear TRS for the end processing, and read ICDRR (dummy read). SCL is free.
5. Clear TDRE.



**Figure 15.9 Slave Transmit Mode Operation Timing 1**



**Figure 15.10 Slave Transmit Mode Operation Timing 2**

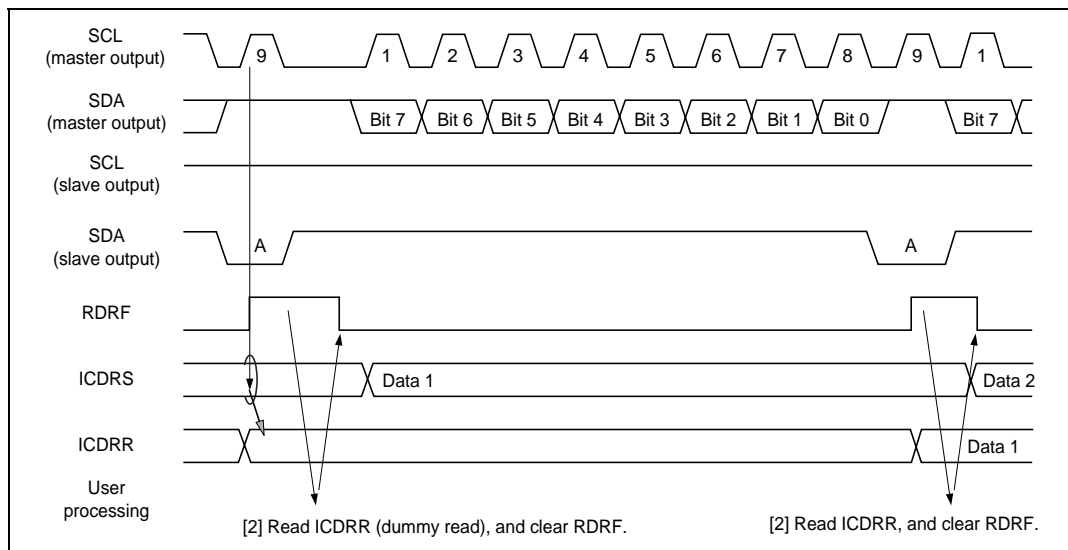
### 15.4.5 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. For slave receive mode operation timing, refer to figures 15.11 and 15.12. The reception procedure and operations in slave receive mode are described below.

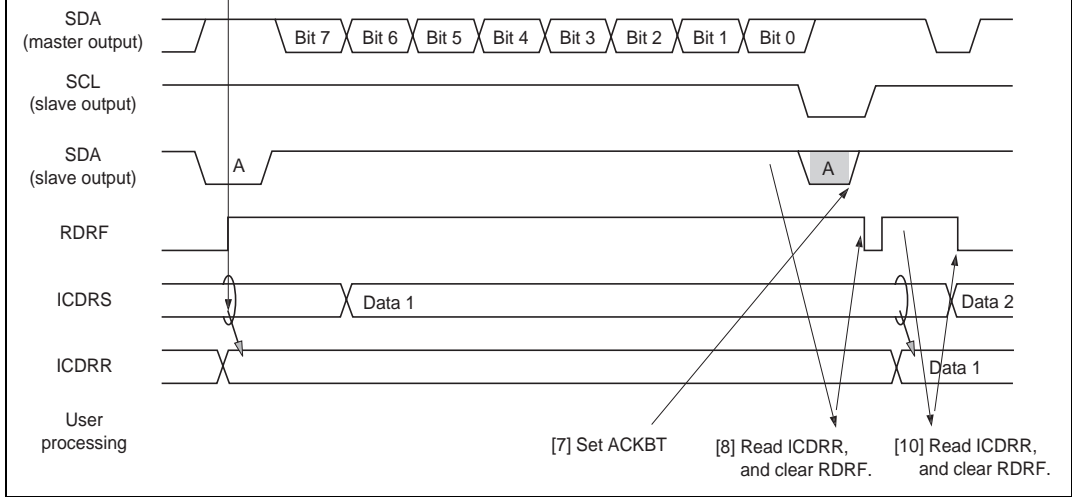
1. Set the ICE bit in ICCRA to 1. Set the MLS and WAIT bits in ICMR and the CKS3 to CKS0 bits in ICCRA to 1. (Initial setting) Set the MST and TRS bits in ICCRA to select slave receive mode, and wait until the slave address matches.
2. When the slave address matches in the first frame following detection of the start condition, the slave device outputs the level specified by ACKBT in ICIER to SDA, at the rise of the 9th clock pulse. At the same time, RDRF in ICSR is set to read ICDRR (dummy read) and RDRF is cleared. (Since the read data show the slave address and R/W, it is not used.)

before reading ICDRR, to be returned to the master device, is reflected to the next transmit frame.

- The last byte data is read by reading ICDRR.



**Figure 15.11 Slave Receive Mode Operation Timing 1**

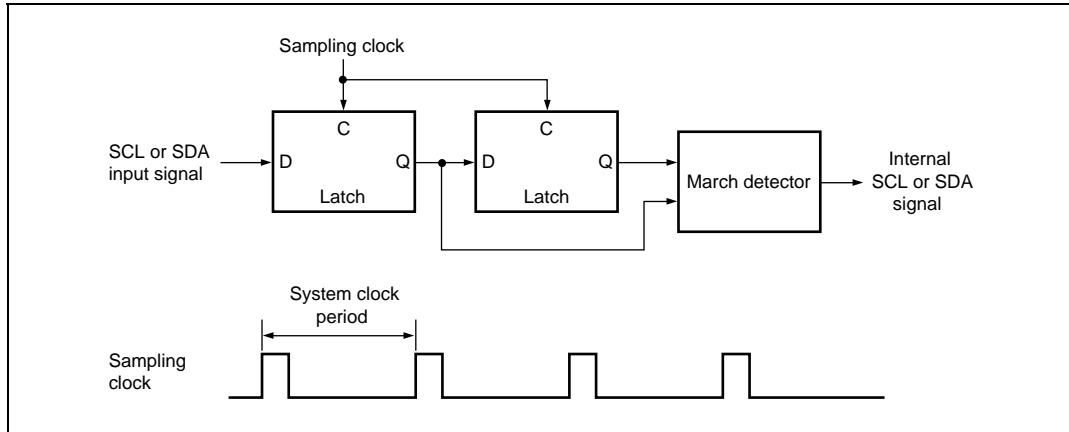


**Figure 15.12 Slave Receive Mode Operation Timing 2**



The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 15.13 shows a block diagram of the noise canceler circuit.

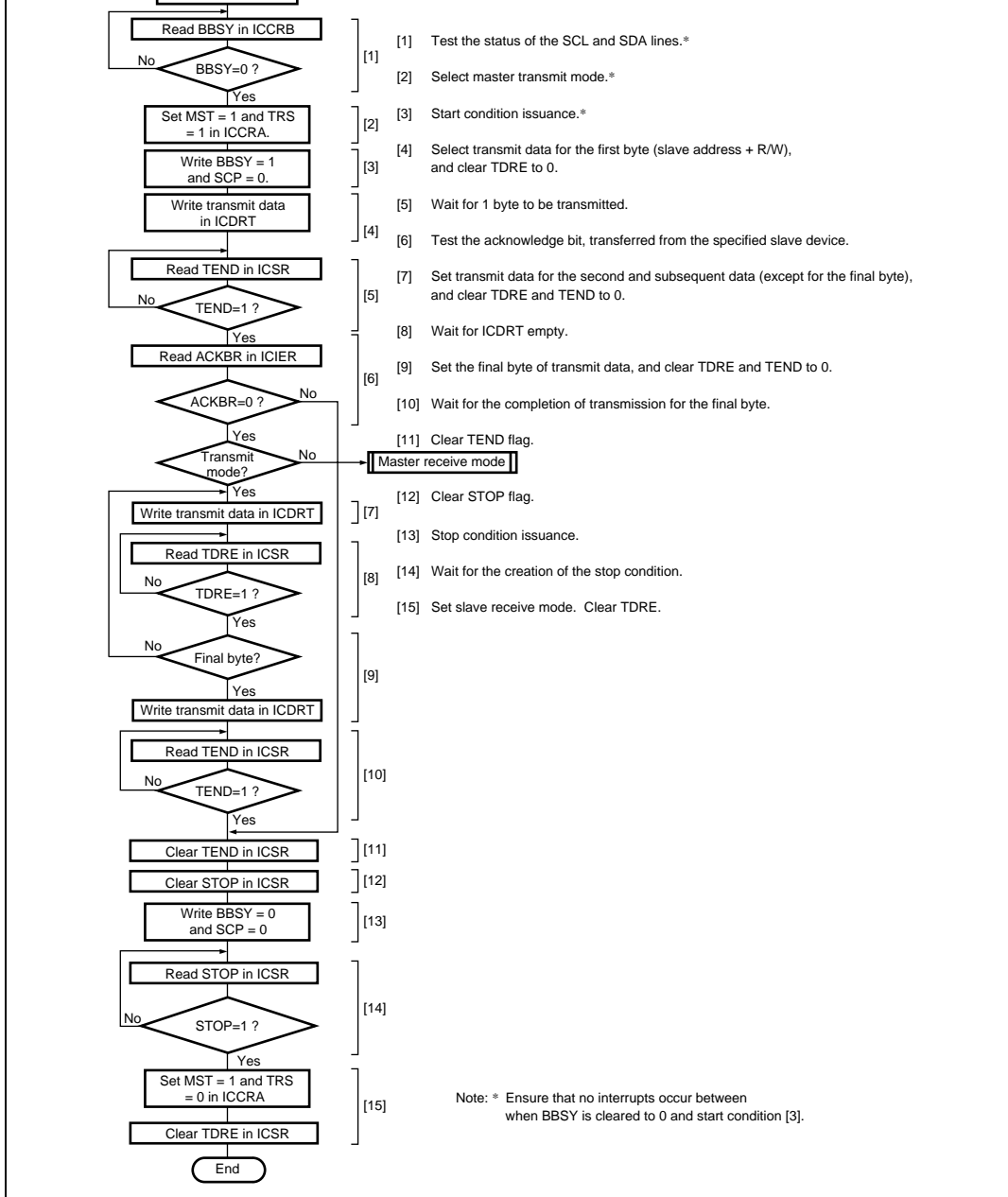
The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.



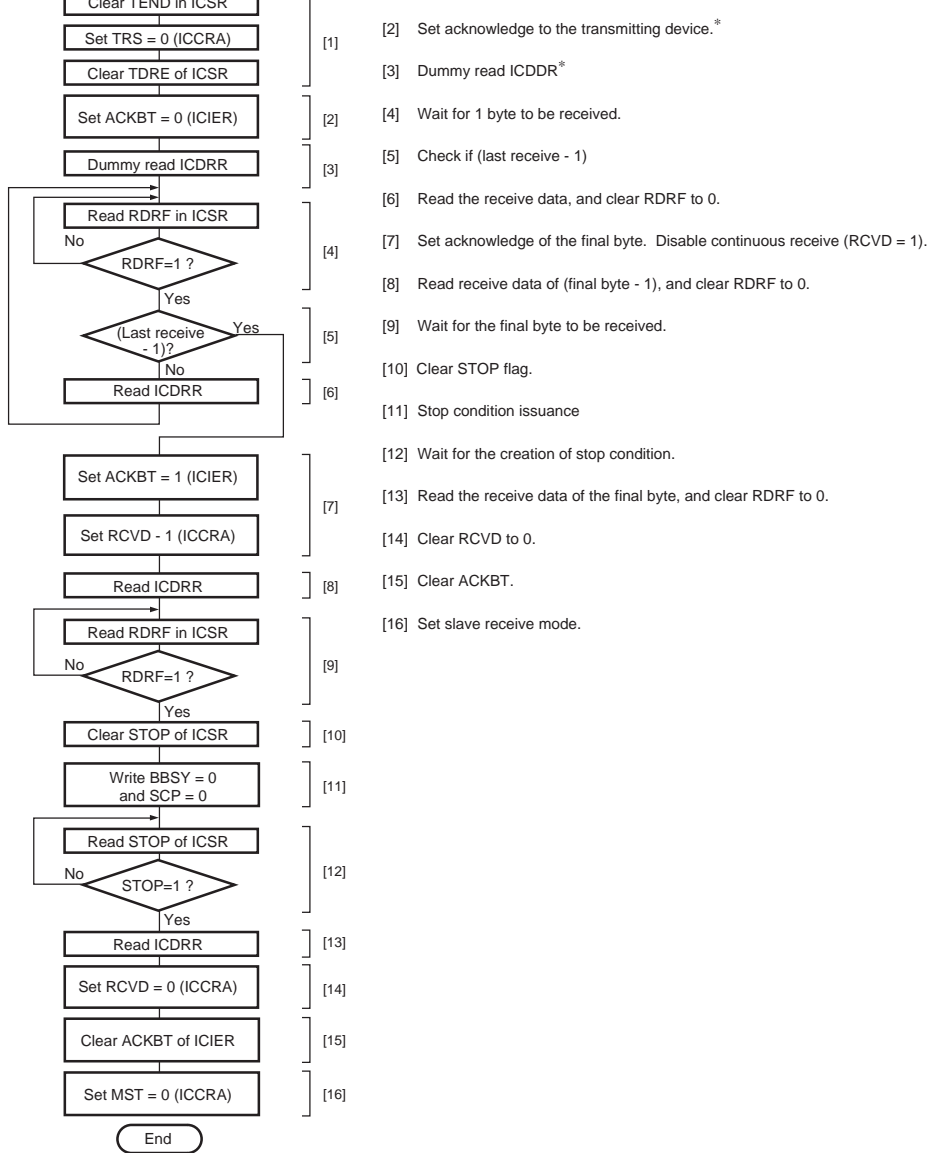
**Figure 15.13 Block Diagram of Noise Canceler**

### 15.4.7 Example of Use

Flowcharts in respective modes that use the I<sup>2</sup>C bus interface are shown in figures 15.14 to 15.17.



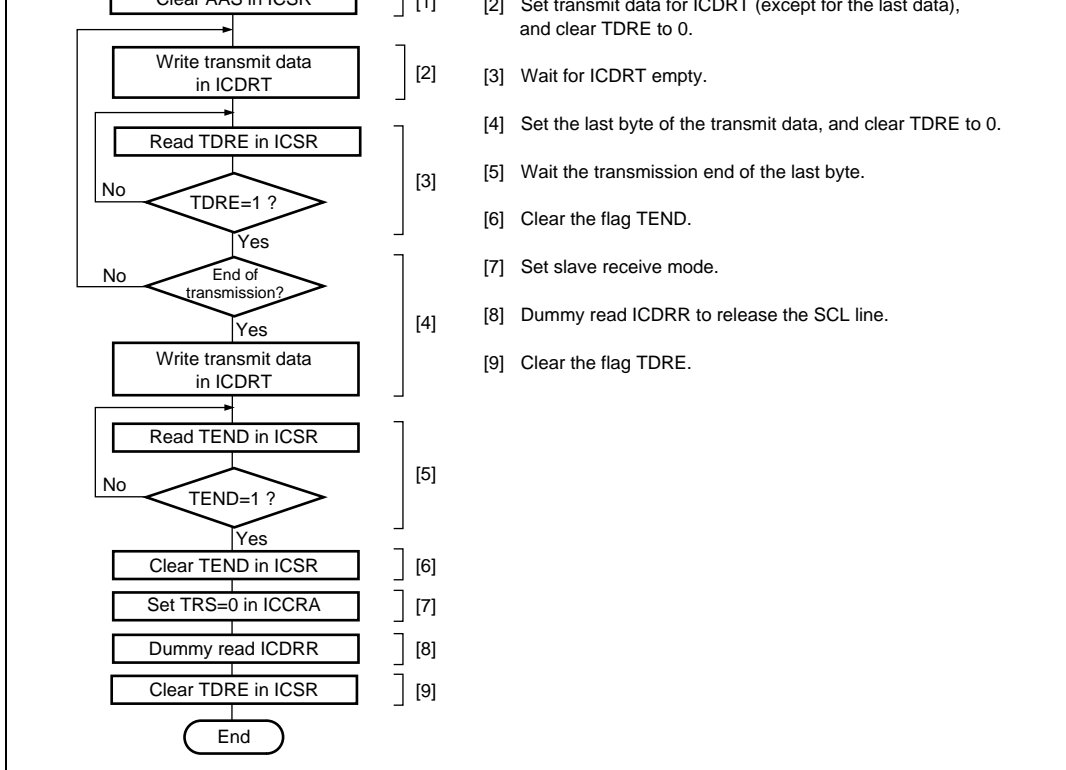
**Figure 15.14 Sample Flowchart for Master Transmit Mode**



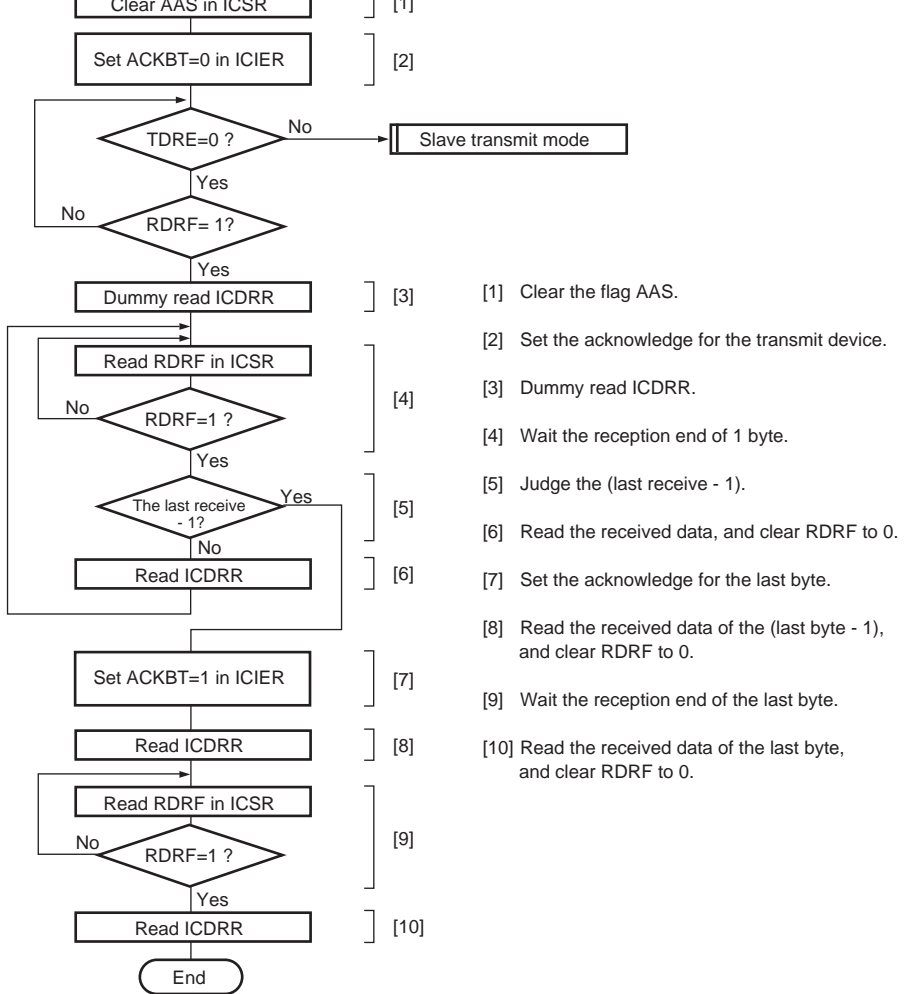
Note: \* Prevent any interrupts while steps [1] to [3] are executed.

Additional information: When receiving one-byte data, execute step [1], and then step [7] omitting steps [2] to [6].  
 In step [8], dummy read ICDRR.

**Figure 15.15 Sample Flowchart for Master Receive Mode**



**Figure 15.16 Sample Flowchart for Slave Transmit Mode**



**Figure 15.17 Sample Flowchart for Slave Receive Mode**

There are six interrupt requests in this module; transmit data empty, transmit end, receive data full, NACK receive, STOP recognition, and arbitration lost. Table 15.3 shows the contents of each interrupt request.

**Table 15.3 Interrupt Requests**

<b>Interrupt Request</b>	<b>Abbreviation</b>	<b>Interrupt Condition</b>
Transmit Data Empty	TXI	$(TDRE = 1) \cdot (TIE = 1)$
Transmit End	TEI	$(TEND = 1) \cdot (TEIE = 1)$
Receive Data Full	RXI	$(RDRF = 1) \cdot (RIE = 1)$
STOP Recognition	STPI	$(STOP = 1) \cdot (STIE = 1)$
NACK Detection	NAKI	$\{(NACKF = 1) + (AL = 1)\} \cdot (NAKIE = 1)$
Arbitration Lost		

When interrupt conditions described in table 15.3 are 1 and the CPU is ready to receive interrupts, an interrupt execution handling is executed. Clear each interrupt source during an interrupt execution handling. Note that TDRE and TEND are automatically cleared by writing the transmit data to ICDRT, and RDRF is automatically cleared by reading ICDRR. When the transmit data is written to ICDRT, TDRE is set again simultaneously. When TDRE is cleared, extra one byte of data may be transmitted.

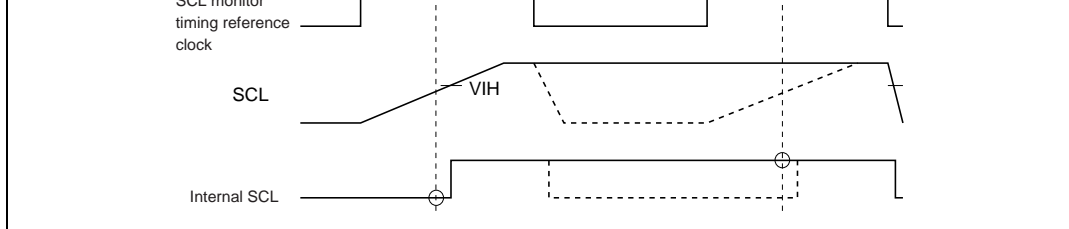
## 15.6 Bit Synchronous Circuit

In master mode,

- When SCL is driven to low by the slave device
- When the rising speed of SCL is lower by the load of the SCL line (load capacitance or pull-up resistance)

This module has a possibility that high level period may be short in the two states described above. Therefore it monitors SCL and communicates by bit with synchronization.

Figure 15.18 shows the timing of the bit synchronous circuit and table 15.4 shows the time when SCL output changes from low to Hi-Z then SCL is monitored.



**Figure 15.18 Timing of the Bit Synchronous Circuit**

**Table 15.4 Time for monitoring SCL**

CKS3	CKS2	Time for monitoring SCL
0	0	7.5 t <sub>cyc</sub> *
	1	19.5 t <sub>cyc</sub>
1	0	17.5 t <sub>cyc</sub>
	1	41.5 t <sub>cyc</sub>

Note: \* If the operating frequency exceeds 20 MHz, it may not be possible to maintain the prescribed transfer rate under certain load conditions. A setting other than 7.5 t<sub>cyc</sub> should therefore be used.

## 15.7 Usage Notes

- Issue (retransmit) the start/stop conditions after the fall of the ninth clock is confirmed. Check SCLO in the I<sup>2</sup>C control register B (ICCRB)\* to confirm the fall of the ninth clock. When the start/stop conditions are issued (retransmitted) at the specific timing under the following condition (i) or (ii), such conditions may not be output successfully. This does not occur in other cases.
  - When the rising of SCL falls behind the time specified in section 15.6, Bit Synchronous Circuit, by the load of the SCL bus (load capacitance or pull-up resistance)
  - When the bit synchronous circuit is activated by extending the low period of eighth and ninth clocks, that is driven by the slave device
- Control WAIT in the I<sup>2</sup>C bus mode register (ICMR) to be set to 0. When WAIT is set to 1, and SCL is driven low for two or more transfer clocks by the slave device at the eighth and ninth clocks, the high period of ninth clock may be shortened. This does not occur in other cases.

edge of the eighth clock cycle. However, when ICDRR is read near the falling edge of the eighth clock cycle, SCL is only fixed low for one clock cycle at the eighth clock cycle of the next receive data, after which SCL is no longer fixed and the ninth clock cycle is output, even if ICDRR is not read. This causes the receive data to overflow.

The following methods can be used to prevent this from occurring.

- In master receive mode, complete processing to read ICDRR before the rising edge of the eighth clock cycle.
- In master receive mode, set RCVD to 1 and perform communication processing one byte at a time.

4. Limitations on transfer rate setting values when using I<sup>2</sup>C bus interface 2 (IIC2) in multi-master mode

When operating in multi-master mode and the IIC transfer rate setting of the MCU is slower than that of another master device, an SCL of an unanticipated width may be output occasionally. To prevent this, set the transfer rate to a value 1/1.8 or greater than the fastest transfer rate among the other master devices. For example, if the fastest transfer rate setting among the other master devices is 400 kbps, set the IIC transfer rate of the MCU to 223 kbps (400/1.8) or higher.

5. Limitations on use of bit manipulation instructions to set MST and TRS when using I<sup>2</sup>C bus interface 2 (IIC2) in multi-master mode

When bit manipulation instructions are used to set MST and TRS in succession to specify master transmit while operating in multi-master mode, an arbitration lost may occur, during execution of the bit manipulation instruction to set TRS, with timing that results in a contradictory state in which AL in ICSR is set to 1 and master transmit mode (MST = 1, TRS = 1) is selected as well.

The following methods can be used to prevent this from occurring.

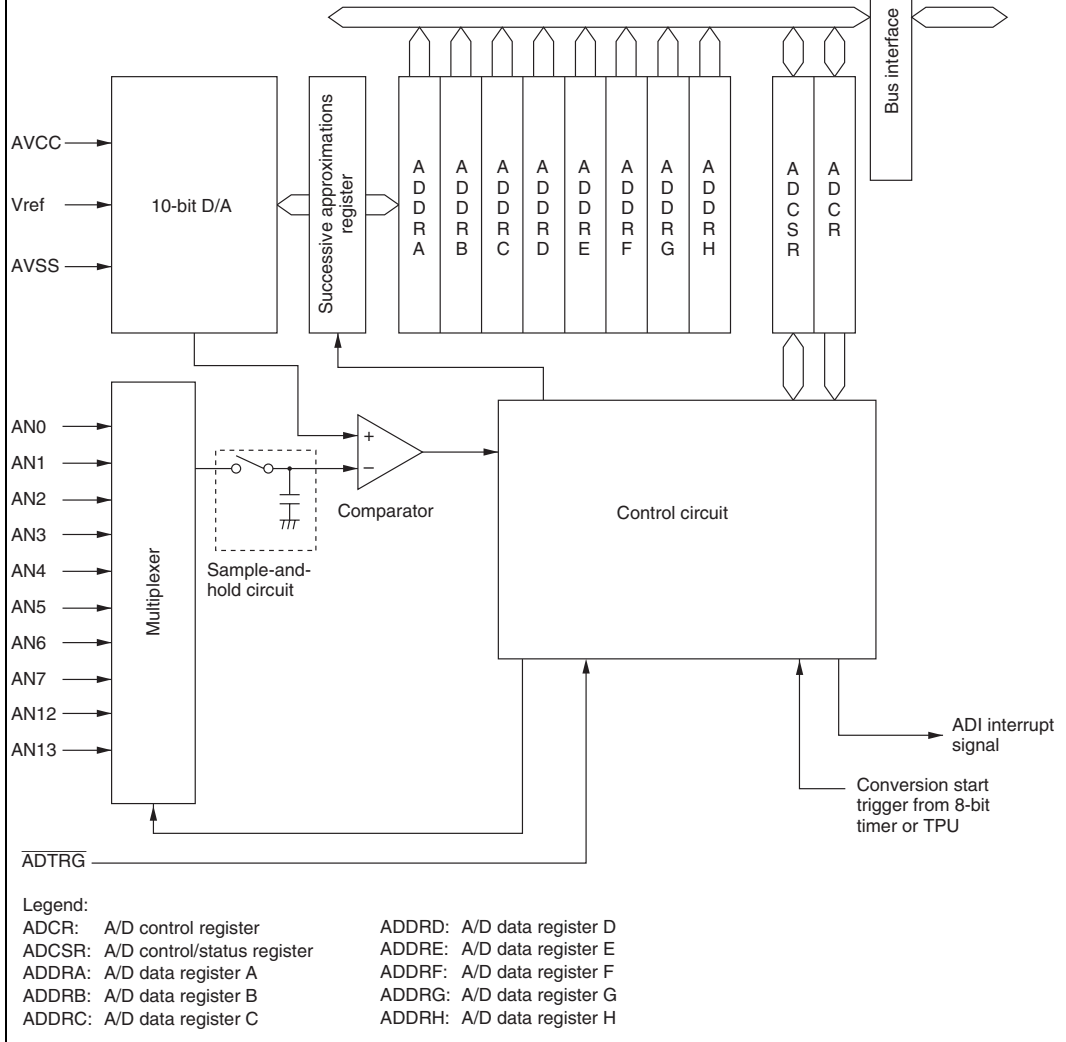
- When operating in multi-master mode, always use the MOV instruction to set MST and TRS.
- When an arbitration lost occurs, confirm that MST and TRS are both cleared to 0. If the settings are other than MST = 0, TRS = 0, clear MST and TRS to 0.



This LSI includes a successive approximation type 10-bit A/D converter that allows up to ten analog input channels to be selected. The block diagram of A/D converter is shown in figure 16.1.

## 16.1 Features

- 10-bit resolution
- Ten input channels
- Conversion time: 7.6  $\mu$ s per channel (at 34-MHz operation)
- Two kinds of operating modes
  - Single mode: Single-channel A/D conversion
  - Scan mode: Continuous A/D conversion on 1 to 4 channels or 1 to 8 channels
- Eight data registers
  - Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  - Conversion can be started by software, 16-bit timer pulse unit (TPU), conversion start trigger by 8-bit timer (TMR), or external trigger signal.
- Interrupt request
  - A/D conversion end interrupt (ADI) request can be generated
- Module stop mode can be set



**Figure 16.1 Block Diagram of A/D Converter**

Table 16.1 summarizes the input pins used by the A/D converter.

The AV<sub>CC</sub> and AV<sub>SS</sub> pins are the power supply pins for the analog block in the A/D converter. The V<sub>ref</sub> pin is the A/D conversion reference voltage pin.

The sixteen analog input pins are divided into two channel sets: channel set 0 (AN0 to AN7) and channel set 1 (AN12 and AN13).

**Table 16.1 Pin Configuration**

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV <sub>CC</sub>	Input	Analog block power supply
Analog ground pin	AV <sub>SS</sub>	Input	Analog block ground
Reference voltage pin	V <sub>ref</sub>	Input	A/D conversion reference voltage
Analog input pin 0	AN0	Input	Channel set 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 12	AN12	Input	Channel set 1 analog inputs
Analog input pin 13	AN13	Input	
A/D external trigger input pin	ADTRG	Input	External trigger input for starting A/D conversion

The A/D converter has the following registers.

- A/D data register A (ADDRA)
- A/D data register B (ADDRB)
- A/D data register C (ADDRC)
- A/D data register D (ADDRD)
- A/D data register E (ADDRE)
- A/D data register F (ADDRF)
- A/D data register G (ADDRG)
- A/D data register H (ADDRH)
- A/D control/status register (ADCSR)
- A/D control register (ADCR)

### **16.3.1 A/D Data Registers A to H (ADDRA to ADDRH)**

There are eight 16-bit read-only ADDR registers, ADDRA to ADDRH, used to store the results of A/D conversion. The ADDR registers, which store a conversion result for each channel, are shown in table 16.2.

The converted 10-bit data is stored to bits 15 to 6. The lower 6-bit data is always read as 0.

The data bus between the CPU and the A/D converter is 16-bit width. The data can be read directly from the CPU.

Analog Input Channel		A/D Data Register which Stores Conversion Result
Channel Set 0 (CH3 = 0)	Channel Set 1 (CH3 = 1)	
AN0	—	ADDRA
AN1	—	ADDRB
AN2	—	ADDRC
AN3	—	ADDRD
AN4	AN12	ADDRE
AN5	AN13	ADDRF
AN6	—	ADDRG
AN7	—	ADDRH

Bit	Bit Name	Initial Value	R/W	Description
7	ADF	0	R/(W)*	<p>A/D End Flag</p> <p>A status flag that indicates the end of A/D conversion.</p> <p>[Setting conditions]</p> <ul style="list-style-type: none"> <li>When A/D conversion ends in single mode</li> <li>When A/D conversion ends on all specified channels in scan mode</li> </ul> <p>[Clearing conditions]</p> <ul style="list-style-type: none"> <li>When 0 is written after reading ADF = 1</li> <li>When the DTC or DMAC is activated by an ADI interrupt and ADDR is read</li> </ul>
6	ADIE	0	R/W	<p>A/D Interrupt Enable</p> <p>A/D conversion end interrupt (ADI) request enabled when 1 is set</p>
5	ADST	0	R/W	<p>A/D Start</p> <p>Clearing this bit to 0 stops A/D conversion, and the A/D converter enters wait state. When this bit is set to 1 by software, TPU (trigger), TMR (trigger), or the <math>\overline{\text{ADTRG}}</math> pin, A/D conversion starts. This bit remains set to 1 during A/D conversion. In single mode, cleared to 0 automatically when conversion on the specified channel ends. In scan mode, conversion continues sequentially on the specified channels until this bit is cleared to 0 by a reset, or a transition to hardware standby mode or software standby mode.</p>
4	—	0	—	<p>Reserved</p> <p>This bit is always read as 0 and cannot be modified.</p>

2	CH2	0	R/W	Selects analog input together with bits SCANE and SCANS in ADCR.
1	CH1	0	R/W	
0	CH0	0	R/W	Set the input channel when conversion is stopped (ADST = 0).
When SCANE = 0 and SCANS = X				
0000:	AN0		10xx:	Cannot be set
0001:	AN1		10xx:	Cannot be set
0010:	AN2		10xx:	Cannot be set
0011:	AN3		10xx:	Cannot be set
0100:	AN4		1100:	AN12
0101:	AN5		1101:	AN13
0110:	AN6		111x:	Cannot be set
0111:	AN7		111x:	Cannot be set
When SCANE = 1 and SCANS = 0				
0000:	AN0		1000:	AN8
0001:	AN0 and AN1		1001:	AN8 and AN9
0010:	AN0 to AN2		1010:	AN8 to AN10
0011:	AN0 to AN3		1011:	AN8 to AN11
0100:	AN4		1100:	AN12
0101:	AN4 and AN5		1101:	AN12 and AN13
0110:	AN4 to AN6		1110:	AN12 to AN14
0111:	AN4 to AN7		1111:	AN12 to AN15
When SCANE = 1 and SCANS = 1				
0000:	AN0		1xxx:	Cannot be set
0001:	AN0 and AN1		1xxx:	Cannot be set
0010:	AN0 to AN2		1xxx:	Cannot be set
0011:	AN0 to AN3		1xxx:	Cannot be set
0100:	AN0 to AN4		1xxx:	Cannot be set
0101:	AN0 to AN5		1xxx:	Cannot be set
0110:	AN0 to AN6		1xxx:	Cannot be set
0111:	AN0 to AN7		1xxx:	Cannot be set

Legend: x: Don't care.

Note: \* Only 0 can be written in bit 7, to clear the flag.

ADCR enables A/D conversion start by an external trigger input.

It also sets the A/D converter operating mode and the A/D conversion time.

Bit	Bit Name	Initial Value	R/W	Description
7	TRGS1	0	R/W	Timer Trigger Select 1 and 0
6	TRGS0	0	R/W	These bits select enabling or disabling of the start of A/D conversion by a trigger signal. 00: A/D conversion start by external trigger is disabled 01: A/D conversion start by external trigger (TPU) is enabled 10: A/D conversion start by external trigger (TMR) is enabled 11: A/D conversion start by external trigger pin (ADTRG) is enabled
5	SCANE	0	R/W	Scan Mode
4	SCANS	0	R/W	Selects single mode or scan mode as the A/D conversion operating mode. 0x: Single mode 10: Scan mode. A/D conversion is performed continuously for channels 1 to 4 11: Scan mode. A/D conversion is performed continuously for channels 1 to 8.
3	CKS1	0	R/W	Clock Select 1 to 0
2	CKS0	0	R/W	Sets the A/D conversion time. Only set bits CKS1 and CKS0 while conversion is stopped (ADST = 0). 00: A/D conversion time = 530 states (max) 01: A/D conversion time = 266 states (max) 10: A/D conversion time = 134 states (max) 11: A/D conversion time = 68 states (max)
1, 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

Legend: x: Don't care.



The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes: single mode and scan mode. When changing the operating mode or analog input channel, to prevent incorrect operation, first clear the bit ADST to 0 in ADCSR to halt A/D conversion. The ADST bit can be set at the same time as the operating mode or analog input channel is changed.

### 16.4.1 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. Operations are as follows.

1. A/D conversion is started when the ADST bit in ADCSR is set to 1, according to the software or external trigger input.
2. When A/D conversion is completed, the result is transferred to the corresponding A/D data register to the channel.
3. On completion of conversion, the ADF bit in ADCSR is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated.
4. The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends. When the ADST bit is cleared to 0 during A/D conversion, A/D conversion stops and the A/D converter enters wait state.

### 16.4.2 Scan Mode

In scan mode, A/D conversion is to be performed sequentially on the specified channels: maximum four channels or maximum eight channels. Operations are as follows.

1. When the ADST bit in ADCSR is set to 1 by software, TPU or external trigger input, A/D conversion starts on the first channel in the group.  
The consecutive A/D conversion on maximum four channels (SCANE and SCANS = 10) or on maximum eight channels (SCANE and SCANS = 11) can be selected. When the consecutive A/D conversion is performed on the four channels, the A/D conversion starts on AN0 when CH3 = 1 and CH2 = 1, AN4 when CH3 and CH2 = 01, or AN12 when CH3 and CH2 = 11. When the consecutive A/D conversion is performed on the eight channels, the A/D conversion starts on AN0 when CH3 = 0.
2. When A/D conversion for each channel is completed, the result is sequentially transferred to the corresponding A/D data register to each channel.

channel in the group starts again.

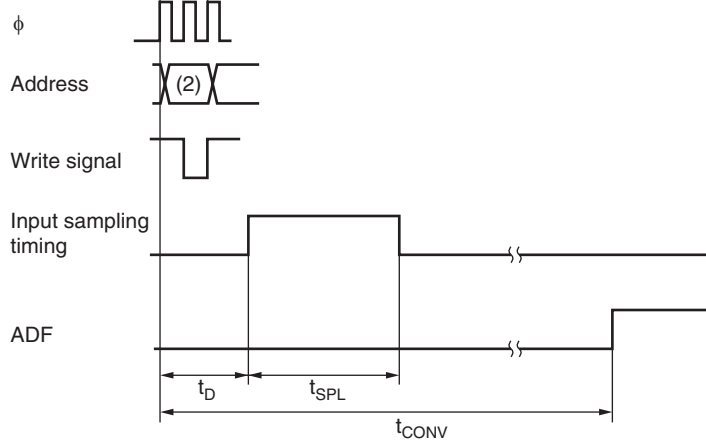
4. The ADST bit is not cleared automatically, and steps [2] and [3] are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters wait state. If the ADST bit is later set to 1, A/D conversion starts again from the first channel in the group.

### 16.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input when A/D conversion start delay time ( $t_D$ ) passes after the ADST bit is set to 1, then starts conversion. Figure 16.2 shows the A/D conversion timing. Table 16.3 indicates the A/D conversion time.

As indicated in figure 16.2, the A/D conversion time ( $t_{CONV}$ ) includes  $t_D$  and the input sampling time ( $t_{SPL}$ ). The length of  $t_D$  varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in tables 16.3.

In scan mode, the values given in table 16.3 apply to the first conversion time. The values given in table 16.4 apply to the second and subsequent conversions.



Legend:

(1): ADCSR write cycle

(2): ADCSR address

$t_D$ : A/D conversion start delay time

$t_{SPL}$ : Input sampling time

$t_{CONV}$ : A/D conversion time

**Figure 16.2 A/D Conversion Timing**

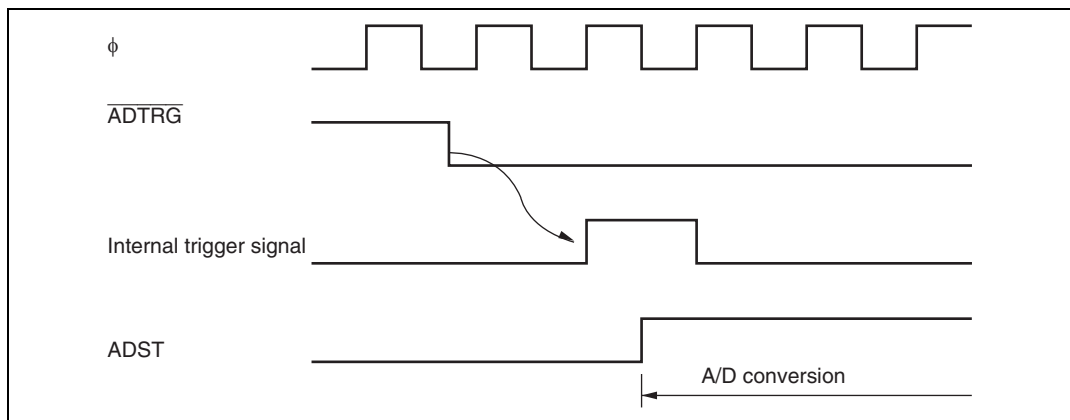
Item	Symbol	CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max
A/D conversion start delay time	$t_D$	18	—	33	10	—	16	6	—	9	4	—	5
Input sampling time	$t_{SPL}$	—	127	—	—	63	—	—	31	—	—	15	—
A/D conversion time	$t_{CONV}$	515	—	530	259	—	266	131	—	134	67	—	68

Note: Values in the table are the number of states.

**Table 16.4 A/D Conversion Time (Scan Mode)**

CKS1	CKS0	Conversion Time (State)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to 11 in ADCR, external trigger input is enabled at the  $\overline{\text{ADTRG}}$  pin. A falling edge at the  $\overline{\text{ADTRG}}$  pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the bit ADST has been set to 1 by software. Figure 16.3 shows the timing.



**Figure 16.3 External Trigger Input Timing**

## 16.5 Interrupts

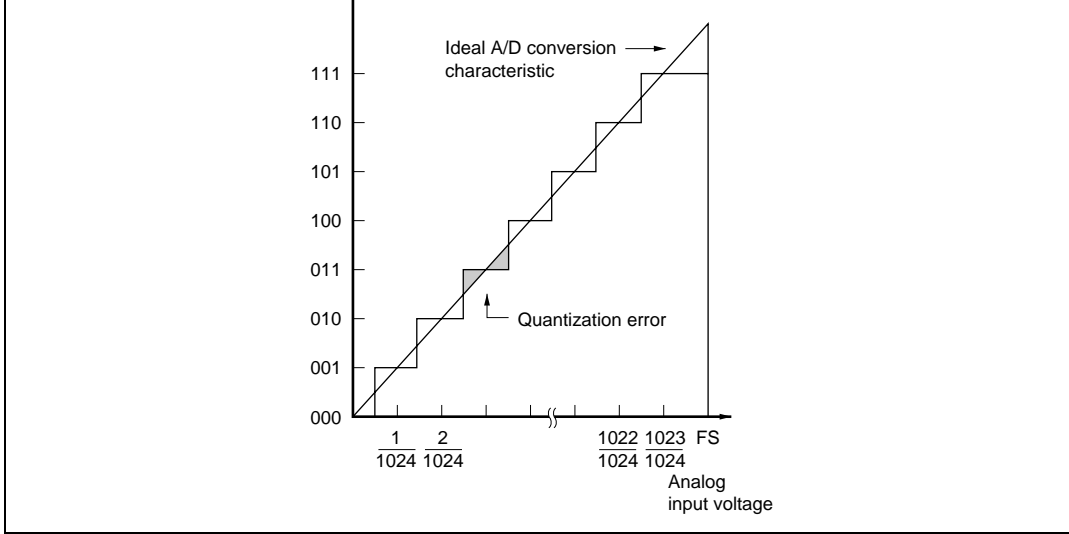
The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. Setting the ADIE bit to 1 enables an ADI interrupt requests while the bit ADF in ADCSR is set to 1 after A/D conversion is completed. The DTC or DMAC can be activated by an ADI interrupt. Having the converted data read by the DTC or DMAC in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

**Table 16.5 A/D Converter Interrupt Source**

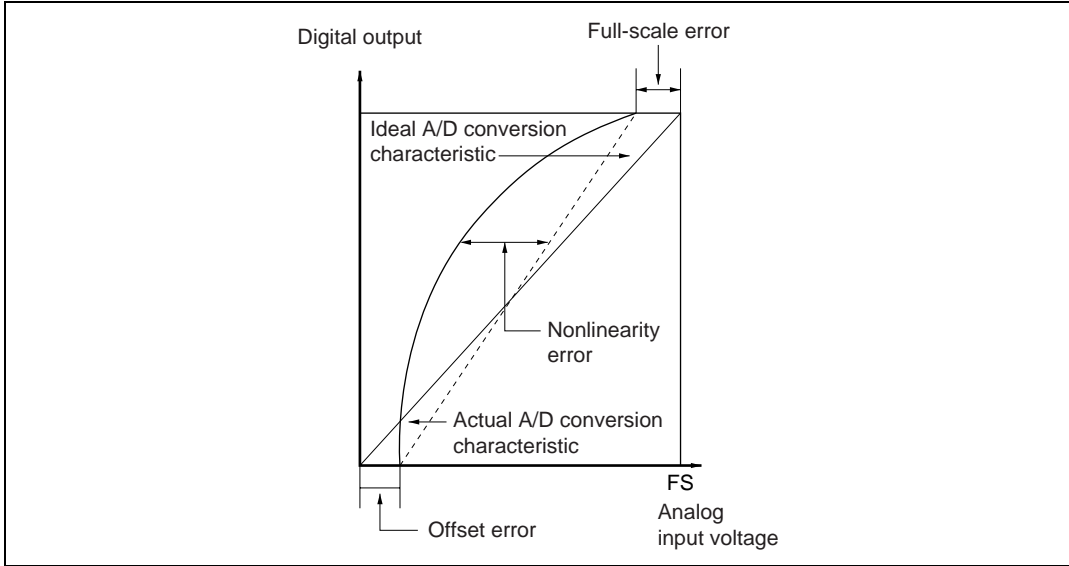
Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ADI	End of conversion	ADF	Possible	Possible

This LSI's A/D conversion precision definitions are given below.

- Resolution  
The number of A/D converter digital output codes
- Quantization error  
The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 16.4).
- 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 B'000000000 (H'000) to B'000000001 (H'001) (see figure 16.5).
- Full-scale error  
The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'111111110 (H'3FE) to B'111111111 (H'3FF) (see figure 16.5).
- 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 (see figure 16.5).
- Absolute precision  
The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.



**Figure 16.4 A/D Conversion Precision Definitions**



**Figure 16.5 A/D Conversion Precision Definitions**

## 16.7.1 Module Stop Mode Setting

Operation of the A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the A/D converter to be halted. Register access is enabled by clearing module stop mode. For details, refer to section 23, Power-Down Modes.

## 16.7.2 Permissible Signal Source Impedance

This LSI's analog input is designed so that conversion precision is guaranteed for an input signal for which the signal source impedance is  $5\text{ k}\Omega$  or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds  $5\text{ k}\Omega$ , charging may be insufficient and it may not be possible to guarantee the A/D conversion precision. However, if a large capacitance is provided externally for conversion in single mode, the input load will essentially comprise only the internal input resistance of  $10\text{ k}\Omega$ , and the signal source impedance is ignored. However, since a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g.,  $5\text{ mV}/\mu\text{s}$  or greater) (see figure 16.6). When converting a high-speed analog signal or conversion in scan mode, a low-impedance buffer should be inserted.

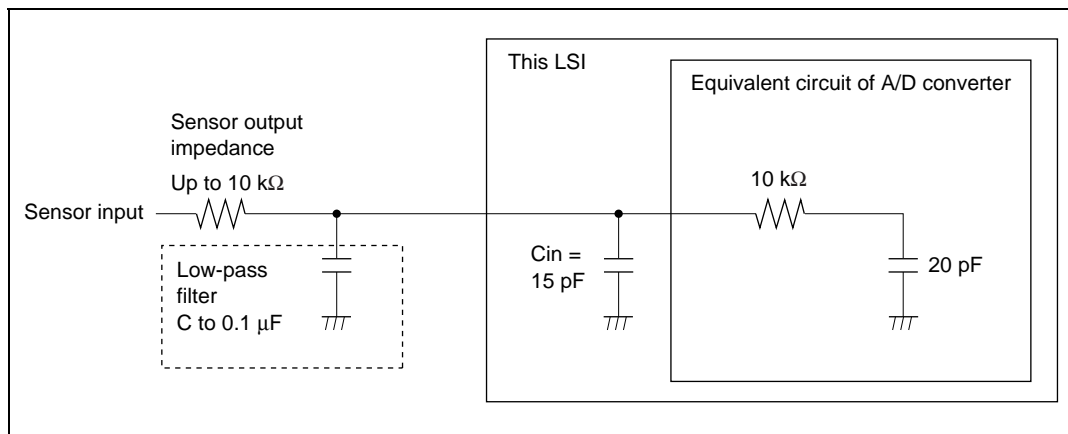


Figure 16.6 Example of Analog Input Circuit



Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AVss.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, so acting as antennas.

#### 16.7.4 Setting Range of Analog Power Supply and Other Pins

If conditions shown below are not met, the reliability of the device may be adversely affected.

- Analog input voltage range  
The voltage applied to analog input pin ANn during A/D conversion should be in the range  $AV_{ss} \leq AV_n \leq V_{ref}$ .
- Relation between AVcc, AVss and Vcc, Vss  
As the relationship between AVcc, AVss and Vcc, Vss, set  $AV_{cc} \geq V_{cc}$  and  $AV_{ss} = V_{ss}$ . If the A/D converter is not used, the AVcc and AVss pins must not be left open.
- Vref setting range

The reference voltage at the Vref pin should be set in the range  $V_{ref} \leq AV_{cc}$ .

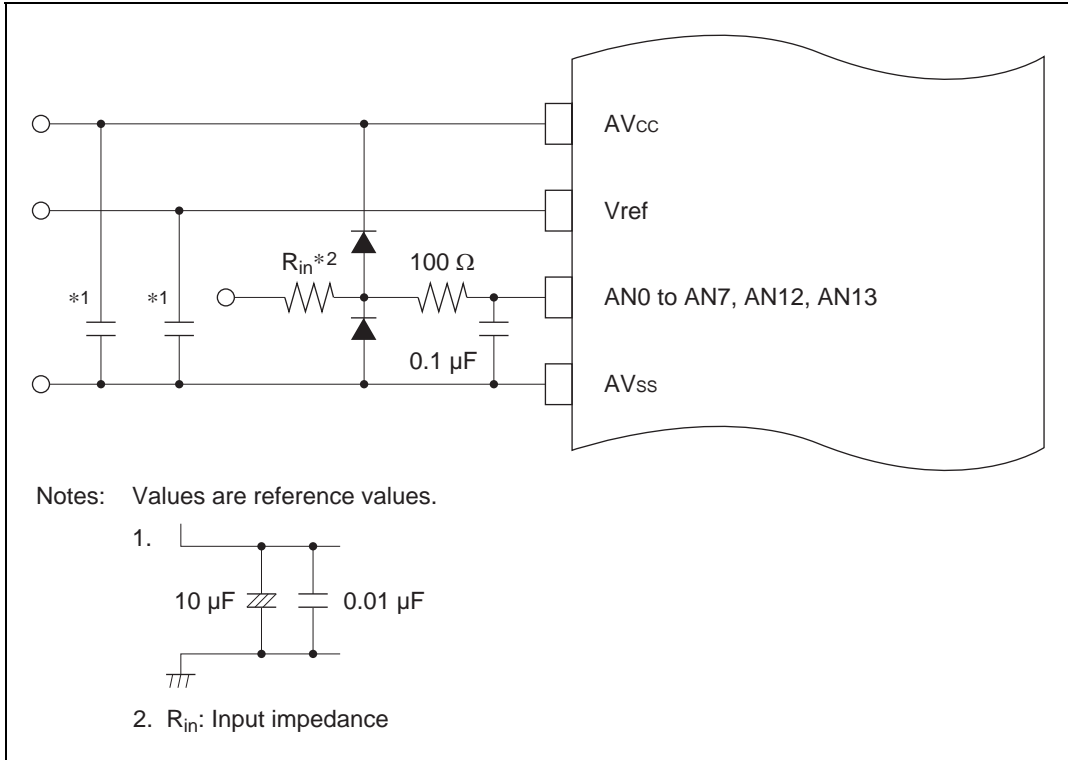
#### 16.7.5 Notes on Board Design

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 (AN0 to AN7, AN12, AN13), analog reference power supply (Vref), and analog power supply (AVcc) by the analog ground (AVss). Also, the analog ground (AVss) should be connected at one point to a stable digital ground (Vss) on the board.

A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN7, AN12, AN13) should be connected between AVcc and AVss as shown in figure 16.7. Also, the bypass capacitors connected to AVcc and the filter capacitor connected to AN0 to AN15 must be connected to AVss.

If a filter capacitor is connected, the input currents at the analog input pins (AN0 to AN7, AN12, AN13) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance ( $R_{in}$ ), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.



**Figure 16.7 Example of Analog Input Protection Circuit**

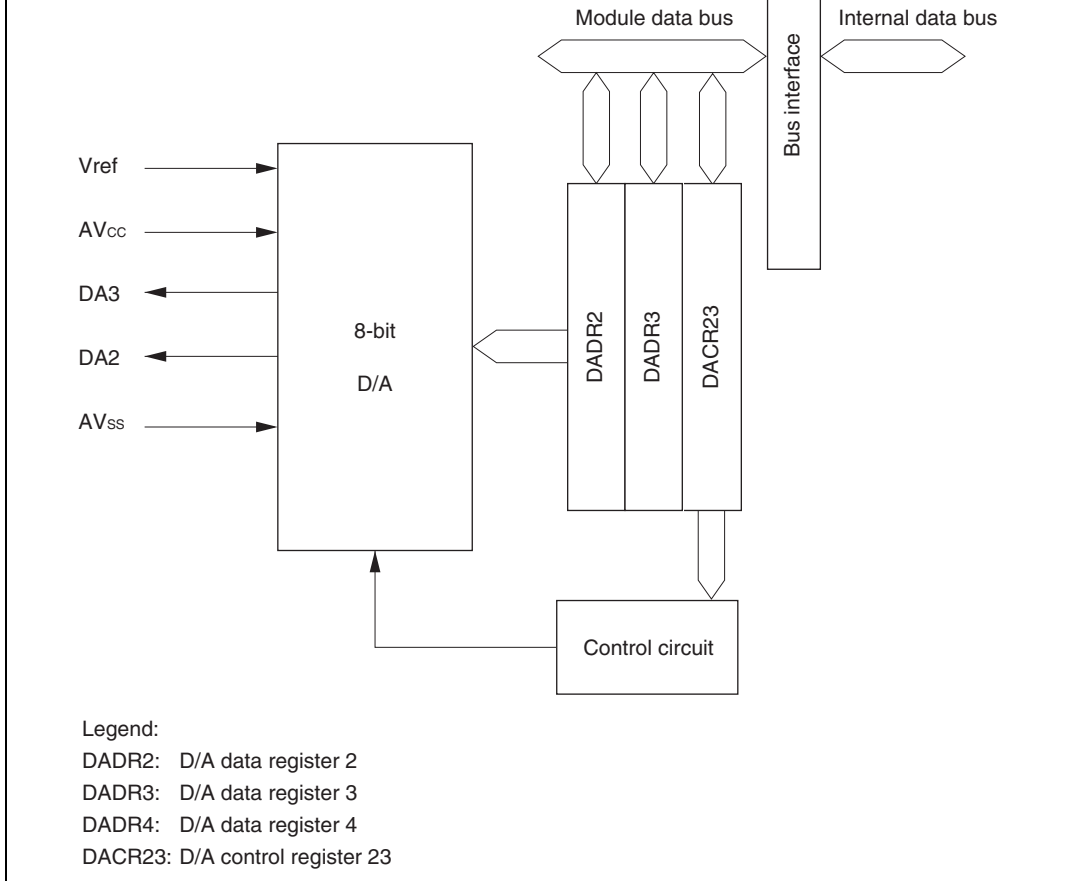
Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	k $\Omega$



## 17.1 Features

D/A converter features are listed below.

- 8-bit resolution
- Two output channels
- Maximum conversion time of 10  $\mu$ s (with 20 pF load)
- Output voltage of 0 V to Vref
- D/A output hold function in software standby mode
- Setting the module stop mode



**Figure 17.1 Block Diagram of D/A Converter**

Table 17.1 summarizes the input and output pins of the D/A converter.

**Table 17.1 Pin Configuration**

<b>Pin Name</b>	<b>Symbol</b>	<b>I/O</b>	<b>Function</b>
Analog power pin	AV <sub>CC</sub>	Input	Analog power
Analog ground pin	AV <sub>SS</sub>	Input	Analog ground
Reference voltage pin	V <sub>ref</sub>	Input	Reference voltage of D/A converter
Analog output pin 2	DA2	Output	Channel 2 analog output
Analog output pin 3	DA3	Output	Channel 3 analog output

## 17.3 Register Descriptions

The D/A converter has the following registers.

- D/A data register 2 (DADR2)
- D/A data register 3 (DADR3)
- D/A control register 01 (DACR01)
- D/A control register 23 (DACR23)

### 17.3.1 D/A Data Registers 2 and 3 (DADR2 and DADR3)

DADR2 and DADR3 are 8-bit readable/writable registers that store data for conversion.

Whenever output is enabled, the values in DADR are converted and output to the analog output pins.

DACR23 control the operation of the D/A converter.

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE3	0	R/W	D/A Output Enable 3 Controls D/A conversion and analog output. 0: Analog output (DA3) is disabled 1: Channel 3 D/A conversion is enabled; analog output (DA3) is enabled
6	DAOE2	0	R/W	D/A Output Enable 2 Controls D/A conversion and analog output. 0: Analog output (DA2) is disabled 1: Channel 2 D/A conversion is enabled; analog output (DA2) is enabled
5	DAE	0	R/W	D/A Enable Used together with the DAOE0 and DAOE1 bits to control D/A conversion. When the DAE bit is cleared to 0, channel 2 and 3 D/A conversions are controlled independently. When the DAE bit is set to 1, channel 2 and 3 D/A conversions are controlled together. Output of conversion results is always controlled independently by the DAOE0 and DAOE1 bits. For details, see table 17.2.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.



Bit 5 DAE	Bit 7 DAOE3	Bit 6 DAOE2	Description
0	0	0	D/A conversion disabled
		1	Channel 2 D/A conversion enabled, channel3 D/A conversion disabled
	1	0	Channel 3 D/A conversion enabled, channel2 D/A conversion disabled
		1	Channel 2 and 3 D/A conversions enabled
1	0	0	D/A conversion disabled
		1	Channel 2 and 3 D/A conversions enabled
	1	0	
		1	

## 17.4 Operation

The D/A converter includes D/A conversion circuits for two channels, each of which can operate independently.

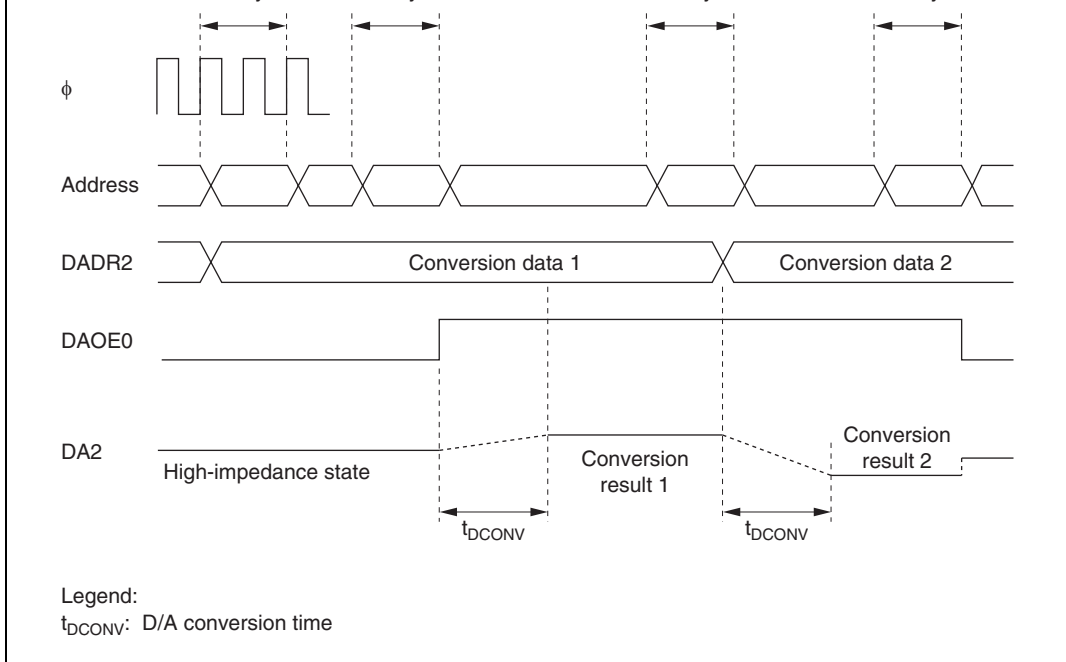
When DAOE bit in DACR23 is set to 1, D/A conversion is enabled and the conversion result is output.

The operation example concerns D/A conversion on channel 2. Figure 17.2 shows the timing of this operation.

- [1] Write the conversion data to DADR2.
- [2] Set the DAOE0 bit in DACR23 to 1. D/A conversion is started. The conversion result is output from the analog output pin DA2 after the conversion time  $t_{\text{DCONV}}$  has elapsed. The conversion result is continued to output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{DADR}2\text{contents}}{256} \times V_{\text{ref}}$$

- [3] If DADR2 is written to again, the conversion is immediately started. The conversion result is output after the conversion time  $t_{\text{DCONV}}$  has elapsed.
- [4] If the DAOE0 bit is cleared to 0, analog output is disabled.



**Figure 17.2 Example of D/A Converter Operation**

## 17.5 Usage Notes

### 17.5.1 Setting for Module Stop Mode

It is possible to enable/disable the D/A converter operation using the module stop control register, the D/A converter does not operate by the initial value of the register. The register can be accessed by releasing the module stop mode. For details, see section 23, Power-Down Modes.

### 17.5.2 D/A Output Hold Function in Software Standby Mode

If D/A conversion is enabled and this LSI enters software standby mode, D/A output is held and analog power supply current remains at the same level during D/A conversion. When the analog power supply current is required to go low in software standby mode, bits DAOE2, DAOE3 and DAE should be cleared to 0, and D/A output should be disabled.

This LSI has an on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU to both byte data and word data.

The on-chip RAM can be enabled or disabled by means of the RAME bit in the system control register (SYSCR). For details on the system control register (SYSCR), refer to section 3.2.2, System Control Register (SYSCR).

Part No.		ROM Type	RAM Capacitance	RAM Address
H8S/2368F	HD64F2368	Flash memory version	32 kbytes	H'FF4000 to H'FFBFFF
	HD64F2367		24 kbytes	H'FF6000 to H'FFBFFF
	HD64F2364		32 kbytes	H'FF4000 to H'FFBFFF
	HD64F2362			
	HD64F2361		24 kbytes	H'FF6000 to H'FFBFFF
	HD64F2360		16 kbytes	H'FF8000 to H'FFBFFF
	HD6432365	Masked ROM version		
	HD6412363	ROMless version		



The features of the flash memory included in the flash memory version are summarized below. The block diagram of the flash memory is shown in figure 19.1.

## 19.1 Features

- Size

Product Classification		ROM Size	ROM Address
H8S/2368 Group	HD64F2367	384 kbytes	H'000000 to H'05FFFF (Modes 3, 4, and 7)

- Programming/erase methods

The flash memory is programmed 128 bytes at a time. Erase is performed in single-block units. The flash memory of 384 kbytes is configured as follows: 64 kbytes × 5 blocks, 32 kbytes × 1 block, and 4 kbytes × 8 blocks. To erase the entire flash memory, each block must be erased in turn.

- Reprogramming capability

The flash memory can be reprogrammed up to 100 times.

- Two on-board programming modes

Boot mode

User program mode

On-board programming/erasing can be done in boot mode in which the on-chip boot program is started for erase or programming of the entire flash memory. In normal user program mode, individual blocks can be erased or programmed.

- Programmer mode

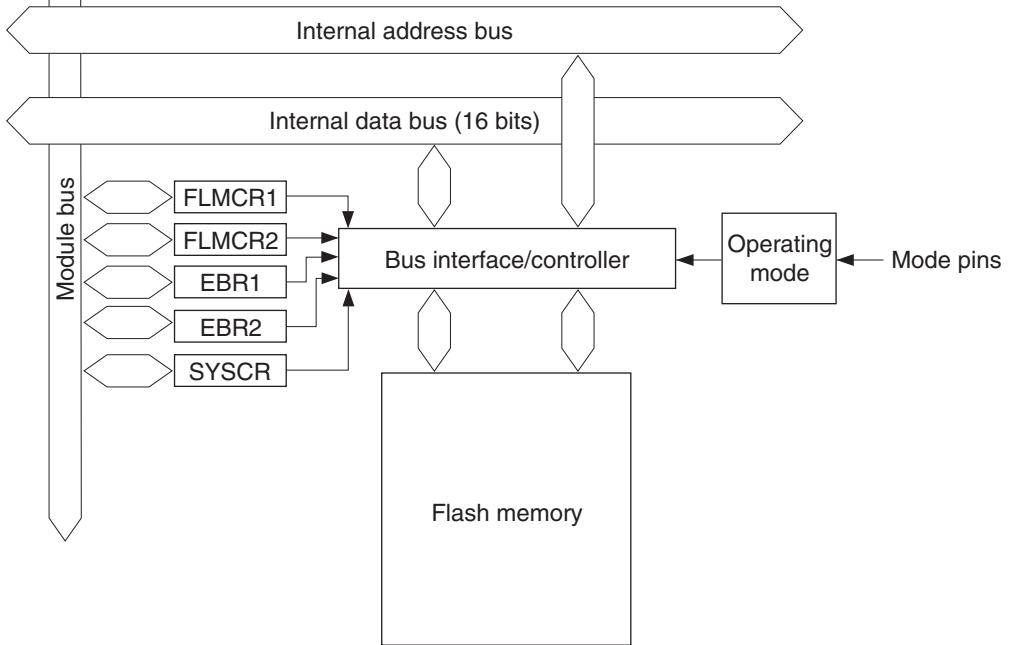
Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

- Automatic bit rate adjustment

With data transfer in boot mode, the bit rate of this LSI can be automatically adjusted to match the transfer bit rate of the host.

- Programming/erasing protection

There are three protect modes, hardware, software, and error protect, which allow protected status to be designated for flash memory program/erase operations.



Legend:

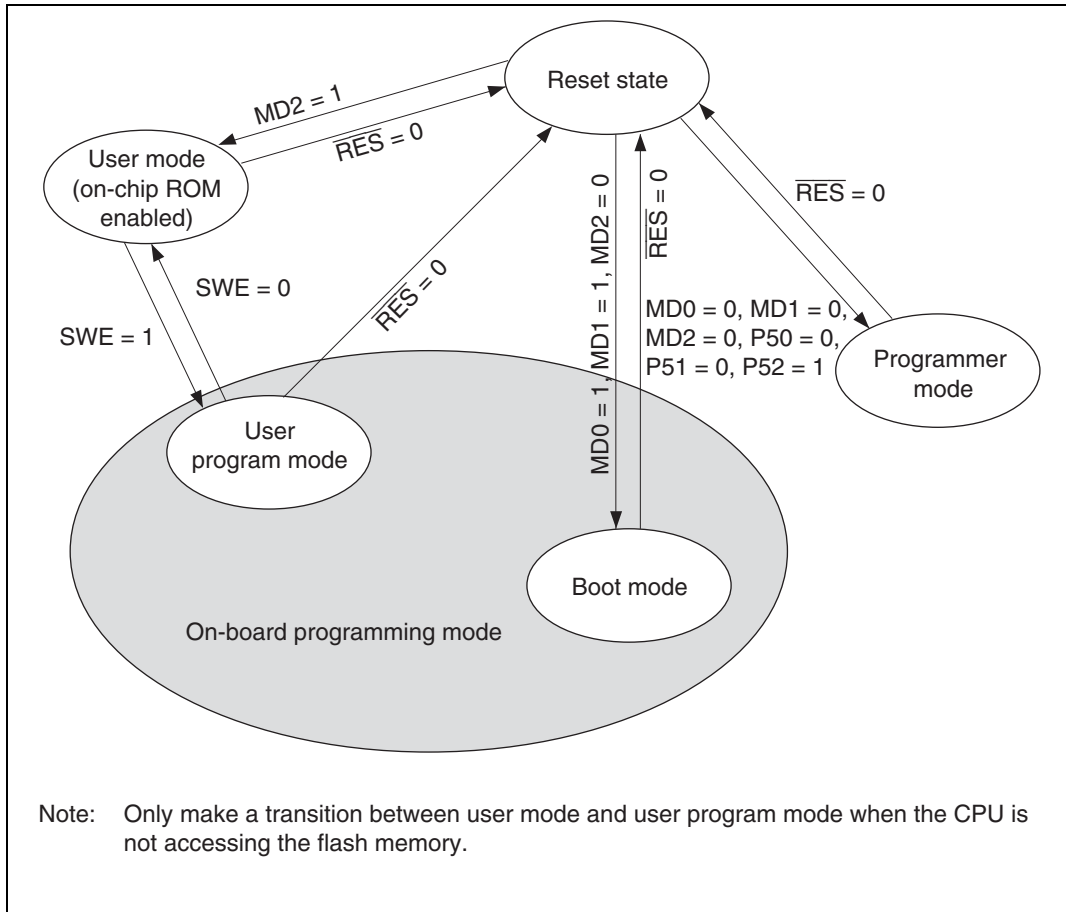
- FLMCR1: Flash memory control register 1
- FLMCR2: Flash memory control register 2
- EBR1: Erase block register 1
- EBR2: Erase block register 2
- SYSCR: System control register

**Figure 19.1 Block Diagram of Flash Memory**

When the mode pins are set in the reset state and a reset-start is executed, this LSI enters an operating mode as shown in figure 19.2. In user mode, flash memory can be read but not programmed or erased.

The boot, user program and programmer modes are provided as modes to write and erase the flash memory.

The differences between boot mode and user program mode are shown in table 19.1. Figure 19.3 shows boot mode. Figure 19.4 shows user program mode.



**Figure 19.2 Flash Memory State Transitions**

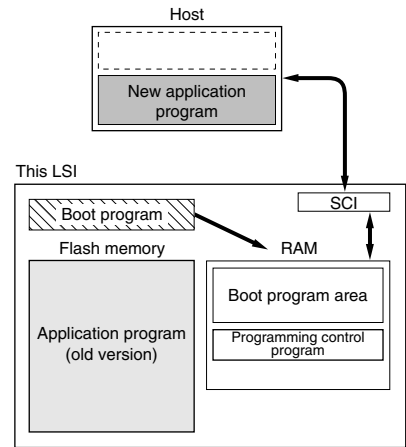
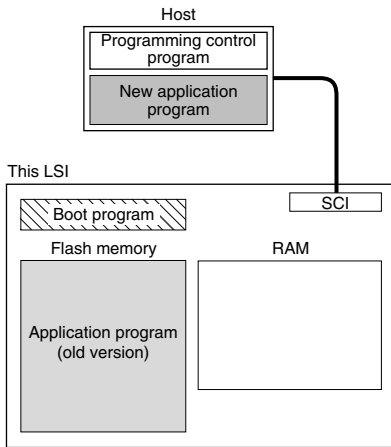
	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	Program/program-verify	Erase/erase-verify/program/ program-verify

Note: \* To be provided by the user, in accordance with the recommended algorithm.



In the flash memory. The user should prepare the programming control program and new application program beforehand in the host.

the chip (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.

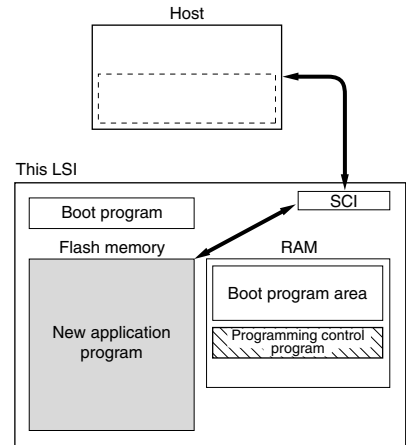
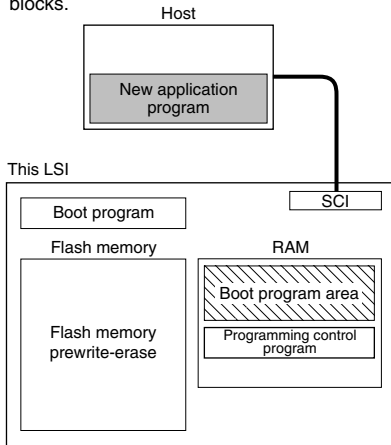


### 3. Flash memory initialization

The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, entire flash memory erasure is performed, without regard to blocks.

### 4. Writing new application program

The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

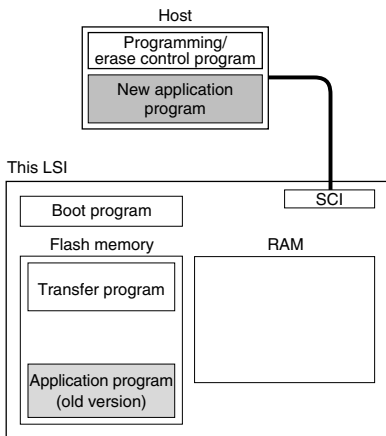


Program execution state

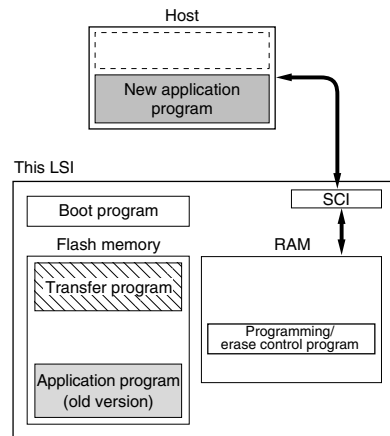
**Figure 19.3 Boot Mode**

programming/erase control program to on-chip RAM should be written into the flash memory by the user beforehand. (2) The programming/erase control program should be prepared in the host or in the flash memory.

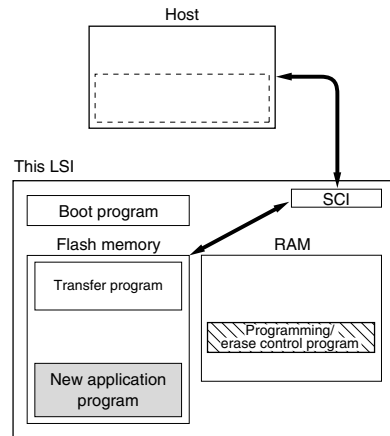
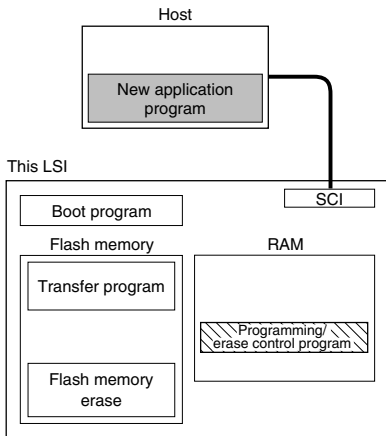
software confirms this fact, executes the transfer program in the flash memory, and transfers the programming/erase control program to RAM.



3. Flash memory initialization  
The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program  
Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



Program execution state

Figure 19.4 User Program Mode

Figure 19.5 shows the block configuration of 384-kbyte flash memory. The thick lines indicate erasing units, the narrow lines indicate programming units, and the values are addresses. The 384-kbyte flash memory is divided into 64 kbytes (5 blocks), 32 kbytes (1 block), and 4 kbytes (8 blocks). Erasing is performed in these divided units. Programming is performed in 128-byte units starting from an address whose lower eight bits are H'00 or H'80.

4 kbytes				← Programming unit: 128 bytes →	H'000FFF
EB1 Erase unit 4 kbytes	H'001000	H'001001	H'001002		H'00107F
					H'001FFF
EB2 Erase unit 4 kbytes	H'002000	H'002001	H'002002	← Programming unit: 128 bytes →	H'00207F
					H'002FFF
EB3 Erase unit 4 kbytes	H'003000	H'003001	H'003002	← Programming unit: 128 bytes →	H'00307F
					H'003FFF
EB4 Erase unit 4 kbytes	H'004000	H'004001	H'004002	← Programming unit: 128 bytes →	H'00407F
EB7 Erase unit 4 kbytes	H'007000	H'007001	H'007002	← Programming unit: 128 bytes →	H'00707F
					H'007FFF
EB8 Erase unit 32 kbytes	H'008000	H'008001	H'008002	← Programming unit: 128 bytes →	H'00807F
					H'00FFFF
EB9 Erase unit 64 kbytes	H'010000	H'010001	H'010002	← Programming unit: 128 bytes →	H'01007F
					H'01FFFF
EB10 Erase unit 64 kbytes	H'020000	H'020001	H'020002	← Programming unit: 128 bytes →	H'02007F
					H'02FFFF
EB11 Erase unit 64 kbytes	H'030000	H'030001	H'030002	← Programming unit: 128 bytes →	H'03007F
					H'03FFFF
EB12 Erase unit 64 kbytes	H'040000	H'040001	H'040002	← Programming unit: 128 bytes →	H'04007F
					H'04FFFF
EB13 Erase unit 64 kbytes	H'050000	H'050001	H'050002	← Programming unit: 128 bytes →	H'05007F
					H'05FFFF

**Figure 19.5 384-kbyte Flash Memory Block Configuration (Modes 3, 4, and 7)**

The flash memory is controlled by means of the pins shown in table 19.2.

**Table 19.2 Pin Configuration**

<b>Pin Name</b>	<b>I/O</b>	<b>Function</b>
$\overline{\text{RES}}$	Input	Reset
MD2	Input	Sets this LSI's operating mode
MD1	Input	Sets this LSI's operating mode
MD0	Input	Sets this LSI's operating mode
P52	Input	Sets operating mode in programmer mode
P51	Input	Sets operating mode in programmer mode
P50	Input	Sets operating mode in programmer mode
TxD1	Output	Serial transmit data output
RxD1	Input	Serial receive data input

## 19.5 Register Descriptions

The flash memory has the following registers. For details on the system control register, refer to section 3.2.2, System Control Register (SYSCR).

- Flash memory control register 1 (FLMCR1)
- Flash memory control register 2 (FLMCR2)
- Erase block register 1 (EBR1)
- Erase block register 2 (EBR2)

FLMCR1 is a register that makes the flash memory transit to program mode, program-verify mode, erase mode, or erase-verify mode. For details on register setting, refer to section 19.7, Flash Memory Programming/Erasing.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0/1	R	This bit is reserved. This bit is always read as 0 in modes 1 and 2. This bit is always read as 1 in modes 3 to 7.
6	SWE	0	R/W	Software Write Enable When this bit is set to 1, flash memory programming/erasing is enabled. When this bit is cleared to 0, other FLMCR1 register bits and all EBR1 and EBR2 bits cannot be set.
5	ESU	0	R/W	Erase Setup When this bit is set to 1 while SWE = 1, the flash memory transits to the erase setup state. When it is cleared to 0, the erase setup state is cancelled.
4	PSU	0	R/W	Program Setup When this bit is set to 1 while SWE = 1, the flash memory transits to the program setup state. When it is cleared to 0, the program setup state is cancelled.
3	EV	0	R/W	Erase-Verify When this bit is set to 1 while SWE = 1, the flash memory transits to erase-verify mode. When it is cleared to 0, erase-verify mode is cancelled.
2	PV	0	R/W	Program-Verify When this bit is set to 1 while SWE = 1, the flash memory transits to program-verify mode. When it is cleared to 0, program-verify mode is cancelled.
1	E	0	R/W	Erase When this bit is set to 1 while SWE = 1, and ESU = 1, the flash memory transits to erase mode. When it is cleared to 0, erase mode is cancelled.
0	P	0	R/W	Program When this bit is set to 1 while SWE = 1, and PSU = 1, the flash memory transits to program mode. When it is cleared to 0, program mode is cancelled.

FLMCR2 is a register that displays the state of flash memory programming/erasing. When the SWE bit in FLMCR1 is cleared to 0, FLMCR2 is initialized to H'00. FLMCR2 is a read-only register, and should not be written to.

Bit	Bit Name	Initial Value	R/W	Description
7	FLER	0	R	Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state. See section 19.8.3, Error Protection, for details.
6 to 0	—	All 0	R	Reserved These bits always read 0.

EBR1 specifies the flash memory erase area block. EBR1 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Set only one bit in EBR1 and EBR2 together (do not set more than one bit at the same time. Setting more than one bit will automatically clear all EBR1 and EBR2 bits to 0). For details, see table 19.3.

Bit	Bit Name	Initial Value	R/W	Description
7	EB7	0	R/W	When this bit is set to 1, 4 kbytes of EB7 are to be erased.
6	EB6	0	R/W	When this bit is set to 1, 4 kbytes of EB6 are to be erased.
5	EB5	0	R/W	When this bit is set to 1, 4 kbytes of EB5 are to be erased.
4	EB4	0	R/W	When this bit is set to 1, 4 kbytes of EB4 are to be erased.
3	EB3	0	R/W	When this bit is set to 1, 4 kbytes of EB3 are to be erased.
2	EB2	0	R/W	When this bit is set to 1, 4 kbytes of EB2 are to be erased.
1	EB1	0	R/W	When this bit is set to 1, 4 kbytes of EB1 are to be erased.
0	EB0	0	R/W	When this bit is set to 1, 4 kbytes of EB0 are to be erased.

#### 19.5.4 Erase Block Register 2 (EBR2)

EBR2 specifies the flash memory erase area block. EBR2 is initialized to H'00 when the SWE bit in FLMCR1 is 0. Set only one bit in EBR2 and EBR1 together (do not set more than one bit at the same time. Setting more than one bit will automatically clear all EBR1 and EBR2 bits to 0). For details, see table 19.3.



				The initial value should not be changed.
5	EB13	0	R/W	When this bit is set to 1, 64 kbytes of EB13 are to be erased.
4	EB12	0	R/W	When this bit is set to 1, 64 kbytes of EB12 are to be erased.
3	EB11	0	R/W	When this bit is set to 1, 64 kbytes of EB11 are to be erased.
2	EB10	0	R/W	When this bit is set to 1, 64 kbytes of EB10 are to be erased.
1	EB9	0	R/W	When this bit is set to 1, 64 kbytes of EB9 are to be erased.
0	EB8	0	R/W	When this bit is set to 1, 32 kbytes of EB8 are to be erased.

**Table 19.3 Erase Blocks**

<b>Block (Size)</b>	<b>Address</b>
	<b>Modes 3, 4, and 7</b>
EB0 (4 kbytes)	H'000000 to H'000FFF
EB1 (4 kbytes)	H'001000 to H'001FFF
EB2 (4 kbytes)	H'002000 to H'002FFF
EB3 (4 kbytes)	H'003000 to H'003FFF
EB4 (4 kbytes)	H'004000 to H'004FFF
EB5 (4 kbytes)	H'005000 to H'005FFF
EB6 (4 kbytes)	H'006000 to H'006FFF
EB7 (4 kbytes)	H'007000 to H'007FFF
EB8 (32 kbytes)	H'008000 to H'00FFFF
EB9 (64 kbytes)	H'010000 to H'01FFFF
EB10 (64 kbytes)	H'020000 to H'02FFFF
EB11 (64 kbytes)	H'030000 to H'03FFFF
EB12 (64 kbytes)	H'040000 to H'04FFFF
EB13 (64 kbytes)	H'050000 to H'05FFFF

In an on-board programming mode, programming, erasing, and verification for the on-chip flash memory can be performed. There are two on-board programming modes: boot mode and user program mode. Table 19.4 shows how to select boot mode. User program mode can be selected by setting the control bits by software. For a diagram that shows mode transitions of flash memory, see figure 19.2.

**Table 19.4 Setting On-Board Programming Mode**

	<b>Mode Setting</b>	<b>MD2</b>	<b>MD1</b>	<b>MD0</b>
Boot mode	Single-chip activation expanded mode with on-chip ROM enabled	0	1	1

### 19.6.1 Boot Mode

When this LSI enters boot mode, the embedded boot program is started. The boot program transfers the programming control program from the externally connected host to the on-chip RAM via the SCI\_1. When the flash memory is all erased, the programming control program is executed.

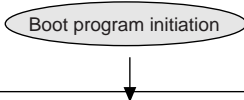
Table 19.5 shows the boot mode operations between reset end and branching to the programming control program.

1. When the boot program is initiated, the SCI\_1 should be set to asynchronous mode, the chip measures the low-level period of asynchronous SCI communication data (H'00) transmitted continuously from the host. The chip then calculates the bit rate of transmission from the host, and adjusts the SCI\_1 bit rate to match that of the host. The transfer format is 8-bit data, 1 stop bit, and no parity. The reset should end with the RxD pin high. The RxD and TxD pins should be pulled up on the board if necessary. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period.
2. After matching the bit rates, the chip transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the chip. If reception could not be performed normally, initiate boot mode again by a reset. Depending on the host's transfer bit rate and system clock frequency of this LSI, there will be a discrepancy between the bit rates of the host and the chip. To operate the SCI properly, set the host's transfer bit rate and system clock frequency of this LSI within the ranges listed in table 19.6.

description in section 19.7, Flash Memory Programming/Erasing.

4. Before branching to the programming control program, the chip terminates transfer operations by the SCI\_1 (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of program data or verify data with the host. The TxD pin is high. The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, since the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
5. In boot mode, if flash memory contains data (all data is not 1), all blocks of flash memory are erased. Boot mode is used for the initial programming in the on-board state or for a forcible return when a program that is to be initiated in user program mode was accidentally erased and could not be executed in user program mode.

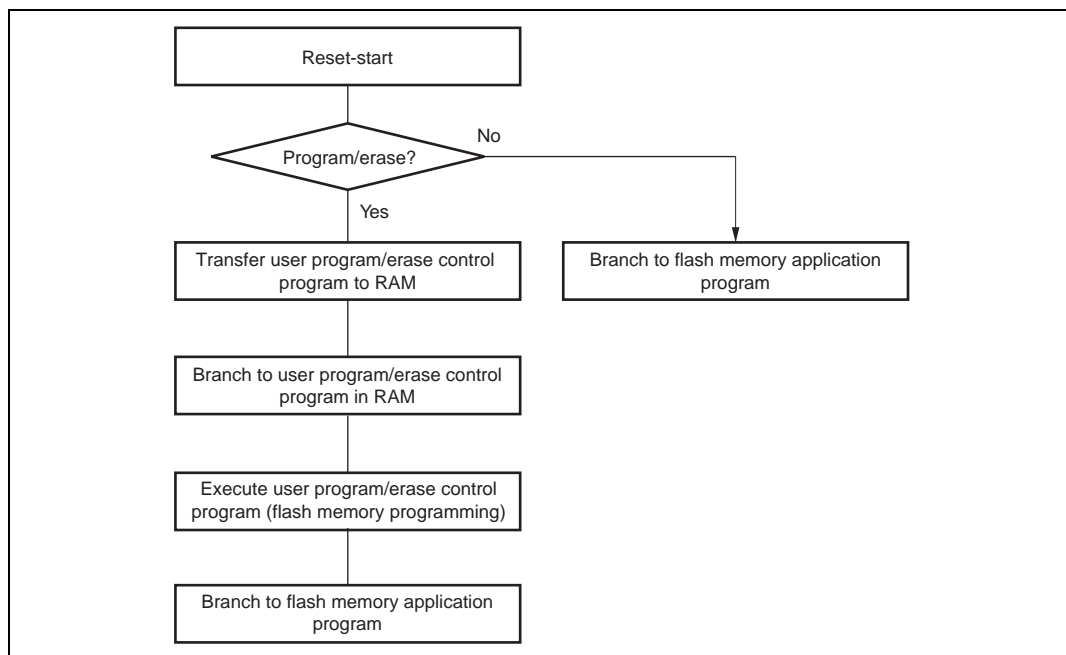
- Notes:
1. In boot mode, a part of the on-chip RAM area (H'FF8000 to H'FF87FF) is used by the boot program. Addresses H'FF8800 to H'FFBFFF is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.
  2. Boot mode can be cleared by a reset. Release the reset by setting the MD pins, after waiting at least 20 states since driving the reset pin low. Boot mode is also cleared when the WDT overflow reset occurs.
  3. Do not change the MD pin input levels in boot mode.
  4. All interrupts are disabled during programming or erasing of the flash memory.

Item	Host Operation	Communication Contents	LSI Operation
	Processing Contents		Processing Contents
Boot mode initiation			Branches to boot program at reset-start.  
Bit rate adjustment	Continuously transmits data H'00 at specified bit rate. ↓ Transmits data H'55 when data H'00 is received error-free. ↓ H'AA reception.	H'00, H'00 ··· H'00  ← H'00 ← H'55 ← H'AA	<ul style="list-style-type: none"> <li>Measures low-level period of receive data H'00.</li> <li>Calculates bit rate and sets BRR in SCI_1.</li> <li>Transmits data H'00 to host as adjustment end indication.</li> </ul> Transmits data H'AA to host when data H'55 is received.
Transfer of number of bytes of programming control program	Transmits number of bytes (N) of programming control program to be transferred as 2-byte data (low-order byte following high-order byte). ↓ Transmits 1-byte of programming control program (repeated for N times).	← Upper bytes, lower bytes ← Echoback  ← H'XX ← Echoback	Echobacks the 2-byte data received to host. ↓ Echobacks received data to host and also transfers it to RAM. (repeated for N times)
Flash memory erase	↓ H'AA reception.	← H'FF ← H'AA	Checks flash memory data, erases all flash memory blocks in case of written data existing, and transmits data H'AA to host. (If erase could not be done, transmits data H'FF to host and aborts operation.)
			↓ Branches to programming control program transferred to on-chip RAM and starts execution.

Host Bit Rate	System Clock Frequency Range of LSI
19,200 bps	8 to 25 MHz
9,600 bps	8 to 25 MHz

## 19.6.2 User Program Mode

On-board programming/erasing of an individual flash memory block can also be performed in user program mode by branching to a user program/erase program. The user must set branching conditions and provide on-board means of supplying programming data. The flash memory must contain the program/erase program or a program which provides the program/erase program from external memory. Because the flash memory itself cannot be read during programming/erasing, transfer the program/erase program to on-chip RAM, as like in boot mode. Figure 19.6 shows a sample procedure for programming/erasing in user program mode. Prepare a program/erase program in accordance with the description in section 19.7, Flash Memory Programming/Erasing.



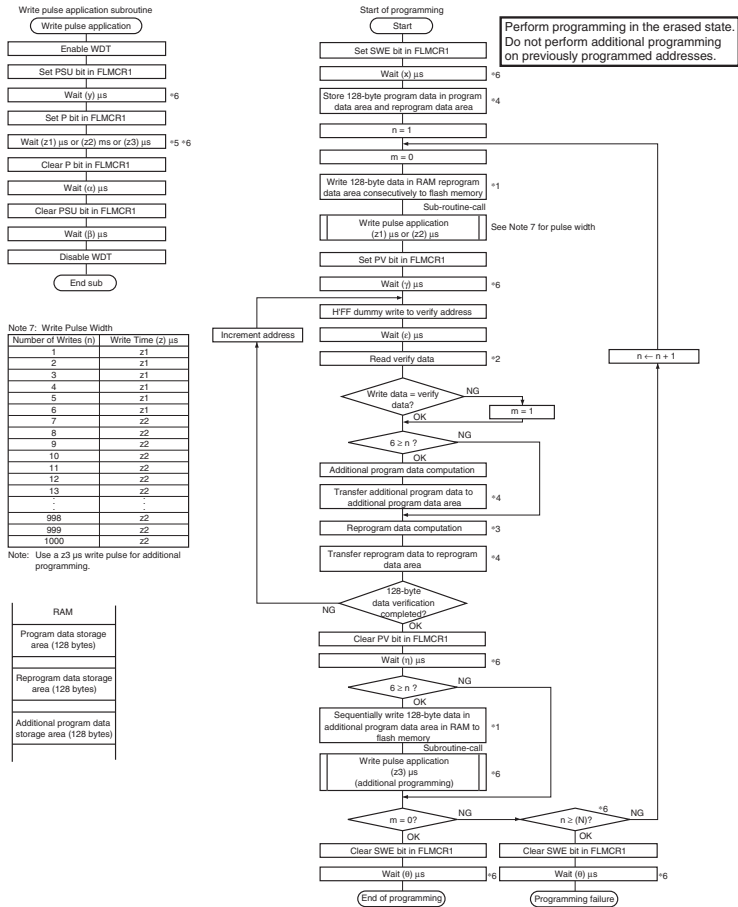
**Figure 19.6 Programming/Erasing Flowchart Example in User Program Mode**

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. Depending on the FLMCR1 and FLMCR2 setting, the flash memory operates in one of the following four modes: program mode, erase mode, program-verify mode, and erase-verify mode. The programming control program in boot mode and the user program/erase program in user mode use these operating modes in combination to perform programming/erasing. Flash memory programming and erasing should be performed in accordance with the descriptions in section 19.7.1, Program/Program-Verify and section 19.7.2, Erase/Erase-Verify, respectively.

### 19.7.1 Program/Program-Verify

When programming data or programs to the flash memory, the program/program-verify flowchart shown in figure 19.7 should be followed. Performing programming operations according to this flowchart will enable data or programs to be programmed to the flash memory without subjecting the chip to voltage stress or sacrificing program data reliability.

1. Programming must be done to an empty address. Do not reprogram an address to which programming has already been performed.
2. Programming should be carried out 128 bytes at a time. A 128-byte data transfer must be performed even if programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. Prepare the following data storage areas in RAM: a 128-byte programming data area, a 128-byte reprogramming data area, and a 128-byte additional-programming data area. Perform reprogramming data computation and additional programming data computation according to figure 19.7.
4. Consecutively transfer 128 bytes of data in byte units from the programming data area, reprogramming data area, or additional-programming data area to the flash memory. The program address and 128-byte data are latched in the flash memory. The lower 8 bits of the start address in the flash memory destination area must be H'00 or H'80.
5. The time during which the P bit is set to 1 is the programming time. Figure 19.7 shows the allowable programming times.
6. The watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. Set a value greater than  $(y + z2 + \alpha + \beta)$   $\mu$ s as the WDT overflow period.
7. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower 2 bits are B'00. Verify data can be read in words from the address to which a dummy write was performed.



- Notes:
- Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, HFF data must be written to the extra addresses.
  - Verify data is read in 16-bit (W) units.
  - The reprogram data is given by the operation of the following tables (comparison between stored data in the program data area and verify data). Programming is executed for the bits of reprogram data 0 in the next reprogram loop. Even bits for which programming has been completed will be subjected to additional programming if they fail the subsequent verify operation.
  - A 128-byte areas for storing program data, reprogram data, and additional program data must be provided in the RAM. The contents of the reprogram and additional program data are modified as programming proceeds.
  - A write pulse of (z1) or (z2) μs should be applied according to the progress of the programming operation. See Note \*7 for the pulse widths. When writing of additional-programming data is executed, a (z3) μs write pulse should be applied. Reprogram data 'X' means reprogram data when the write pulse is applied.
  - For the values of x, y, z1, z2, z3, α, β, γ, ε, η, θ, and N, see section 25.2.5, Flash Memory Characteristics.

Program Data Operation Chart

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
0	1	0	Programming incomplete; reprogram
1	0	1	Programming completed
1	1	1	Still in erased state; no action

Additional Program Data Operation Chart

Reprogram Data (X)	Verify Data (V)	Additional Program Data (Y)	Comments
0	0	0	Additional programming executed
0	1	1	Additional programming not executed
1	0	0	Additional programming not executed
1	1	1	Additional programming not executed

**Figure 19.7 Program/Program-Verify Flowchart**

When erasing flash memory, the erase/erase-verify flowchart shown in figure 19.8 should be followed.

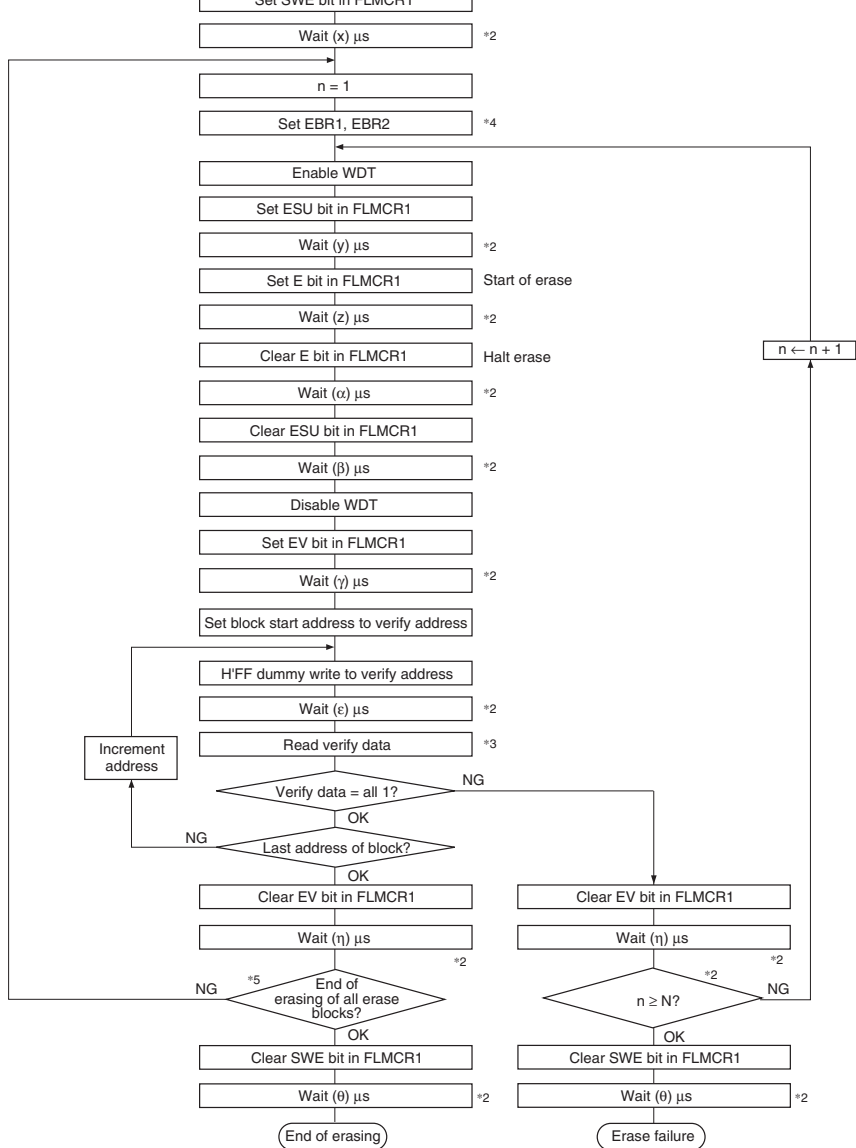
1. Prewriting (setting erase block data to all 0s) is not necessary.
2. Erasing is performed in block units. Make only a single-bit specification in the erase block registers (EBR1 and EBR2). To erase multiple blocks, each block must be erased in turn.
3. The time during which the E bit is set to 1 is the flash memory erase time.
4. The watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. Set a value greater than  $(y + z + \alpha + \beta)$  ms as the WDT overflow period.
5. For a dummy write to a verify address, write 1-byte data H'FF to an address whose lower two bits are B'00. Verify data can be read in longwords from the address to which a dummy write was performed.
6. If the read data is not erased, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence (N) must not be exceeded.

### 19.7.3 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI input, are disabled when flash memory is being programmed or erased, and while the boot program is executing in boot mode. There are three reasons for this:

1. Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. If the interrupt exception handling is started when the vector address has not been programmed yet or the flash memory is being programmed or erased, the vector would not be read correctly, possibly resulting in CPU runaway.
3. If an interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.





- Notes:
1. Prewriting (setting erase block data to all 0) is not necessary.
  2. The values of  $x$ ,  $y$ ,  $z$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\epsilon$ ,  $\eta$ ,  $\theta$ , and  $N$  are shown in section 25.2.5, Flash Memory Characteristics.
  3. Verify data is read in 16-bit (W) units.
  4. Set only one bit in EBR1 or EBR2. More than one bit cannot be set.
  5. Erasing is performed in block units. To erase a number of blocks, the individual blocks must be erased sequentially.

**Figure 19.8 Erase/Erase-Verify Flowchart**

There are three kinds of flash memory program/erase protection: hardware protection, software protection, and error protection.

### 19.8.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted because of a transition to reset (including an overflow reset by the WDT) or standby mode. Flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2) are initialized. In a reset via the  $\overline{\text{RES}}$  pin, the reset state is not entered unless the  $\overline{\text{RES}}$  pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the  $\overline{\text{RES}}$  pin low for the RES pulse width specified in the AC Characteristics section.

### 19.8.2 Software Protection

Protection can be implemented against programming/erasing of all flash memory blocks by clearing the SWE bit in FLMCR1 to 0. When protection is in effect, setting the P or E bit in FLMCR1 does not cause a transition to program mode or erase mode. By setting the erase block register 1 (EBR1) and erase block register 2 (EBR2), erase protection can be set for individual blocks. When EBR1 and EBR2 are set to H'00, erase protection is set for all blocks.

### 19.8.3 Error Protection

In error protection, an error is detected when the CPU's runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is forcibly aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

When the following errors are detected during programming/erasing of flash memory, the FLER bit in FLMCR2 is set to 1, and the error protection state is entered.

- When flash memory is read during programming/erasing (including a vector read or instruction fetch)
- When an exception handling (excluding a reset) is started during programming/erasing
- When a SLEEP instruction is executed during programming/erasing
- When the CPU releases the bus during programming/erasing

cannot be re-entered by re-setting the P or E bit. However, since PV and EV bit setting is enabled, and a transition can be made to verify mode. The error protection state can be canceled by a reset or in hardware standby mode.

## 19.9 Programmer Mode

In programmer mode, a PROM programmer can perform programming/erasing via a socket adapter, just like for a discrete flash memory. Use a PROM programmer which supports the Renesas 512-kbyte flash memory on-chip MCU device type (FZTAT512V3A). A 12-MHz input clock is needed.

## 19.10 Power-Down States for Flash Memory

In user mode, the flash memory will operate in either of the following states:

- Normal operating mode  
The flash memory can be read.
- Standby mode  
All flash memory circuits are halted.

Table 19.7 shows the correspondence between the operating modes of this LSI and the flash memory. When the flash memory returns to normal operation from a standby state, a power supply circuit stabilization period is needed. When the flash memory returns to its normal operating state, bits STS3 to STS0 in SBYCR must be set to provide a wait time of at least 100  $\mu$ s, even when the external clock is being used.

**Table 19.7 Flash Memory Operating States**

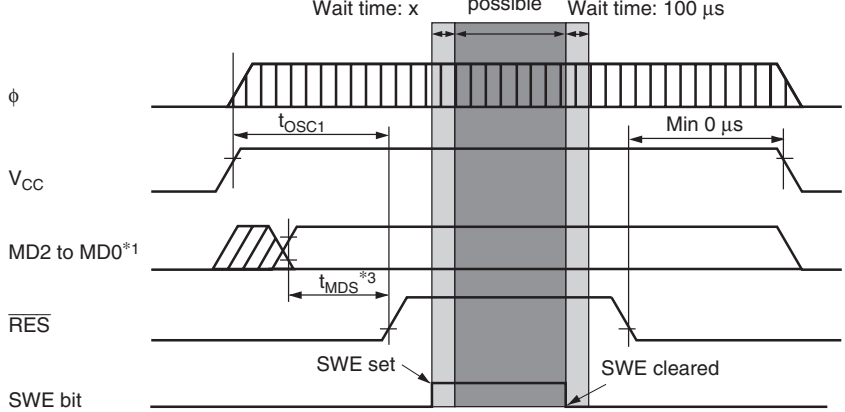
<b>Operating Mode</b>	<b>Flash Memory Operating State</b>
Active mode	Normal operating state
Sleep mode	Normal operating state
Standby mode	Standby state

Precautions concerning the use of on-board programming mode, and programmer mode are summarized below.

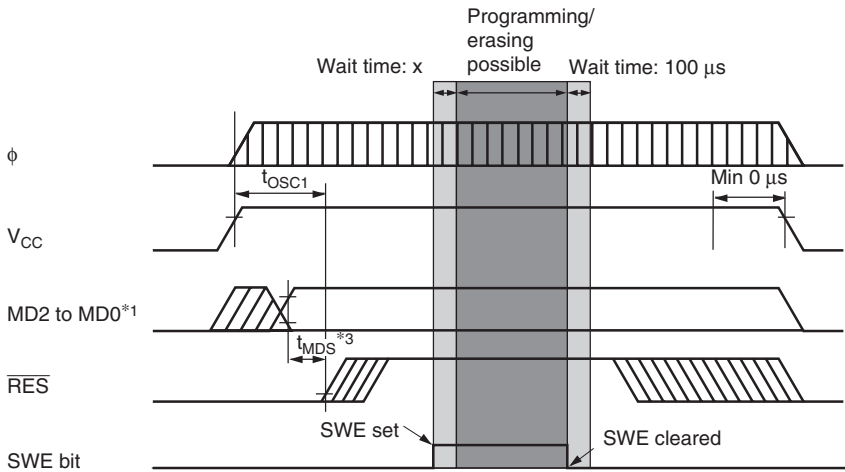
1. Use the specified voltages and timing for programming and erasing.  
Applied voltages in excess of the rating can permanently damage the device. Use a PROM programmer that supports the Renesas microcomputer device type with 512-kbyte on-chip flash memory (FZTAT512V3A).  
Do not select the HN27C4096 setting for the PROM programmer, and only use the specified socket adapter.
2. Reset the flash memory before turning on/off the power.  
When applying or disconnecting Vcc power, fix the  $\overline{\text{RES}}$  pin low and place the flash memory in the hardware protection state. The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery.
3. Use the recommended algorithm when programming and erasing flash memory.  
The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P or E bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.
4. Do not set or clear the SWE bit during execution of a program in flash memory.  
Wait for at least 100  $\mu\text{s}$  after clearing the SWE bit before executing a program or reading data in flash memory.  
When the SWE bit is set, data in flash memory can be rewritten. When the SWE bit is set to 1, data in flash memory can be read only in program-verify/erase-verify mode. Access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE bit during programming, erasing, or verifying.
5. Do not use interrupts while flash memory is being programmed or erased.  
All interrupt requests, including NMI, should be disabled during programming/erasing the flash memory to give priority to program/erase operations.
6. Do not perform additional programming. Erase the memory before reprogramming.  
In on-board programming, perform only one programming operation on a 128-byte programming unit block. In programmer mode, too, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.
7. Before programming, check that the chip is correctly mounted in the PROM programmer.  
Overcurrent damage to the device can result if the index marks on the PROM programmer socket, socket adapter, and chip are not correctly aligned.

9. Apply the reset signal after the SWE, bit is cleared during its operation.

The reset signal is applied at least 100  $\mu$ s after the SWE bit has been cleared.



(1) Boot Mode

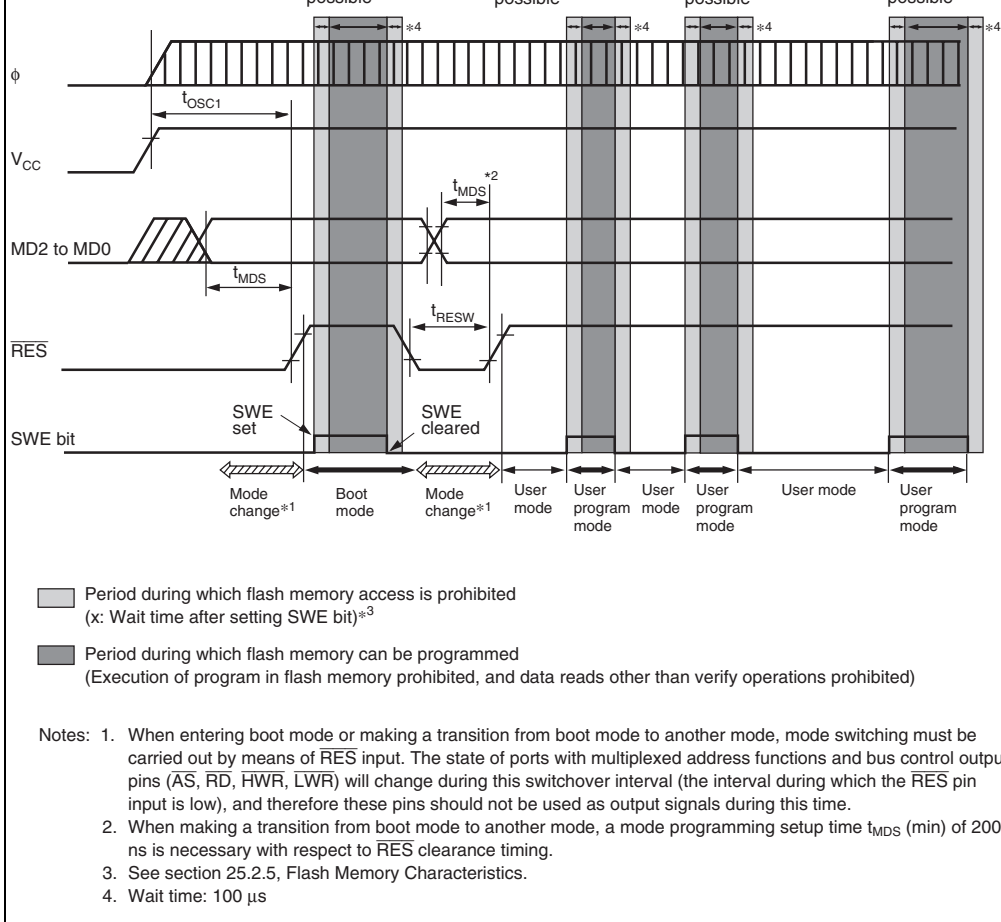


(2) User Program Mode

- Period during which flash memory access is prohibited (x: Wait time after setting SWE bit)\*2
- Period during which flash memory can be programmed (Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

- Notes:
1. Except when switching modes, the level of the mode pins (MD2 to MD0) must be fixed until power-off by pulling the pins up or down.
  2. See section 25.2.5, Flash Memory Characteristics.
  3. Mode programming setup time  $t_{MDS}$  (min) = 200 ns

**Figure 19.9 Power-On/Off Timing**



**Figure 19.10 Mode Transition Timing**  
**(Example: Boot Mode → User Mode ↔ User Program Mode)**





The flash memory has the following features. Figure 20.1 shows a block diagram of the flash memory.

## 20.1 Features

- Size

Product Classification		ROM Size	ROM Address
H8S/2368F	HD64F2368	512 kbytes	H'000000 to H'07FFFF (Modes 3 to 5 and 7)
H8S/2364F	HD64F2364	384 kbytes	H'000000 to H'05FFFF (Modes 3 to 5 and 7)
H8S/2362F	HD64F2362	256 kbytes	H'000000 to H'03FFFF (Modes 3 to 5 and 7)
H8S/2361F	HD64F2361		
H8S/2360F	HD64F2360		

- 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 memory MAT is initiated at a power-on reset in user mode: 512 kbytes (H8S/2368F), 384 kbytes (H8S/2364F), 256 kbytes (H8S/2362F, H8S/2361F, H8S/2360F)
  - The user boot memory MAT is initiated at a power-on reset in user boot mode: 8 kbytes
- 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.
- Programming/erasing time  
The flash memory programming time is 1 ms (typ) in 128-byte simultaneous programming and 8 μs per byte. The erasing time is 750 ms (typ) per 64-kbyte block.
- Number of programming  
The number of flash memory programming can be up to 100 times.

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 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.

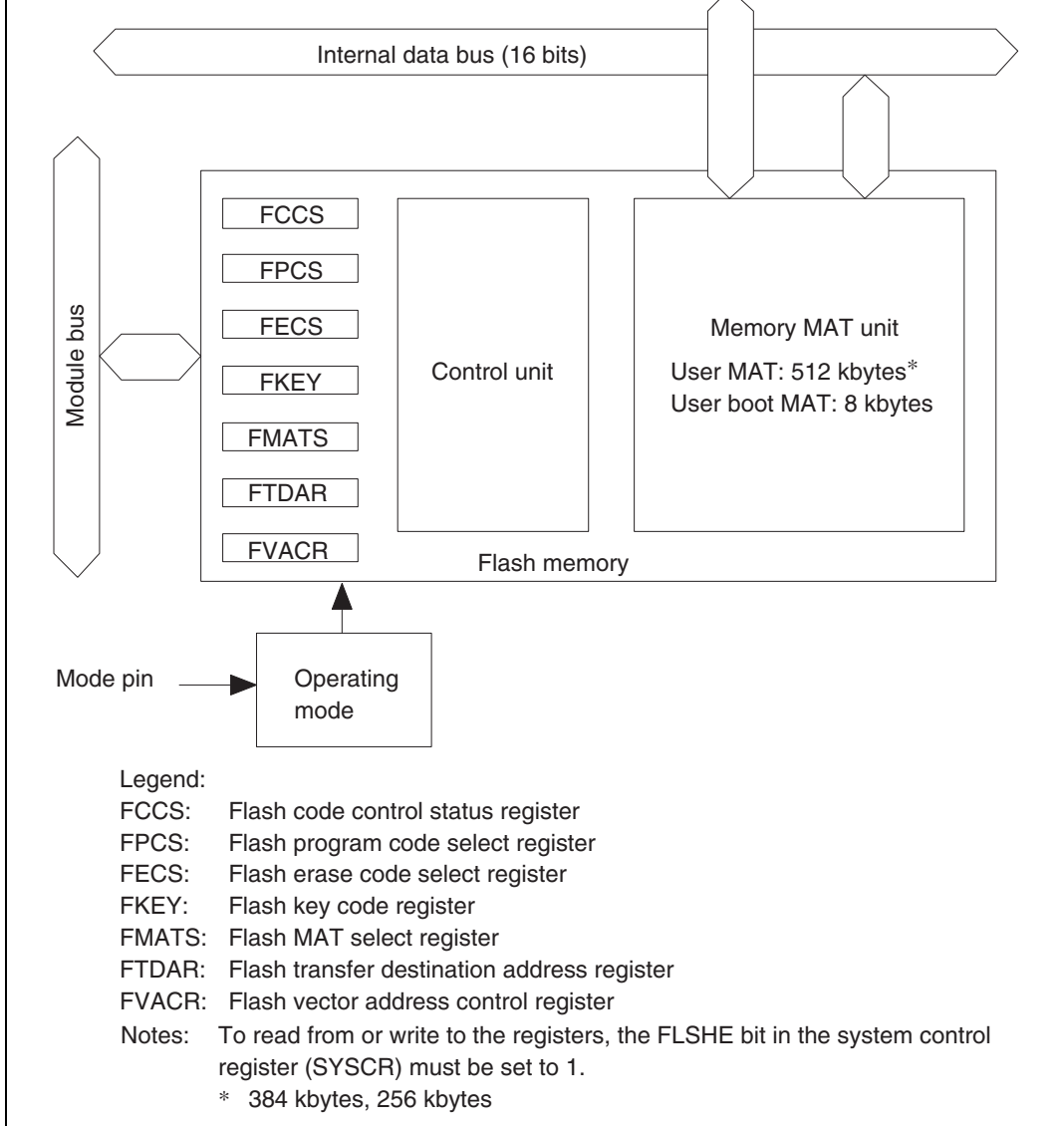
• One off-board programming mode

— PROM mode

This mode uses the PROM programmer. The user MAT and user boot MAT can be programmed.

• Protection modes

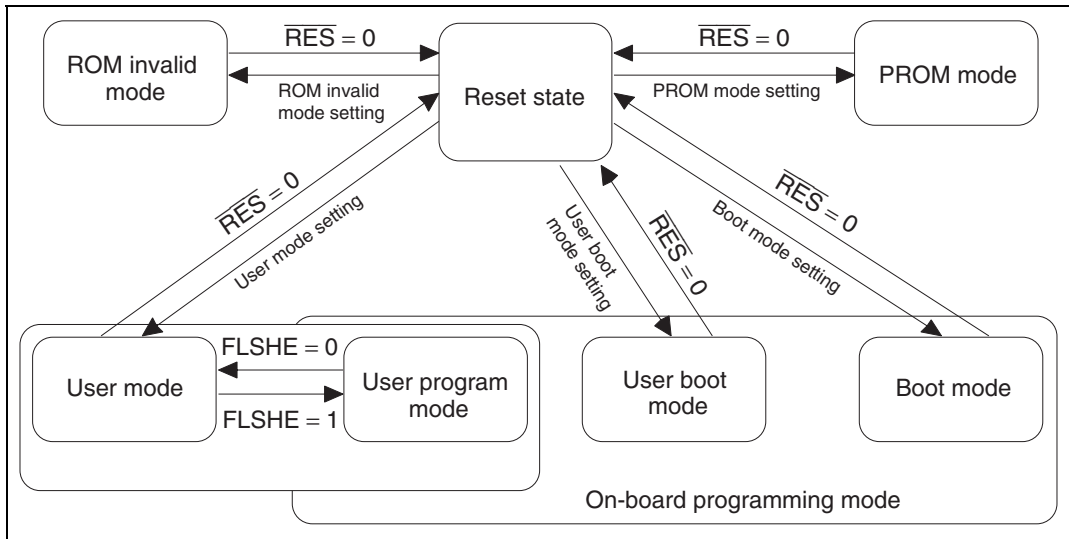
There are two protection modes: software protection by the register setting and hardware protection by the FWE pin. The protection state for flash memory programming/erasing can be set.



**Figure 20.1 Block Diagram of Flash Memory**

When each mode pin and the FWE pin are set in the reset state and reset start is performed, the microcomputer enters each operating mode as shown in figure 20.2.

- Flash memory cannot be read, programmed, or erased in ROM invalid mode.
- 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 PROM mode.



**Figure 20.2 Mode Transition of Flash Memory**

The comparison table of programming and erasing related items about boot mode, user program mode, user boot mode, and PROM mode is shown in table 20.1.

**Table 20.1 Comparison of Programming Modes**

	<b>Boot mode</b>	<b>User program mode</b>	<b>User boot mode</b>	<b>PROM 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
All erasure	○ (Automatic)	○	○	○ (Automatic)
Block division erasure	○ *1	○	○	×
Program data transfer	From host via SCI	From optional device via RAM	From optional device via RAM	Via programmer
Reset initiation MAT	Embedded program storage MAT	User MAT	User boot MAT*2	—
Transition to user mode	Changing mode setting and reset	Changing FLSHE bit	Changing mode setting and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.

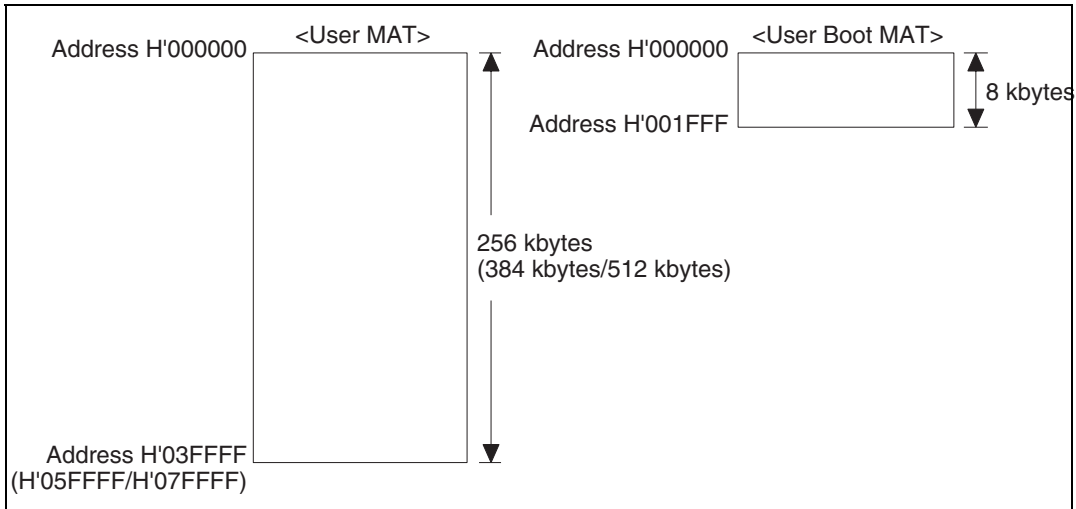
2. Firstly, the reset vector is fetched from the embedded program storage MAT. After the flash memory related registers are checked, the reset vector is fetched from the user boot MAT.

- The user boot MAT can be programmed or erased only in boot mode and PROM mode.
- The user MAT and user boot MAT are 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.
- The boot operation of the optional interface can be performed by the mode pin setting different from user program mode in user boot mode.

This LSI's flash memory is configured by the 256-kbyte/384-kbyte/512-kbyte 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 two MATs, the MAT must be switched by using FMATS.

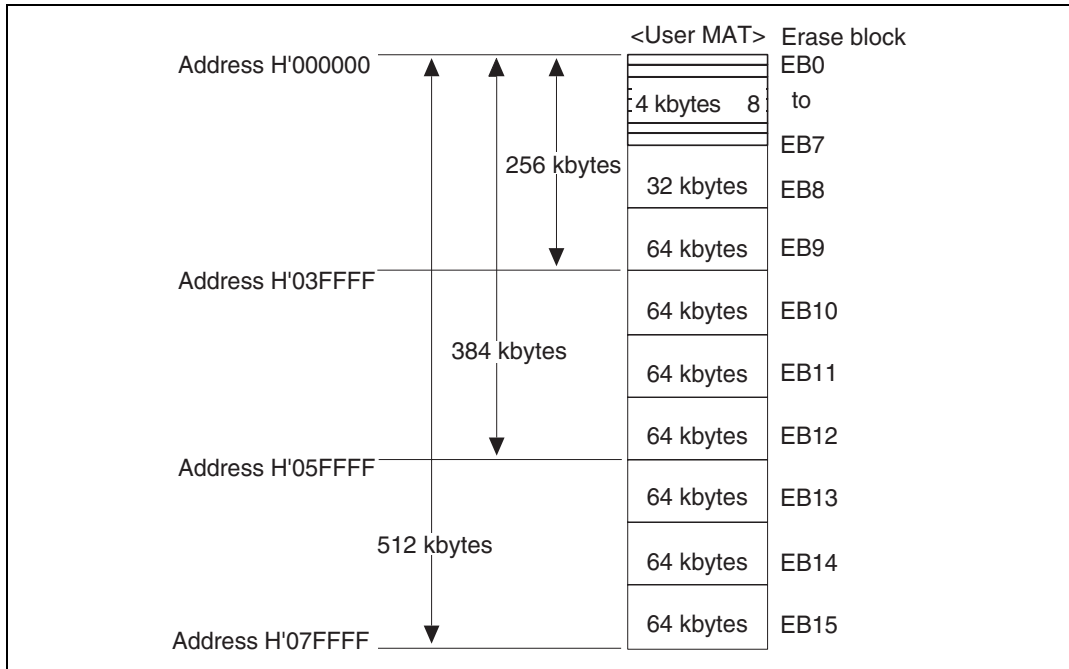
The user MAT or user boot MAT can be read in all modes if it is in ROM valid mode. However, the user boot MAT can be programmed only in boot mode and PROM mode.



**Figure 20.3 Flash Memory Configuration**

The size of the user MAT is different from that of the user boot MAT. An address which exceeds the size of the 8-kbyte user boot MAT should not be accessed. If the attempt is made, data is read as undefined value.

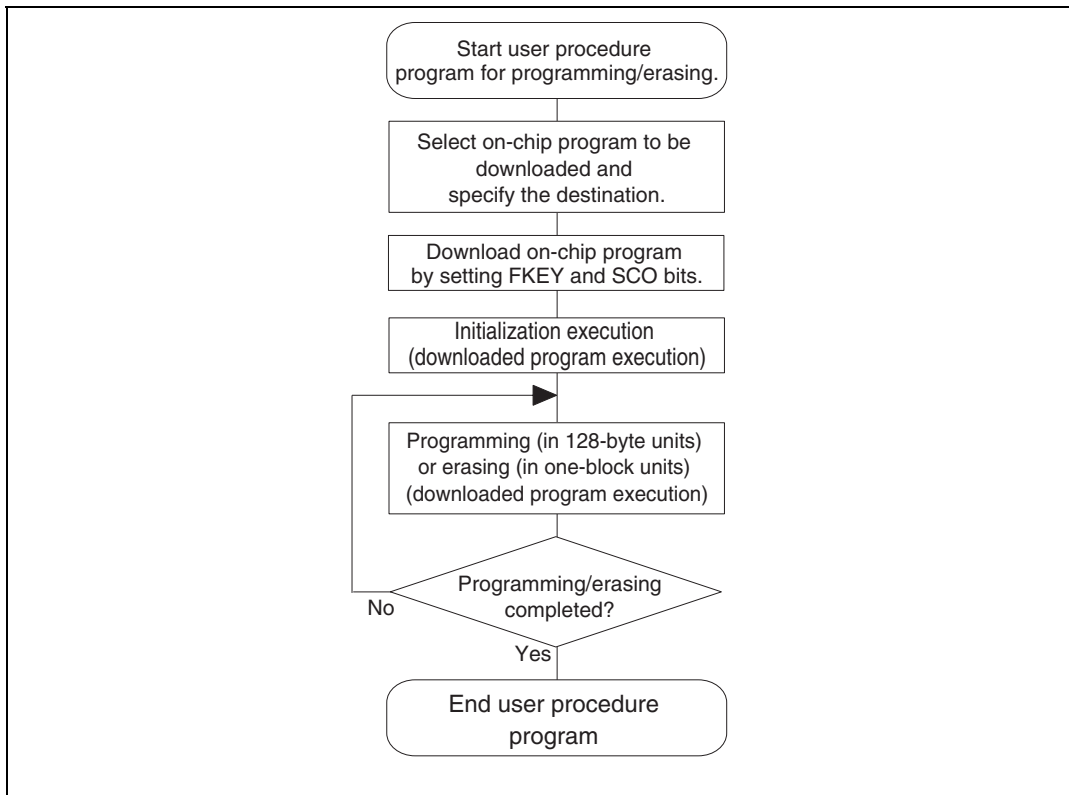
The user MAT is divided into 64 kbytes (seven blocks), 32 kbytes (one block), and 4 kbytes (eight blocks) as shown in figure 20.4. The user MAT can be erased in this divided-block units and the erase-block number of EB0 to EB15 is specified when erasing.



**Figure 20.4 Block Division of User MAT**

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 register/parameter.

The procedure program is made by the user in user program mode and user boot mode. An overview of the procedure is given as follows. For details, see section 20.4.2, User Program Mode.



**Figure 20.5 Overview of User Procedure Program**

#### 1. Selection of on-chip program to be downloaded

For programming/erasing execution, the FLSHE bit must be set to 1 to transition to user program mode.

This LSI has programming/erasing programs which 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 register. The address of the programming destination is specified by the FTDAR.



register (FKEY) and the flash control register (FCCS) of the programming/erasing interface register.

The flash memory 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 the 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 parameter, whether the normal download is executed or not can be confirmed.

### 3. Initialization of programming/erasing

The operating frequency is set before execution of programming/erasing. This setting is performed by using the programming/erasing interface parameter.

### 4. Programming/erasing execution

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 parameter and the on-chip program is initiated. The on-chip program is executed by using the JSR or BSR instruction and performing the subroutine call of the specified address in the on-chip RAM. The execution result is returned to the programming/erasing interface parameter.

The area to be programmed must be erased in advance when programming flash memory. All interrupts are prohibited during programming and erasing. Interrupts must be masked within the user system.

### 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.

Table 20.2 shows the flash memory pin configuration.

**Table 20.2 Pin Configuration**

Pin Name	Abbreviation	Input/Output	Function
Reset	$\overline{\text{RES}}$	Input	Reset
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
Port 52	P52	Input	Sets operating mode of programmer mode
Port 51	P51	Input	Sets operating mode of programmer mode
Port 50	P50	Input	Sets operating mode of programmer mode
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 20.7, Programmer Mode.

## 20.3 Register Descriptions

The registers/parameters which control flash memory are shown as follows.

- Flash code control status register (FCCS)
- Flash program code select register (FPCS)
- Flash erase code select register (FECS)
- Flash key code register (FKEY)
- Flash MAT select register (FMATS)
- Flash transfer destination address register (FTDAR)
- Download pass and fail result (DPFP)
- Flash pass and fail result (FPFR)
- Flash multipurpose address area (FMPAR)
- Flash multipurpose data destination area (FMPDR)
- Flash erase Block select (FEBS)

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 20.3.

**Table 20.3 Register/Parameter and Target Mode**

		Download	Initiali- zation	Program- ming	Erasure	Read
Programming/erasing interface register	FCCS	○	—	—	—	—
	FPFS	○	—	—	—	—
	FECS	○	—	—	—	—
	FKEY	○	—	○	○	—
	FMATS	—	—	○*1	○*1	○*2
	FTDAR	○	—	—	—	—
Programming/erasing interface parameter	DPFR	○	—	—	—	—
	FPFR	—	○	○	○	—
	FPEFEQ	—	○	—	—	—
	FUBRA	—	○	—	—	—
	FMPAR	—	—	○	—	—
	FMPDR	—	—	○	—	—
	FEBS	—	—	—	○	—

- 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.

The programming/erasing interface registers are as described below. They are all 8-bit registers that can be accessed in byte. Except for the FLER bit in FCCS, these registers are initialized at a power-on reset, in hardware standby mode, or in software standby mode. The FLER bit is not initialized in software standby mode.

- 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 on-chip program.

Bit	Bit Name	Initial Value	R/W	Description
7	—	1	R	Reserved This bit is always read as 0. The write value should always be 1.
6, 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

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 transition to a power-on reset or 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 must be released after the reset period of 100  $\mu$ s which is longer than normal.

0: Flash memory operates normally

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 conditions]

- When an interrupt, such as NMI, occurs during programming/erasing flash memory.
- When the flash memory is read during programming/erasing flash memory
- When the SLEEP instruction is executed during programming/erasing flash memory
- When a bus master other than the CPU, such as the DMAC, DTC, or BREQ, gets bus mastership during programming/erasing flash memory.

---

3 to 1 — All 0 R

Reserved

These bits are always read as 0. The write value should always be 0.

---

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 specified by FTDAR.

In order to set this bit to 1, H'A5 must be written to FKEY, and this operation must be executed in the on-chip RAM.

Four NOP instructions must be executed immediately after setting this bit to 1.

Since this bit is cleared to 0 when download is completed, this bit cannot be read as 1.

All interrupts must be disabled. This should be made in the user system.

0: Download of the on-chip programming/erasing program to the on-chip RAM is not executed

[Clear condition] When download is completed

1: Request that the on-chip programming/erasing program is downloaded to the on-chip RAM is occurred

[Set conditions] When all of the following conditions are satisfied and 1 is written to this bit

- H'A5 is written to FKEY
  - During execution in the on-chip RAM
-

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	PPVS	0	R/W	Program Pulse Verify Selects the programming program. 0: On-chip programming program is not selected [Clear condition] When transfer is completed 1: On-chip programming program is selected

- Flash Erase Code Select Register (FECS)

FECS selects download of the on-chip erasing program.

Bit	Bit Name	Initial Value	R/W	Description
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	EPVB	0	R/W	Erase Pulse Verify Block Selects the erasing program. 0: On-chip erasing program is not selected [Clear condition] When transfer is completed 1: On-chip erasing program is selected

programming/erasing of flash memory. Before setting the SCO bit to 1 in order to download on-chip program or executing the downloaded programming/erasing program, these processing cannot be executed if the key code is not written.

Bit	Bit Name	Initial Value	R/W	Description
7	K7	0	R/W	Key Code
6	K6	0	R/W	Only when H'A5 is written, writing to the SCO bit is valid.
5	K5	0	R/W	When the 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.
4	K4	0	R/W	
3	K3	0	R/W	
2	K2	0	R/W	Only when H'5A is written, programming/erasing can be executed. Even if the on-chip programming/erasing program is executed, the flash memory cannot be programmed or erased when the value other than H'5A is written to FKEY.
1	K1	0	R/W	
0	K0	0	R/W	
				H'A5: Writing to the SCO bit is enabled (The SCO bit cannot be set by the value other than H'A5.)
				H'5A: Programming/erasing is enabled (The value other than H'A5 is in software protection state.)
				H'00: Initial value



Bit	Bit Name	Initial Value	R/W	Description
7	MS7	0/1*	R/W	<b>MAT Select</b>
6	MS6	0	R/W	<p>These bits are in user-MAT selection state when the value other than H'AA is written and in user-boot-MAT selection state when H'AA is written.</p> <p>The MAT is switched by writing the value in FMATS.</p> <p>When the MAT is switched, follow section 20.6, 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 PROM mode.)</p> <p>H'AA: The user boot MAT is selected (in user-MAT selection state when the value of these bits are other than H'AA)</p> <p>Initial value when these bits are initiated in user boot mode.</p> <p>H'00: Initial value when these bits are initiated in a mode except for user boot mode (in user-MAT selection state)</p> <p>[Programmable condition] These bits are in the execution state in the on-chip RAM.</p>
5	MS5	0/1*	R/W	
4	MS4	0	R/W	
3	MS3	0/1*	R/W	
2	MS2	0	R/W	
1	MS1	0/1*	R/W	
0	MS0	0	R/W	

Note: \* Set to 1 when in user boot mode, otherwise set to 0.

must be specified before setting the SCO bit in FCCS to 1.

Bit	Bit Name	Initial Value	R/W	Description
7	TDER	0	R/W	<p>Transfer Destination Address Setting Error</p> <p>This bit is set to 1 when the address specified by bits TDA6 to TDA0, which is the start address to download an on-chip program, is over the range. Whether or not the range specified by bits TDA6 to TDA0 is within the range of H'00 to H'03 is determined when an on-chip program is downloaded by setting the SCO bit in FCCS. Make sure that this bit is cleared to 0 before setting the SCO bit to 1 and the value specified by TDA6 to TDA0 is within the range of H'00 to H'03.</p> <p>0: The value specified by bits TDA6 to TDA0 is within the range.</p> <p>1: The value specified by is TDA6 to TDA0 is over the range (H'04 to H'FF) and the download is stopped.</p>
6	TDA6	0	R/W	Transfer Destination Address
5	TDA5	0	R/W	Specifies the start address to download an on-chip program. H'00 to H'03 can be specified meaning that the start address in the on-chip RAM space can be specified in units of 4 kbytes.
4	TDA4	0	R/W	
3	TDA3	0	R/W	<p>H'00: H'FF9000 is specified as a start address to download an on-chip program.</p> <p>H'01: H'FFA000 is specified as a start address to download an on-chip program.</p> <p>H'02: H'FFB000 is specified as a start address to download an on-chip program.</p> <p>H'03: H'FF8000 is specified as a start address to download an on-chip program.</p> <p>H'04 to H'07: Setting prohibited. Specifying this value sets the TDRE bit to 1 and stops the download.</p>
2	TDA2	0	R/W	
1	TDA1	0	R/W	
0	TDA0	0	R/W	

The programming/erasing interface parameter specifies the operating frequency, 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 (ER0 and ER1) or the on-chip RAM area. The initial value is undefined at a power-on reset or in hardware standby mode.

When download, initialization, or on-chip program is executed, registers of the CPU except for R0L are stored. The return value of the processing result is written in ER0, ER1. Since the stack area is used for storing the registers except for ER0, ER1, the stack area must be saved at the processing start. (A maximum size of a stack area to be used is 128 bytes.)

The programming/erasing interface parameter is 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 20.4. The meaning of the bits in FPFER varies in each processing program: initialization, programming, or erasure. For details, see descriptions of FPFER for each process.

Name of Parameter	Abbreviation	Download	Initialization	Programming	Erase	R/W	Initial Value	Allocation
Download pass and fail result	DPFR	○	—	—	—	R/W	Undefined	On-chip RAM*
Flash pass and fail result	FPFR	○	○	○	○	R/W	Undefined	R0L of CPU
Flash programming/erasing frequency control	FPEFEQ	—	○	—	—	R/W	Undefined	ER0 of CPU
Flash user branch address set	FUBRA	—	○	—	—	R/W	Undefined	ER1 of CPU
Flash multipurpose address area	FMPAR	—	—	○	—	R/W	Undefined	ER1 of CPU
Flash multipurpose data destination area	FMPDR	—	—	○	—	R/W	Undefined	ER0 of CPU
Flash erase block select	FEBS	—	—	—	○	R/W	Undefined	R0L of CPU

Note: \* A single byte of the start address to download an on-chip program, which is 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 128-kbyte area starting from the address specified by FTDAR.

Download control is set in the program/erase interface register, and the return value is passed using the DPFR parameter.

### (a) Download pass/fail result parameter (DPFR: single byte of start address 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 writing the single byte of the start address specified by FTDAR to the value other than the return value of download (for example, H'FF) before the download start (before setting the SCO bit to 1).

7 to 3	—	—	—	Unused Return 0
2	SS	—	R/W	<p>Source Select Error Detect</p> <p>Only one type for the on-chip program which can be downloaded can be specified. When more than two types of the program are selected, the program is not selected, or the program is selected without mapping, error is occurred.</p> <p>0: Download program can be selected normally 1: Download error is occurred (multi-selection or program which is not mapped is selected)</p>
1	FK	—	R/W	<p>Flash Key Register Error Detect (FK)</p> <p>Returns the check result whether the value of FKEY is set to H'A5.</p> <p>0: KEY setting is normal (FKEY = H'A5) 1: Setting value of FKEY becomes error (FKEY = value other than H'A5)</p>
0	SF	—	R/W	<p>Success/Fail</p> <p>Returns the result whether download is ended normally or not. The determination result whether program that is downloaded to the on-chip RAM is read back and then transferred to the on-chip RAM is returned.</p> <p>0: Downloading on-chip program is ended normally (no error) 1: Downloading on-chip program is ended abnormally (error occurs)</p>

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.

The initial program is set as a parameter of the programming/erasing program which has downloaded these settings.

(a) Flash programming/erasing frequency parameter (FPEFEQ: general register ER0 of CPU)

This parameter sets the operating frequency of the CPU and enables the user branch function.

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	FUBE15 to 0	—	R/W	Set to H'AAFF if the user branch function is enabled by the flash user branch enable bit. Otherwise, set to H'0000.
15 to 0	F15 to F0	—	R/W	<p>Frequency Set</p> <p>Set the operating frequency of the CPU. The setting value must be calculated as the following methods.</p> <ol style="list-style-type: none"> <li>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.</li> <li>2. The value multiplied by 100 is converted to the binary digit and is written to the FPEFEQ parameter (general register ER0).</li> </ol> <p>For example, when the operating frequency of the CPU is 34.000 MHz, the value is as follows.</p> <ol style="list-style-type: none"> <li>1. The number to three decimal places of 34.000 is rounded and the value is thus 34.00.</li> <li>2. The formula that <math>34.00 \times 100 = 3400</math> is converted to the binary digit and B'0000,1101,0100,1000 (H'0D48) is set to R0.</li> </ol>

This parameter sets the user branch destination address. A specified user program can be used to perform programming or erasing of processing units of predetermined size. When using the user branch function, set the flash user branch enable bits in FPEFEQ to H'AAFF in addition to the settings in this register.

### Bits 31 to 0: User branch destination address (UA31 to UA0)

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	UA31 to UA0	—	R/W	<p>User branch destination address</p> <p>When no user branching is required, set address 0 (H'00000000). A user branch destination must be within the RAM space other than the area occupied by the internal program transferred, or the external bus space.</p> <p>Proceed with caution to avoid branching to an area without execution code, which would cause a runaway, and avoid corrupting the internal program area or a stack area. In the event of a program runaway, flash memory values are not guaranteed.</p> <p>During user-branched processing, do not download, initialize, or invoke any programming/erase program routines of the internal program. Programming/erasing subsequent to the user branch routine cannot be otherwise guaranteed. In addition, do not modify the prepared data to be written. Likewise, do not rewrite the programming/erase interface register during user branched processing. After completing the user-branch processing, return to the programming/erase program by the RTS instructions.</p>

This parameter indicates the return value of the initialization result.

Bit	Bit Name	Initial Value	R/W	Description
7 to 2	—	—	—	Unused Return 0
1	FQ	—	R/W	Frequency Error Detect Returns the check result whether the specified operating frequency of the CPU is in the range of the supported operating frequency. 0: Setting of operating frequency is normal 1: Setting of operating frequency is abnormal
0	SF	—	R/W	Success/Fail Indicates whether initialization is completed normally. 0: Initialization is ended normally (no error) 1: Initialization is 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 must be stored in a general register ER1. This parameter is called as FMPAR (flash multipurpose address area parameter). Since the program data is always in units of 128 bytes, the lower eight bits (A7 to A0) 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 in other than the flash memory space.

When data to be programmed does not satisfy 128 bytes, the 128-byte program data must be prepared by filling with the dummy code H'FF.

The start address of the area in which the prepared program data is stored must be stored in a general register ER0. This parameter is called as FMPDR (flash multipurpose data destination area parameter).

For details on the program processing procedure, see section 20.4.2, User Program Mode.



This parameter stores the start address of the programming destination on the user MAT.

When the address in the area other than 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 FPFRR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOA31 to MOA0	—	R/W	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. Therefore, the specified programming start address becomes a 128-byte boundary and MOA6 to MOA0 are always 0.

(b) Flash multipurpose data destination parameter (FMPDR: general register ER0 of CPU):

This parameter stores 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 in FPFRR.

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	MOD31 to MOD0	—	R/W	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.

This parameter indicates the return value of the program processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Returns 0.
6	MD	—	R/W	Programming Mode Related Setting Error Detect Returns the check result that the error protection state is not entered. When the error protection state is entered, 1 is written to this bit. The state can be confirmed with the FLER bit in FCCS. For conditions to enter the error protection state, see section 20.5.3, Error Protection. 0: FLER setting is normal (FLER = 0) 1: Programming cannot be performed (FLER = 1)
5	EE	—	R/W	Programming Execution Error Detect 1 is returned to this bit when the specified data could not be written because the user MAT was not erased. 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 should be performed in boot mode or PROM mode. 0: Programming has ended normally 1: Programming has ended abnormally (programming result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of the value of FKEY before the start of the programming processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other than H'5A)
3	—	—	—	Unused Returns 0.

2	WD	—	R/W	<p>Write Data Address Detect</p> <p>When the address in the flash memory area is specified as the start address of the storage destination of the program data, an error occurs.</p> <p>0: Setting of write data address is normal</p> <p>1: Setting of write data address is abnormal</p>
1	WA	—	R/W	<p>Write Address Error Detect</p> <p>When the following items are specified as the start address of the programming destination, an error occurs.</p> <ul style="list-style-type: none"> <li>• When the programming destination address in the area other than flash memory is specified</li> <li>• When the specified address is not a 128-byte boundary (the value of A6 to A0 is not H'0).</li> </ul> <p>0: Setting of programming destination address is normal</p> <p>1: Setting of programming destination address is abnormal</p>
0	SF	—	R/W	<p>Success/Fail</p> <p>Indicates whether the program processing is ended normally or not.</p> <p>0: Programming is ended normally (no error)</p> <p>1: Programming is ended abnormally (error occurs)</p>

When flash memory is erased, the erase-block number on the user MA1 must be passed to the erasing program which is downloaded. This is set to the FEBS parameter (general register ER0).

One block is specified from the block number 0 to 15.

For details on the erasing processing procedure, see section 20.4.2, User Program Mode.

(a) Flash erase block select parameter (FEBS: general register ER0 of CPU)

This parameter specifies the erase-block number.

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	—	Unused These bits should be cleared to H'0.
7 to 0	Ebn7 to 0	—	R/W	Erase Block Number Set an erase-block number within the range from 0 to 15. H'00 corresponds to the EB0 block and H'0F corresponds to the EB15 block. An error occurs if a number outside the range from H'00 to H'0F* is set Note: * For the H8S/2362F, H8S/2361F, and H8S/2360F choose a setting value within the range from H'00 to H'0B. For the H8S/2364F, choose a setting value within the range from H'00 to H'0D.

This parameter returns value of the erasing processing result.

Bit	Bit Name	Initial Value	R/W	Description
7	—	—	—	Unused Return 0.
6	MD	—	—	Programming Mode Related Setting Error Detect Returns the check result of whether the error protection state is entered. The error protection state is entered, 1 is written to this bit. The error protection state can be confirmed with the FLER bit in FCCS. For conditions to enter the error protection state, see section 20.5.3, Error Protection. 0: FLER setting is normal (FLER = 0) 1: FLER = 1 and programming cannot be performed
5	EE	—	R/W	Erase Execution Error Detect 1 is returned to this bit when the user MAT could not be erased or when flash-memory related register settings are partially changed. 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. Erasing of the user boot MAT should be performed in boot mode or PROM mode. 0: Erasure has ended normally 1: Erasure has ended abnormally (erasure result is not guaranteed)
4	FK	—	R/W	Flash Key Register Error Detect Returns the check result of FKEY value before start of the erasing processing. 0: FKEY setting is normal (FKEY = H'5A) 1: FKEY setting is error (FKEY = value other than H'5A)

3	EB	—	R/W	<p>Erase Block Select Error Detect</p> <p>Returns the check result whether the specified erase-block number is in the block range of the user MAT.</p> <p>0: Setting of erase-block number is normal</p> <p>1: Setting of erase-block number is abnormal</p>
2, 1	—	—	—	<p>Unused</p> <p>Return 0.</p>
0	SF	—	R/W	<p>Success/Fail</p> <p>Indicates whether the erasing processing is ended normally or not.</p> <p>0: Erasure is ended normally (no error)</p> <p>1: Erasure is ended abnormally (error occurs)</p>

FVACR modifies the space from which the NMI interrupts is read. Normally the vector table data is read from the address spaces from H'00001C to H'00001F. However, the vector table can be read from the on-chip RAM by the FVACR setting. FVACR is initialized to H'00 at a power-on reset or in hardware standby mode.

All interrupts including NMI must be prohibited in the programming/erasing processing or during downloading on-chip program. When the NMI interrupt is necessary, FVACR must be set and the interrupt exception processing routine must be set in the on-chip RAM space or in the external space.

Bit	Bit Name	Initial Value	R/W	Description
7	FVCHGE	0	R/W	<p>Vector Switch Function Valid</p> <p>Selects whether the function for modifying the space from which the vector table data is read is valid or invalid. When FVCHGE = 1, the vector table data can be read from the on-chip RAM space.</p> <p>0: Function for modifying the space from which the vector table data is read is invalid (Initial value)</p> <p>1: Function for modifying the space from which the vector table data is read is valid</p>
6 to 4	—	All 0	R	<p>Reserved</p> <p>These bits are always read as 0. The write value should always be 0.</p>
3	FVSEL3	0	R/W	Interrupt Source Select
2	FVSEL2	0	R/W	The vector table of the NMI interrupt processing can be in the on-chip RAM space by setting this bit.
1	FVSEL1	0	R/W	0000: Vector table data is in area 0 (H'00001C to H'00001F)
0	FVSEL0	0	R/W	<p>0001: Setting prohibited</p> <p>001×: Setting prohibited</p> <p>01××: Setting prohibited</p> <p>1000: Vector table data is in the on-chip RAM space (H'FFA01C to H'FFA01F)</p> <p>1001: Setting prohibited</p> <p>101×: Setting prohibited</p> <p>11××: Setting prohibited</p>

Legend:

×: Don't care

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 of the pin setting for entering each mode, see table 20.5. User programming mode can be used by setting the control bit (FLSHE) by software. For details of the state transition of each mode for flash memory, see figure 20.2.

**Table 20.5 Setting On-Board Programming Mode**

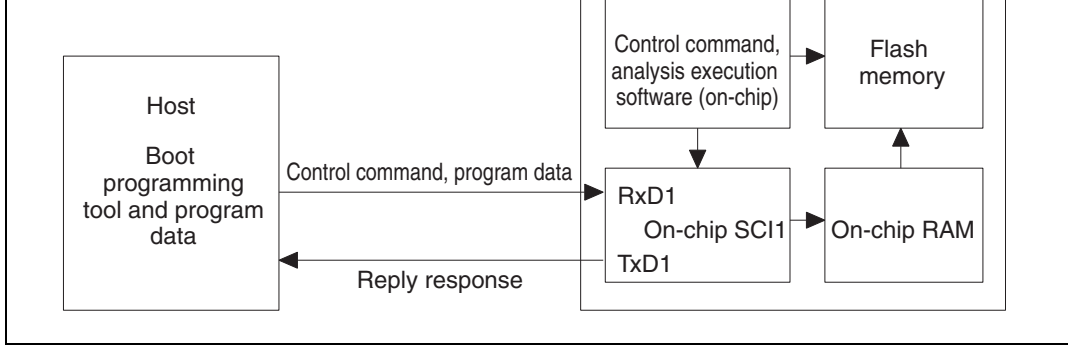
Mode Setting		MD2	MD1	MD0
Boot mode	Advanced mode	0	1	1
User boot mode	Advanced mode	1	0	1

### 20.4.1 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 20.6. For details on the pin setting in boot mode, see table 20.5. The NMI and other interrupts are ignored in boot mode. However, the NMI and other interrupts should be disabled in the user system.



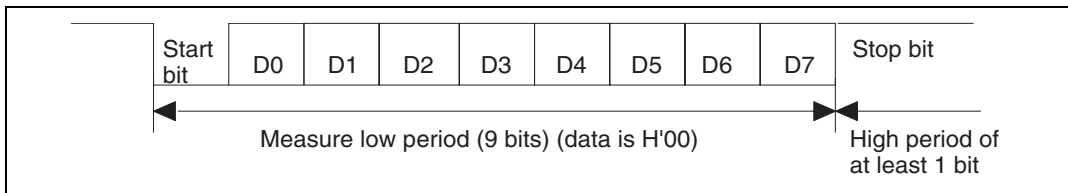


**Figure 20.6 System Configuration in Boot Mode**

**(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 by 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 20.6. Boot mode must be initiated in the range of this system clock.



**Figure 20.7 Automatic-Bit-Rate Adjustment Operation of SCI**

Bit rate of Host	System Clock Frequency
9,600 bps	8 to 25 MHz
19,200 bps	8 to 25 MHz

## (2) State Transition Diagram

The overview of the state transition diagram after boot mode is initiated is shown in figure 20.8.

### 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

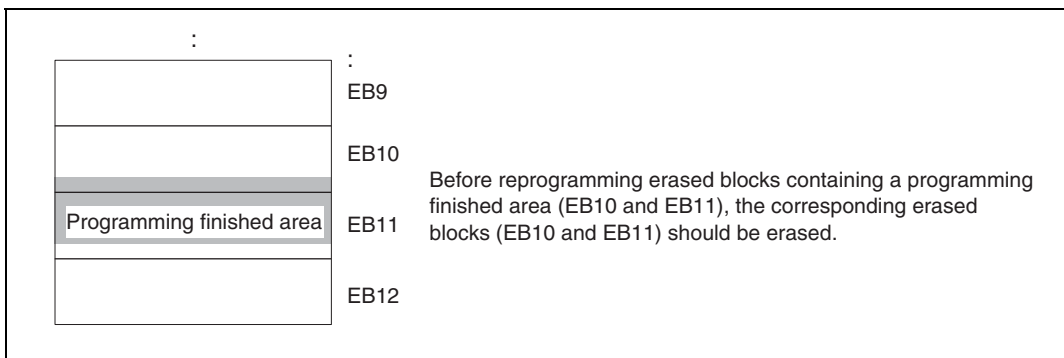
For inquiries about user-MAT size and configuration, MAT start address, and support state, the required information is transmitted to the host.

### 3. Automatic erasure of all user MAT and user boot MAT

After inquiries have finished, all user MAT and user boot MAT are automatically erased.

### 4. Waiting for programming/erasing command

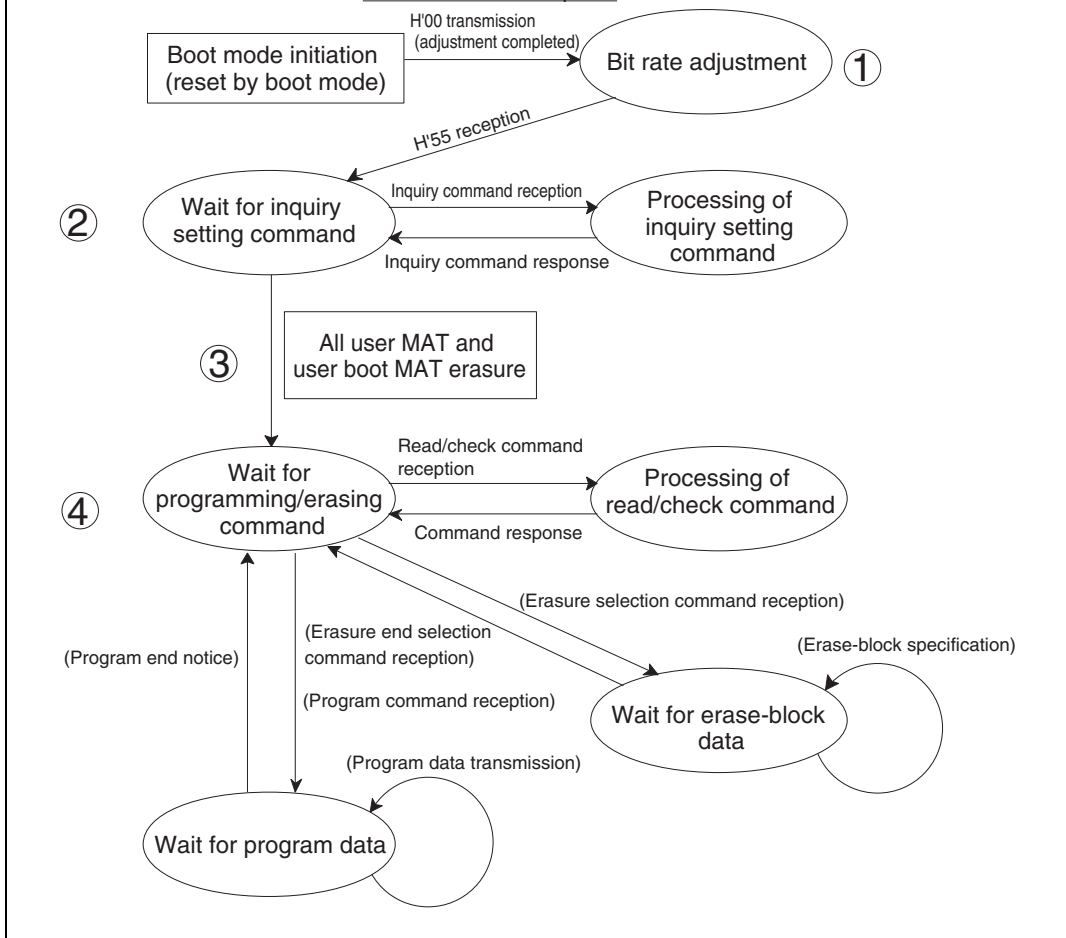
— When the program preparation notice is received, the state for waiting program data is entered. The programming start address and program data must be transmitted following the programming command. When programming is finished, the programming start address must be set to H'FFFFFFFF and transmitted. Then the state for waiting program data is returned to the state of programming/erasing command wait. Before reprogramming erased blocks containing a programming finished area for which the programming finished command has been issued, make sure to erase the corresponding erased blocks.



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 used when the specified block is programmed without a reset start 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 programmed data after all user MAT/user boot MAT has automatically been erased.



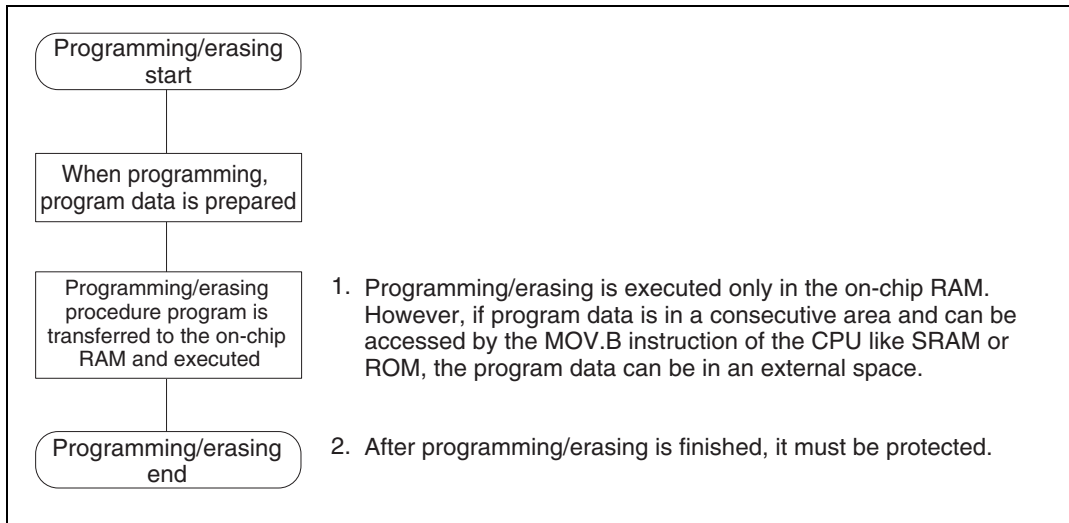
**Figure 20.8 Overview of Boot Mode State Transition Diagram**

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 20.9.

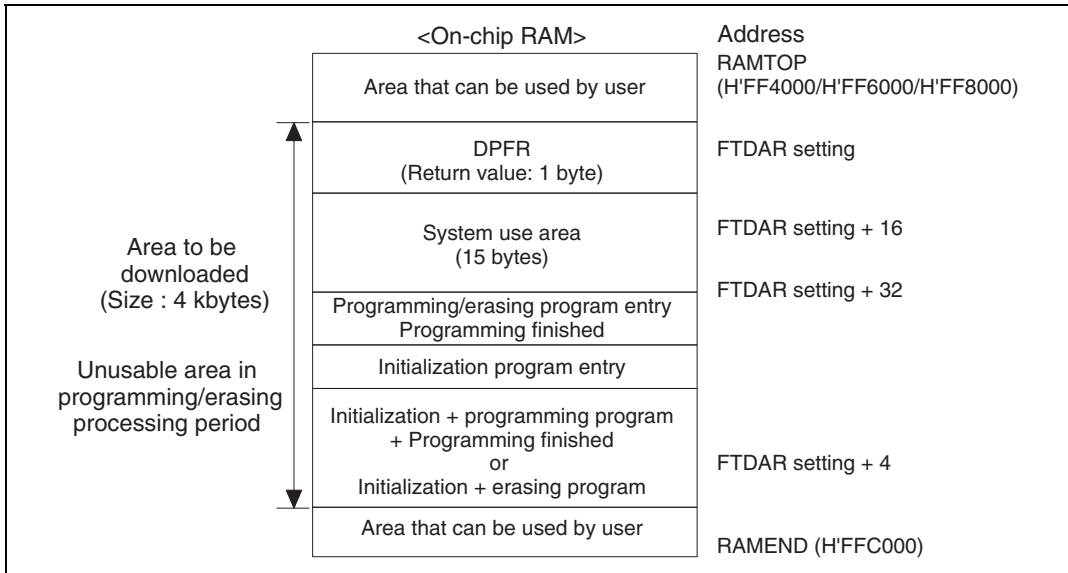
High voltage is applied to internal flash memory during the programming/erasing processing. Therefore, transition to reset or hardware standby must not be executed. Doing so may cause damage or destroy flash memory. If reset is executed accidentally, reset must be released after the reset input period, which is longer than normal 100  $\mu$ s.



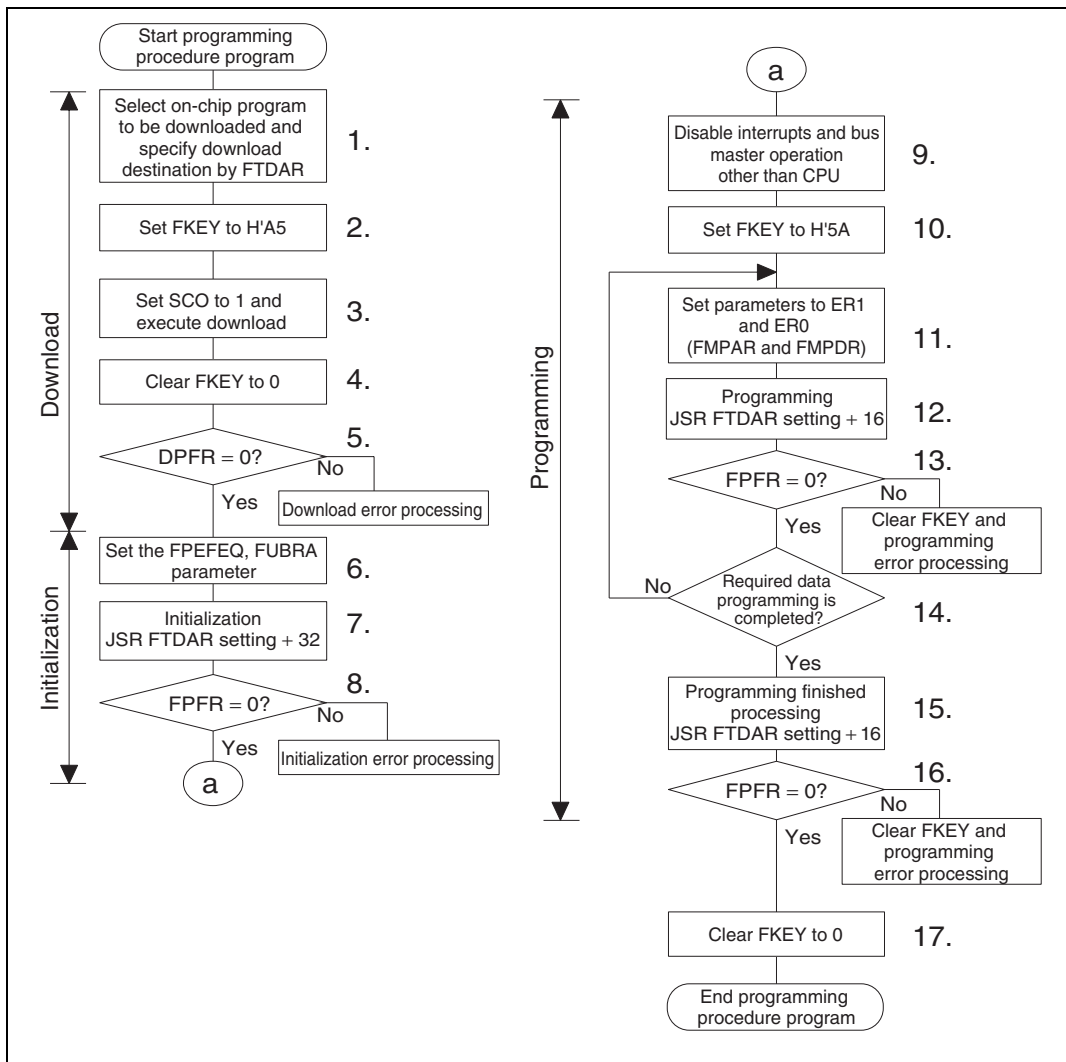
**Figure 20.9 Programming/Erasing Overview Flow**

Parts of the procedure program that are made by the user, like download request, programming/erasing procedure, and determination of the result, must be executed in the on-chip RAM. The on-chip program that is to be downloaded is all in the on-chip RAM. Note that area in the on-chip RAM must be controlled so that these parts do not overlap.

Figure 20.10 shows the program area to be downloaded.



**Figure 20.10 RAM Map when Programming/Erasing Is Executed**



**Figure 20.11 Programming Procedure**

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 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 is not executed, erasing is executed 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 dummy data to be added is H'FF, the program processing period can be shortened.

1. Select the on-chip program to be downloaded and specify a download destination

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 SS bit in DPFR. The start address of a download destination is specified by FTDAR.

2. Program H'A5 in FKEY

If H'A5 is not written to FKEY for protection, 1 cannot be written to the SCO bit for download request.

3. 1 is written to the SCO bit of FCCS and then download is executed.

To write 1 to the SCO bit, the following conditions must be satisfied.

- 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 the SCO bit is returned to the user procedure program, the SCO 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 DPFR. Before the SCO bit is set to 1, incorrect determination must be prevented by setting the one byte of the start address (to be used as DPFR) 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. 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.



- 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, the user procedure program is returned.

The notes on download are as follows.

- In the download processing, the values of CPU general registers other than ERO and ER1 are retained.
  - In the download processing, any interrupts are not accepted. However, interrupt requests other than the NMI are held. Therefore, when the user procedure program is returned, the NMI interrupts occur. NMI requests are discarded if the FVACR value is H'00. However, if H'88 has been written to FVACR, they are held and the interrupts are generated when processing returns to the user procedure program.
  - The sources of the interrupt requests from the on-chip module and at the falling edge of the IRQ are held during downloading. The refresh cycles for the DRAM can be inserted.
  - When the level-detection interrupt requests are to be held, interrupts must be input until the download is ended.
  - When hardware standby mode is entered during download processing, the normal download cannot be guaranteed in the on-chip RAM. Therefore, download must be executed again.
  - Since a stack area of a maximum 128-byte is used, the area must be allocated before setting the SCO bit to 1.
  - If a flash memory access by the DMAC, DTC, or  $\overline{\text{BREQ}}$  signal is requested during downloading, the operation cannot be guaranteed. Therefore, an access request by the DMAC, DTC, or  $\overline{\text{BREQ}}$  signal must not be generated.
4. FKEY is cleared to H'00 for protection.
  5. The value of the DPFR parameter must be checked and the download result must be confirmed.
    - 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.

selection and FKEY setting were normal, respectively.

6. The operating frequency and user branch destination are set to the FPEFEQ and FUBRA parameters for initialization.
  - The current frequency of the CPU clock is set to the FPEFEQ parameter (general register ER0).

The allowable setting range for the FPEFEQ parameter is 8 MHz to 34 MHz. When the frequency is set to 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 20.3.2 (2) (a), Flash programming/erasing frequency parameter (FPEFEQ: general register ER0 of CPU).
  - Set the user branch destination address as the FUBRA parameter (general register ER1) and the user branch enable bits (FUBE15 to FUBE0) as the FPEFEQ parameter (general register ER0). Set FUBRA and FUBE15 to FUBE0 to 0 if the user branch function is not required.

Do use programmable user MAT as the user branch destination. Also, do not use an area containing a downloaded internal program as the user branch destination. After user branch processing completes, use the RTS instruction to return to programming processing. For details, see the descriptions in 21.3.2 (2) (a), Flash programming/erasing frequency parameter (FPEFEQ: general register ER0 of CPU), and 21.3.2 (2) (b), Flash user branch address setting parameter (FUBRA: general register ER1 of CPU).

## 7. Initialization

When a programming program is downloaded, the initialization program is also downloaded to the on-chip RAM. There is an entry point of the initialization program in the area from the start address specified by FTDAR + 32 bytes of the on-chip RAM. The subroutine is called and initialization is executed by using the following steps.

MOV .L	DLTOP+32 , ER2 ;	Set entry address to ER2
JSR	@ER2 ;	Call initialization routine
NOP		

- The general registers other than ER0, ER1 are held in the initialization program.
  - R0L is a return value of the FPFR parameter.
  - Since the stack area is used in the initialization program, a stack area of a maximum 128 bytes must be allocated in RAM.
  - Interrupts can be accepted during the execution of the initialization program. The program storage area and stack area in the on-chip RAM and register values must not be destroyed.
8. The return value in the initialization program, FPFR (general register R0L) is determined.

interrupts occur or the bus mastership is moved to other than the CPU during this time, the voltage for more than the specified time will be applied and flash memory may be damaged. Therefore, interrupts, movement of bus mastership to other than the CPU (DMAC, DTC, or  $\overline{I}$  set to B'1 in interrupt control mode 0 or bits 2 to 0 (I2 to I0) in the extend control register of the CPU should be set to B'111 in interrupt control mode 2. Then interrupts other than NMI are held and are not executed.

The NMI interrupts must be masked within the user system.

The interrupts that are held must be executed after all program processing.

When the bus mastership is moved to other than the CPU by the DMAC, DTC, or  $\overline{BREQ}$  signal or DRAM refresh cycles are entered, the error protection state is entered. Therefore, taking bus mastership by the DMAC, DTC, or  $\overline{BREQ}$  signal is prohibited.

10. FKEY must be set to H'5A and the user MAT must be prepared for programming.

11. 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 ER1. The start address of the program data area (FMPDR) is set to general register ER0.

— Example of the FMPAR setting

FMPAR specifies the programming destination 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 FPFPR. Since the unit is 128 bytes, the lower eight bits (A7 to A0) must be H'00 or H'80 as the boundary of 128 bytes.

— Example of the FMPDR setting

When the storage destination of the program data is flash memory, even if the program execution routine is executed, programming is not executed and an error is returned to the FPFPR parameter. In this case, the program data must be transferred to the on-chip RAM and then programming must be executed.

12. Programming

There is an entry point of the programming program in the area from the start address specified by FTDAR + 16 bytes of the on-chip RAM. The subroutine is called and programming is executed by using the following steps.

MOV .L	#DLTOP+16 , ER2 ;	Set entry address to ER2
JSR	@ER2 ;	Call programming routine
NOF		

— Since the stack area is used in the programming program, a stack area of a maximum 128 bytes must be allocated in RAM

13. The return value in the programming program, FPFR (general register R0L) is determined.

14. 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 12 to 14. 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.

15. Execution of Programming Finished Processing

The entry point of the programming library is in the area beginning at the download destination start address specified by FTDAR plus 16 bytes. Subroutine calls should therefore be performed as follows.

MOV.L	#H'F0F0F0F0, ER0;	
MOV.L	#H'0F0F0F0F, ER1;	
MOV.L	#DLTOP+16, ER2;	Set entry address to ER2
JSR	@ER2;	Call programming finished routine

— Data is stored in a general register other than ER0, ER1 by the programming finished program.

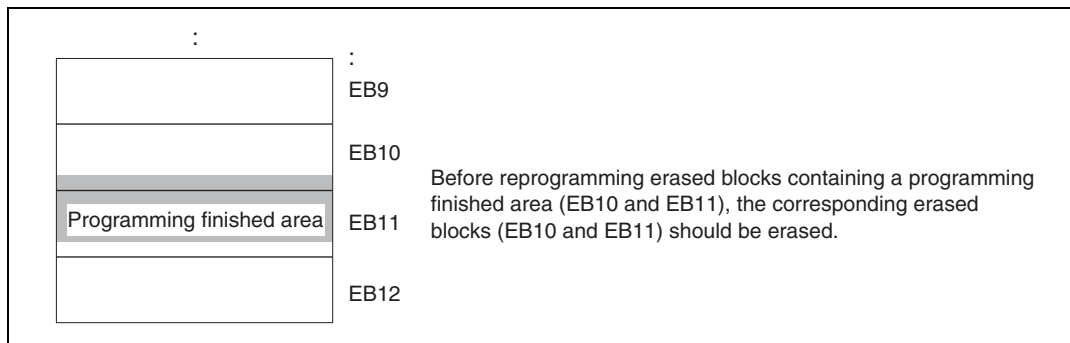
— R0L is the return value of the FPFR parameter.

— The programming finished program uses the stack area, so a maximum 128-byte stack area should be reserved in RAM beforehand.

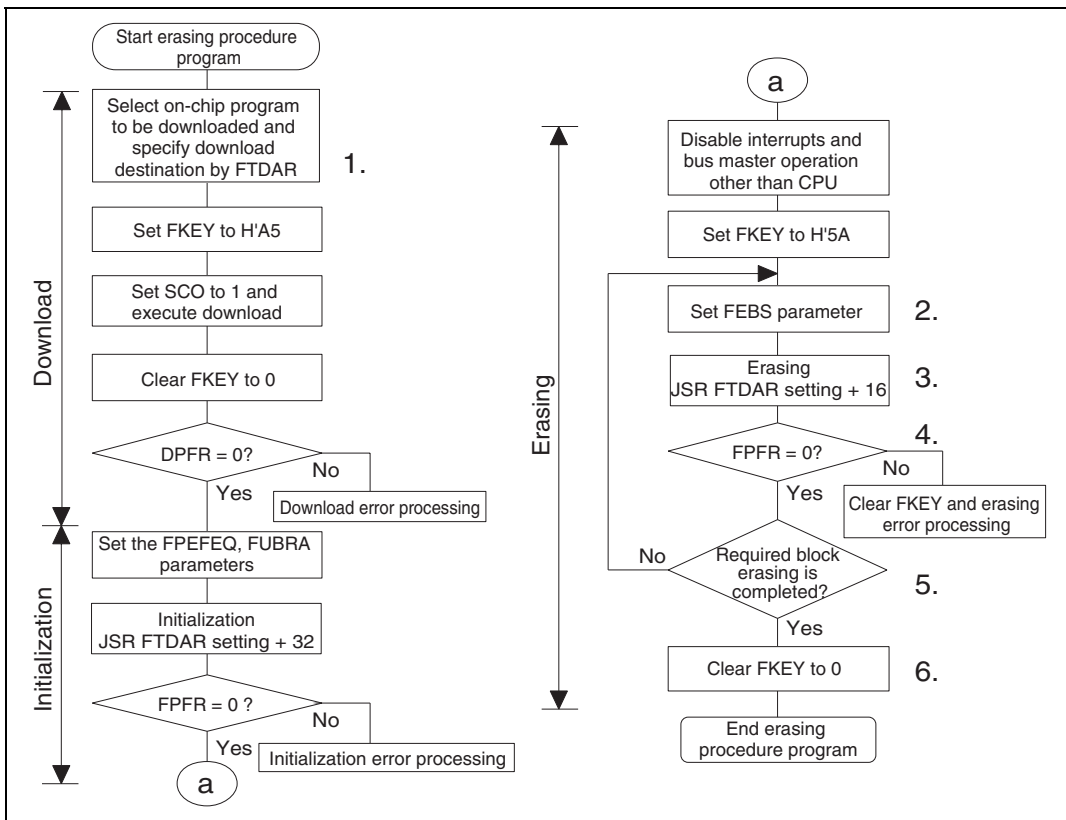
— Only perform programming finished processing once per block. Even if multiple 128-byte programming operations have been performed to the same block, programming finished processing should only be carried out once. (Do not perform programming finished processing multiple times.) If it is necessary to reprogram blocks within a previously programmed area on which programming finished processing has been performed, first erase the blocks in question and then reprogram them.

— Programming finished processing should be performed on all blocks containing areas that have been programmed after initialization processing. For example, if programming finished processing is to be carried out once after programming blocks EB1 to EB3, programming finished processing should be performed individually on EB1, EB2, and EB3.

initialization processing, internal program downloading, rewriting an area of RAM that is a download destination, or MAT switching is performed before programming finished processing, programming will not take place correctly.



16. Determine the FPFR (general-purpose register R0L) value returned by the programming program.
17. After programming finishes, clear FKEY and specify software protection.  
If this LSI is restarted by a power-on reset immediately after user MAT programming has finished, secure a reset period (period of  $\overline{RES} = 0$ ) that is at least as long as normal 100  $\mu$ s.



**Figure 20.12 Erasing Procedure**

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 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 20.4.4, Procedure Program and Storable Area for Programming Data.

For the downloaded on-chip program area, refer to figure 20.10.

processing for each block.

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 reported to the SS bit in the DPFR parameter.

Specify the start address of a 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, refer to Programming Procedure in User Program Mode in section 20.4.2 (2) Programming Procedure in User Program Mode.

The procedures after setting parameters for erasing programs are as follows:

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 ER0). 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. Erasure

Similar to as in programming, there is an entry point of the erasing program in the area from the start address of a download destination specified 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, ER2;	Set entry address to ER2
JSR	@ER2;	Call erasing routine
NOP		

- The general registers other than ER0, ER1 are held in the erasing program.
  - R0L is a return value of the FPFR parameter.
  - Since the stack area is used in the erasing program, a stack area of a maximum 128 bytes must be allocated in RAM
4. The return value in the erasing program, FPFR (general register R0L) is determined.
  5. Determine whether erasure of the necessary blocks has completed.  
If more than one block is to be erased, update the FEBS parameter and repeat steps 3 to 5.  
Blocks that have already been erased can be erased again.

secure a reset period (period of  $\overline{\text{RES}} = 0$ ) that is at least as long as normal 100  $\mu\text{s}$ .

### 20.4.3 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 20.5.

When the reset start is executed in user boot mode, the built-in check routine runs. The user MAT and user boot MAT states are checked by this check routine.

While the check routine is running, NMI and all other interrupts cannot be accepted.

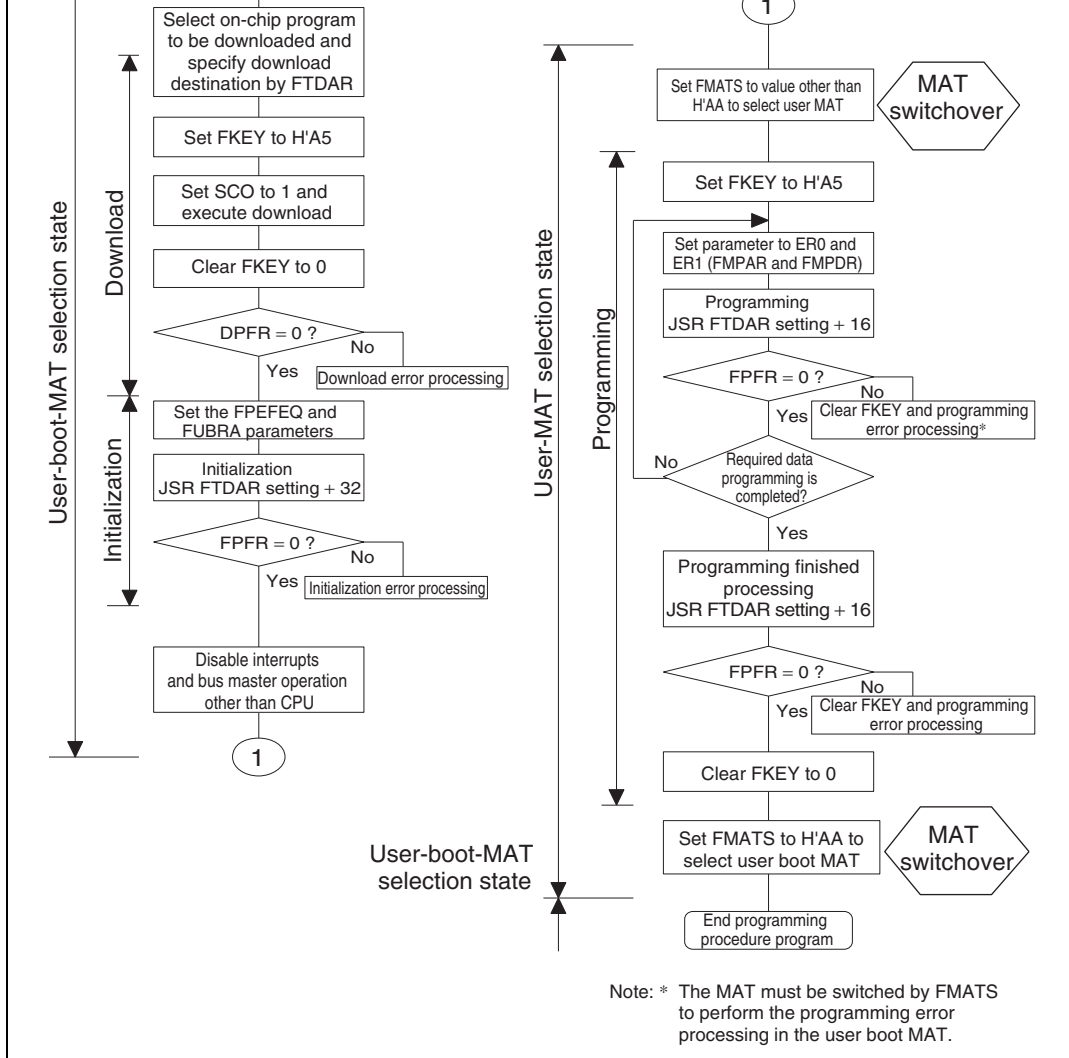
Next, processing starts from the execution start address of the reset vector in the user boot MAT. At this point, H'AA is set to FMATS because the execution MAT is the user boot MAT.

#### (2) User MAT Programming in User Boot Mode

For programming the user MAT in user boot mode, additional processing made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after programming completes.

Figure 20.13 shows the procedure for programming the user MAT in user boot mode.





**Figure 20.13 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 20.13.

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

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, and if an interrupt occurs, from which MAT the interrupt vector is read is undetermined. Perform MAT switching in accordance with the description in section 20.6, Switching between User MAT and User Boot MAT.

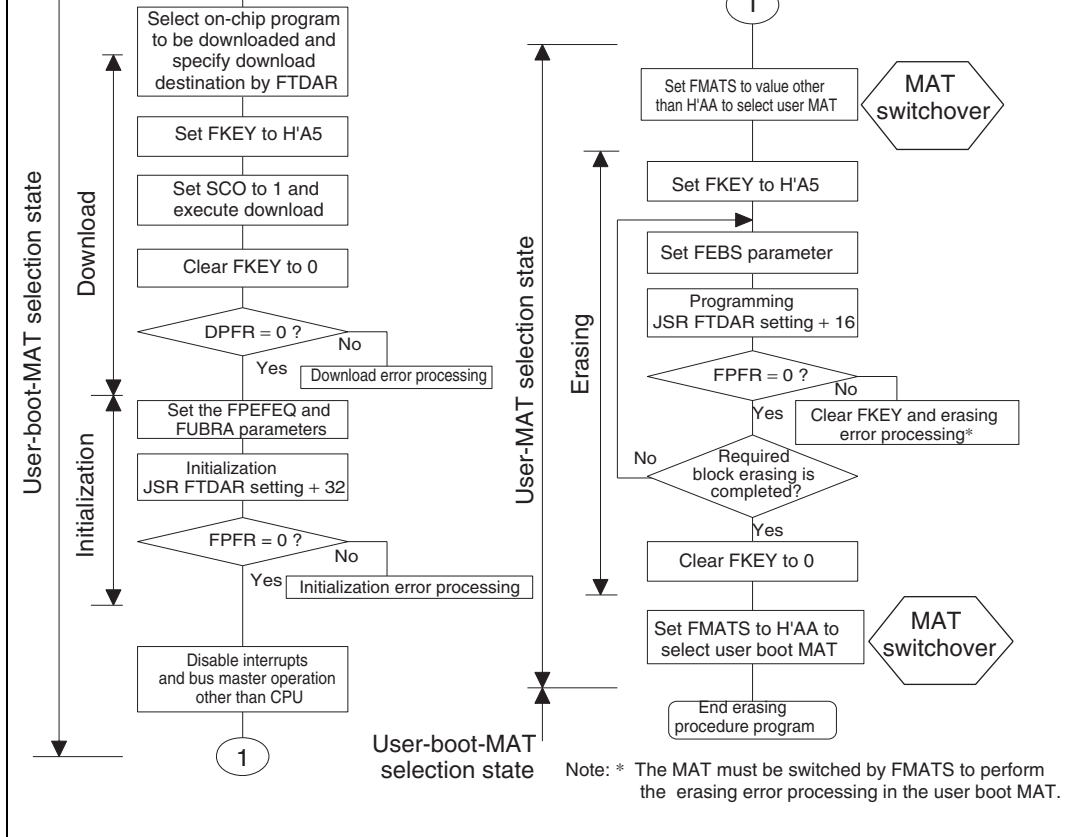
Except for MAT switching, the programming procedure is the same as that in user program mode.

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 20.4.4, Procedure Program and Storable Area for Programming Data.

### **(3) User MAT Erasing in User Boot Mode**

For erasing the user MAT in user boot mode, additional processing made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after erasing completes.

Figure 20.14 shows the procedure for erasing the user MAT in user boot mode.



**Figure 20.14 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 20.14.

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, and if an interrupt occurs, from which MAT the interrupt vector is read is undetermined. Perform MAT switching in accordance with the description in section 20.6, Switching between User MAT and User Boot MAT.

Except for MAT switching, the erasing procedure is the same as that in user program mode.

#### 20.4.4 Procedure Program and Storable Area for Programming Data

In the descriptions in the previous section, the programming/erasing procedure programs and storable areas for program data are assumed to be in the on-chip RAM. However, the program and the data can be stored in and executed from other areas, such as part of flash memory which is not to be programmed or erased, or somewhere in the external address space.

##### (1) Conditions that Apply to Programming/Erasing

1. The on-chip programming/erasing program is downloaded from the address in the on-chip RAM specified by FTDAR, therefore, this area is not available for use.
2. The on-chip programming/erasing program will use the 128 bytes as a stack. So, make sure that this area is secured.
3. Download by setting the SCO bit to 1 will lead to switching of the MAT. If, therefore, this operation is used, it should be executed from the 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 determined. When in a mode in which the external address space is not accessible, such as single-chip mode, the required procedure programs, NMI handling vector and NMI handler should be transferred to the on-chip RAM before programming/erasing of the flash memory starts.
5. The flash memory is not accessible during programming/erasing operations, therefore, the operation program is downloaded to the on-chip RAM to be executed. The NMI-handling vector and programs such as that which activate the operation program, and NMI handler should thus be stored in on-chip memory other than flash memory or the external address space.
6. After programming/erasing, the flash memory should be inhibited until FKEY is cleared. The reset state ( $\overline{\text{RES}} = 0$ ) must be in place for more than 100  $\mu\text{s}$  when the LSI mode is changed to reset on completion of a programming/erasing operation.  
Transitions to the reset state, and hardware standby mode are inhibited during programming/erasing. When the reset signal is accidentally input to the chip, 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 should be needed when programming/erasing of the user boot MAT is operated in user-boot mode. The program which switches the MATs should be executed from the on-chip RAM. See section 20.6, Switching between User MAT and User Boot MAT. Please make sure you know which MAT is selected when switching between them.

should be transferred to the on-chip RAM to place the address that FMPDR indicates in an area other than the flash memory.

In consideration of these conditions, there are three factors; operating mode, the bank structure of the user MAT, and operations.

The areas in which the programming data can be stored for execution are shown in tables.

**Table 20.7 Executable MAT**

<b>Operation</b>	<b>Initiated Mode</b>	
	<b>User Program Mode</b>	<b>User Boot Mode*</b>
Programming	Table 20.8 (1)	Table 20.8 (3)
Erasing	Table 20.8 (2)	Table 20.8 (4)

Note: \* Programming/Erasing is possible to user MATs.

Item	Storable/Executable Area		Selected MAT		
	On-chip RAM	User MAT	External Space (Expanded Mode)	User MAT	Embedded Program Storage Area
Storage Area for Program Data	○	×*	○	—	—
Operation for Selection of On-chip Program to be Downloaded	○	○	○	○	
Operation for Writing H'A5 to FKEY	○	○	○	○	
Execution of Writing SC0 = 1 to FCCS (Download)	○	×	×		○
Operation for FKEY Clear	○	○	○	○	
Determination of Download Result	○	○	○	○	
Operation for Download Error	○	○	○	○	
Operation for Settings of Initial Parameter	○	○	○	○	
Execution of Initialization	○	×	×	○	
Determination of Initialization Result	○	○	○	○	
Operation for Initialization Error	○	○	○	○	
NMI Handling Routine	○	×	○	○	
Operation for Inhibit of Interrupt	○	○	○	○	
Operation for Writing H'5A to FKEY	○	○	○	○	
Operation for Settings of Program Parameter	○	×	○	○	

Item	On-chip RAM	Flash Memory	(Expanded Mode)	User MAT	Program Storage Area
Execution of Programming	○	×	×	○	
Determination of Program Result	○	×	○	○	
Operation for Program Error	○	×	○	○	
Operation for FKEY Clear	○	×	○	○	

Note: \* Transferring the data to the on-chip RAM enables this area to be used.

Item	Storable/Executable Area		Selected MAT		Embedded Program Storage Area
	On-chip RAM	User MAT	External Space (Expanded Mode)	User MAT	
Operation for Selection of On-chip Program to be Downloaded	○	○	○	○	
Operation for Writing H'A5 to FKEY	○	○	○	○	
Execution of Writing SC0 = 1 to FCCS (Download)	○	×	×		○
Operation for FKEY Clear	○	○	○	○	
Determination of Download Result	○	○	○	○	
Operation for Download Error	○	○	○	○	
Operation for Settings of Initial Parameter	○	○	○	○	
Execution of Initialization	○	×	×	○	
Determination of Initialization Result	○	○	○	○	
Operation for Initialization Error	○	○	○	○	
NMI Handling Routine	○	×	○	○	
Operation for Inhibit of Interrupt	○	○	○	○	
Operation for Writing H'5A to FKEY	○	○	○	○	
Operation for Settings of Erasure Parameter	○	×	○	○	
Execution of Erasure	○	×	×	○	
Determination of Erasure Result	○	×	○	○	



Item	On-chip RAM	User MAT	(Expanded Mode)	User MAT	Program Storage Area
Operation for Erasure Error	○	×	○	○	
Operation for FKEY Clear	○	×	○	○	

Item	Storable/Executable Area			Selected MAT		
	On-chip RAM	User Boot MAT	External Space (Expanded Mode)	User MAT	User Boot MAT	Embedded Program Storage Area
Storage Area for Program Data	○	x <sup>*1</sup>	○	—	—	—
Operation for Selection of On-chip Program to be Downloaded	○	○	○		○	
Operation for Writing H'A5 to FKEY	○	○	○		○	
Execution of Writing SC0 = 1 to FCCS (Download)	○	x	x			○
Operation for FKEY Clear	○	○	○		○	
Determination of Download Result	○	○	○		○	
Operation for Download Error	○	○	○		○	
Operation for Settings of Initial Parameter	○	○	○		○	
Execution of Initialization	○	x	x		○	
Determination of Initialization Result	○	○	○		○	
Operation for Initialization Error	○	○	○		○	
NMI Handling Routine	○	x	○		○	
Operation for Interrupt Inhibit	○	○	○		○	
Switching MATs by FMATS	○	x	x	○		

Item	On-chip RAM	Boot MAT	(Expanded Mode)	User MAT	Boot MAT	Program Storage Area
Operation for Writing H'5A to FKEY	○	×	○	○		
Operation for Settings of Program Parameter	○	×	○	○		
Execution of Programming	○	×	×	○		
Determination of Program Result	○	×	○	○		
Operation for Program Error	○	× <sup>*2</sup>	○	○		
Operation for FKEY Clear	○	×	○	○		
Switching MATs by FMATS	○	×	×		○	

- Notes: 1. Transferring the data to the on-chip RAM enables this area to be used.  
2. Switching FMATS by a program in the on-chip RAM enables this area to be used.

Item	Storable/Executable Area			Selected MAT		Embedded Program Storage Area
	On-chip RAM	User Boot MAT	External Space (Expanded Mode)	User MAT	User Boot MAT	
Operation for Selection of On-chip Program to be Downloaded	○	○	○		○	
Operation for Writing H'A5 to FKEY	○	○	○		○	
Execution of Writing SC0 = 1 to FCCS (Download)	○	×	×			○
Operation for FKEY Clear	○	○	○		○	
Determination of Download Result	○	○	○		○	
Operation for Download Error	○	○	○		○	
Operation for Settings of Initial Parameter	○	○	○		○	
Execution of Initialization	○	×	×		○	
Determination of Initialization Result	○	○	○		○	
Operation for Initialization Error	○	○	○		○	
NMI Handling Routine	○	×	○		○	
Operation for Interrupt Inhibit	○	○	○		○	
Switching MATs by FMATS	○	×	×		○	
Operation for Writing H'5A to FKEY	○	×	○	○		

Item	On-chip RAM	Boot MAT	(Expanded Mode)	User MAT	Boot MAT	Program Storage Area
Operation for Settings of Erasure Parameter	○	×	○	○		
Execution of Erasure	○	×	×	○		
Determination of Erasure Result	○	×	○	○		
Operation for Erasure Error	○	×*	○	○		
Operation for FKEY Clear	○	×	○	○		
Switching MATs by FMATS	○	×	×	○		

Note: \* Switching FMATS by a program in the on-chip RAM enables this area to be used.

There are two kinds of flash memory program/erase protection: hardware and software protection.

### 20.5.1 Hardware Protection

Programming and erasing of flash memory is forcibly disabled or suspended by hardware protection. In this state, the downloading of a on-chip program and initialization 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 parameter FPFR.

**Table 20.9 Hardware Protection**

Item	Description	Function to be Protected	
		Download	Program/Erase
Reset/standby protection	<ul style="list-style-type: none"><li>• The program/erase interface registers are initialized in the power-on reset state (including a power-on reset by the WDT) and standby mode and the program/erase-protected state is entered.</li><li>• The reset state will not be entered by a reset using the <math>\overline{\text{RES}}</math> pin unless the RES pin is held low until oscillation has stabilized after power is initially supplied. In the case of a reset during operation, hold the RES pin low for the RES pulse width that is specified in the section on AC characteristics section. If a reset is input during programming or erasure, data values in the flash memory are not guaranteed. In this case, execute erasure and then execute program again.</li></ul>	○	○

Software protection is set up by disabling the downloading of on-chip programs for programming and erasing or by means of a key code register.

**Table 20.10 Software Protection**

Item	Description	Function to be Protected	
		Download	Program/ Erase
Protection by the SCO bit	<ul style="list-style-type: none"> <li>The program/erase-protected state is entered by clearing the SCO bit in FCCS which disables the downloading of the programming/erasing programs.</li> </ul>	○	○
Protection by the FKEY register	<ul style="list-style-type: none"> <li>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.</li> </ul>	○	○

### 20.5.3 Error Protection

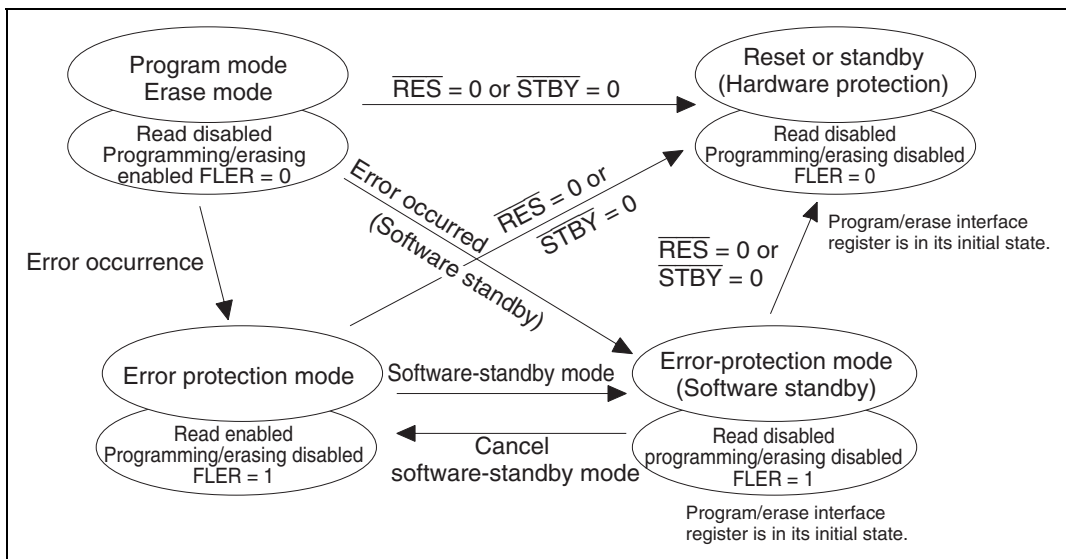
Error protection is a mechanism for aborting programming or erasure when an error occurs, in the form of the microcomputer entering runaway during programming/erasing of the flash memory or operations that are not according to the established procedures for programming/erasing. Aborting programming or 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 the FCCS register is set to 1 and the error-protection state is entered, and this aborts the programming or erasure.

1. When an interrupt such as NMI occurs during programming/erasing.
2. When the flash memory is read during programming/erasing (including a vector read or an instruction fetch).
3. When a SLEEP instruction (including software-standby mode) is executed during programming/erasing.
4. When a bus master other than the CPU such as the DMAC or DTC gets bus mastership during programming/erasing.

Error protection is cancelled only by a power-on reset or by hardware-standby mode. Note that the reset should only be released after providing a reset input over a period longer than the normal 100  $\mu$ s period. Since high voltages are applied during programming/erasing of the flash memory, some voltage may remain 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 20.15 shows transitions to and from the error-protection state.



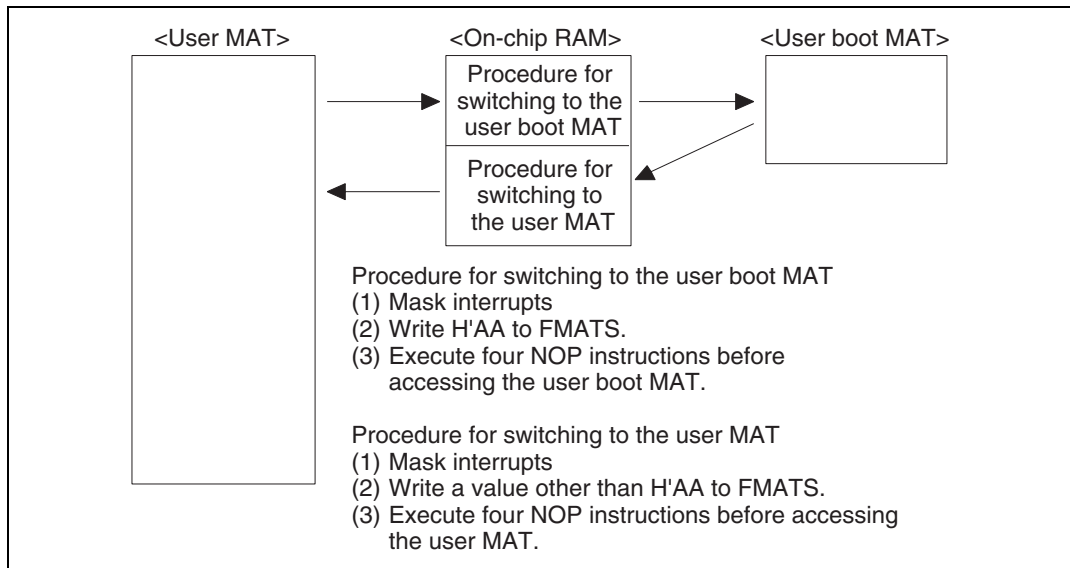
**Figure 20.15 Transitions to Error-Protection State**



It is possible to alternate 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 should take place in boot mode or PROM mode.)

1. MAT switching by FMATS should always be executed from the on-chip RAM.
2. To ensure that the MAT that has been switched to is accessible, execute four NOP instructions in the on-chip RAM immediately after writing to FMATS of the on-chip RAM (this prevents access to the flash memory during MAT switching).
3. If an interrupt has occurred during switching, there is no guarantee of which memory MAT is being accessed. Always mask the maskable interrupts before switching between MATs. In addition, configure the system so that NMI interrupts do not occur during MAT switching.
4. After the MATs have been switched, take care because the interrupt vector table will also have been switched. If interrupt processing is to be the same before and after MAT switching, transfer the interrupt-processing routines to the on-chip RAM, and use the settings of FVACR to place the interrupt-vector table in the on-chip RAM .
5. Memory sizes of the user MAT and user boot MAT are different. When accessing the user boot MAT, do not access addresses above the top of its 8-kbyte memory space. If access goes beyond the 8-kbyte space, the values read are undefined.



**Figure 20.16 Switching between the User MAT and User Boot MAT**

Along with its on-board programming mode, this LSI also has a PROM mode as a further mode for the writing and erasing of programs and data. In the PROM mode, a general-purpose PROM programmer can freely be used to write programs to the on-chip ROM. Program/erase is possible on the user MAT and user boot MAT. The PROM programmer must support Renesas microcomputers with 512-kbyte flash memory as a device type.

A status-polling system is adopted for operation in automatic program, automatic erase, and status-read modes. In the status-read mode, details of the system's internal signals are output after execution of automatic programming or automatic erasure. In the PROM mode, provide a 12-MHz input-clock signal.

## **20.8 Serial Communication Interface Specification for Boot Mode**

Initiating boot mode enables the boot program to communicate with the host by using the internal SCI. The serial communication interface specification is shown below.

### **(1) Status**

The boot program has three states.

#### **1. Bit-Rate-Adjustment State**

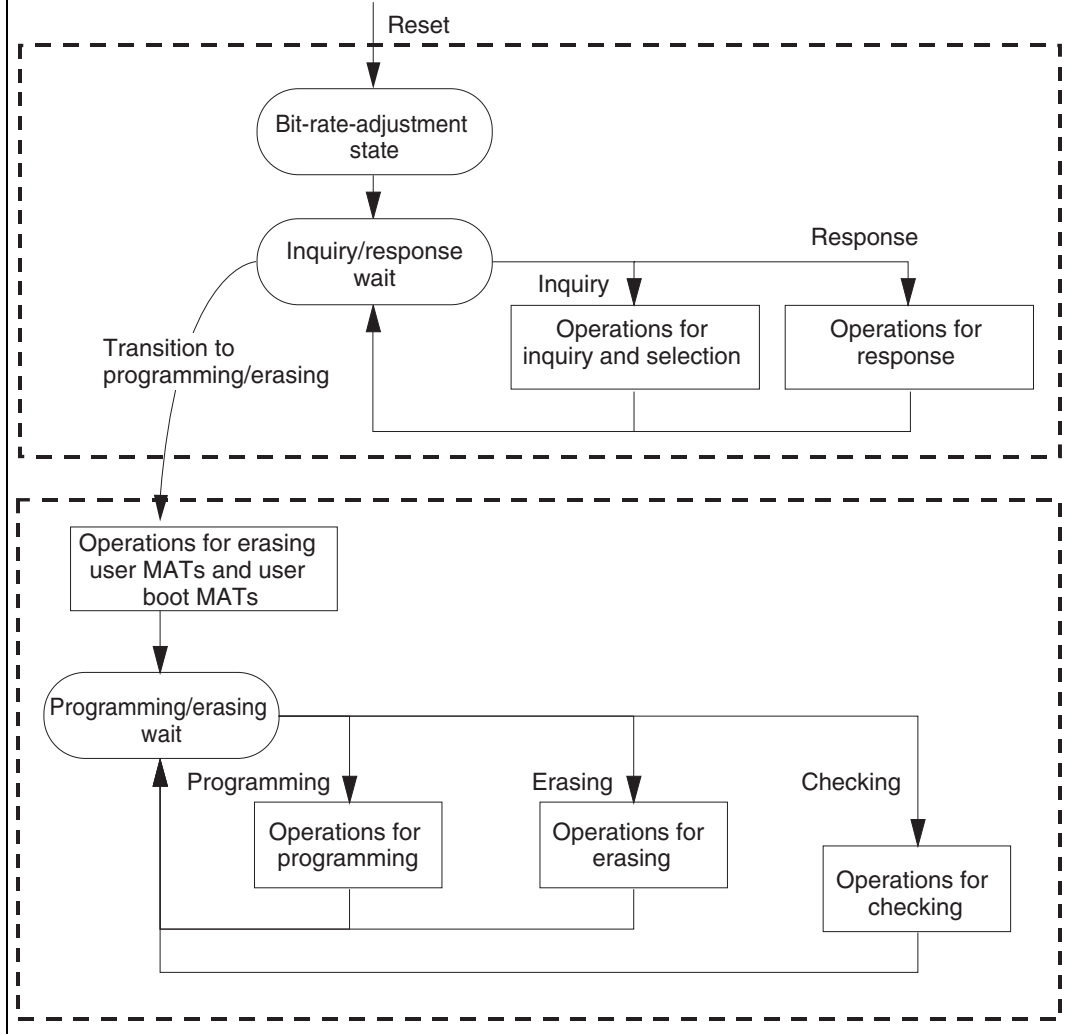
In this state, the boot program adjusts the bit rate to communicate with the host. Initiating boot mode enables starting of the boot program and entry to the bit-rate-adjustment state. The program receives the command from the host to adjust the bit rate. After adjusting the bit rate, the program enters 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 program transfers the libraries required for erasure to the RAM and erases the user MATs and user boot MATs before the transition.

#### **3. Programming/erasing state**

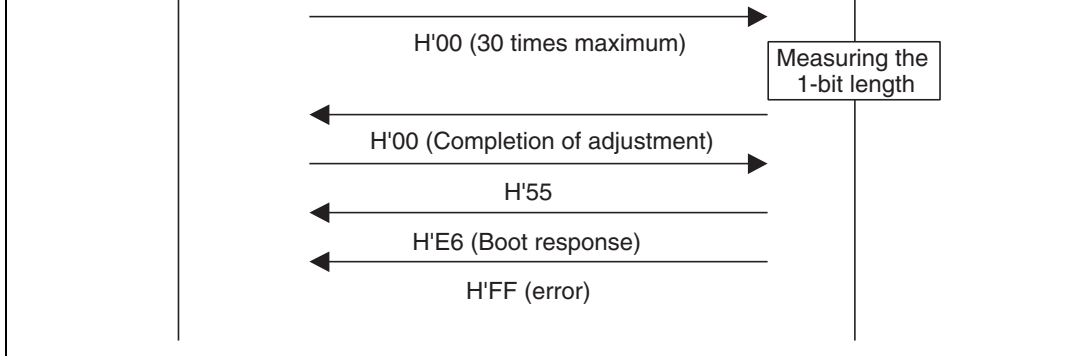
Programming and erasure by the boot program take place in this state. The boot program is made to transfer the programming/erasing programs to the RAM by commands from the host. Sum checks and blank checks are executed by sending these commands from the host.



**Figure 20.17 Boot Program States**

## (2) 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 and selection state. The bit-rate-adjustment sequence is shown in figure 20.18.



**Figure 20.18 Bit-Rate-Adjustment Sequence**

### (3) Communications Protocol

After adjustment of the bit rate, the protocol for 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 are consists of the inquiries and the 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

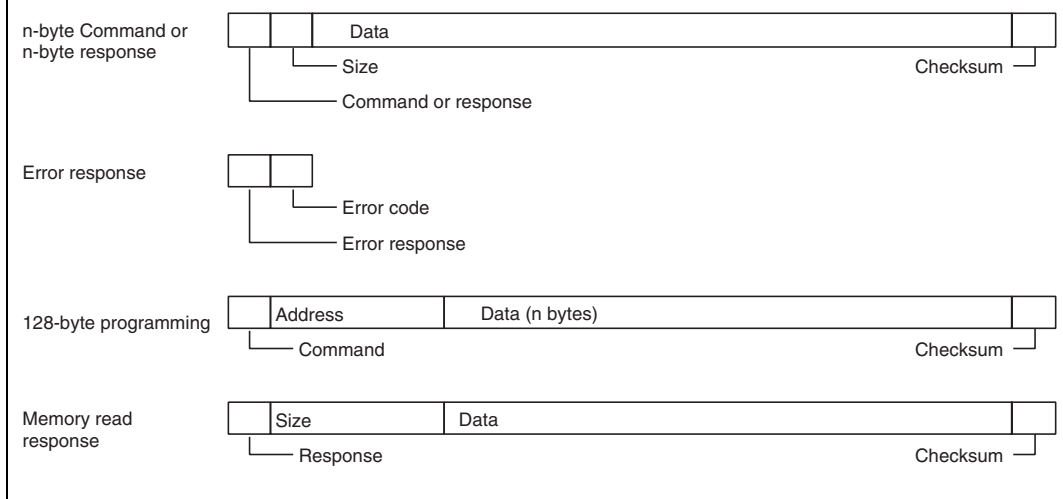
The error response is a response to inquiries. It consists of an error response and an error code and comes two bytes.

#### 4. Programming of 128 bytes

The size is not specified in commands. The size of n is indicated in response to the programming unit inquiry.

#### 5. Memory read response

This response consists of 4 bytes of data.



**Figure 20.19 Communication Protocol Format**

- **Command (one byte):** Commands including inquiries, selection, programming, erasing, and checking
- **Response (one byte):** Response to an inquiry
- **Size (one byte):** The amount of data for transmission excluding the command, amount of data, and checksum
- **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.
- **Data (n bytes):** Detailed data of a command or response
- **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 (the size is indicated in the response to the programming unit inquiry.)
- **Size (four bytes):** Four-byte response to a memory read

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.

Inquiry and selection commands are listed below.

**Table 20.11 Inquiry and Selection Commands**

<b>Command</b>	<b>Command Name</b>	<b>Description</b>
H'20	Supported Device Inquiry	Inquiry regarding device codes
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 frequency-multiplied clock types, the number of multiplication ratios, and the values of each multiple
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 operated 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 that order. These commands will certainly be needed. When two or more selection commands are sent at once, the last command will be valid.

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).

(a) Supported Device Inquiry

The boot program will return the device codes of supported devices and the product code 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, size, and checksum, that is, the amount of data contributes by the number of devices, characters, device codes and product names
- Number of devices (one byte): The number of device types supported by the boot program
- Number of characters (one byte): The number of characters in the device codes and boot program's name
- Device code (four bytes): ASCII code of the supporting product
- Product name (n bytes): Type name of the boot program in ASCII-coded characters
- 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.

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): Amount of device-code data  
This is fixed at 4
- Device code (four bytes): Device code (ASCII code) returned in response to the supported device inquiry
- 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 error, that is, the device code does not match

### (c) 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 clock modes (one byte): The number of supported clock modes  
H'00 indicates no clock mode or the device allows to read the clock mode.
- Mode (one byte): Values of the supported clock modes (i.e. H'01 means clock mode 1.)
- SUM (one byte): Checksum



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 commands.

Command 

H'11	Size	Mode	SUM
------	------	------	-----

- Command, H'11, (one byte): Selection of clock mode
- Size (one byte): Amount of data that represents the modes
- Mode (one byte): A 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: Checksum error
  - H'22: Clock mode error, that is, the clock mode does not match.

Even if the clock mode numbers are H'00 and H'01 by a clock mode inquiry, the clock mode must be selected using these respective values.

The boot program will return the supported multiplication and division ratios.

Command H'22

- Command, H'22, (one byte): Inquiry regarding multiplication ratio

Response	H'32	Size	Number of types					
	Number of multiplication ratios	Multiplication ratio	...					
	...							
	SUM							

- Response, H'32, (one byte): Response to the multiplication ratio inquiry
- Size (one byte): The amount of data that represents the number of clock sources and multiplication ratios and the multiplication ratios
- Number of types (one byte): The number of supported multiplied clock types (e.g. when there are two multiplied clock types, which are the main and peripheral clocks, the number of types will be H'02.)
- Number of multiplication ratios (one byte): The number of multiplication ratios for each type (e.g. the number of multiplication ratios to which the main clock can be set and the peripheral clock can be set.)
- Multiplication ratio (one byte)
  - Multiplication ratio: The 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: Not supported by the H8S/2368F 0.18  $\mu$ m F-ZTAT Group.

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

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): The number of bytes that represents the minimum values, maximum values, and the number of frequencies.
- Number of operating clock frequencies (one byte): The 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): The minimum value of the 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 64 MHz, it will be D'6400 and H'1900.)
- Maximum value (two bytes): Maximum value among the multiplied or divided clock frequencies.  
There are as many pairs of minimum and maximum values as there are operating clock frequencies.
- SUM (one byte): Checksum

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	
	Area-start address			Area-last address
	...			
	SUM			

- Response, H'34, (one byte): Response to user boot MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start addresses, and area-last address
- Number of Areas (one byte): The number of consecutive user boot MAT areas  
When user boot MAT areas are consecutive, the number of areas returned is H'01.
- Area-start address (four byte): Start address of the area
- Area-last address (four byte): 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

#### (h) 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 area			Last address area
	...			
	SUM			

- Response, H'35, (one byte): Response to the user MAT information inquiry
- Size (one byte): The number of bytes that represents the number of areas, area-start address and area-last address
- Number of areas (one byte): The number of consecutive user MAT areas  
When the user MAT areas are consecutive, the number of areas is H'01.
- Area-start address (four byte): Start address of the area

- SUM (one byte): Checksum

(i) Erased Block Information Inquiry

The boot program will return the number of erased blocks and their addresses.

Command 

H'26
------

- Command, H'26, (two bytes): Inquiry regarding erased block information

Response	H'36	Size	Number of blocks	
	Block start address			Block last address
	...			
	SUM			

- Response, H'36, (one byte): Response to the number of erased blocks and addresses
- Size (three byte): The number of bytes that represents the number of blocks, block-start addresses, and block-last addresses.
- Number of blocks (one byte): The number of erased blocks
- Block start address (four bytes): Start address of a block
- Block last Address (four bytes): Last address of a block  
There are as many groups of data representing the start and last addresses as there are areas.
- SUM (one byte): Checksum

(j) 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): The number of bytes that indicate the programming unit, which is fixed to 2
- Programming unit (two bytes): A unit for programming  
This is the unit for reception of programming.
- SUM (one byte): Checksum

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): The number of bytes that represents the bit rate, input frequency, number of multiplication ratios, and multiplication ratio
- Bit rate (two bytes): New bit rate  
One hundredth of the value (e.g. when the value is 19,200 bps, the bit rate is H'00C0, which is D'192.)
- Input frequency (two bytes): Frequency of the clock input to the boot program  
This is valid to the hundredths place and represents the value in MHz multiplied by 100. (e.g. when the value is 64 MHz, the input frequency is H'1900 (= D'6400).)
- Number of multiplication ratios (one byte): The number of multiplication ratios to which the device can be set.
- Multiplication ratio 1 (one byte): The value of multiplication or division ratios for the main operating frequency  
Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)  
Division ratio: The inverse of the division ratio, as a negative number (e.g. when the clock frequency is divided by two, the value of division ratio will be H'FE. H'FE = D'-2)
- Multiplication ratio 2 (one byte): The value of multiplication or division ratios for the peripheral frequency  
Multiplication ratio (one byte): The value of the multiplication ratio (e.g. when the clock frequency is multiplied by four, the multiplication ratio will be H'04.)  
(Division ratio: The inverse of the division ratio, as a negative number (E.g. when the clock is divided by two, the value of division ratio will be H'FE. H'FE = D'-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: (one byte): Error code
  - H'11: Sum checking error
  - H'24: Bit-rate selection error  
The rate is not available.
  - H'25: Error in input frequency  
This input frequency is not within the specified range.
  - H'26: Multiplication-ratio error  
The ratio does not match an available ratio.
  - H'27: Operating frequency error  
The frequency is not within the specified range.

## (5) Received Data Check

The methods for checking of received data are listed below.

### 1. 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.

### 2. 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, an input-frequency error is generated.

### 3. Operating frequency error

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 operated at the operating frequency. The expression is given below.

Operating frequency = Input frequency  $\times$  Multiplication ratio, or  
Operating frequency = Input frequency  $\div$  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.

To facilitate error checking, the value (n) of clock select (CKS) in the serial mode register (SMR), and the value(N) in the bit rate register (BRR), which are found from the peripheral operating clock frequency ( $\phi$ ) and bit rate (B), are used to calculate the error rate to ensure that it is less than 4%. If the error is more than 4%, a bit rate error is generated. The error is calculated using the following expression:

$$\text{Error (\%)} = \left\{ \left[ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{(2 \times n - 1)}} \right] - 1 \right\} \times 100$$

When the new bit rate is selectable, the rate will be set in the register after sending ACK in response. The host will send an ACK with the new bit rate for confirmation and the boot program will respond with that rate.

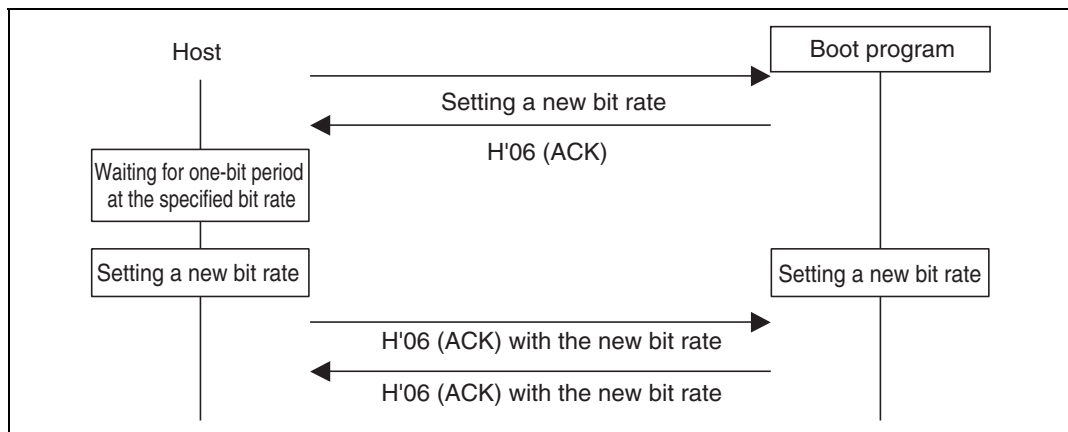
Confirmation H'06

- 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 20.20.



**Figure 20.20 New Bit-Rate Selection Sequence**



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 will enter 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. These procedure should be carried out before sending of 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 MAT and user boot MAT have been erased by the transferred erasing program.

Error Response 

H'C0	H'51
------	------

- Error response, H'C0, (one byte): Error response for user boot MAT blank check
- Error code, H'51, (one byte): Erasing error  
An error occurred and erasure was not completed.

### (7) 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 an inquiry command after the transition to programming/erasing state command, are examples.

Error Response 

H'80	H'xx
------	------

- Error response, H'80, (one byte): Command error
- Command, H'xx, (one byte): Received command

### (8) 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.

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), which are needed for a new bit-rate selection.
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 MATs information inquiry (H'24), user MATs information inquiry (H'25), erased block information inquiry (H'26), and programming unit inquiry (H'27).
8. After making inquiries and selecting a new bit rate, issue the transition to programming/erasing state command (H'40). The boot program will then enter the programming/erasing state.

A programming selection command makes the boot program select the programming method, an 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. The programming/erasing commands are listed below.

**Table 20.12 Programming/Erasing Command**

<b>Command</b>	<b>Command Name</b>	<b>Description</b>
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 the blank data of the user boot MAT
H'4D	User MAT blank check	Checks the blank data 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 status

command.

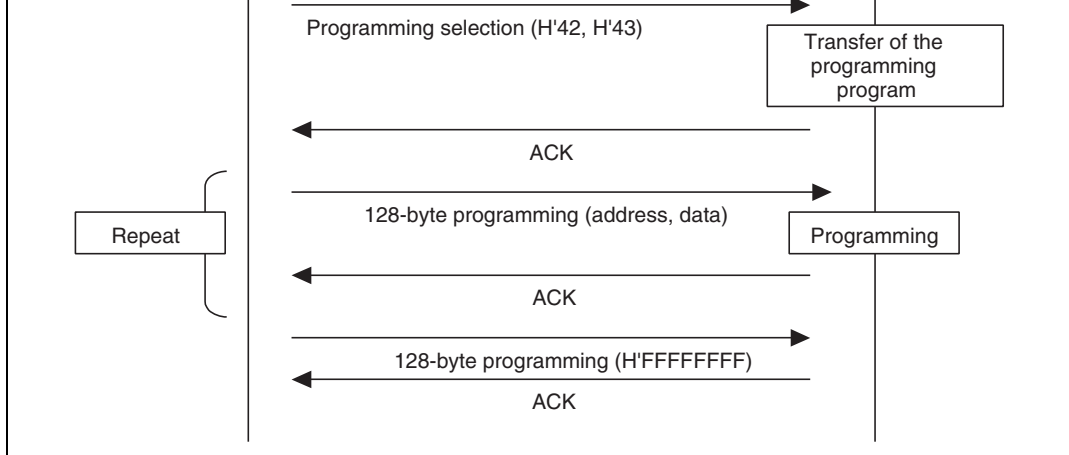
Firstly, the host should send the programming-selection command and select the programming method and programming MATs. There are two programming selection commands, and selection is according to the area and method for programming.

1. User boot MAT programming selection
2. User MAT 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 an 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.

Where the sequence of programming operations that is executed includes programming with another method or of another MAT, the procedure must be repeated from the programming selection command.

The sequence for programming-selection and 128-byte programming commands is shown in figure 20.21.



**Figure 20.21 Programming Sequence**

(a) 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-program programming selection

Response H'06

- Response, H'06, (one byte): Response to user boot-program programming selection  
When the programming program has been transferred, the boot program will return ACK.

Error Response H'C2 ERROR

- Error response : H'C2 (1 byte): Error response to user boot MAT programming selection
- ERROR : (1 byte): Error code  
H'54 : Selection processing error (transfer error occurs and processing is not completed)

• User-program programming selection

The boot program will transfer a program for programming. The data is programmed to the user MATs by the transferred program for programming.

Command H'43

- Command, H'43, (one byte): User-program programming selection

When the programming program has been transferred, the boot program will return ACK.

Error Response 

H'C3	ERROR
------	-------

- Error response : H'C3 (1 byte): Error response to user MAT programming selection
- ERROR : (1 byte): Error code  
H'54 : Selection processing error (transfer error occurs and processing is not completed)

(b) 128-byte programming

The boot program will use the programming program transferred by the programming selection to program the user boot MATs or user MATs in response to 128-byte programming.

Command	H'50	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 (i.e. H'00, H'01, H'00, H'00 : H'00010000)
- Programming Data (128 bytes): Data to be programmed  
The size is specified in the 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.

Error Response 

H'D0	ERROR
------	-------

- Error response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code  
H'11: Checksum Error  
H'2A: Address error  
The address is not within the specified MAT.  
H'53: Programming error  
A programming error has occurred and programming cannot be continued.

When there are less than 128 bytes of data to be programmed, the host should fill the rest with H'FF.

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 the programming and wait for selection of programming or erasing.

Command	H'50	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

Response	H'06
----------	------

- Response, H'06, (one byte): Response to 128-byte programming  
On completion of programming, the boot program will return ACK.

Error Response	H'D0	ERROR
----------------	------	-------

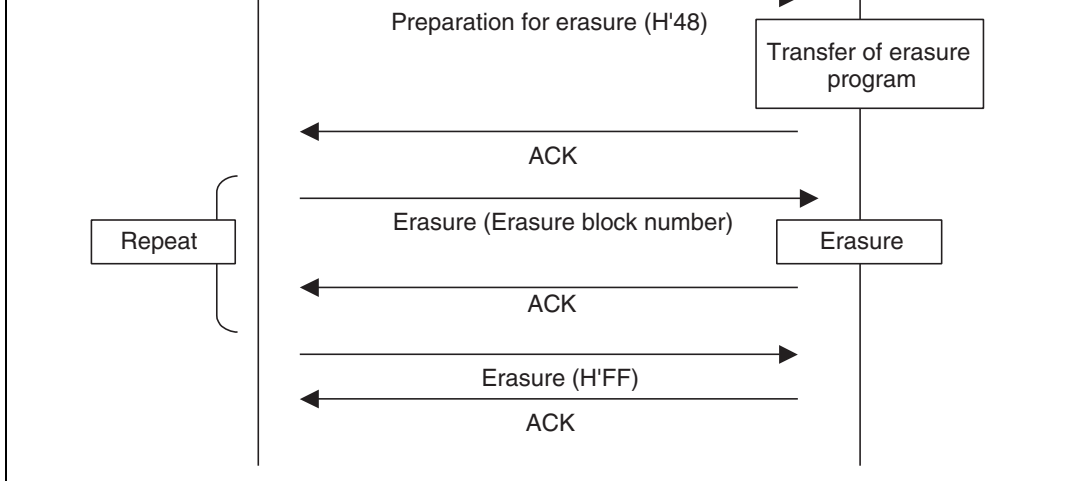
- Error Response, H'D0, (one byte): Error response for 128-byte programming
- ERROR: (one byte): Error code  
H'11: Checksum error  
H'53: Programming error  
An error has occurred in programming and programming cannot be continued.

## (10) Erasure

Erasure is performed with the erasure selection and block erasure command.

Firstly, erasure is selected by the erasure selection command and the boot program then erases the specified block. The command should be repeatedly executed if two or more blocks are to be erased. Sending a block-erasure command from the host with the block number H'FF will stop the erasure operating. On completion of erasing, the boot program will wait for selection of programming or erasing.

The sequences of the issuing of erasure selection commands and the erasure of data are shown in figure 20.22.



**Figure 20.22 Erasure Sequence**

(a) Erasure Selection

The boot program will transfer the erasure program. User MAT data is erased by the transferred erasure program.

Command 

H'48
------

- Command, H'48, (one byte): Erasure selection

Response 

H'06
------

- Response, H'06, (one byte): Response for erasure selection

After the erasure program has been transferred, the boot program will return ACK.

Error Response 

H'C8	ERROR
------	-------

- Error Response, H'C8, (one byte): Error response to erasure selection
- ERROR: (one byte): Error code
  - H'54: Selection processing error (transfer error occurs and processing is not completed)



The boot program will erase the contents of the specified block.

Command	H'58	Size	Block number	SUM
---------	------	------	--------------	-----

- Command, H'58, (one byte): Erasure
- Size (one byte): The number of bytes that represents the erasure block number  
This is fixed to 1.
- Block number (one byte): Number of the block to be erased
- SUM (one byte): Checksum

Response	H'06
----------	------

- Response, H'06, (one byte): Response to Erasure  
After erasure has been completed, the boot program will return ACK.

Error Response	H'D8	ERROR
----------------	------	-------

- Error Response, H'D8, (one byte): Response to Erasure
- ERROR (one byte): Error code
  - 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): The number of bytes that represents the block number  
This is fixed to 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 after the block number H'FF has been sent, the procedure should be executed from the erasure selection command.

The boot program will return the data in the specified address.

Command	H'52	Size	Area	Read address
	Read size			SUM

- Command: H'52 (1 byte): Memory read
- Size (1 byte): Amount of data that represents the area, read address, and read size (fixed at 9)
- Area (1 byte)  
H'00: User boot MAT  
H'01: User MAT  
An address error occurs when the area setting is incorrect.
- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response	H'52	Read size					
	Data	...					
	SUM						

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

Error Response	H'D2	ERROR
----------------	------	-------

- Error response: H'D2 (1 byte): Error response to memory read
- ERROR: (1 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.

## (12) User-Boot Program Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the user-boot program, as a four-byte value.

Command	H'4A
---------	------

- Command, H'4A, (one byte): Sum check for user-boot program

- Size (one byte): The number of bytes that represents the checksum  
This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user boot MATs  
The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

### (13) User-Program Sum Check

The boot program will return the byte-by-byte total of the contents of the bytes of the user program.

Command 

H'4B
------

- Command, H'4B, (one byte): Sum check for user program

Response 

H'5B	Size	Checksum of user program	SUM
------	------	--------------------------	-----

- Response, H'5B, (one byte): Response to the sum check of the user program
- Size (one byte): The number of bytes that represents the checksum  
This is fixed to 4.
- Checksum of user boot program (four bytes): Checksum of user MATs  
The total of the data is obtained in byte units.
- SUM (one byte): Sum check for data being transmitted

### (14) 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 for user boot MAT

Response 

H'06
------

- Response, H'06, (one byte): Response to the blank check of user boot MAT  
If all user MATs are blank (H'FF), the boot program will return ACK.

Error Response 

H'CC	H'52
------	------

- Error Response, H'CC, (one byte): Response to blank check for user boot MAT
- Error Code, H'52, (one byte): Erasure has not been completed.

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 for user MATs

Response 

H'06
------

- Response, H'06, (one byte): Response to the blank check for user boot MATs  
If the contents of 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 the blank check of user MATs.
- Error code, H'52, (one byte): Erasure has not been completed.

## (16) Boot Program State 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's state

Response 

H'5F	Size	Status	ERROR	SUM
------	------	--------	-------	-----

- Response, H'5F, (one byte): Response to boot program state inquiry
- Size (one byte): The number of bytes. This is fixed to 2.
- Status (one byte): State of the boot program

**Table 20.13 Status Code**

<b>Code</b>	<b>Description</b>
H'11	Device Selection Wait
H'12	Clock Mode Selection Wait
H'13	Bit Rate Selection Wait
H'1F	Programming/Erasing State Transition Wait (Bit rate selection is completed)
H'31	Programming State for Erasure
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 20.14 Error Code**

<b>Code</b>	<b>Description</b>
H'00	No Error
H'11	Sum Check Error
H'12	Program Size 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 Incomplete Error
H'53	Programming Error
H'54	Selection Processing Error
H'80	Command Error
H'FF	Bit-Rate-Adjustment Confirmation Error

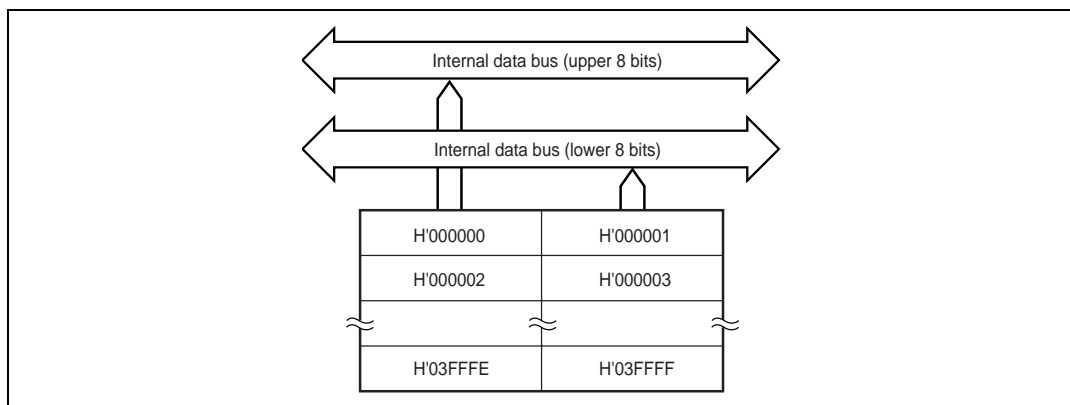
- SUM (one byte): Sum check

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 4 kbytes or less. Accordingly, when the CPU clock frequency is 34 MHz, the download for each program takes approximately 60  $\mu$ s at maximum.
2. Write to flash-memory related registers by DMAC  
While an instruction in on-chip RAM is being executed, the 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. Do not use DMAC to program FLASH related registers.
3. Compatibility with programming/erasing program of conventional F-ZTAT H8 microcomputer  
A programming/erasing program for flash memory used in the conventional F-ZTAT H8 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.
4. Monitoring runaway by WDT  
Unlike the conventional F-ZTAT H8 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 periodic timer interrupts) for WDT while taking the programming/erasing time into consideration as required.
5. Notes on supplying power  
When the power is supplied, the reset signal must be a low level and the external-input clock must be supplied.
6. User branch processing intervals  
The user branch processing interval differs for programming and erasing operations. Table 20.15 shows the maximum start intervals when the CPU clock frequency is 35 MHz.

**Table 20.15 User Branch Processing Start Intervals**

	<b>Maximum Interval</b>
Programming operation	1 ms
Erasing operation	30 ms

H8S/2635 has 256 kbytes of mask ROM. The on-chip ROM is connected to the CPU, data transfer controller (DTC), and DMA controller (DMAC) with a 16-bit data bus. The on-chip ROM can be accessed by the CPU, DTC, and DMAC in 8 or 16-bit units. The data in the on-chip ROM can always be accessed in one state.



**Figure 21.1 Block Diagram of 256-kbyte Mask ROM (HD6432365)**

The on-chip ROM is enabled or disabled according to the operating mode. The operating mode is selected by the mode setting pins MD2 to MD0 as shown in table 3.1. Select mode 4 or 7 when the on-chip ROM is used, and mode 1 or 2 when the on-chip ROM is not used. The on-chip ROM is allocated in area 0.

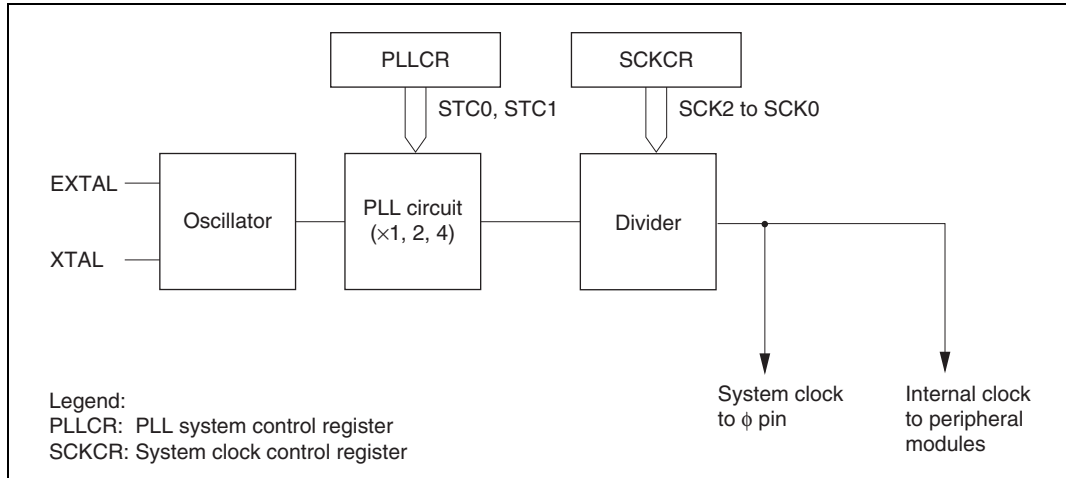




This LSI has an on-chip clock pulse generator (CPG) that generates the system clock ( $\phi$ ) and internal clocks.

The clock pulse generator consists of an oscillator circuit, PLL circuit, and divider.

Figure 22.1 shows a block diagram of the clock pulse generator.



**Figure 22.1 Block Diagram of Clock Pulse Generator**

The frequency can be changed by means of the PLL circuit. Frequency changes are made by software by means of settings in the PLL control register (PLLCR) and the system clock control register (SCKCR).

## 22.1 Register Descriptions

The clock pulse generator has the following registers.

- System clock control register (SCKCR)
- PLL control register (PLLCR)

SCKCR controls  $\phi$  clock output and selects operation when the frequency multiplication factor used by the PLL circuit is changed, and the division ratio used by the divider.

Bit	Bit Name	Initial Value	R/W	Description
7	PSTOP	0	R/W	<p><math>\phi</math> Clock Output Disable</p> <p>Controls <math>\phi</math> output.</p> <p>Normal Operation</p> <p>0: <math>\phi</math> output</p> <p>1: Fixed high</p> <p>Sleep Mode</p> <p>0: <math>\phi</math> output</p> <p>1: Fixed high</p> <p>Software Standby Mode</p> <p>0: Fixed high</p> <p>1: Fixed high</p> <p>Hardware Standby Mode</p> <p>0: High impedance</p> <p>1: High impedance</p> <p>All module clock stop mode</p> <p>0: <math>\phi</math> output</p> <p>1: Fixed high</p>
6	—	0	R/W	<p>Reserved</p> <p>Though this bit can be read from or written to, the write value should always be 0.</p>
5, 4	—	All 0	—	<p>Reserved</p> <p>These bits are always read as 0 and cannot be modified.</p>
3	STCS	0	R/W	<p>Frequency Multiplication Factor Switching Mode Select</p> <p>Selects the operation when the PLL circuit frequency multiplication factor is changed.</p> <p>0: Specified multiplication factor is valid after transition to software standby mode</p> <p>1: Specified multiplication factor is valid immediately after STC1 and STC0 bits are rewritten</p>

1	SCK1	0	R/W	Select the division ratio.
0	SCK0	0	R/W	000: 1/1
				001: 1/2
				010: 1/4
				011: Setting prohibited
				100: Setting prohibited
				101: Setting prohibited
				11x: Setting prohibited

Legend:

x: Don't care

### 22.1.2 PLL Control Register (PLLCR)

PLLCR sets the frequency multiplication factor used by the PLL circuit.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.
3	—	0	R/W	Reserved Though this bit can be read from or written to, the write value should always be 0.
2	—	0	R/W	Reserved This bit is always read as 0 and cannot be modified.
1 0	STC1 STC0	0 0	R/W R/W	Frequency Multiplication Factor The STC bits specify the frequency multiplication factor used by the PLL circuit. 00: ×1 01: ×2 10: ×4 11: Setting prohibited

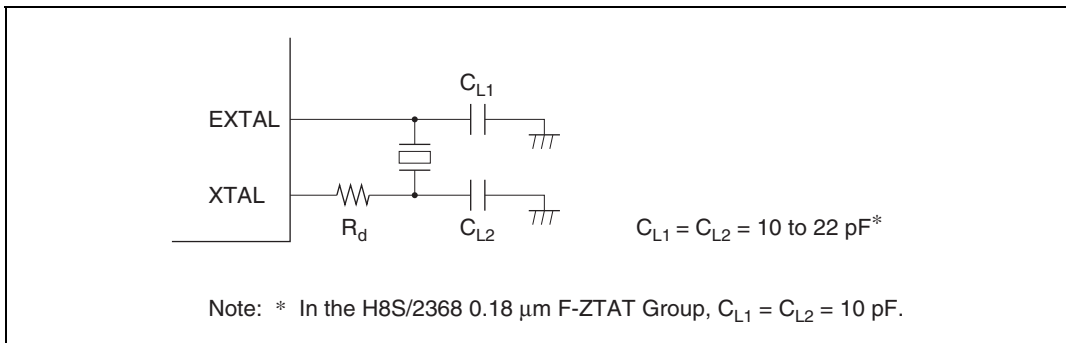
Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

### 22.2.1 Connecting a Crystal Oscillator

A crystal oscillator can be connected as shown in the example in figure 22.2. Select the damping resistance  $R_d$  according to table 22.1. An AT-cut parallel-resonance type should be used.

When a clock is supplied with a crystal resonator connected, the frequency of the crystal resonator should be 8 MHz to 25 MHz.

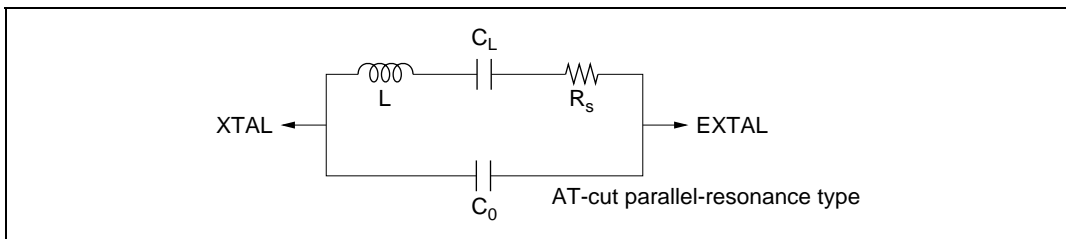
Figure 22.3 shows the equivalent circuit of the crystal oscillator. Use a crystal oscillator that has the characteristics shown in table 22.2.



**Figure 22.2 Connection of Crystal Oscillator (Example)**

**Table 22.1 Damping Resistance Value**

Frequency (MHz)	8	12	16	20	25
$R_d$ ( $\Omega$ )	200	0	0	0	0



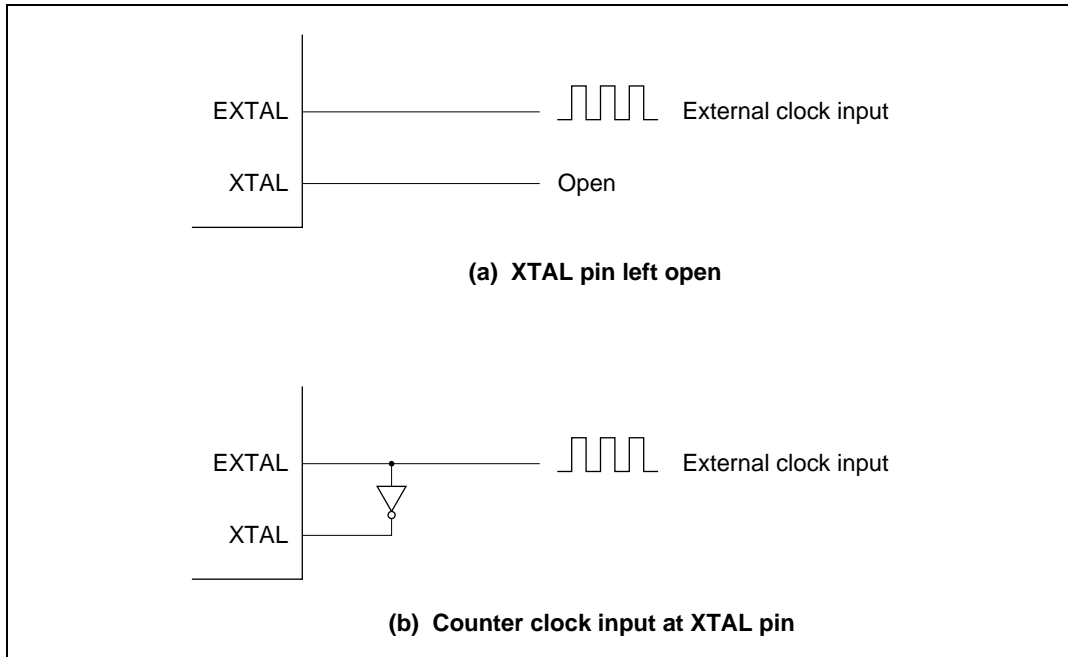
**Figure 22.3 Crystal Oscillator Equivalent Circuit**

Frequency (MHz)	8	12	16	20	25
$R_s$ max ( $\Omega$ )	80	60	50	40	40
$C_0$ max (pF)	7	7	7	7	7

### 22.2.2 External Clock Input

An external clock signal can be input as shown in the examples in figure 22.4. If the XTAL pin is left open, make sure that parasitic capacitance is no more than 10 pF. When the counter clock is input to the XTAL pin, make sure that the external clock is held high in standby mode.

Table 22.3 shows the input conditions for the external clock. The frequency of an external clock to be input should be 8 MHz to 25 MHz.



**Figure 22.4 External Clock Input (Examples)**

Item	Symbol	Min	Max	Unit	Test Conditions
External clock input low pulse width	t <sub>EXL</sub>	15	—	ns	Figure 22.5
External clock input high pulse width	t <sub>EXH</sub>	15	—	ns	
External clock rise time	t <sub>EXr</sub>	—	5	ns	
External clock fall time	t <sub>EXf</sub>	—	5	ns	
Clock low pulse width	t <sub>CL</sub>	0.4	0.6	t <sub>cyc</sub>	
Clock high pulse width	t <sub>CH</sub>	0.4	0.6	t <sub>cyc</sub>	

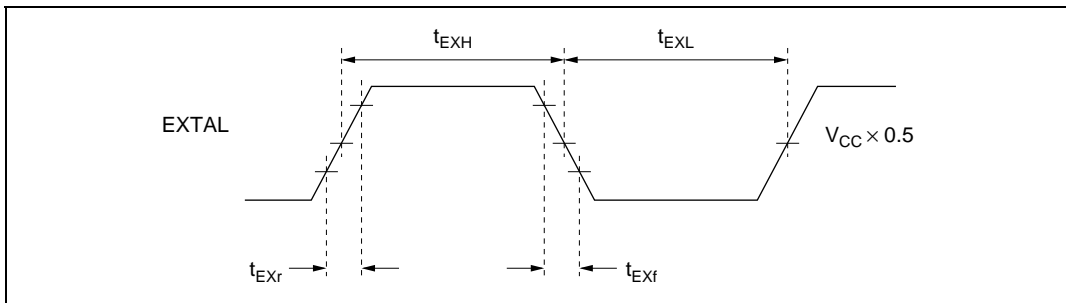


Figure 22.5 External Clock Input Timing

## 22.3 PLL Circuit

The PLL circuit has the function of multiplying the frequency of the clock from the oscillator by a factor of 1, 2, or 4. The multiplication factor is set with the STC1 and the STC0 bits in PLLCR. The phase of the rising edge of the internal clock is controlled so as to match that of the rising edge of the EXTAL pin.

When the multiplication factor of the PLL circuit is changed, the operation varies according to the setting of the STCS bit in SCKCR.

When STCS = 0, the setting becomes valid after a transition to software standby mode. The transition time count is performed in accordance with the setting of bits STS3 to STS0 in SBYCR. For details on SBYCR, refer to section 23.1.1, Standby Control Register (SBYCR).

1. The initial PLL circuit multiplication factor is 1.

mode.

4. The clock pulse generator stops and the value set in STC1 and STC0 becomes valid.
5. Software standby mode is cleared, and a transition time is secured in accordance with the setting in STS3 to STS0.
6. After the set transition time has elapsed, this LSI resumes operation using the target multiplication factor.

When STCS = 1, this LSI operates using the new multiplication factor immediately after bits STC1 and STC0 are rewritten.

## 22.4 Frequency Divider

The frequency divider divides the PLL output clock to generate a 1/2 or 1/4 clock.

## 22.5 Usage Notes

### 22.5.1 Notes on Clock Pulse Generator

1. The following points should be noted since the frequency of  $\phi$  changes according to the setting of SCKCR and PLLCR.

Select the clock division ratio that is within the operation guaranteed range of clock cycle time  $t_{cyc}$  shown in the AC timing of Electrical Characteristics. In other words, the range of  $\phi$  must be specified from 8 MHz (min) to 33 MHz\* (max); outside of this range must be prevented.

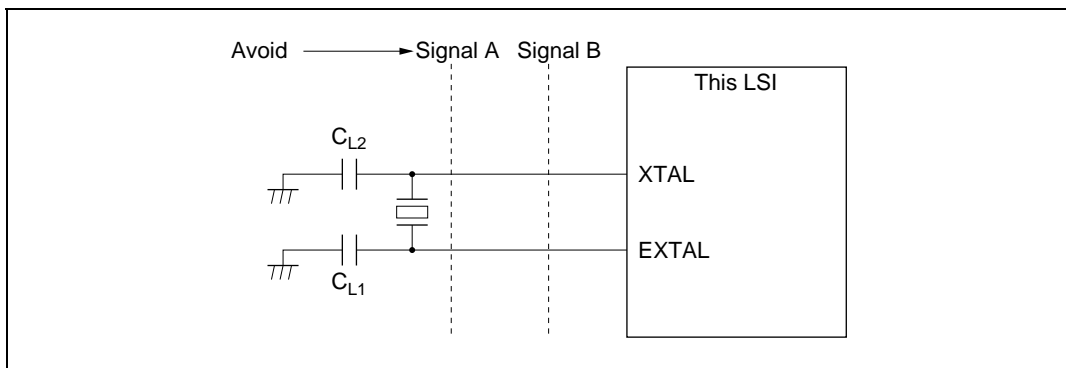
2. All the on-chip peripheral modules operate on the  $\phi$ . Therefore, note that the time processing of modules such as a timer and SCI differ before and after changing the clock division ratio. In addition, wait time for clearing software standby mode differs by changing the clock division ratio. See the description, Setting Oscillation Stabilization Time after Clearing Software Standby Mode in section 23.2.3, Software Standby Mode, for details.
3. Note that the frequency of  $\phi$  will be changed when setting SCKCR or PLLCR while executing the external bus cycle with the write-data-buffer function.

Note: \* 34 MHz for the H8S/2368 0.18 $\mu$ m F-ZTAT Group

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

### 22.5.3 Notes on Board Design

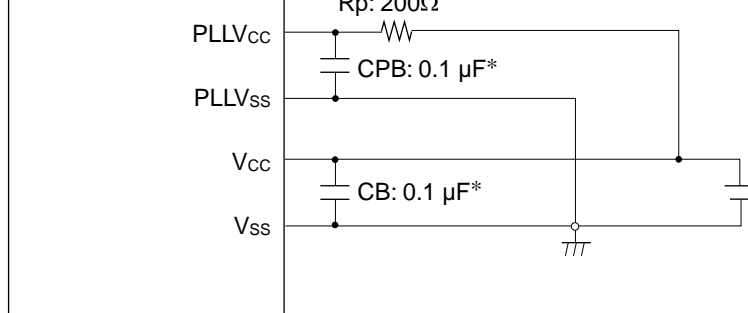
When using the crystal oscillator, place the crystal oscillator and its load capacitors as close as possible to the XTAL and EXTAL pins. Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 22.6.



**Figure 22.6 Note on Oscillator Board Design**

Figure 22.7 shows the external circuitry recommended for the PLL circuit. Separate PLLVcc and PLLVss from the other Vcc and Vss lines at the board power supply source, and be sure to insert bypass capacitors CPB and CB close to the pins.





Note: \* CB and CPB are laminated ceramic capacitors.

**Figure 22.7 Recommended External Circuitry for PLL Circuit**



In addition to the normal program execution state, this LSI has power-down modes in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip peripheral modules, and so on.

This LSI's operating modes are high-speed mode and six power down modes:

- Clock division mode
- Sleep mode
- Module stop mode
- All module clock stop mode
- Software standby mode
- Hardware standby mode

Sleep mode is a CPU state, clock division mode is an on-chip peripheral function (including bus masters and the CPU) state, and module stop mode is an on-chip peripheral function (including bus masters other than the CPU) state. A combination of these modes can be set.

After a reset, this LSI is in high-speed mode.

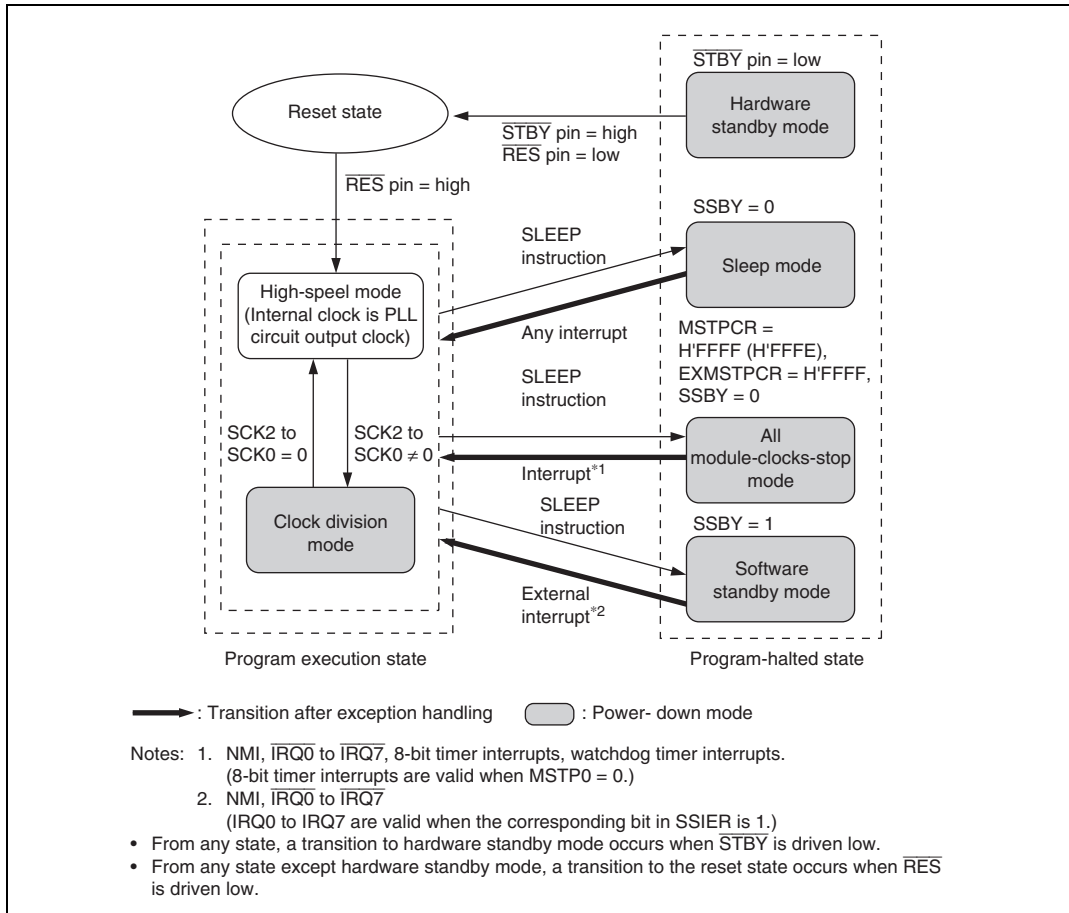
Table 23.1 shows the internal states of this LSI in each mode. Figure 23.1 shows the mode transition diagram.

Operating State	High Speed Mode	Clock Division Mode	Sleep Mode	Module Stop Mode	All Module Clock Stop Mode	Software Standby Mode	Hardware Standby Mode	
Clock pulse generator	Functions	Functions	Functions	Functions	Functions	Halted	Halted	
CPU	Instruction execution	Functions	Functions	Halted	Functions	Halted	Halted	Halted
	Register			Retained			Retained	Undefined
External interrupts	NMI	Functions	Functions	Functions	Functions	Functions	Functions	Halted
	IRQ0 to 7							
Peripheral functions	WDT	Functions	Functions	Functions	Functions	Functions	Halted (Retained)	Halted (Reset)
	TMR	Functions	Functions	Functions	Halted (Retained)	Functions/Halted (Retained) <sup>*1</sup>	Halted (Retained)	Halted (Reset)
	DMAC	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	DTC	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	TPU	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	PPG	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	D/A	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	A/D	Functions	Functions	Functions	Halted (Retained)	Halted (Retained)	Halted (Retained)	Halted (Reset)
	SCI	Functions	Functions	Functions	Halted <sup>*2</sup> (Reset/Retained)	Halted <sup>*2</sup> (Reset/Retained)	Halted <sup>*2</sup> (Reset/Retained)	Halted (Reset)
	IIC2	Functions	Functions	Functions	Halted <sup>*3</sup> (Reset/Retained)	Halted <sup>*3</sup> (Reset/Retained)	Halted <sup>*3</sup> (Reset/Retained)	Halted (Reset)
RAM	Functions	Functions	Functions	Functions	Functions	Retained	Retained	
I/O	Functions	Functions	Functions	Functions	Retained	Retained	High impedance	

Notes: “Halted (Retained)” in the table means that internal register values are retained and internal operations are suspended.

“Halted (Reset)” in the table means that internal register values and internal states are initialized.

1. The active or halted state can be selected by means of the MSTP0 bit in MSTPCR.
2. TDR, SSR, and RDR are halted (reset) and other registers are halted (retained).
3. BC2 to BC0 are halted (reset) and other registers are halted (retained).



**Figure 23.1 Mode Transitions**

The registers relating to the power-down mode are shown below. For details on the system clock control register (SCKCR), refer to section 22.1.1, System Clock Control Register (SCKCR).

- System clock control register (SCKCR)
- Standby control register (SBYCR)
- Module stop control register H (MSTPCRH)
- Module stop control register L (MSTPCRL)
- Extension module stop control register H (EXMSTPCRH)
- Extension module stop control register L (EXMSTPCRL)

### 23.1.1 Standby Control Register (SBYCR)

SBYCR performs software standby mode control.

Bit	Bit Name	Initial Value	R/W	Description
7	SSBY	0	R/W	<p>Software Standby</p> <p>This bit specifies the transition mode after executing the SLEEP instruction</p> <p>0: Shifts to sleep mode after the SLEEP instruction is executed</p> <p>1: Shifts to software standby mode after the SLEEP instruction is executed</p> <p>This bit does not change when clearing the software standby mode by using external interrupts and shifting to normal operation. This bit should be written 0 when clearing.</p>
6	OPE	1	R/W	<p>Output Port Enable</p> <p>Specifies whether the output of the address bus and bus control signals (<math>\overline{CS0}</math> to <math>\overline{CS7}</math>, <math>\overline{AS}</math>, <math>\overline{RD}</math>, <math>\overline{HWR}</math>, <math>\overline{LWR}</math>, <math>\overline{UCAS}</math>, <math>\overline{LCAS}</math>) is retained or set to the high-impedance state in software standby mode.</p> <p>0: In software standby mode, address bus and bus control signals are high-impedance</p> <p>1: In software standby mode, address bus and bus control signals retain output state</p>

These bits are always read as 0. The initial value should not be changed.

---

3	STS3	1	R/W	Standby Timer Select 3 to 0
2	STS2	1	R/W	These bits select the time the MCU waits for the clock to stabilize when software standby mode is cleared by an external interrupt. With crystal oscillation, refer to table 23.2 and make a selection according to the operating frequency so that the standby time is at least the oscillation stabilization time. With an external clock, a PLL circuit stabilization time is necessary. Refer to table 23.2 to set the wait time. When DRAM is used and self-refreshing in the software standby state is selected, note that the DRAM's tRAS (self-refresh RAS pulse width) specification must be satisfied.  With the F-ZTAT version, a flash memory stabilization time must be provided.  0000: Setting prohibited 0001: Setting prohibited 0010: Setting prohibited 0011: Setting prohibited 0100: Setting prohibited 0101: Standby time = 64 states 0110: Standby time = 512 states 0111: Standby time = 1024 states 1000: Standby time = 2048 states 1001: Standby time = 4096 states 1010: Standby time = 16384 states 1011: Standby time = 32768 states 1100: Standby time = 65536 states 1101: Standby time = 131072 states 1110: Standby time = 262144 states 1111: Standby time = 524288 states
1	STS1	1	R/W	
0	STS0	1	R/W	

---

Setting a bit to 1, the corresponding module enters module stop mode, while clearing the bit to 0 clears the module stop mode.

- **MSTPCRH**

Bit	Bit Name	Initial Value	R/W	Module
15	ACSE	0	R/W	All-Module-Clocks-Stop Mode Enable Enables or disables all-module-clocks-stop mode, in which, when the CPU executes a SLEEP instruction after module stop mode has been set for all the on-chip peripheral functions controlled by MSTPCR or the on-chip peripheral functions except the TMR. 0: All-module-clocks-stop mode disabled 1: All-module-clocks-stop mode enabled
14	MSTP14	0	R/W	—
13	MSTP13	0	R/W	DMA controller (DMAC)
12	MSTP12	0	R/W	Data transfer controller (DTC)
11	MSTP11	1	R/W	16-bit timer-pulse unit (TPU)
10	MSTP10	1	R/W	Programmable pulse generator (PPG)
9	MSTP9	1	R/W	—
8	MSTP8	1	R/W	D/A converter (channels 2 and 3)

- **MSTPCRL**

Bit	Bit Name	Initial Value	R/W	Module
7	MSTP7	1	R/W	—
6	MSTP6	1	R/W	A/D converter
5	MSTP5	1	R/W	Serial communication interface 4 (SCI_4)
4	MSTP4	1	R/W	Serial communication interface 3 (SCI_3)
3	MSTP3	1	R/W	Serial communication interface 2 (SCI_2)
2	MSTP2	1	R/W	Serial communication interface 1 (SCI_1)
1	MSTP1	1	R/W	Serial communication interface 0 (SCI_0)
0	MSTP0	1	R/W	8-bit timer (TMR)



EXMSTPCR performs all-module-clocks-stop mode control with MSTPCR.

When entering all-module-clocks-stop mode, set EXMSTPCR to H'FFFF. Otherwise, set EXMSTPCR to H'FFFD.

- EXMSTPCRH

Bit	Bit Name	Initial Value	R/W	Module
15 to 12	—	All 1	R/W	Reserved Read/write is enabled. 1 should be written in writing.
11	MSTP27	1	R/W	—
10	MSTP26	1	R/W	—
9	MSTP25	1	R/W	—
8	MSTP24	1	R/W	—

- EXMSTPCRL

Bit	Bit Name	Initial Value	R/W	Module
7	MSTP23	1	R/W	—
6	MSTP22	1	R/W	—
5	MSTP21	1	R/W	—
4	MSTP20	1	R/W	I <sup>2</sup> C bus interface 2_1 (IIC2_1)
3	MSTP19	1	R/W	I <sup>2</sup> C bus interface 2_0 (IIC2_0)
2	MSTP18	1	R/W	—
1	MSTP17	0	R/W	—
0	MSTP16	1	R/W	—

### 23.2.1 Clock Division Mode

When bits SCK2 to SCK0 in SCKCR are set to a value from 001 to 101, a transition is made to clock division mode at the end of the bus cycle. In clock division mode, the CPU, bus masters, and on-chip peripheral functions all operate on the operating clock (1/2 or 1/4) specified by bits SCK2 to SCK0.

Clock division mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode at the end of the bus cycle, and clock division mode is cleared.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is cleared to 0, the chip enters sleep mode. When sleep mode is cleared by an interrupt, clock division mode is restored.

If a SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1, the chip enters software standby mode. When software standby mode is cleared by an external interrupt, clock division mode is restored.

When the  $\overline{\text{RES}}$  pin is driven low, the reset state is entered and clock division mode is cleared. The same applies to a reset caused by watchdog timer overflow.

When the  $\overline{\text{STBY}}$  pin is driven low, a transition is made to hardware standby mode.

### 23.2.2 Sleep Mode

**Transition to Sleep Mode:** When the SLEEP instruction is executed when the SSBY bit is 0 in SBYCR, the CPU enters the sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other peripheral functions do not stop.

**Exiting Sleep Mode:** Sleep mode is exited by any interrupt, or signals at the  $\overline{\text{RES}}$ , or  $\overline{\text{STBY}}$  pins.

- **Exiting Sleep Mode by Interrupts:**  
When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by the CPU.
- **Exiting Sleep Mode by  $\overline{\text{RES}}$  pin:**  
Setting the  $\overline{\text{RES}}$  pin level low selects the reset state. After the stipulated reset input duration, driving the  $\overline{\text{RES}}$  pin high starts the CPU performing reset exception processing.
- **Exiting Sleep Mode by  $\overline{\text{STBY}}$  Pin:**

### 23.2.3 Software Standby Mode

**Transition to Software Standby Mode:** If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, software standby mode is entered. In this mode, the CPU, on-chip peripheral functions, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip peripheral functions other than the SCI and A/D converter, and I/O ports, are retained. Whether the address bus and bus control signals are placed in the high-impedance state or retain the output state can be specified by the OPE bit in SBYCR.

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

**Clearing Software Standby Mode:** Software standby mode is cleared by an external interrupt (NMI pin, or pins  $\overline{\text{IRQ0}}$  to  $\overline{\text{IRQ7}}$ ), or by means of the  $\overline{\text{RES}}$  pin or  $\overline{\text{STBY}}$  pin. Setting the SSI bit in SSIER to 1 enables  $\overline{\text{IRQ0}}$  to  $\overline{\text{IRQ7}}$  to be used as software standby mode clearing sources.

Clearing with an Interrupt:

When an NMI or IRQ0 to IRQ7 interrupt request signal is input, clock oscillation starts, and after the elapse of the time set in bits STS3 to STS0 in SBYCR, stable clocks are supplied to the entire LSI, software standby mode is cleared, and interrupt exception handling is started.

When clearing software standby mode with an IRQ0 to IRQ7 interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ7 is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.

Clearing with the  $\overline{\text{RES}}$  Pin:

When the  $\overline{\text{RES}}$  pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire LSI. Note that the  $\overline{\text{RES}}$  pin must be held low until clock oscillation stabilizes. When the  $\overline{\text{RES}}$  pin goes high, the CPU begins reset exception handling.

Clearing with the  $\overline{\text{STBY}}$  Pin:

When the  $\overline{\text{STBY}}$  pin is driven low, a transition is made to hardware standby mode.

Using a Crystal Oscillator:

Set bits STS3 to STS0 so that the standby time is more than the oscillation stabilization time.

Table 23.2 shows the standby times for operating frequencies and settings of bits STS3 to STS0.

Using an External Clock:

A PLL circuit stabilization time is necessary. Refer to table 23.2 to set the wait time.

**Table 23.2 Oscillation Stabilization Time Settings**

STS3	STS2	STS1	STS0	Standby Time	$\phi^{*1}$ [MHz]							Unit
					34 <sup>*2</sup>	33	25	20	13	10	8	
0	0	0	0	Reserved	—	—	—	—	—	—	—	$\mu$ s
			1	Reserved	—	—	—	—	—	—	—	
		1	0	Reserved	—	—	—	—	—	—	—	
			1	Reserved	—	—	—	—	—	—	—	
	1	0	0	Reserved	—	—	—	—	—	—	—	
			1	64	1.9	1.9	2.6	3.2	4.9	6.4	8.0	
		1	0	512	15.1	15.5	20.5	25.6	39.4	51.2	64.0	
			1	1024	30.1	31.0	41.0	51.2	78.8	102.4	128.0	
1	0	0	0	2048	60.2	62.1	81.9	102.4	157.5	204.8	256.0	ms
			1	4096	0.12	0.12	0.16	0.20	0.32	0.41	0.51	
		1	0	16384	0.48	0.50	0.66	0.82	1.26	1.64	2.05	
			1	32765	0.96	0.99	1.31	1.64	2.52	3.28	4.10	
	1	0	0	65536	1.93	1.99	2.62	3.28	5.04	6.55	8.19	
			1	131072	3.86	3.97	5.24	6.55	10.08	13.11	16.38	
		1	0	262144	7.71	7.94	10.49	13.11	20.16	26.21	32.77	
			1	524288	15.42	15.89	20.97	26.21	40.33	52.43	65.54	

: Recommended time setting

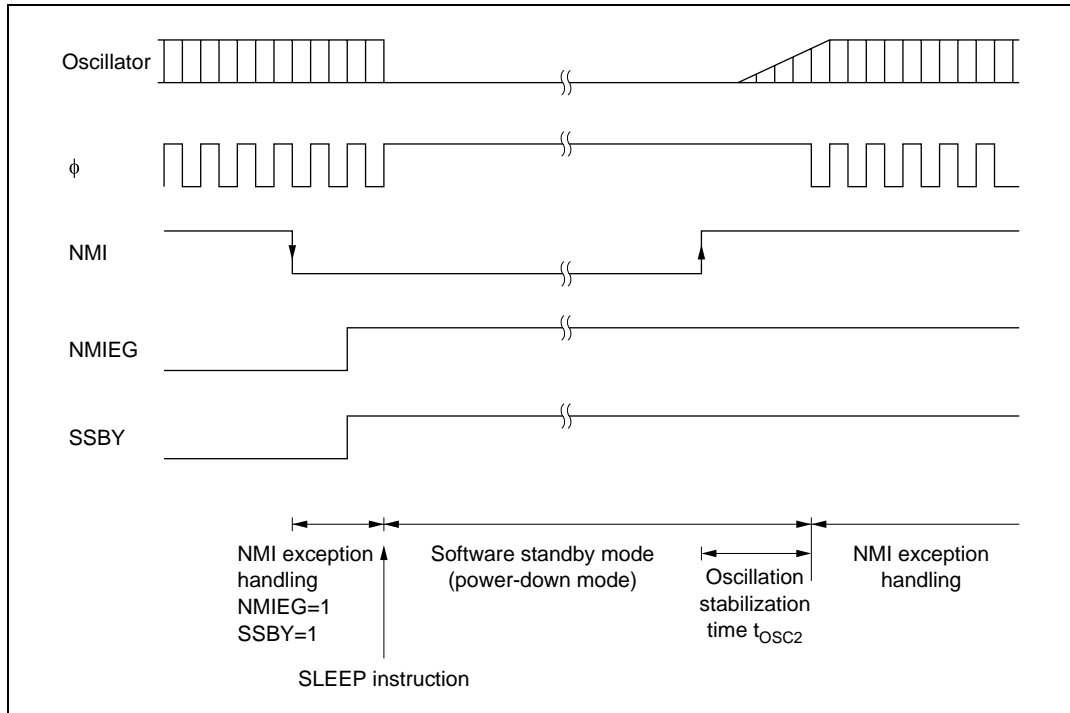
Notes: 1.  $\phi$  is the frequency divider output.

2. Supported only by the H8S/2368 0.18  $\mu$ m F-ZTAT Group.

standby mode is cleared at the rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in INTCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.



**Figure 23.2 Software Standby Mode Application Example**

**Transition to Hardware Standby Mode:** When the  $\overline{\text{STBY}}$  pin is driven low, a transition is made to hardware standby mode from any mode.

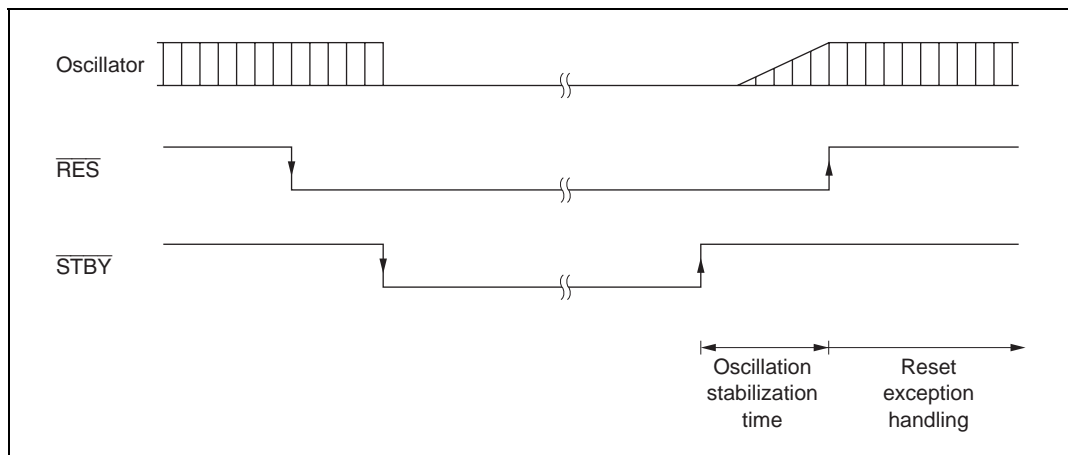
In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power dissipation. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the  $\overline{\text{STBY}}$  pin low. Do not change the state of the mode pins (MD2 to MD0) while this LSI is in hardware standby mode.

**Clearing Hardware Standby Mode:** Hardware standby mode is cleared by means of the  $\overline{\text{STBY}}$  pin and the  $\overline{\text{RES}}$  pin. When the  $\overline{\text{STBY}}$  pin is driven high while the  $\overline{\text{RES}}$  pin is low, the reset state is set and clock oscillation is started. Ensure that the  $\overline{\text{RES}}$  pin is held low until the clock oscillator stabilizes (for details on the oscillation stabilization time, refer to table 23.2). When the  $\overline{\text{RES}}$  pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

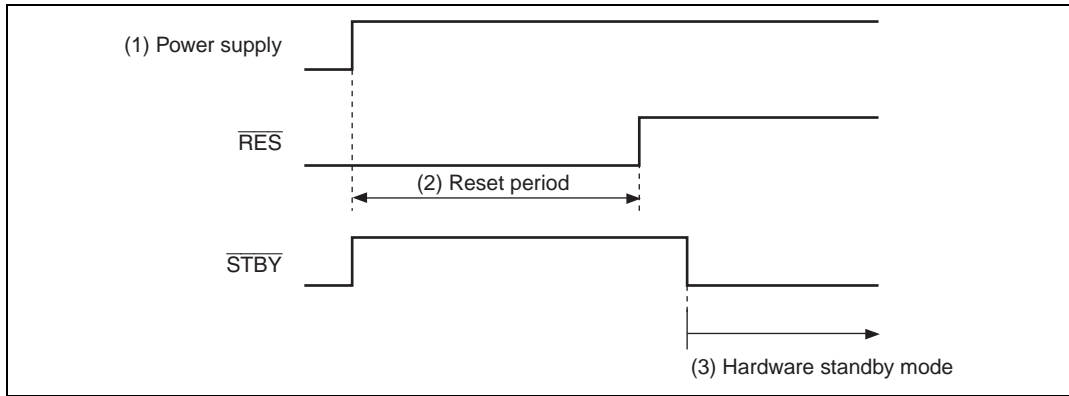
**Hardware Standby Mode Timing:** Figure 23.3 shows an example of hardware standby mode timing.

When the  $\overline{\text{STBY}}$  pin is driven low after the  $\overline{\text{RES}}$  pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the  $\overline{\text{STBY}}$  pin high, waiting for the oscillation stabilization time, then changing the  $\overline{\text{RES}}$  pin from low to high.



**Figure 23.3 Hardware Standby Mode Timing**

signal must be driven low for a given period with retaining the STBY signal high. After the RES signal is canceled, drive the STBY signal low.



**Figure 23.4 Hardware Standby Mode Timing when Power Is Supplied**

### 23.2.5 Module Stop Mode

Module stop mode can be set for individual on-chip peripheral modules.

When the corresponding MSTP bit in MSTPCR or EXMSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI are retained.

After reset clearance, all modules other than the DMAC, and DTC are in module stop mode.

The module registers which are set in module stop mode cannot be read or written to.

When the ACSE bit in MSTPCRH is set to 1 and module stop mode is set for all the on-chip peripheral functions controlled by MSTPCR or EXMSTPCR (MSTPCR = H'FFFF, EXMSTPCR = H'FFFF), or for all the on-chip peripheral functions except the 8-bit timer (MSTPCR = H'FFFE, EXMSTPCR = H'FFFF), executing a SLEEP instruction while the SSBY bit in SBYCR is cleared to 0 will cause all the on-chip peripheral functions (except the 8-bit timer and watchdog timer), the bus controller, and the I/O ports to stop operating, and a transition to be made to all-module-clocks-stop mode, at the end of the bus cycle.

Operation or halting of the 8-bit timer can be selected by means of the MSTP0 bit.

All-module-clocks-stop mode is cleared by an external interrupt (NMI,  $\overline{\text{IRQ0}}$  to  $\overline{\text{IRQ7}}$  pins),  $\overline{\text{RES}}$  pin input, or an internal interrupt (8-bit timer, watchdog timer), and the CPU returns to the normal program execution state via the exception handling state. All-module-clocks-stop mode is not cleared if interrupts are disabled, if interrupts other than NMI are masked by the CPU, or if the relevant interrupt is designated as a DTC activation source.

When the  $\overline{\text{STBY}}$  pin is driven low, a transition is made to hardware standby mode.

### 23.3 $\phi$ Clock Output Control

Output of the  $\phi$  clock can be controlled by means of the PSTOP bit in SCKCR, and DDR for the corresponding port. When the PSTOP bit is set to 1, the  $\phi$  clock stops at the end of the bus cycle, and  $\phi$  output goes high.  $\phi$  clock output is enabled when the PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0,  $\phi$  clock output is disabled and input port mode is set. Table 23.3 shows the state of the  $\phi$  pin in each processing state.

**Table 23.3  $\phi$  Pin State in Each Processing State**

Register Setting		Normal operating state	Sleep mode	Software standby mode	Hardware standby mode	All-module-clocks-stop mode
DDR	PSTOP					
0	X	High impedance	High impedance	High impedance	High impedance	High impedance
1	0	$\phi$ output	$\phi$ output	Fixed high	High impedance	$\phi$ output
1	1	Fixed high	Fixed high	Fixed high	High impedance	Fixed high



### **23.4.1 I/O Port Status**

In software standby mode, I/O port states are retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

### **23.4.2 Current Dissipation during Oscillation Stabilization Standby Period**

Current dissipation increases during the oscillation stabilization standby period.

### **23.4.3 DMAC/DTC Module Stop**

Depending on the operating status of the DMAC or DTC, the MSTP14 to MSTP13 and may not be set to 1. Setting of the DMAC or DTC module stop mode should be carried out only when the respective module is not activated.

For details, refer to section 7, DMA Controller (DMAC), and section 8, Data Transfer Controller (DTC).

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

Relevant interrupt operations cannot be performed in module stop mode. Consequently, if module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DMAC or DTC activation source.

Interrupts should therefore be disabled before entering module stop mode.

### **23.4.5 Writing to MSTPCR, EXMSTPCR**

MSTPCR and EXMSTPCR should only be written to by the CPU.

The following points should be noted in clock division mode.

- Select the clock division ratio specified by the SCK2 to SCK0 bits so that the frequency of  $\phi$  is within the operation guaranteed range of clock cycle time ( $t_{cyc}$ ) shown in the Electrical Characteristics. In other words, the range of  $\phi$  must be specified to 8 MHz (min.); outside of this range ( $\phi < 8$  MHz) must be prevented.
- All the on-chip peripheral modules operate on the  $\phi$ . Therefore, note the time processing of modules such as a timer and SCI differ before and after changing the clock division ratio. In addition, wait time for clearing software standby mode differs by changing the clock division ratio.
- Note that the frequency of  $\phi$  will be changed by changing the clock division ratio.

The address list gives information on the on-chip I/O register addresses, how the register bits are configured, and the register states in each operating mode. The information is given as shown below.

1. Register addresses (address order)
  - Registers are listed from the lower allocation addresses.
  - Registers are classified by functional modules.
  - The access size is indicated.
2. Register bits
  - Bit configurations of the registers are described in the same order as the register addresses.
  - Reserved bits are indicated by — in the bit name column.
  - No entry in the bit-name column indicates that the whole register is allocated as a counter or for holding data.
  - For the registers of 16 or 32 bits, the MSB is described first.
3. Register states in each operating mode
  - Register states are described in the same order as the register addresses.
  - The register states described here are for the basic operating modes. If there is a specific reset for an on-chip peripheral module, refer to the section on that on-chip peripheral module.

The data bus width indicates the numbers of bits by which the register is accessed.

The number of access states indicates the number of states based on the specified reference clock.

Register Name	Abbreviation	Bit No.	Address	Module	Data Width	Access States
DTC mode register A	MRA	8	H'BC00	DTC	16/32	2
DTC source address register	SAR	24		DTC	16/32	2
DTC mode register B	MRB	8	to	DTC	16/32	2
DTC destination address register	DAR	24		DTC	16/32	2
DTC transfer count register A	CRA	16		DTC	16/32	2
DTC transfer count register B	CRB	16	H'BFFF	DTC	16/32	2
I <sup>2</sup> C bus control register A_0	ICCRA_0	8	H'FD58	IIC2_0	8	2
I <sup>2</sup> C bus control register B_0	ICCRB_0	8	H'FD59	IIC2_0	8	2
I <sup>2</sup> C bus mode register_0	ICMR_0	8	H'FD5A	IIC2_0	8	2
I <sup>2</sup> C bus interrupt enable register_0	ICIER_0	8	H'FD5B	IIC2_0	8	2
I <sup>2</sup> C bus status register_0	ICSR_0	8	H'FD5C	IIC2_0	8	2
Slave address register_0	SAR_0	8	H'FD5D	IIC2_0	8	2
I <sup>2</sup> C transfer data register_0	ICDRT_0	8	H'FD5E	IIC2_0	8	2
I <sup>2</sup> C receive data register_0	ICDRR_0	8	H'FD5F	IIC2_0	8	2
I <sup>2</sup> C bus control register A_1	ICCRA_1	8	H'FD60	IIC2_1	8	2
I <sup>2</sup> C bus control register B_1	ICCRB_1	8	H'FD61	IIC2_1	8	2
I <sup>2</sup> C bus mode register_1	ICMR_1	8	H'FD62	IIC2_1	8	2
I <sup>2</sup> C bus interrupt enable register_1	ICIER_1	8	H'FD63	IIC2_1	8	2
I <sup>2</sup> C bus status register_1	ICSR_1	8	H'FD64	IIC2_1	8	2
Slave address register_1	SAR_1	8	H'FD65	IIC2_1	8	2
I <sup>2</sup> C transfer data register_1	ICDRT_1	8	H'FD66	IIC2_1	8	2
I <sup>2</sup> C receive data register_1	ICDRR_1	8	H'FD67	IIC2_1	8	2
Serial expansion mode register_2	SEMR_2	8	H'FDA8	SCI_2	8	2
Interrupt priority register A	IPRA	16	H'FE00	INT	16	2
Interrupt priority register B	IPRB	16	H'FE02	INT	16	2
Interrupt priority register C	IPRC	16	H'FE04	INT	16	2
Interrupt priority register D	IPRD	16	H'FE06	INT	16	2

Interrupt priority register E	IPRE	16	H'FE08	INT	16	2
Interrupt priority register F	IPRF	16	H'FE0A	INT	16	2
Interrupt priority register G	IPRG	16	H'FE0C	INT	16	2
Interrupt priority register H	IPRH	16	H'FE0E	INT	16	2
Interrupt priority register I	IPRI	16	H'FE10	INT	16	2
Interrupt priority register J	IPRJ	16	H'FE12	INT	16	2
Interrupt priority register K	IPRK	16	H'FE14	INT	16	2
IRQ pin select register	ITSR	16	H'FE16	INT	16	2
Software standby release IRQ enable register	SSIER	16	H'FE18	INT	16	2
IRQ sense control register L	ISCRL	16	H'FE1C	INT	16	2
IrDA control register_0	IrCR_0	8	H'FE1E	IrDA_0	8	2
Port 1 data direction register	P1DDR	8	H'FE20	PORT	8	2
Port 2 data direction register	P2DDR	8	H'FE21	PORT	8	2
Port 3 data direction register	P3DDR	8	H'FE22	PORT	8	2
Port 5 data direction register	P5DDR	8	H'FE24	PORT	8	2
Port 8 data direction register	P8DDR	8	H'FE27	PORT	8	2
Port A data direction register	PADDR	8	H'FE29	PORT	8	2
Port B data direction register	PBDDR	8	H'FE2A	PORT	8	2
Port C data direction register	PCDDR	8	H'FE2B	PORT	8	2
Port D data direction register	PDDDR	8	H'FE2C	PORT	8	2
Port E data direction register	PEDDR	8	H'FE2D	PORT	8	2
Port F data direction register	PFDDR	8	H'FE2E	PORT	8	2
Port G data direction register	PGDDR	8	H'FE2F	PORT	8	2
Port function control register 0	PFCR0	8	H'FE32	PORT	8	2
Port function control register 1	PFCR1	8	H'FE33	PORT	8	2
Port function control register 2	PFCR2	8	H'FE34	PORT	8	2
Port A MOS pull-up control register	PAPCR	8	H'FE36	PORT	8	2
Port B MOS pull-up control register	PBPCR	8	H'FE37	PORT	8	2
Port C MOS pull-up control register	PCPCR	8	H'FE38	PORT	8	2
Port D MOS pull-up control register	PDPCR	8	H'FE39	PORT	8	2
Port E MOS pull-up control register	PEPCR	8	H'FE3A	PORT	8	2

Port 3 open drain control register	P3ODR	8	H'FE3C	PORT	8	2
Port A open drain control register	PAODR	8	H'FE3D	PORT	8	2
Serial mode register_3	SMR_3	8	H'FE40	SCI_3	8	2
Bit rate register_3	BRR_3	8	H'FE41	SCI_3	8	2
Serial control register_3	SCR_3	8	H'FE42	SCI_3	8	2
Transmit data register_3	TDR_3	8	H'FE43	SCI_3	8	2
Serial status register_3	SSR_3	8	H'FE44	SCI_3	8	2
Receive data register_3	RDR_3	8	H'FE45	SCI_3	8	2
Smart card mode register_3	SCMR_3	8	H'FE46	SCI_3	8	2
Serial mode register_4	SMR_4	8	H'FE48	SCI_4	8	2
Bit rate register_4	BRR_4	8	H'FE49	SCI_4	8	2
Serial control register_4	SCR_4	8	H'FE4A	SCI_4	8	2
Transmit data register_4	TDR_4	8	H'FE4B	SCI_4	8	2
Serial status register_4	SSR_4	8	H'FE4C	SCI_4	8	2
Receive data register_4	RDR_4	8	H'FE4D	SCI_4	8	2
Smart card mode register_4	SCMR_4	8	H'FE4E	SCI_4	8	2
Timer control register_3	TCR_3	8	H'FE80	TPU_3	16	2
Timer mode register_3	TMDR_3	8	H'FE81	TPU_3	16	2
Timer I/O control register H_3	TIORH_3	8	H'FE82	TPU_3	16	2
Timer I/O control register L_3	TIORL_3	8	H'FE83	TPU_3	16	2
Timer interrupt enable register_3	TIER_3	8	H'FE84	TPU_3	16	2
Timer status register_3	TSR_3	8	H'FE85	TPU_3	16	2
Timer counter_3	TCNT_3	16	H'FE86	TPU_3	16	2
Timer general register A_3	TGRA_3	16	H'FE88	TPU_3	16	2
Timer general register B_3	TGRB_3	16	H'FE8A	TPU_3	16	2
Timer general register C_3	TGRC_3	16	H'FE8C	TPU_3	16	2
Timer general register D_3	TGRD_3	16	H'FE8E	TPU_3	16	2
Timer control register_4	TCR_4	8	H'FE90	TPU_4	16	2
Timer mode register_4	TMDR_4	8	H'FE91	TPU_4	16	2
Timer I/O control register_4	TIOR_4	8	H'FE92	TPU_4	16	2
Timer interrupt enable register_4	TIER_4	8	H'FE94	TPU_4	16	2
Timer status register_4	TSR_4	8	H'FE95	TPU_4	16	2

Timer counter_4	TCNT_4	16	H'FE96	TPU_4	16	2
Timer general register A_4	TGRA_4	16	H'FE98	TPU_4	16	2
Timer general register B_4	TGRB_4	16	H'FE9A	TPU_4	16	2
Timer control register_5	TCR_5	8	H'FEA0	TPU_5	16	2
Timer mode register_5	TMDR_5	8	H'FEA1	TPU_5	16	2
Timer I/O control register_5	TIOR_5	8	H'FEA2	TPU_5	16	2
Timer interrupt enable register_5	TIER_5	8	H'FEA4	TPU_5	16	2
Timer status register_5	TSR_5	8	H'FEA5	TPU_5	16	2
Timer counter_5	TCNT_5	16	H'FEA6	TPU_5	16	2
Timer general register A_5	TGRA_5	16	H'FEA8	TPU_5	16	2
Timer general register B_5	TGRB_5	16	H'FEAA	TPU_5	16	2
Bus width control register	ABWCR	8	H'FEC0	BSC	16	2
Access state control register	ASTCR	8	H'FEC1	BSC	16	2
Wait control register AH	WTCRAH	8	H'FEC2	BSC	16	2
Wait control register AL	WTCRAL	8	H'FEC3	BSC	16	2
Wait control register BH	WTCRBH	8	H'FEC4	BSC	16	2
Wait control register BL	WTCRBL	8	H'FEC5	BSC	16	2
Read strobe timing control register	RDNCR	8	H'FEC6	BSC	16	2
$\overline{CS}$ assertion period control registers H	CSACRH	8	H'FEC8	BSC	16	2
$\overline{CS}$ assertion period control register L	CSACRL	8	H'FEC9	BSC	16	2
Burst ROM interface control register H	BROMCRH	8	H'FECA	BSC	16	2
Burst ROM interface control register L	BROMCRL	8	H'FECB	BSC	16	2
Bus control register	BCR	16	H'FECC	BSC	16	2
DRAM control register	DRAMCR	16	H'FED0	BSC	16	2
DRAM access control register	DRACCR	8	H'FED2	BSC	16	2
Refresh control register	REFCR	16	H'FED4	BSC	16	2
Refresh timer counter	RTCNT	8	H'FED6	BSC	16	2
Refresh time constant register	RTCOR	8	H'FED7	BSC	16	2
Memory address register_0AH	MAR_0AH	16	H'FEE0	DMAC	16	2
Memory address register_0AL	MAR_0AL	16	H'FEE2	DMAC	16	2

I/O address register_0A	IOAR_0A	16	H'FEE4	DMAC	16	2
Transfer count register_0A	ETCR_0A	16	H'FEE6	DMAC	16	2
Memory address register_0BH	MAR_0BH	16	H'FEE8	DMAC	16	2
Memory address register_0BL	MAR_0BL	16	H'FEEA	DMAC	16	2
I/O address register_0B	IOAR_0B	16	H'FEEC	DMAC	16	2
Transfer count register_0B	ETCR_0B	16	H'FEEE	DMAC	16	2
Memory address register_1AH	MAR_1AH	16	H'FEF0	DMAC	16	2
Memory address register_1AL	MAR_1AL	16	H'FEF2	DMAC	16	2
I/O address register_1A	IOAR_1A	16	H'FEF4	DMAC	16	2
Transfer count register_1A	ETCR_1A	16	H'FEF6	DMAC	16	2
Memory address register_1BH	MAR_1BH	16	H'FEF8	DMAC	16	2
Memory address register_1BL	MAR_1BL	16	H'FEFA	DMAC	16	2
I/O address register_1B	IOAR_1B	16	H'FEFC	DMAC	16	2
Transfer count register_1B	ETCR_1B	16	H'FEFE	DMAC	16	2
DMA write enable register	DMAWER	8	H'FF20	DMAC	8	2
DMA terminal control register	DMATCR	8	H'FF21	DMAC	8	2
DMA control register_0A	DMACR_0A	8	H'FF22	DMAC	16	2
DMA control register_0B	DMACR_0B	8	H'FF23	DMAC	16	2
DMA control register_1A	DMACR_1A	8	H'FF24	DMAC	16	2
DMA control register_1B	DMACR_1B	8	H'FF25	DMAC	16	2
DMA band control register H	DMABCRH	8	H'FF26	DMAC	16	2
DMA band control register L	DMABCRL	8	H'FF27	DMAC	16	2
DTC enable register A	DTCERA	8	H'FF28	DTC	16	2
DTC enable register B	DTCERB	8	H'FF29	DTC	16	2
DTC enable register C	DTCERC	8	H'FF2A	DTC	16	2
DTC enable register D	DTCERD	8	H'FF2B	DTC	16	2
DTC enable register E	DTCERE	8	H'FF2C	DTC	16	2
DTC enable register F	DTCERF	8	H'FF2D	DTC	16	2
DTC enable register G	DTCERG	8	H'FF2E	DTC	16	2
DTC enable register H	DTCERH	8	H'FF2F	DTC	16	2
DTC vector register	DTVECR	8	H'FF30	DTC	16	2



Interrupt control register	INTCR	8	H'FF31	INT	16	2
IRQ enable register	IER	16	H'FF32	INT	16	2
IRQ status register	ISR	16	H'FF34	INT	16	2
Standby control register	SBYCR	8	H'FF3A	SYSTEM	8	2
System clock control register	SCKCR	8	H'FF3B	SYSTEM	8	2
System control register	SYSCR	8	H'FF3D	SYSTEM	8	2
Mode control register	MDCR	8	H'FF3E	SYSTEM	8	2
Module stop control register H	MSTPCR <sub>H</sub>	8	H'FF40	SYSTEM	8	2
Module stop control register L	MSTPCL	8	H'FF41	SYSTEM	8	2
Extension module stop control register H	EXMSTPCR <sub>H</sub>	8	H'FF42	SYSTEM	8	2
Extension module stop control register L	EXMSTPCL	8	H'FF43	SYSTEM	8	2
PLL control register	PLLCR	8	H'FF45	SYSTEM	8	2
PPG output control register	PCR	8	H'FF46	PPG	8	2
PPG output mode register	PMR	8	H'FF47	PPG	8	2
Next data enable register H	NDERH	8	H'FF48	PPG	8	2
Next data enable register L	NDERL	8	H'FF49	PPG	8	2
Output data register H	PODRH	8	H'FF4A	PPG	8	2
Output data register L	PODRL	8	H'FF4B	PPG	8	2
Next data register H <sup>*1</sup>	NDRH	8	H'FF4C	PPG	8	2
Next data register L <sup>*1</sup>	NDRL	8	H'FF4D	PPG	8	2
Next data register H <sup>*1</sup>	NDRH	8	H'FF4E	PPG	8	2
Next data register L <sup>*1</sup>	NDRL	8	H'FF4F	PPG	8	2
Port 1 register	PORT1	8	H'FF50	PORT	8	2
Port 2 register	PORT2	8	H'FF51	PORT	8	2
Port 3 register	PORT3	8	H'FF52	PORT	8	2
Port 4 register	PORT4	8	H'FF53	PORT	8	2
Port 5 register	PORT5	8	H'FF54	PORT	8	2
Port 8 register	PORT8	8	H'FF57	PORT	8	2
Port 9 register	PORT9	8	H'FF58	PORT	8	2

Port A register	PORTA	8	H'FF59	PORT	8	2
Port B register	PORTB	8	H'FF5A	PORT	8	2
Port C register	PORTC	8	H'FF5B	PORT	8	2
Port D register	PORTD	8	H'FF5C	PORT	8	2
Port E register	PORTE	8	H'FF5D	PORT	8	2
Port F register	PORTF	8	H'FF5E	PORT	8	2
Port G register	PORTG	8	H'FF5F	PORT	8	2
Port 1 data register	P1DR	8	H'FF60	PORT	8	2
Port 2 data register	P2DR	8	H'FF61	PORT	8	2
Port 3 data register	P3DR	8	H'FF62	PORT	8	2
Port 5 data register	P5DR	8	H'FF64	PORT	8	2
Port 8 data register	P8DR	8	H'FF67	PORT	8	2
Port A data register	PADR	8	H'FF69	PORT	8	2
Port B data register	PBDR	8	H'FF6A	PORT	8	2
Port C data register	PCDR	8	H'FF6B	PORT	8	2
Port D data register	PDDR	8	H'FF6C	PORT	8	2
Port E data register	PEDR	8	H'FF6D	PORT	8	2
Port F data register	PFDR	8	H'FF6E	PORT	8	2
Port G data register	PGDR	8	H'FF6F	PORT	8	2
Serial mode register_0	SMR_0	8	H'FF78	SCI_0	8	2
Bit rate register_0	BRR_0	8	H'FF79	SCI_0	8	2
Serial control register_0	SCR_0	8	H'FF7A	SCI_0	8	2
Transmit data register_0	TDR_0	8	H'FF7B	SCI_0	8	2
Serial status register_0	SSR_0	8	H'FF7C	SCI_0	8	2
Receive data register_0	RDR_0	8	H'FF7D	SCI_0	8	2
Smart card mode register_0	SCMR_0	8	H'FF7E	SCI_0	8	2
Serial mode register_1	SMR_1	8	H'FF80	SCI_1	8	2
Bit rate register_1	BRR_1	8	H'FF81	SCI_1	8	2
Serial control register_1	SCR_1	8	H'FF82	SCI_1	8	2
Transmit data register_1	TDR_1	8	H'FF83	SCI_1	8	2
Serial status register_1	SSR_1	8	H'FF84	SCI_1	8	2
Receive data register_1	RDR_1	8	H'FF85	SCI_1	8	2

Smart card mode register_1	SCMR_1	8	H'FF86	SCI_1	8	2
Serial mode register_2	SMR_2	8	H'FF88	SCI_2	8	2
Bit rate register_2	BRR_2	8	H'FF89	SCI_2	8	2
Serial control register_2	SCR_2	8	H'FF8A	SCI_2	8	2
Transmit data register_2	TDR_2	8	H'FF8B	SCI_2	8	2
Serial status register_2	SSR_2	8	H'FF8C	SCI_2	8	2
Receive data register_2	RDR_2	8	H'FF8D	SCI_2	8	2
Smart card mode register_2	SCMR_2	8	H'FF8E	SCI_2	8	2
A/D data register A	ADDRA	16	H'FF90	A/D	16	2
A/D data register B	ADDRB	16	H'FF92	A/D	16	2
A/D data register C	ADDRC	16	H'FF94	A/D	16	2
A/D data register D	ADDRD	16	H'FF96	A/D	16	2
A/D data register E	ADDRE	16	H'FF98	A/D	16	2
A/D data register F	ADDRF	16	H'FF9A	A/D	16	2
A/D data register G	ADDRG	16	H'FF9C	A/D	16	2
A/D data register H	ADDRH	16	H'FF9E	A/D	16	2
A/D control/status register	ADCSR	8	H'FFA0	A/D	16	2
A/D control register	ADCR	8	H'FFA1	A/D	16	2
D/A data register 2	DADR2	8	H'FFA8	D/A	8	2
D/A data register 3	DADR3	8	H'FFA9	D/A	8	2
D/A control register 23	DACR23	8	H'FFAA	D/A	8	2
Timer control register_0	TCR_0	8	H'FFB0	TMR_0	16	2
Timer control register_1	TCR_1	8	H'FFB1	TMR_1	16	2
Timer control/status register_0	TCSR_0	8	H'FFB2	TMR_0	16	2
Timer control/status register_1	TCSR_1	8	H'FFB3	TMR_1	16	2
Time constant register_A0	TCORA_0	8	H'FFB4	TMR_0	16	2
Time constant register_A1	TCORA_1	8	H'FFB5	TMR_1	16	2
Time constant register_B0	TCORB_0	8	H'FFB6	TMR_0	16	2
Time constant register_B1	TCORB_1	8	H'FFB7	TMR_1	16	2
Timer counter_0	TCNT_0	8	H'FFB8	TMR_0	16	2
Timer counter_1	TCNT_1	8	H'FFB9	TMR_1	16	2

Timer control/status register	TCSR_0	8	H'FFBC* <sup>2</sup> (Write)	WDT	16	2
			H'FFBC (Read)			
Timer counter	TCNT	8	H'FFBC* <sup>2</sup> (Write)	WDT	16	2
			H'FFBD (Read)			
Reset control/status register	RSTCSR	8	H'FFBE* <sup>2</sup> (Write)	WDT	16	2
			H'FFBF (Read)			
Timer start register	TSTR	8	H'FFC0	TPU	16	2
Timer synchronous register	TSYR	8	H'FFC1	TPU	16	2
Flash code control status register	FCCS* <sup>3</sup>	8	H'FFC4* <sup>4</sup>	FLASH	8	2
Flash program code select register	FPCS* <sup>3</sup>	8	H'FFC5* <sup>4</sup>	FLASH	8	2
Flash erase code select register	FECS* <sup>3</sup>	8	H'FFC6* <sup>4</sup>	FLASH	8	2
Flash memory control register 1	FLMCR1	8	H'FFC8	FLASH	8	2
Flash key code register	FKEY* <sup>3</sup>	8	H'FFC8	FLASH	8	2
Flash memory control register 2	FLMCR2	8	H'FFC9	FLASH	8	2
Flash MAT select register	FMATS* <sup>3</sup>	8	H'FFC9	FLASH	8	2
Flash transfer destination address register	FTDAR* <sup>3</sup>	8	H'FFCA	FLASH	8	2
Erase block register 1	EBR1	8	H'FFCA	FLASH	8	2
Erase block register 2	EBR2	8	H'FFCB	FLASH	8	2
Flash vector address control register	FVACR* <sup>3</sup>	8	H'FFCB	FLASH	8	2
Timer control register_0	TCR_0	8	H'FFD0	TPU_0	16	2
Timer mode register_0	TMDR_0	8	H'FFD1	TPU_0	16	2
Timer I/O control register H_0	TIORH_0	8	H'FFD2	TPU_0	16	2
Timer I/O control register L_0	TIORL_0	8	H'FFD3	TPU_0	16	2
Timer interrupt enable register_0	TIER_0	8	H'FFD4	TPU_0	16	2
Timer status register_0	TSR_0	8	H'FFD5	TPU_0	16	2
Timer counter_0	TCNT_0	16	H'FFD6	TPU_0	16	2
Timer general register A_0	TGRA_0	16	H'FFD8	TPU_0	16	2
Timer general register B_0	TGRB_0	16	H'FFDA	TPU_0	16	2

Timer general register C_0	TGRC_0	16	H'FFDC	TPU_0	16	2
Timer general register D_0	TGRD_0	16	H'FFDE	TPU_0	16	2
Timer control register_1	TCR_1	8	H'FFE0	TPU_1	16	2
Timer mode register_1	TMDR_1	8	H'FFE1	TPU_1	16	2
Timer I/O control register_1	TIOR_1	8	H'FFE2	TPU_1	16	2
Timer interrupt enable register_1	TIER_1	8	H'FFE4	TPU_1	16	2
Timer status register_1	TSR_1	8	H'FFE5	TPU_1	16	2
Timer counter_1	TCNT_1	16	H'FFE6	TPU_1	16	2
Timer general register A_1	TGRA_1	16	H'FFE8	TPU_1	16	2
Timer general register B_1	TGRB_1	16	H'FFEA	TPU_1	16	2
Timer control register_2	TCR_2	8	H'FFF0	TPU_2	16	2
Timer mode register_2	TMDR_2	8	H'FFF1	TPU_2	16	2
Timer I/O control register_2	TIOR_2	8	H'FFF2	TPU_2	16	2
Timer interrupt enable register_2	TIER_2	8	H'FFF4	TPU_2	16	2
Timer status register_2	TSR_2	8	H'FFF5	TPU_2	16	2
Timer counter_2	TCNT_2	16	H'FFF6	TPU_2	16	2
Timer general register A_2	TGRA_2	16	H'FFF8	TPU_2	16	2
Timer general register B_2	TGRB_2	16	H'FFFA	TPU_2	16	2

- Notes:
1. If the pulse output group 2 and pulse output group 3 output triggers are the same according to the PCR setting, the NDRH address will be H'FF4C, and if different, the address of NDRH for group 2 will be H'FF4E, and that for group 3 will be H'FF4C. Similarly, if the pulse output group 0 and pulse output group 1 output triggers are the same according to the PCR setting, the NDRL address will be H'FF4D, and if different, the address of NDRL for group 0 will be H'FF4F, and that for group 1 will be H'FF4D.
  2. For writing, refer to section 13.6.1, Notes on Register Access.
  3. Supported only by the H8S/2368 0.18  $\mu$ m F-ZTAT Group.
  4. Cannot be accessed by other than H8S/2368 0.18  $\mu$ m F-ZTAT Group.

Register bit names of the on-chip peripheral modules are described below.

Each line covers eight bits, and 16- or 32-bit registers are shown as 2 or 4 lines.

Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
MRA	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz	DTC <sup>*1</sup>
SAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
MRB	CHNE	DISEL	CHNS	—	—	—	—	—	
DAR	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRA	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
CRB	—	—	—	—	—	—	—	—	
	—	—	—	—	—	—	—	—	
ICCRA_0	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2_0
ICCRB_0	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—	
ICMR_0	—	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER_0	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_0	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ	
SAR_0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	—	
ICDRT_0	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_0	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	
ICCRA_1	ICE	RCVD	MST	TRS	CKS3	CKS2	CKS1	CKS0	IIC2_1
ICCRB_1	BBSY	SCP	SDAO	—	SCLO	—	IICRST	—	
ICMR_1	—	WAIT	—	—	BCWP	BC2	BC1	BC0	
ICIER_1	TIE	TEIE	RIE	NAKIE	STIE	ACKE	ACKBR	ACKBT	
ICSR_1	TDRE	TEND	RDRF	NACKF	STOP	AL	AAS	ADZ	
SAR_1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	—	
ICDRT_1	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0	
ICDRR_1	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0	

SEMR_2	—	—	—	—	ABCS	ACS2	ACS1	ACS0	SCI_2 Smart card interface 2
IPRA	—	IPRA14	IPRA13	IPRA12	—	IPRA10	IPRA9	IPRA8	INT
	—	IPRA6	IPRA5	IPRA4	—	IPRA2	IPRA1	IPRA0	
IPRB	—	IPRB14	IPRB13	IPRB12	—	IPRB10	IPRB9	IPRB8	
	—	IPRB6	IPRB5	IPRB4	—	IPRB2	IPRB1	IPRB0	
IPRC	—	IPRC14	IPRC13	IPRC12	—	IPRC10	IPRC9	IPRC8	
	—	IPRC6	IPRC5	IPRC4	—	IPRC2	IPRC1	IPRC0	
IPRD	—	IPRD14	IPRD13	IPRD12	—	IPRD10	IPRD9	IPRD8	
	—	IPRD6	IPRD5	IPRD4	—	IPRD2	IPRD1	IPRD0	
IPRE	—	IPRE14	IPRE13	IPRE12	—	IPRE10	IPRE9	IPRE8	
	—	IPRE6	IPRE5	IPRE4	—	IPRE2	IPRE1	IPRE0	
IPRF	—	IPRF14	IPRF13	IPRF12	—	IPRF10	IPRF9	IPRF8	
	—	IPRF6	IPRF5	IPRF4	—	IPRF2	IPRF1	IPRF0	
IPRG	—	IPRG14	IPRG13	IPRG12	—	IPRG10	IPRG9	IPRG8	
	—	IPRG6	IPRG5	IPRG4	—	IPRG2	IPRG1	IPRG0	
IPRH	—	IPRH14	IPRH13	IPRH12	—	IPRH10	IPRH9	IPRH8	
	—	IPRH6	IPRH5	IPRH4	—	IPRH2	IPRH1	IPRH0	
IPRI	—	IPRI14	IPRI13	IPRI12	—	IPRI10	IPRI9	IPRI8	
	—	IPRI6	IPRI5	IPRI4	—	IPRI2	IPRI1	IPRI0	
IPRJ	—	IPRJ14	IPRJ13	IPRJ12	—	IPRJ10	IPRJ9	IPRJ8	
	—	IPRJ6	IPRJ5	IPRJ4	—	IPRJ2	IPRJ1	IPRJ0	
IPRK	—	IPRK14	IPRK13	IPRK12	—	IPRK10	IPRK9	IPRK8	
	—	IPRK6	IPRK5	IPRK4	—	IPRK2	IPRK1	IPRK0	
ITSR	—	—	—	—	—	—	—	—	
	—	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	
SSIER	—	—	—	—	—	—	—	—	
	—	SSI7	SSI6	SSI5	SSI4	SSI3	SSI2	SSI1	
ISCR	—	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
	—	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
IrCR_0	IrE	IrCKS2	IrCKS1	IrCKS0	—	—	—	—	IrDA_0

P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT
P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR	
P3DDR	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
P5DDR	—	—	—	—	P53DDR	P52DDR	P51DDR	P50DDR	
P8DDR	—	—	P85DDR	—	P83DDR	—	P81DDR	—	
PADDR	PA7DDR	PA6DDR	PA5DDR	PA4DDR	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR	
PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
PGDDR	—	PG6DDR	PG5DDR	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	
PFCR0	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E	
PFCR1	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E	
PFCR2	—	—	—	—	ASOE	LWROE	OES	—	
PAPCR	PA7PCR	PA6PCR	PA5PCR	PA4PCR	PA3PCR	PA2PCR	PA1PCR	PA0PCR	
PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR	
PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR	
PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR	
PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR	
P3ODR	—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR	
PAODR	PA7ODR	PA6ODR	PA5ODR	PA4ODR	PA3ODR	PA2ODR	PA1ODR	PA0ODR	
SMR_3 <sup>*4</sup>	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS1	CKS0	SCI_3
SMR_3 <sup>*5</sup>	GM	BLK	PE	O/ $\bar{E}$	BCP1	BCP0	CKS1	CKS0	Smart card interface 3
BRR_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_3 <sup>*4</sup>	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_3 <sup>*5</sup>	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_3	—	—	—	—	SDIR	SINV	—	SMIF	



SMR_4*4	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS1	CKS0	SCI_4
SMR_4*5	GM	BLK	PE	O/ $\bar{E}$	BCP1	BCP0	CKS1	CKS0	Smart card interface 4
BRR_4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_4	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_4*4	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_4*5	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_4	—	—	—	—	SDIR	SINV	—	SMIF	
TCR_3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_3
TMDR_3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_3	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_3	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_3	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_4
TMDR_4	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	

TCNT_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	TPU_4
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_4	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_5
TMDR_5	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_5	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	BSC
ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0	
WTCRAH	—	W72	W71	W70	—	W62	W61	W60	
WTCRAL	—	W52	W51	W50	—	W42	W41	W40	
WTCRBH	—	W32	W31	W30	—	W22	W21	W20	
WTCRBL	—	W12	W11	W10	—	W02	W01	W00	
RDNCR	RDN7	RDN6	RDN5	RDN4	RDN3	RDN2	RDN1	RDN0	
CSACRH	CSXH7	CSXH6	CSXH5	CSXH4	CSXH3	CSXH2	CSXH1	CSXH0	
CSACRL	CSXT7	CSXT6	CSXT5	CSXT4	CSXT3	CSXT2	CSXT1	CSXT0	
BROMCRH	BSRM0	BSTS02	BSTS01	BSTS00	—	—	BSWD01	BSWD00	
BROMCRL	BSRM1	BSTS12	BSTS11	BSTS10	—	—	BSWD11	BSWD10	

BCR	BRLE	BREQ0E	—	IDLC	ICIS1	ICIS0	WDBE	WAITE	BSC
	—	—	—	—	—	—	—	—	
DRAMCR	OEE	RAST	—	CAST	—	RMTS2	RMTS1	RMTS0	
	BE	RCDM	DDS	—	—	MXC2	MXC1	MXC0	
DRACCR	DRMI	—	TPC1	TPC0	—	—	RCD1	RCD0	
REFCR	CMF	CMIE	RCW1	RCW0	—	RTCK2	RTCK1	RTCK0	
	RFSHE	CBRM	RLW1	RLW0	SLFRF	TPCS2	TPCS1	TPCS0	
RTCNT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
RTCOR	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_0AH	—	—	—	—	—	—	—	—	DMAC
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_0AL	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IOAR_0A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
ETCR_0A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_0BH	—	—	—	—	—	—	—	—	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_0BL	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IOAR_0B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
ETCR_0B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_1AH	—	—	—	—	—	—	—	—	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_1AL	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IOAR_1A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

ETCR_1A	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	DMAC
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_1BH	—	—	—	—	—	—	—	—	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
MAR_1BL	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
IOARV1B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
ETCR_1B	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DMAWER	—	—	—	—	WE1B	WE1A	WE0B	WE0A	
DMATCR	—	—	TEE1	TEE0	—	—	—	—	
DMACR_0A <sup>*2</sup>	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_0A <sup>*3</sup>	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	
DMACR_0B <sup>*2</sup>	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_0B <sup>*3</sup>	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	
DMACR_1A <sup>*2</sup>	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_1A <sup>*3</sup>	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—	
DMACR_1B <sup>*2</sup>	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0	
DMACR_1B <sup>*3</sup>	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0	
DMABCRH <sup>*2</sup>	FAE1	FAE0	SAE1	SAE0	DTA1B	DTA1A	DTA0B	DTA0A	
DMABCRH <sup>*3</sup>	FAE1	FAE0	—	—	DTA1	—	DTA0	—	
DMABCRL <sup>*2</sup>	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A	
DMABCRL <sup>*3</sup>	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A	
DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC
DTCERB	—	—	—	—	—	—	—	—	
DTCERC	—	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0	
DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0	
DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0	
DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0	
DTCERG	DTCEG7	DTCEG6	DTCEG5	DTCEG4	DTCEG3	DTCEG2	—	—	
DTCERH	—	—	—	—	—	—	—	—	

DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	DTC
INTCR	—	—	INTM1	INTM0	NMIEG	—	—	—	INT
IER	—	—	—	—	—	—	—	—	
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
ISR	—	—	—	—	—	—	—	—	
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
SBYCR	SSBY	OPE	—	—	STS3	STS2	STS1	STS0	SYSTEM
SCKCR	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0	
SYSCR	—	—	—	—	FLSHE	—	EXPE	RAME	
MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	
MSTPCRH	ACSE	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	
MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	
EXMSTPCRH	—	—	—	—	MSTP27	MSTP26	MSTP25	MSTP24	
EXMSTPCRL	MSTP23	MSTP22	MSTP21	MSTP20	MSTP19	MSTP18	MSTP17	MSTP16	
PLLCR	—	—	—	—	—	—	STC1	STC0	
PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG
PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV	
NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8	
NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0	
PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8	
PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0	
NDRH <sup>*6</sup>	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	
NDRL <sup>*6</sup>	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	
NDRH <sup>*6</sup>	—	—	—	—	NDR11	NDR10	NDR9	NDR8	
NDRL <sup>*6</sup>	—	—	—	—	NDR3	NDR2	NDR1	NDR0	
PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT
PORT2	P27	P26	P25	P24	P23	P22	P21	P20	
PORT3	—	—	P35	P34	P33	P32	P31	P30	
PORT4	P47	P46	P45	P44	P43	P42	P41	P40	
PORT5	—	—	—	—	P53	P52	P51	P50	
PORT8	—	—	P85	—	P83	—	P81	—	
PORT9	—	—	P95	P94	—	—	—	—	

PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORT
PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	
PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	
PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	
PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	
PORTG	—	PG6	PG5	PG4	PG3	PG2	PG1	PG0	
P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	
P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR	
P3DR	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR	
P5DR	—	—	—	—	P53DR	P52DR	P51DR	P50DR	
P8DR	—	—	P85DR	—	P83DR	—	P81DR	—	
PADR	PA7DR	PA6DR	PA5DR	PA4DR	PA3DR	PA2DR	PA1DR	PA0DR	
PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	
PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR	
PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR	
PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR	
PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR	
PGDR	—	PG6DR	PG5DR	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR	
SMR_0 <sup>*4</sup>	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS1	CKS0	SCI_0, Smart card interface_0
SMR_0 <sup>*5</sup>	GM	BLK	PE	O/ $\bar{E}$	BCP1	BCP0	CKS1	CKS0	
BRR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_0 <sup>*4</sup>	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_0 <sup>*5</sup>	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_0	—	—	—	—	SDIR	SINV	—	SMIF	

SMR_1* <sup>4</sup>	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS1	CKS0	SCI_1, Smart card interface_1
SMR_1* <sup>5</sup>	GM	BLK	PE	O/ $\bar{E}$	BCP1	BCP0	OXS1	OXS0	
BRR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_1* <sup>4</sup>	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_1* <sup>5</sup>	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_1	—	—	—	—	SDIR	SINV	—	SMIF	
SMR_2* <sup>4</sup>	C/ $\bar{A}$	CHR	PE	O/ $\bar{E}$	STOP	MP	CKS1	CKS0	SCI_2, Smart card interface_2
SMR_2* <sup>5</sup>	GM	BLK	PE	O/ $\bar{E}$	BCP1	BCP0	CKS1	CKS0	
BRR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCR_2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
TDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SSR_2* <sup>4</sup>	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
SSR_2* <sup>5</sup>	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT	
RDR_2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
SCMR_2	—	—	—	—	SDIR	SINV	—	SMIF	
ADDRA	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
	AD1	AD0	—	—	—	—	—	—	
ADDRB	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRC	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRD	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRE	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRF	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRG	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	
ADDRH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
	AD1	AD0	—	—	—	—	—	—	

ADCSR	ADF	ADIE	ADST	—	CH3	CH2	CH1	CH0	A/D
ADCR	TRGS1	TRGS0	SCANE	SCANS	CKS1	CKS0	—	—	
DADR2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	D/A
DADR3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
DACR23	DAOE3	DAOE2	DAE	—	—	—	—	—	
TCR_0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_0
TCR_1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_1
TCSR_0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0	
TCSR_1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
TCORA_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCORA_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCORB_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCORB_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCNT_0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCNT_1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCSR	OVF	WT/IT	TME	—	—	CKS2	CKS1	CKS0	WDT
TCNT	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
RSTCSR	WOVF	RSTE	—	—	—	—	—	—	
TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU
TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
FCCS <sup>*7</sup>	—	—	—	FLER	—	—	—	SCO	FLASH
FPCS <sup>*7</sup>	—	—	—	PPVD	—	—	—	PPVS	
FECS <sup>*7</sup>	—	—	—	—	—	—	—	EPVB	
FLMCR1	—	SWE	ESU	PSU	EV	PV	E	P	
FKEY <sup>*7</sup>	K7	K6	K5	K4	K3	K2	K1	K0	
FLMCR2	FLER	—	—	—	—	—	—	—	
FMATS <sup>*7</sup>	MS7	MS6	MS5	MS4	MS3	MS2	MS1	MS0	
FTDAR <sup>*7</sup>	TDER	TDA6	TDA5	TDA4	TDA3	TDA2	TDA1	TDA0	
EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
EBR2	—	—	EB13	EB12	EB11	EB10	EB9	EB8	
FVACR <sup>*7</sup>	FVCHGE	—	—	—	FVSEL3	FVSEL2	FVSEL1	FVSEL0	



TCR_0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_0
TMDR_0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
TIORH_0	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIORL_0	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
TIER_0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
TSR_0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
TCNT_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRC_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRD_0	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TCR_1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_1
TMDR_1	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_1	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

TCR_2	—	CCLR1	CCLR0	TKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU_2
TMDR_2	—	—	—	—	MD3	MD2	MD1	MD0	
TIOR_2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
TIER_2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
TSR_2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
TCNT_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRA_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
TGRB_2	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

- Notes:
1. Loaded in on-chip RAM. The bus width is 32 bits when the DTC accesses this area as register information, and 16 bits otherwise.
  2. For short address mode
  3. For full address mode
  4. For normal mode
  5. For smart card interface mode
  6. If the pulse output group 2 and pulse output group 3 output triggers are the same according to the PCR setting, the NDRH address will be H'FF4C, and if different, the address of NDRH for group 2 will be H'FF4E, and that for group 3 will be H'FF4C. Similarly, if the pulse output group 0 and pulse output group 1 output triggers are the same according to the PCR setting, the NDRL address will be H'FF4D, and if different, the address of NDRL for group 0 will be H'FF4F, and that for group 1 will be H'FF4D.
  7. Supported only by the H8S/2368 0.18  $\mu\text{m}$  F-ZTAT Group.

Register Name	Reset	High-Speed	Clock Division	Sleep	Module Stop	All Module Clock Stop	Software Standby	Hardware Standby	Module	
MRA	Initialized	—	—	—	—	—	—	Initialized	DTC	
SAR	Initialized	—	—	—	—	—	—	Initialized		
MRB	Initialized	—	—	—	—	—	—	Initialized		
DAR	Initialized	—	—	—	—	—	—	Initialized		
CRA	Initialized	—	—	—	—	—	—	Initialized		
CRB	Initialized	—	—	—	—	—	—	Initialized		
ICCRA_0	Initialized	—	—	—	—	—	—	Initialized	IIC2_0	
ICCRB_0	Initialized	—	—	—	—	—	—	Initialized		
ICMR_0	Initialized	—	—	—	—	—	—	Initialized		
ICIER_0	Initialized	—	—	—	—	—	—	Initialized		
ICSR_0	Initialized	—	—	—	—	—	—	Initialized		
SAR_0	Initialized	—	—	—	—	—	—	Initialized		
ICDRT_0	Initialized	—	—	—	—	—	—	Initialized		
ICDRR_0	Initialized	—	—	—	—	—	—	Initialized		
ICCRA_1	Initialized	—	—	—	—	—	—	Initialized		IIC2_1
ICCRB_1	Initialized	—	—	—	—	—	—	Initialized		
ICMR_1	Initialized	—	—	—	—	—	—	Initialized		
ICIER_1	Initialized	—	—	—	—	—	—	Initialized		
ICSR_1	Initialized	—	—	—	—	—	—	Initialized		
SAR_1	Initialized	—	—	—	—	—	—	Initialized		
ICDRT_1	Initialized	—	—	—	—	—	—	Initialized		
ICDRR_1	Initialized	—	—	—	—	—	—	Initialized		
SEMR_2	Initialized	—	—	—	—	—	—	Initialized	SCI2	
IPRA	Initialized	—	—	—	—	—	—	Initialized	INT	
IPRB	Initialized	—	—	—	—	—	—	Initialized		
IPRC	Initialized	—	—	—	—	—	—	Initialized		
IPRD	Initialized	—	—	—	—	—	—	Initialized		
IPRE	Initialized	—	—	—	—	—	—	Initialized		
IPRF	Initialized	—	—	—	—	—	—	Initialized		

IPRG	Initialized	—	—	—	—	—	—	Initialized	INT
IPRH	Initialized	—	—	—	—	—	—	Initialized	
IPRI	Initialized	—	—	—	—	—	—	Initialized	
IPRJ	Initialized	—	—	—	—	—	—	Initialized	
IPRK	Initialized	—	—	—	—	—	—	Initialized	
ITSR	Initialized	—	—	—	—	—	—	Initialized	
SSIER	Initialized	—	—	—	—	—	—	Initialized	
ISCR	Initialized	—	—	—	—	—	—	Initialized	
IrCR_0	Initialized	—	—	—	—	—	—	Initialized	IrDA_0
P1DDR	Initialized	—	—	—	—	—	—	Initialized	PORT
P2DDR	Initialized	—	—	—	—	—	—	Initialized	
P3DDR	Initialized	—	—	—	—	—	—	Initialized	
P5DDR	Initialized	—	—	—	—	—	—	Initialized	
P8DDR	Initialized	—	—	—	—	—	—	Initialized	
PADDR	Initialized	—	—	—	—	—	—	Initialized	
PBDDR	Initialized	—	—	—	—	—	—	Initialized	
PCDDR	Initialized	—	—	—	—	—	—	Initialized	
PDDDR	Initialized	—	—	—	—	—	—	Initialized	
PEDDR	Initialized	—	—	—	—	—	—	Initialized	
PFDDR	Initialized	—	—	—	—	—	—	Initialized	
PGDDR	Initialized	—	—	—	—	—	—	Initialized	
PFCR0	Initialized	—	—	—	—	—	—	Initialized	
PFCR1	Initialized	—	—	—	—	—	—	Initialized	
PFCR2	Initialized	—	—	—	—	—	—	Initialized	
PAPCR	Initialized	—	—	—	—	—	—	Initialized	
PBPCR	Initialized	—	—	—	—	—	—	Initialized	
PCPCR	Initialized	—	—	—	—	—	—	Initialized	
PDPCR	Initialized	—	—	—	—	—	—	Initialized	
PEPCR	Initialized	—	—	—	—	—	—	Initialized	
P3ODR	Initialized	—	—	—	—	—	—	Initialized	
PAODR	Initialized	—	—	—	—	—	—	Initialized	

SMR_3	Initialized	—	—	—	—	—	—	—	Initialized	SCI_3
BRR_3	Initialized	—	—	—	—	—	—	—	Initialized	
SCR_3	Initialized	—	—	—	—	—	—	—	Initialized	
TDR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_3	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_3	Initialized	—	—	—	—	—	—	—	Initialized	
SMR_4	Initialized	—	—	—	—	—	—	—	Initialized	SCI_4
BRR_4	Initialized	—	—	—	—	—	—	—	Initialized	
SCR_4	Initialized	—	—	—	—	—	—	—	Initialized	
TDR_4	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_4	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_4	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_4	Initialized	—	—	—	—	—	—	—	Initialized	
TCR_3	Initialized	—	—	—	—	—	—	—	Initialized	TPU_3
TMDR_3	Initialized	—	—	—	—	—	—	—	Initialized	
TIORH_3	Initialized	—	—	—	—	—	—	—	Initialized	
TIORL_3	Initialized	—	—	—	—	—	—	—	Initialized	
TIER_3	Initialized	—	—	—	—	—	—	—	Initialized	
TSR_3	Initialized	—	—	—	—	—	—	—	Initialized	
TCNT_3	Initialized	—	—	—	—	—	—	—	Initialized	
TGRA_3	Initialized	—	—	—	—	—	—	—	Initialized	
TGRB_3	Initialized	—	—	—	—	—	—	—	Initialized	
TGRC_3	Initialized	—	—	—	—	—	—	—	Initialized	
TGRD_3	Initialized	—	—	—	—	—	—	—	Initialized	
TCR_4	Initialized	—	—	—	—	—	—	—	Initialized	TPU_4
TMDR_4	Initialized	—	—	—	—	—	—	—	Initialized	
TIOR_4	Initialized	—	—	—	—	—	—	—	Initialized	
TIER_4	Initialized	—	—	—	—	—	—	—	Initialized	
TSR_4	Initialized	—	—	—	—	—	—	—	Initialized	

TCNT_4	Initialized	—	—	—	—	—	—	Initialized	TPU_4
TGRA_4	Initialized	—	—	—	—	—	—	Initialized	
TGRB_4	Initialized	—	—	—	—	—	—	Initialized	
TCR_5	Initialized	—	—	—	—	—	—	Initialized	TPU_5
TMDR_5	Initialized	—	—	—	—	—	—	Initialized	
TIOR_5	Initialized	—	—	—	—	—	—	Initialized	
TIER_5	Initialized	—	—	—	—	—	—	Initialized	
TSR_5	Initialized	—	—	—	—	—	—	Initialized	
TCNT_5	Initialized	—	—	—	—	—	—	Initialized	
TGRA_5	Initialized	—	—	—	—	—	—	Initialized	
TGRB_5	Initialized	—	—	—	—	—	—	Initialized	
ABWCR	Initialized	—	—	—	—	—	—	Initialized	BSC
ASTCR	Initialized	—	—	—	—	—	—	Initialized	
WTCRAH	Initialized	—	—	—	—	—	—	Initialized	
WTCRAL	Initialized	—	—	—	—	—	—	Initialized	
WTCRBH	Initialized	—	—	—	—	—	—	Initialized	
WTCRBL	Initialized	—	—	—	—	—	—	Initialized	
RDNCR	Initialized	—	—	—	—	—	—	Initialized	
CSACRH	Initialized	—	—	—	—	—	—	Initialized	
CSACRL	Initialized	—	—	—	—	—	—	Initialized	
BROMCRH	Initialized	—	—	—	—	—	—	Initialized	
BROMCRL	Initialized	—	—	—	—	—	—	Initialized	
BCR	Initialized	—	—	—	—	—	—	Initialized	
DRAMCR	Initialized	—	—	—	—	—	—	Initialized	
DRACCR	Initialized	—	—	—	—	—	—	Initialized	
REFCR	Initialized	—	—	—	—	—	—	Initialized	
RTCNT	Initialized	—	—	—	—	—	—	Initialized	
RTCOR	Initialized	—	—	—	—	—	—	Initialized	

MAR_0AH	Initialized	—	—	—	—	—	—	Initialized	DMAC
MAR_0AL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_0A	Initialized	—	—	—	—	—	—	Initialized	
ETCR_0A	Initialized	—	—	—	—	—	—	Initialized	
MAR_0BH	Initialized	—	—	—	—	—	—	Initialized	
MAR_0BL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_0B	Initialized	—	—	—	—	—	—	Initialized	
ETCR_0B	Initialized	—	—	—	—	—	—	Initialized	
MAR_1AH	Initialized	—	—	—	—	—	—	Initialized	
MAR_1AL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_1A	Initialized	—	—	—	—	—	—	Initialized	
ETCR_1A	Initialized	—	—	—	—	—	—	Initialized	
MAR_1BH	Initialized	—	—	—	—	—	—	Initialized	
MAR_1BL	Initialized	—	—	—	—	—	—	Initialized	
IOAR_1B	Initialized	—	—	—	—	—	—	Initialized	
ETCR_1B	Initialized	—	—	—	—	—	—	Initialized	
DMAWER	Initialized	—	—	—	—	—	—	Initialized	
DMATCR	Initialized	—	—	—	—	—	—	Initialized	
DMACR_0A	Initialized	—	—	—	—	—	—	Initialized	
DMACR_0B	Initialized	—	—	—	—	—	—	Initialized	
DMACR_1A	Initialized	—	—	—	—	—	—	Initialized	
DMACR_1B	Initialized	—	—	—	—	—	—	Initialized	
DMABCRH	Initialized	—	—	—	—	—	—	Initialized	
DMABCRL	Initialized	—	—	—	—	—	—	Initialized	
DTCERA	Initialized	—	—	—	—	—	—	Initialized	DTC
DTCERB	Initialized	—	—	—	—	—	—	Initialized	
DTCERC	Initialized	—	—	—	—	—	—	Initialized	
DTCERD	Initialized	—	—	—	—	—	—	Initialized	
DTCERE	Initialized	—	—	—	—	—	—	Initialized	
DTCERF	Initialized	—	—	—	—	—	—	Initialized	
DTCERG	Initialized	—	—	—	—	—	—	Initialized	

DTVECH	Initialized	—	—	—	—	—	—	Initialized	DTC
DTVECR	Initialized	—	—	—	—	—	—	Initialized	
INTCR	Initialized	—	—	—	—	—	—	Initialized	INT
IER	Initialized	—	—	—	—	—	—	Initialized	
ISR	Initialized	—	—	—	—	—	—	Initialized	
SBYCR	Initialized	—	—	—	—	—	—	Initialized	SYSTEM
SCKCR	Initialized	—	—	—	—	—	—	Initialized	
SYSCR	Initialized	—	—	—	—	—	—	Initialized	
MDCR	Initialized	—	—	—	—	—	—	Initialized	
MSTPCRH	Initialized	—	—	—	—	—	—	Initialized	
MSTPCRL	Initialized	—	—	—	—	—	—	Initialized	
EXMSTPCRH	Initialized	—	—	—	—	—	—	Initialized	
EXMSTPCRL	Initialized	—	—	—	—	—	—	Initialized	
PLLCR	Initialized	—	—	—	—	—	—	Initialized	
PCR	Initialized	—	—	—	—	—	—	Initialized	PPG
PMR	Initialized	—	—	—	—	—	—	Initialized	
NDERH	Initialized	—	—	—	—	—	—	Initialized	
NDERL	Initialized	—	—	—	—	—	—	Initialized	
PODRH	Initialized	—	—	—	—	—	—	Initialized	
PODRL	Initialized	—	—	—	—	—	—	Initialized	
NDRH	Initialized	—	—	—	—	—	—	Initialized	
NDRL	Initialized	—	—	—	—	—	—	Initialized	
NDRH	Initialized	—	—	—	—	—	—	Initialized	
NDRL	Initialized	—	—	—	—	—	—	Initialized	
PORT1	—	—	—	—	—	—	—	—	PORT
PORT2	—	—	—	—	—	—	—	—	
PORT3	—	—	—	—	—	—	—	—	
PORT4	—	—	—	—	—	—	—	—	
PORT5	—	—	—	—	—	—	—	—	
PORT8	—	—	—	—	—	—	—	—	
PORT9	—	—	—	—	—	—	—	—	



PORTA	—	—	—	—	—	—	—	—	PORT
PORTB	—	—	—	—	—	—	—	—	
PORTC	—	—	—	—	—	—	—	—	
PORTD	—	—	—	—	—	—	—	—	
PORTE	—	—	—	—	—	—	—	—	
PORTF	—	—	—	—	—	—	—	—	
PORTG	—	—	—	—	—	—	—	—	
P1DR	Initialized	—	—	—	—	—	—	Initialized	
P2DR	Initialized	—	—	—	—	—	—	Initialized	
P3DR	Initialized	—	—	—	—	—	—	Initialized	
P5DR	Initialized	—	—	—	—	—	—	Initialized	
P8DR	Initialized	—	—	—	—	—	—	Initialized	
PADR	Initialized	—	—	—	—	—	—	Initialized	
PBDR	Initialized	—	—	—	—	—	—	Initialized	
PCDR	Initialized	—	—	—	—	—	—	Initialized	
PDDR	Initialized	—	—	—	—	—	—	Initialized	
PEDR	Initialized	—	—	—	—	—	—	Initialized	
PFDR	Initialized	—	—	—	—	—	—	Initialized	
PGDR	Initialized	—	—	—	—	—	—	Initialized	
SMR_0	Initialized	—	—	—	—	—	—	Initialized	SCI_0
BRR_0	Initialized	—	—	—	—	—	—	Initialized	
SCR_0	Initialized	—	—	—	—	—	—	Initialized	
TDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SSR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
RDR_0	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	
SCMR_0	Initialized	—	—	—	—	—	—	Initialized	

SMR_1	Initialized	—	—	—	—	—	—	—	Initialized	SCI_1
BRR_1	Initialized	—	—	—	—	—	—	—	Initialized	
SCR_1	Initialized	—	—	—	—	—	—	—	Initialized	
TDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_1	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_1	Initialized	—	—	—	—	—	—	—	Initialized	
SMR_2	Initialized	—	—	—	—	—	—	—	Initialized	SCI_2
BRR_2	Initialized	—	—	—	—	—	—	—	Initialized	
SCR_2	Initialized	—	—	—	—	—	—	—	Initialized	
TDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_2	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_2	Initialized	—	—	—	—	—	—	—	Initialized	
ADDRA	Initialized	—	—	—	—	—	—	—	Initialized	A/D
ADDRB	Initialized	—	—	—	—	—	—	—	Initialized	
ADDRC	Initialized	—	—	—	—	—	—	—	Initialized	
ADDRD	Initialized	—	—	—	—	—	—	—	Initialized	
ADDRE	Initialized	—	—	—	—	—	—	—	Initialized	
ADDRF	Initialized	—	—	—	—	—	—	—	Initialized	
ADDRG	Initialized	—	—	—	—	—	—	—	Initialized	
ADDRH	Initialized	—	—	—	—	—	—	—	Initialized	
ADCSR	Initialized	—	—	—	—	—	—	—	Initialized	
ADCR	Initialized	—	—	—	—	—	—	—	Initialized	
DADR2	Initialized	—	—	—	—	—	—	—	Initialized	D/A
DADR3	Initialized	—	—	—	—	—	—	—	Initialized	
DACR23	Initialized	—	—	—	—	—	—	—	Initialized	

TCR_0	Initialized	—	—	—	—	—	—	Initialized	TMR_0
TCR_1	Initialized	—	—	—	—	—	—	Initialized	TMR_1
TCSR_0	Initialized	—	—	—	—	—	—	Initialized	
TCSR_1	Initialized	—	—	—	—	—	—	Initialized	
TCORA_0	Initialized	—	—	—	—	—	—	Initialized	
TCORA_1	Initialized	—	—	—	—	—	—	Initialized	
TCORB_0	Initialized	—	—	—	—	—	—	Initialized	
TCORB_1	Initialized	—	—	—	—	—	—	Initialized	
TCNT_0	Initialized	—	—	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	—	—	Initialized	
TCSR	Initialized	—	—	—	—	—	—	Initialized	WDT
TCNT	Initialized	—	—	—	—	—	—	Initialized	
RSTCSR	Initialized	—	—	—	—	—	—	Initialized	
TSTR	Initialized	—	—	—	—	—	—	Initialized	TPU
TSYR	Initialized	—	—	—	—	—	—	Initialized	
FCCS*	Initialized	—	—	—	—	—	—	Initialized	FLASH
FPCS*	Initialized	—	—	—	—	—	—	Initialized	
FECS*	Initialized	—	—	—	—	—	—	Initialized	
FLMCR1	Initialized	—	—	—	—	—	—	Initialized	
FKEY*	Initialized	—	—	—	—	—	—	Initialized	
FLMCR2	Initialized	—	—	—	—	—	—	Initialized	
FMATS*	Initialized	—	—	—	—	—	—	Initialized	
FTDAR*	Initialized	—	—	—	—	—	—	Initialized	
EBR1	Initialized	—	—	—	—	—	—	Initialized	
EBR2	Initialized	—	—	—	—	—	—	Initialized	
FVACR*	Initialized	—	—	—	—	—	—	Initialized	

TCR_0	Initialized	—	—	—	—	—	—	Initialized	TPU_0
TMDR_0	Initialized	—	—	—	—	—	—	Initialized	
TIORH_0	Initialized	—	—	—	—	—	—	Initialized	
TIORL_0	Initialized	—	—	—	—	—	—	Initialized	
TIER_0	Initialized	—	—	—	—	—	—	Initialized	
TSR_0	Initialized	—	—	—	—	—	—	Initialized	
TCNT_0	Initialized	—	—	—	—	—	—	Initialized	
TGRA_0	Initialized	—	—	—	—	—	—	Initialized	
TGRB_0	Initialized	—	—	—	—	—	—	Initialized	
TGRC_0	Initialized	—	—	—	—	—	—	Initialized	
TGRD_0	Initialized	—	—	—	—	—	—	Initialized	
TCR_1	Initialized	—	—	—	—	—	—	Initialized	TPU_1
TMDR_1	Initialized	—	—	—	—	—	—	Initialized	
TIOR_1	Initialized	—	—	—	—	—	—	Initialized	
TIER_1	Initialized	—	—	—	—	—	—	Initialized	
TSR_1	Initialized	—	—	—	—	—	—	Initialized	
TCNT_1	Initialized	—	—	—	—	—	—	Initialized	
TGRA_1	Initialized	—	—	—	—	—	—	Initialized	
TGRB_1	Initialized	—	—	—	—	—	—	Initialized	
TCR_2	Initialized	—	—	—	—	—	—	Initialized	TPU_2
TMDR_2	Initialized	—	—	—	—	—	—	Initialized	
TIOR_2	Initialized	—	—	—	—	—	—	Initialized	
TIER_2	Initialized	—	—	—	—	—	—	Initialized	
TSR_2	Initialized	—	—	—	—	—	—	Initialized	
TCNT_2	Initialized	—	—	—	—	—	—	Initialized	
TGRA_2	Initialized	—	—	—	—	—	—	Initialized	
TGRB_2	Initialized	—	—	—	—	—	—	Initialized	

Note: \* Supported only by the H8S/2368 0.18 μm F-ZTAT Group.

## 25.1 Electrical Characteristics of Masked ROM and ROMless Versions

### 25.1.1 Absolute Maximum Ratings

Table 25.1 lists the absolute maximum ratings.

**Table 25.1 Absolute Maximum Ratings**

Item	Symbol	Value	Unit
Power supply voltage	$V_{CC}$ $PLL V_{CC}$	-0.3 to +4.0	V
Input voltage (except ports 4, 9)	$V_{in}$	-0.3 to $V_{CC} + 0.3$	V
Input voltage (ports 4, 9)	$V_{in}$	-0.3 to $AV_{CC} + 0.3$	V
Reference power supply voltage	$V_{ref}$	-0.3 to $AV_{CC} + 0.3$	V
Analog power supply voltage	$AV_{CC}$	-0.3 to +4.0	V
Analog input voltage	$V_{AN}$	-0.3 to $AV_{CC} + 0.3$	V
Operating temperature	$T_{opr}$	Regular specifications: -20 to +75	°C
		Wide-range specifications: -40 to +85	°C
Storage temperature	$T_{stg}$	-55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

## Table 25.2 DC Characteristics (1)

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0\text{ V}^{*1}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions	
Schmitt trigger input voltage	Ports 1, 2, and 4 <sup>*2</sup> , P50 to P53 <sup>*2</sup> , PA4 to PA7 <sup>*2</sup>	$VT^-$	$V_{CC} \times 0.2$	—	—	V	
		$VT^+$	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{STBY}$ , MD2 to MD0	$V_{IH}$	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	$\overline{RES}$ , NMI, EMLE		$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Port 3, P50 to P53 <sup>*3</sup> , port 8, ports A to G <sup>*3</sup>		2.2V	—	$V_{CC} + 0.3$	V	
	Ports 4 and 9		2.2V	—	$V_{CC} + 0.3$	V	
Input low voltage	$\overline{RES}$ , $\overline{STBY}$ , MD2 to MD0, EMLE	$V_{IL}$	-0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL		-0.3	—	$V_{CC} \times 0.2$	V	
	Ports 3 to 5 <sup>*3</sup> , 8, 9, A to G <sup>*3</sup>		-0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	$V_{OH}$	$V_{CC} - 0.5$	—	—	V	$I_{OH} = -200\ \mu\text{A}$
			$V_{CC} - 1.0$	—	—	V	$I_{OH} = -1\ \text{mA}$
Output low voltage	All output pins P32 to P35 <sup>*4</sup>	$V_{OL}$	—	—	0.4	V	$I_{OL} = 1.6\ \text{mA}$
			—	—	0.5	V	$I_{OL} = 8.0\ \text{mA}$

- Notes: 1. When the A/D and D/A converters are not used, do not leave the  $AV_{CC}$ ,  $V_{ref}$ , and  $AV_{SS}$  pins open. Connect the  $AV_{CC}$  and  $V_{ref}$  pins to  $V_{CC}$ , and the  $AV_{SS}$  pin to  $V_{SS}$ .
2. When used as  $\overline{IRQ0}$  to  $\overline{IRQ7}$ .
3. When used as other than  $\overline{IRQ0}$  to  $\overline{IRQ7}$ .
4. When used as SCL0 to SCL1, SDA0 to SDA1.

Conditions:  $V_{CC} = 5.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V to }5.5\text{ V}$ ,  $V_{ref} = 5.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0\text{ V}^{*1}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input leakage current	$\overline{\text{RES}}$	$ I_{in} $	—	—	10.0	$\mu\text{A}$	$V_{in} = 0.5\text{ to }V_{CC} - 0.5\text{ V}$
	$\overline{\text{STBY}}$ , NMI, MD2 to MD0		—	—	1.0	$\mu\text{A}$	
	Ports 4 and 9		—	—	1.0	$\mu\text{A}$	$V_{in} = 0.5\text{ to }AV_{CC} - 0.5\text{ V}$
Three-state leakage current (off state)	Ports 1 to 3, P50 to P53, ports 8, A to G	$ I_{TSL} $	—	—	1.0	$\mu\text{A}$	$V_{in} = 0.5\text{ to }V_{CC} - 0.5\text{ V}$
Input pull-up MOS current	Ports A to E	$-I_p$	10	—	300	$\mu\text{A}$	$V_{CC} = 3.0\text{ to }3.6\text{ V}$ $V_{in} = 0\text{ V}$
Input capacitance	$\overline{\text{RES}}$	$C_{in}$	—	—	30	pF	$V_{in} = 0\text{ V}$
	NMI		—	—	30	pF	$f = 1\text{ MHz}$
	All input pins except $\overline{\text{RES}}$ and NMI		—	—	15	pF	$T_a = 25^\circ\text{C}$
Current dissipation <sup>*2</sup>	Normal operation	$I_{CC}^{*4}$	—	75 (3.3 V)	115	$\text{mA}$	$f = 33\text{ MHz}$
	Sleep mode		—	55 (3.3 V)	95	$\text{mA}$	$f = 33\text{ MHz}$
	Standby mode <sup>*3</sup>		—	0.01	10	$\mu\text{A}$	$T_a \leq 50^\circ\text{C}$
			—	—	80	$\mu\text{A}$	$50^\circ\text{C} < T_a$
Analog power supply current	During A/D and D/A conversion	$AI_{CC}$	—	0.3 (3.0 V)	2.0	$\text{mA}$	
	Idle		—	0.01	5.0	$\mu\text{A}$	

Reference power supply current	During A/D and D/A conversion Idle	$I_{CC}$	—	2.0 (3.0 V)	3.5	mA
RAM standby voltage		$V_{RAM}$	2.0	—	—	V

- Notes:
- When the A/D and D/A converters are not used, do not leave the  $AV_{CC}$ ,  $V_{ref}$ , and  $AV_{SS}$  pins open. Connect the  $AV_{CC}$  and  $V_{ref}$  pins to  $V_{CC}$ , and the  $AV_{SS}$  pin to  $V_{SS}$ .
  - Current dissipation values are for  $V_{IH,min} = V_{CC} - 0.2$  V and  $V_{IL,max} = 0.2$  V with all output pins unloaded and all MOS input pull-ups in the off state.
  - The values are for  $V_{RAM} \leq V_{CC} < 3.0$  V,  $V_{IH,min} = V_{CC} \times 0.9$ , and  $V_{IL,max} = 0.3$  V.
  - $I_{CC}$  depends on  $V_{CC}$  and  $f$  as follows:  
 $I_{CC,max} = 1.0$  (mA) +  $0.95$  (mA/(MHz  $\times$  V))  $\times V_{CC} \times f$  (normal operation)  
 $I_{CC,max} = 1.0$  (mA) +  $0.8$  (mA/(MHz  $\times$  V))  $\times V_{CC} \times f$  (sleep mode)

### Table 25.4 Permissible Output Currents

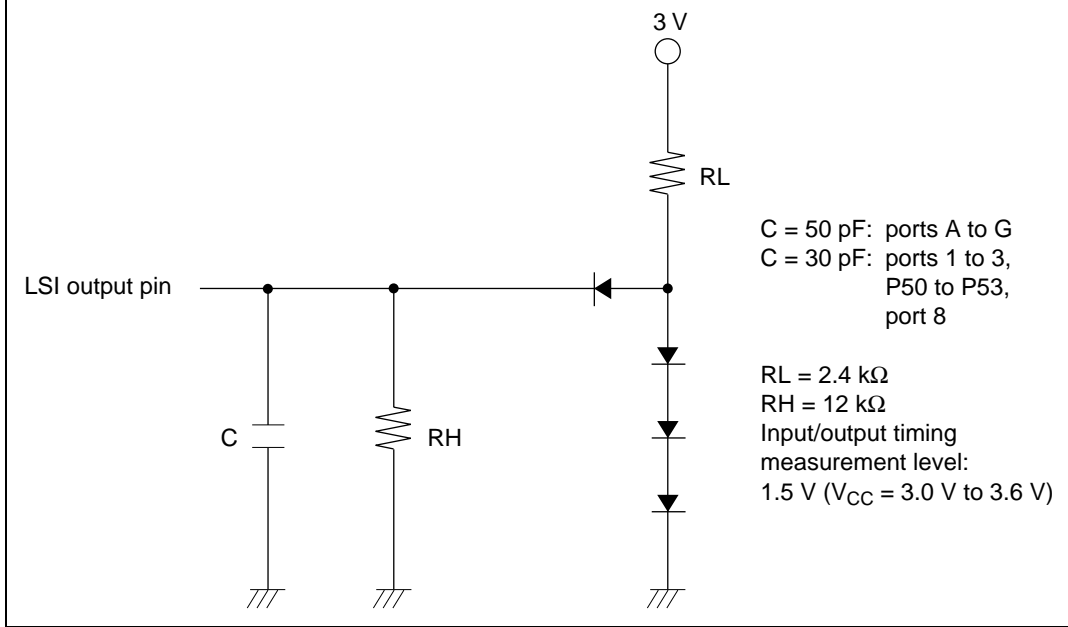
Conditions:  $V_{CC} = 3.0$  V to 3.6 V,  $AV_{CC} = 3.0$  V to 3.6 V,  $V_{ref} = 3.0$  V to  $AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0$  V\*,  $T_a = -20^\circ\text{C}$  to  $+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C}$  to  $+85^\circ\text{C}$  (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	SCL0, 1, SDA0, 1	$I_{OL}$	—	—	8.0	mA
	Output pins other than the above		—	—	2.0	
Permissible output low current (total)	Total of all output pins	$\Sigma I_{OL}$	—	—	80	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	40	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 25.4.

Note: \* When the A/D and D/A converters are not used, do not leave the  $AV_{CC}$ ,  $V_{ref}$ , and  $AV_{SS}$  pins open. Connect the  $AV_{CC}$  and  $V_{ref}$  pins to  $V_{CC}$ , and the  $AV_{SS}$  pin to  $V_{SS}$ .



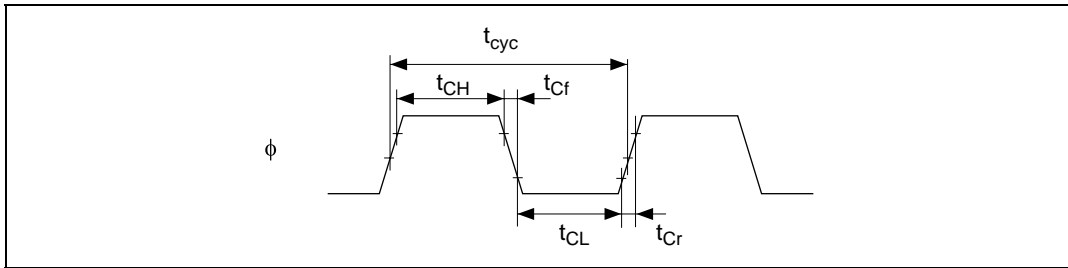


**Figure 25.1 Output Load Circuit**

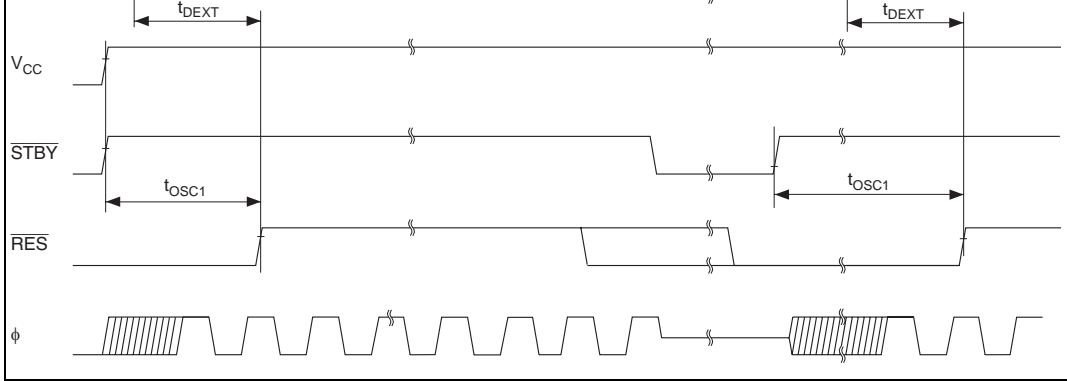
**Table 25.5 Clock Timing**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

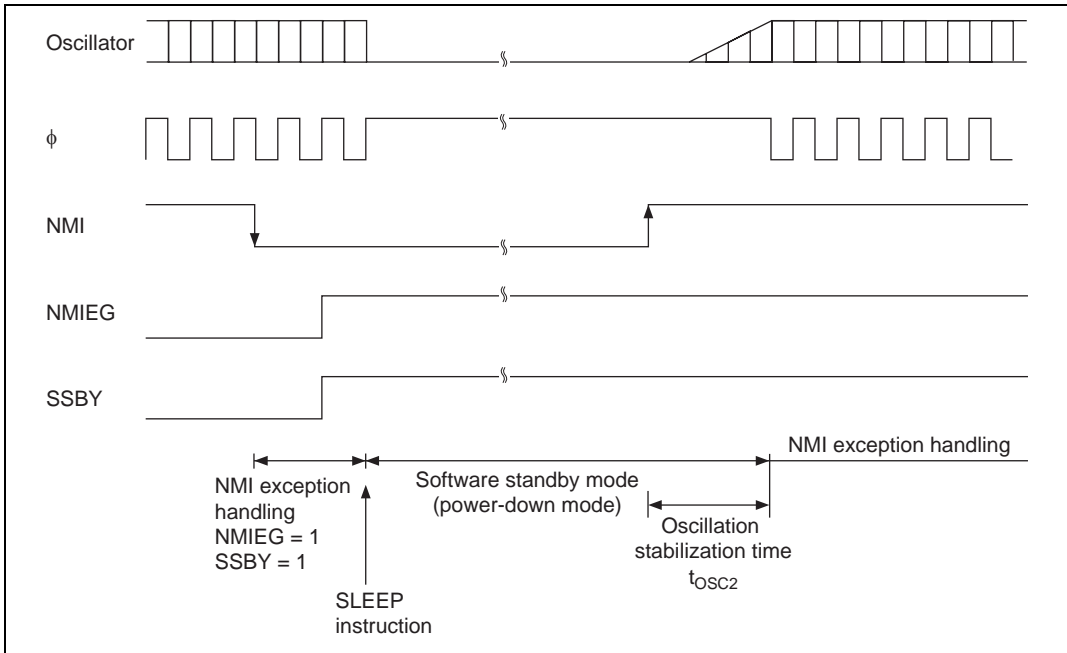
Item	Symbol	Min	Max	Unit	Test Conditions
Clock cycle time	$t_{cyc}$	30.3	125	ns	Figure 25.2
Clock pulse high width	$t_{CH}$	10	—	ns	Figure 25.2
Clock pulse low width	$t_{CL}$	10	—	ns	
Clock rise time	$t_{Cr}$	—	5	ns	
Clock fall time	$t_{Cf}$	—	5	ns	
Reset oscillation stabilization time (crystal)	$t_{OSC1}$	10	—	ms	Figure 25.3(1)
Software standby oscillation stabilization time (crystal)	$t_{OSC2}$	10	—	ms	Figure 25.3(2)
External clock output delay stabilization time	$t_{DEXT}$	1	—	ms	Figure 25.3(1)



**Figure 25.2 System Clock Timing**



**Figure 25.3 Oscillation Stabilization Timing (1)**



**Figure 25.3 Oscillation Stabilization Timing (2)**

## Table 25.6 Control Signal Timing

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{\text{RES}}$ setup time	$t_{\text{RESS}}$	200	—	ns	Figure 25.4
$\overline{\text{RES}}$ pulse width	$t_{\text{RESW}}$	20	—	$t_{\text{cyc}}$	
NMI setup time	$t_{\text{NMIS}}$	150	—	ns	Figure 25.5
NMI hold time	$t_{\text{NMIH}}$	10	—		
NMI pulse width (in recovery from software standby mode)	$t_{\text{NMIW}}$	200	—		
$\overline{\text{IRQ}}$ setup time	$t_{\text{IRQS}}$	150	—	ns	
$\overline{\text{IRQ}}$ hold time	$t_{\text{IROH}}$	10	—		
$\overline{\text{IRQ}}$ pulse width (in recovery from software standby mode)	$t_{\text{IRQW}}$	200	—		

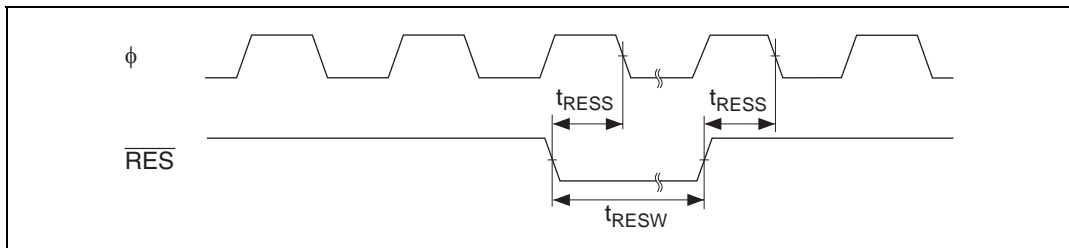
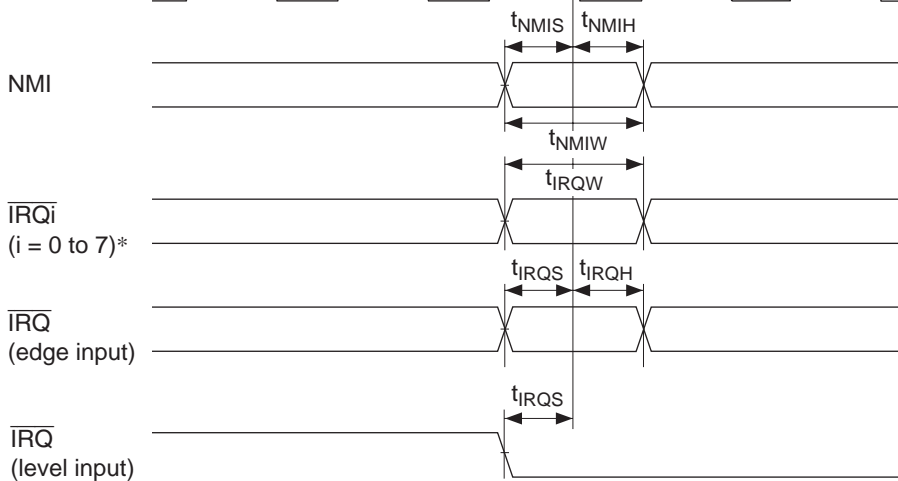


Figure 25.4 Reset Input Timing



Note: \* Necessary for SSIER setting to clear software standby mode.

**Figure 25.5 Interrupt Input Timing**

**Table 25.7 Bus Timing (1)**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
Address delay time	$t_{AD}$	—	20	ns	Figures 25.6 to 25.19
Address setup time 1	$t_{AS1}$	$0.5 \times t_{cyc} - 13$	—	ns	
Address setup time 2	$t_{AS2}$	$1.0 \times t_{cyc} - 13$	—	ns	
Address setup time 3	$t_{AS3}$	$1.5 \times t_{cyc} - 13$	—	ns	
Address setup time 4	$t_{AS4}$	$2.0 \times t_{cyc} - 13$	—	ns	
Address hold time 1	$t_{AH1}$	$0.5 \times t_{cyc} - 8$	—	ns	
Address hold time 2	$t_{AH2}$	$1.0 \times t_{cyc} - 8$	—	ns	
Address hold time 3	$t_{AH3}$	$1.5 \times t_{cyc} - 8$	—	ns	
$\overline{CS}$ delay time 1	$t_{CSD1}$	—	15	ns	
$\overline{CS}$ delay time 2	$t_{CSD2}$	—	15	ns	
$\overline{CS}$ delay time 3	$t_{CSD3}$	—	20	ns	
$\overline{AS}$ delay time	$t_{ASD}$	—	15	ns	
$\overline{RD}$ delay time 1	$t_{RSD1}$	—	15	ns	
$\overline{RD}$ delay time 2	$t_{RSD2}$	—	15	ns	
Read data setup time 1	$t_{RDS1}$	15	—	ns	
Read data setup time 2	$t_{RDS2}$	15	—	ns	
Read data hold time 1	$t_{RDH1}$	0	—	ns	
Read data hold time 2	$t_{RDH2}$	0	—	ns	
Read data access time 1	$t_{AC1}$	—	$1.0 \times t_{cyc} - 20$	ns	
Read data access time 2	$t_{AC2}$	—	$1.5 \times t_{cyc} - 20$	ns	
Read data access time 3	$t_{AC3}$	—	$2.0 \times t_{cyc} - 20$	ns	
Read data access time 4	$t_{AC4}$	—	$2.5 \times t_{cyc} - 20$	ns	
Read data access time 5	$t_{AC5}$	—	$1.0 \times t_{cyc} - 20$	ns	
Read data access time 6	$t_{AC6}$	—	$2.0 \times t_{cyc} - 20$	ns	
Read data access time 7	$t_{AC7}$	—	$4.0 \times t_{cyc} - 20$	ns	
Read data access time 8	$t_{AC8}$	—	$3.0 \times t_{cyc} - 20$	ns	

---

Address read data access time 2	$t_{AA2}$	—	$1.5 \times t_{cyc} - 20$	ns
Address read data access time 3	$t_{AA3}$	—	$2.0 \times t_{cyc} - 20$	ns
Address read data access time 4	$t_{AA4}$	—	$2.5 \times t_{cyc} - 20$	ns
Address read data access time 5	$t_{AA5}$	—	$3.0 \times t_{cyc} - 20$	ns

---

Conditions:  $V_{CC} = 3.0\text{ V to }5.0\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }5.0\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{WR}$ delay time 1	$t_{WRD1}$	—	15	ns	Figures 25.6 to 25.19
$\overline{WR}$ delay time 2	$t_{WRD2}$	—	15	ns	
$\overline{WR}$ pulse width 1	$t_{WSW1}$	$1.0 \times t_{cyc} - 13$	—	ns	
$\overline{WR}$ pulse width 2	$t_{WSW2}$	$1.5 \times t_{cyc} - 13$	—	ns	
Write data delay time	$t_{WDD}$	—	20	ns	
Write data setup time 1	$t_{WDS1}$	$0.5 \times t_{cyc} - 13$	—	ns	
Write data setup time 2	$t_{WDS2}$	$1.0 \times t_{cyc} - 13$	—	ns	
Write data setup time 3	$t_{WDS3}$	$1.5 \times t_{cyc} - 13$	—	ns	
Write data hold time 1	$t_{WDH1}$	$0.5 \times t_{cyc} - 8$	—	ns	
Write data hold time 2	$t_{WDH2}$	$1.0 \times t_{cyc} - 8$	—	ns	
Write data hold time 3	$t_{WDH3}$	$1.5 \times t_{cyc} - 8$	—	ns	
Write command setup time 1	$t_{WCS1}$	$0.5 \times t_{cyc} - 10$	—	ns	
Write command setup time 2	$t_{WCS2}$	$1.0 \times t_{cyc} - 10$	—	ns	
Write command hold time 1	$t_{WCH1}$	$0.5 \times t_{cyc} - 10$	—	ns	
Write command hold time 2	$t_{WCH2}$	$1.0 \times t_{cyc} - 10$	—	ns	
Read command setup time 1	$t_{RCS1}$	$1.5 \times t_{cyc} - 10$	—	ns	
Read command setup time 2	$t_{RCS2}$	$2.0 \times t_{cyc} - 10$	—	ns	
Read command hold time	$t_{RCH}$	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ delay time 1	$t_{CASD1}$	—	15	ns	
$\overline{CAS}$ delay time 2	$t_{CASD2}$	—	15	ns	
$\overline{CAS}$ setup time 1	$t_{CSR1}$	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ setup time 2	$t_{CSR2}$	$1.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ pulse width 1	$t_{CASW1}$	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ pulse width 2	$t_{CASW2}$	$1.5 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ precharge time 1	$t_{CPW1}$	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ precharge time 2	$t_{CPW2}$	$1.5 \times t_{cyc} - 20$	—	ns	
$\overline{OE}$ delay time 1	$t_{OED1}$	—	15	ns	
$\overline{OE}$ delay time 2	$t_{OED2}$	—	15	ns	



Precharge time 2	$t_{PCH2}$	$1.5 \times t_{cyc} - 20$	—	ns	25.19
Self-refresh precharge time 1	$t_{RPS1}$	$2.5 \times t_{cyc} - 20$	—	ns	Figures 25.20 and 25.21
Self-refresh precharge time 2	$t_{RPS2}$	$3.0 \times t_{cyc} - 20$	—	ns	
$\overline{WAIT}$ setup time	$t_{WTS}$	25	—	ns	Figures 25.8 and 25.14
$\overline{WAIT}$ hold time	$t_{WTH}$	5	—	ns	
$\overline{BREQ}$ setup time	$t_{BREQS}$	30	—	ns	Figure 25.22
$\overline{BACK}$ delay time	$t_{BACD}$	—	15	ns	
Bus floating time	$t_{BZD}$	—	40	ns	
$\overline{BREQO}$ delay time	$t_{BRQOD}$	—	25	ns	Figure 25.23

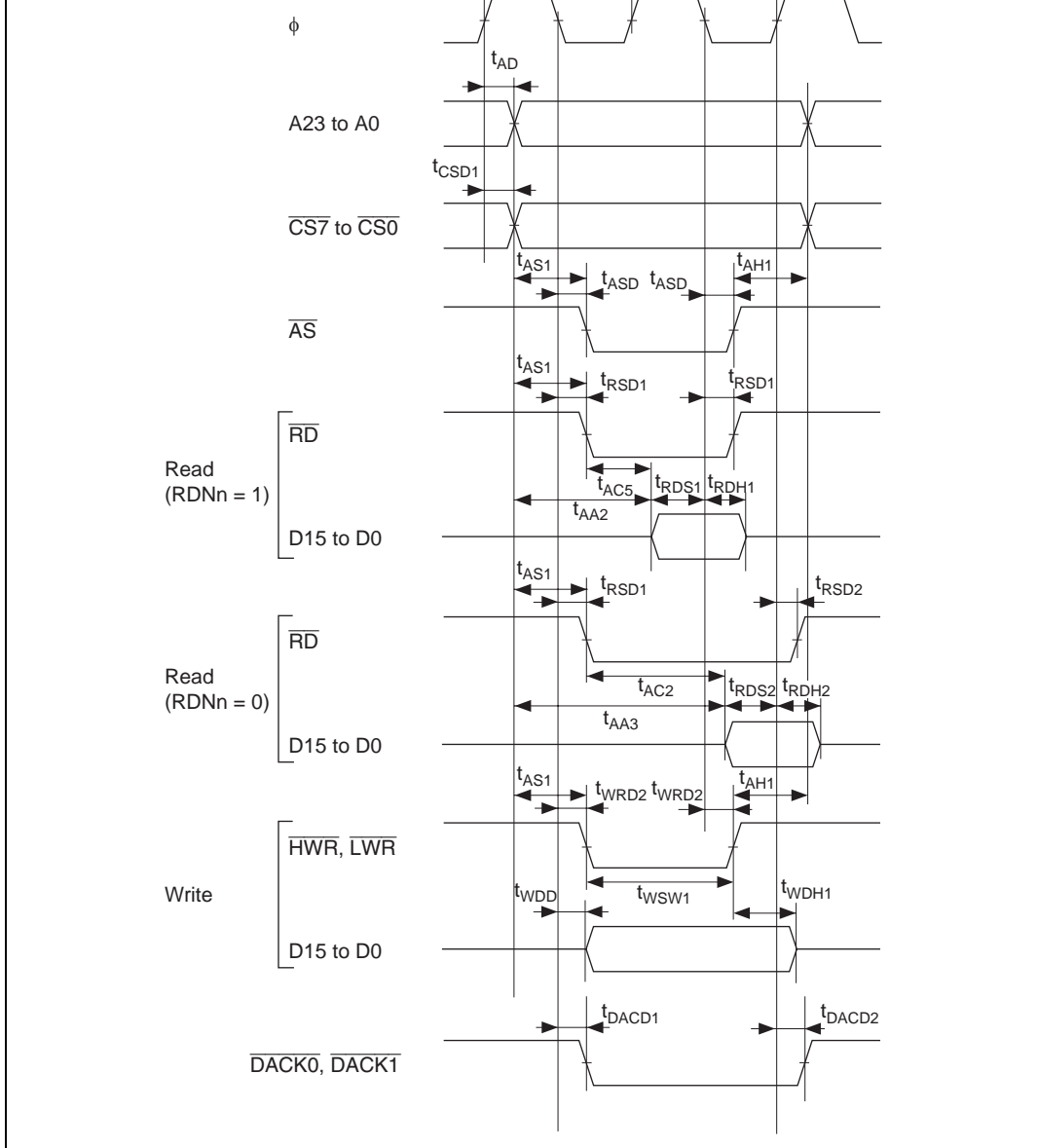


Figure 25.6 Basic Bus Timing: Two-State Access

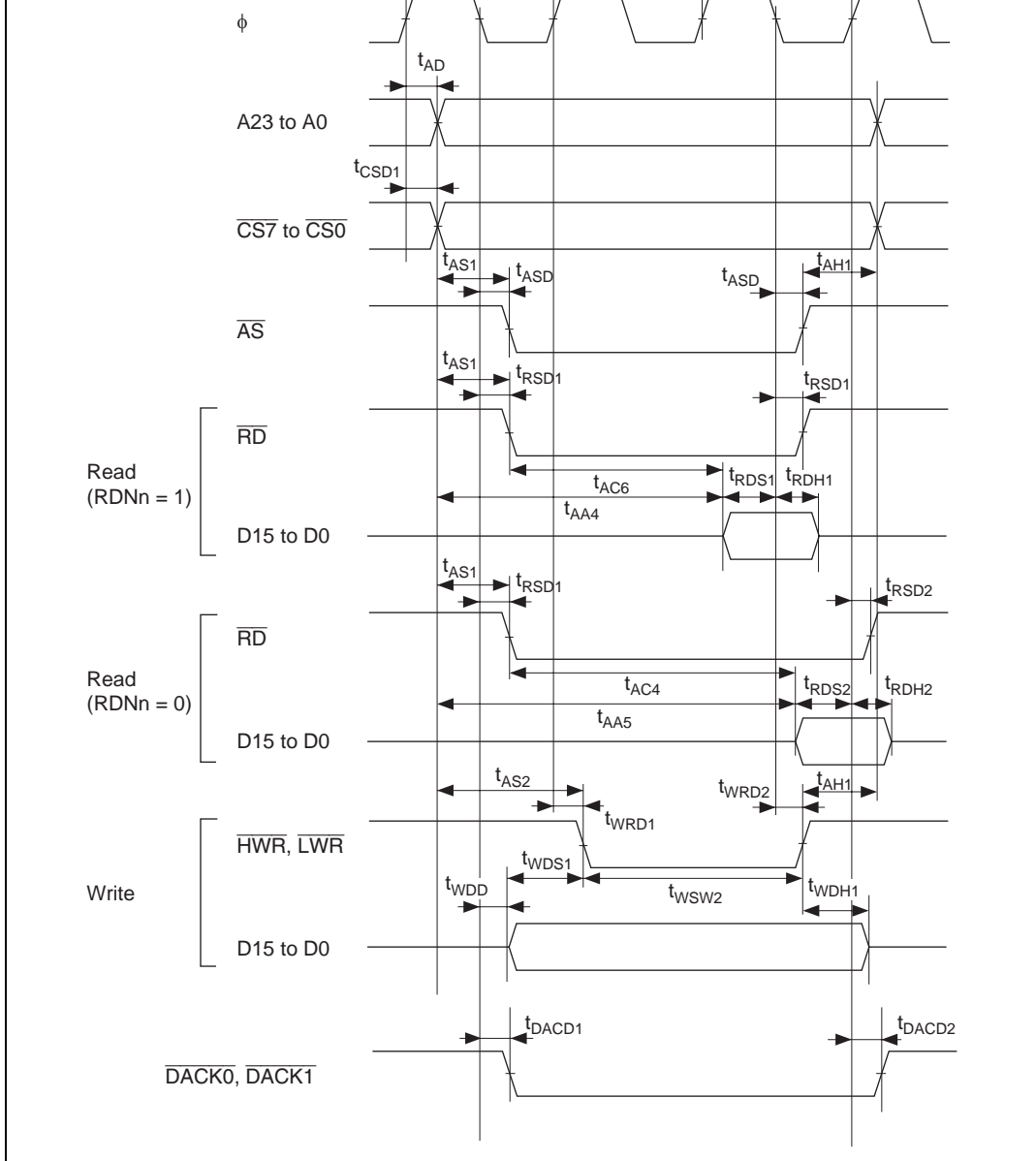
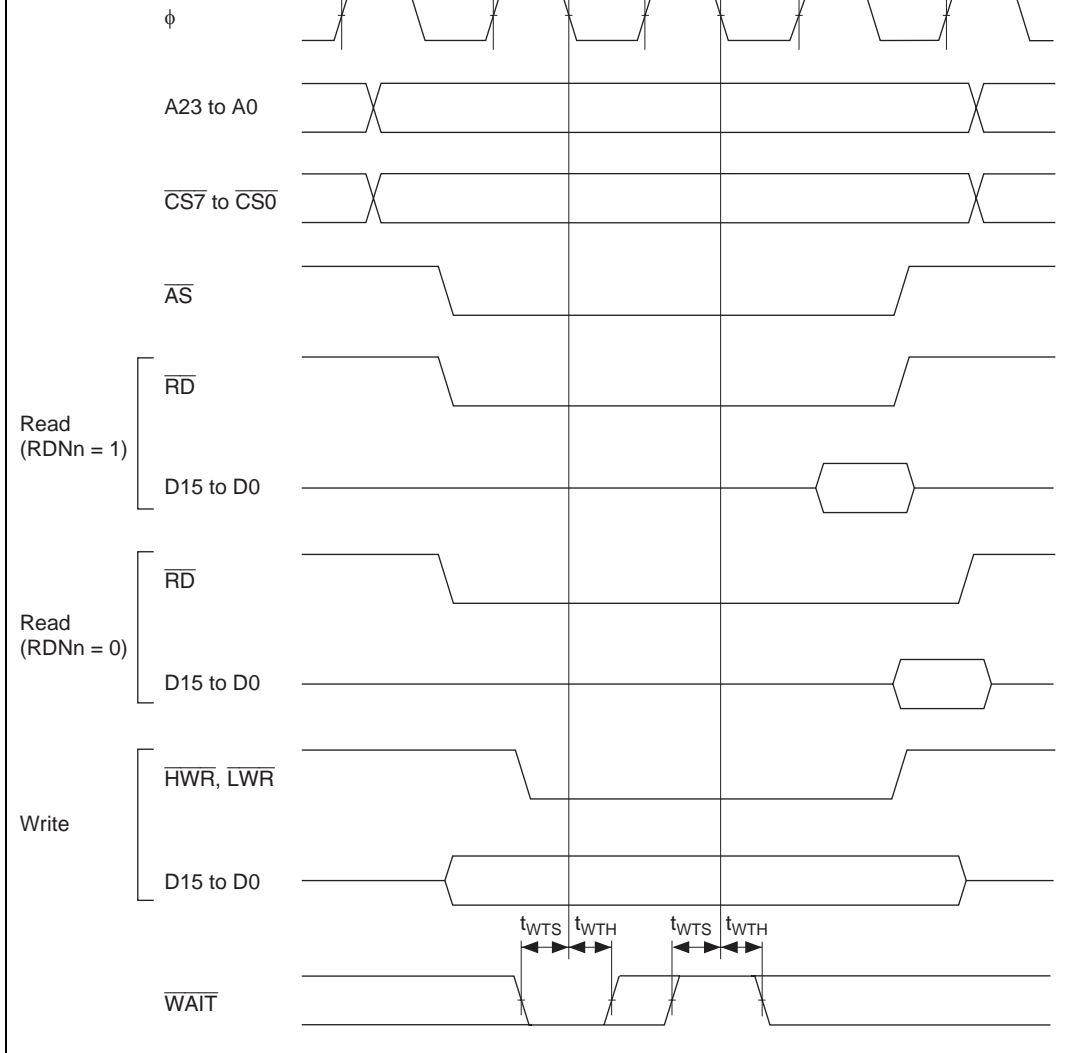
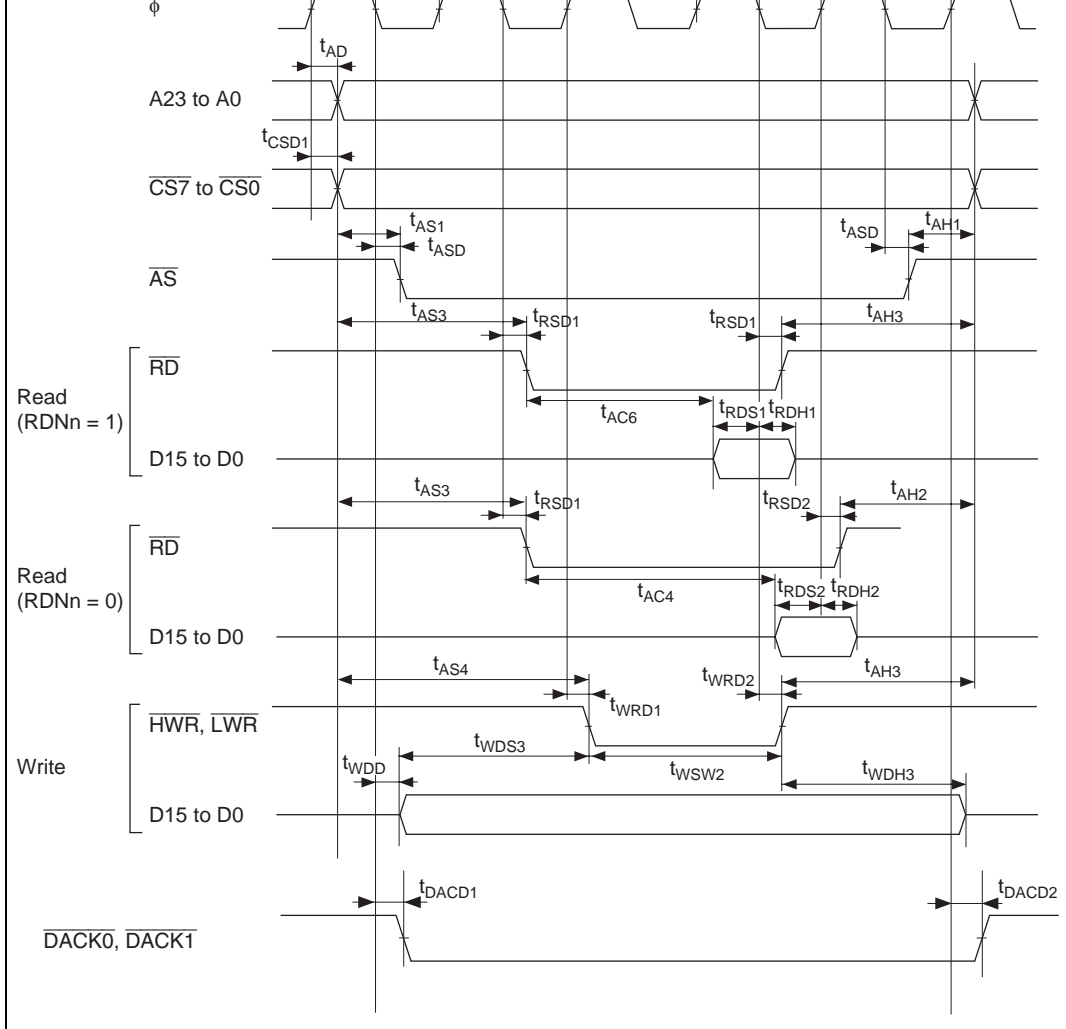


Figure 25.7 Basic Bus Timing: Three-State Access

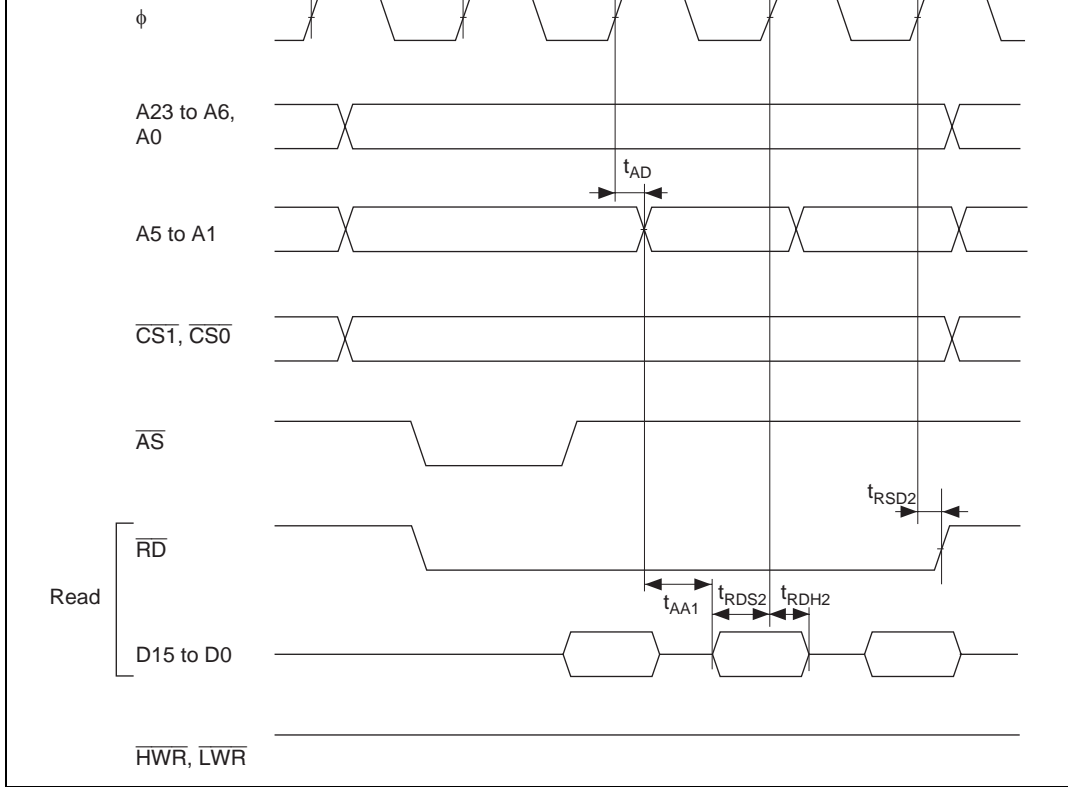


**Figure 25.8 Basic Bus Timing: Three-State Access, One Wait**

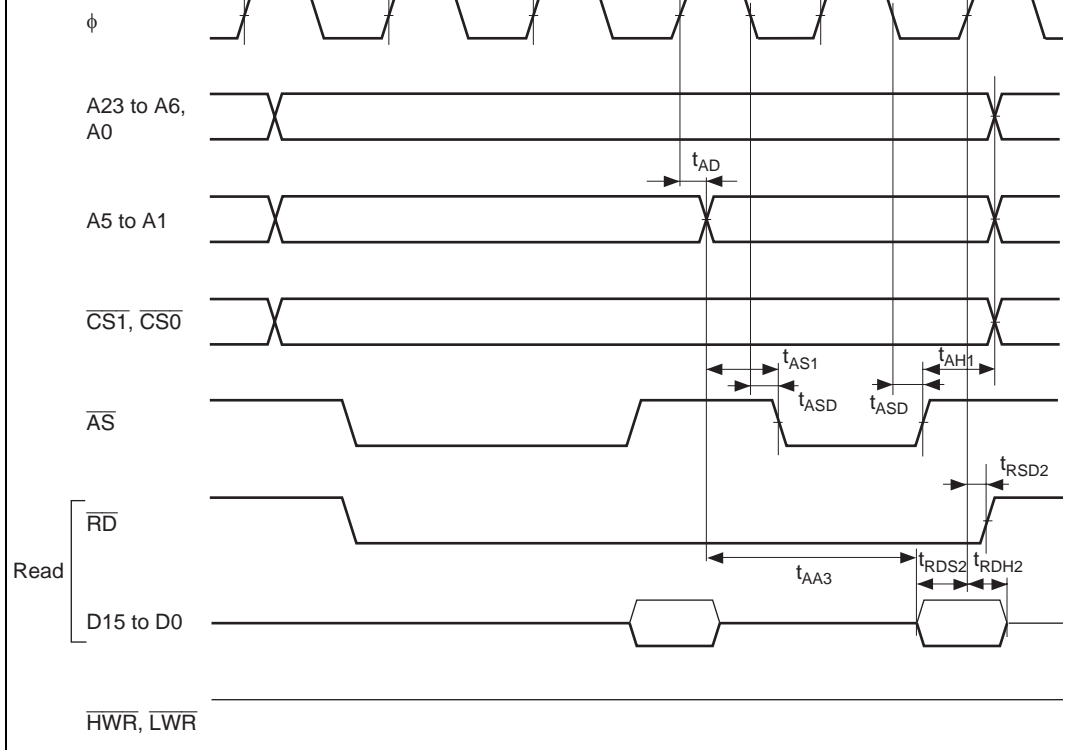




**Figure 25.10 Basic Bus Timing: Three-State Access ( $\overline{CS}$  Assertion Period Extended)**

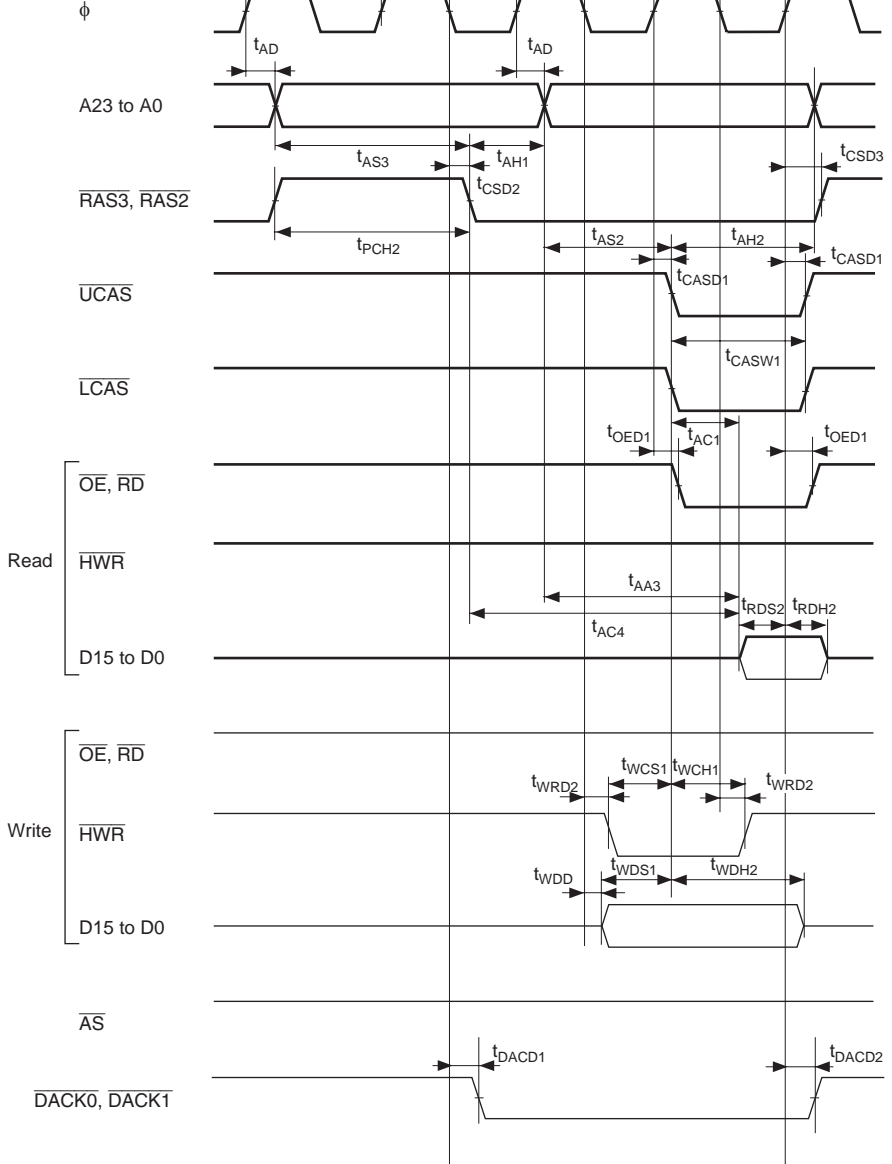


**Figure 25.11 Burst ROM Access Timing: One-State Burst Access**



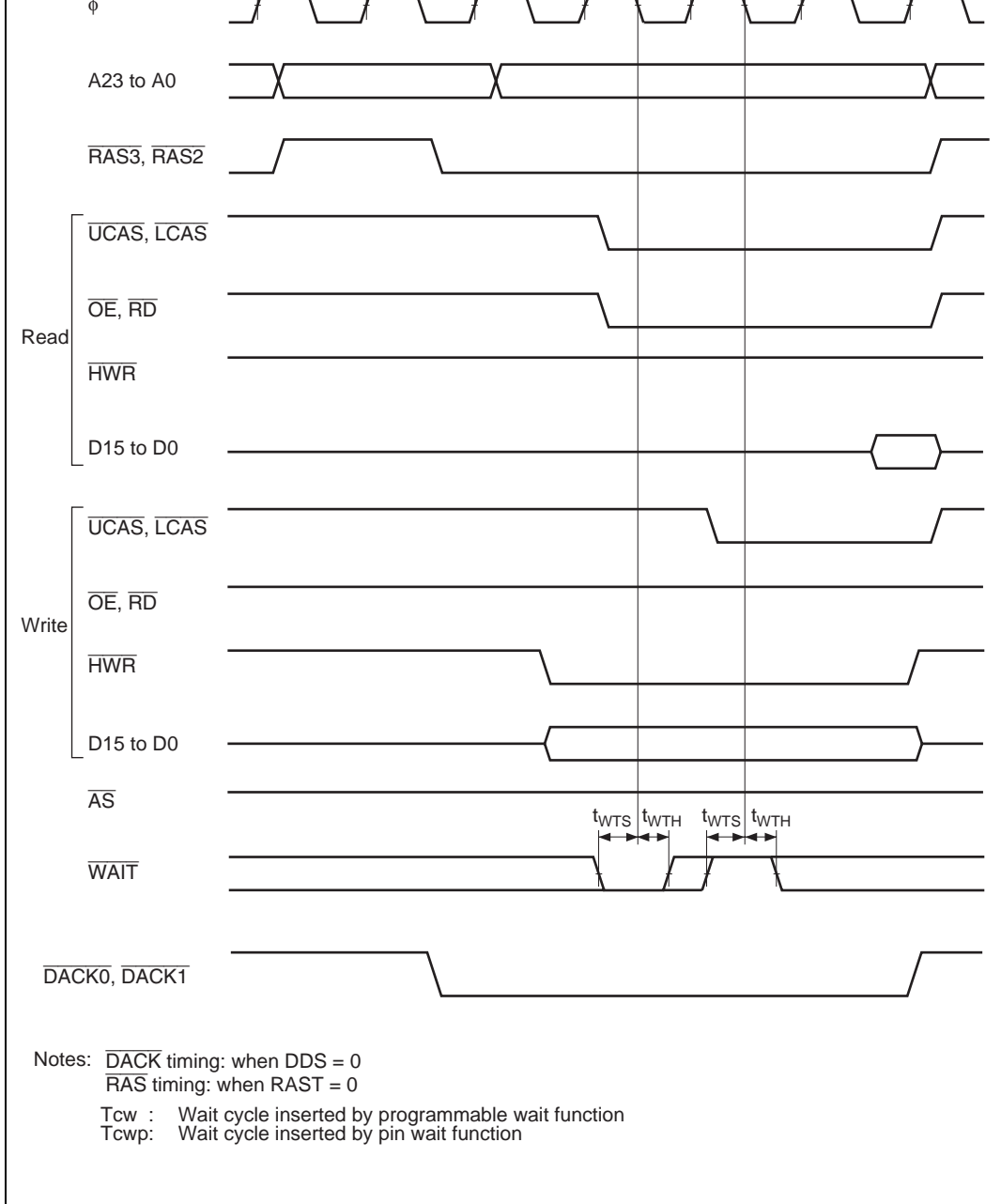
**Figure 25.12 Burst ROM Access Timing: Two-State Burst Access**





Notes:  $\overline{\text{DACK}}$  timing: when  $\text{DDS} = 0$   
 RAS timing: when  $\text{RAST} = 0$

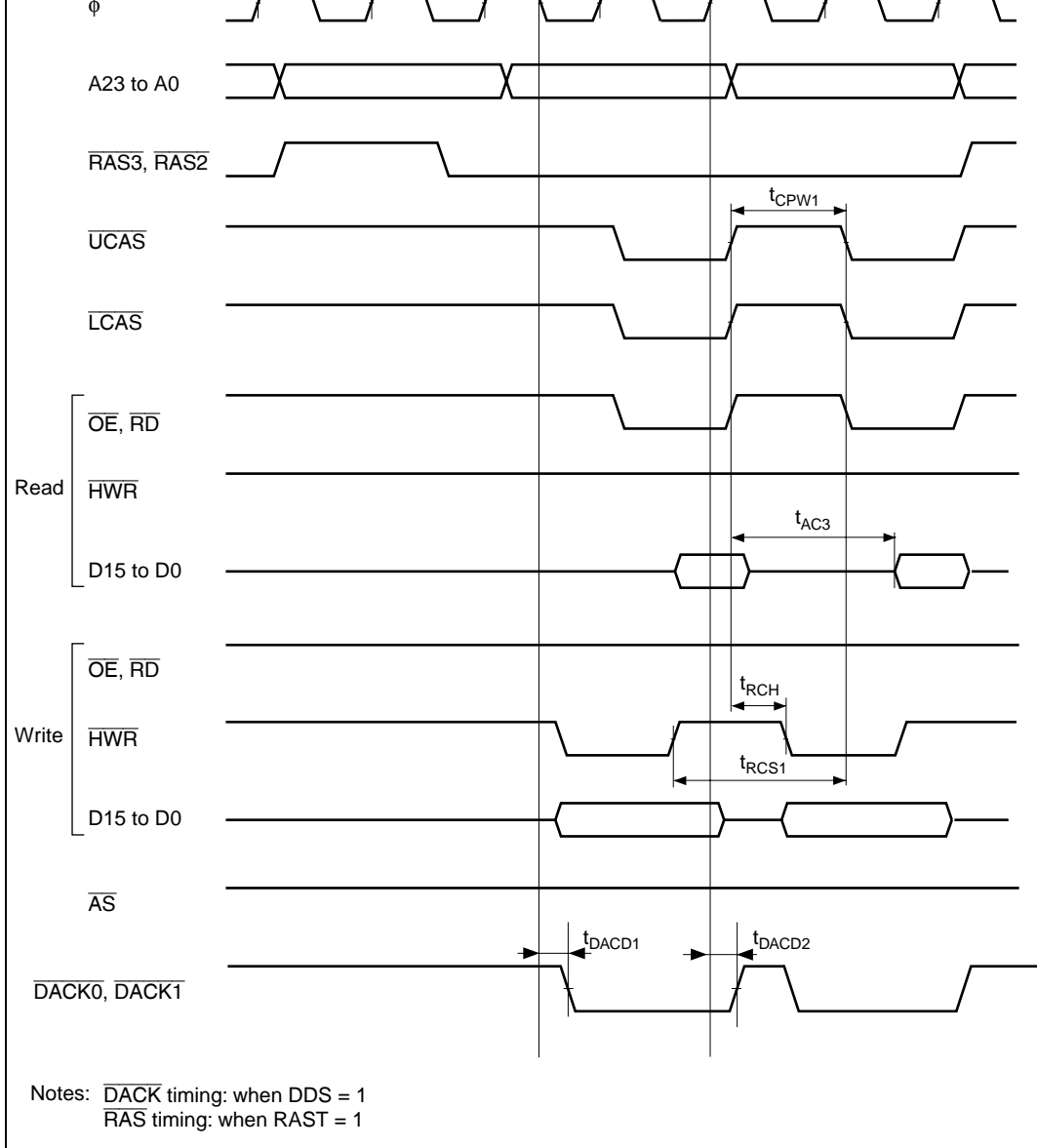
**Figure 25.13 DRAM Access Timing: Two-State Access**



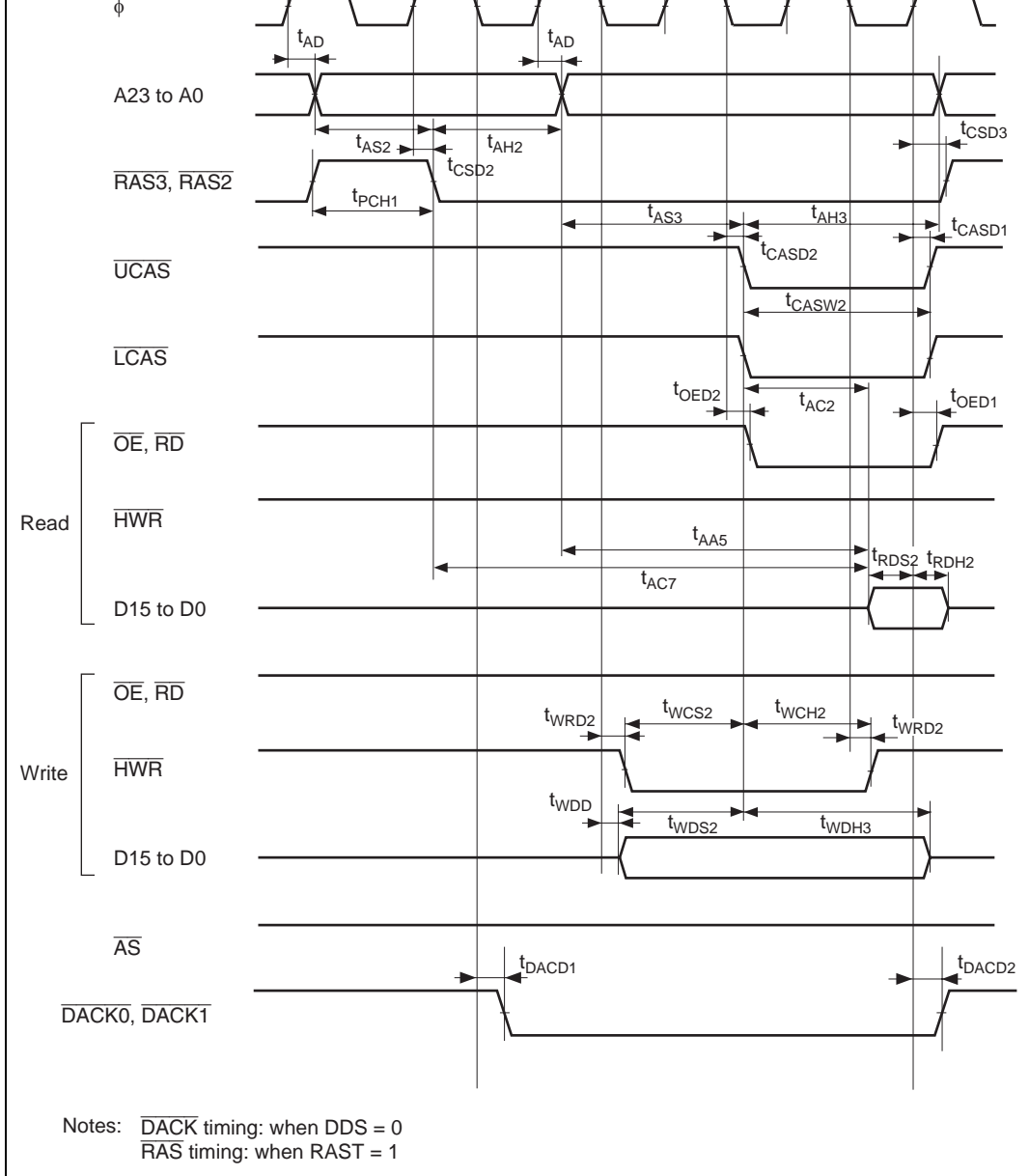
Notes:  $\overline{DACK}$  timing: when DDS = 0  
 RAS timing: when RAST = 0  
 Tcw : Wait cycle inserted by programmable wait function  
 Tcwp: Wait cycle inserted by pin wait function

**Figure 25.14 DRAM Access Timing: Two-State Access, One Wait**

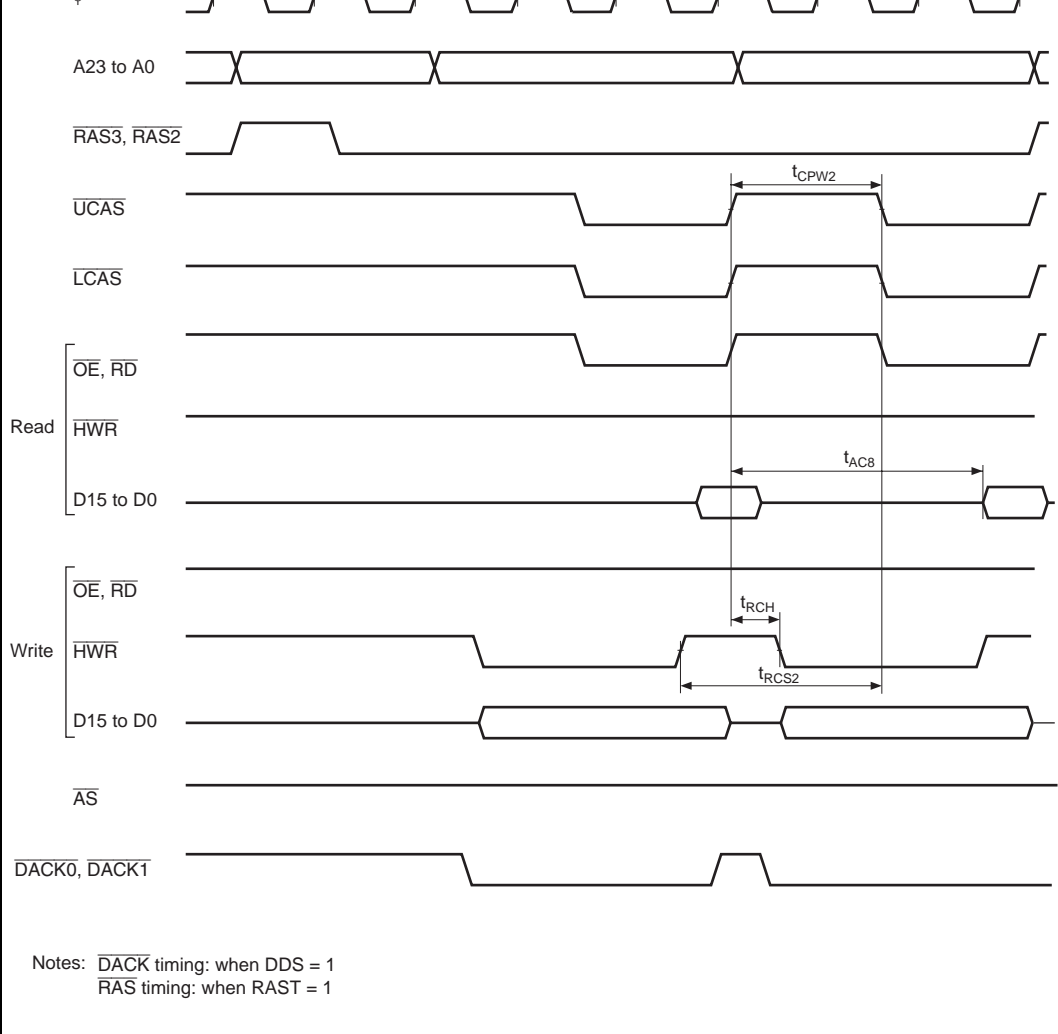




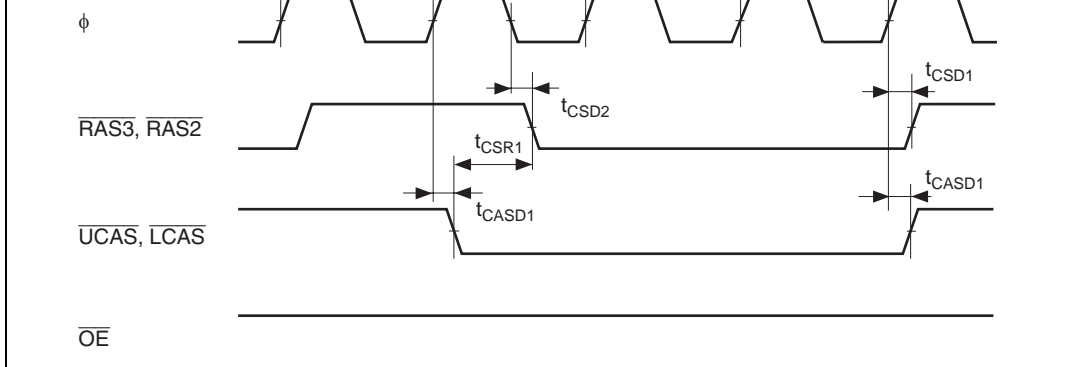
**Figure 25.15 DRAM Access Timing: Two-State Burst Access**



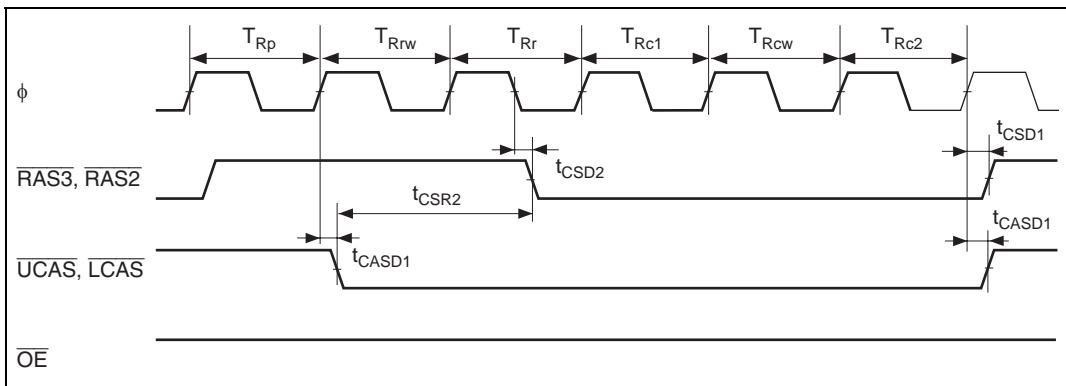
**Figure 25.16 DRAM Access Timing: Three-State Access (RAST = 1)**



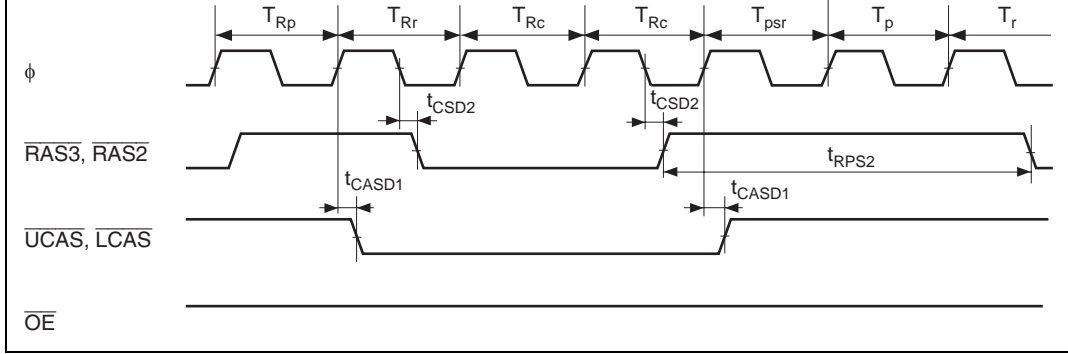
**Figure 25.17 DRAM Access Timing: Three-State Burst Access**



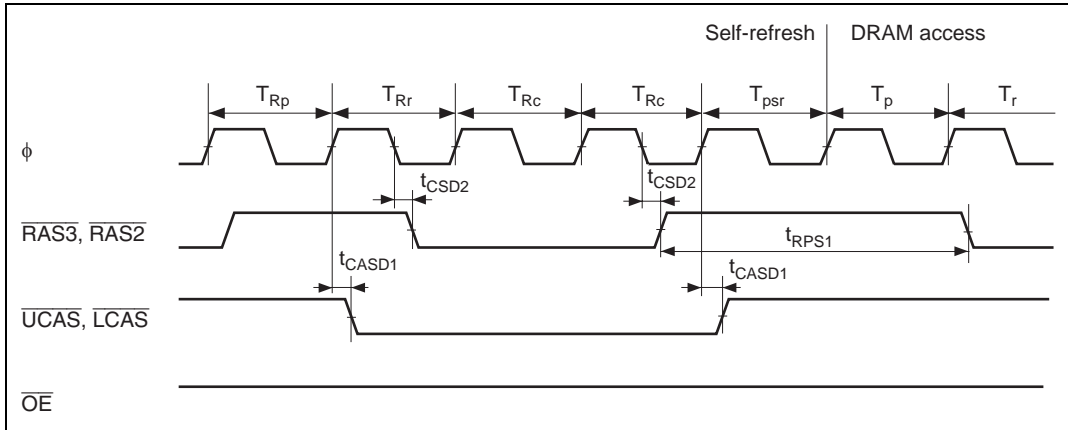
**Figure 25.18 CAS-Before-RAS Refresh Timing**



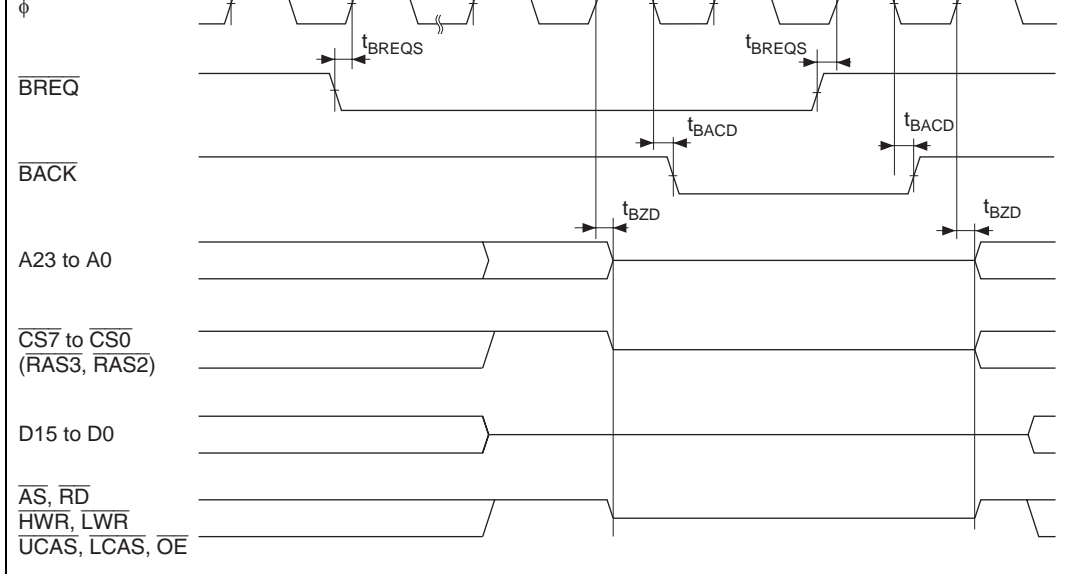
**Figure 25.19 CAS-Before-RAS Refresh Timing (with Wait Cycle Insertion)**



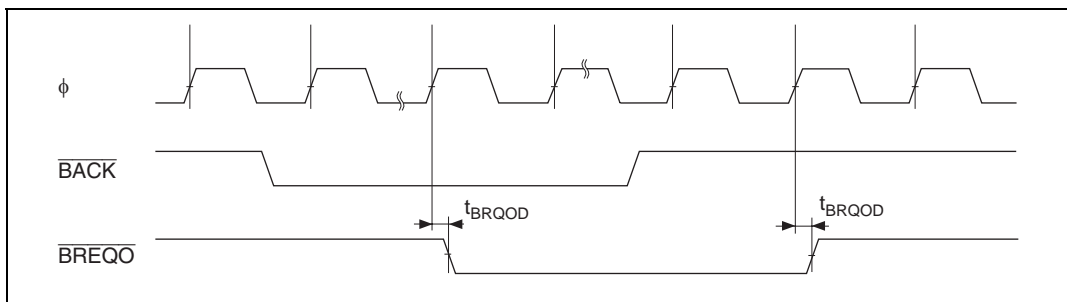
**Figure 25.20 Self-Refresh Timing (Return from Software Standby Mode: RAST = 0)**



**Figure 25.21 Self-Refresh Timing (Return from Software Standby Mode: RAST = 1)**



**Figure 25.22 External Bus Release Timing**



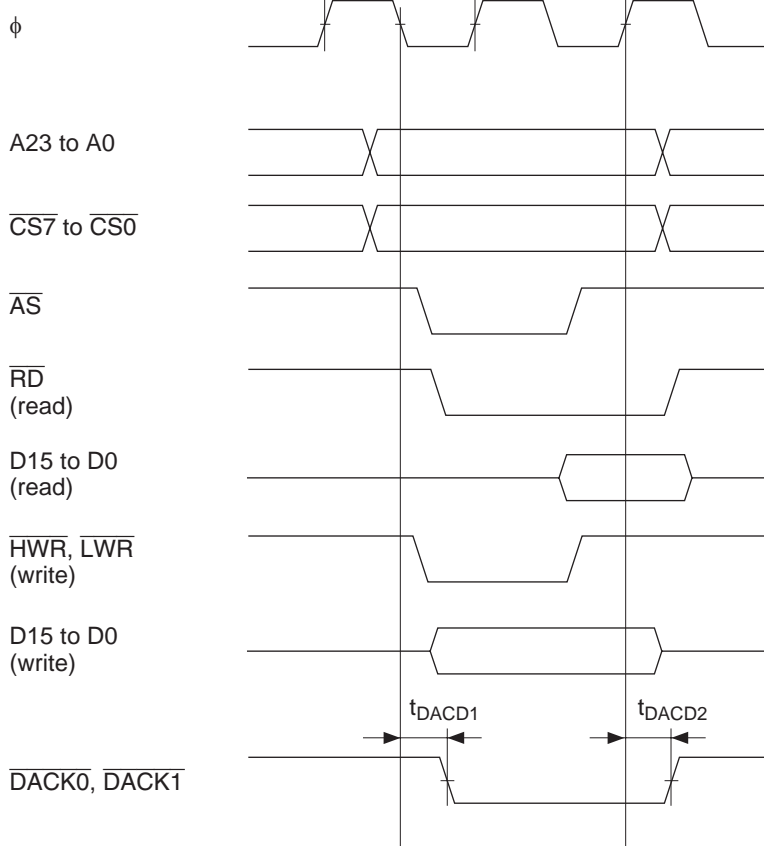
**Figure 25.23 External Bus Request Output Timing**



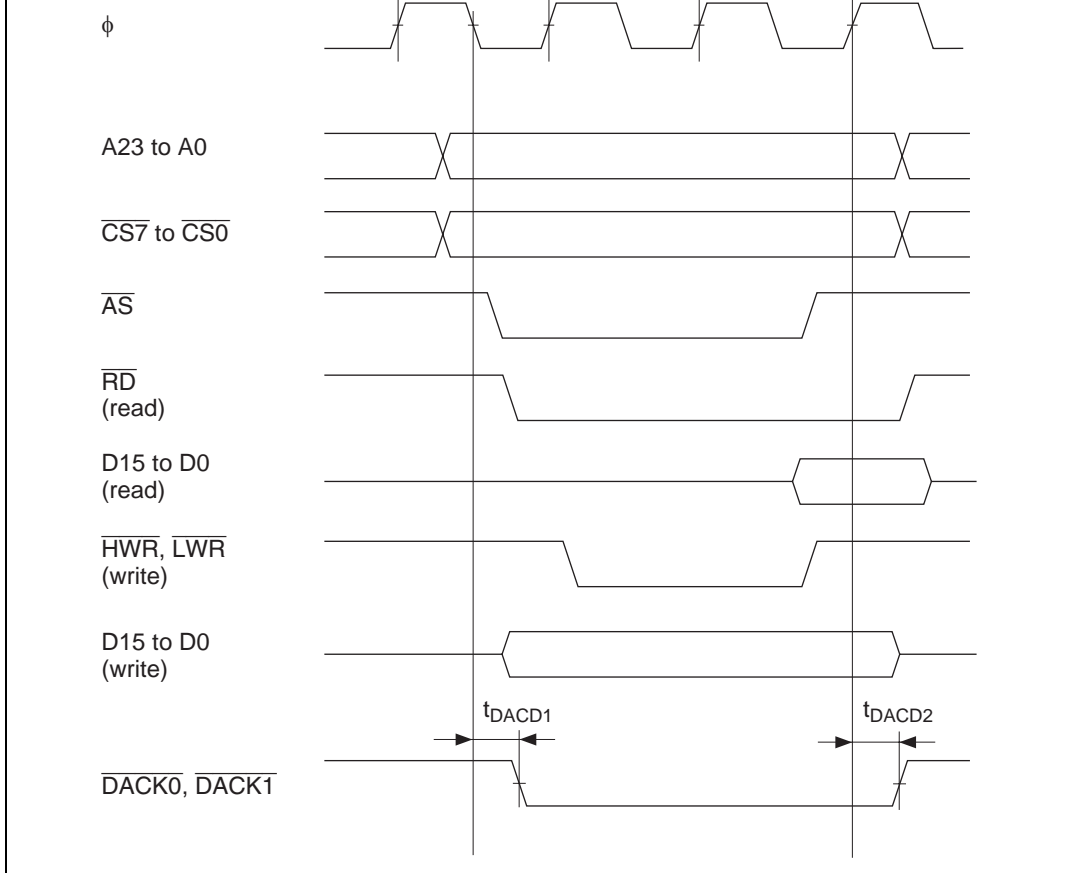
**Table 25.9 DMAC Timing**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

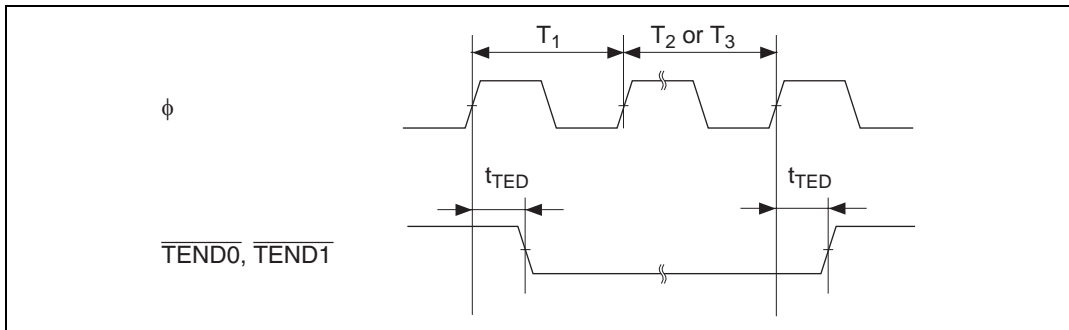
Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{\text{DREQ}}$ setup time	$t_{\text{DRQS}}$	25	—	ns	Figure 25.27
$\overline{\text{DREQ}}$ hold time	$t_{\text{DRQH}}$	10	—		
$\overline{\text{TEND}}$ delay time	$t_{\text{TED}}$	—	18	ns	Figure 25.26
$\overline{\text{DACK}}$ delay time 1	$t_{\text{DACD1}}$	—	18		Figures 25.24 and 25.25
$\overline{\text{DACK}}$ delay time 2	$t_{\text{DACD2}}$	—	18		



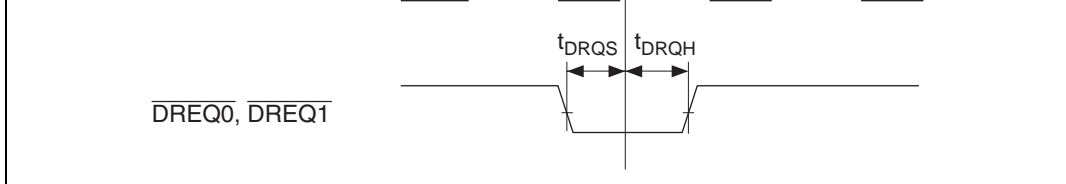
**Figure 25.24 DMAC Single Address Transfer Timing: Two-State Access**



**Figure 25.25 DMAC Single Address Transfer Timing: Three-State Access**



**Figure 25.26 DMAC  $\overline{TEND}$  Output Timing**



**Figure 25.27 DMAC  $\overline{\text{DREQ}}$  Input Timing**

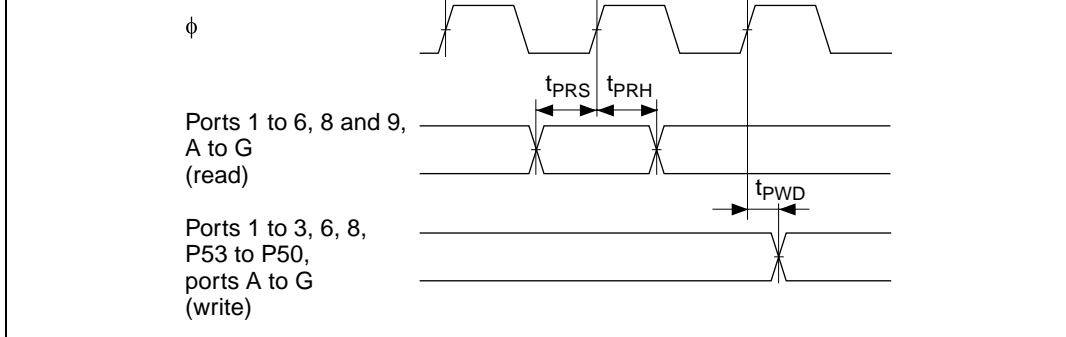
**(5) Timing of On-Chip Peripheral Modules**

**Table 25.10 Timing of On-Chip Peripheral Modules**

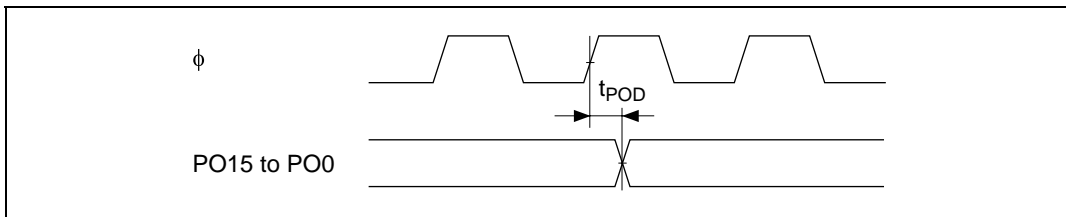
Conditions:  $V_{CC} = 3.0 \text{ V to } 3.6 \text{ V}$ ,  $AV_{CC} = 3.0 \text{ V to } 3.6 \text{ V}$ ,  $V_{ref} = 3.0 \text{ V to } AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0 \text{ V}$ ,  
 $\phi = 8 \text{ MHz to } 33 \text{ MHz}$ ,  $T_a = -20^\circ\text{C to } +75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to } +85^\circ\text{C}$  (wide-range specifications)

Item		Symbol	Min	Max	Unit	Test Conditions	
I/O ports	Output data delay time	$t_{PWD}$	—	40	ns	Figure 25.28	
	Input data setup time	$t_{PRS}$	25	—	ns		
	Input data hold time	$t_{PRH}$	25	—	ns		
PPG	Pulse output delay time	$t_{POD}$	—	40	ns	Figure 25.29	
TPU	Timer output delay time	$t_{TOCD}$	—	40	ns	Figure 25.30	
	Timer input setup time	$t_{TICS}$	25	—	ns		
	Timer clock input setup time	$t_{TCKS}$	25	—	ns	Figure 25.31	
	Timer clock pulse width	Single-edge specification	$t_{TCKWH}$	1.5	—	$t_{cyc}$	
		Both-edge specification	$t_{TCKWL}$	2.5	—	$t_{cyc}$	
8-bit timer	Timer output delay time	$t_{TMOD}$	—	40	ns	Figure 25.32	
	Timer reset input setup time	$t_{TMRS}$	25	—	ns	Figure 25.34	
	Timer clock input setup time	$t_{TMCS}$	25	—	ns	Figure 25.33	
	Timer clock pulse width	Single-edge specification	$t_{TMCWH}$	1.5	—	$t_{cyc}$	
		Both-edge specification	$t_{TMCWL}$	2.5	—	$t_{cyc}$	

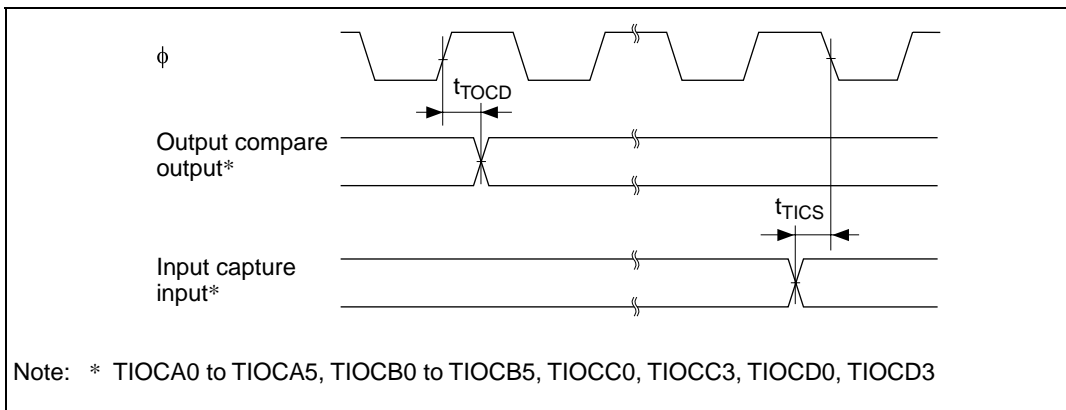
SCI	Input clock cycle	Asynchronous	$t_{S_{cyc}}$	4	—	$t_{cyc}$	Figure 25.36
		Synchronous		6	—		
	Input clock pulse width		$t_{S_{CKW}}$	0.4	0.6	$t_{S_{cyc}}$	
	Input clock rise time		$t_{S_{CKr}}$	—	1.5	$t_{cyc}$	
	Input clock fall time		$t_{S_{CKf}}$	—	1.5		
	Transmit data delay time		$t_{TXD}$	—	40	ns	Figure 25.37
	Receive data setup time (synchronous)		$t_{RXS}$	40	—	ns	
	Receive data hold time (synchronous)		$t_{RXH}$	40	—	ns	
A/D converter	Trigger input setup time		$t_{TRGS}$	30	—	ns	Figure 25.38
IIC2	SCL input cycle time		$t_{SCL}$	$12 t_{cyc} + 600$	—	ns	Figure 25.39
	SCL input high pulse width		$t_{SCLH}$	$3 t_{cyc} + 300$	—	ns	
	SCL input low pulse width		$t_{SCLL}$	$5 t_{cyc} + 300$	—	ns	
	SCL, SDA Input fall time		$t_{Sf}$	—	300	ns	
	SCL, SDA Input spike pulse removal time		$t_{SP}$	—	$1 t_{cyc}$	ns	
	SDA input bus free time		$t_{BUF}$	$5 t_{cyc}$	—	ns	
	Start condition input hold time		$t_{STAH}$	$3 t_{cyc}$	—	ns	
	Retransmit start condition input setup time		$t_{STAS}$	$3 t_{cyc}$	—	ns	
	Stop condition input setup time		$t_{STOS}$	$1 t_{cyc} + 20$	—	ns	
	Data input setup time		$t_{SDAS}$	0	—	ns	
	Data input hold time		$t_{SDAH}$	0	—	ns	
	SCL, SDA capacitive load		$C_b$	—	400	pF	
	SCL, SDA fall time		$t_{Sf}$	—	300	ns	



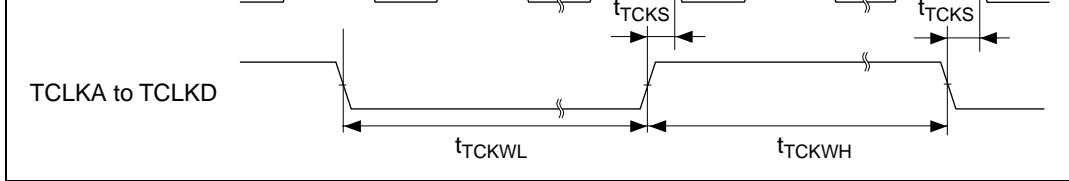
**Figure 25.28 I/O Port Input/Output Timing**



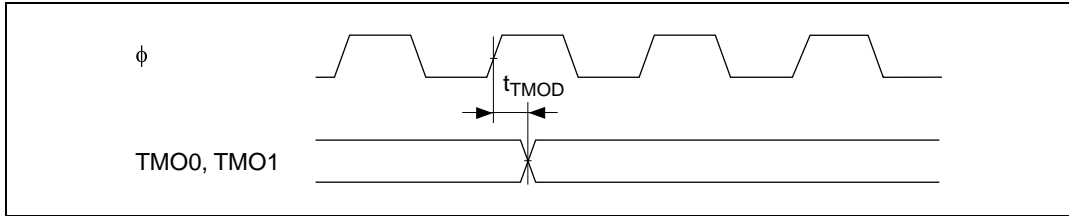
**Figure 25.29 PPG Output Timing**



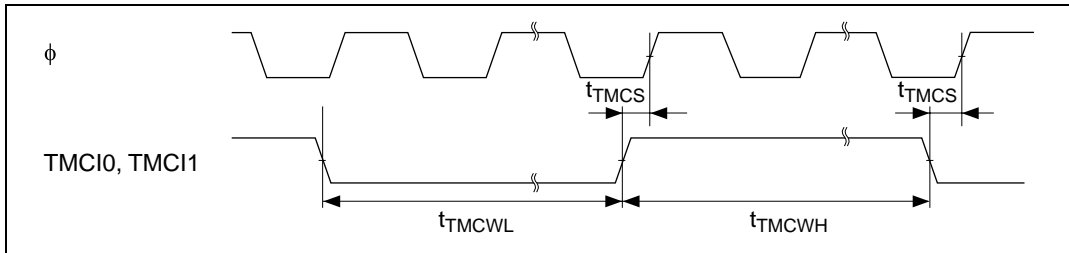
**Figure 25.30 TPU Input/Output Timing**



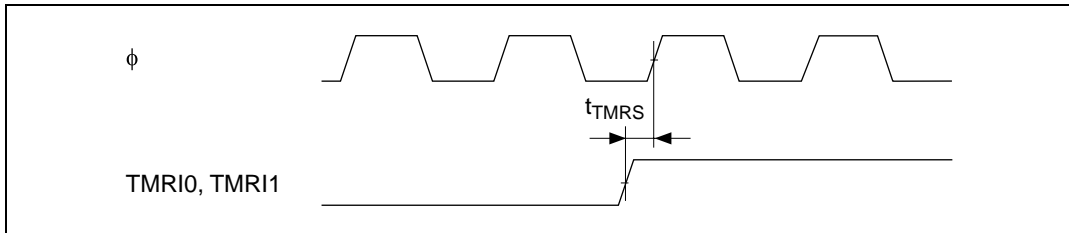
**Figure 25.31 TPU Clock Input Timing**



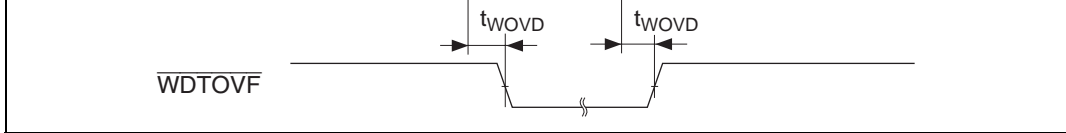
**Figure 25.32 8-Bit Timer Output Timing**



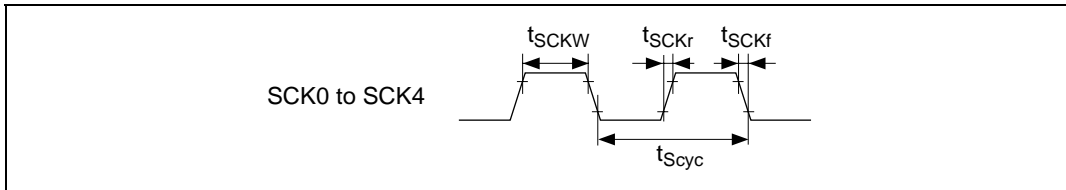
**Figure 25.33 8-Bit Timer Clock Input Timing**



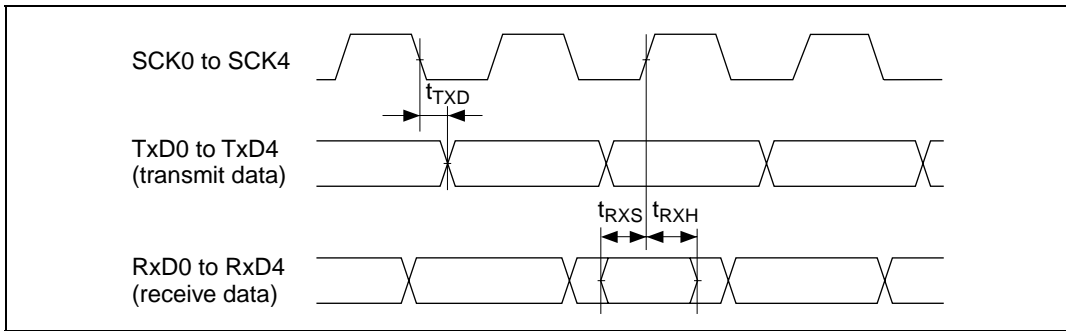
**Figure 25.34 8-Bit Timer Reset Input Timing**



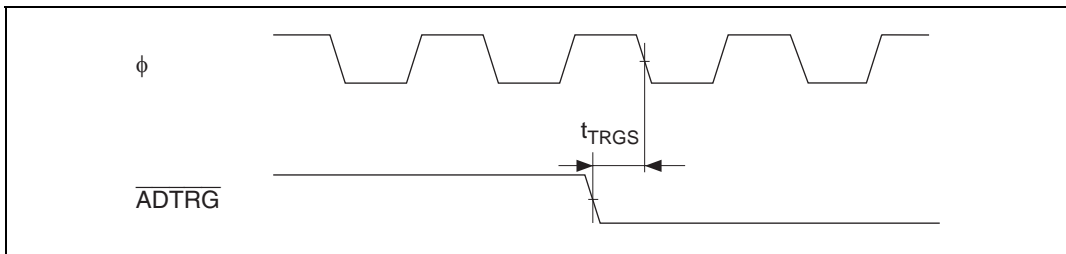
**Figure 25.35 WDT Output Timing**



**Figure 25.36 SCK Clock Input Timing**

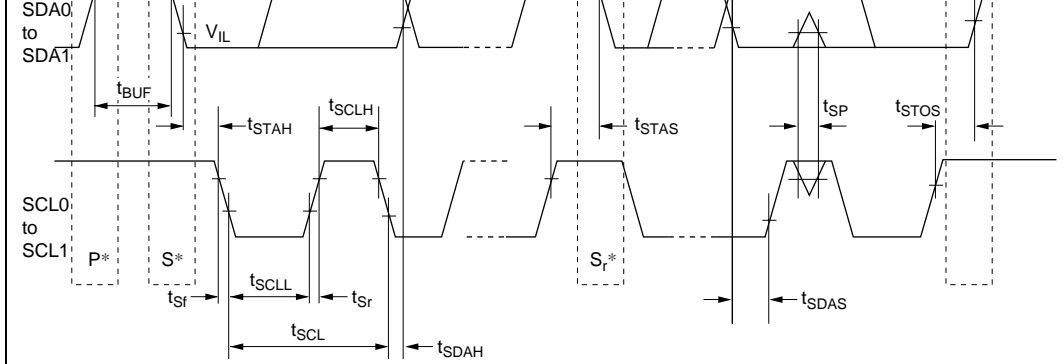


**Figure 25.37 SCI Input/Output Timing: Synchronous Mode**



**Figure 25.38 A/D Converter External Trigger Input Timing**





Note: \* S, P, and Sr represent the following conditions:  
 S: Start condition  
 P: Stop condition  
 Sr: Retransmit start condition

**Figure 25.39 I<sup>2</sup>C Bus Interface Input/Output Timing (Option)**

**Table 25.11 A/D Conversion Characteristics**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Min	Typ	Max	Unit
Resolution	10	10	10	Bit
Conversion time	8.1	—	—	$\mu\text{s}$
Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	5	$\text{k}\Omega$
Nonlinearity error	—	—	$\pm 5.5$	LSB
Offset error	—	—	$\pm 5.5$	LSB
Full-scale error	—	—	$\pm 5.5$	LSB
Quantization error	—	—	$\pm 0.5$	LSB
Absolute accuracy	—	—	$\pm 6.0$	LSB

### 25.1.5 D/A Conversion Characteristics

**Table 25.12 D/A Conversion Characteristics**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Min	Typ	Max	Unit	Test Conditions
Resolution	8	8	8	Bit	
Conversion time	—	—	10	$\mu\text{s}$	20 pF capacitive load
Absolute accuracy	—	$\pm 2.0$	$\pm 3.0$	LSB	2 M $\Omega$ resistive load
	—	—	$\pm 2.0$	LSB	4 M $\Omega$ resistive load

## 25.2.1 Absolute Maximum Ratings

Table 25.13 lists the absolute maximum ratings.

**Table 25.13 Absolute Maximum Ratings**

Item	Symbol	Value	Unit
Power supply voltage	$V_{CC}$ $PLL V_{CC}$	-0.3 to +4.0	V
Input voltage (except ports 4, 9)	$V_{in}$	-0.3 to $V_{CC} + 0.3$	V
Input voltage (ports 4, 9)	$V_{in}$	-0.3 to $AV_{CC} + 0.3$	V
Reference power supply voltage	$V_{ref}$	-0.3 to $AV_{CC} + 0.3$	V
Analog power supply voltage	$AV_{CC}$	-0.3 to +4.0	V
Analog input voltage	$V_{AN}$	-0.3 to $AV_{CC} + 0.3$	V
Operating temperature	$T_{opr}$	Regular specifications: -20 to +75*	°C
		Wide-range specifications: -40 to +85*	°C
Storage temperature	$T_{stg}$	-55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note: \* F-ZTAT version:

Ranges of operating temperature when flash memory is programmed/erased:

Regular specifications: 0 to +75°C

Wide-range specifications: 0 to +85°C

**Table 25.14 DC Characteristics (1)**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0\text{ V}^{*1}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions	
Schmitt trigger input voltage	Ports 1, 2, and 4 <sup>*2</sup> , P50 to P53 <sup>*2</sup> , PA4 to PA7 <sup>*2</sup>	$VT^-$	$V_{CC} \times 0.2$	—	—	V	
		$VT^+$	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{STBY}$ , MD2 to MD0	$V_{IH}$	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	$\overline{RES}$ , NMI, EMLE		$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Port 3, P50 to P53 <sup>*3</sup> , port 8, ports A to G <sup>*3</sup>		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Ports 4 and 9		$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$	V	
Input low voltage	$\overline{RES}$ , $\overline{STBY}$ , MD2 to MD0, EMLE	$V_{IL}$	-0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL		-0.3	—	$V_{CC} \times 0.2$	V	
	Ports 3 to 5 <sup>*3</sup> , 8, 9, A to G <sup>*3</sup>		-0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	$V_{OH}$	$V_{CC} - 0.5$	—	—	V	$I_{OH} = -200\ \mu\text{A}$
			$V_{CC} - 1.0$	—	—	V	$I_{OH} = -1\ \text{mA}$
Output low voltage	All output pins	$V_{OL}$	—	—	0.4	V	$I_{OL} = 1.6\ \text{mA}$
	P32 to P35 <sup>*4</sup>		—	—	0.5	V	$I_{OL} = 8.0\ \text{mA}$

- Notes: 1. When the A/D and D/A converters are not used, do not leave the  $AV_{CC}$ ,  $V_{ref}$ , and  $AV_{SS}$  pins open. Connect the  $AV_{CC}$  and  $V_{ref}$  pins to  $V_{CC}$  and the  $AV_{SS}$  pin to  $V_{SS}$ .
2. When used as  $\overline{IRQ0}$  to  $\overline{IRQ7}$ .
3. When used as other than  $\overline{IRQ0}$  to  $\overline{IRQ7}$ .
4. When used as SCL0 to SCL1, SDA0 to SDA1.

Conditions:  $V_{CC} = 5.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V to }5.5\text{ V}$ ,  $V_{ref} = 5.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0\text{ V}^{*1}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input leakage current	$\overline{\text{RES}}$	$ I_{in} $	—	—	10.0	$\mu\text{A}$	$V_{in} = 0.5\text{ to }V_{CC} - 0.5\text{ V}$
	$\overline{\text{STBY}}$ , NMI, MD2 to MD0		—	—	1.0	$\mu\text{A}$	
	Ports 4 and 9		—	—	1.0	$\mu\text{A}$	$V_{in} = 0.5\text{ to }AV_{CC} - 0.5\text{ V}$
Three-state leakage current (off state)	Ports 1 to 3, P50 to P53, ports 8, A to G	$ I_{TSL} $	—	—	1.0	$\mu\text{A}$	$V_{in} = 0.5\text{ to }V_{CC} - 0.5\text{ V}$
Input pull-up MOS current	Ports A to E	$-I_p$	10	—	300	$\mu\text{A}$	$V_{CC} = 3.0\text{ to }3.6\text{ V}$ $V_{in} = 0\text{ V}$
Input capacitance	$\overline{\text{RES}}$	$C_{in}$	—	—	30	pF	$V_{in} = 0\text{ V}$
	NMI		—	—	30	pF	$f = 1\text{ MHz}$
	All input pins except $\overline{\text{RES}}$ and NMI		—	—	15	pF	$T_a = 25^\circ\text{C}$
Current dissipation <sup>*2</sup>	Normal operation	$I_{CC}^{*4}$	—	75 (3.3 V)	115	$\text{mA}$	$f = 33\text{ MHz}$
	Sleep mode		—	55 (3.3 V)	95	$\text{mA}$	$f = 33\text{ MHz}$
	Standby mode <sup>*3</sup>		—	0.01	10	$\mu\text{A}$	$T_a \leq 50^\circ\text{C}$
			—	—	80	$\mu\text{A}$	$50^\circ\text{C} < T_a$
Analog power supply current	During A/D and D/A conversion	$AI_{CC}$	—	0.3 (3.0 V)	2.0	$\text{mA}$	
	Idle		—	0.01	5.0	$\mu\text{A}$	

Reference power supply current	During A/D and D/A conversion Idle	$I_{CC}$	—	2.0 (3.0 V)	3.5	mA
RAM standby voltage		$V_{RAM}$	2.0	—	—	V

- Notes: 1. When the A/D and D/A converters are not used, do not leave the  $AV_{CC}$ ,  $V_{ref}$ , and  $AV_{SS}$  pins open. Connect the  $AV_{CC}$  and  $V_{ref}$  pins to  $V_{CC}$ , and the  $AV_{SS}$  pin to  $V_{SS}$ .
2. Current dissipation values are for  $V_{IHmin} = V_{CC} - 0.2$  V and  $V_{ILmax} = 0.2$  V with all output pins unloaded and all MOS input pull-ups in the off state.
3. The values are for  $V_{RAM} \leq V_{CC} < 3.0$  V,  $V_{IHmin} = V_{CC} \times 0.9$ , and  $V_{ILmax} = 0.3$  V.
4.  $I_{CC}$  depends on  $V_{CC}$  and  $f$  as follows:  
 $I_{CCmax} = 1.0$  (mA) +  $0.95$  (mA/(MHz  $\times$  V))  $\times V_{CC} \times f$  (normal operation)  
 $I_{CCmax} = 1.0$  (mA) +  $0.8$  (mA/(MHz  $\times$  V))  $\times V_{CC} \times f$  (sleep mode)

**Table 25.16 Permissible Output Currents**

Conditions:  $V_{CC} = 3.0$  V to 3.6 V,  $AV_{CC} = 3.0$  V to 3.6 V,  $V_{ref} = 3.0$  V to  $AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0$  V\*,  $T_a = -20^\circ\text{C}$  to  $+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C}$  to  $+85^\circ\text{C}$  (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	SCL0 to SCL1, SDA0 to SDA1	$I_{OL}$	—	—	8.0	mA
	Output pins other than the above		—	—	2.0	
Permissible output low current (total)	Total of all output pins	$\Sigma I_{OL}$	—	—	80	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0	mA
Permissible output high current (total)	Total of all output pins	$\Sigma -I_{OH}$	—	—	40	mA

Caution: To protect the LSI's reliability, do not exceed the output current values in table 25.16.

Note: \* When the A/D and D/A converters are not used, do not leave the  $AV_{CC}$ ,  $V_{ref}$ , and  $AV_{SS}$  pins open. Connect the  $AV_{CC}$  and  $V_{ref}$  pins to  $V_{CC}$ , and the  $AV_{SS}$  pin to  $V_{SS}$ .

**Table 25.17 Clock Timing**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

<b>Item</b>	<b>Symbol</b>	<b>Min</b>	<b>Max</b>	<b>Unit</b>	<b>Test Conditions</b>
Clock cycle time	$t_{cyc}$	30.3	125	ns	Figure 25.2
Clock pulse high width	$t_{CH}$	10	—	ns	Figure 25.2
Clock pulse low width	$t_{CL}$	10	—	ns	
Clock rise time	$t_{Cr}$	—	5	ns	
Clock fall time	$t_{Cf}$	—	5	ns	
Reset oscillation stabilization time (crystal)	$t_{OSC1}$	10	—	ms	Figure 25.3(1)
Software standby oscillation stabilization time (crystal)	$t_{OSC2}$	10	—	ms	Figure 25.3(2)
External clock output stabilization time	$t_{DEXT}$	1	—	ms	Figure 25.3(1)

**Table 25.18 Control Signal Timing**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{\text{RES}}$ setup time	$t_{\text{RESS}}$	200	—	ns	Figure 25.4
$\overline{\text{RES}}$ pulse width	$t_{\text{RESW}}$	20	—	$t_{\text{cyc}}$	
NMI setup time	$t_{\text{NMIS}}$	150	—	ns	Figure 25.5
NMI hold time	$t_{\text{NMIH}}$	10	—		
NMI pulse width (in recovery from software standby mode)	$t_{\text{NMIW}}$	200	—		
$\overline{\text{IRQ}}$ setup time	$t_{\text{IRQS}}$	150	—	ns	
$\overline{\text{IRQ}}$ hold time	$t_{\text{IROH}}$	10	—		
$\overline{\text{IRQ}}$ pulse width (in recovery from software standby mode)	$t_{\text{IRQW}}$	200	—		



**Table 25.19 Bus Timing (1)**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
Address delay time	$t_{AD}$	—	20	ns	Figures 25.6 to 25.19
Address setup time 1	$t_{AS1}$	$0.5 \times t_{cyc} - 13$	—	ns	
Address setup time 2	$t_{AS2}$	$1.0 \times t_{cyc} - 13$	—	ns	
Address setup time 3	$t_{AS3}$	$1.5 \times t_{cyc} - 13$	—	ns	
Address setup time 4	$t_{AS4}$	$2.0 \times t_{cyc} - 13$	—	ns	
Address hold time 1	$t_{AH1}$	$0.5 \times t_{cyc} - 8$	—	ns	
Address hold time 2	$t_{AH2}$	$1.0 \times t_{cyc} - 8$	—	ns	
Address hold time 3	$t_{AH3}$	$1.5 \times t_{cyc} - 8$	—	ns	
$\overline{CS}$ delay time 1	$t_{CSD1}$	—	15	ns	
$\overline{CS}$ delay time 2	$t_{CSD2}$	—	15	ns	
$\overline{CS}$ delay time 3	$t_{CSD3}$	—	20	ns	
$\overline{AS}$ delay time	$t_{ASD}$	—	15	ns	
$\overline{RD}$ delay time 1	$t_{RSD1}$	—	15	ns	
$\overline{RD}$ delay time 2	$t_{RSD2}$	—	15	ns	
Read data setup time 1	$t_{RDS1}$	15	—	ns	
Read data setup time 2	$t_{RDS2}$	15	—	ns	
Read data hold time 1	$t_{RDH1}$	0	—	ns	
Read data hold time 2	$t_{RDH2}$	0	—	ns	
Read data access time 1	$t_{AC1}$	—	$1.0 \times t_{cyc} - 20$	ns	
Read data access time 2	$t_{AC2}$	—	$1.5 \times t_{cyc} - 20$	ns	
Read data access time 3	$t_{AC3}$	—	$2.0 \times t_{cyc} - 20$	ns	
Read data access time 4	$t_{AC4}$	—	$2.5 \times t_{cyc} - 20$	ns	
Read data access time 5	$t_{AC5}$	—	$1.0 \times t_{cyc} - 20$	ns	
Read data access time 6	$t_{AC6}$	—	$2.0 \times t_{cyc} - 20$	ns	
Read data access time 7	$t_{AC7}$	—	$4.0 \times t_{cyc} - 20$	ns	
Read data access time 8	$t_{AC8}$	—	$3.0 \times t_{cyc} - 20$	ns	

---

Address read data access time 2	$t_{AA2}$	—	$1.5 \times t_{oyc} - 20$	ns
Address read data access time 3	$t_{AA3}$	—	$2.0 \times t_{oyc} - 20$	ns
Address read data access time 4	$t_{AA4}$	—	$2.5 \times t_{oyc} - 20$	ns
Address read data access time 5	$t_{AA5}$	—	$3.0 \times t_{oyc} - 20$	ns

---

Conditions:  $V_{CC} = 5.0\text{ V to }5.0\text{ V}$ ,  $AV_{CC} = 5.0\text{ V to }5.0\text{ V}$ ,  $V_{ref} = 5.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{WR}$ delay time 1	$t_{WRD1}$	—	15	ns	Figures 25.6 to 25.19
$\overline{WR}$ delay time 2	$t_{WRD2}$	—	15	ns	
$\overline{WR}$ pulse width 1	$t_{WSW1}$	$1.0 \times t_{cyc} - 13$	—	ns	
$\overline{WR}$ pulse width 2	$t_{WSW2}$	$1.5 \times t_{cyc} - 13$	—	ns	
Write data delay time	$t_{WDD}$	—	20	ns	
Write data setup time 1	$t_{WDS1}$	$0.5 \times t_{cyc} - 13$	—	ns	
Write data setup time 2	$t_{WDS2}$	$1.0 \times t_{cyc} - 13$	—	ns	
Write data setup time 3	$t_{WDS3}$	$1.5 \times t_{cyc} - 13$	—	ns	
Write data hold time 1	$t_{WDH1}$	$0.5 \times t_{cyc} - 8$	—	ns	
Write data hold time 2	$t_{WDH2}$	$1.0 \times t_{cyc} - 8$	—	ns	
Write data hold time 3	$t_{WDH3}$	$1.5 \times t_{cyc} - 8$	—	ns	
Write command setup time 1	$t_{WCS1}$	$0.5 \times t_{cyc} - 10$	—	ns	
Write command setup time 2	$t_{WCS2}$	$1.0 \times t_{cyc} - 10$	—	ns	
Write command hold time 1	$t_{WCH1}$	$0.5 \times t_{cyc} - 10$	—	ns	
Write command hold time 2	$t_{WCH2}$	$1.0 \times t_{cyc} - 10$	—	ns	
Read command setup time 1	$t_{RCS1}$	$1.5 \times t_{cyc} - 10$	—	ns	
Read command setup time 2	$t_{RCS2}$	$2.0 \times t_{cyc} - 10$	—	ns	
Read command hold time	$t_{RCH}$	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ delay time 1	$t_{CASD1}$	—	15	ns	
$\overline{CAS}$ delay time 2	$t_{CASD2}$	—	15	ns	
$\overline{CAS}$ setup time 1	$t_{CSR1}$	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ setup time 2	$t_{CSR2}$	$1.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ pulse width 1	$t_{CASW1}$	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ pulse width 2	$t_{CASW2}$	$1.5 \times t_{cyc} - 20$	—	ns	
CAS precharge time 1	$t_{CPW1}$	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ precharge time 2	$t_{CPW2}$	$1.5 \times t_{cyc} - 20$	—	ns	
$\overline{OE}$ delay time 1	$t_{OED1}$	—	15	ns	
$\overline{OE}$ delay time 2	$t_{OED2}$	—	15	ns	

Precharge time 2	$t_{PCH2}$	$1.5 \times t_{cyc} - 20$	—	ns	25.19
Self-refresh precharge time 1	$t_{RPS1}$	$2.5 \times t_{cyc} - 20$	—	ns	Figures 25.20 and 25.21
Self-refresh precharge time 2	$t_{RPS2}$	$3.0 \times t_{cyc} - 20$	—	ns	
WAIT setup time	$t_{WTS}$	25	—	ns	Figure 25.14
WAIT hold time	$t_{WTH}$	5	—	ns	
BREQ setup time	$t_{BREQS}$	30	—	ns	Figure 25.22
BACK delay time	$t_{BACD}$	—	15	ns	
Bus floating time	$t_{BZD}$	—	40	ns	
BREQO delay time	$t_{BROOD}$	—	25	ns	Figure 25.23

Table 0. DMAC Timing

Table 25.21 DMAC Timing

Conditions:  $V_{CC} = 3.0 \text{ V to } 3.6 \text{ V}$ ,  $AV_{CC} = 3.0 \text{ V to } 3.6 \text{ V}$ ,  $V_{ref} = 3.0 \text{ V to } AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0 \text{ V}$ ,  
 $\phi = 8 \text{ MHz to } 33 \text{ MHz}$ ,  $T_a = -20^\circ\text{C to } +75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to } +85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
DREQ setup time	$t_{DRQS}$	25	—	ns	Figure 25.27
DREQ hold time	$t_{DRQH}$	10	—		
TEND delay time	$t_{TED}$	—	18	ns	Figure 25.26
DACK delay time 1	$t_{DACD1}$	—	18		Figures 25.24 and 25.25
DACK delay time 2	$t_{DACD2}$	—	18		

**Table 25.22 Timing of On-Chip Peripheral Modules**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item		Symbol	Min	Max	Unit	Test Conditions	
I/O ports	Output data delay time	$t_{PWD}$	—	40	ns	Figure 25.28	
	Input data setup time	$t_{PRS}$	25	—	ns		
	Input data hold time	$t_{PRH}$	25	—	ns		
PPG	Pulse output delay time	$t_{POD}$	—	40	ns	Figure 25.29	
TPU	Timer output delay time	$t_{TOCD}$	—	40	ns	Figure 25.30	
	Timer input setup time	$t_{TICS}$	25	—	ns		
	Timer clock input setup time	$t_{TCKS}$	25	—	ns	Figure 25.31	
	Timer clock pulse width	Single-edge specification	$t_{TCKWH}$	1.5	—		$t_{cyc}$
Both-edge specification		$t_{TCKWL}$	2.5	—	$t_{cyc}$		
8-bit timer	Timer output delay time	$t_{TMOD}$	—	40	ns	Figure 25.32	
	Timer reset input setup time	$t_{TMRS}$	25	—	ns	Figure 25.34	
	Timer clock input setup time	$t_{TMCS}$	25	—	ns	Figure 25.33	
	Timer clock pulse width	Single-edge specification	$t_{TMCWH}$	1.5	—		$t_{cyc}$
		Both-edge specification	$t_{TMCWL}$	2.5	—		$t_{cyc}$
WDT	Overflow output delay time	$t_{WOVD}$	—	40	ns	Figure 25.35	
SCI	Input clock cycle	Asynchronous	$t_{Scyc}$	4	—	$t_{cyc}$	Figure 25.36
		Synchronous		6	—		
	Input 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			
	Transmit data delay time	$t_{TXD}$	—	40	ns	Figure 25.37	
	Receive data setup time (synchronous)	$t_{RXS}$	40	—	ns		
	Receive data hold time (synchronous)	$t_{RXH}$	40	—	ns		
A/D converter	Trigger input setup time	$t_{TRGS}$	30	—	ns	Figure 25.38	

SCL input high pulse width	$t_{SCLH}$	$3 t_{cyc} + 300$	—	ns
SCL input low pulse width	$t_{SCLL}$	$5 t_{cyc} + 300$	—	ns
SCL, SDA Input fall time	$t_{Sf}$	—	300	ns
SCL, SDA Input spike pulse removal time	$t_{SP}$	—	$1 t_{cyc}$	ns
SDA input bus free time	$t_{BUF}$	$5 t_{cyc}$	—	ns
Start condition input hold time	$t_{STAH}$	$3 t_{cyc}$	—	ns
Retransmit start condition input setup time	$t_{STAS}$	$3 t_{cyc}$	—	ns
Stop condition input setup time	$t_{STOS}$	$1 t_{cyc} + 20$	—	ns
Data input setup time	$t_{SDAS}$	0	—	ns
Data input hold time	$t_{SDAH}$	0	—	ns
SCL, SDA capacitive load	Cb	—	400	pF
SCL, SDA fall time	$t_{Sf}$	—	300	ns

**Table 25.23 A/D Conversion Characteristics**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Min	Typ	Max	Unit
Resolution	10	10	10	Bit
Conversion time	8.1	—	—	$\mu\text{s}$
Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	5	$\text{k}\Omega$
Nonlinearity error	—	—	$\pm 5.5$	LSB
Offset error	—	—	$\pm 5.5$	LSB
Full-scale error	—	—	$\pm 5.5$	LSB
Quantization error	—	—	$\pm 0.5$	LSB
Absolute accuracy	—	—	$\pm 6.0$	LSB

**25.2.4 D/A Conversion Characteristics****Table 25.24 D/A Conversion Characteristics**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }33\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Min	Typ	Max	Unit	Test Conditions
Resolution	8	8	8	Bit	
Conversion time	—	—	10	$\mu\text{s}$	20 pF capacitive load
Absolute accuracy	—	$\pm 2.0$	$\pm 3.0$	LSB	2 $\text{M}\Omega$ resistive load
	—	—	$\pm 2.0$	LSB	4 $\text{M}\Omega$ resistive load

**Table 25.25 Flash Memory Characteristics**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = 0^\circ\text{C to }75^\circ\text{C}$  (program/erase operating temperature range:  
regular specifications),  $T_a = 0^\circ\text{C to }85^\circ\text{C}$  (program/erase operating temperature  
range: wide-range specifications)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions		
Programming time <sup>*1*2*4</sup>	$t_P$	—	10	200	ms/ 128 bytes			
Erase time <sup>*1*3*6</sup>	$t_E$	—	50	1000	ms/ block			
Rewrite times	$N_{WEC}$	—	—	100	Times			
Programming	Wait time after SWE bit setting <sup>*1</sup>	x	1	—	—	$\mu\text{s}$		
	Wait time after PSU bit setting <sup>*1</sup>	y	50	—	—	$\mu\text{s}$		
	Wait time after P bit setting <sup>*1*4</sup>	z	z1	—	—	30	$\mu\text{s}$	$1 \leq n \leq 6$
			z2	—	—	200	$\mu\text{s}$	$7 \leq n \leq 1000$
			z3	—	—	10	$\mu\text{s}$	Additional programming wait
	Wait time after P bit clearing <sup>*1</sup>	$\alpha$	5	—	—	$\mu\text{s}$		
	Wait time after PSU bit clearing <sup>*1</sup>	$\beta$	5	—	—	$\mu\text{s}$		
	Wait time after PV bit setting <sup>*1</sup>	$\gamma$	4	—	—	$\mu\text{s}$		
	Wait time after H'FF dummy write <sup>*1</sup>	$\varepsilon$	2	—	—	$\mu\text{s}$		
	Wait time after PV bit clearing <sup>*1</sup>	$\eta$	2	—	—	$\mu\text{s}$		
	Wait time after SWE bit clearing <sup>*1</sup>	$\theta$	100	—	—	$\mu\text{s}$		
	Maximum number of writes <sup>*1*4</sup>	N	—	—	1000 <sup>*5</sup>	Times		



Erasing	Wait time after SWE bit setting <sup>*1</sup>	x	1	—	—	μs	
	Wait time after ESU bit setting <sup>*1</sup>	y	100	—	—	μs	
	Wait time after E bit setting <sup>*1*6</sup>	z	—	—	10	ms	Erase time wait
	Wait time after E bit clearing <sup>*1</sup>	α	10	—	—	μs	
	Wait time after ESU bit clearing <sup>*1</sup>	β	10	—	—	μs	
	Wait time after EV bit setting <sup>*1</sup>	γ	20	—	—	μs	
	Wait time after H'FF dummy write <sup>*1</sup>	ε	2	—	—	μs	
	Wait time after EV bit clearing <sup>*1</sup>	η	4	—	—	μs	
	Wait time after SWE bit clearing <sup>*1</sup>	θ	100	—	—	μs	
	Maximum number of erases <sup>*1*6</sup>	N	—	—	100	Times	

- Notes:
1. Follow the program/erase algorithms when making the time settings.
  2. Programming time per 128 bytes. (Indicates the total time during which the P bit is set in flash memory control register 1 (FLMCR1). Does not include the program-verify time.)
  3. Time to erase one block. (Indicates the time during which the E bit is set in FLMCR1. Does not include the erase-verify time.)
  4. Maximum programming time

$$t_p(\max) = \sum_{i=1}^N \text{wait time after P bit setting (z)}$$

5. The maximum number of writes (N) should be set as shown below according to the actual set value of (z) so as not to exceed the maximum programming time ( $t_p(\max)$ ). The wait time after P bit setting (z) should be changed as follows according to the number of writes (n).

Number of writes (n)

$$1 \leq n \leq 6 \quad z = 30 \mu\text{s}$$

$$7 \leq n \leq 1000 \quad z = 200 \mu\text{s}$$

(Additional programming)

Number of writes (n)

$$1 \leq n \leq 6 \quad z = 10 \mu\text{s}$$

6. For the maximum erase time ( $t_E(\max)$ ), the following relationship applies between the wait time after E bit setting (z) and the maximum number of erases (N):

$$t_E(\max) = \text{Wait time after E bit setting (z)} \times \text{maximum number of erases (N)}$$

### 25.3.1 Absolute Maximum Ratings

Table 25.26 lists the absolute maximum ratings.

**Table 25.26 Absolute Maximum Ratings**

Item	Symbol	Value	Unit
Power supply voltage	$V_{CC}$ $PLL V_{CC}$	-0.3 to +4.3	V
Input voltage (except ports 4 and 9)	$V_{in}$	-0.3 to $V_{CC} + 0.3$	V
Input voltage (ports 4 and 9)	$V_{in}$	-0.3 to $AV_{CC} + 0.3$	V
Reference power supply voltage	$V_{ref}$	-0.3 to $AV_{CC} + 0.3$	V
Analog power supply voltage	$AV_{CC}$	-0.3 to +4.0	V
Analog input voltage	$V_{AN}$	-0.3 to $AV_{CC} + 0.3$	V
Operating temperature	$T_{opr}$	Regular specifications: -20 to +75*	°C
		Wide-range specifications: -40 to +85*	°C
Storage temperature	$T_{slg}$	-55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note: \* Ranges of operating temperature when flash memory is programmed/erased:  
Regular specifications: 0 to +75°C  
Wide-range specifications: 0 to +85°C

**Table 25.27 DC Characteristics**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0\text{ V}^{*1}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	Port 1, 2, 4 <sup>*2</sup> , P50 to P53 <sup>*2</sup> , PA4 to PA7 <sup>*2</sup>	$VT^-$	$V_{CC} \times 0.2$	—	—	V	
		$VT^+$	—	—	$V_{CC} \times 0.7$	V	
		$VT^+ - VT^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{STBY}$ , MD2 to MD0	$V_{IH}$	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	$\overline{RES}$ , NMI, FWE		$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Port 3, P50 to P53 <sup>*3</sup> , port 8 <sup>*3</sup> , ports A to G <sup>*3</sup>		2.2	—	$V_{CC} + 0.3$	V	
	Port 4, Port 9		2.2	—	$AV_{CC} + 0.3$	V	
Input low voltage	$\overline{RES}$ , $\overline{STBY}$ , MD2 to MD0, EMLE	$V_{IL}$	-0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL		-0.3	—	$V_{CC} \times 0.2$	V	
	Ports 3 to 5 <sup>*3</sup> , port 8, port 9, ports A to G <sup>*3</sup>		-0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	$V_{OH}$	$V_{CC} - 0.5$	—	—	V	$I_{OH} = -200\ \mu\text{A}$
			$V_{CC} - 1.0$	—	—	V	$I_{OH} = -1\ \text{mA}$
Output low voltage	All output pins P32 to P35 <sup>*4</sup>	$V_{OL}$	—	—	0.4	V	$I_{OL} = 1.6\ \text{mA}$
			—	—	0.5	V	$I_{OL} = 8.0\ \text{mA}$

- Notes: 1. When the A/D and D/A converters are not used, the  $AV_{CC}$ ,  $V_{ref}$ , and  $AV_{SS}$  pins should not be open. Connect the  $AV_{CC}$  and  $V_{ref}$  pins to  $V_{CC}$  and the  $AV_{SS}$  pin to  $V_{SS}$ .
2. When used as  $\overline{IRQ0}$  to  $\overline{IRQ7}$ .
3. When used as other than  $\overline{IRQ0}$  to  $\overline{IRQ7}$ .
4. When used as SCL0, SCL1, SDA0, and SDA1.

Conditions.  $V_{CC} = 3.0\text{ V to }5.0\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }5.0\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0\text{ V}^{*1}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	$\overline{\text{RES}}$	$ I_{in} $	—	—	10.0	$\mu\text{A}$ $V_{in} = 0.5\text{ to }V_{CC} - 0.5\text{ V}$
	$\overline{\text{STBY}}$ , NMI, MD2 to MD0		—	—	1.0	$\mu\text{A}$
	Port 4, Port 9		—	—	1.0	$\mu\text{A}$ $V_{in} = 0.5\text{ to }AV_{CC} - 0.5\text{ V}$
Three-state leakage current (off state)	Ports 1 to 3, P50 to P53, port 8, ports A to G	$ I_{TSL} $	—	—	1.0	$\mu\text{A}$ $V_{in} = 0.5\text{ to }V_{CC} - 0.5\text{ V}$
Input pull-up MOS current	Ports A to E	$-I_p$	10	—	300	$\mu\text{A}$ $V_{CC} = 3.0\text{ to }3.6\text{ V}$ $V_{in} = 0\text{ V}$
Input capacitance	$\overline{\text{RES}}$	$C_{in}$	—	—	30	$\text{pF}$ $V_{in} = 0\text{ V}$
	NMI		—	—	30	$\text{pF}$ $f = 1\text{ MHz}$
	All input pins except RES and NMI		—	—	15	$\text{pF}$ $T_a = 25^\circ\text{C}$
Current consumption <sup>*2</sup>	Normal operation	$I_{CC}^{*4}$	—	40 (3.3 V)	60	$\text{mA}$ $f = 34\text{ MHz}$
	Sleep mode		—	20 (3.3 V)	40	$\text{mA}$ $f = 34\text{ MHz}$
	Standby mode <sup>*3</sup>		—	5	20	$\mu\text{A}$ $T_a \leq 50^\circ\text{C}$
			—	—	80	$\mu\text{A}$ $50^\circ\text{C} < T_a$
Analog power supply current	During A/D and D/A conversion	$AI_{CC}$	—	0.5 (3.0 V)	2.0	$\text{mA}$
	Idle		—	0.01	5.0	$\mu\text{A}$
Reference power supply current	During A/D and D/A conversion	$AI_{CC}$	—	3.0 (3.0 V)	6.0	$\text{mA}$
	Idle		—	0.01	5.0	$\mu\text{A}$
RAM standby voltage		$V_{RAM}$	2.5	—	—	$\text{V}$
$V_{CC}$ start voltage <sup>*5</sup>		$V_{CCstart}$	—	—	—	$\text{V}$



The clock, control signal, bus, DMAC, and on-chip peripheral function timings are shown below. The measurement conditions of the AC characteristics are shown in figure 25.1.

## (1) Clock Timing

**Table 25.30 Clock Timing**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }34\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Clock cycle time	$t_{cyc}$	29.4	125	ns	Figure 25.2
Clock pulse high width	$t_{CH}$	9	—	ns	Figure 25.2
Clock pulse low width	$t_{CL}$	9	—	ns	
Clock rising time	$t_{Cr}$	—	5	ns	
Clock falling time	$t_{Cf}$	—	5	ns	
Reset oscillation settling time (crystal)	$t_{OSC1}$	10	—	ms	Figure 25.3 (1)
Software standby oscillation settling time (crystal)	$t_{OSC2}$	10	—	ms	Figure 25.3 (2)
External clock output settling time	$t_{DEXT}$	1	—	ms	Figure 25.3 (1)

**Table 25.31 Control Signal Timing**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }34\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{\text{RES}}$ setup time	$t_{\text{RESS}}$	200	—	ns	Figure 25.4
$\overline{\text{RES}}$ pulse width	$t_{\text{RESW}}$	20	—	$t_{\text{cyc}}$	
NMI setup time	$t_{\text{NMIS}}$	150	—	ns	Figure 25.5
NMI hold time	$t_{\text{NMIH}}$	10	—		
NMI pulse width (in recovery from software standby mode)	$t_{\text{NMIW}}$	200	—		
$\overline{\text{IRQ}}$ setup time	$t_{\text{IRQS}}$	150	—	ns	
$\overline{\text{IRQ}}$ hold time	$t_{\text{IRQH}}$	10	—		
$\overline{\text{IRQ}}$ pulse width (in recovery from software standby mode)	$t_{\text{IRQW}}$	200	—		

**Table 25.32 Bus Timing (1)**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }34\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	$T_{AD}$	—	20	ns	Figures 25.6 to 25.19
Address setup time 1	$T_{AS1}$	$0.5 \times t_{cyc} - 13$	—	ns	
Address setup time 2	$T_{AS2}$	$1.0 \times t_{cyc} - 13$	—	ns	
Address setup time 3	$T_{AS3}$	$1.5 \times t_{cyc} - 13$	—	ns	
Address setup time 4	$T_{AS4}$	$2.0 \times t_{cyc} - 13$	—	ns	
Address hold time 1	$T_{AH1}$	$0.5 \times t_{cyc} - 8$	—	ns	
Address hold time 2	$T_{AH2}$	$1.0 \times t_{cyc} - 8$	—	ns	
Address hold time 3	$T_{AH3}$	$1.5 \times t_{cyc} - 8$	—	ns	
$\overline{CS}$ delay time 1	$t_{CSD1}$	—	15	ns	
$\overline{CS}$ delay time 2	$t_{CSD2}$	—	15	ns	
$\overline{CS}$ delay time 3	$t_{CSD3}$	—	20	ns	
$\overline{AS}$ delay time	$T_{ASD}$	—	15	ns	
$\overline{RD}$ delay time 1	$t_{RSD1}$	—	15	ns	
$\overline{RD}$ delay time 2	$t_{RSD2}$	—	15	ns	
Read data setup time 1	$t_{RDS1}$	15	—	ns	
Read data setup time 2	$t_{RDS2}$	15	—	ns	
Read data hold time 1	$t_{RDH1}$	0	—	ns	
Read data hold time 2	$t_{RDH2}$	0	—	ns	
Read data access time 1	$T_{AC1}$	—	$1.0 \times t_{cyc} - 25$	ns	
Read data access time 2	$T_{AC2}$	—	$1.5 \times t_{cyc} - 25$	ns	
Read data access time 3	$T_{AC3}$	—	$2.0 \times t_{cyc} - 25$	ns	
Read data access time 4	$T_{AC4}$	—	$2.5 \times t_{cyc} - 25$	ns	
Read data access time 5	$T_{AC5}$	—	$1.0 \times t_{cyc} - 25$	ns	
Read data access time 6	$T_{AC6}$	—	$2.0 \times t_{cyc} - 25$	ns	
Read data access time 7	$T_{AC7}$	—	$4.0 \times t_{cyc} - 25$	ns	
Read data access time 8	$T_{AC8}$	—	$3.0 \times t_{cyc} - 25$	ns	



---

Address read data access time 2	$T_{AA2}$	—	$1.5 \times t_{cyc}$	-25	ns
Address read data access time 3	$T_{AA3}$	—	$2.0 \times t_{cyc}$	-25	ns
Address read data access time 4	$T_{AA4}$	—	$2.5 \times t_{cyc}$	-25	ns
Address read data access time 5	$T_{AA5}$	—	$3.0 \times t_{cyc}$	-25	ns

---

Conditions:  $V_{CC} = 5.0\text{ V to }5.5\text{ V}$ ,  $AV_{CC} = 5.0\text{ V to }5.5\text{ V}$ ,  $V_{ref} = 5.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }34\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{WR}$ delay time 1	$t_{WRD1}$	—	15	ns	Figures 25.6 to 25.19
$\overline{WR}$ delay time 2	$t_{WRD2}$	—	15	ns	
$\overline{WR}$ pulse width 1	$t_{WSW1}$	$1.0 \times t_{cyc} - 13$	—	ns	
$\overline{WR}$ pulse width 2	$t_{WSW2}$	$1.5 \times t_{cyc} - 13$	—	ns	
Write data delay time	$t_{WDD}$	—	23	ns	
Write data setup time 1	$t_{WDS1}$	$0.5 \times t_{cyc} - 15$	—	ns	
Write data setup time 2	$t_{WDS2}$	$1.0 \times t_{cyc} - 15$	—	ns	
Write data setup time 3	$t_{WDS3}$	$1.5 \times t_{cyc} - 15$	—	ns	
Write data hold time 1	$t_{WDH1}$	$0.5 \times t_{cyc} - 13$	—	ns	
Write data hold time 2	$t_{WDH2}$	$1.0 \times t_{cyc} - 13$	—	ns	
Write data hold time 3	$t_{WDH3}$	$1.5 \times t_{cyc} - 13$	—	ns	
Write command setup time 1	$t_{WCS1}$	$0.5 \times t_{cyc} - 10$	—	ns	
Write command setup time 2	$t_{WCS2}$	$1.0 \times t_{cyc} - 10$	—	ns	
Write command hold time 1	$t_{WCH1}$	$0.5 \times t_{cyc} - 10$	—	ns	
Write command hold time 2	$t_{WCH2}$	$1.0 \times t_{cyc} - 10$	—	ns	
Read command setup time 1	$t_{RCS1}$	$1.5 \times t_{cyc} - 10$	—	ns	
Read command setup time 2	$t_{RCS2}$	$2.0 \times t_{cyc} - 10$	—	ns	
Read command hold time	$t_{RCH}$	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ delay time 1	$t_{CASD1}$	—	15	ns	
$\overline{CAS}$ delay time 2	$t_{CASD2}$	—	15	ns	
$\overline{CAS}$ setup time 1	$t_{CSR1}$	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ setup time 2	$t_{CSR2}$	$1.5 \times t_{cyc} - 10$	—	ns	
$\overline{CAS}$ pulse width 1	$t_{CASW1}$	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ pulse width 2	$t_{CASW2}$	$1.5 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ precharge time 1	$t_{CPW1}$	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{CAS}$ precharge time 2	$t_{CPW2}$	$1.5 \times t_{cyc} - 20$	—	ns	
$\overline{OE}$ delay time 1	$t_{OED1}$	—	15	ns	
$\overline{OE}$ delay time 2	$t_{OED2}$	—	15	ns	

Precharge time 2	$t_{PCH2}$	$1.5 \times t_{cyc} - 20$	—	ns	25.19
Self-refresh precharge time 1	$t_{RPS1}$	$2.5 \times t_{cyc} - 20$	—	ns	Figures 25.20 and 25.21
Self-refresh precharge time 2	$t_{RPS2}$	$3.0 \times t_{cyc} - 20$	—	ns	
$\overline{WAIT}$ setup time	$t_{WTS}$	25	—	ns	Figures 25.14
$\overline{WAIT}$ hold time	$t_{WTH}$	5	—	ns	
$\overline{BREQ}$ setup time	$t_{BREQS}$	30	—	ns	Figure 25.22
$\overline{BACK}$ delay time	$t_{BACD}$	—	15	ns	
Bus floating time	$t_{BZD}$	—	40	ns	
$\overline{BREQO}$ delay time	$t_{BRQOD}$	—	25	ns	Figure 25.23

#### (4) DMAC Timing

**Table 25.34 DMAC Timing**

Conditions:  $V_{CC} = 3.0 \text{ V to } 3.6 \text{ V}$ ,  $AV_{CC} = 3.0 \text{ V to } 3.6 \text{ V}$ ,  $V_{ref} = 3.0 \text{ V to } AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0 \text{ V}$ ,  
 $\phi = 8 \text{ MHz to } 34 \text{ MHz}$ ,  $T_a = -20^\circ\text{C to } +75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to } +85^\circ\text{C}$  (wide-range specifications)

Item	Symbol	Min.	Max.	Unit	Test Conditions
$\overline{DREQ}$ setup time	$t_{DRQS}$	25	—	ns	Figure 25.27
$\overline{DREQ}$ hold time	$t_{DRQH}$	10	—		
$\overline{TEND}$ delay time	$t_{TED}$	—	18	ns	Figure 25.26
$\overline{DACK}$ delay time 1	$t_{DACD1}$	—	18		Figures 25.24 and 25.25
$\overline{DACK}$ delay time 2	$t_{DACD2}$	—	18		

**Table 25.35 Timing of On-Chip Peripheral Modules**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }34\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item		Symbol	Min.	Max.	Unit	Test Conditions	
I/O ports	Output data delay time	$t_{PWD}$	—	40	ns	Figure 25.28	
	Input data setup time	$t_{PRS}$	25	—	ns		
	Input data hold time	$t_{PRH}$	25	—	ns		
PPG	Pulse output delay time	$t_{POD}$	—	40	ns	Figure 25.29	
TPU	Timer output delay time	$t_{TOCD}$	—	40	ns	Figure 25.30	
	Timer input setup time	$t_{TICS}$	25	—	ns		
	Timer clock input setup time	$t_{TCKS}$	25	—	ns	Figure 25.31	
	Timer clock pulse width	Single-edge specification	$t_{TCKWH}$	1.5	—		$t_{cyc}$
Both-edge specification		$t_{TCKWL}$	2.5	—	$t_{cyc}$		
8-bit timer	Timer output delay time	$t_{TMOD}$	—	40	ns	Figure 25.32	
	Timer reset input setup time	$t_{TMRS}$	25	—	ns	Figure 25.34	
	Timer clock input setup time	$t_{TMCS}$	25	—	ns	Figure 25.33	
	Timer clock pulse width	Single-edge specification	$t_{TMCWH}$	1.5	—		$t_{cyc}$
		Both-edge specification	$t_{TMCWL}$	2.5	—		$t_{cyc}$
WDT	Overflow output delay time	$t_{WOVD}$	—	40	ns	Figure 25.35	
SCI	Input clock cycle	Asynchronous	$t_{Scyc}$	4	—	$t_{cyc}$	Figure 25.36
		Synchronous		6	—		
	Input clock pulse width	$t_{SCKW}$	0.4	0.6	$t_{Scyc}$		
	Input clock rising time	$t_{SCKr}$	—	1.5	$t_{cyc}$		
	Input clock falling time	$t_{SCKf}$	—	1.5			
	Transmit data delay time	$t_{TXD}$	—	40	ns	Figure 25.37	
	Receive data setup time (synchronous)	$t_{RXS}$	40	—	ns		
	Receive data hold time (synchronous)	$t_{RXH}$	40	—	ns		

IIC2	SCL input cycle time	$t_{SCL}$	$12 t_{cyc} + 600$	—	ns	Figure 25.39
	SCL input high pulse width	$t_{SCLH}$	$3 t_{cyc} + 300$	—	ns	
	SCL input low pulse width	$t_{SCLL}$	$5 t_{cyc} + 300$	—	ns	
	SCL, SDA Input falling time	$t_{Sf}$	—	300	ns	
	SCL, SDA Input spike pulse removal time	$t_{SP}$	—	$1 t_{cyc}$	ns	
	SDA input bus free time	$t_{BUF}$	$5 t_{cyc}$	—	ns	
	Start condition input hold time	$t_{STAH}$	$3 t_{cyc}$	—	ns	
	Retransmit start condition input setup time	$t_{STAS}$	$3 t_{cyc}$	—	ns	
	Stop condition input setup time	$t_{STOS}$	$1 t_{cyc} + 20$	—	ns	
	Data input setup time	$t_{SDAS}$	0	—	ns	
	Data input hold time	$t_{SDAH}$	0	—	ns	
	SCL, SDA capacitive load	$C_b$	—	400	pF	
	SCL, SDA falling time	$t_{Sf}$	—	300	ns	

**Table 25.36 A/D Conversion Characteristics**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }34\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Min.	Typ.	Max.	Unit
Resolution	10	10	10	Bit
Conversion time	7.6	—	—	$\mu\text{s}$
Analog input capacitance	—	—	20	pF
Permissible signal source impedance	—	—	5	$\text{k}\Omega$
Nonlinearity error	—	—	$\pm 5.5$	LSB
Offset error	—	—	$\pm 5.5$	LSB
Full-scale error	—	—	$\pm 5.5$	LSB
Quantization error	—	—	$\pm 0.5$	LSB
Absolute accuracy	—	—	$\pm 6.0$	LSB

### 25.3.5 D/A Conversion Characteristics

**Table 25.37 D/A Conversion Characteristics**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  $V_{SS} = AV_{SS} = 0\text{ V}$ ,  
 $\phi = 8\text{ MHz to }34\text{ MHz}$ ,  $T_a = -20^\circ\text{C to }+75^\circ\text{C}$  (regular specifications),  
 $T_a = -40^\circ\text{C to }+85^\circ\text{C}$  (wide-range specifications)

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	8	8	Bit	
Conversion time	—	—	10	$\mu\text{s}$	20 pF capacitive load
Absolute accuracy	—	$\pm 2.0$	$\pm 3.0$	LSB	2 M $\Omega$ resistive load
	—	—	$\pm 2.0$	LSB	4 M $\Omega$ resistive load

**Table 25.38 Flash Memory Characteristics (0.18- $\mu$ m F-ZTAT Version)**

Conditions:  $V_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $AV_{CC} = 3.0\text{ V to }3.6\text{ V}$ ,  $V_{ref} = 3.0\text{ V to }AV_{CC}$ ,  
 $V_{SS} = AV_{SS} = 0\text{ V}$ ,  $T_a = 0^\circ\text{C to }75^\circ\text{C}$  (Programming/Erasing Operating Temperature  
 Range: Normal Specifications),  $T_a = 0^\circ\text{C to }85^\circ\text{C}$  (Programming/Erasing Operating  
 Temperature Range: Extended Temperature Range Specifications)

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Programming time <sup>*1*2*4</sup>	$t_p$	—	1	10	ms/ 128 bytes	
Erase time <sup>*1*2*4</sup>	$t_E$	—	250	1500	ms/ 4 kbytes	
		—	500	4000	ms/ 32 kbytes	
		—	750	6500	ms/ 64 kbytes	
Programming time (total) <sup>*1*2*4</sup>	$\Sigma t_p$	—	2	6	s/256 kbytes	$T_a = 25^\circ\text{C}$
		—	3	9	s/384 kbytes	
		—	4	12	s/512 kbytes	
Erase time (total) <sup>*1*2*4</sup>	$\Sigma t_E$	—	7	20	s/All blocks	$T_a = 25^\circ\text{C}$
Programming and erase time (total) <sup>*1*2*4</sup>	$\Sigma t_{PE}$	—	9	26	s/256 kbytes	$T_a = 25^\circ\text{C}$
		—	10	29	s/384 kbytes	
		—	11	32	s/512 kbytes	
Rewrite times	$N_{WEC}$	100 <sup>*3</sup>	—	—	Times	
Data storage time <sup>*4</sup>	$t_{DRP}$	10	—	—	Year	

- Notes: 1. Actual programming and erase times are dependent on data characteristics.  
 2. Programming and erase times do not include data transfer time.  
 3. The minimum number of times for which all characteristics are guaranteed. (The guaranteed range is from 1 to the minimum number of times.)  
 4. Rewrite characteristics are for the operating range including the minimum value.

The F-ZTAT and masked ROM versions both satisfy the electrical characteristics shown in this manual, but actual electrical characteristic values, operating margins, noise margins, and other properties may vary due to differences in manufacturing process, on-chip ROM, layout patterns, and so on.

When system evaluation testing is carried out using the F-ZTAT version, the same evaluation testing should also be conducted for the masked ROM version when changing over to that version.



## A. I/O Port States in Each Pin State

Port Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port 1	1, 2, 4, 7	T	T	Keep	Keep	I/O port
Port 2	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P34 to P30	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P35/( $\overline{OE}$ )	1, 2, 4, 7	T	T	[OPE = 0, $\overline{OE}$ (CKE) output] T [OPE = 1, $\overline{OE}$ output] H [Other than the above] Keep	[ $\overline{OE}$ (CKE) output] T [Other than the above] Keep	[ $\overline{OE}$ output] $\overline{OE}$ (CKE) [Other than the above] I/O port
Port 4	1, 2, 4, 7	T	T	T	T	Input port
P53 to P50	1, 2, 4, 7	T	T	Keep	Keep	I/O port
Port 8	1, 2, 4, 7	T	T	Keep	Keep	I/O port
P95/DA3	1, 2, 4, 7	T	T	[DAOE3 = 1] Keep [DAOE3 = 0] T	Keep	Input port
P94/DA2	1, 2, 4, 7	T	T	[DAOE2 = 1] Keep [DAOE2 = 0] T	Keep	Input port

Port Name	Operating Mode	Reset	Standby Mode	Software Standby Mode	Bus Release State	State Sleep Mode
PA7/A23/ $\overline{CS7}$	1, 2, 4, 7	T	T	[OPE = 0, $\overline{CS}$ output] T [OPE = 1, $\overline{CS}$ output] H [OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	$\overline{CS}$ output] $\overline{CS}$ [Address output] A23 [Other than the above] I/O port
PA6/A22 PA5/A21	1, 2, 4, 7	T	T	[OPE = 0, address output] T [OPE = 1, address output] [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A22 to A21 [Other than the above] I/O port
PA4/A20 PA3/A19 PA2/A18 PA1/A17 PA0/A16	1, 2	L	T	[OPE = 0] T [OPE = 1] Keep	T	Address output A20 to A16
	3, 4, 7	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A20 to A16 [Other than the above] I/O port

Port Name	Operating Mode	Reset	Standby Mode	Software Standby Mode	Bus Release State	State Sleep Mode
Port B	1, 2	L	T	[OPE = 0]	T	Address output
				T		A15 to A8
				[OPE = 1] Keep		
	4	T	T	[OPE = 0, address output]	[Address output] T	[Address output]
				T	[Other than the above]	A15 to A8
				[OPE = 1, address output] Keep [Other than the above] Keep	Keep	[Other than the above] I/O port
	3, 7	T	T	[OPE = 0, address output]	[Address output] T	[Address output]
				T	[Other than the above]	A15 to A8
				[OPE = 1, address output] Keep [Other than the above] Keep	Keep	[Other than the above] I/O port
Port C	1, 2	L	T	[OPE = 0]	T	Address output
				T		A7 to A0
				[OPE = 1] Keep		
	4	T	T	[OPE = 0, address output]	[Address output] T	[Address output]
				T	[Other than the above]	A7 to A0
				[OPE = 1, address output] Keep [Other than the above] Keep	Keep	[Other than the above] I/O port

Port Name	Operating Mode	Reset	Standby Mode	Software Standby Mode	Bus Release State	State Sleep Mode
Port C	3, 7	T	T	[OPE = 0, address output] T [OPE = 1, address output] Keep [Other than the above] Keep	[Address output] T [Other than the above] Keep	[Address output] A7 to A0 [Other than the above] I/O port
	1, 2, 4	T	T	T	T	D15 to D8
Port D	3, 7	T	T	[Data bus] T [Other than the above] Keep	[Data bus] T [Other than the above] Keep	[Data bus] D15 to D8 [Other than the above] I/O port
	1, 2, 4	8-bit bus T	T	Keep	Keep	I/O port
Port E	4	16-bit bus T	T	T	T	D7 to D0
	3, 7	8-bit bus T	T	Keep	Keep	I/O port
PF7/ $\phi$	1, 2, 4	16-bit bus T	T	[Data bus] T [Other than the above] Keep	[Data bus] T [Other than the above] Keep	[Data bus] D7 to D0 [Other than the above] I/O port
	3, 7	Clock output T	T	[Clock output] H [Other than the above] Keep	[Clock output] Clock output [Other than the above] Keep	[Clock output] Clock output [Other than the above] Input port

Port Name	Operating Mode	Reset	Standby Mode	Software Standby Mode	Bus Release State	State Sleep Mode
PF6/ $\overline{AS}$	1, 2, 4	H	T	[OPE = 0, $\overline{AS}$ output] T	$\overline{AS}$ output] T [Other than the above]	$\overline{AS}$ output] $\overline{AS}$ [Other than the above]
	3, 7	T		[OPE = 1, $\overline{AS}$ output] H [Other than the above] Keep	Keep [Other than the above] Keep	[Other than the above] I/O port
PF5/ $\overline{RD}$ PF4/ $\overline{HWR}$	1, 2, 4	H	T	[OPE = 0] T [OPE = 1] H	T [Other than the above] Keep	$\overline{RD}$ , $\overline{HWR}$
	3, 7	T		[OPE = 0, $\overline{RD}$ , $\overline{HWR}$ output] T [OPE = 1, $\overline{RD}$ , $\overline{HWR}$ output] H [Other than the above] Keep	$\overline{RD}$ , $\overline{HWR}$ output] T [Other than the above] Keep	$\overline{RD}$ , $\overline{HWR}$ output] $\overline{RD}$ , $\overline{HWR}$ [Other than the above] I/O port
PF3/ $\overline{LWR}$	1, 2, 4	H	T	[OPE = 0, $\overline{LWR}$ output] T	$\overline{LWR}$ output] T [Other than the above]	$\overline{LWR}$ output] $\overline{LWR}$ [Other than the above]
	3, 7	T		[OPE = 1, $\overline{LWR}$ output] H [Other than the above] Keep	Keep [Other than the above] Keep	[Other than the above] I/O port

Port Name	Operating Mode	Reset	Standby Mode	Software Standby Mode	Bus Release State	State Sleep Mode
PF2/ $\overline{\text{LCAS}}$ / $\overline{\text{CS6}}$	1, 2, 4, 7	T	T	[ $\text{OPE} = 0, \overline{\text{LCAS}}$ (DQML) output]	[ $\overline{\text{LCAS}}$ ( $\overline{\text{DQML}}$ ) output]	[ $\overline{\text{LCAS}}$ ( $\overline{\text{DQML}}$ ) output]
				T	T	output]
				[ $\text{OPE} = 1, \overline{\text{LCAS}}$ (DQML) output]	[ $\overline{\text{CS}}$ output]	[ $\overline{\text{LCAS}}$ ( $\overline{\text{DQML}}$ ) output]
				H	[Other than the above]	[ $\overline{\text{CS}}$ output]
				[ $\text{OPE} = 0, \overline{\text{CS}}$ output]	Keep	$\overline{\text{CS}}$
				T	[Other than the above]	[Other than the above]
				[ $\text{OPE} = 1, \overline{\text{CS}}$ output]		I/O port
H	[Other than the above]					
			Keep			
PF1/ $\overline{\text{UCAS}}$ / $\overline{\text{CS5}}$	1, 2, 4, 7	T	T	[ $\text{OPE} = 0, \overline{\text{UCAS}}$ ( $\overline{\text{DQMU}}$ ) output]	[ $\overline{\text{UCAS}}$ ( $\overline{\text{DQMU}}$ ) output]	[ $\overline{\text{UCAS}}$ ( $\overline{\text{DQMU}}$ ) output]
				T	T	output]
				[ $\text{OPE} = 1, \overline{\text{UCAS}}$ ( $\overline{\text{DQMU}}$ ) output]	[ $\overline{\text{CS}}$ output]	[ $\overline{\text{UCAS}}$ ( $\overline{\text{DQMU}}$ ) output]
				H	[Other than the above]	[ $\overline{\text{CS}}$ output]
				[ $\text{OPE} = 0, \overline{\text{CS}}$ output]	Keep	CS
				T	[Other than the above]	[Other than the above]
				[ $\text{OPE} = 1, \overline{\text{CS}}$ output]		I/O port
H	[Other than the above]					
			Keep			
PF0/ $\overline{\text{WAIT}}$ / $\overline{\text{OE}}$	1, 2, 4, 7	T	T	[ $\overline{\text{WAIT}}$ input]	[ $\overline{\text{WAIT}}$ input]	[ $\overline{\text{WAIT}}$ input]
				T	T	$\overline{\text{WAIT}}$
				[ $\overline{\text{OE}}$ output, $\text{OPE} = 0$ ]	[ $\overline{\text{OE}}$ output]	[ $\overline{\text{OE}}$ output]
				T	[Other than the above]	OE
				[ $\overline{\text{OE}}$ output, $\text{OPE} = 1$ ]	Keep	[Other than the above]
				H	[Other than the above]	I/O port
			Keep			

Port Name	Operating Mode	Reset	Standby Mode	Software Standby Mode	Bus Release State	State Sleep Mode
PG6/BREQ	1, 2, 4, 7	T	T	[BREQ input] T [Other than the above] Keep	BREQ input $\overline{\text{BREQ}}$	$\overline{\text{BREQ}}$ input BREQ [Other than the above] I/O port
PG5/BACK	1, 2, 4, 7	T	T	$\overline{\text{BACK}}$ output] $\overline{\text{BACK}}$ [Other than the above] Keep	BACK	$\overline{\text{BACK}}$ output] BACK [Other than the above] I/O port
PG4/ BREQO/ CS4	1, 2, 4, 7	T	T	$\overline{\text{BREQO}}$ output] BREQO [OPE = 0, $\overline{\text{CS}}$ output] T [OPE = 1, CS output] H [Other than the above] Keep	$\overline{\text{BREQO}}$ output] BREQO [ $\overline{\text{CS}}$ output] T [Other than the above] Keep	$\overline{\text{BREQO}}$ output] BREQO [ $\overline{\text{CS}}$ output] CS [Other than the above] I/O port
PG3/CS3 PG2/CS2 PG1/CS1	1, 2, 4, 7	T	T	[OPE = 0, $\overline{\text{CS}}$ output] T [OPE = 1, CS output] H [Other than the above] Keep	[ $\overline{\text{CS}}$ output] T [Other than the above] Keep	[ $\overline{\text{CS}}$ output] CS [Other than the above] I/O port

Port Name	Operating Mode	Reset	Standby Mode	Software Standby Mode	Bus Release State	State Sleep Mode
PG0/ $\overline{CS0}$	1, 2	H	T	[OPE = 0, $\overline{CS}$ output]	$\overline{CS}$ output]	$\overline{CS}$ output]
	3, 4, 7	T		T [OPE = 1, $\overline{CS}$ output] H [Other than the above] Keep	T [Other than the above] Keep	$\overline{CS}$ [Other than the above] I/O port
$\overline{WDTOVF}$	1, 2, 3, 4, 7	H	H	H	H	H*

Legend:

L: Low level

H: High level

Keep: Input port becomes high-impedance, output port retains state

T: High impedance

DDR: Data direction register

OPE: Output port enable

Notes: Indicates the state after the bus cycle being executed is completed.

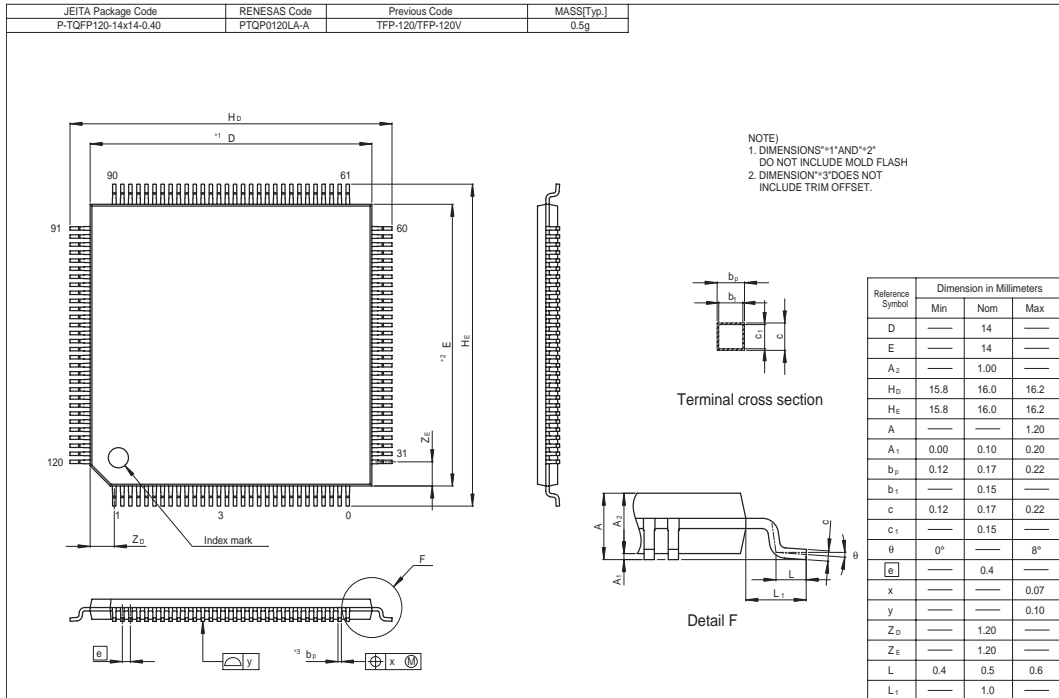
\* Low output if a watchdog overflow occurs when WT/IT is set to 1.



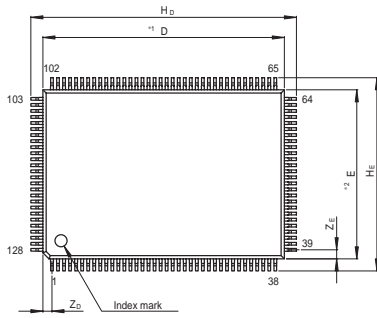
Product		Part No.	Model Marking	Package (Code)
H8S/2368F	F-ZTAT version	HD64F2368	HD64F2368VTE	120-pin TFP (TFP-120, TFP-120V)* <sup>2</sup>
H8S/2367F	F-ZTAT version	HD64F2367	HD64F2367VTE	
H8S/2367F	F-ZTAT version	HD64F2367	HD64F2367F	128-pin* <sup>1</sup> QFP (FP-128B, FP-128B)* <sup>2</sup>
H8S/2366F	F-ZTAT version	HD64F2366	HD64F2366F	128-pin* <sup>1</sup> QFP (FP-128B, FP-128B)* <sup>2</sup>
H8S/2364F	F-ZTAT version	HD64F2364	HD64F2364VTE	120-pin TFP (TFP-120, TFP-120V)* <sup>2</sup>
H8S/2362F	F-ZTAT version	HD64F2362	HD64F2362VTE	
H8S/2361F	F-ZTAT version	HD64F2361	HD64F2361VTE	
H8S/2360F	F-ZTAT version	HD64F2360	HD64F2360VTE	
H8S/2365	Masked ROM version	HD6432365	HD6432365F	128-pin* <sup>1</sup> QFP (FP-128B, FP-128B)* <sup>2</sup>
H8S/2365	Masked ROM version	HD6432365	HD6432365VTE	120-pin TFP (TFP-120, TFP-120V)* <sup>2</sup>
H8S/2363	ROMless version	HD6412363	HD6412363F	128-pin* <sup>1</sup> QFP (FP-128B, FP-128B)* <sup>2</sup>
H8S/2363	ROMless version	HD6412363	HD6412363VTE	120-pin TFP (TFP-120, TFP-120V)* <sup>2</sup>

Notes: When using the optional functions for the F-ZTAT version, ROMless version, which has the common type name, contact a Renesas sales office.

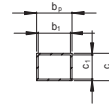
1. Supported only by the H8S/2368 0.18 μm F-ZTAT Group.
2. Pb free version



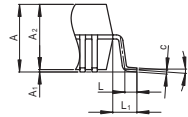
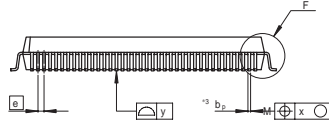
**Figure C.1 Package Dimensions (TFP-120)**



NOTE)  
 1. DIMENSIONS \*1\* AND \*2\*  
 DO NOT INCLUDE MOLD FLASH  
 2. DIMENSION \*3\* DOES NOT  
 INCLUDE TRIM OFFSET.



Terminal cross section



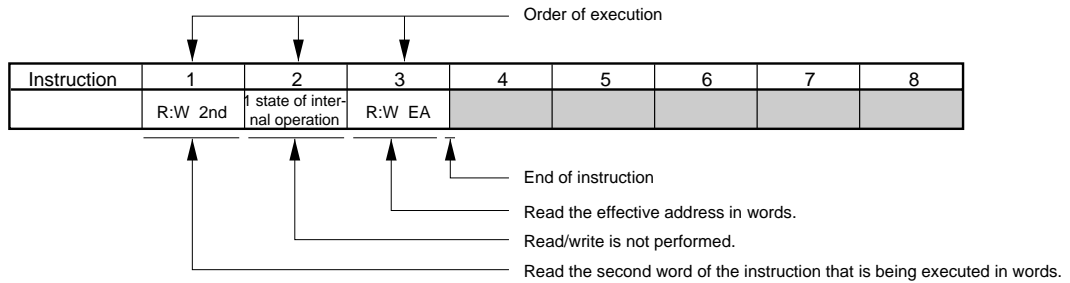
Detail F

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	20	—
E	—	14	—
A <sub>2</sub>	—	2.70	—
H <sub>D</sub>	21.8	22.0	22.2
H <sub>E</sub>	15.8	16.0	16.2
A	—	—	3.15
A <sub>1</sub>	0.00	0.10	0.25
b <sub>D</sub>	0.17	0.22	0.27
b <sub>1</sub>	—	0.20	—
c	0.12	0.17	0.22
c <sub>1</sub>	—	0.15	—
θ	0°	—	8°
—	—	0.5	—
x	—	—	0.10
y	—	—	0.10
Z <sub>D</sub>	—	0.75	—
Z <sub>E</sub>	—	0.75	—
L	0.3	0.5	0.7
L <sub>1</sub>	—	1.0	—

Figure C.2 Package Dimensions (FP-128B)

Table D.1 shows the execution state of each instruction in this LSI.

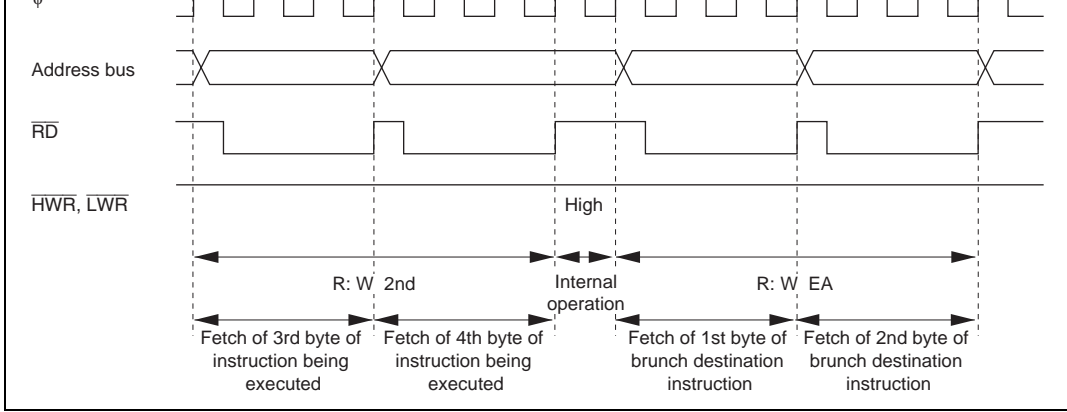
[Explanation of Table Contents]



Legend:

R:B	Reading in bytes
R:W	Reading in words
W:B	Writing in bytes
W:W	Writing in words
:M	Bus right cannot be handed over immediately after this cycle
2nd	Address of second word (3rd and 4th bytes)
3rd	Address of third word (5th and 6th bytes)
4th	Address of fourth word (7th and 8th bytes)
5th	Address of fifth word (9th and 10th bytes)
NEXT	Start address of instruction immediately following the instruction being executed
EA	Effective address
VEC	Vector address

Figure D.1 shows the timing of the address bus,  $\overline{RD}$ ,  $\overline{HWR}$ , and  $\overline{LWR}$  during execution of the sample instruction above (example in "Explanation of Table Contents") with an 8-bit bus, 3-state access, and no wait.



**Figure D.1** Timing of Address Bus,  $\overline{RD}$ ,  $\overline{HWR}$ , and  $\overline{LWR}$  (8-bit bus, 3-state access, no wait)

ADD.B #xx:8,Rd	R:W NEXT			
ADD.B Rs,Rd	R:W NEXT			
ADD.W #xx:16,Rd	R:W 2nd NEXT	R:W NEXT		
ADD.W Rs,Rd	R:W NEXT			
ADD.L #xx:32,ERd	R:W 2nd NEXT	R:W 3rd NEXT	R:W NEXT	
ADD.L ERs,ERd	R:W NEXT			
ADDS #1/2/4,ERd	R:W NEXT			
ADDX #xx:8,Rd	R:W NEXT			
ADDX Rs,Rd	R:W NEXT			
AND.B #xx:8,Rd	R:W NEXT			
AND.B Rs,Rd	R:W NEXT			
AND.W #xx:16,Rd	R:W 2nd NEXT	R:W: NEXT		
AND.W Rs,Rd	R:W NEXT			
AND.L #xx:32,ERd	R:W 2nd NEXT	R:W 3rd NEXT	R:W NEXT	
AND.L ERs,ERd	R:W 2nd NEXT	R:W NEXT		
ANDC #xx:8,CCR	R:W NEXT			
ANDC #xx:8,EXR	R:W 2nd NEXT	R:W NEXT		
BAND #xx:3,Rd	R:W NEXT			

#xx:3,@ERd				NEXT	
BAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W NEXT		
BAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W NEXT	
BAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT
BRA d:8 (BT d:8)	R:W NEXT	R:W EA			
BRN d:8 (BF d:8)	R:W NEXT	R:W EA			
BHI d:8	R:W NEXT	R:W EA			
BLS d:8	R:W NEXT	R:W EA			
BCC d:8 (BHS d:8)	R:W NEXT	R:W EA			
BCS d:8 (BLO d:8)	R:W NEXT	R:W EA			
BNE d:8	R:W NEXT	R:W EA			
BEQ d:8	R:W NEXT	R:W EA			
BVC d:8	R:W NEXT	R:W EA			
BVS d:8	R:W NEXT	R:W EA			
BPL d:8	R:W NEXT	R:W EA			
BMI d:8	R:W NEXT	R:W EA			
BGE d:8	R:W NEXT	R:W EA			
BLT d:8	R:W NEXT	R:W EA			
BGT d:8	R:W NEXT	R:W EA			

## NEXT

BRA d:16 (BT d:16)	R:W 2nd	1 state of internal operation	R:W EA	
BRN d:16 (BF d:16)	R:W 2nd	1 state of internal operation	R:W EA	
BHI d:16	R:W 2nd	1 state of internal operation	R:W EA	
BLS d:16	R:W 2nd	1 state of internal operation	R:W EA	
BCC d:16 (BHS d:16)	R:W 2nd	1 state of internal operation	R:W EA	
BCS d:16 (BLO d:16)	R:W 2nd	1 state of internal operation	R:W EA	
BNE d:16	R:W 2nd	1 state of internal operation	R:W EA	
BEQ d:16	R:W 2nd	1 state of internal operation	R:W EA	
BVC d:16	R:W 2nd	1 state of internal operation	R:W EA	
BVS d:16	R:W 2nd	1 state of internal operation	R:W EA	
BPL d:16	R:W 2nd	1 state of internal operation	R:W EA	
BMI d:16	R:W 2nd	1 state of internal operation	R:W EA	



		internal operation						
BLT d:16	R:W 2nd	1 state of internal operation	R:W EA					
BGT d:16	R:W 2nd	1 state of internal operation	R:W EA					
BLE d:16	R:W 2nd	1 state of internal operation	R:W EA					
BCLR #xx:3,Rd	R:W NEXT							
BCLR #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA				
BCLR #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA				
BCLR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA			
BCLR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA		
BCLR Rn,Rd	R:W NEXT							
BCLR Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA				
BCLR Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA				
BCLR Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA			
BCLR Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA		
BIAND #xx:3,Rd	R:W NEXT							
BIAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W NEXT					
BIAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W NEXT					

#xx:3,@aa:16					NEXT		
BIAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT		
BILD #xx:3,Rd	R:W NEXT						
BILD #xx:3,@ERd	R:W 2nd	R:B EA	R:W NEXT				
BILD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W NEXT				
BILD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W NEXT			
BILD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT		
BIOR #xx:3,Rd	R:W NEXT						
BIOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W NEXT				
BIOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W NEXT				
BIOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W NEXT			
BIOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT		
BIST #xx:3,Rd	R:W NEXT						
BIST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BIST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA			
BIST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BIST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BIXOR #xx:3,Rd	R:W NEXT						
BIXOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W NEXT				



#xx:3,@ERd									NEXT
BOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W NEXT						
BOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W NEXT					
BOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT				
BSET #xx:3,Rd	R:W NEXT								
BSET #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BSET #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BSET Rn,Rd	R:W NEXT								
BSET Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BSET Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BSR d:8	Advanced NEXT	R:W EA	W:W:M Stack (H)	W:W Stack (L)					
BSR d:16	Advanced	R:W 2nd	1 state of internal operation	R:W EA	W:W:M Stack (H)	W:W Stack (L)			
BST #xx:3,Rd	R:W NEXT								
BST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					

#xx:3, @aa:8		EA	NEXT				
BST #xx:3, @aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA		
BST #xx:3, @aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA	
BTST #xx:3, Rd	R:W NEXT						
BTST #xx:3, @ERd	R:W 2nd	R:B EA	R:W NEXT				
BTST #xx:3, @aa:8	R:W 2nd	R:B EA	R:W NEXT				
BTST #xx:3, @aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W NEXT			
BTST #xx:3, @aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT		
BTST Rn, Rd	R:W NEXT						
BTST Rn, @ERd	R:W 2nd	R:B EA	R:W NEXT				
BTST Rn, @aa:8	R:W 2nd	R:B EA	R:W NEXT				
BTST Rn, @aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W NEXT			
BTST Rn, @aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT		
BXOR #xx:3, Rd	R:W NEXT						
BXOR #xx:3, @ERd	R:W 2nd	R:B EA	R:W NEXT				
BXOR #xx:3, @aa:8	R:W 2nd	R:B EA	R:W NEXT				
BXOR #xx:3, @aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W NEXT			
BXOR #xx:3, @aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W NEXT		

	NEXT	internal operation		
CMP.B #xx:8,Rd	R:W NEXT			
CMP.B Rs,Rd	R:W NEXT			
CMP.W #xx:16,Rd	R:W 2nd NEXT	R:W NEXT		
CMP.W Rs,Rd	R:W NEXT			
CMP.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT	
CMP.L ERs,ERd	R:W NEXT			
DAA Rd	R:W NEXT			
DAS Rd	R:W NEXT			
DEC.B Rd	R:W NEXT			
DEC.W #1/2,Rd	R:W NEXT			
DEC.L #1/2,ERd	R:W NEXT			
DIVXS.B Rs,Rd	R:W 2nd NEXT	R:W NEXT	11 states of internal operation	
DIVXS.W Rs,ERd	R:W 2nd NEXT	R:W NEXT	19 states of internal operation	
DIVXU.B Rs,Rd	R:W NEXT		11 states of internal operation	
DIVXU.W Rs,ERd	R:W NEXT		19 states of internal operation	
EEMOV.B	R:W 2nd	2 states of internal operation	R:B EAs *1	W:B EAd *1 R:W NEXT
EEMOV.W	R:W 2nd	2 states of internal operation	R:B EAs *1	W:B EAd *1 R:W NEXT

					times*1		
EXTS.L ERd	R:W NEXT						
EXTU.W Rd	R:W NEXT						
EXTU.L ERd	R:W NEXT						
INC.B Rd	R:W NEXT						
INC.W #1/2,Rd	R:W NEXT						
INC.L #1/2,ERd	R:W NEXT						
JMP @ERn	R:W NEXT	R:W EA					
JMP @aa:24	R:W 2nd	1 state of internal operation	R:W EA				
JMP @ @aa :8	Advanced R:W NEXT	R:W:M aa:8	R:W aa:8	1 state of internal operation	R:W EA		
JSR @ERn	Advanced R:W NEXT	R:W EA	W:W:M Stack (H)	W:W Stack (L)			
JSR @aa:2 4	Advanced R:W 2nd	1 state of internal operation	R:W EA	W:W:M Stack (H)	W:W Stack (L)		
JSR @ @aa :8	Advanced R:W NEXT	R:W:M aa:8	R:W aa:8	W:W:M Stack (H)	W:W Stack (L)	R:W EA	
LDC #xx:8,CCR	R:W NEXT						
LDC #xx:8,EXR	R:W 2nd	R:W NEXT					
LDC Rs,CCR	R:W NEXT						
LDC Rs,EXR	R:W NEXT						

@ERs,CCR			NEXT				
LDC @ERs,EXR	R:W 2nd	R:W	R:W EA NEXT				
LDC@(d:16,ERs),CCR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC@(d:16,ERs),EXR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC@(d:32,ERs),CCR	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA	
LDC@(d:32,ERs),EXR	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA	
LDC @ERs+,CCR	R:W 2nd	R:W NEXT	1 state of internal operation	R:W EA			
LDC @ERs+,EXR	R:W 2nd	R:W NEXT	1 state of internal operation	R:W EA			
LDC @aa:16,CCR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC @aa:16,EXR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA			
LDC @aa:32,CCR	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA		
LDC @aa:32,EXR	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA		
LDM.L @SP+, (ERn-ERn+1) *8	R:W 2nd	R:W NEXT	1 state of internal operation	R:W:M Stack (H) *2	R:W Stack (L) *2		
LDM.L @SP+, (ERn-ERn+2) *8	R:W 2nd	R:W NEXT	1 state of internal operation	R:W:M Stack (H) *2	R:W Stack (L) *2		
LDM.L @SP+, (ERn-ERn+3) *8	R:W 2nd	R:W NEXT	1 state of internal operation	R:W:M Stack (H) *2	R:W Stack (L) *2		
LDMAC ERs,MACH	R:W NEXT	1 state of internal operation					



ERs,MACL	NEXT	internal operation				
MAC @ERn+,@ERm+	R:W 2nd R:W NEXT	R:W EAn	R:W EAm			
MOV.B #xx:8,Rd	R:W NEXT					
MOV.B Rs,Rd	R:W NEXT					
MOV.B @ERs,Rd	R:W NEXT	R:B EA				
MOV.B @(d:16,ERs),Rd	R:W 2nd R:W NEXT	R:W 3rd	R:B EA			
MOV.B @(d:32,ERs),Rd	R:W 2nd R:W 3rd R:W 4th	R:W 3rd	R:W 4th	R:W NEXT	R:B EA	
MOV.B @ERs+,Rd	R:W NEXT	1 state of internal operation	R:B EA			
MOV.B @aa:8,Rd	R:W NEXT	R:B EA				
MOV.B @aa:16,Rd	R:W 2nd R:W NEXT	R:W 3rd	R:B EA			
MOV.B @aa:32,Rd	R:W 2nd R:W 3rd	R:W 3rd	R:W NEXT	R:B EA		
MOV.B Rs,@ERd	R:W NEXT	W:B EA				
MOV.B Rs,@(d:16,ERd)	R:W 2nd R:W NEXT	R:W 3rd	W:B EA			
MOV.B Rs,@(d:32,ERd)	R:W 2nd R:W 3rd R:W 4th	R:W 3rd	R:W 4th	R:W NEXT	W:B EA	
MOV.B Rs,@-ERd	R:W NEXT	1 state of internal operation	W:B EA			
MOV.B Rs,@aa:8	R:W NEXT	W:B EA				

Rs, @aa:16			NEXT			
MOV.B Rs, @aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:B EA		
MOV.W #xx:16,Rd	R:W 2nd	R:W NEXT				
MOV.W Rs,Rd	R:W NEXT					
MOV.W @ERs,Rd	R:W NEXT	R:W EA				
MOV.W @(d:16,ERs), Rd	R:W 2nd	R:W NEXT	R:W EA			
MOV.W @(d:32,ERs), Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA	
MOV.W @ERs+,Rd	R:W NEXT	1 state of internal operation	R:W EA			
MOV.W @aa:16,Rd	R:W 2nd	R:W NEXT	R:W EA			
MOV.W @aa:32,Rd	R:W 2nd	R:W 3rd	R:W NEXT	R:B EA		
MOV.W Rs, @ERd	R:W NEXT	W:W EA				
MOV.W Rs, @(d:16,ERd)	R:W 2nd	R:W NEXT	W:W EA			
MOV.W Rs, @(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA	
MOV.W Rs, @- ERd	R:W NEXT	1 state of internal operation	W:W EA			
MOV.W Rs, @aa:16	R:W 2nd	R:W NEXT	W:W EA			
MOV.W Rs, @aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA		
MOV.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT			

ERs,ERd	NEXT							
MOV.L @ERs,ERd	R:W 2nd	R:W NEXT	R:W:M EA	R:W EA+2				
MOV.L @(d:16,ERs), ERd	R:W 2nd	R:W 3rd	R:W NEXT	R:W:M EA	R:W EA+2			
MOV.L @(d:32,ERs), ERd	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W:M EA	R:W EA+2	
MOV.L @ERs+,ERd	R:W 2nd	R:W NEXT	1 state of internal operation	R:W:M EA	R:W EA+2			
MOV.L @aa:16,ERd	R:W 2nd	R:W 3rd	R:W NEXT	R:W:M EA	R:W EA+2			
MOV.L @aa:32,ERd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W:M EA	R:W EA+2		
MOV.L ERs,@ERd	R:W 2nd	R:W NEXT	W:W:M EA	W:W EA+2				
MOV.L ERs, @(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W:M EA	W:W EA+2			
MOV.L ERs, @(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W:M EA	W:W EA+2	
MOV.L ERs,@-ERd	R:W 2nd	R:W NEXT	1 state of internal operation	W:W:M EA	W:W EA+2			
MOV.L ERs,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W:M EA	W:W EA+2			
MOV.L ERs,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W:M EA	W:W EA+2		
MOVFPPE @aa:16,Rd	Not available in this LSI.							
MOVTPPE Rs,@aa:16								
MULXS.B Rs,Rd	R:W 2nd	R:W NEXT	2 status of internal operation					
MULXS.W Rs,ERd	R:W 2nd	R:W NEXT	3 status of internal operation					

Rs,Rd	NEXT	operation	
MULXU.W Rs,ERd	R:W NEXT	3 status of internal operation	
NEG.B Rd	R:W NEXT		
NEG.W Rd	R:W NEXT		
NEG.L ERd	R:W NEXT		
NOP	R:W NEXT		
NOT.B Rd	R:W NEXT		
NOT.W Rd	R:W NEXT		
NOT.L ERd	R:W NEXT		
OR.B #xx:8,Rd	R:W NEXT		
OR.B Rs,Rd	R:W NEXT		
OR.W #xx:16,Rd	R:W 2nd NEXT	R:W NEXT	
OR.W Rs,Rd	R:W NEXT		
OR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT
OR.L ERs,ERd	R:W 2nd	R:W NEXT	
ORC #xx:8,CCR	R:W NEXT		
ORC #xx:8,EXR	R:W 2nd	R:W NEXT	
POP.W Rn	R:W NEXT	1 state of internal operation	R:W EA

		NEXT	internal operation	EA	EA+2			
PUSH.W Rn	R:W NEXT	1 state of internal operation	W:W EA					
PUSH.L ERn	R:W 2nd	R:W NEXT	1 state of internal operation	W:W:M EA	W:W EA+2			
ROTL.B Rd	R:W NEXT							
ROTL.B #2,Rd	R:W NEXT							
ROTL.W Rd	R:W NEXT							
ROTL.W #2,Rd	R:W NEXT							
ROTL.L ERd	R:W NEXT							
ROTL.L #2,ERd	R:W NEXT							
ROTR.B Rd	R:W NEXT							
ROTR.B #2,Rd	R:W NEXT							
ROTR.W Rd	R:W NEXT							
ROTR.W #2,Rd	R:W NEXT							
ROTR.L ERd	R:W NEXT							
ROTR.L #2,ERd	R:W NEXT							
ROTXL.B Rd	R:W NEXT							
ROTXL.B #2,Rd	R:W NEXT							
ROTXL.W Rd	R:W NEXT							

#2,Rd	NEXT					
ROTXL.L ERd	R:W NEXT					
ROTXL.L #2,ERd	R:W NEXT					
ROTXR.B Rd	R:W NEXT					
ROTXR.B #2,Rd	R:W NEXT					
ROTXR.W Rd	R:W NEXT					
ROTXR.W #2,Rd	R:W NEXT					
ROTXR.L ERd	R:W NEXT					
ROTXR.L #2,ERd	R:W NEXT					
RTE	R:W NEXT	R:W Stack (EXR)	R:W Stack (H)	R:W Stack (L) internal operation	1 state of R:W* <sup>3</sup>	
RTS <small>Advanced</small>	R:W NEXT	R:W:M Stack (H)	R:W Stack (L) internal operation	1 state of R:W* <sup>3</sup>		
SHAL.B Rd	R:W NEXT					
SHAL.B #2,Rd	R:W NEXT					
SHAL.W Rd	R:W NEXT					
SHAL.W #2,Rd	R:W NEXT					
SHAL.L ERd	R:W NEXT					
SHAL.L #2,ERd	R:W NEXT					
SHAR.B Rd	R:W NEXT					

	NEXT		
SHAR.W Rd	R:W NEXT		
SHAR.W #2,Rd	R:W NEXT		
SHAR.L ERd	R:W NEXT		
SHAR.L #2,ERd	R:W NEXT		
SHLL.B Rd	R:W NEXT		
SHLL.B #2,Rd	R:W NEXT		
SHLL.W Rd	R:W NEXT		
SHLL.W #2,Rd	R:W NEXT		
SHLL.L ERd	R:W NEXT		
SHLL.L #2,ERd	R:W NEXT		
SHLR.B Rd	R:W NEXT		
SHLR.B #2,Rd	R:W NEXT		
SHLR.W Rd	R:W NEXT		
SHLR.W #2,Rd	R:W NEXT		
SHLR.L ERd	R:W NEXT		
SHLR.L #2,ERd	R:W NEXT		
SLEEP	R:W NEXT	Internal operation: M	

STC EXR,Rd	R:W NEXT						
STC CCR,@ERd	R:W 2nd NEXT	R:W	W:W EA				
STC EXR,@ERd	R:W 2nd NEXT	R:W	W:W EA				
STC CCR, @(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
STC EXR, @(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
STC CCR, @(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA	
STC EXR, @(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA	
STC CCR,@- ERd	R:W 2nd	R:W NEXT	1 state of internal operation	W:W EA			
STC EXR,@- ERd	R:W 2nd	R:W NEXT	1 state of internal operation	W:W EA			
STC CCR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
STC EXR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA			
STC CCR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA		
STC EXR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA		
STM.L (ERn- ERn+1), @-SP*8	R:W 2nd	R:W NEXT	1 state of internal operation	W:W:M Stack (H)*2	W:W Stack (L) *2		
STM.L (ERn- ERn+2), @-SP*8	R:W 2nd	R:W NEXT	1 state of internal operation	W:W:M Stack (H)*2	W:W Stack (L) *2		



ERn+3), @-SP* <sup>8</sup>	NEXT	internal operation	Stack (H)* <sup>2</sup>	Stack (L) * <sup>2</sup>					
STMAC MACH,ERd	R:W NEXT								
STMAC MACL,ERd	R:W NEXT								
SUB.B Rs,Rd	R:W NEXT								
SUB.W #xx:16,Rd	R:W 2nd NEXT	R:W							
SUB.W Rs,Rd	R:W NEXT								
SUB.L #xx:32,ERd	R:W 2nd NEXT	R:W 3rd	R:W NEXT						
SUB.L ERs,ERd	R:W NEXT								
SUBS #1/2/4,ERd	R:W NEXT								
SUBX #xx:8,Rd	R:W NEXT								
SUBX Rs,Rd	R:W								
TAS @ERd* <sup>7</sup>	R:W 2nd NEXT	R:W	R:B:M EA	W:B EA					
TRAPA Advanced #x:2	R:W NEXT	1 state of internal operation	W:W Stack (L)	W:W Stack (H)	W:W Stack (EXR)	R:W:M VEC	R:W VEC+2	1 state of internal operation	R:W* <sup>6</sup>
XOR.B #xx8,Rd	R:W NEXT								
XOR.B Rs,Rd	R:W NEXT								
XOR.W #xx:16,Rd	R:W 2nd NEXT	R:W							
XOR.W Rs,Rd	R:W NEXT								
XOR.L #xx:32,ERd	R:W 2nd NEXT	R:W 3rd	R:W NEXT						

ERs,ERd			NEXT						
XORC #xx:8,CCR	R:W NEXT								
XORC #xx:8,EXR	R:W 2nd NEXT	R:W	R:W						
Reset exception handling	Advanced VEC	R:W:M VEC	R:W VEC+2	1 state of internal operation	R:W* <sup>4</sup>				
Interrupt exception handling	Advanced	R:W* <sup>5</sup>	1 state of internal operation	W:W Stack (L)	W:W Stack (H)	W:W Stack (EXR)	R:W:M VEC	R:W VEC+2	1 state of R:W* <sup>6</sup> internal operation

- Notes:
1. EAs is the ER5 value and EAd the ER6 value. 1 is added to each of them after execution.  
n is the initial value of R4L or R4, and the processing is not executed when n = 0.
  2. Repeated two times when two registers are stored/retrieved, three times when three registers are stored/retrieved, and four times when four registers are stored/retrieved.
  3. Start address on returning.
  4. Start address of program.
  5. Prefetch address that is obtained by adding 2 to the saved PC.  
Reading is not performed on returning from sleep mode or software standby mode, and this is regarded as internal operation.
  6. Start address of interrupt handling routine.
  7. Registers ER0, ER1, ER4, and ER5 are used for a TAS instruction.
  8. Registers ER0 to ER6 are used for an STM/LDM instruction.

16-Bit Timer Pulse Unit.....	401	BusController	
Buffer Operation .....	447	Valid Strobes.....	152
Cascaded Operation .....	452	Clock Pulse Generator .....	813
Input Capture Function .....	444	PLL Circuit .....	818
Phase Counting Mode.....	459	Condition Field .....	47
PWM Modes .....	454	Condition-Code Register.....	32
Synchronous Operation.....	445	CPU Operating Modes .....	24
8-Bit Timer		Advanced Mode .....	26
16-Bit Counter Mode .....	520	Normal Mode .....	24
Compare Match Count Mode .....	520	D/A Converter.....	681
Operation with Cascaded Connection.	520	Data Transfer Controller	
Pulse Output .....	515	Block Transfer Mode .....	310
TCNT Incrementation Timing .....	516	Data Transfer Controller .....	293
8-Bit Timers.....	507	Activation by Software .....	316
A/D Converter .....	661	Chain Transfer .....	311
A/D Conversion Time.....	670	Chain Transfer when Counter = 0.....	318
A/D Converter Activation.....	468	DTC Vector Table.....	302
External Trigger.....	673	Interrupts .....	312
Scan Mode .....	669	Location of Register Information.....	302
Single Mode.....	669	Normal Mode .....	308, 317
Address Space.....	28	Repeat Mode .....	309
Addressing Mode.....	49	Software Activation .....	320
Absolute Address.....	50	DMA Controller.....	213
Immediate .....	51	Activation by Auto-Request.....	243
Memory Indirect .....	51	Activation by External Request .....	242
Program-Counter Relative .....	51	Full Address Mode.....	268
Register Direct.....	49	Interrupt Sources .....	287
Register Indirect.....	49	Repeat Mode .....	252
Register Indirect with Displacement....	50	Sequential Mode .....	246
Register Indirect with Post-Increment ..	50	Short Address Mode.....	267
Register indirect with pre-decrement....	50	Single Address Mode .....	256
Bcc .....	45	Transfer Modes .....	244
Bus Controller.....	119	Effective Address.....	49, 52, 53
Basic Bus Interface .....	150	Effective Address Extension .....	47
Basic Timing.....	153	Exception Handling.....	79
DRAM Interface .....	165	Interrupts .....	84
Read Strobe (RD) Timing.....	162	Reset exception handling .....	81
		Stack Status after Exception Handling..	86

Extended Control Register (EXR) .....	31	Stop condition .....	644
Flash Memory .....	689	Instruction Set .....	37
Boot Mode .....	702	Arithmetic Operations Instructions .....	40
Erase/Erase-Verify .....	708	Bit Manipulation Instructions .....	43
erasing units .....	695	Block Data Transfer Instruction .....	47
Error Protection .....	710	Branch Instructions .....	45
Hardware Protection .....	710	Data Transfer Instructions .....	39
Program/Program-Verify .....	706	Logic Operations Instructions .....	42
Programmer Mode .....	711	Shift Instructions .....	42
programming units .....	695	System Control Instruction .....	46
Software Protection .....	710	Interrupt Control Modes .....	108
Flash Memory (0.18- $\mu$ m F-ZTAT Version)		Interrupt Controller .....	89
Boot Mode .....	748	Interrupt Mask Bit .....	33
Communications Protocol .....	784	Interrupts	
Error Protection .....	779	ADI .....	673
Flash MAT Configuration .....	722	CMIA .....	521
Hardware Protection .....	778	CMIB .....	521
Mode Comparison .....	721	ERI .....	620
off-board programming mode .....	718	OVI .....	521
On-Board Programming .....	748	RXI .....	620
on-board programming mode .....	718	TCI1U .....	467
Procedure Program .....	768	TCI1V .....	467
Programmer Mode .....	782	TCI2U .....	467
Protection .....	778	TCI2V .....	467
Serial Communication Interface		TCI3V .....	467
Specification .....	782	TCI4U .....	467
Software Protection .....	779	TCI4V .....	467
user boot MAT .....	781	TCI5U .....	467
user boot memory MAT .....	717	TCI5V .....	467
User Boot Mode .....	764	TEI .....	620
user MAT .....	781	TGI0A .....	467
user memory MAT .....	717	TGI0B .....	467
User Program Mode .....	753	TGI0C .....	467
General Registers .....	30	TGI0D .....	467
I <sup>2</sup> C Bus Data Formats .....	643	TGI0E .....	467
I <sup>2</sup> C Bus Interface (IIC) .....	629	TGI1A .....	467
acknowledge .....	644, 646, 648, 650	TGI1B .....	467
general call address .....	641	TGI2A .....	467

TGI3B.....	467	DMABCR .....	226
TGI3C.....	467	DMATCR .....	239
TGI3D.....	467	DMAWER .....	238
TGI4A.....	467	DRACCR.....	140, 843, 866
TGI4B.....	467	DRAMCR .....	135, 843, 855, 866
TGI5A.....	467	DTCER .....	299, 844, 856, 867
TGI5B.....	467	DTVECR.....	300
TXI .....	620	EBR1.....	700
WOVI .....	536	EBR2.....	700
MCU Operating Modes .....	57	FLMCR1 .....	698
Operation Field .....	47	FLMCR2 .....	699
Program Counter.....	31	ICMR .....	636
Programmable Pulse Generator .....	485	ICSR.....	639, 642
Non-Overlapping Pulse Output.....	499	IER .....	95, 845, 857, 868
Programming/Erasing Interface Parameter		INTCR .....	92, 845, 857, 868
Download pass/fail result parameter...	736	IPR .....	93, 840, 851, 863
Flash erase block select parameter.....	744	IrCR .....	573, 841, 851, 864
Flash multipurpose address area		ISCR.....	851, 864
parameter .....	741	ISCRL .....	96
Flash multipurpose data destination		ISR .....	99, 845, 857, 868
parameter .....	741	ITSR.....	100, 864
Flash pass/fail parameter .....	745	MDCR.....	58, 845, 857, 868
Flash programming/erasing frequency		MRA .....	295, 840, 850, 863
parameter .....	738	MRB.....	297, 840, 850, 863
Programming/Erasing Interface Register	728	MSTPCR.....	828, 845, 857, 868
Register		NDER.....	488, 845, 857, 868
ABWCR.....	123, 843, 854, 866	NDR .....	490, 845, 857, 868
ADCR .....	668, 847, 860, 870	P1DDR.....	327, 841, 852, 864
ADCSR.....	666, 847, 860, 870	P1DR.....	328, 846, 858, 869
ADDR.....	664, 847, 859, 870	P2DDR.....	337, 841, 852, 864
ASTCR .....	123, 843, 854, 866	P2DR.....	338, 846, 858, 869
BCR .....	133, 843, 855, 866	P3DDR.....	347, 841, 852, 864
BROMCR .....	132, 843, 854, 866	P3DR.....	348, 846, 858, 869
BRR .....	564, 846, 858, 869	P3ODR.....	349, 842, 852, 864
CRA .....	298, 840, 850, 863	P5DDR.....	357, 841, 852, 864
CRB .....	298, 840, 850, 863	P5DR.....	357, 846, 858, 869
CSACR.....	130, 843, 854, 866	P8DDR.....	360, 841, 852, 864
DACR .....	684, 860	P8DR.....	361, 846, 858, 869

PAODR.....	367, 842, 852, 864	RDR .....	546, 846, 858, 869
PAPCR.....	367, 841, 852, 864	REFCR.....	141, 843, 855, 866
PBDDR.....	373, 841, 852, 864	RSR.....	546
PBDR.....	374, 846, 858, 869	RSTCSR.....	533, 848, 860, 871
PBPCR.....	375, 841, 852, 864	RTCNT .....	144, 843, 855, 866
PCDDR.....	377, 841, 852, 864	RTCOR .....	144, 843, 855, 866
PCDR.....	378, 846, 858, 869	SAR.....	297, 641, 840, 850, 863
PCPCR.....	379, 841, 852, 864	SBYCR .....	826, 845, 857, 868
PCR.....	492, 845, 857, 868	SCKCR .....	814, 845, 857, 868
PDDDR.....	381, 841, 852, 864	SCMR .....	563, 846, 858, 869
PDDR.....	382, 846, 858, 869	SCR.....	551, 846, 858, 869
PDPCR.....	383, 841, 852, 864	SEMR.....	574, 840, 851
PEDDR .....	385, 841, 852, 864	SMR .....	547, 846, 858, 869
PEDR .....	386, 846, 858, 869	SSIER.....	101, 841, 851, 864
PEPCR.....	387, 841, 852, 864	SSR .....	556, 846, 858, 869
PFCR0 .....	368, 841, 852, 864	SYSCR.....	58, 845, 857, 868
PFCR1 .....	369, 841, 852, 864	TCNT .....	437, 509, 531, 847, 848, ..... 860, 861, 871, 872
PFCR2 .....	350, 841, 852, 864	TCOR.....	510, 847, 860
PFDDR .....	389, 841, 852, 864	TCR.....	408, 510, 847, 848, ..... 860, 861, 871, 872
PFDR .....	390, 846, 858, 869	TCSR .....	512, 531, 847, 848, 860, 871
PGDDR.....	395, 841, 852, 864	TDR .....	546, 846, 858, 869
PGDR.....	396, 846, 858, 869	TGR .....	437, 848, 861, 872
PLLCR.....	815, 845, 857, 868	TIER.....	432, 848, 861, 872
PMR.....	493, 845, 857, 868	TIOR.....	414, 848, 861, 872
PODR.....	489, 845, 857, 868	TMDR.....	413, 848, 861, 872
PORT1 .....	328, 845, 857, 868	TSR .....	434, 547, 848, 861, 872
PORT2 .....	338, 845, 857, 868	TSTR.....	437, 860, 871
PORT3 .....	348, 845, 857, 868	TSYR .....	438, 860, 871
PORT4 .....	354, 845, 857, 868	Wait Control.....	161
PORT5 .....	358, 845, 857, 868	WTCR.....	124, 843, 854, 866
PORT8 .....	361, 845, 857, 868	Register Field.....	47
PORT9 .....	363, 845, 857, 868	Registers	
PORTA .....	366, 846, 858, 869	FCCS.....	726
PORTB .....	374, 846, 858, 869	FECS .....	726
PORTC .....	378, 846, 858, 869	FKEY .....	726
PORTD .....	382, 846, 858, 869	FMATS .....	726
PORTE.....	386, 846, 858, 869		
PORTF.....	390, 846, 858, 869		

FVACR .....	727	parity error .....	584
Reset .....	81	Smart Card Interface Mode .....	604
Serial Communication Interface .....	541	stack pointer (SP) .....	30
Asynchronous Mode .....	576	Trace Bit .....	31
bit rates .....	564	TRAPA .....	51
Break .....	622	Watchdog Timer .....	529
Clocked Synchronous Mode .....	595	Interval Timer Mode .....	535
framing error .....	584	Watchdog Timer Mode .....	534





---

**Renesas 16-Bit Single-Chip Microcomputer  
Hardware Manual  
H8S/2368 Group**

Publication Date: 1st Edition, March 2002

Rev.6.00, March 18, 2009

Published by: Sales Strategic Planning Div.  
Renesas Technology Corp.

Edited by: Customer Support Department  
Global Strategic Communication Div.  
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

---



## RENESAS SALES OFFICES

<http://www.renesas.com>

Refer to "<http://www.renesas.com/en/network>" for the latest and detailed information.

### **Renesas Technology America, Inc.**

450 Holger Way, San Jose, CA 95134-1368, U.S.A  
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

### **Renesas Technology Europe Limited**

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

### **Renesas Technology (Shanghai) Co., Ltd.**

Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120  
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

### **Renesas Technology Hong Kong Ltd.**

7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong  
Tel: <852> 2265-6688, Fax: <852> 2377-3473

### **Renesas Technology Taiwan Co., Ltd.**

10th Floor, No.99, Fushing North Road, Taipei, Taiwan  
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

### **Renesas Technology Singapore Pte. Ltd.**

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
Tel: <65> 6213-0200, Fax: <65> 6278-8001

### **Renesas Technology Korea Co., Ltd.**

Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea  
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

### **Renesas Technology Malaysia Sdn. Bhd**

Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: <603> 7955-9390, Fax: <603> 7955-9510





# H8S/2368 Group Hardware Manual



Renesas Electronics Corporation

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0050-0600

## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [16-bit Microcontrollers - MCU category](#):*

*Click to view products by [Renesas manufacturer](#):*

Other Similar products are found below :

[M30302FCFP#U3](#) [MB90F036APMC-GSE1](#) [MB90F428GCPFR-GSE1](#) [MB96F683RBPMC-GSAE1](#) [R5F10MMGDFB#30](#)  
[R5F111PGGFB#30](#) [R5F117BCGNA#20](#) [DF3026XBL25V](#) [DF36014GFTV](#) [DF36014GFXV](#) [DF36034GFPV](#) [R5F11B7EANA#U0](#)  
[R5F21172DSP#U0](#) [MB90092PF-G-BNDE1](#) [MB90F335APMC1-G-SPE1](#) [MB90F345CAPFR-GSE1](#) [MB90F568PMCR-GE1](#)  
[MB96F395RSAPMC-GSE2](#) [DF36024GFXV](#) [UPD78F1018F1-BA4-A](#) [MB96F018RBPMC-GSE1](#) [MB90F867ASPFR-GE1](#)  
[M30290FCHP#U3A](#) [DF2239FA20IV](#) [R5F117BCGFP#30](#) [LC88F58B0AU-SQFPH](#) [MB90F548GPF-GE1](#) [MB90214PF-GT-310-BND-AE1](#)  
[MB90F342CESPQC-GSE2](#) [MB90F428GAPF-GSE1](#) [ML62Q504H-NNNTBWBX](#) [S912ZVH128F2VLL](#) [UPD78F1500AGK-GAK-AX](#)  
[HD64F3337SF16V](#) [MB90F428GCPF-GSE1](#) [MB90F342ESPMC-G-JNE1](#) [MB90022PF-GS-358E1](#) [MB96F395RWAPMC-GSE2](#)  
[MB96395RSAPMC-GS-110E2](#) [MB90F883CSPMC-GE1](#) [S912ZVHY64F1CLL](#) [S912ZVHY64F1VLQ](#) [ST10F280](#) [MB96F338RSAPMCR-](#)  
[GK5E2](#) [CY90096PF-G-002-BND-ERE1](#) [ML62Q1569-NNNGAZ0AX](#) [ML62Q1739-NNNGAZ0AX](#) [ML62Q1749-NNNGAZ0AX](#)  
[ML62Q1579-NNNGAZ0AX](#) [ML62Q1559-NNNGAZ0AX](#)