To all our customers

Regarding the change of names mentioned in the document, such as Hitachi Electric and Hitachi XX, to Renesas Technology Corp.

The semiconductor operations of Mitsubishi Electric and Hitachi were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: http://www.renesas.com

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

Cautions

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

Hitachi 16-Bit Single-Chip Microcomputer

H8S/2214, H8S/2214 F-ZTAT™

Hardware Manual

ADE-602-213A
Rev. 2.0
7/26/01
Hitachi Ltd.

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

Preface

This LSI is a single-chip microcomputer made up of the H8S/2000 CPU with an internal 32-bit architecture as its core, and the peripheral functions required to configure a system.

This LSI is equipped with ROM, RAM, a bus controller, data transfer controller (DTC), a DMA controller (DMAC), two types of timers, a serial communication interface (SCI), a D/A converter, an A/D converter, and I/O ports as on-chip supporting modules. This LSI is suitable for use as an embedded processor for high-level control systems. Its on-chip ROM are flash memory (F-ZTAT™*) and mask ROM that 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 Hitachi, Ltd.

Target Users: This manual was written for users who will be using the H8S/2214, H8S/2214F-ZTAT™ in the design of application systems. Members of this audience 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/2214, H8S/2214F-ZTAT™ to the above audience. Refer to the H8S/2600 Series, H8S/2000 Series Programming Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the H8S/2600 Series, H8S/2000 Series Programming Manual.
- In order to understand the details of a register when its name is known
  The addresses, bits, and initial values of the registers are summarized in Appendix B, Internal I/O Registers.
  Example: Bit order: The MSB is on the left and the LSB is on the right.

Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require.
http://www.hitachi.co.jp/Sicd/English/Products/micome.htm

H8S/2214, H8S/2214F-ZTAT™ manuals:

Manual Title	ADE No.
H8S/2214, H8S/2214F-ZTAT™ Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Programming Manual	ADE-602-083

Users manuals for development tools:

Manual Title	ADE No.
C/C++ Complier, Assembler, Optimized Linkage Editor User's Manual	ADE-702-247
Simulator Debugger (for Windows) Users Manual	ADE-702-037
Hitachi Embedded Workshop Users Manual	ADE-702-201

Application Notes:

Manual Title	ADE No.
H8S Series Technical Q & A	ADE-502-059

List of Items Revised or Added for This Version

Section		Title	Page	Revisions (See Manual for Details)
Section 1 Overview	1.1	Overview	2	Table 1-1 Overview Description of DMA Controller (DMAC) amended.
	1.2	Internal Block Diagrams	5	Figure 1-1 H8S/2214 Internal Block Diagram amended.
	1.3.1	Pin Arrangements	6	Figure 1-2 H8S/2214 Pin Arrangement (TFP-100B, TFP-100G: Top View) amended.
	1.3.2	Pin Functions in Each Operating Mode	9	Table 1-2 Pin Functions in Each Operating Mode Mode of pin Nos. 34 and 35 revised from 4 to 7.
	1.3.3	Pin Functions	14	Table 1-3 Pin Functions. Description of DMA Controller (DMAC) amended.
Section 7 DMA Controller	7.1.1	Features	163	Description of Short Address Mode amended.
	7.1.2	Block Diagram	164	Figure 7-1 Block Diagram of DMAC amended.
	7.1.3	Overview of Functions	165	Table 7-1 Overview of DMAC Functions (Short Address Mode) Description of Single Address Mode deleted.
	7.1.4	Pin Configuration	167	Description of Pin Configuration amended. Table 7-3 DMAC Pins amended.
	7.2.4	DMA Control Register (DMACR)	173	Description of Bit 4: Data Transfer Direction (DTDIR) amended, and Table amended.
	7.2.5	DMA Band Control Register (DMABCR)	176	DMABCRH of Bits 13 and 12 changed to "—." Bits 13 and 12—Reserved bits. Description of 12 and 13 grouped together.
	7.3.5	DMA Band Control Register (DMABCR)	189	Revised as follows: Bits 10 and 8—Reserved (DTA1A, DTA0A): Reserved bits in full address mode. Read and write possible.

Section		Title	Page	Revisions (See Manual for Details)
Section 7 DMA Controller	7.5	Operation	—	7.5.5 Single Address Mode deleted.
			—	7.5.11 DMAC Bus Cycles (Single Address Mode) deleted.
			—	7.5.12 Write Data Buffer Function deleted.
	7.5.1	Transfer Modes	196	Table 7-6 DMAC Transfer Modes Single Address Mode deleted. Note revise.
			—	Description of Single Address Mode deleted.
	7.5.7	DMAC Activation Sources	—	Description of Single Address Mode deleted.
	7.5.11	Relation between the DMAC, External Bus Requests, and the DTC	229	Description of Refresh Cycle deleted.
	7.7	Usage Notes	235	Module Stop Description of DACK pin enable (FAE=0 and SAE=1) deleted.
			—	Write data buffer function deleted.
Section 9 I/O Ports	9.1	Overview	266	Table 9-1 H8S/2214 Port Functions Description of port 1 DMAC output pins (DACK0 and DACK1) deleted.
			269	Description of DMAC output pins (DACK0 and DACK1) related to port 1 deleted.
Section 10 16-Bit Timer Pulse Unit (TPU)	10.1.2	Block Diagram	345	Figure 10-1 Block Diagram of H8S/2214 TPU A/D conversion start request signal deleted.
Section 11 Watchdog Timer (WDT)	11.2.2	Timer Control/Status Register (TCSR)	419	Table Bit 7: Overflow Flag (OVH) amended and Note added.
	11.2.4	Notes on Register Access	422	Writing to RSTCSR H'FFBB in description changed to H'FF76.

Section		Title	Page	Revisions (See Manual for Details)
Section 15 ROM	15.4.3	Mode Transitions	522	PF3=1 added to Notes: 2 in Figure 15-3 Flash Memory State Transitions.
	15.13	Flash Memory Programmer Mode	557	PF3 added to Table 15-12 Programmer Mode Pin Settings.
	15.13.1	Socket Adapter Pin Correspondence Diagram	558	Figure 15-20 Socket Adapter Pin Correspondence Diagram amended.
Section 18 Electrical Characteristics	18.1	Absolute Maximum Ratings	603	Preliminary deleted from Table 18-1 Absolute Maximum Ratings.
	18.2	Power Supply Voltage and Operating Frequency Range	604	Preliminary deleted from Figure 18-1 Power Supply Voltage and Operating Frequency Range.
	18.3	DC Characteristics	605	Table 18-2 DC Characteristics (1) amended. Preliminary deleted.
			607	Table 18-3 DC Characteristics (2) amended. Preliminary deleted, and Notes amended.
			608	Table 18-4 DC Characteristics (3) amended. Preliminary deleted, and Notes amended.
			609	Preliminary deleted from Table 18-5 Permissible Output Currents.
	18.4.1	Clock Timing	610	Preliminary deleted from Table 18-6 Clock Timing.
	18.4.2	Control Signal Timing	611	Preliminary deleted from Table 18-7 Control Signal Timing.
	18.4.3	Bus Timing	613, 614	Table 18-8 Bus Timing amended. Preliminary deleted.
	18.4.4	Timing of On-Chip Supporting Modules	620	Table 18-9 Timing of On-Chip Supporting Modules amended. Preliminary deleted.
	18.4.5	DMAC Timing	623	added.
	18.5	D/A Convervion Characteristics	624	Preliminary deleted from Table 18-11 D/A Conversion Characteristics.

Section		Title	Page	Revisions (See Manual for Details)
Section 18 Electrical Characteristics	18.6	Flash Memory Characteristics	625	Table 18-12 Flash Memory Characteristics amended. Preliminary deleted.
Appendix	B.1	Address	707	Address H'FF66 bits 5 and 4 revised to "—."
	B.2	Functions	764	Address H'FF66 bits 13 and 12 revised Note added.
	F	Product Code Lineup	826	Table F-1 H8S/2214 Product Code Lineup amended.

Contents

Figures

Tables

Section 1 Overview

1.1 Overview

The H8S/2214 is a microcomputer (MCU: microcomputer unit), built around the H8S/2000 CPU, employing Hitachi's proprietary architecture, and equipped with the on-chip peripheral functions necessary for system configuration.

The H8S/2000 CPU has an internal 32-bit architecture, is provided with sixteen 16-bit general registers and a concise, optimized instruction set designed for high-speed operation, and can address a 16-Mbyte linear address space. The instruction set is upward-compatible with H8/300 and H8/300H CPU instructions at the object-code level, facilitating migration from the H8/300, H8/300L, or H8/300H Series.

On-chip peripheral functions required for system configuration include DMA controller (DMAC) data transfer controller (DTC) bus masters, ROM and RAM memory, a16-bit timer-pulse unit (TPU), watchdog timer (WDT), serial communication interface (SCI), D/A converter, and I/O ports.

The on-chip ROM is either flash memory (F-ZTAT™*) or mask ROM, with a capacity of 128 kbytes. ROM is connected to the CPU via a 16-bit data bus, enabling both byte and word data to be accessed in one state. Instruction fetching has been speeded up, and processing speed increased.

Four operating modes, modes 4 to 7, are provided, and there is a choice of single-chip mode or external expansion mode.

The features of the H8S/2214 are shown in Table 1-1.

Note: * F-ZTAT™ is a trademark of Hitachi, Ltd.

Table 1-1 Overview

Item	Specification
CPU	• General-register machine — Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers) • High-speed operation suitable for realtime control — Maximum clock rate 16 MHz — High-speed arithmetic operations (at 16 MHz operation) 8/16/32-bit register-register add/subtract: 62.5 ns 16 × 16-bit register-register multiply : 1250 ns 32 ÷ 16-bit register-register divide : 1250 ns • Instruction set suitable for high-speed operation — Sixty-five basic instructions — 8/16/32-bit move/arithmetic and logic instructions — Unsigned/signed multiply and divide instructions — Powerful bit-manipulation instructions • Two CPU operating modes — Normal mode : 64-kbyte address space (not available in the H8S/2214) — Advanced mode : 16-Mbyte address space
Bus controller	• Address space divided into 8 areas, with bus specifications settable independently for each area • Chip select output possible for each area • Choice of 8-bit or 16-bit access space for each area • 2-state or 3-state access space can be designated for each area • Number of program wait states can be set for each area • Burst ROM directly connectable • External bus release function
DMA controller (DMAC)	• Choice of short address mode or full address mode • Four channels in short address mode Two channels in full address mode • Transfer possible in repeat mode, block transfer mode, etc. • Can be activated by internal interrupt

Item	Specification
Data transfer controller (DTC)	• Can be activated by internal interrupt or software • Multiple transfers or multiple types of transfer possible for one activation source • Transfer possible in repeat mode, block transfer mode, etc. • Request can be sent to CPU for interrupt that activated DTC
16-bit timer-pulse unit (TPU)	• 3-channel 16-bit timer on-chip • Pulse I/O processing capability for up to 8 pins • Automatic 2-phase encoder count capability
Watchdog timer (WDT) × 1 channel	• Watchdog timer or interval timer selectable
Serial communication interface (SCI) × 3 channels (SCI0 to SCI2)	• Asynchronous mode or synchronous mode selectable • Multiprocessor communication function
D/A converter	• Resolution: 8 bits • Output: 1 channel
I/O ports	• 72 I/O pins, 9 input-only pins
Memory	• Flash memory or mask ROM • High-speed static RAM
Interrupt controller	• Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$) Eight external expansion interrupt pins (EXIRQ7 to EXIRQ0) • 31 internal interrupt sources • Eight priority levels settable
Power-down state	• Medium-speed mode • Sleep mode • Module stop mode • Software standby mode • Hardware standby mode

Product Name	ROM	RAM
H8S/2214	128 kbytes	12 kbytes

Item | **Specification**

Operating modes: Four MCU operating modes

Mode	CPU Operating Mode	Description	On-Chip ROM	External Data Bus: Initial Value	External Data Bus: Maximum Value
4	Advanced	On-chip ROM disabled expansion mode	Disabled	16 bits	16 bits
5		On-chip ROM disabled expansion mode	Disabled	8 bits	16 bits
6		On-chip ROM enabled expansion mode	Enabled	8 bits	16 bits
7		Single-chip mode	Enabled	—	

Clock pulse generator: Clock pulse generators

- System clock pulse generator: 2 to 16 MHz

Built-in duty correction circuit

Packages

- 100-pin plastic TQFP (TFP-100B)
- 100-pin plastic TQFP (TFP-100G)
- 112-pin plastic TFBGA (TBP-112)

Product lineup

Model Name: Mask ROM Version	Z-TAT™ Version	ROM/RAM (Bytes)	Packages
HD6432214	HD64F2214	128 k/12 k	TFP-100B, TFP-100G, TBP-112

1.2 Internal Block Diagrams

Figures 1-1 shows internal block diagram of the H8S/2214.

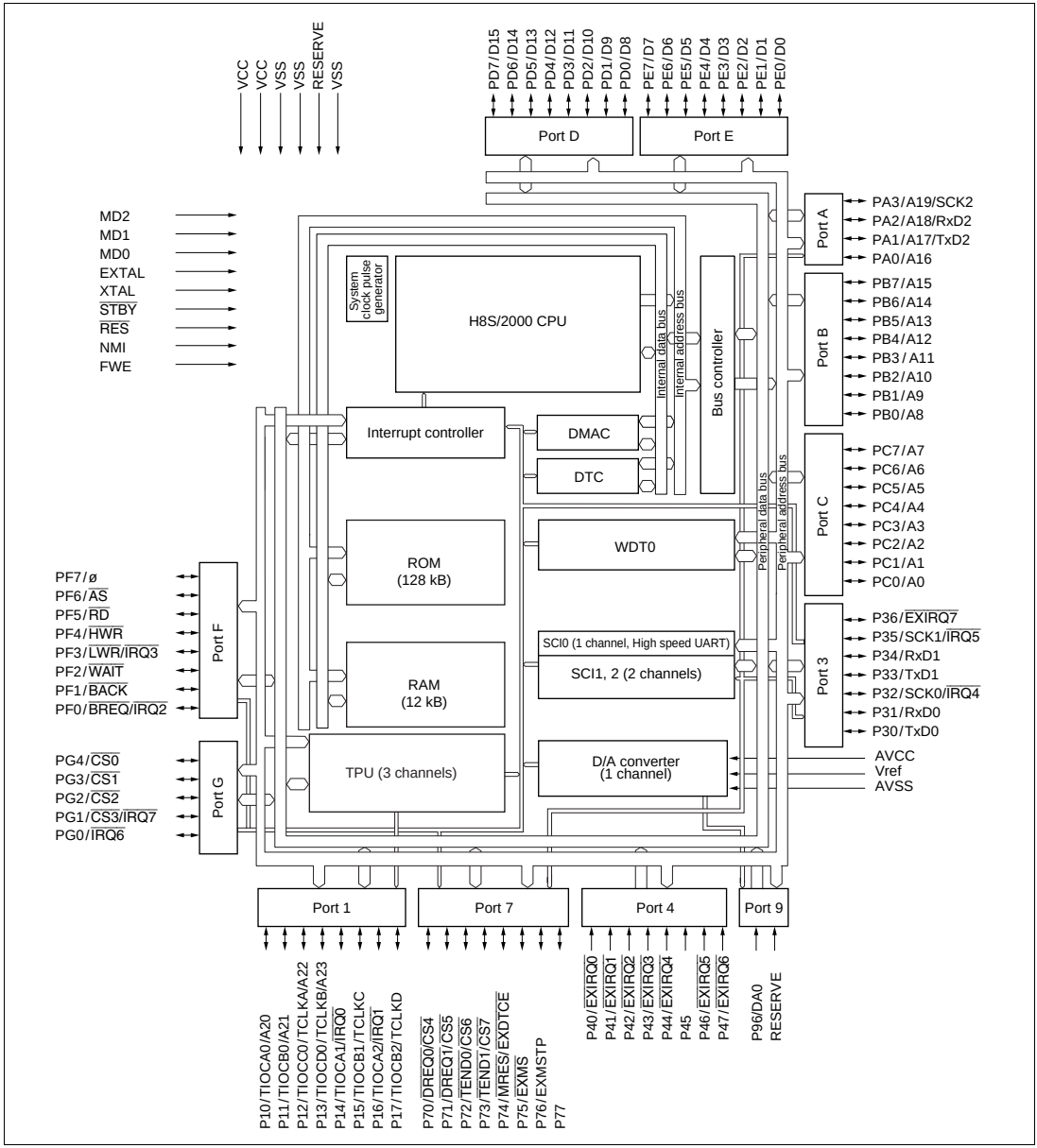


Figure 1-1 H8S/2214 Internal Block Diagram

1.3 Pin Description

1.3.1 Pin Arrangements

Figures 1-2 and 1-3 show the pin arrangements of the H8S/2214.

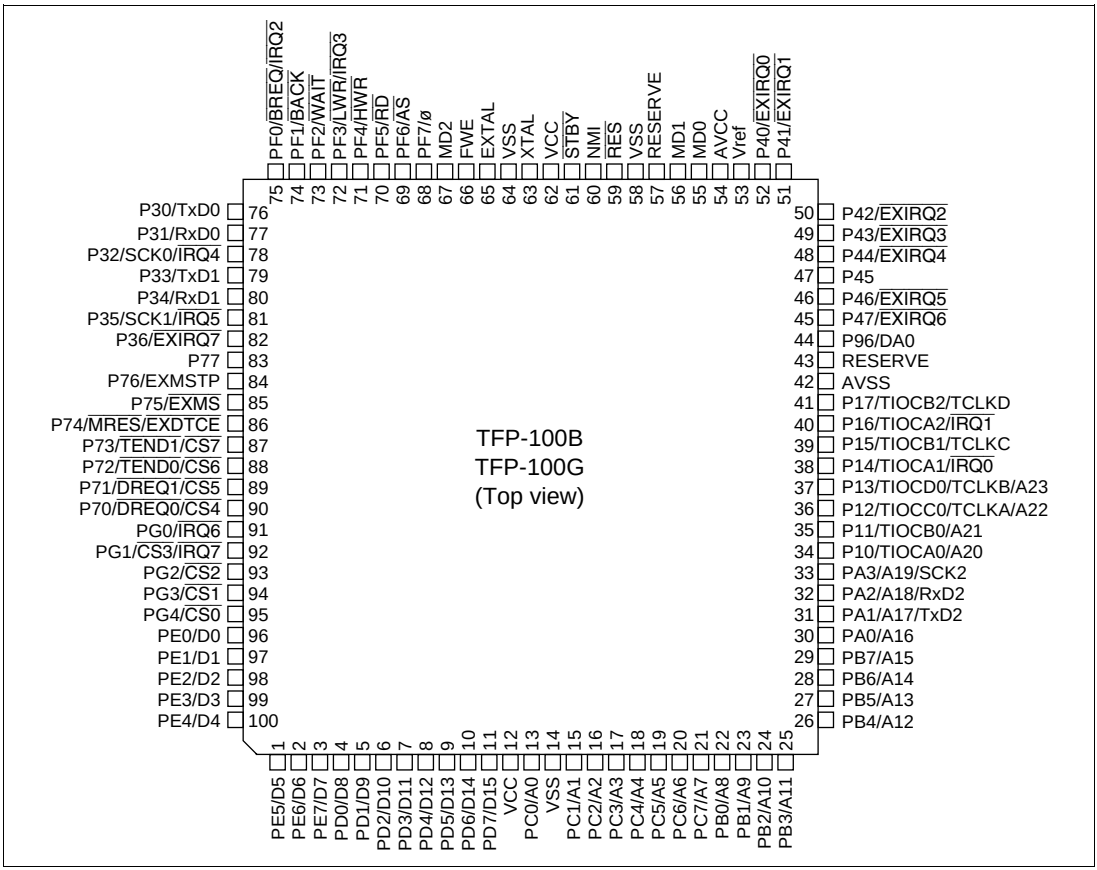


Figure 1-2 H8S/2214 Pin Arrangement (TFP-100B, TFP-100G: Top View)

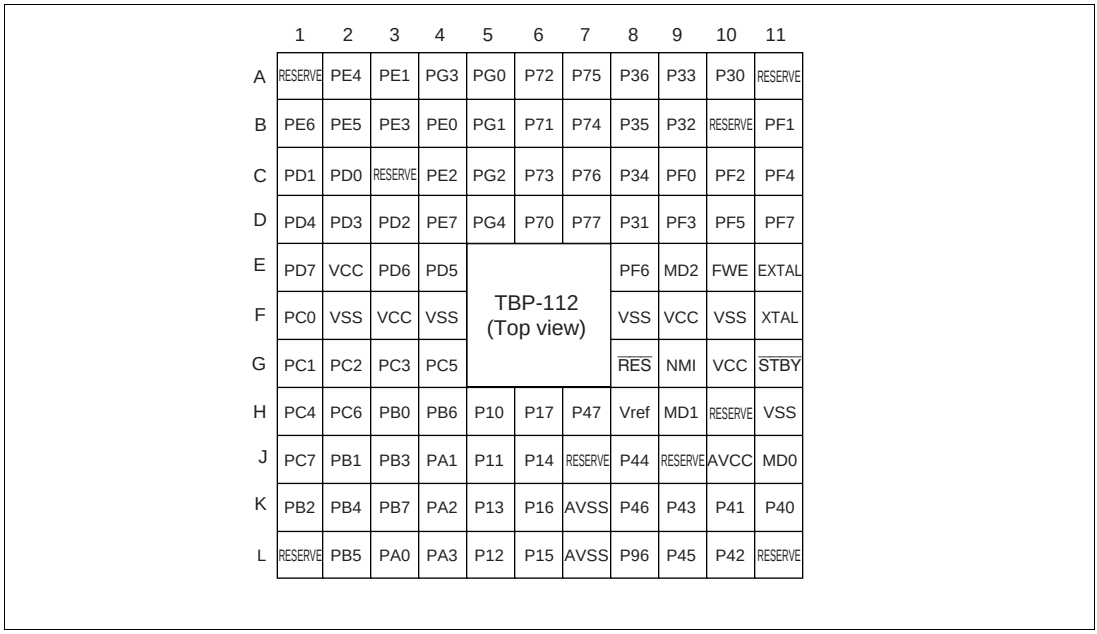


Figure 1-3 H8S/2214 Pin Arrangement (TBP-112: Top View)

1.3.2 Pin Functions in Each Operating Mode

Table 1-2 shows the pin functions of the H8S/2214 in each of the operating modes.

Table 1-2 Pin Functions in Each Operating Mode

Pin No.		Pin Name				
TFP-100B, TFP-100G	TBP-112	Mode 4	Mode 5	Mode 6	Mode 7	PROM Mode
1	B2	PE5/D5	PE5/D5	PE5/D5	PE5	NC
2	B1	PE6/D6	PE6/D6	PE6/D6	PE6	NC
3	D4	PE7/D7	PE7/D7	PE7/D7	PE7	NC
4	C2	D8	D8	D8	PD0	D0
5	C1	D9	D9	D9	PD1	D1
6	D3	D10	D10	D10	PD2	D2
7	D2	D11	D11	D11	PD3	D3
8	D1	D12	D12	D12	PD4	D4
9	E4	D13	D13	D13	PD5	D5
10	E3	D14	D14	D14	PD6	D6
11	E1	D15	D15	D15	PD7	D7
12	E2, F3	VCC	VCC	VCC	VCC	VCC
13	F1	A0	A0	PC0/A0	PC0	A0
14	F2, F4	VSS	VSS	VSS	VSS	VSS
15	G1	A1	A1	PC1/A1	PC1	A1
16	G2	A2	A2	PC2/A2	PC2	A2
17	G3	A3	A3	PC3/A3	PC3	A3
18	H1	A4	A4	PC4/A4	PC4	A4
19	G4	A5	A5	PC5/A5	PC5	A5
20	H2	A6	A6	PC6/A6	PC6	A6
21	J1	A7	A7	PC7/A7	PC7	A7
22	H3	PB0/A8	PB0/A8	PB0/A8	PB0	A8
23	J2	PB1/A9	PB1/A9	PB1/A9	PB1	$\overline{OE}$
24	K1	PB2/A10	PB2/A10	PB2/A10	PB2	A10
25	J3	PB3/A11	PB3/A11	PB3/A11	PB3	A11
26	K2	PB4/A12	PB4/A12	PB4/A12	PB4	A12
27	L2	PB5/A13	PB5/A13	PB5/A13	PB5	A13
28	H4	PB6/A14	PB6/A14	PB6/A14	PB6	A14

Pin No.		Pin Name				
TFP-100B, TFP-100G	TBP-112	Mode 4	Mode 5	Mode 6	Mode 7	PROM Mode
29	K3	PB7/A15	PB7/A15	PB7/A15	PB7	A15
30	L3	PA0/A16	PA0/A16	PA0/A16	PA0	A16
31	J4	PA1/A17/TxD2	PA1/A17/TxD2	PA1/A17/TxD2	PA1/TxD2	VCC
32	K4	PA2/A18/RxD2	PA2/A18/RxD2	PA2/A18/RxD2	PA2/RxD2	VCC
33	L4	PA3/A19/SCK2	PA3/A19/SCK2	PA3/A19/SCK2	PA3/SCK2	NC
34	H5	P10/TIOCA0/A20	P10/TIOCA0/A20	P10/TIOCA0/A20	P10/TIOCA0	NC
35	J5	P11/TIOCB0/A21	P11/TIOCB0/A21	P11/TIOCB0/A21	P11/TIOCB0	NC
36	L5	P12/TIOCC0/ TCLKA/A22	P12/TIOCC0/ TCLKA/A22	P12/TIOCC0/ TCLKA/A22	P12/TIOCC0/ TCLKA	NC
37	K5	P13/TIOCD0/ TCLKB/A23	P13/TIOCD0/ TCLKB/A23	P13/TIOCD0/ TCLKB/A23	P13/TIOCD0/ TCLKB	NC
38	J6	P14/TIOCA1/ $\overline{\text{IRQ0}}$	P14/TIOCA1/ $\overline{\text{IRQ0}}$	P14/TIOCA1/ $\overline{\text{IRQ0}}$	P14/TIOCA1/ $\overline{\text{IRQ0}}$	NC
39	L6	P15/TIOCB1/ TCLKC	P15/TIOCB1/ TCLKC	P15/TIOCB1/ TCLKC	P15/TIOCB1/ TCLKC	NC
40	K6	P16/TIOCA2/ $\overline{\text{IRQ1}}$	P16/TIOCA2/ $\overline{\text{IRQ1}}$	P16/TIOCA2/ $\overline{\text{IRQ1}}$	P16/TIOCA2/ $\overline{\text{IRQ1}}$	NC
41	H6	P17/TIOCB2/ TCLKD	P17/TIOCB2/ TCLKD	P17/TIOCB2/ TCLKD	P17/TIOCB2/ TCLKD	NC
42	K7, L7	AVSS	AVSS	AVSS	AVSS	VSS
43	J7	Reserve	Reserve	Reserve	Reserve	NC
44	L8	P96/DA0	P96/DA0	P96/DA0	P96/DA0	NC
45	H7	P47/$\overline{\text{EXIRQ6}}$	P47/$\overline{\text{EXIRQ6}}$	P47/$\overline{\text{EXIRQ6}}$	P47/$\overline{\text{EXIRQ6}}$	NC
46	K8	P46/$\overline{\text{EXIRQ5}}$	P46/$\overline{\text{EXIRQ5}}$	P46/$\overline{\text{EXIRQ5}}$	P46/$\overline{\text{EXIRQ5}}$	NC
47	L9	P45	P45	P45	P45	NC
48	J8	P44/$\overline{\text{EXIRQ4}}$	P44/$\overline{\text{EXIRQ4}}$	P44/$\overline{\text{EXIRQ4}}$	P44/$\overline{\text{EXIRQ4}}$	NC
49	K9	P43/$\overline{\text{EXIRQ3}}$	P43/$\overline{\text{EXIRQ3}}$	P43/$\overline{\text{EXIRQ3}}$	P43/$\overline{\text{EXIRQ3}}$	NC
50	L10	P42/$\overline{\text{EXIRQ2}}$	P42/$\overline{\text{EXIRQ2}}$	P42/$\overline{\text{EXIRQ2}}$	P42/$\overline{\text{EXIRQ2}}$	NC
51	K10	P41/$\overline{\text{EXIRQ1}}$	P41/$\overline{\text{EXIRQ1}}$	P41/$\overline{\text{EXIRQ1}}$	P41/$\overline{\text{EXIRQ1}}$	NC
52	K11	P40/$\overline{\text{EXIRQ0}}$	P40/$\overline{\text{EXIRQ0}}$	P40/$\overline{\text{EXIRQ0}}$	P40/$\overline{\text{EXIRQ0}}$	NC
53	H8	Vref	Vref	Vref	Vref	VCC
54	J10	AVCC	AVCC	AVCC	AVCC	VCC
55	J11	MD0	MD0	MD0	MD0	VSS

Pin No.		Pin Name				
TFP-100B, TFP-100G	TBP-112	Mode 4	Mode 5	Mode 6	Mode 7	PROM Mode
56	H9	MD1	MD1	MD1	MD1	VSS
57	H10	Reserve	Reserve	Reserve	Reserve	NC
58	H11	VSS	VSS	VSS	VSS	NC
59	G8	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	VPP
60	G9	NMI	NMI	NMI	NMI	A9
61	G11	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	VSS
62	F9, G10	VCC	VCC	VCC	VCC	VCC
63	F11	XTAL	XTAL	XTAL	XTAL	NC
64	F8, F10	VSS	VSS	VSS	VSS	VSS
65	E11	EXTAL	EXTAL	EXTAL	EXTAL	NC
66	E10	FWE	FWE	FWE	FWE	FWE
67	E9	MD2	MD2	MD2	MD2	VSS
68	D11	PF7/ø	PF7/ø	PF7/ø	PF7/ø	NC
69	E8	$\overline{\text{AS}}$	$\overline{\text{AS}}$	$\overline{\text{AS}}$	PF6	NC
70	D10	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PF5	NC
71	C11	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	PF4	NC
72	D9	PF3/$\overline{\text{LWR}}$/$\overline{\text{IRQ3}}$	PF3/$\overline{\text{LWR}}$/$\overline{\text{IRQ3}}$	PF3/$\overline{\text{LWR}}$/$\overline{\text{IRQ3}}$	PF3/$\overline{\text{IRQ3}}$	NC
73	C10	PF2/$\overline{\text{WAIT}}$	PF2/$\overline{\text{WAIT}}$	PF2/$\overline{\text{WAIT}}$	PF2	$\overline{\text{CE}}$
74	B11	PF1/$\overline{\text{BACK}}$	PF1/$\overline{\text{BACK}}$	PF1/$\overline{\text{BACK}}$	PF1	$\overline{\text{PGM}}$
75	C9	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{BREQ}}$/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{IRQ2}}$	NC
76	A10	P30/TxD0	P30/TxD0	P30/TxD0	P30/TxD0	NC
77	D8	P31/RxD1	P31/RxD1	P31/RxD1	P31/RxD1	NC
78	B9	P32/SCK0/$\overline{\text{IRQ4}}$	P32/SCK0/$\overline{\text{IRQ4}}$	P32/SCK0/$\overline{\text{IRQ4}}$	P32/SCK0/$\overline{\text{IRQ4}}$	NC
79	A9	P33/TxD1	P33/TxD1	P33/TxD1	P33/TxD1	NC
80	C8	P34/RxD1	P34/RxD1	P34/RxD1	P34/RxD1	NC
81	B8	P35/SCK1/$\overline{\text{IRQ5}}$	P35/SCK1/$\overline{\text{IRQ5}}$	P35/SCK1/$\overline{\text{IRQ5}}$	P35/SCK1/$\overline{\text{IRQ5}}$	NC
82	A8	P36/$\overline{\text{EXIRQ7}}$	P36/$\overline{\text{EXIRQ7}}$	P36/$\overline{\text{EXIRQ7}}$	P36/$\overline{\text{EXIRQ7}}$	NC
83	D7	P77	P77	P77	P77	NC
84	C7	P76/EXMSTP	P76/EXMSTP	P76/EXMSTP	P76/EXMSTP	NC
85	A7	P75/$\overline{\text{EXMS}}$	P75/$\overline{\text{EXMS}}$	P75/$\overline{\text{EXMS}}$	P75/$\overline{\text{EXMS}}$	NC

Pin No.		Pin Name				
TFP-100B, TFP-100G	**TBP-112**	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**	**PROM Mode**
86	B7	P74/$\overline{\text{MRES}}$/ $\overline{\text{EXDTCE}}$	P74/$\overline{\text{MRES}}$/ $\overline{\text{EXDTCE}}$	P74/$\overline{\text{MRES}}$/ $\overline{\text{EXDTCE}}$	P74/$\overline{\text{MRES}}$/ $\overline{\text{EXDTCE}}$	NC
87	C6	P73/$\overline{\text{TEND1}}$/$\overline{\text{CS7}}$	P73/$\overline{\text{TEND1}}$/$\overline{\text{CS7}}$	P73/$\overline{\text{TEND1}}$/$\overline{\text{CS7}}$	P73/$\overline{\text{TEND1}}$	NC
88	A6	P72/$\overline{\text{TEND0}}$/$\overline{\text{CS6}}$	P72/$\overline{\text{TEND0}}$/$\overline{\text{CS6}}$	P72/$\overline{\text{TEND0}}$/$\overline{\text{CS6}}$	P72/$\overline{\text{TEND0}}$	NC
89	B6	P71/$\overline{\text{DREQ1}}$/$\overline{\text{CS5}}$	P71/$\overline{\text{DREQ1}}$/$\overline{\text{CS5}}$	P71/$\overline{\text{DREQ1}}$/$\overline{\text{CS5}}$	P71/$\overline{\text{DREQ1}}$	NC
90	D6	P70/$\overline{\text{DREQ0}}$/$\overline{\text{CS4}}$	P70/$\overline{\text{DREQ0}}$/$\overline{\text{CS4}}$	P70/$\overline{\text{DREQ0}}$/$\overline{\text{CS4}}$	P70/$\overline{\text{DREQ0}}$	NC
91	A5	PG0/$\overline{\text{IRQ6}}$	PG0/$\overline{\text{IRQ6}}$	PG0/$\overline{\text{IRQ6}}$	PG0/$\overline{\text{IRQ6}}$	NC
92	B5	PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$	PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$	PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$	PG1/$\overline{\text{IRQ7}}$	NC
93	C5	PG2/$\overline{\text{CS2}}$	PG2/$\overline{\text{CS2}}$	PG2/$\overline{\text{CS2}}$	PG2	NC
94	A4	PG3/$\overline{\text{CS1}}$	PG3/$\overline{\text{CS1}}$	PG3/$\overline{\text{CS1}}$	PG3	NC
95	D5	PG4/$\overline{\text{CS0}}$	PG4/$\overline{\text{CS0}}$	PG4/$\overline{\text{CS0}}$	PG4	NC
96	B4	PE0/D0	PE0/D0	PE0/D0	PE0	NC
97	A3	PE1/D1	PE1/D1	PE1/D1	PE1	NC
98	C4	PE2/D2	PE2/D2	PE2/D2	PE2	NC
99	B3	PE3/D3	PE3/D3	PE3/D3	PE3	NC
100	A2	PE4/D4	PE4/D4	PE4/D4	PE4	VSS
—	A1, A11, B10, C3, J9, L1, L11	Reserve	Reserve	Reserve	Reserve	Reserve

1.3.3 Pin Functions

Table 1-3 outlines the pin functions of the H8S/2214.

Table 1-3 Pin Functions

Type	Symbol	I/O	Name and Function
Power	VCC	Input	Power supply: For connection to the power supply. All V_{CC} pins should be connected to the system power supply.
	VSS	Input	Ground: For connection to ground (0 V). All V_{SS} pins should be connected to the system power supply (0 V).
Clock	XTAL	Input	Crystal: Connects to a crystal oscillator. See section 16, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator and external clock input.
	EXTAL	Input	External clock: Connects to a crystal oscillator. The EXTAL pin can also input an external clock. See section 16, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator and external clock input.
	ø	Output	System clock: Supplies the system clock to an external device.
Operating mode control	MD2 to MD0	Input	Mode pins: These pins set the operating mode. The relation between the settings of pins MD2 to MD0 and the operating mode is shown below. These pins should not be changed while the H8S/2214 is operating. Except when the mode is changed, the mode pins (MD2 to MD0) must be pulled down or pulled up to a fixed level until powering off.
System control	$\overline{\text{RES}}$	Input	Reset input: When this pin is driven low, the chip enters the power-on reset state.
	$\overline{\text{MRES}}$	Input	Manual reset: When this pin is driven low, the chip enters the manual reset state.

MD2	MD1	MD0	Operating Mode
0	0	0	—
		1	—
	1	0	—
		1	—
1	0	0	Mode 4
		1	Mode 5
	1	0	Mode 6
		1	Mode 7

Type	Symbol	I/O	Name and Function
System control	$\overline{\text{STBY}}$	Input	Standby: When this pin is driven low, a transition is made to hardware standby mode.
	$\overline{\text{BREQ}}$	Input	Bus request: Used by an external bus master to issue a bus request to the H8S/2214.
	$\overline{\text{BACK}}$	Output	Bus request acknowledge: Indicates that the bus has been released to an external bus master.
	FWE	Input	Flash write enable Enables/disables flash memory programming.
Interrupts	NMI	Input	Nonmaskable interrupt: Requests a nonmaskable interrupt. When this pin is not used, it should be fixed high.
	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	Input	Interrupt request 7 to 0: These pins request a maskable interrupt.
Address bus	A23 to A0	Output	Address bus: These pins output an address.
Data bus	D15 to D0	I/O	Data bus: These pins constitute a bidirectional data bus.
Bus control	$\overline{\text{CS7}}$ to $\overline{\text{CS0}}$	Output	Chip select: Signals for selecting areas 7 to 0.
	$\overline{\text{AS}}$	Output	Address strobe: When this pin is low, it indicates that address output on the address bus is enabled.
	$\overline{\text{RD}}$	Output	Read: When this pin is low, it indicates that the external address space can be read.
	$\overline{\text{HWR}}$	Output	High write: A strobe signal that writes to external space and indicates that the upper half (D15 to D8) of the data bus is enabled.
	$\overline{\text{LWR}}$	Output	Low write: A strobe signal that writes to external space and indicates that the lower half (D7 to D0) of the data bus is enabled.
	$\overline{\text{WAIT}}$	Input	Wait: Requests insertion of a wait state in the bus cycle when accessing external 3-state address space.
External expansion	$\overline{\text{EXIRQ7}}$ to $\overline{\text{EXIRQ0}}$	Input	External expansion interrupt request 7 to 0: Input pins for interrupt requests from external modules.
	$\overline{\text{EXMS}}$	Output	External expansion module select: Select signal for external modules.
	$\overline{\text{EXDTC}}$	Output	External expansion DTC transfer end: DTC data transfer end signal for $\overline{\text{EXIRQ7}}$ to $\overline{\text{EXIRQ0}}$ input.
	$\overline{\text{EXMSTP}}$	Output	External expansion module stop: Module stop signal for external modules.

Type	Symbol	I/O	Name and Function
DMA controller (DMAC)	$\overline{\text{DREQ1}}$, $\overline{\text{DREQ0}}$	Input	DMA request 1 and 0: These pins request DMAC activation.
	$\overline{\text{TEND1}}$, $\overline{\text{TEND0}}$	Output	DMA transfer end 1 and 0: These pins indicate the end of DMAC data transfer.
16-bit timer-pulse unit (TPU)	TCLKD to TCLKA	Input	Clock input D to A: These pins input an external clock.
	TIOCA0, TIOCB0, TIOCC0, TIOCD0	I/O	Input capture/output compare match A0 to D0: The TGR0A to TGR0D input capture input or output compare output, or PWM output pins.
	TIOCA1, TIOCB1	I/O	Input capture/output compare match A1 and B1: The TGR1A and TGR1B input capture input or output compare output, or PWM output pins.
	TIOCA2, TIOCB2	I/O	Input capture/output compare match A2 and B2: The TGR2A and TGR2B input capture input or output compare output, or PWM output pins.
Serial communication interface (SCI)	TxD2, TxD1, TxD0	Output	Transmit data: Data output pins.
	RxD2, RxD1, RxD0	Input	Receive data: Data input pins.
	SCK2, SCK1 SCK0	I/O	Serial clock: Clock I/O pins.
D/A converter	DA0	Output	Analog output: D/A converter analog output pins.
	AVCC	Input	This is the power supply pin for the D/A converter. When the D/A converter is not used, this pin should be connected to the system power supply (VCC).
	AVSS	Input	This is the ground pin for the D/A converter. This pin should be connected to the system power supply (0 V).
	Vref	Input	This is the reference voltage input pin for the D/A converter. When the D/A converter is not used, this pin should be connected to the system power supply (VCC).

Type	Symbol	I/O	Name and Function
I/O ports	P17 to P10	I/O	Port 1: An 8-bit I/O port. Input or output can be designated for each bit by means of the port 1 data direction register (P1DDR).
	P36 to P30	I/O	Port 3: A 7-bit I/O port. Input or output can be designated for each bit by means of the port 3 data direction register (P3DDR).
	P47 to P40	Input	Port 4: An 8-bit input port.
	P77 to P70	I/O	Port 7: An 8-bit I/O port. Input or output can be designated for each bit by means of the port 7 data direction register (P7DDR).
	P96	Input	Port 9: A 1-bit input port.
	PA3 to PA0	I/O	Port A: A 4-bit I/O port. Input or output can be designated for each bit by means of the port A data direction register (PADDR).
	PB7 to PB0	I/O	Port B: An 8-bit I/O port. Input or output can be designated for each bit by means of the port B data direction register (PBDDR).
	PC7 to PC0	I/O	Port C: An 8-bit I/O port. Input or output can be designated for each bit by means of the port C data direction register (PCDDR).
	PD7 to PD0	I/O	Port D: An 8-bit I/O port. Input or output can be designated for each bit by means of the port D data direction register (PDDDR).
	PE7 to PE0	I/O	Port E: An 8-bit I/O port. Input or output can be designated for each bit by means of the port E data direction register (PEDDR).
	PF7 to PF0	I/O	Port F: An 8-bit I/O port. Input or output can be designated for each bit by means of the port F data direction register (PFDDR).
	PG4 to PG0	I/O	Port G: A 5-bit I/O port. Input or output can be designated for each bit by means of the port G data direction register (PGDDR).

Section 2 CPU

2.1 Overview

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 (architecturally 4-Gbyte) linear address space, and is ideal for realtime control.

2.1.1 Features

The H8S/2000 CPU has the following features.

- Upward-compatible with H8/300 and H8/300H CPUs
  - Can execute H8/300 and H8/300H 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 (4 Gbytes architecturally)

- High-speed operation
  - — All frequently-used instructions execute in one or two states
  - — Maximum clock rate : 16 MHz
  - — 8/16/32-bit register-register add/subtract : 62.5 ns
  - — 8 × 8-bit register-register multiply : 750 ns
  - — 16 ÷ 8-bit register-register divide : 750 ns
  - — 16 × 16-bit register-register multiply : 1250 ns
  - — 32 ÷ 16-bit register-register divide : 1250 ns

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

  Note: * Not available in the H8S/2214.

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

2.1.2 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.

- Number of execution states

  The number of exection states of the MULXU and MULXS instructions.

		Internal Operation	
Instruction	**Mnemonic**	**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

There are also differences in the address space, CCR and EXR register functions, power-down state, etc., depending on the product.

2.1.3 Differences from H8/300 CPU

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

- More general registers and control registers
  - Eight 16-bit expanded registers, plus 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.

  Note: * Not available in the H8S/2214.
- 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 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 execute twice as fast.

2.1.4 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 and two 32-bit control registers have been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift 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 execute twice as fast.

2.2 CPU Operating Modes

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 total address space (architecturally a maximum 16-Mbyte program area and a maximum of 4 Gbytes for program and data areas combined). The mode is selected by the mode pins of the microcontroller.

Note: * Not available in the H8S/2214.

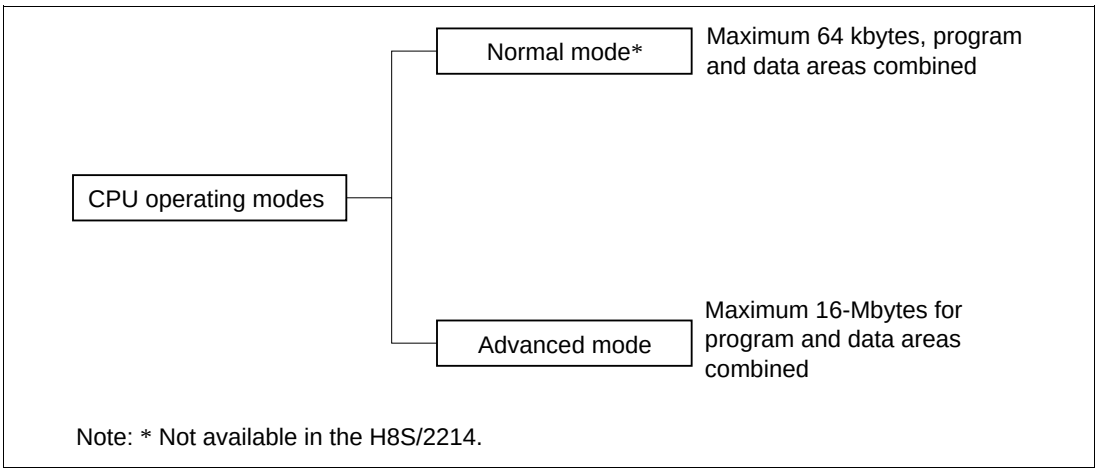


Figure 2-1 CPU Operating Modes

(1) Normal Mode (not available in the H8S/2214)

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

Address Space: A maximum address space of 64 kbytes can be accessed.

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 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 the general register is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, 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 configuration of the exception vector table in normal mode is shown in figure 2-2. For details of the exception vector table, see section 4, Exception Handling.

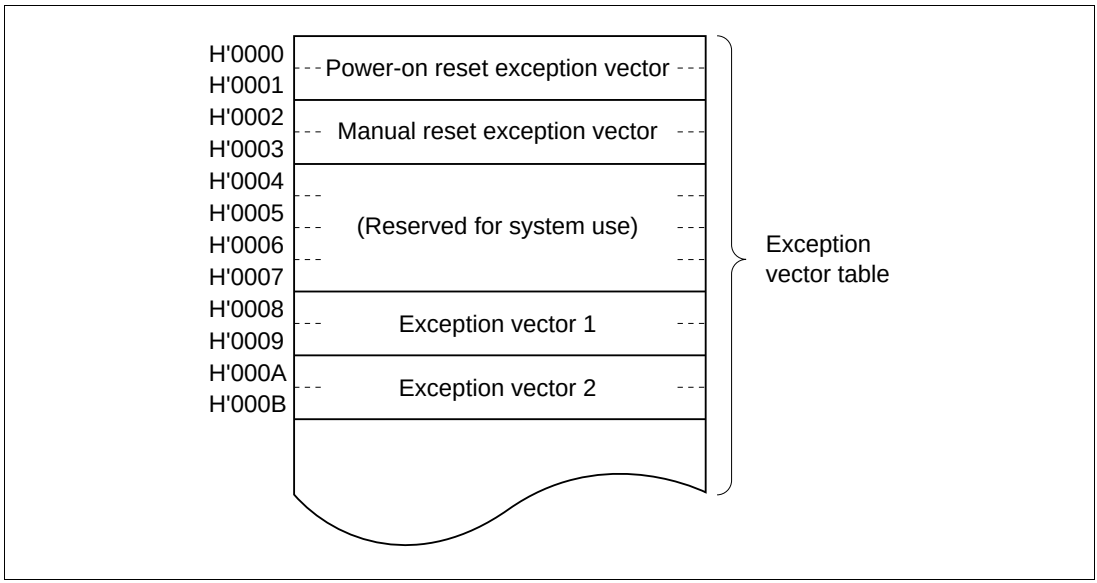
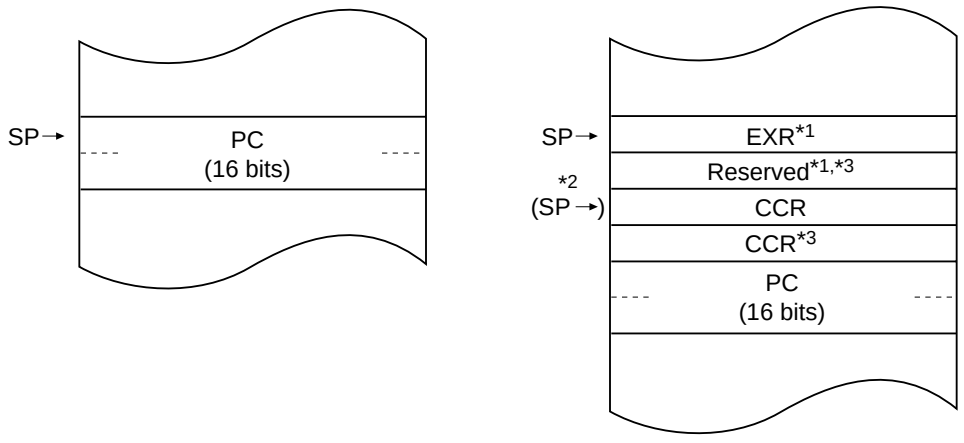


Figure 2-2 Exception Vector Table (Normal Mode)

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 2-3. When EXR is invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.



(a) Subroutine Branch **(b) Exception Handling**

Notes: *1 When EXR is not used it is not stored on the stack.
*2 SP when EXR is not used.
*3 Ignored when returning.

Figure 2-3 Stack Structure in Normal Mode

(2) Advanced Mode

Address Space: Linear access is provided to a 16-Mbyte maximum address space (architecturally a maximum 16-Mbyte program area and a maximum 4-Gbyte data area, with a maximum of 4 Gbytes for program and data areas combined).

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 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 units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2-4). For details of the exception vector table, see section 4, Exception Handling.

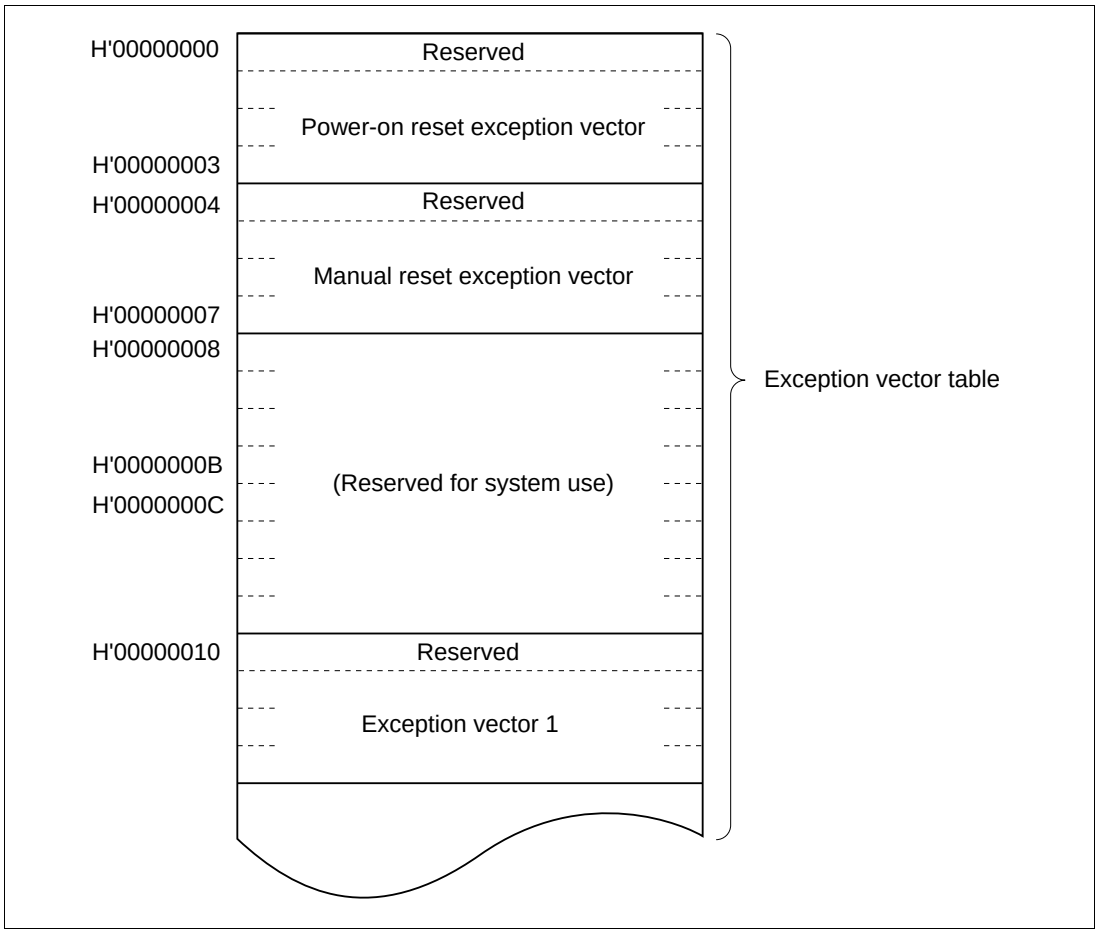
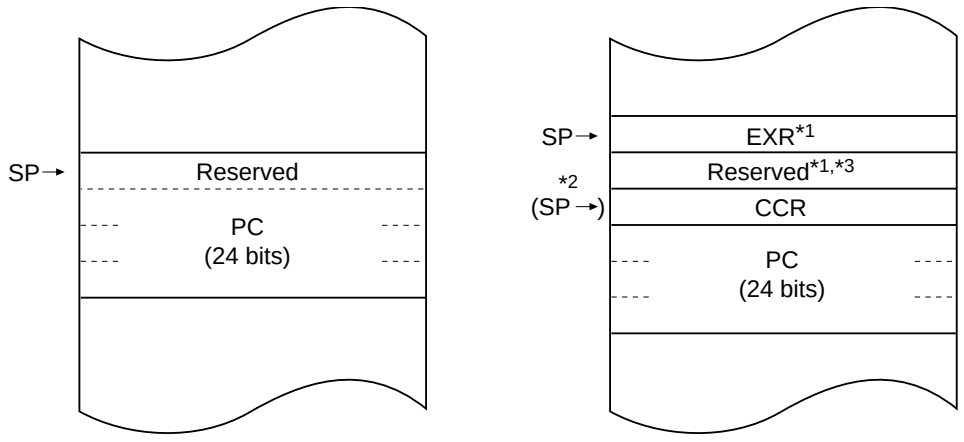


Figure 2-4 Exception Vector Table (Advanced Mode)

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 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 first part of this range is also 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 2-5. When EXR is invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.



(a) Subroutine Branch **(b) Exception Handling**

Notes: *1 When EXR is not used it is not stored on the stack.
*2 SP when EXR is not used.
*3 Ignored when returning.

Figure 2-5 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2-6 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.

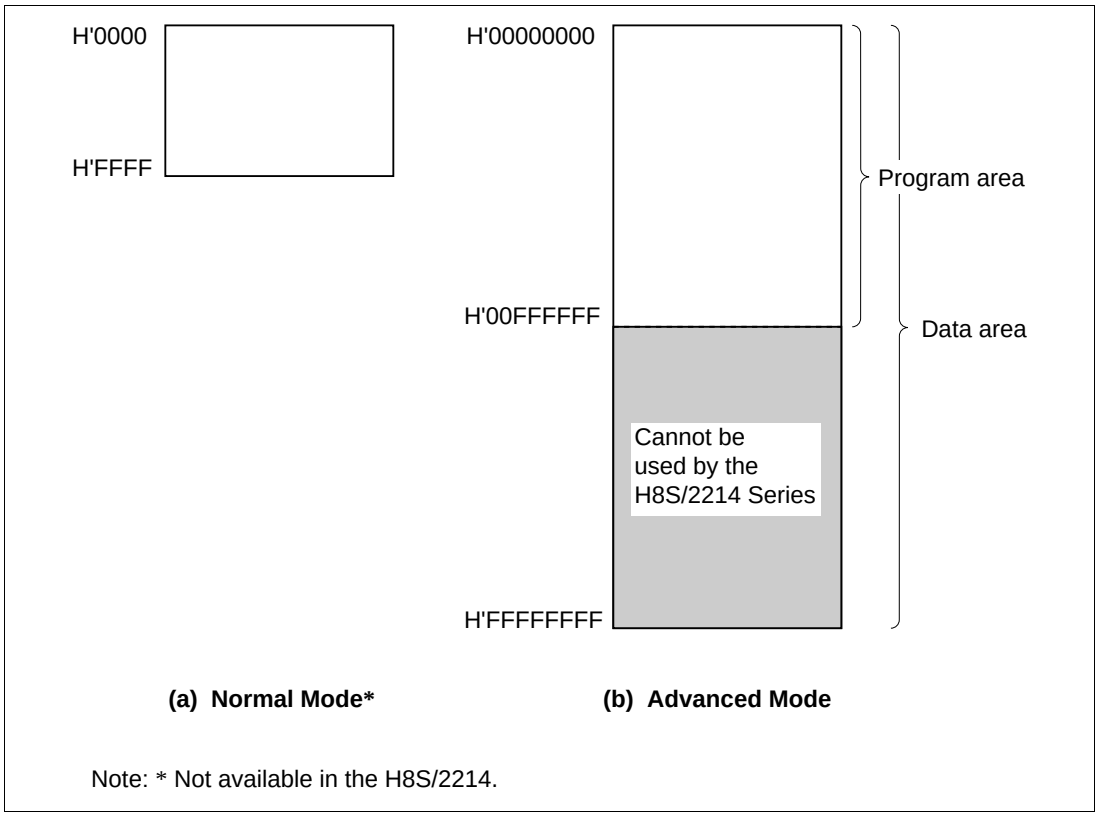


Figure 2-6 Memory Map

2.4 Register Configuration

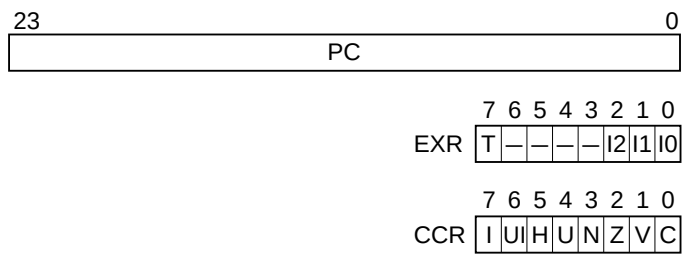
2.4.1 Overview

The CPU has the internal registers shown in figure 2-7. There are two types of registers: general registers and control registers.

General Registers (Rn) and Extended Registers (En)

	15 — 07	07 — 0	
ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers (CR)



Legend

SP:	Stack pointer	H:	Half-carry flag
PC:	Program counter	U:	User bit
EXR:	Extended control register	N:	Negative flag
T:	Trace bit	Z:	Zero flag
I2 to I0:	Interrupt mask bits	V:	Overflow flag
CCR:	Condition-code register	C:	Carry flag
I:	Interrupt mask bit		
UI:	User bit or interrupt mask bit*		

Note: * In the H8S/2214, this bit cannot be used as an interrupt mask.

Figure 2-7 CPU Registers

2.4.2 General Registers

The 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. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide 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.

The R registers divide 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.

Figure 2-8 illustrates the usage of the general registers. The usage of each register can be selected independently.

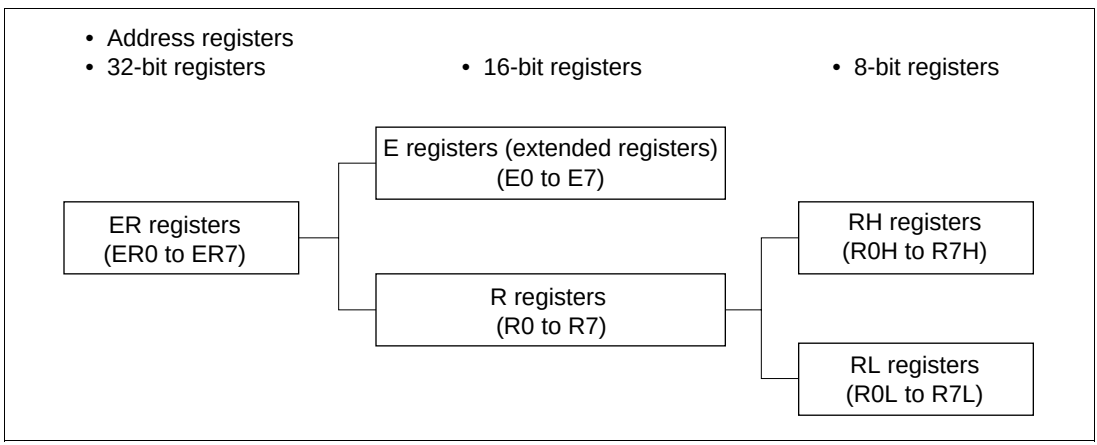


Figure 2-8 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2-9 shows the stack.

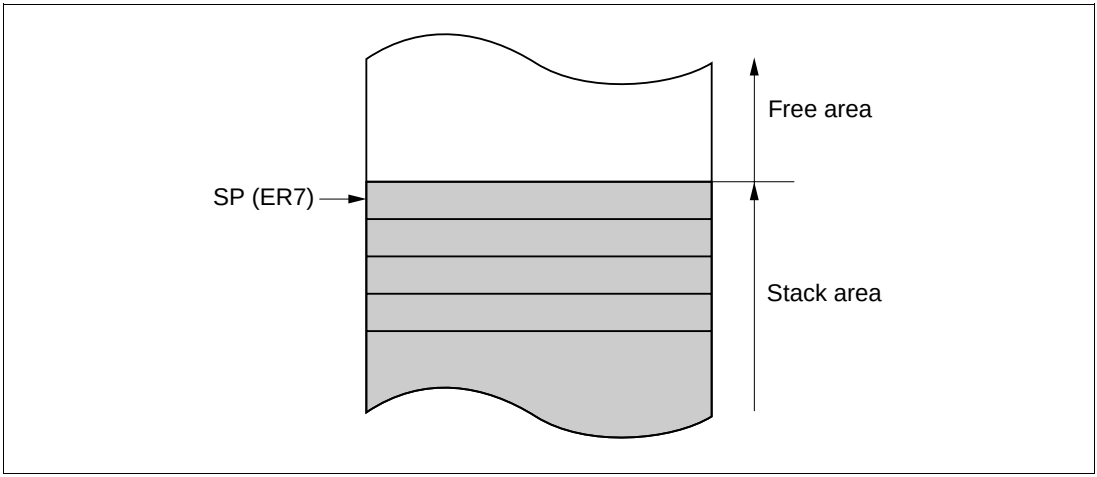


Figure 2-9 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC), 8-bit extended control register (EXR), and 8-bit condition-code register (CCR).

(1) 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, the least significant PC bit is regarded as 0.)

(2) Extended Control Register (EXR): This 8-bit register contains the trace bit (T) and interrupt mask bit.

Bit 7—Trace Bit (T): Selects trace mode. When this bit is cleared to 0, instructions are executed in sequence. When this bit is set to 1, a trace exception is generated each time an instruction is executed.

Bits 6 to 3—Reserved: These bits are reserved. They are always read as 1.

Bits 2 to 0—Interrupt Mask Bits (I2 to I0): These bits designate the interrupt mask level (0 to 7). For details, refer to section 5, Interrupt Controller.

Operations can be performed on the EXR bits by the LDC, STC, ANDC, ORC, and XORC instructions. All interrupts, including NMI, are disabled for three states after one of these instructions is executed, except for STC.

(3) Condition-Code Register (CCR): 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.

Bit 7—Interrupt Mask Bit (I): 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 by hardware at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. With the H8S/2214, this bit cannot be used as an interrupt mask bit.

Bit 5—Half-Carry Flag (H): 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.

Bit 4—User Bit (U): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.

Bit 3—Negative Flag (N): Stores the value of the most significant bit (sign bit) of data.

Bit 2—Zero Flag (Z): Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.

Bit 1—Overflow Flag (V): Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.

Bit 0—Carry Flag (C): Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by:

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions, to indicate a carry

The carry flag is also used as a bit accumulator by bit manipulation instructions.

Some instructions leave some or all of the flag bits unchanged. For the action of each instruction on the flag bits, refer to Appendix A.1, List of Instructions.

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.

2.4.4 Initial Register Values

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

2.5 Data Formats

The 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, …, 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

Figures 2-10 and 2-11 show the data formats in general registers.

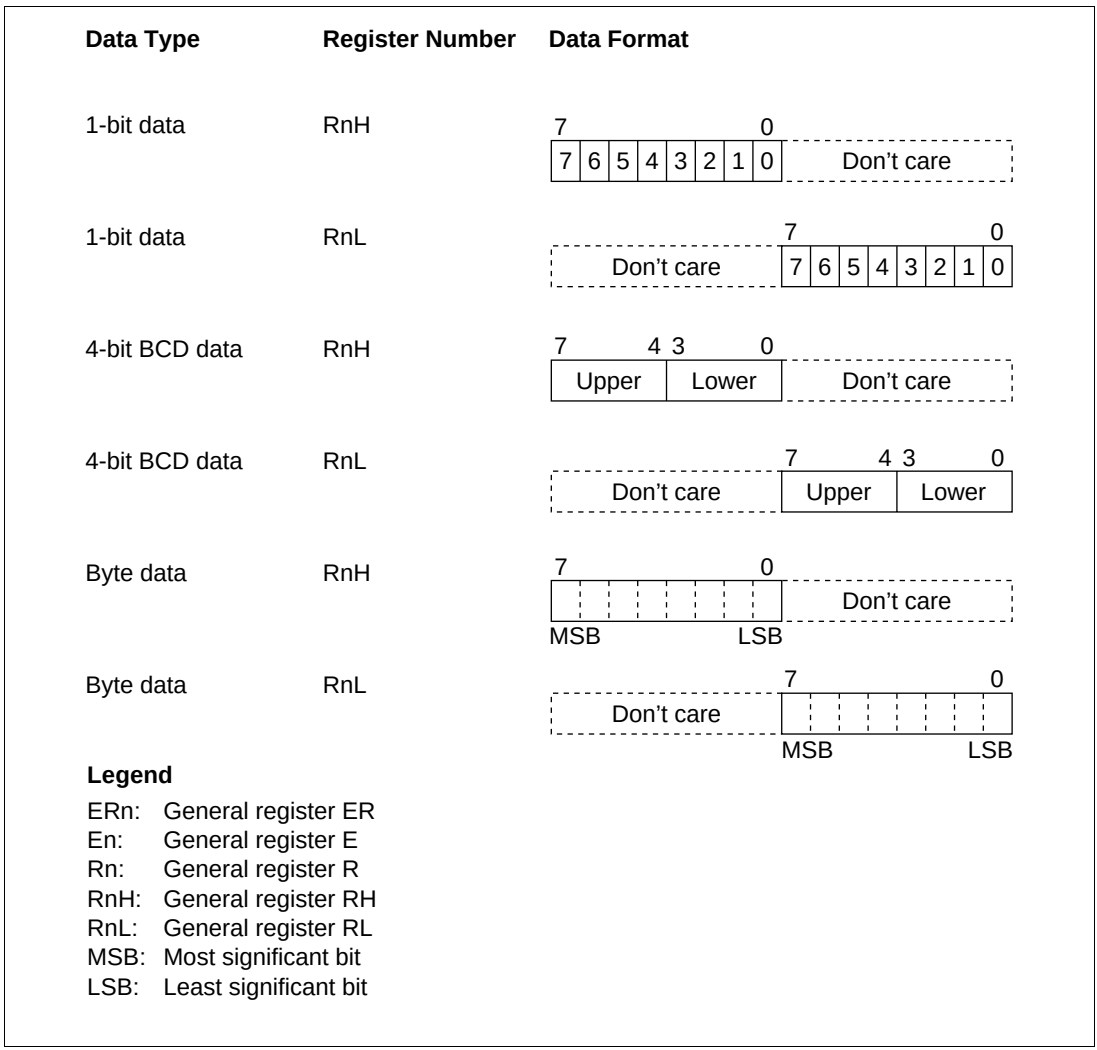
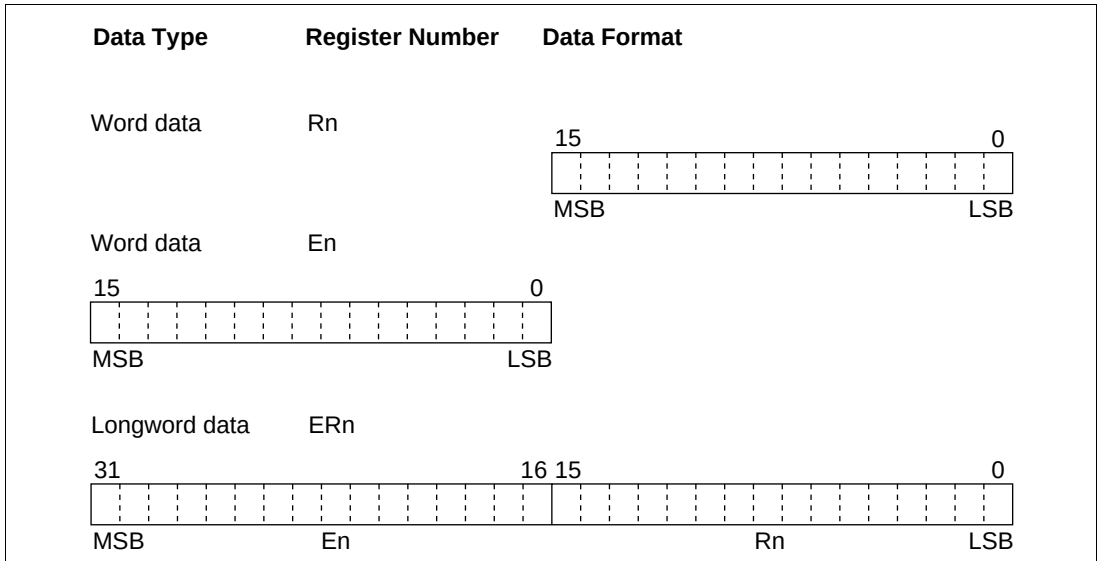


Figure 2-10 General Register Data Formats (1)



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 2-11 General Register Data Formats (2)

2.5.2 Memory Data Formats

Figure 2-12 shows the data formats in memory. The 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.

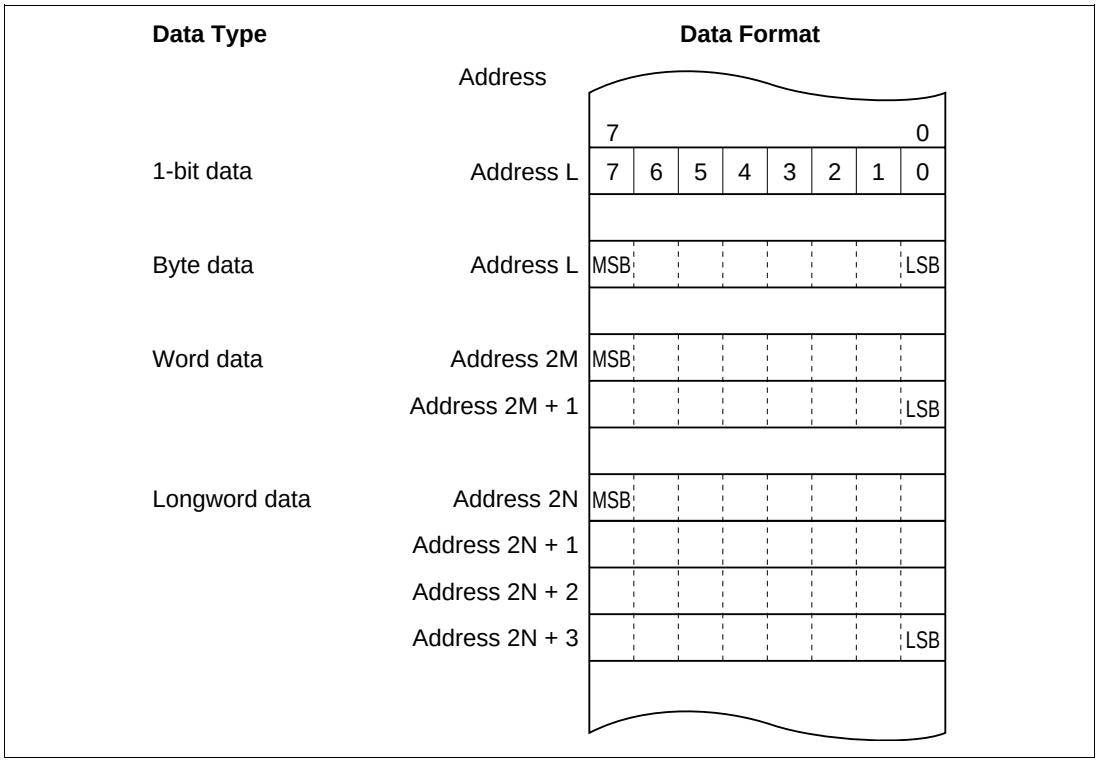


Figure 2-12 Memory Data Formats

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

2.6 Instruction Set

2.6.1 Overview

The H8S/2000 CPU has 65 types of instructions. The instructions are classified by function in table 2-1.

Table 2-1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	BWL	5
	POP*1, PUSH*1	WL	
	LDM, STM	L	
	MOVFPE, MOVTPE*3	B	
Arithmetic operations	ADD, SUB, CMP, NEG	BWL	19
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	BWL	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	BW	
	EXTU, EXTS	WL	
	TAS*4	B	
Logic operations	AND, OR, XOR, NOT	BWL	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	BWL	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc*2, 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 the H8S/2214.

*4 This instruction should be used with the ER0, ER1, ER4, or ER5 general register only.

2.6.2 Instructions and Addressing Modes

Table 2-2 indicates the combinations of instructions and addressing modes that the H8S/2600 CPU can use.

Table 2-2 Combinations of Instructions and Addressing Modes

Function	Instruction	Addressing Modes													
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:32,ERn)	@–ERn/@ERn+	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC)	@@aa:8	—
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	—	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	—	WL
	LDM, STM	—	—	—	—	—	—	—	—	—	—	—	—	—	L
	MOVFPE*1, MOVTPE*1	—	—	—	—	—	—	—	B	—	—	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—	—	—
	MULXU, DIVXU	—	BW	—	—	—	—	—	—	—	—	—	—	—	—
	MULXS, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—	—	—
	TAS*2	—	—	B	—	—	—	—	—	—	—	—	—	—	—

Notes: *1 Cannot be used in the H8S/2214.
*2 This instruction should be used with the ER0, ER1, ER4, or ER5 general register only.

Function	Instruction	Addressing Modes													
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:32,ERn)	@–ERn/@ERn+	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC)	@@aa:8	—
Logic operations	AND, OR, XOR	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
Shift		—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
Bit manipulation		—	B	B	—	—	—	B	B	—	B	—	—	—	—
Branch	Bcc, BSR	—	—	—	—	—	—	—	—	—	—	○	○	—	—
	JMP, JSR	—	—	—	—	—	—	—	—	○	—	—	—	○	—
	RTS	—	—	—	—	—	—	—	—	—	—	—	—	—	○
System control	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	RTE	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	LDC	B	B	W	W	W	W	—	W	—	W	—	—	—	—
	STC	—	B	W	W	W	W	—	W	—	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—	—	—	○
Block data transfer		—	—	—	—	—	—	—	—	—	—	—	—	—	BW

Legend
B: Byte
W: Word
L: Longword

2.6.3 Table of Instructions Classified by Function

Table 2-3 summarizes the instructions in each functional category. The notation used in table 2-3 is defined below.

Operation Notation

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).

Table 2-3 Instructions Classified by Function

Type	Instruction	Size*[1]	Function
Data transfer	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.
	MOVFPE	B	Cannot be used in the H8S/2214.
	MOVTPE	B	Cannot be used in the H8S/2214.
	POP	W/L	@SP+ → Rn Pops a 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 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: *1 Size refers to the operand size.
B: Byte
W: Word
L: Longword

Type	Instruction	Size*[1]	Function
Arithmetic operations	ADD SUB	B/W/L	Rd ± Rs → Rd, Rd ± #IMM → Rd Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register. (Immediate byte data cannot be subtracted from byte data in a general register. Use the SUBX or ADD instruction.)
	ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry or borrow on byte data in two general registers, or on immediate data and data in a general register.
	INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
	ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → 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 → Rd Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data.
	MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
	MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
	DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned 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.

Note: *1 Size refers to the operand size.
B: Byte
W: Word
L: Longword

Type	Instruction	Size*[1]	Function
Arithmetic operations	DIVXS	B/W	Rd ÷ Rs → Rd 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 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	B	@ERd – 0, 1 → (<bit 7> of @ERd)*[2] 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 This instruction should be used with the ER0, ER1, ER4, or ER5 general register only.

Type	Instruction	Size*[1]	Function
Logic operations	AND	B/W/L	Rd ∧ Rs → Rd, Rd ∧ #IMM → Rd Performs a logical AND operation on a general register and another general register or immediate data.
	OR	B/W/L	Rd ∨ Rs → Rd, Rd ∨ #IMM → Rd Performs a logical OR operation on a general register and another general register or immediate data.
	XOR	B/W/L	Rd ⊕ Rs → Rd, Rd ⊕ #IMM → Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data.
	NOT	B/W/L	¬ (Rd) → (Rd) Takes the one's complement of general register contents.
Shift operations	SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register contents. 1-bit or 2-bit shift is possible.
	SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register contents. 1-bit or 2-bit shift is possible.
	ROTL ROTR	B/W/L	Rd (rotate) → Rd Rotates general register contents. 1-bit or 2-bit rotation is possible.
	ROTXL ROTXR	B/W/L	Rd (rotate) → Rd Rotates general register contents through the carry flag. 1-bit or 2-bit rotation is possible.

Note: *1 Size refers to the operand size.
B: Byte
W: Word
L: Longword

Type	Instruction	Size*1	Function
Bit-manipulation instructions	BSET	B	1 → (<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 → (<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	¬ (<bit-No.> of <EAd>) → (<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	¬ (<bit-No.> of <EAd>) → 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 ∧ (<bit-No.> of <EAd>) → C 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 ∧ ¬ (<bit-No.> of <EAd>) → C 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 ∨ (<bit-No.> of <EAd>) → C 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 ∨ ¬ (<bit-No.> of <EAd>) → C 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.

Note: *1 Size refers to the operand size.
B: Byte

Type	Instruction	Size*1	Function
Bit-manipulation instructions	BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C 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 ⊕ ¬ (<bit-No.> of <EAd>) → C 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	(<bit-No.> of <EAd>) → C Transfers a specified bit in a general register or memory operand to the carry flag.
	BILD	B	¬ (<bit-No.> of <EAd>) → 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 → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in a general register or memory operand.
	BIST	B	¬ C → (<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: *1 Size refers to the operand size.
B: Byte

Type	Instruction	Size	Function
Branch instructions	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 ∨ Z = 0
BLS	Low or same	C ∨ 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 ⊕ V = 0
BLT	Less than	N ⊕ V = 1
BGT	Greater than	Z∨(N ⊕ V) = 0
BLE	Less or equal	Z∨(N ⊕ 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

Type	Instruction	Size*[1]	Function
System control instructions	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 source 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. 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: *1 Size refers to the operand size.
B: Byte
W: Word

Type	Instruction	Size	Function
Block data transfer instruction	EEPMOV.B	—	if R4L ≠ 0 then Repeat @ER5+ → @ER6+ R4L–1 → R4L Until R4L = 0 else next;
	EEPMOV.W	—	if R4 ≠ 0 then Repeat @ER5+ → @ER6+ R4–1 → R4 Until R4 = 0 else next;
			Transfers a data block according to parameters set in general registers R4L or R4, ER5, and ER6.
			R4L or R4: size of block (bytes) ER5: starting source address ER6: starting destination address
			Execution of the next instruction begins as soon as the transfer is completed.

2.6.4 Basic Instruction Formats

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

Figure 2-13 shows examples of instruction formats.

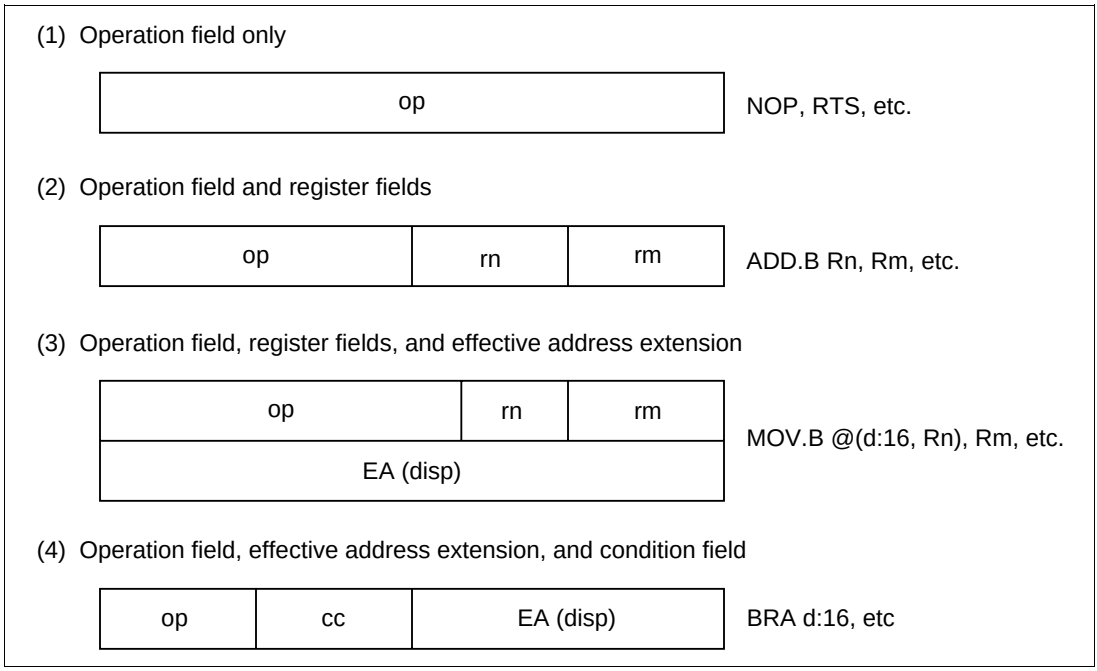


Figure 2-13 Instruction Formats (Examples)

(1) 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.

(2) Register Field: Specifies a general register. Address registers are specified by 3 bits, data registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no register field.

(3) Effective Address Extension: Eight, 16, or 32 bits specifying immediate data, an absolute address, or a displacement.

(4) Condition Field: Specifies the branching condition of Bcc instructions.

2.6.5 Notes on Use of Bit-Manipulation Instructions

The BSET, BCLR, BNOT, BST, and BIST instructions read a byte of data, carry out bit manipulation, then write back the byte of data. Caution is therefore required when using these instructions on a register containing write-only bits, or a port.

The BCLR instruction can be used to clear internal I/O register flags to 0. In this case, the relevant flag need not be read beforehand if it is clear that it has been set to 1 in an interrupt handling routine, etc.

2.7 Addressing Modes and Effective Address Calculation

2.7.1 Addressing Mode

The CPU supports the eight addressing modes listed in table 2-4. Each instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions 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 2-4 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

(1) Register Direct—Rn: The register field of the instruction specifies an 8-, 16-, or 32-bit general register containing 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) Register Indirect—@ERn: The register field of the instruction code specifies an address register (ERn) which contains the address of the operand on memory. 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).

(3) Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn): A 16-bit or 32-bit displacement contained in the instruction 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.

(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 transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, 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 transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, the register value should be even.

(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).

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'FFFFFF). For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute address can access the entire address space.

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).

Table 2-5 indicates the accessible absolute address ranges.

Table 2-5 Absolute Address Access Ranges

Absolute Address		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)		

Note: * Not available in the H8S/2214.

(6) Immediate—#xx:8, #xx:16, or #xx:32: The instruction contains 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data as an operand.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. 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.

(7) Program-Counter Relative—@(d:8, PC) or @(d:16, PC): This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction is sign-extended and added to the 24-bit PC contents to generate a 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.

(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. This memory operand contains a branch address. The upper bits of the 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 all 0 (H'00).

Note that the first part of the address range is also the exception vector area. For further details, refer to section 4, Exception Handling.

Note: * Not available in the H8S/2214.

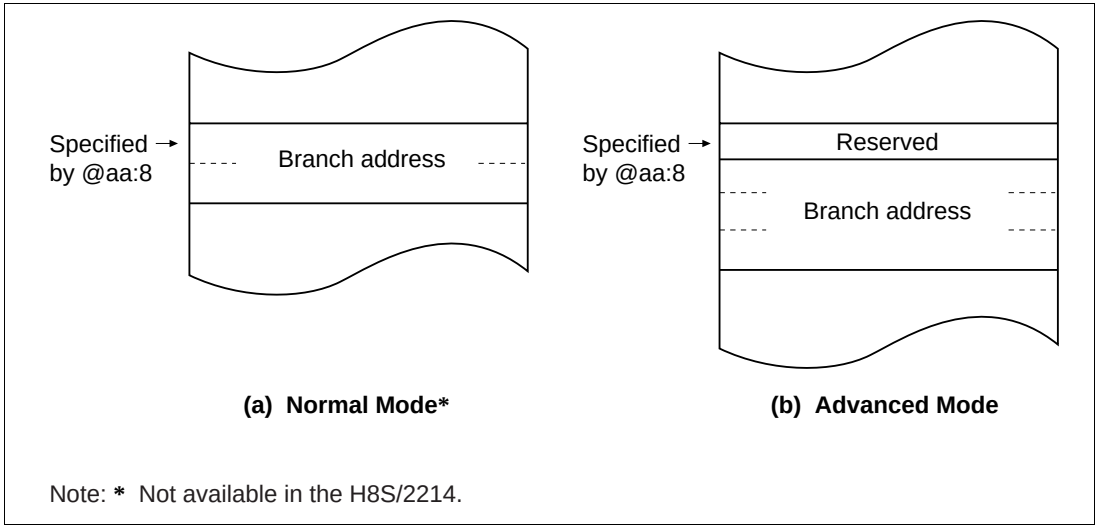


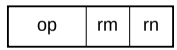
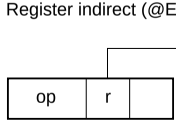
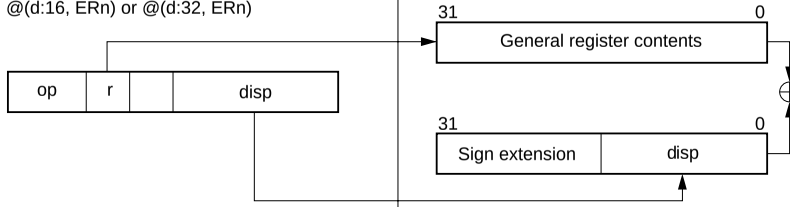
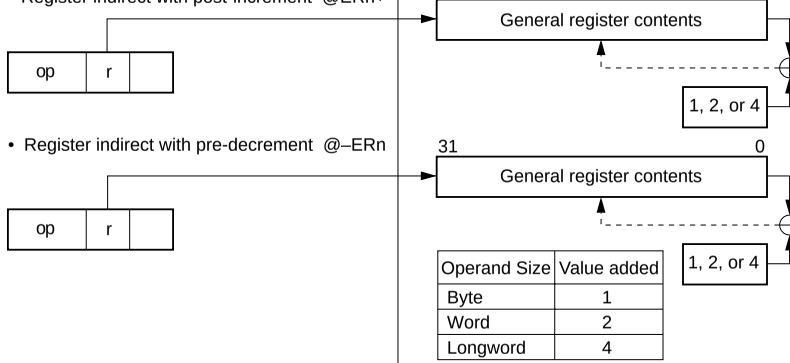
Figure 2-14 Branch Address Specification in Memory Indirect Mode

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 instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

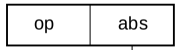
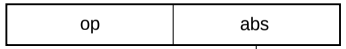
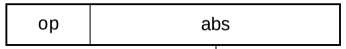
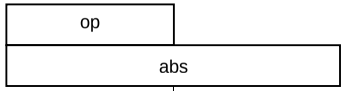
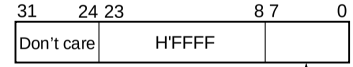
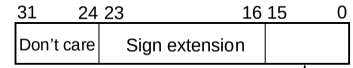
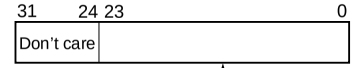
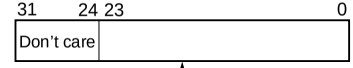
2.7.2 Effective Address Calculation

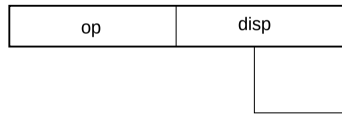
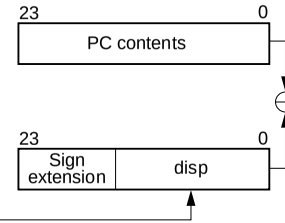
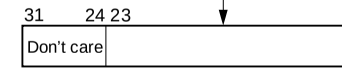
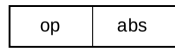
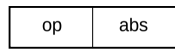
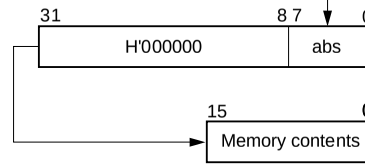
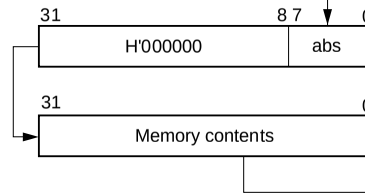
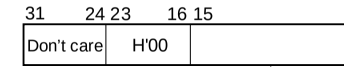
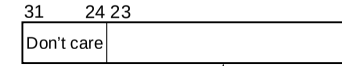
Table 2-6 indicates how effective addresses are calculated in each addressing mode. In normal mode the upper 8 bits of the effective address are ignored in order to generate a 16-bit address.

Table 2-6 Effective Address Calculation

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 		

Operand Size	Value added
Byte	1
Word	2
Longword	4

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	Absolute address @aa:8  @aa:16  @aa:24  @aa:32  		   
6	Immediate #xx:8/#xx:16/#xx:32  		Operand is immediate data.

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
7	Program-counter relative @(d:8, PC)/@(d:16, PC) 		
8	Memory indirect @@aa:8 • Normal mode*  • Advanced mode 	 	 

Note: * Not available in the H8S/2214.

2.8 Processing States

2.8.1 Overview

The CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and power-down state. Figure 2-15 shows a diagram of the processing states. Figure 2-16 indicates the state transitions.

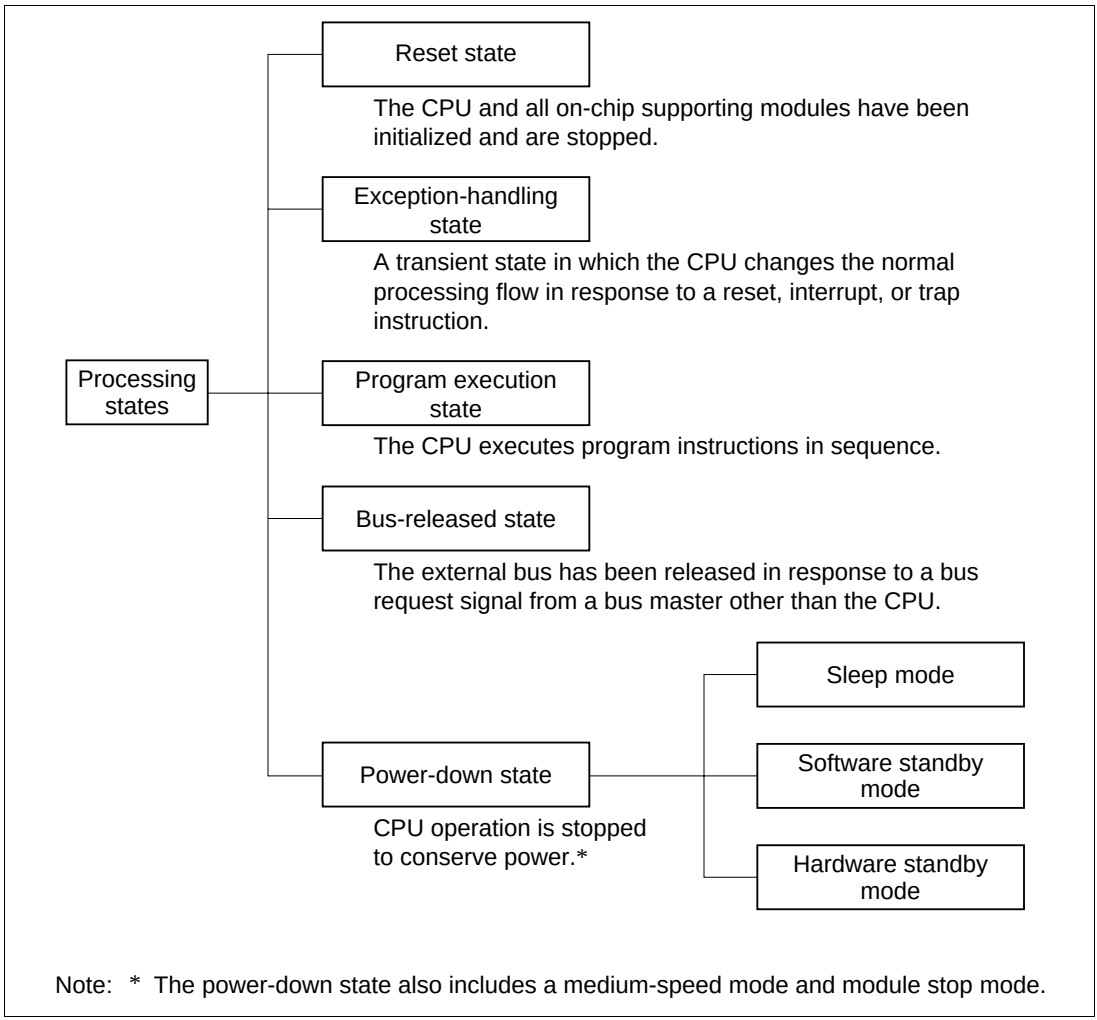
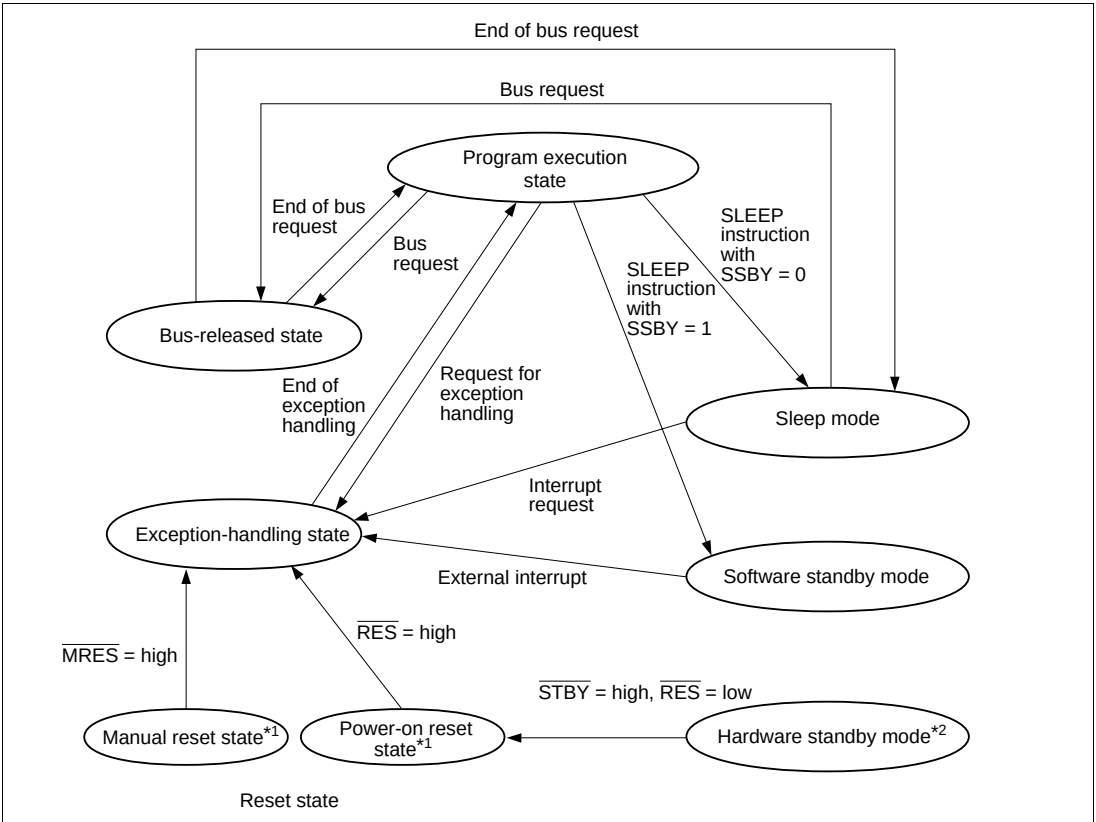


Figure 2-15 Processing States



Notes: *1 From any state except hardware standby mode, a transition to the power-on reset state occurs whenever $\overline{RES}$ goes low. From any state except hardware standby mode and the power-on reset state, a transition to the manual reset state occurs whenever $\overline{MRES}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.

*2 From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.

Figure 2-16 State Transitions

2.8.2 Reset State

When the $\overline{RES}$ input goes low all current processing stops and the CPU enters the power-on reset state. When the $\overline{MRES}$ input goes low, the CPU enters the manual reset state. All interrupts are disabled in the reset state. Reset exception handling starts when the $\overline{RES}$ or $\overline{MRES}$ signal changes from low to high.

The reset state can also be entered by a watchdog timer overflow. For details, refer to section 11, Watchdog Timer.

2.8.3 Exception-Handling State

The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to a reset, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address.

(1) Types of Exception Handling and Their Priority

Exception handling is performed for resets, traces, interrupts, and trap instructions. Table 2-7 indicates the types of exception handling and their priority. Trap instruction exception handling is always accepted, in the program execution state.

Exception handling and the stack structure depend on the interrupt control mode set in SYSCR.

Table 2-7 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception Handling
High ↑ Low	Reset	Synchronized with clock	Exception handling starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin, or when the watchdog timer overflows.
	Trace	End of instruction execution or end of exception-handling sequence*1	When the trace (T) bit is set to 1, the trace starts at the end of the current instruction or current exception-handling sequence
	Interrupt	End of instruction execution or end of exception-handling sequence*2	When an interrupt is requested, exception handling starts at the end of the current instruction or current exception-handling sequence
	Trap instruction	When TRAPA instruction is executed	Exception handling starts when a trap (TRAPA) instruction is executed*3

Notes: *1 Traces are enabled only in interrupt control mode 2. Trace exception-handling is not executed at the end of the RTE instruction.

*2 Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC instructions, or immediately after reset exception handling.

*3 Trap instruction exception handling is always accepted, in the program execution state.

(2) Reset Exception Handling

After the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin has gone low and the reset state has been entered, reset exception handling starts when $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ goes high again. The CPU enters the power-on reset state when the $\overline{\text{RES}}$ pin is low, and the manual reset state when the $\overline{\text{MRES}}$ pin is low. When reset exception handling starts the CPU fetches a start address (vector) from the exception vector table and starts program execution from that address. All interrupts, including NMI, are disabled during reset exception handling and after it ends.

(3) Traces

Traces are enabled only in interrupt control mode 2. Trace mode is entered when the T bit of EXR is set to 1. When trace mode is established, trace exception handling starts at the end of each instruction.

At the end of a trace exception-handling sequence, the T bit of EXR is cleared to 0 and trace mode is cleared. Interrupt masks are not affected.

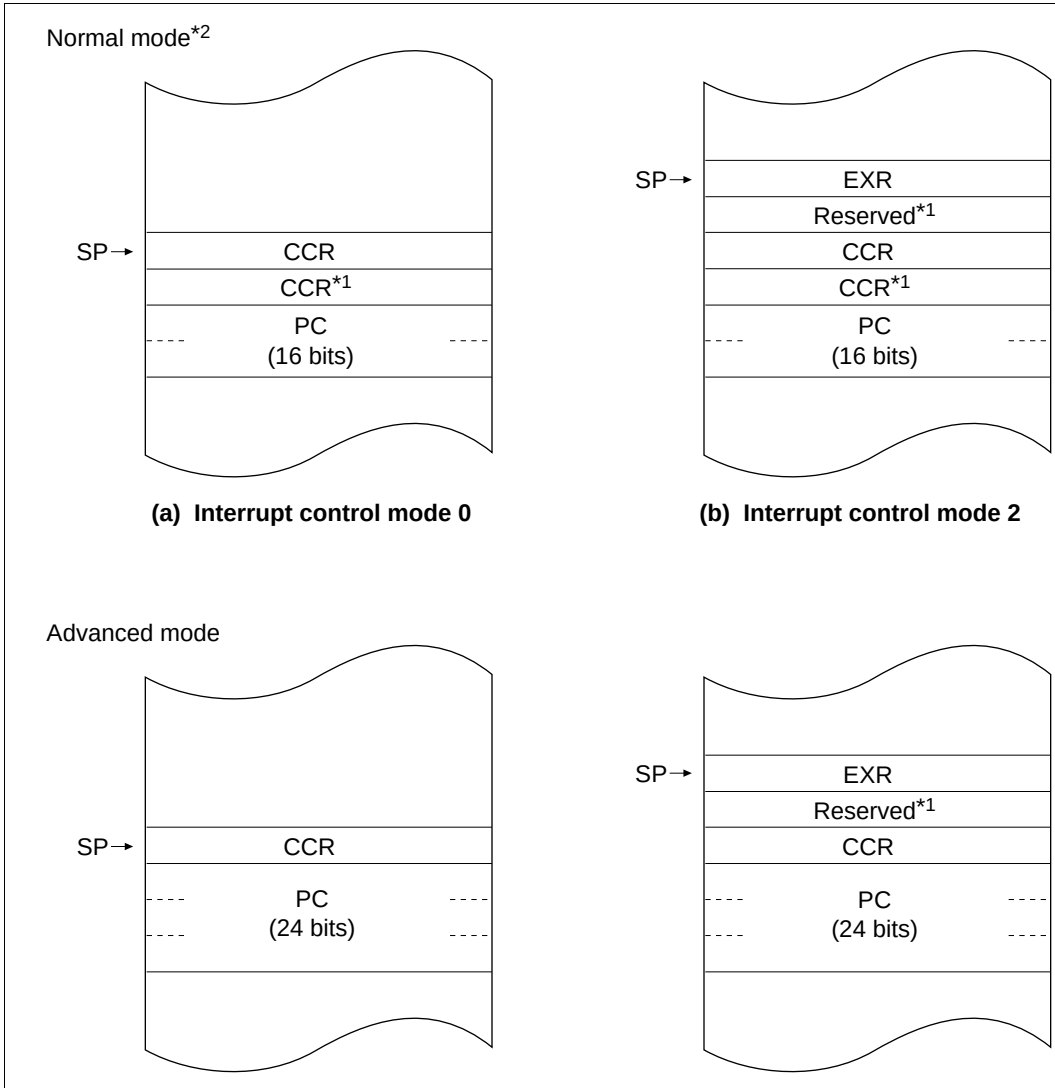
The T bit saved on the stack retains its value of 1, and when the RTE instruction is executed to return from the trace exception-handling routine, trace mode is entered again. Trace exception-handling is not executed at the end of the RTE instruction.

Trace mode is not entered in interrupt control mode 0, regardless of the state of the T bit.

(4) Interrupt Exception Handling and Trap Instruction Exception Handling

When interrupt or trap-instruction exception handling begins, the CPU references the stack pointer (ER7) and pushes the program counter and other control registers onto the stack. Next, the CPU alters the settings of the interrupt mask bits in the control registers. Then the CPU fetches a start address (vector) from the exception vector table and program execution starts from that start address.

Figure 2-17 shows the stack after exception handling ends.



Notes: *1 Ignored when returning.
*2 Not available in the H8S/2214.

Figure 2-17 Stack Structure after Exception Handling (Examples)

2.8.4 Program Execution State

In this state the CPU executes program instructions in sequence.

2.8.5 Bus-Released State

This is a state in which 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.

There are two other bus masters in addition to the CPU: the DMA controller (DMAC) and data transfer controller (DTC).

For further details, refer to section 6, Bus Controller.

2.8.6 Power-Down State

The power-down state includes both modes in which the CPU stops operating and modes in which the CPU does not stop. There are five modes in which the CPU stops operating: sleep mode, software standby mode, and hardware standby mode. There are also three other power-down modes: medium-speed mode, module stop mode, and subactive mode. In medium-speed mode the CPU and other bus masters operate on a medium-speed clock. Module stop mode permits halting of the operation of individual modules, other than the CPU. For details, refer to section 17, Power-Down State.

(1) Sleep Mode: A transition to sleep mode is made if the SLEEP instruction is executed while the SSBY bit in SBYCR and the LSON bit in LPWRCR are both cleared to 0. In sleep mode, CPU operations stop immediately after execution of the SLEEP instruction. The contents of CPU registers are retained.

(2) Software Standby Mode: A transition to software standby mode is made if the SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1, and the LSON bit in LPWRCR and the PSS bit in TCSR (WDT1) are both cleared to 0. In software standby mode, the CPU and clock halt and all MCU operations stop. As long as a specified voltage is supplied, the contents of CPU registers and on-chip RAM are retained. The I/O ports also remain in their existing states.

(3) Hardware Standby Mode: A transition to hardware standby mode is made when the $\overline{\text{STBY}}$ pin goes low. In hardware standby mode, the CPU and clock halt and all MCU operations stop. The on-chip supporting modules are reset, but as long as a specified voltage is supplied, on-chip RAM contents are retained.

2.9 Basic Timing

2.9.1 Overview

The CPU is driven by a system clock, denoted by the symbol ø. The period from one rising edge of ø to the next is referred to as a "state." The memory cycle or bus cycle consists of one, two, or three states. Different methods are used to access on-chip memory, on-chip supporting modules, and the external address space.

2.9.2 On-Chip Memory (ROM, RAM)

On-chip memory is accessed in one state. The data bus is 16 bits wide, permitting both byte and word transfer instruction. Figure 2-18 shows the on-chip memory access cycle. Figure 2-19 shows the pin states.

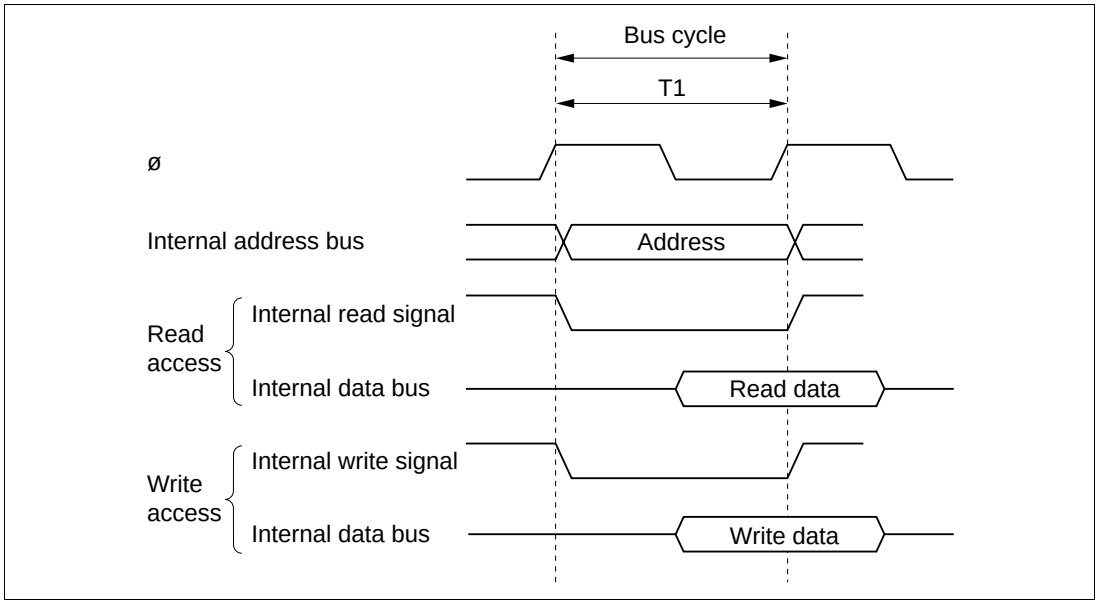


Figure 2-18 On-Chip Memory Access Cycle

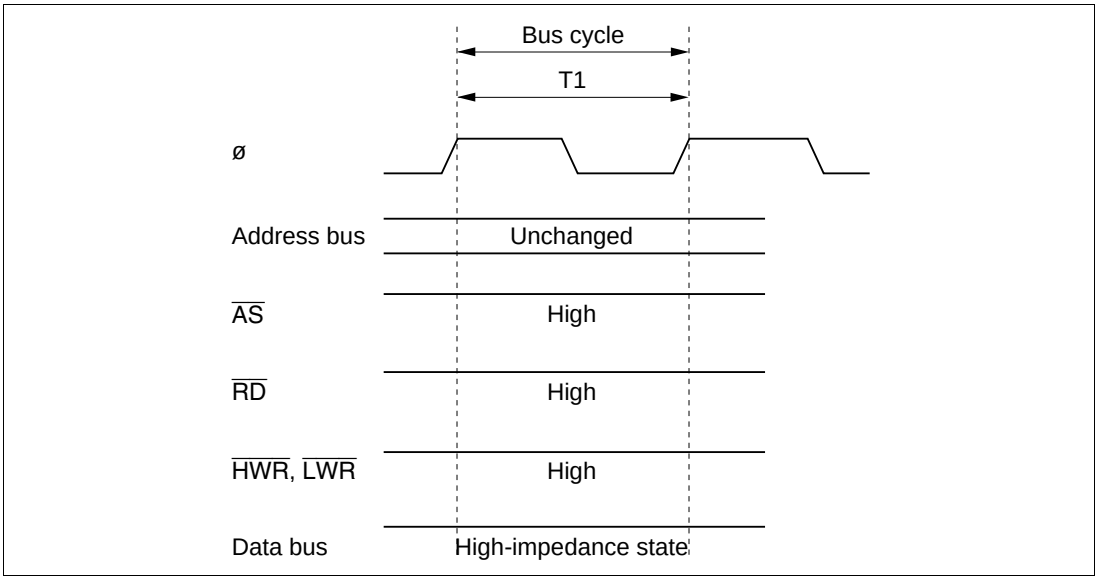


Figure 2-19 Pin States during On-Chip Memory Access

2.9.3 On-Chip Supporting Module Access Timing

The on-chip supporting modules are accessed in two states. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed. Figure 2-20 shows the access timing for the on-chip supporting modules. Figure 2-21 shows the pin states.

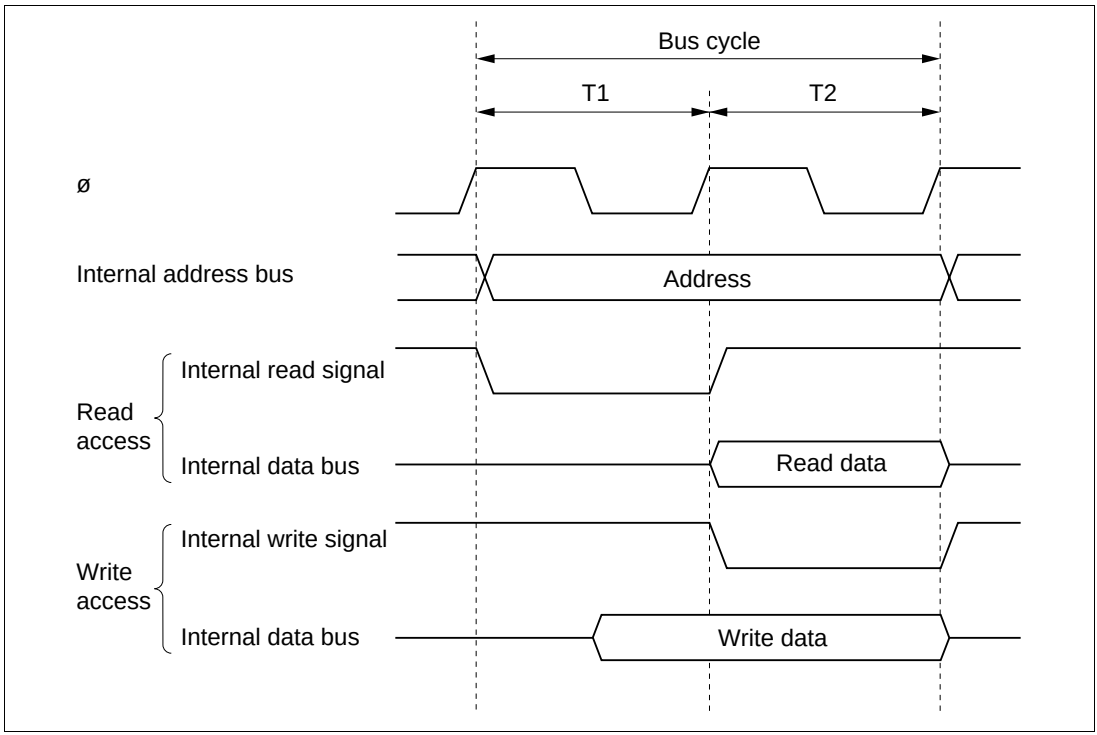


Figure 2-20 On-Chip Supporting Module Access Cycle

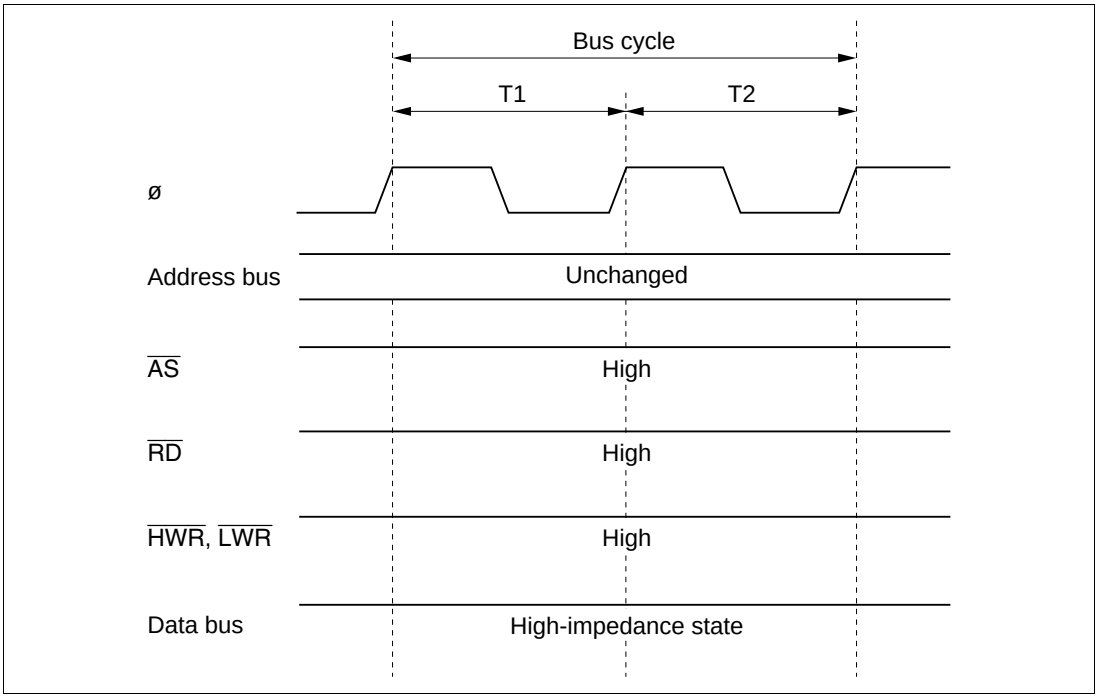


Figure 2-21 Pin States during On-Chip Supporting Module Access

2.9.4 External Address Space Access Timing

The external address space is accessed with an 8-bit or 16-bit data bus width in a two-state or three-state bus cycle. In three-state access, wait states can be inserted. For further details, refer to section 6, Bus Controller.

2.10 Usage Note

Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction. The TAS instruction is not generated by the Hitachi H8S and H8/300 series C/C++ compilers. If the TAS instruction is used as a user-defined intrinsic function, ensure that only register ER0, ER1, ER4, or ER5 is used.

Section 3 MCU Operating Modes

3.1 Overview

3.1.1 Operating Mode Selection

The H8S/2214 has four operating modes (modes 4 to 7). These modes enable selection of the CPU operating mode, enabling/disabling of on-chip ROM, and the initial bus width setting, by setting the mode pins (MD2 to MD0).

Table 3-1 lists the MCU operating modes.

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	External Data Bus: Max. Width
0*	0	0	0	—	—	—	—	
1*			1					
2*		1	0					
3*			1					
4	1	0	0	Advanced	On-chip ROM disabled, expanded mode	Disabled	16 bits	16 bits
5			1				8 bits	16 bits
6		1	0		On-chip ROM enabled, expanded mode	Enabled	8 bits	16 bits
7			1		Single-chip mode		—	

Note: * Not available in the H8S/2214.

The CPU's architecture allows for 4 Gbytes of address space, but the H8S/2214 actually accesses a maximum of 16 Mbytes.

Modes 4 to 6 are externally expanded modes that allow access to external memory and peripheral devices.

The external expansion modes allow switching between 8-bit and 16-bit bus modes. After program execution starts, an 8-bit or 16-bit address space can be set for each area, depending on the bus controller setting. If 16-bit access is selected for any one area, 16-bit bus mode is set; if 8-bit access is selected for all areas, 8-bit bus mode is set.

Note that the functions of each pin depend on the operating mode.

The H8S/2214 can be used only in modes 4 to 7. This means that the mode pins must be set to select one of these modes. Do not change the inputs at the mode pins during operation.

3.1.2 Register Configuration

The H8S/2214 has a mode control register (MDCR) that indicates the inputs at the mode pins (MD2 to MD0), and a system control register (SYSCR) that controls the operation of the H8S/2214. Table 3-2 summarizes these registers.

Table 3-2 MCU Registers

Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R	Undetermined	H'FDE7
System control register	SYSCR	R/W	H'01	H'FDE5

Note: * Lower 16 bits of the address.

3.2 Register Descriptions

3.2.1 Mode Control Register (MDCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	MDS2	MDS1	MDS0
Initial value:		1	0	0	0	0	—*	—*	—*
R/W	:	—	—	—	—	—	R	R	R

Note: * Determined by pins MD2 to MD0.

MDCR is an 8-bit read-only register that indicates the current operating mode of the H8S/2214.

Bit 7—Reserved: Read-only bit, always read as 1.

Bits 6 to 3—Reserved: Read-only bits, always read as 0.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): 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-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 power-on reset, but are retained after a manual reset.

3.2.2 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	INTM1	INTM0	NMIEG	MRESE	—	RAME
Initial value:		0	0	0	0	0	0	0	1
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—	R/W

SYSCR is an 8-bit readable/writable register that selects the interrupt control mode, the detected edge for NMI, and enables or disables $\overline{\text{MRES}}$ pin input and on-chip RAM.

SYSCR is initialized to H'01 by a power-on reset and in hardware standby mode. In a manual reset, the INTM1, INTM0, NMIEG, and RAME bits are initialized, but the MRESE bit is not. SYSCR is not initialized in software standby mode.

Bit 7—Reserved: Only 0 should be written to this bit.

Bit 6—Reserved: Read-only bit, always read as 0.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select the control mode of the interrupt controller. For details of the interrupt control modes, see section 5.4.1, Interrupt Control Modes and Interrupt Operation.

Bit 5 INTM1	Bit 4 INTM0	Interrupt Control Mode	Description	
0	0	0	Control of interrupts by I bit	(Initial value)
	1	—	Setting prohibited	
1	0	2	Control of interrupts by I2 to I0 bits and IPR	
	1	—	Setting prohibited	

Bit 3—NMI Edge Select (NMIEG): Selects the valid edge of the NMI interrupt input.

Bit 3 NMIEG	Description	
0	An interrupt is requested at the falling edge of NMI input	(Initial value)
1	An interrupt is requested at the rising edge of NMI input	

Bit 2—Manual Reset Select (MRESE): Enables or disables the $\overline{\text{MRES}}$ pin. Table 3-3 shows the relationship between the $\overline{\text{RES}}$ and $\overline{\text{MRES}}$ pin values and type of reset. For details of resets, see section 4.2, Resets.

Bit 2

MRESE	Description	
0	Manual reset is disabled P74/$\overline{\text{MRES}}$ pin can be used as P74 I/O pin	(Initial value)
1	Manual reset is enabled P74/$\overline{\text{MRES}}$ pin can be used as $\overline{\text{MRES}}$ input pin	

Table 3-3 Relationship between $\overline{\text{RES}}$ and $\overline{\text{MRES}}$ pin Values and Type of Reset

Pins		
$\overline{\text{RES}}$	$\overline{\text{MRES}}$	Type of Reset
0	*	Power-on reset
1	0	Manual reset
1	1	Operating state

*: Don't care

Bit 1—Reserved: Read-only bit, always read as 0.

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized when the reset status is released. It is not initialized in software standby mode.

Bit 0

RAME	Description	
0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

Note: When the DTC is used, the RAME bit should not be cleared to 0.

3.3 Operating Mode Descriptions

3.3.1 Mode 4

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.

Pins P13 to P10, and ports A, B, and C function as an address bus, ports D and E function as a data bus, and part of port F carries bus control signals.

Pins P13 to P11 function as input ports immediately after a reset. Address (A23 to A21) output can be enabled or disabled by bits AE3 to AE0 in the pin function control register (PFCR) regardless of the corresponding data direction register (DDR) values. Pin 10 and ports A and B function as address (A20 to A8) outputs immediately after a reset. Address output can be enabled or disabled by bits AE3 to AE0 in PFCR regardless of the corresponding DDR values. Pins for which address output is disabled among pins P13 to P10 and in ports A and B become port outputs when the corresponding DDR bits are set to 1.

Port C always has an address (A7 to A0) output function.

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

3.3.2 Mode 5

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.

Pins P13 to P10, and ports A, B, and C function as an address bus, ports D and E function as a data bus, and part of port F carries bus control signals.

Pins P13 to P11 function as input ports immediately after a reset. Address (A23 to A21) output can be enabled or disabled by bits AE3 to AE0 in the pin function control register (PFCR) regardless of the corresponding data direction register (DDR) values. Pin 10 and ports A and B function as address (A20 to A8) outputs immediately after a reset. Address output can be enabled or disabled by bits AE3 to AE0 in PFCR regardless of the corresponding DDR values. Pins for which address output is disabled among pins P13 to P10 and in ports A and B become port outputs when the corresponding DDR bits are set to 1.

Port C always has an address (A7 to A0) output function.

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

3.3.3 Mode 6

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled.

Pins P13 to P10, and ports A and B function as input ports immediately after a reset. Address (A23 to A8) output can be enabled or disabled by bits AE3 to AE0 in the pin function control register (PFCR) regardless of the corresponding data direction register (DDR) values. Pins for which address output is disabled among pins P13 to P10 and in ports A and B become port outputs when the corresponding DDR bits are set to 1.

Ports D and E function as a data bus, and part of port F carries data bus signals.

Port C is an input port immediately after a reset. Addresses A7 to A0 are output by setting the corresponding DDR bits to 1.

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

3.3.4 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, but external addresses cannot be accessed.

All I/O ports are available for use as input-output ports.

3.4 Pin Functions in Each Operating Mode

The pin functions of ports 1, and A to F vary depending on the operating mode. Table 3-3 shows their functions in each operating mode.

Table 3-3 Pin Functions in Each Mode

Port		Mode 4	Mode 5	Mode 6	Mode 7
Port 1	P13 to P11	P*/A	P*/A	P*/A	P
	P10	P/A*	P/A*	P*/A	P
Port A	PA3 to PA0	P/A*	P/A*	P*/A	P
Port B		P/A*	P/A*	P*/A	P
Port C		A	A	P*/A	P
Port D		D	D	D	P
Port E		P/D*	P*/D	P*/D	P
Port F	PF7	P/C*	P/C*	P/C*	P*/C
	PF6 to PF4	C	C	C	P
	PF3	P/C*	P*/C	P*/C	
	PF2 to PF0	P*/C	P*/C	P*/C	

Legend

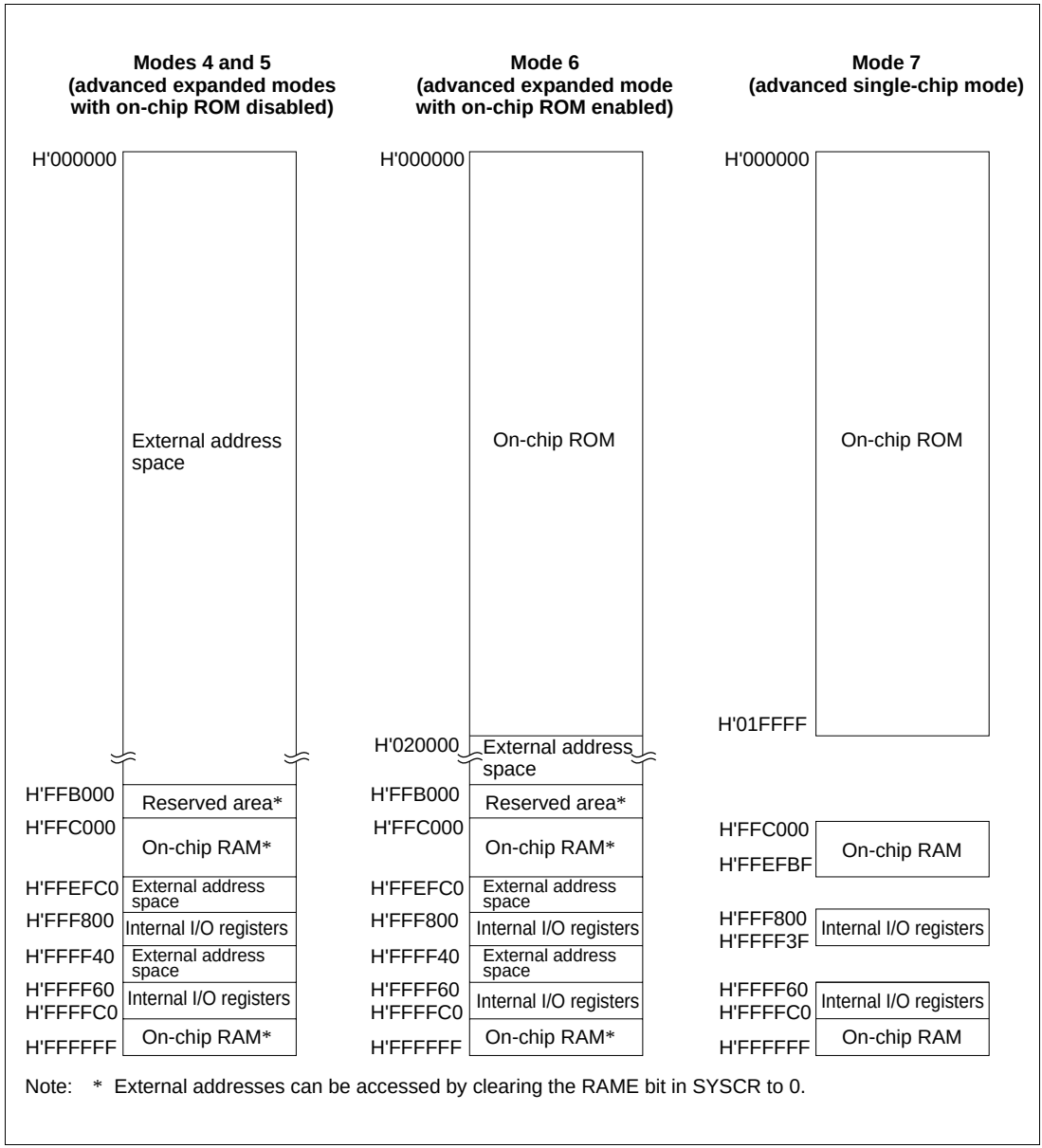
P: I/O port
A: Address bus output
D: Data bus I/O
C: Control signals, clock I/O
*: After reset

3.5 Memory Map in Each Operating Mode

The H8S/2214 memory map is shown in figure 3-1.

The address space is 16 Mbytes in modes 4 to 7 (advanced modes).

The address space is divided into eight areas for modes 4 to 7. For details, see section 6, Bus Controller.



Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3-1 Memory Map in Each Operating Mode in the H8S/2214

Section 4 Exception Handling

4.1 Overview

4.1.1 Exception Handling Types and Priority

As table 4-1 indicates, exception handling may be caused by a reset, trace, trap instruction, or interrupt. 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. Trap instruction exceptions are accepted at all times, in the program execution state.

Exception handling sources, the stack structure, and the operation of the CPU vary depending on the interrupt control mode set by the INTM0 and INTM1 bits of SYSCR.

Table 4-1 Exception Handling Types and Priority

Priority	Exception Handling Type	Start of Exception Handling
High ↑	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin, or when the watchdog timer overflows. The CPU enters the power-on reset state when the $\overline{\text{RES}}$ pin is low, and the manual reset state when the $\overline{\text{MRES}}$ pin is low.
	Trace*1	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit is set to 1
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued*2
Low	Trap instruction (TRAPA)*3	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 Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.

*3 Trap instruction exception handling requests are accepted at all times in program execution state.

4.1.2 Exception Handling Operation

Exceptions originate from various sources. Trap instructions and interrupts are handled as follows:

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

For a reset exception, steps 2 and 3 above are carried out.

4.1.3 Exception Sources and Vector Table

The exception sources are classified as shown in figure 4-1. Different vector addresses are assigned to different exception sources.

Table 4-2 lists the exception sources and their vector addresses.

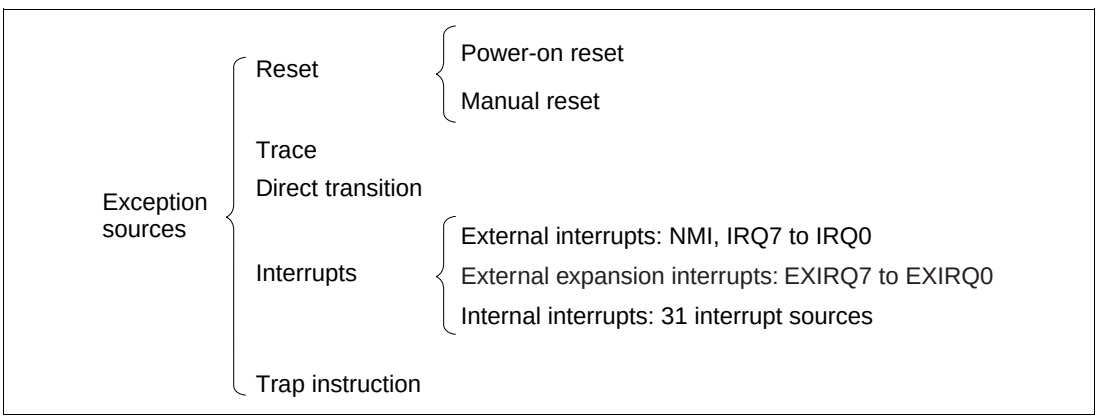


Figure 4-1 Exception Sources

Table 4-2 Exception Vector Table

Exception Source		Vector Number	Vector Address*[1] Advanced Mode
Power-on reset		0	H'0000 to H'0003
Manual reset		1	H'0004 to H'0007
Reserved for system use		2	H'0008 to H'000B
		3	H'000C to H'000F
		4	H'0010 to H'0013
Trace		5	H'0014 to H'0017
Direct transition		6	H'0018 to H'001B
External interrupt	NMI	7	H'001C to H'001F
Trap instruction (4 sources)		8	H'0020 to H'0023
		9	H'0024 to H'0027
		10	H'0028 to H'002B
		11	H'002C to H'002F
Reserved for system use		12	H'0030 to H'0033
		13	H'0034 to H'0037
		14	H'0038 to H'003B
		15	H'003C to H'003F
External interrupt	IRQ0	16	H'0040 to H'0043
	IRQ1	17	H'0044 to H'0047
	IRQ2	18	H'0048 to H'004B
	IRQ3	19	H'004C to H'004F
	IRQ4	20	H'0050 to H'0053
	IRQ5	21	H'0054 to H'0057
	IRQ6	22	H'0058 to H'005B
	IRQ7	23	H'005C to H'005F
Internal interrupt*[2]		24 ⎮ 111	H'0060 to H'0063 ⎮ H'01BC to H'01BF

Notes: *1 Lower 16 bits of the address.

*2 For details of internal interrupt vectors, see section 5.3.3, Interrupt Exception Handling Vector Table.

4.2 Reset

4.2.1 Overview

A reset has the highest exception priority.

When the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin goes low, all processing halts and the H8S/2214 enters the reset state. A reset initializes the internal state of the CPU and the registers of on-chip supporting modules. Immediately after a reset, interrupt control mode 0 is set.

Reset exception handling begins when the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin changes from low to high.

The levels of the $\overline{\text{RES}}$ and $\overline{\text{MRES}}$ pins at reset determine whether a power-on reset or a manual reset is effected.

The H8S/2214 can also be reset by overflow of the watchdog timer. For details see section 11, Watchdog Timer.

4.2.2 Reset Types

A reset can be of either of two types: a power-on reset or a manual reset. Reset types are shown in table 4-3. A power-on reset should be used when powering on.

The internal state of the CPU is initialized by either type of reset. A power-on reset also initializes all the registers in the on-chip supporting modules, while a manual reset initializes all the registers in the on-chip supporting modules except for the bus controller and I/O ports, which retain their previous states.
With a manual reset, since the on-chip supporting modules are initialized, ports used as on-chip supporting module I/O pins are switched to I/O ports controlled by DDR and DR.

Table 4-3 Reset Types

	Reset Transition Conditions		Internal State	
Type	**$\overline{\text{MRES}}$**	**$\overline{\text{RES}}$**	**CPU**	**On-Chip Supporting Modules**
Power-on reset	*	Low	Initialized	Initialized
Manual reset	Low	High	Initialized	Initialized, except for bus controller and I/O ports

*: Don't care

A reset caused by the watchdog timer can also be of either of two types: a power-on reset or a manual reset.

When the $\overline{\text{MRES}}$ pin is used, $\overline{\text{MRES}}$ pin input must be enabled by setting the MRESE bit to 1 in SYSCR.

4.2.3 Reset Sequence

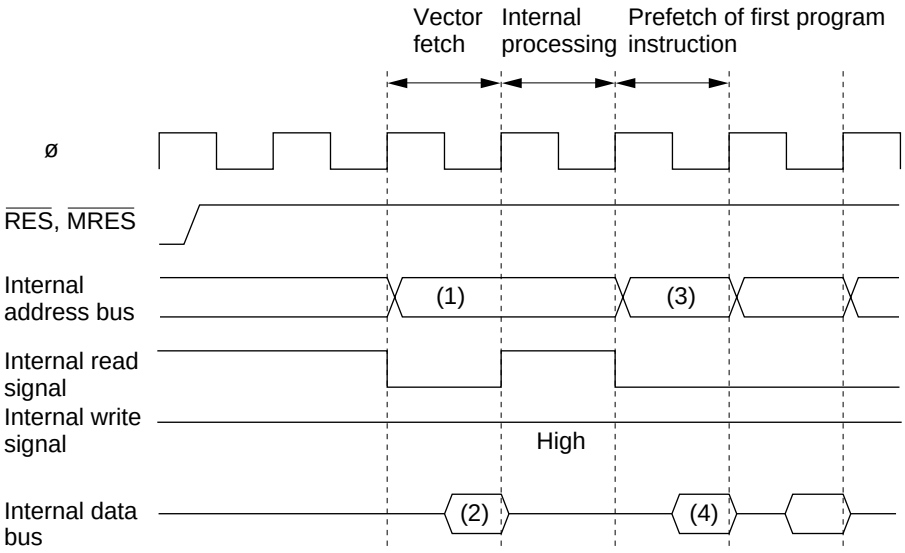
The H8S/2214 enters the reset state when the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin goes low.

To ensure that the H8S/2214 is reset, hold the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low for at least 20 ms at power-up. To reset the H8S/2214 during operation, hold the $\overline{\text{RES}}$ or $\overline{\text{MRES}}$ pin low for at least 20 states.

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

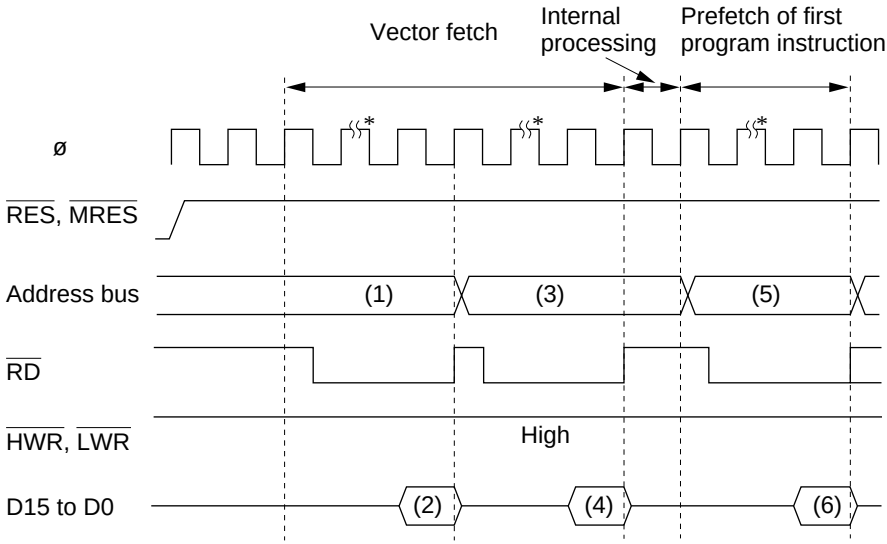
1. The internal state of the CPU and the registers of the on-chip supporting modules are initialized, the T bit is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR.
2. The reset exception handling vector address is read and transferred to the PC, and program execution starts from the address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

Figures 4-2 and 4-3 show examples of the reset sequence.



(1) Reset exception handling vector address (for a power-on reset, (1) = H'0000; for a manual reset, (1) = H'0002)
(2) Start address (contents of reset exception handling vector address)
(3) Start address ((3) = (2))
(4) First program instruction

Figure 4-2 Reset Sequence (Modes 2 and 3: Not available in the H8S/2214)



(1) (3) Reset exception handling vector address (for a power-on reset, (1) = H'000000, (3) = H'000002; for a manual reset, (1) = H'000004, (3) = H'000006)
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Note: * Three program wait states are inserted.

Figure 4-3 Reset Sequence (Mode 4)

4.2.4 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).

4.2.5 State of On-Chip Supporting Modules after Reset Release

After reset release, MSTPCRA is initialized to H'3F, MSTPCRB and MSTPCRC are initialized to H'FF, and all modules except the DMAC and DTC enter module stop mode. Consequently, on-chip supporting module registers cannot be read or written to. Register reading and writing is enabled when module stop mode is exited.

4.3 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 of 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 canceled by clearing the T bit in EXR to 0. It is not affected by interrupt masking.

Table 4-4 shows the state of CCR and EXR after execution of trace exception handling.

Interrupts are accepted even within the trace exception handling routine.

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.

Table 4-4 Status of CCR and EXR after Trace Exception Handling

	CCR		EXR	
Interrupt Control Mode	**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.

4.4 Interrupts

Interrupt exception handling can be requested by nine external sources (NMI, IRQ7 to IRQ0), eight external expansion sources (EXIRQ7 to EXIRQ0), and 31 internal sources in the on-chip supporting modules. Figure 4-4 classifies the interrupt sources and the number of interrupts of each type.

The on-chip supporting modules that can request interrupts include the watchdog timer (WDT), 16-bit timer-pulse unit (TPU), serial communication interface (SCI), data transfer controller (DTC), and DMA controller (DMAC). Each interrupt source has a separate vector address.

NMI is the highest-priority interrupt. 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.

For details of interrupts, see section 5, Interrupt Controller.

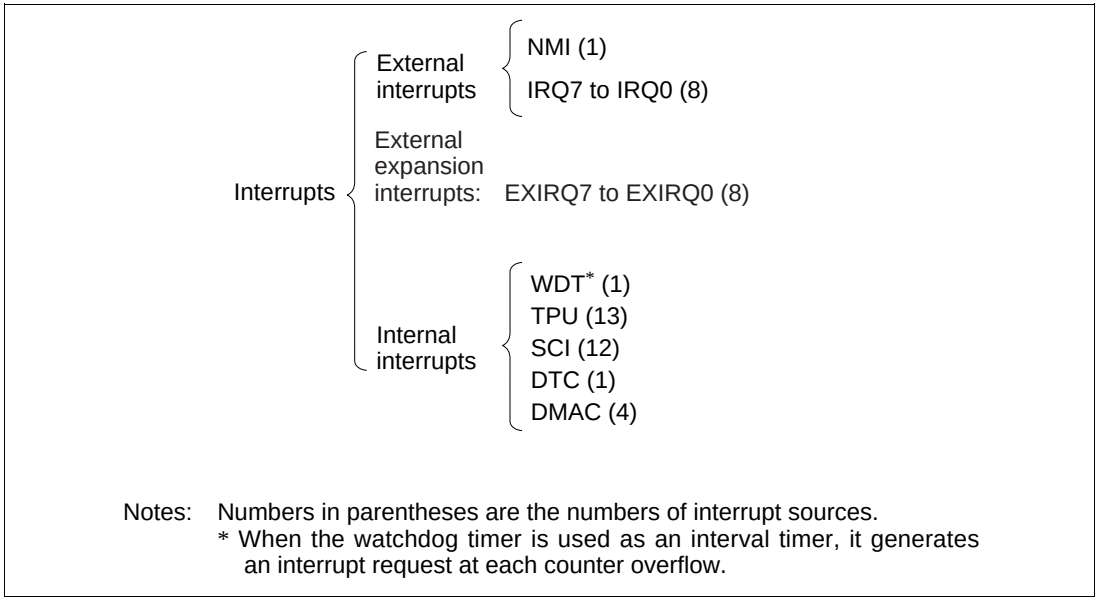


Figure 4-4 Interrupt Sources and Number of Interrupts

4.5 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 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 4-5 shows the status of CCR and EXR after execution of trap instruction exception handling.

Table 4-5 Status of CCR and EXR after Trap Instruction Exception Handling

	CCR		EXR	
Interrupt Control Mode	**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.

4.6 Stack Status after Exception Handling

Figures 4-5 and 4-6 show the stack after completion of trap instruction exception handling and interrupt exception handling.

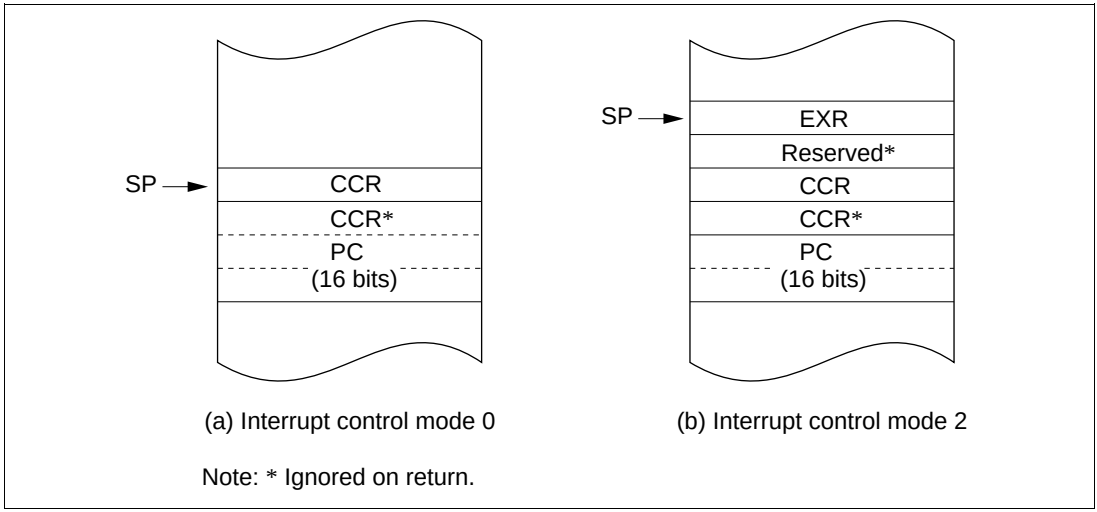


Figure 4-5 Stack Status after Exception Handling (Normal Modes: Not available in the H8S/2214)

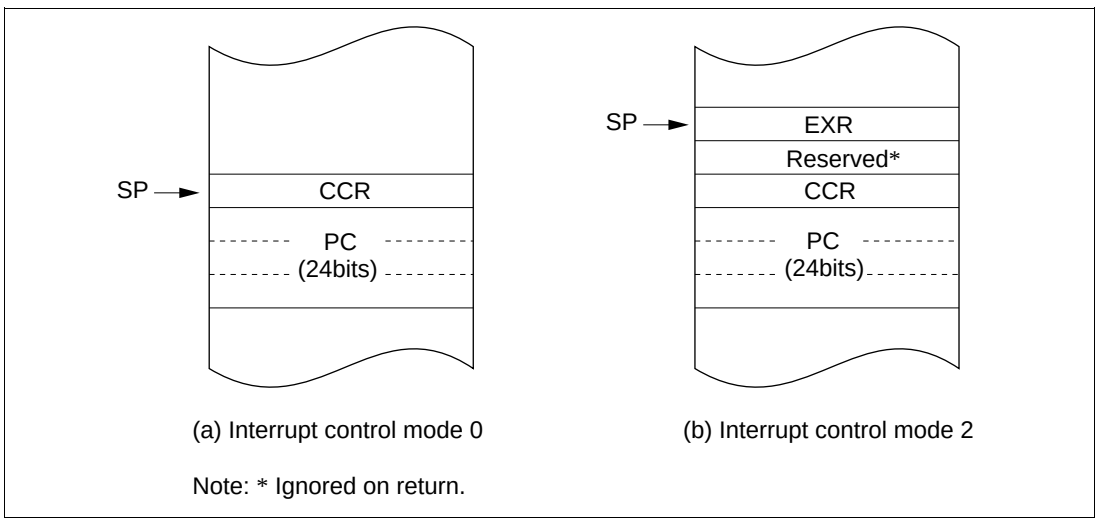


Figure 4-6 Stack Status after Exception Handling (Advanced Modes)

4.7 Notes on Use of the Stack

When accessing word data or longword data, the H8S/2214 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 4-7 shows an example of what happens when the SP value is odd.

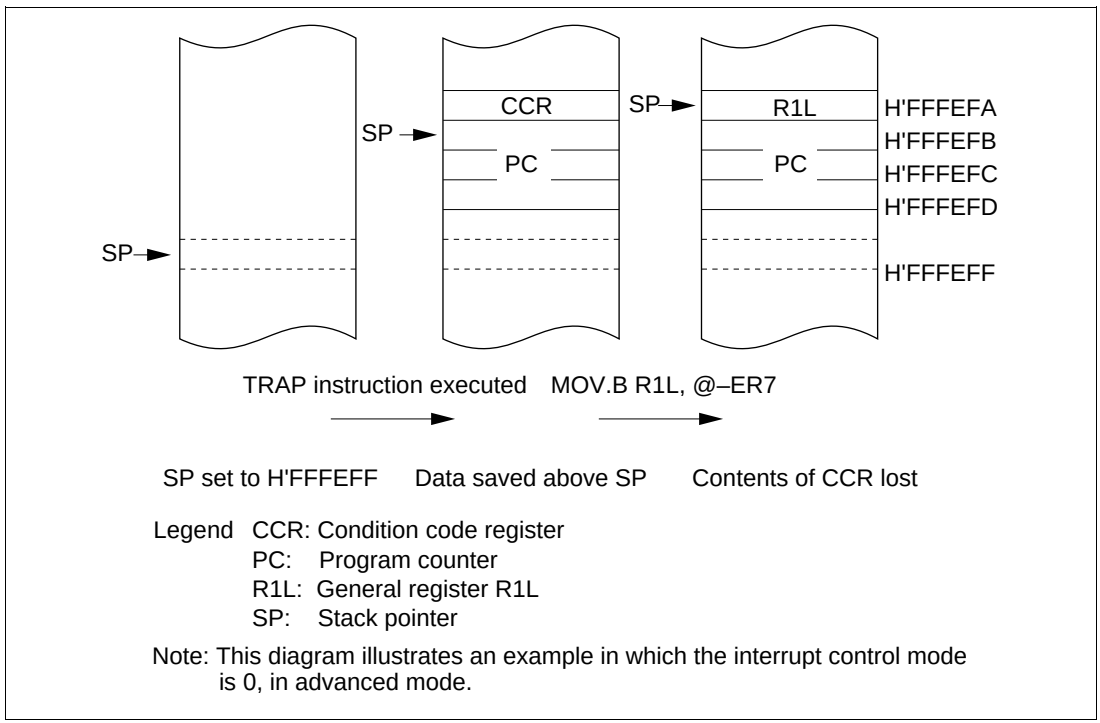


Figure 4-7 Operation when SP Value is Odd

Section 5 Interrupt Controller

5.1 Overview

5.1.1 Features

The H8S/2214 controls interrupts by means of an interrupt controller. The interrupt controller has the following features:

- Two interrupt control modes
  - Any of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR).
- 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 IRQ7 to IRQ0.
- DTC or DMAC control
  - DTC or DMAC activation is performed by means of interrupts.
- Eight external expansion interrupt input pins

5.1.2 Block Diagram

A block diagram of the interrupt controller is shown in Figure 5-1.

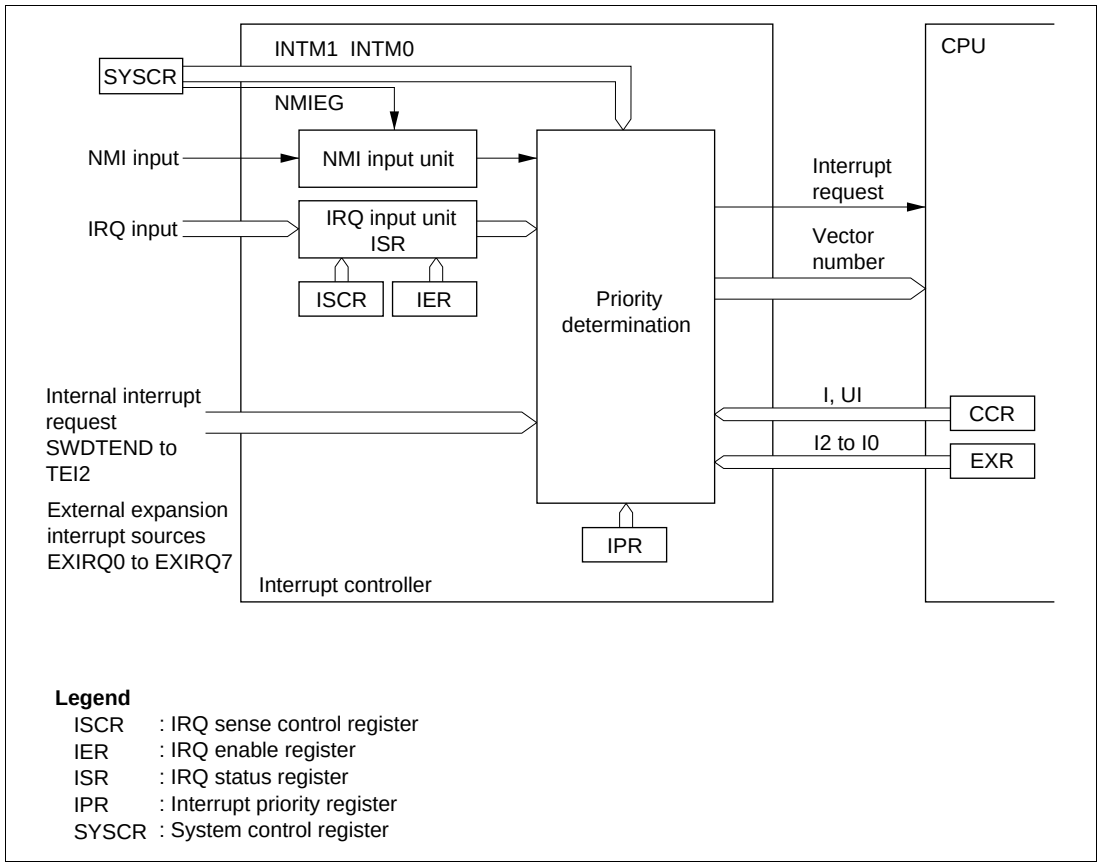


Figure 5-1 Block Diagram of Interrupt Controller

5.1.3 Pin Configuration

Table 5-1 summarizes the pins of the interrupt controller.

Table 5-1 Interrupt Controller Pins

Name	Symbol	I/O	Function
Nonmaskable interrupt	NMI	Input	Nonmaskable external interrupt; rising or falling edge can be selected
External interrupt requests 7 to 0	$\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$	Input	Maskable external interrupts; rising, falling, or both edges, or level sensing, can be selected
External expansion interrupt sources 7 to 0	$\overline{\text{EXIRQ7}}$ to $\overline{\text{EXIRQ0}}$	Input	Interrupts from external expansion modules. Interrupt is accepted on low level.

5.1.4 Register Configuration

Table 5-2 summarizes the registers of the interrupt controller.

Table 5-2 Interrupt Controller Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
System control register	SYSCR	R/W	H'01	H'FDE5
IRQ sense control register H	ISCRH	R/W	H'00	H'FE12
IRQ sense control register L	ISCRL	R/W	H'00	H'FE13
IRQ enable register	IER	R/W	H'00	H'FE14
IRQ status register	ISR	R/(W)*[2]	H'00	H'FE15
Interrupt priority register A	IPRA	R/W	H'77	H'FEC0
Interrupt priority register B	IPRB	R/W	H'77	H'FEC1
Interrupt priority register C	IPRC	R/W	H'77	H'FEC2
Interrupt priority register D	IPRD	R/W	H'77	H'FEC3
Interrupt priority register F	IPRF	R/W	H'77	H'FEC5
Interrupt priority register G	IPRG	R/W	H'77	H'FEC6
Interrupt priority register J	IPRJ	R/W	H'77	H'FEC9
Interrupt priority register K	IPRK	R/W	H'77	H'FECA
Interrupt priority register M	IPRM	R/W	H'77	H'FECC

Notes: *1 Lower 16 bits of the address.
*2 Can only be written with 0 for flag clearing.

5.2 Register Descriptions

5.2.1 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	INTM1	INTM0	NMIEG	MRESE	—	RAME
Initial value:		0	0	0	0	0	0	0	1
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—	R/W

SYSCR is an 8-bit readable/writable register that selects the interrupt control mode, and the detected edge for NMI.

Only bits 5 to 3 are described here; for details of the other bits, see section 3.2.2, System Control Register (SYSCR).

SYSCR is initialized to H'01 by a power-on reset and in hardware standby mode. In a manual reset, the INTM1, INTM0, NMIEG, and RAME bits are initialized, but the MRESE bit is not. SYSCR is not initialized in software standby mode.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select one of two interrupt control modes for the interrupt controller.

Bit 5	Bit 4	Interrupt		
INTM1	**INTM0**	**Control Mode**	**Description**	
0	0	0	Interrupts are controlled by I bit	(Initial value)
	1	—	Setting prohibited	
1	0	2	Interrupts are controlled by bits I2 to I0, and IPR	
	1	—	Setting prohibited	

Bit 3—NMI Edge Select (NMIEG): Selects the input edge for the NMI pin.

Bit 3		
NMIEG	**Description**	
0	Interrupt request generated at falling edge of NMI input	(Initial value)
1	Interrupt request generated at rising edge of NMI input	

5.2.2 Interrupt Priority Registers A to D, F, G, J, K, M (IPRA to IPRD, IPRF, IPRG, IPRJ, IPRK, IPRM)

Bit	:	7	6	5	4	3	2	1	0
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0
Initial value:		0	1	1	1	0	1	1	1
R/W	:	—	R/W	R/W	R/W	—	R/W	R/W	R/W

The IPR registers are nine 8-bit readable/writable registers that set priorities (levels 7 to 0) for interrupts other than NMI.

The correspondence between IPR settings and interrupt sources is shown in table 5-3.

The IPR registers set a priority (level 7 to 0) for each interrupt source other than NMI.

The IPR registers are initialized to H'77 by a reset and in hardware standby mode.

They are not initialized in software standby mode.

Bits 7 and 3—Reserved: Read-only bits, always read as 0.

Table 5-3 Correspondence between Interrupt Sources and IPR Settings

	Bits	
Register	**6 to 4**	**2 to 0**
IPRA	IRQ0	IRQ1
IPRB	IRQ2 IRQ3	IRQ4 IRQ5
IPRC	IRQ6 IRQ7	DTC
IPRD	Watchdog timer 0	—*
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	—
IPRJ	DMAC	SCI channel 0
IPRK	SCI channel 1	SCI channel 2
IPRM	EXIRQ3 to EXIRQ0	EXIRQ7 to EXIRQ4

Note: * Reserved bits. These bits cannot be modified and are always read as 1.

As shown in table 5-3, multiple interrupts are assigned to one IPR. Setting a value in the range from H'0 to H'7 in the 3-bit groups of bits 6 to 4 and 2 to 0 sets the priority of the corresponding interrupt. The lowest priority level, level 0, is assigned by setting H'0, and the highest priority level, level 7, by setting H'7.

When interrupt requests are generated, the highest-priority interrupt according to the priority levels set in the IPR registers is selected. This interrupt level is then compared with the interrupt mask level set by the interrupt mask bits (I2 to I0) in the extend register (EXR) in the CPU, and if the priority level of the interrupt is higher than the set mask level, an interrupt request is issued to the CPU.

5.2.3 IRQ Enable Register (IER)

Bit	:	7	6	5	4	3	2	1	0
		IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IER is an 8-bit readable/writable register that controls enabling and disabling of interrupt requests IRQ7 to IRQ0.

IER is initialized to H'00 by a reset and in hardware standby mode.

It is not initialized in software standby mode.

Bits 7 to 0—IRQ7 to IRQ0 Enable (IRQ7E to IRQ0E): These bits select whether IRQ7 to IRQ0 are enabled or disabled.

Bit n		
IRQnE	**Description**	
0	IRQn interrupts disabled	(Initial value)
1	IRQn interrupts enabled	

(n = 7 to 0)

5.2.4 IRQ Sense Control Registers H and L (ISCRH, ISCRL)

ISCRH

Bit	:	15	14	13	12	11	10	9	8
		IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ISCRL

Bit	:	7	6	5	4	3	2	1	0
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The ISCR registers are 16-bit readable/writable registers that select rising edge, falling edge, or both edge detection, or level sensing, for the input at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

The ISCR registers are initialized to H'0000 by a reset and in hardware standby mode.

They are not initialized in software standby mode.

Bits 15 to 0: IRQ7 Sense Control A and B (IRQ7SCA, IRQ7SCB) to IRQ0 Sense Control A and B (IRQ0SCA, IRQ0SCB)

Bits 15 to 0

IRQ7SCB to IRQ0SCB	IRQ7SCA to IRQ0SCA	Description
0	0	Interrupt request generated at $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input low level (initial value)
	1	Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input
1	0	Interrupt request generated at rising edge of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input
	1	Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input

5.2.5 IRQ Status Register (ISR)

Bit	:	7	6	5	4	3	2	1	0
		IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

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

ISR is an 8-bit readable/writable register that indicates the status of IRQ7 to IRQ0 interrupt requests.

ISR is initialized to H'00 by a reset and in hardware standby mode.

It is not initialized in software standby mode.

Bits 7 to 0—IRQ7 to IRQ0 flags (IRQ7F to IRQ0F): These bits indicate the status of IRQ7 to IRQ0 interrupt requests.

Bit n **IRQnF**	**Description**
0	[Clearing conditions] (Initial value) • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed when low-level detection is set (IRQnSCB = IRQnSCA = 0) and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set (IRQnSCB = 1 or IRQnSCA = 1) • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0
1	[Setting conditions] • When $\overline{\text{IRQn}}$ input goes low when low-level detection is set (IRQnSCB = IRQnSCA = 0) • When a falling edge occurs in $\overline{\text{IRQn}}$ input when falling edge detection is set (IRQnSCB = 0, IRQnSCA = 1) • When a rising edge occurs in $\overline{\text{IRQn}}$ input when rising edge detection is set (IRQnSCB = 1, IRQnSCA = 0) • When a falling or rising edge occurs in $\overline{\text{IRQn}}$ input when both-edge detection is set (IRQnSCB = IRQnSCA = 1)

(n = 7 to 0)

5.3 Interrupt Sources

Interrupt sources comprise external interrupts (NMI and IRQ7 to IRQ0) and internal interrupts (53 sources).

5.3.1 External Interrupts

There are nine external interrupts: NMI and IRQ7 to IRQ0. Of these, NMI and IRQ2 to IRQ0 can be used to restore the H8S/2214 from software standby mode.

NMI Interrupt: 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 SYSCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

The vector number for NMI interrupt exception handling is 7.

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

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.
- 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.

A block diagram of interrupts IRQ7 to IRQ0 is shown in figure 5-2.

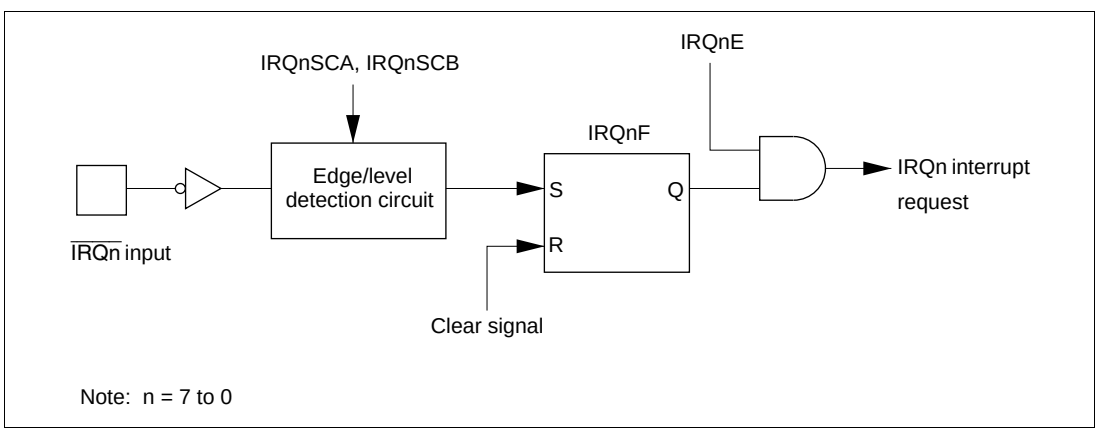


Figure 5-2 Block Diagram of Interrupts IRQ7 to IRQ0

Figure 5-3 shows the timing of setting IRQnF.

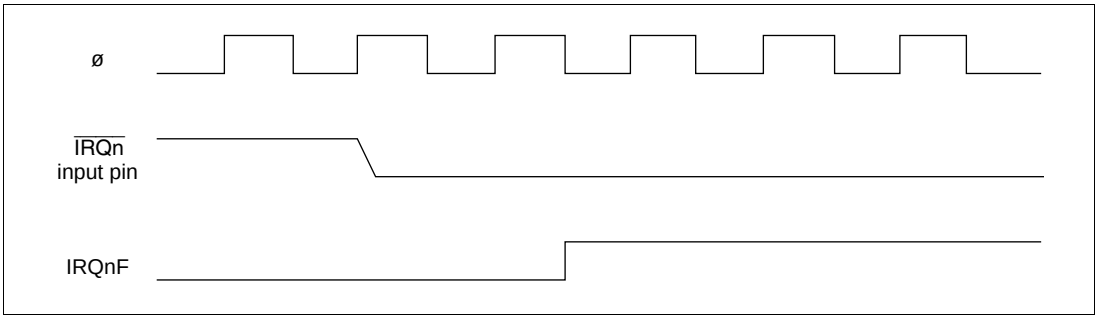


Figure 5-3 Timing of Setting IRQnF

The vector numbers for IRQ7 to IRQ0 interrupt exception handling are 23 to 16.

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. Since interrupt request flags IRQ7F to IRQ0F are set when the setting condition is satisfied, regardless of the IER setting, only the necessary flags should be referenced.

EXIRQ7 to EXIRQ0 Interrupts: Interrupts EXIRQ7 to EXIRQ0 are for use by external expansion modules. An interrupt is requested by a low-level input signal at one of pins EXIRQ7 to EXIRQ0.

5.3.2 Internal Interrupts

There are 31 sources for internal interrupts from on-chip supporting modules.

- For each on-chip supporting module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. If both of these are set to 1 for a particular interrupt source, 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, 8-bit timer, SCI, or other interrupt request. When the DMAC and DTC is activated by an interrupt, the interrupt control mode and interrupt mask bits are not affected.

5.3.3 Interrupt Exception Handling Vector Table

Table 5-4 shows interrupt exception handling sources, vector addresses, and interrupt priorities. For default priorities, the lower the vector number, the higher the priority.

Priorities among modules can be set by means of the IPR. The situation when two or more modules are set to the same priority, and priorities within a module, are fixed as shown in table 5-4.

Table 5-4 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
NMI	External pin	7	H'001C		High
IRQ0		16	H'0040	IPRA6 to IPRA 4	
IRQ1		17	H'0044	IPRA2 to IPRA 0	
IRQ2 IRQ3		18 19	H'0048 H'004C	IPRB6 to IPRB 4	
IRQ4 IRQ5		20 21	H'0050 H'0054	IPRB2 to IPRB 0	
IRQ6 IRQ7		22 23	H'0058 H'005C	IPRC6 to IPRC 4	
SWDTEND (software activation interrupt end)	DTC	24	H'0060	IPRC2 to IPRC 0	
WOVI0 (interval timer)	Watchdog timer 0	25	H'0064	IPRD6 to IPRD 4	
TGI0A (TGR0A input capture/compare match) TGI0B (TGR0B input capture/compare match) TGI0C (TGR0C input capture/compare match) TGI0D (TGR0D input capture/compare match) TCI0V (overflow 0)	TPU channel 0	32 33 34 35 36	H'0080 H'0084 H'0088 H'008C H'0090	IPRF6 to IPRF 4	
Reserved	—	37 38 39	H'0094 H'0098 H'009C		Low

Note: * Lower 16 bits of the start address.

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address* Advanced Mode	IPR	Priority
TGI1A (TGR1A input capture/compare match)	TPU channel 1	40	H'00A0	IPRF2 to IPRF 0	High
TGI1B (TGR1B input capture/compare match)		41	H'00A4		
TCI1V (overflow 1)		42	H'00A8		
TCI1U (underflow 1)		43	H'00AC		
TGI2A (TGR2A input capture/compare match)	TPU channel 2	44	H'00B0	IPRG6 to IPRG 4	
TGI2B (TGR2B input capture/compare match)		45	H'00B4		
TCI2V (overflow 2)		46	H'00B8		
TCI2U (underflow 2)		47	H'00BC		
DEND0A (channel 0/channel 0A transfer end)	DMAC	72	H'0120	IPRJ6 to IPRJ4	
DEND0B (channel 0B transfer end)		73	H'0124		
DEND1A (channel 1/channel 1A transfer end)		74	H'0128		
DEND1B (channel 1B transfer end)		75	H'012C		
ERI0 (receive error 0)	SCI channel 0	80	H'0140	IPRJ2 to IPRJ 0	
RXI0 (reception completed 0)		81	H'0144		
TXI0 (transmit data empty 0)		82	H'0148		
TEI0 (transmission end 0)		83	H'014C		
ERI1 (receive error 1)	SCI channel 1	84	H'0150	IPRK6 to IPRK 4	
RXI1 (reception completed 1)		85	H'0154		
TXI1 (transmit data empty 1)		86	H'0158		
TEI1 (transmission end 1)		87	H'015C		
ERI2 (receive error 2)	SCI channel 2	88	H'0160	IPRK2 to IPRK 0	
RXI2 (reception completed 2)		89	H'0164		
TXI2 (transmit data empty 2)		90	H'0168		
TEI2 (transmission end 2)		91	H'016C		
EXIRQ0	External module	104	H'01A0	IPRM6 to IPRM4	
EXIRQ1		105	H'01A4		
EXIRQ2		106	H'01A8		
EXIRQ3		107	H'01AC		
EXIRQ4		108	H'01B0	IPRM2 to IPRM0	
EXIRQ5		109	H'01B4		
EXIRQ6		110	H'01B8		
EXIRQ7		111	H'01DC		Low

Note: * Lower 16 bits of the start address.

5.4 Interrupt Operation

5.4.1 Interrupt Control Modes and Interrupt Operation

Interrupt operations in the H8S/2214 differ depending on the interrupt control mode.

NMI interrupts are accepted at all times except in the reset state and the hardware standby state. In the case of IRQ interrupts and on-chip supporting module interrupts, an enable bit is provided for each interrupt. Clearing an enable bit to 0 disables the corresponding interrupt request. Interrupt sources for which the enable bits are set to 1 are controlled by the interrupt controller.

Table 5-5 shows the interrupt control modes.

The interrupt controller performs interrupt control according to the interrupt control mode set by the INTM1 and INTM0 bits in SYSCR, the priorities set in IPR, and the masking state indicated by the I and UI bits in the CPU's CCR, and bits I2 to I0 in EXR.

Table 5-5 Interrupt Control Modes

Interrupt Control Mode	SYSCR		Priority Setting Registers	Interrupt Mask Bits	Description
	INTM1	INTM0			
0	0	0	—	I	Interrupt mask control is performed by the I bit.
—		1	—	—	Setting prohibited
2	1	0	IPR	I2 to I0	8-level interrupt mask control is performed by bits I2 to I0. 8 priority levels can be set with IPR.
—		1	—	—	Setting prohibited

Figure 5-4 shows a block diagram of the priority decision circuit.

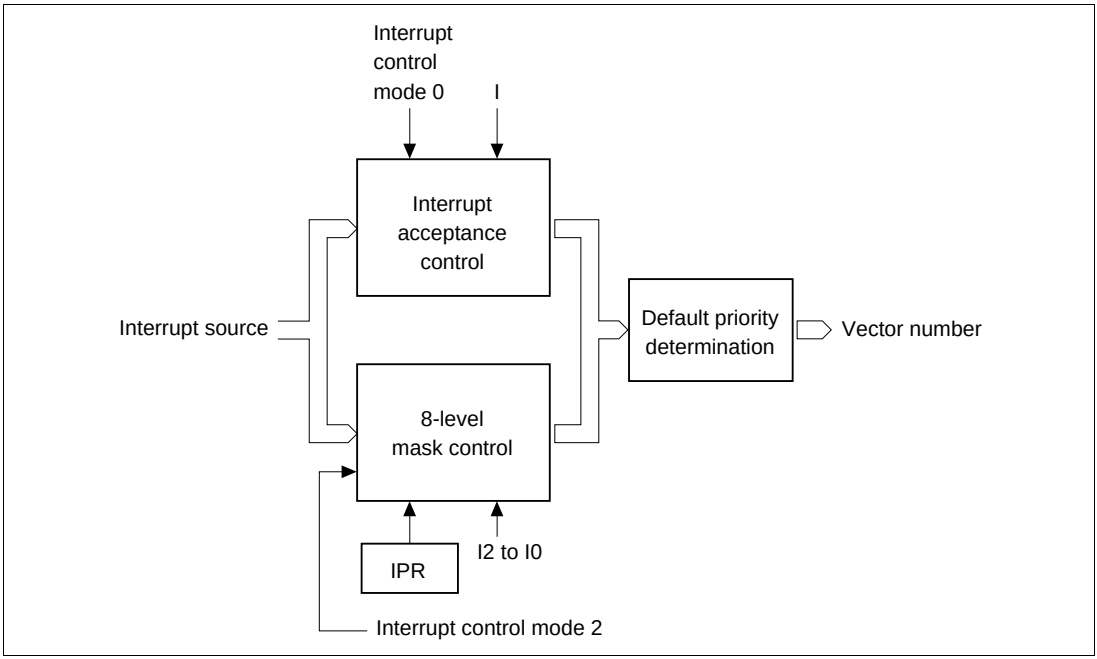


Figure 5-4 Block Diagram of Interrupt Control Operation

(1) Interrupt Acceptance Control

In interrupt control mode 0, interrupt acceptance is controlled by the I bit in CCR.

Table 5-6 shows the interrupts selected in each interrupt control mode.

Table 5-6 Interrupts Selected in Each Interrupt Control Mode (1)

	Interrupt Mask Bits	
Interrupt Control Mode	**I**	**Selected Interrupts**
0	0	All interrupts
	1	NMI interrupts
2	*	All interrupts

* : Don't care

(2) 8-Level Control

In interrupt control mode 2, 8-level mask level determination is performed for the selected interrupts in interrupt acceptance control according to the interrupt priority level (IPR).

The interrupt source selected is the interrupt with the highest priority level, and whose priority level set in IPR is higher than the mask level.

Table 5-7 Interrupts Selected in Each Interrupt Control Mode (2)

Interrupt Control Mode	Selected Interrupts
0	All interrupts
2	Highest-priority-level (IPR) interrupt whose priority level is greater than the mask level (IPR > I2 to I0).

(3) Default Priority Determination

When an interrupt is selected by 8-level control, its priority is determined and a vector number is generated.

If the same value is set for IPR, acceptance of multiple interrupts is enabled, and so only the interrupt source with the highest priority according to the preset default priorities is selected and has a vector number generated.

Interrupt sources with a lower priority than the accepted interrupt source are held pending.

Table 5-8 shows operations and control signal functions in each interrupt control mode.

Table 5-8 Operations and Control Signal Functions in Each Interrupt Control Mode

Interrupt Control	Setting		Interrupt Acceptance Control		8-Level Control			Default Priority	T
Mode	INTM1	INTM0		I		I2 to I0	IPR	Determination	(Trace)
0	0	0	○	IM	X	—	—*2	○	—
2	1	0	X	—*1	○	IM	PR	○	T

Legend

○ : Interrupt operation control performed
X : No operation. (All interrupts enabled)
IM : Used as interrupt mask bit
PR : Sets priority.
— : Not used.

Notes: *1 Set to 1 when interrupt is accepted.
*2 Keep the initial setting.

5.4.2 Interrupt Control Mode 0

Enabling and disabling of IRQ interrupts and on-chip supporting module interrupts can be set by means of the I bit in the CPU's CCR. Interrupts are enabled when the I bit is cleared to 0, and disabled when set to 1.

Figure 5-5 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] The I bit is then referenced. If the I bit is cleared to 0, the interrupt request is accepted. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending.

[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 an interrupt request is accepted, interrupt exception handling starts 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.

[7] A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.

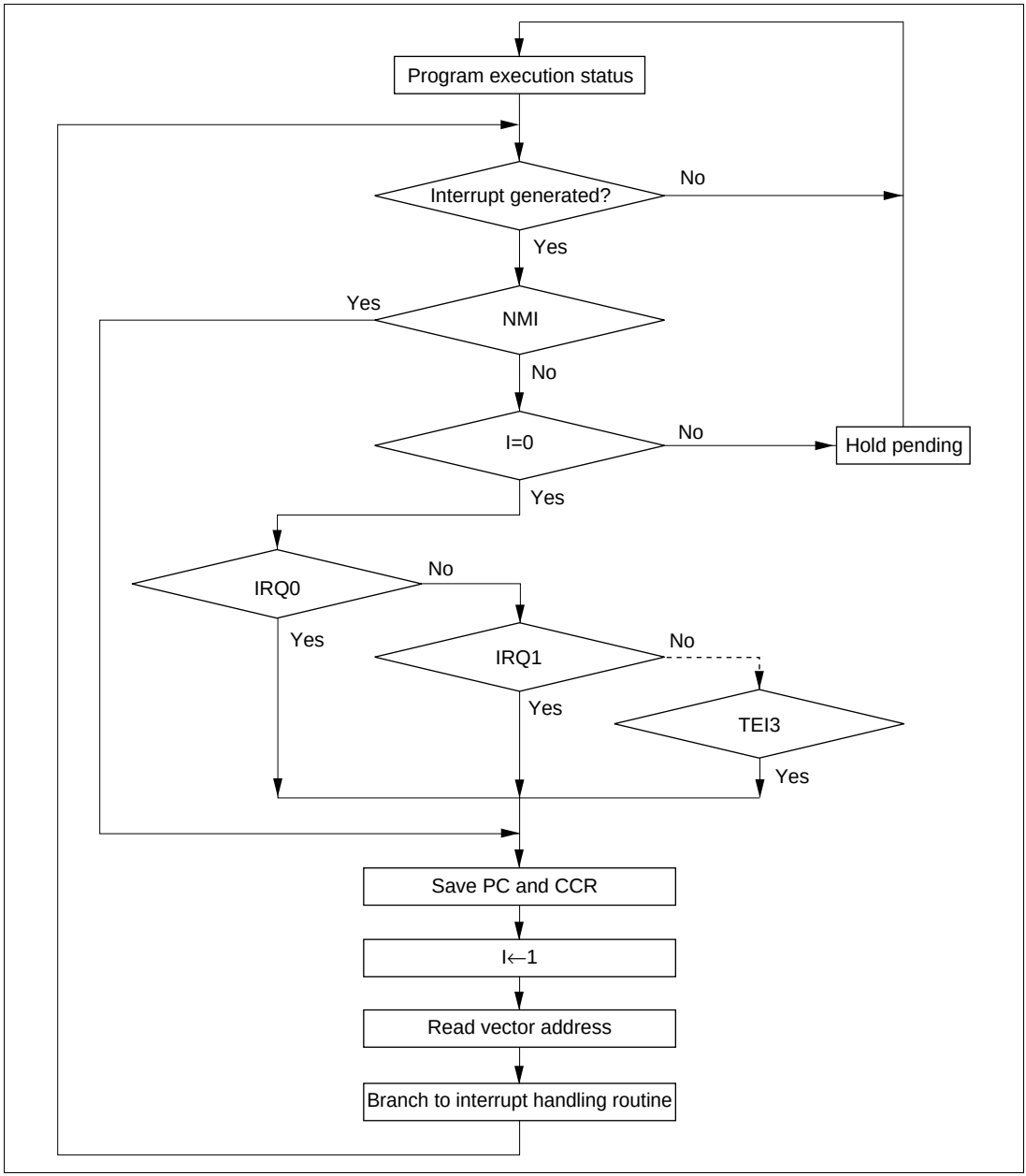


Figure 5-5 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

5.4.3 Interrupt Control Mode 2

Eight-level masking is implemented for IRQ interrupts and on-chip supporting module interrupts by comparing the interrupt mask level set by bits I2 to I0 of EXR in the CPU with IPR.

Figure 5-6 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 5-4 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 an interrupt request is accepted, interrupt exception handling starts 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] A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.

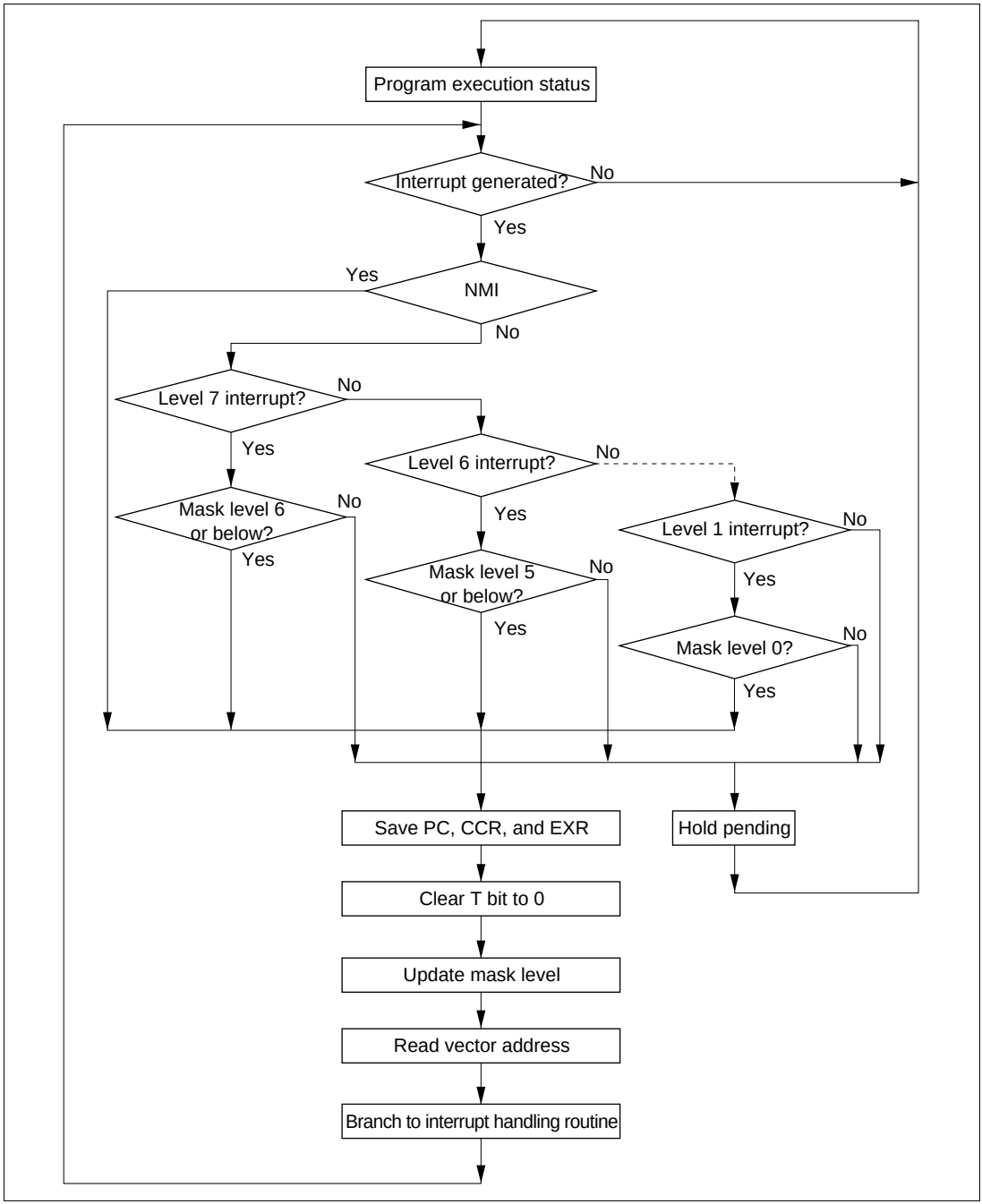


Figure 5-6 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

5.4.4 Interrupt Exception Handling Sequence

Figure 5-7 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.

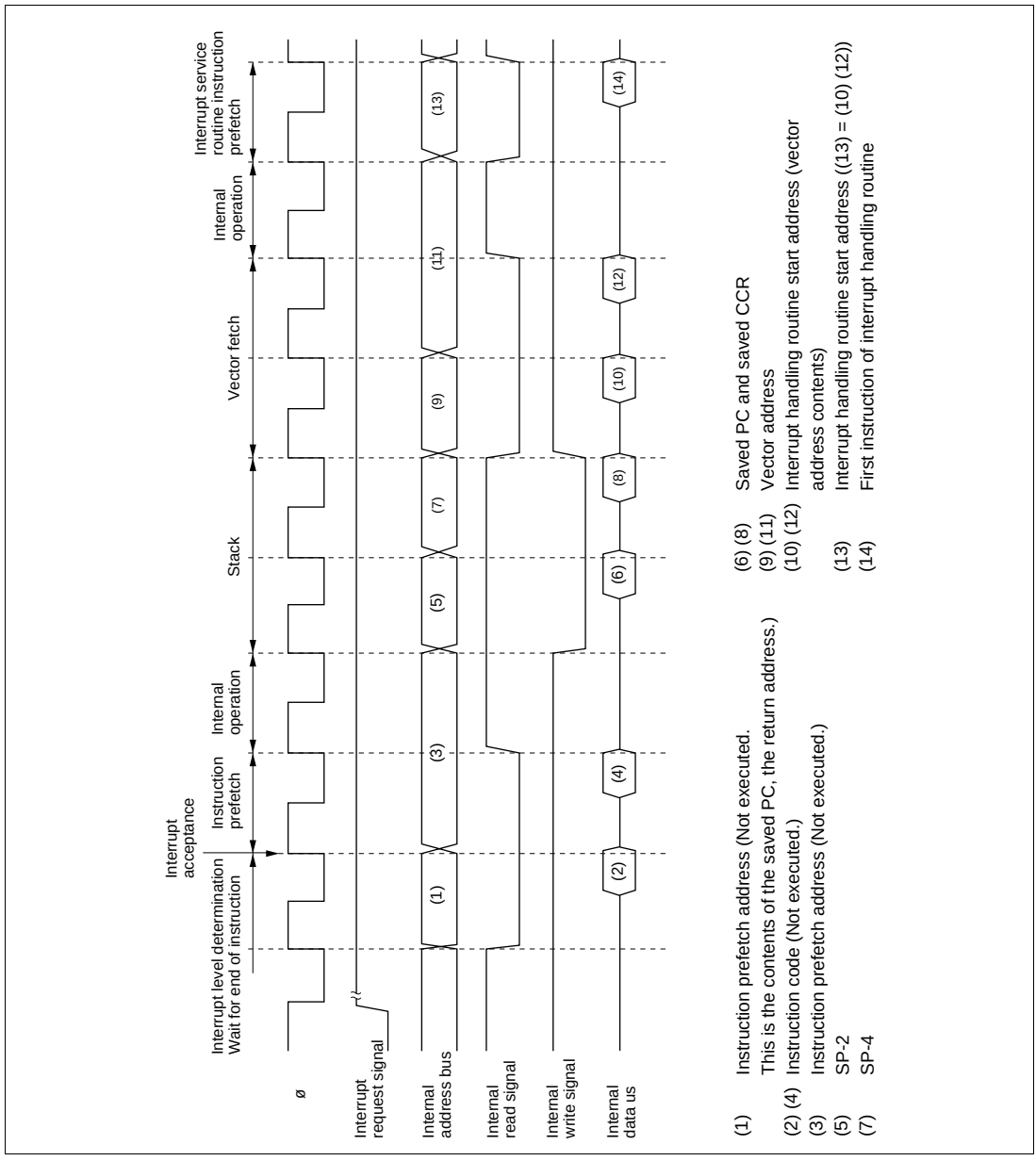


Figure 5-7 Interrupt Exception Handling

5.4.5 Interrupt Response Times

The H8S/2214 is capable of fast word transfer instruction to on-chip memory, and the program area is provided in on-chip ROM and the stack area in on-chip RAM, enabling high-speed processing.

Table 5-9 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 5-9 are explained in table 5-10.

Table 5-9 Interrupt Response Times

		Normal Mode*5		Advanced Mode	
No.	**Execution Status**	**INTM1 = 0**	**INTM1 = 1**	**INTM1 = 0**	**INTM1 = 1**
1	Interrupt priority determination*1	3	3	3	3
2	Number of wait states until executing instruction ends*2	(1 to 19) + 2·S_I	(1 to 19) + 2·S_I	(1 to 19) + 2·S_I	(1 to 19) + 2·S_I
3	PC, CCR, EXR stack save	2·S_K	3·S_K	2·S_K	3·S_K
4	Vector fetch	S_I	S_I	2·S_I	2·S_I
5	Instruction fetch*3	2·S_I	2·S_I	2·S_I	2·S_I
6	Internal processing*4	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 the H8S/2214.

Table 5-10 Number of States in Interrupt Handling Routine Execution Statuses

		Object of Access				
			External Device			
			8 Bit Bus		16 Bit Bus	
Symbol		**Internal Memory**	**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					

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

5.5 Usage Notes

5.5.1 Contention between Interrupt Generation and Disabling

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

In other words, 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 5-8 shows and example in which the TGIEA bit in 16-bit timer TIER0 is cleared to 0.

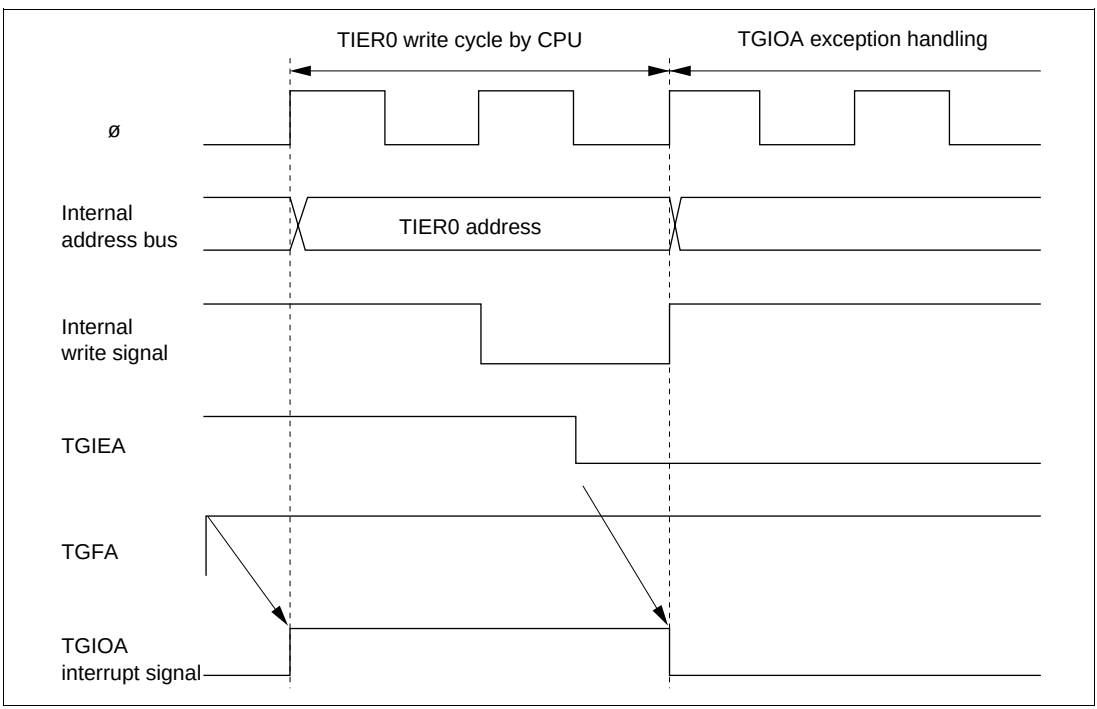


Figure 5-8 Contention between Interrupt Generation and Disabling

The above contention will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

5.5.2 Instructions that Disable Interrupts

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.

5.5.3 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.

5.5.4 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 move 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
```

5.6 DTC and DMAC Activation by Interrupt

5.6.1 Overview

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 with to activate the DTC and DMAC, see section 8, Data Transfer Controller and section 7, DMA Controller.

5.6.2 Block Diagram

Figure 5-9 shows a block diagram of the DTC interrupt controller.

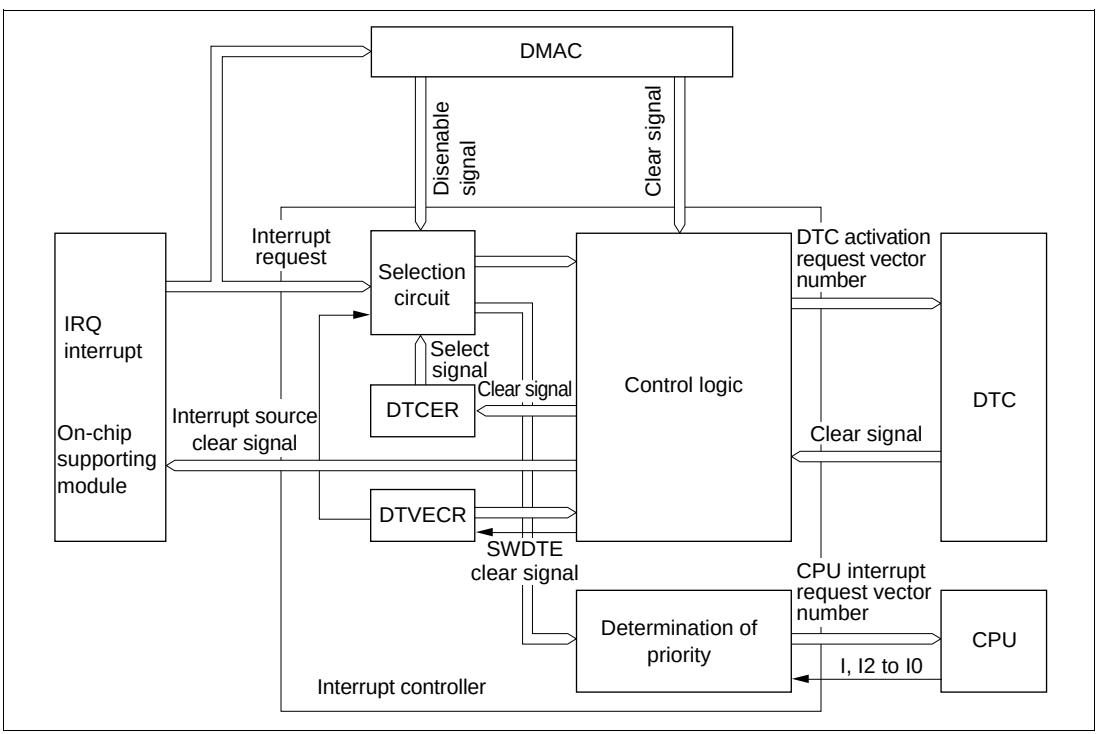


Figure 5-9 Interrupt Control for DTC and DMAC

5.6.3 Operation

The interrupt controller has three main functions in DTC and DMAC control.

(1) Selection of Interrupt Source: DMAC inputs activation factor directly to each channel. The activation factors for each channel of DMAC are selected by DTF3 to DTF0 bits of DMACR. The DTA bit of DMABCR can be used to select whether the selected activation factors are managed by DMAC. By setting the DTA bit to 1, the interrupt factor which were the activation factor for that DMAC do not act as the DTC activation factor or the CPU interrupt factor.

Interrupt factors other than the interrupts managed by the DMAC are selected as DTC activation request or CPU interrupt request by the DTCEA to DTCEF of DTC and the DTCE bit of DTCEI.

By specifying the DISEL bit of the DTC's MRB, it is possible to clear the DTCE bit to 0 after DTC data transfer, and request a CPU interrupt.

If DTC carries out the designate number of data transfers and the transfer counter reads 0, after DTC data transfer, the DTCE bit is also cleared to 0, and a CPU interrupt requested.

(2) Determination of Priority: The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels. See 8.4 Interrupts and 8.3.3 DTC Vector Table for the respective priority.

(3) Operation Order: If the same interrupt is selected as a DTC activation source and a CPU interrupt source, the DTC data transfer is performed first, followed by CPU interrupt exception handling.

If the same interrupt is selected as the DMAC activation factor and as the DTC activation factor or CPU interrupt factor, these operate independently. They operate in accordance with the respective operating states and bus priorities.

Table 5-11 shows the interrupt factor clear control and selection of interrupt factors by specification of the DTA bit of DMAC's DMABCR, DTC's DTCEA to DTCEF, DTCEI's DTCE bits, and the DISEL bit of DTC's MRB.

Table 5-11 Interrupt Source Selection and Clearing Control

Settings					
DMAC	**DTC**		**Interrupt Sources Selection/Clearing Control**		
DTA	**DTCE**	**DISEL**	**DMAC**	**DTC**	**CPU**
0	0	*	○	X	◎
	1	0	○	◎	X
		1	○	○	◎
1	*	*	◎	X	X

Legend

◎ : The relevant interrupt is used. Interrupt source clearing is performed.
(The CPU should clear the source flag in the interrupt handling routine.)

○ : The relevant interrupt is used. The interrupt source is not cleared.

X : The relevant bit cannot be used.

* : Don't care

(4) Notes on Use: The SCI interrupt source is cleared when the DMAC or DTC reads or writes to the prescribed register, and is not dependent upon the DTA bit, DTCE bit, or DISEL bit.

Section 6 Bus Controller

6.1 Overview

The H8S/2214 has a built-in bus controller (BSC) that manages the external address space divided into eight areas. The bus specifications, such as bus width and number of access states, can be set independently for each area, enabling multiple memories to be connected easily.

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

6.1.1 Features

The features of the bus controller are listed below.

- Manages external address space in area units
  - Manages the external space as 8 areas of 2-Mbytes
  - Bus specifications can be set independently for each area
  - Burst ROM interface can be set
- Basic bus interface
  - Chip select ($\overline{CS0}$ 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 for area 0
  - Choice of 1- or 2-state burst access
- Idle cycle insertion
  - An idle cycle can be inserted in case of an external read cycle between different areas
  - An idle cycle can be inserted in case of an external write cycle immediately after an external read cycle
- Bus arbitration function
  - Includes a bus arbiter that arbitrates bus mastership among the CPU, DMAC, and DTC
- Other features
  - External bus release function

6.1.2 Block Diagram

Figure 6-1 shows a block diagram of the bus controller.

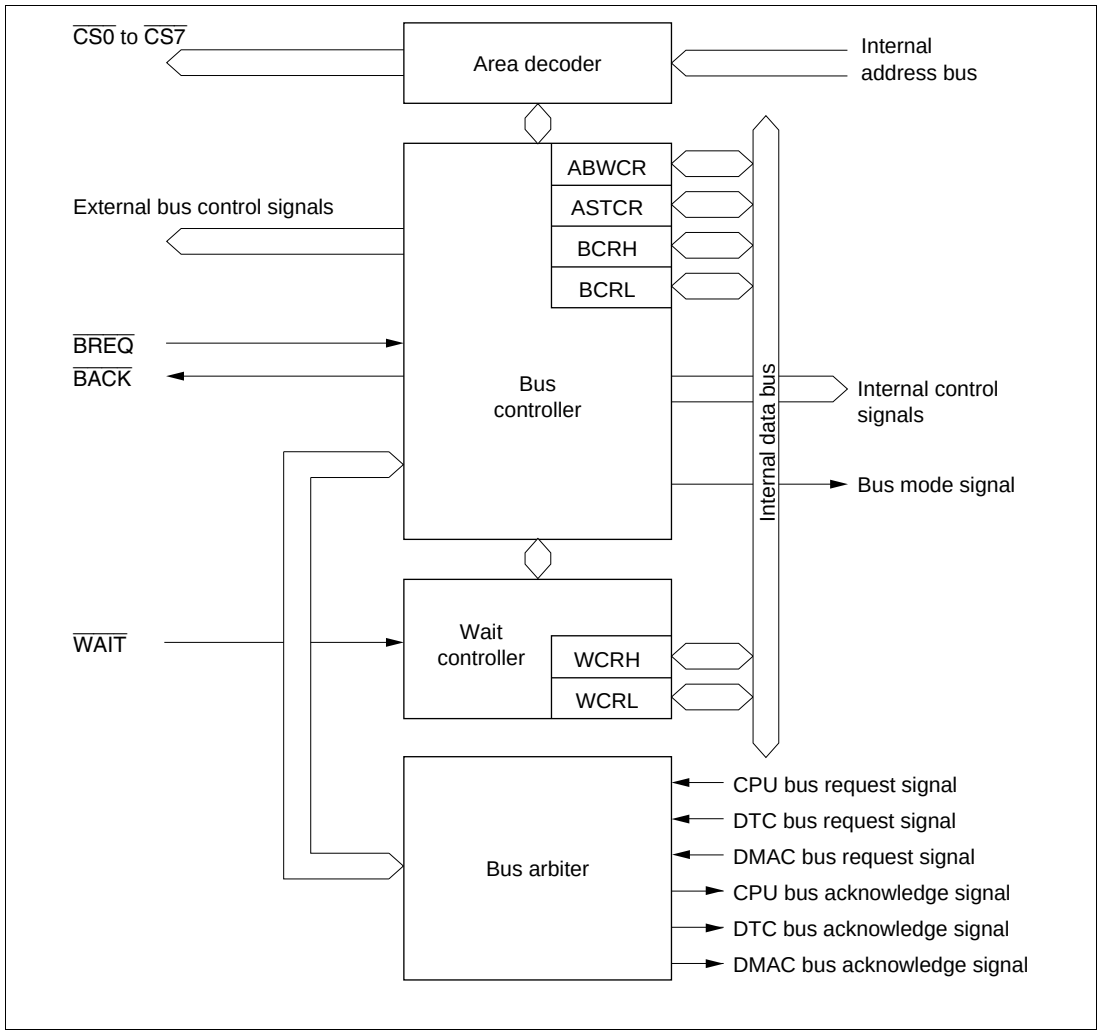


Figure 6-1 Block Diagram of Bus Controller

6.1.3 Pin Configuration

Table 6-1 summarizes the pins of the bus controller.

Table 6-1 Bus Controller Pins

Name	Symbol	I/O	Function
Address strobe	$\overline{AS}$	Output	Strobe signal indicating that address output on address bus is enabled.
Read	$\overline{RD}$	Output	Strobe signal indicating that external space is being read.
High write	$\overline{HWR}$	Output	Strobe signal indicating that external space is to be written, and upper half (D15 to D8) of data bus is enabled.
Low write	$\overline{LWR}$	Output	Strobe signal indicating that external space is to be written, and lower half (D7 to D0) of data bus is enabled.
Chip select 0 to 7	$\overline{CS0}$ to $\overline{CS7}$	Output	Strobe signal indicating that areas 0 to 7 are selected.
Wait	$\overline{WAIT}$	Input	Wait request signal when accessing external 3-state access space.
Bus request	$\overline{BREQ}$	Input	Request signal that releases bus to external device.
Bus request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating that bus has been released.

6.1.4 Register Configuration

Table 6-2 summarizes the registers of the bus controller.

Table 6-2 Bus Controller Registers

Name	Abbreviation	R/W	Initial Value: Power-On Reset	Initial Value: Manual Reset	Address*[1]
Bus width control register	ABWCR	R/W	H'FF/H'00*[2]	Retained	H'FED0
Access state control register	ASTCR	R/W	H'FF	Retained	H'FED1
Wait control register H	WCRH	R/W	H'FF	Retained	H'FED2
Wait control register L	WCRL	R/W	H'FF	Retained	H'FED3
Bus control register H	BCRH	R/W	H'D0	Retained	H'FED4
Bus control register L	BCRL	R/W	H'08	Retained	H'FED5
Pin function control register	PFCR	R/W	H'0D/H'00*[3]	Retained	H'FDEB

Notes: *1 Lower 16 bits of the address.

*2 Determined by the MCU operating mode. Initialized to H'00 in mode 4, and to H'FF in modes 5 to 7.

*3 Initialized to H'0D in modes 4 and 5, and to H'00 in modes 6 and 7.

6.2 Register Descriptions

6.2.1 Bus Width Control Register (ABWCR)

Bit	:	7	6	5	4	3	2	1	0
		ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
Modes 5 to 7									
Initial value	:	1	1	1	1	1	1	1	1
RW	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Mode 4									
Initial value	:	0	0	0	0	0	0	0	0
RW	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ABWCR is an 8-bit readable/writable register that designates each area for either 8-bit access or 16-bit access.

ABWCR sets the data bus width for the external memory space. The bus width for on-chip memory and internal I/O registers is fixed regardless of the settings in ABWCR.

After a power-on reset and in hardware standby mode, ABWCR is initialized to H'FF in modes 5, 6, and 7, and to H'00 in mode 4. It is not initialized by a manual reset or in software standby mode.

Bits 7 to 0—Area 7 to 0 Bus Width Control (ABW7 to ABW0): These bits select whether the corresponding area is to be designated for 8-bit access or 16-bit access.

Bit n	
ABWn	**Description**
0	Area n is designated for 16-bit access
1	Area n is designated for 8-bit access

(n = 7 to 0)

6.2.2 Access State Control Register (ASTCR)

Bit	:	7	6	5	4	3	2	1	0
		AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ASTCR is an 8-bit readable/writable register that designates each area as either a 2-state access space or a 3-state access space.

ASTCR sets the number of access states for the external memory space. The number of access states for on-chip memory and internal I/O registers is fixed regardless of the settings in ASTCR.

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

Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select whether the corresponding area is to be designated as a 2-state access space or a 3-state access space.

Wait state insertion is enabled or disabled at the same time.

Bit n		
ASTn	**Description**	
0	Area n is designated for 2-state access Wait state insertion in area n external space is disabled	
1	Area n is designated for 3-state access Wait state insertion in area n external space is enabled	(Initial value)

(n = 7 to 0)

6.2.3 Wait Control Registers H and L (WCRH, WCRL)

WCRH and WCRL are 8-bit readable/writable registers that select the number of program wait states for each area.

Program waits are not inserted in the case of on-chip memory or internal I/O registers.

WCRH and WCRL are initialized to H'FF by a power-on reset and in hardware standby mode. They are not initialized by a manual reset or in software standby mode.

(1) WCRH

Bit	:	7	6	5	4	3	2	1	0
		W71	W70	W61	W60	W51	W50	W41	W40
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Area 7 Wait Control 1 and 0 (W71, W70): These bits select the number of program wait states when area 7 in external space is accessed while the AST7 bit in ASTCR is set to 1.

Bit 7	Bit 6	
W71	**W70**	**Description**
0	0	Program wait not inserted when external space area 7 is accessed
	1	1 program wait state inserted when external space area 7 is accessed
1	0	2 program wait states inserted when external space area 7 is accessed
	1	3 program wait states inserted when external space area 7 is accessed (Initial value)

Bits 5 and 4—Area 6 Wait Control 1 and 0 (W61, W60): These bits select the number of program wait states when area 6 in external space is accessed while the AST6 bit in ASTCR is set to 1.

Bit 5	Bit 4	
W61	**W60**	**Description**
0	0	Program wait not inserted when external space area 6 is accessed
	1	1 program wait state inserted when external space area 6 is accessed
1	0	2 program wait states inserted when external space area 6 is accessed
	1	3 program wait states inserted when external space area 6 is accessed (Initial value)

Bits 3 and 2—Area 5 Wait Control 1 and 0 (W51, W50): These bits select the number of program wait states when area 5 in external space is accessed while the AST5 bit in ASTCR is set to 1.

Bit 3	Bit 2	
W51	**W50**	**Description**
0	0	Program wait not inserted when external space area 5 is accessed
	1	1 program wait state inserted when external space area 5 is accessed
1	0	2 program wait states inserted when external space area 5 is accessed
	1	3 program wait states inserted when external space area 5 is accessed (Initial value)

Bits 1 and 0—Area 4 Wait Control 1 and 0 (W41, W40): These bits select the number of program wait states when area 4 in external space is accessed while the AST4 bit in ASTCR is set to 1.

Bit 1	Bit 0	
W41	**W40**	**Description**
0	0	Program wait not inserted when external space area 4 is accessed
	1	1 program wait state inserted when external space area 4 is accessed
1	0	2 program wait states inserted when external space area 4 is accessed
	1	3 program wait states inserted when external space area 4 is accessed (Initial value)

(2) WCRL

Bit	:	7	6	5	4	3	2	1	0
		W31	W30	W21	W20	W11	W10	W01	W00
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Area 3 Wait Control 1 and 0 (W31, W30): These bits select the number of program wait states when area 3 in external space is accessed while the AST3 bit in ASTCR is set to 1.

Bit 7	Bit 6	
W31	**W30**	**Description**
0	0	Program wait not inserted when external space area 3 is accessed
	1	1 program wait state inserted when external space area 3 is accessed
1	0	2 program wait states inserted when external space area 3 is accessed
	1	3 program wait states inserted when external space area 3 is accessed (Initial value)

Bits 5 and 4—Area 2 Wait Control 1 and 0 (W21, W20): These bits select the number of program wait states when area 2 in external space is accessed while the AST2 bit in ASTCR is set to 1.

Bit 5	Bit 4	
W21	**W20**	**Description**
0	0	Program wait not inserted when external space area 2 is accessed
	1	1 program wait state inserted when external space area 2 is accessed
1	0	2 program wait states inserted when external space area 2 is accessed
	1	3 program wait states inserted when external space area 2 is accessed (Initial value)

Bits 3 and 2—Area 1 Wait Control 1 and 0 (W11, W10): These bits select the number of program wait states when area 1 in external space is accessed while the AST1 bit in ASTCR is set to 1.

Bit 3	Bit 2	
W11	**W10**	**Description**
0	0	Program wait not inserted when external space area 1 is accessed
	1	1 program wait state inserted when external space area 1 is accessed
1	0	2 program wait states inserted when external space area 1 is accessed
	1	3 program wait states inserted when external space area 1 is accessed (Initial value)

Bits 1 and 0—Area 0 Wait Control 1 and 0 (W01, W00): These bits select the number of program wait states when area 0 in external space is accessed while the AST0 bit in ASTCR is set to 1.

Bit 1	Bit 0	
W01	**W00**	**Description**
0	0	Program wait not inserted when external space area 0 is accessed
	1	1 program wait state inserted when external space area 0 is accessed
1	0	2 program wait states inserted when external space area 0 is accessed
	1	3 program wait states inserted when external space area 0 is accessed (Initial value)

6.2.4 Bus Control Register H (BCRH)

Bit	:	7	6	5	4	3	2	1	0
		ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—
Initial value	:	1	1	0	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCRH is an 8-bit readable/writable register that selects enabling or disabling of idle cycle insertion, and the memory interface for area 0.

BCRH is initialized to H'D0 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

Bit 7—Idle Cycle Insert 1 (ICIS1): Selects whether or not one idle cycle state is to be inserted between bus cycles when successive external read cycles are performed in different areas.

Bit 7

ICIS1	Description
0	Idle cycle not inserted in case of successive external read cycles in different areas
1	Idle cycle inserted in case of successive external read cycles in different areas (Initial value)

Bit 6—Idle Cycle Insert 0 (ICIS0): Selects whether or not one idle cycle state is to be inserted between bus cycles when successive external read and external write cycles are performed .

Bit 6

ICIS0	Description
0	Idle cycle not inserted in case of successive external read and external write cycles
1	Idle cycle inserted in case of successive external read and external write cycles (Initial value)

Bit 5—Burst ROM Enable (BRSTRM): Selects whether area 0 is used as a burst ROM interface.

Bit 5

BRSTRM	Description
0	Area 0 is basic bus interface (Initial value)
1	Area 0 is burst ROM interface

Bit 4—Burst Cycle Select 1 (BRSTS1): Selects the number of burst cycles for the burst ROM interface.

Bit 4

BRSTS1	Description	
0	Burst cycle comprises 1 state	
1	Burst cycle comprises 2 states	(Initial value)

Bit 3—Burst Cycle Select 0 (BRSTS0): Selects the number of words that can be accessed in a burst ROM interface burst access.

Bit 3

BRSTS0	Description	
0	Max. 4 words in burst access	(Initial value)
1	Max. 8 words in burst access	

Bits 2 to 0—Reserved: Only 0 should be written to these bits.

6.2.5 Bus Control Register L (BCRL)

Bit	:	7	6	5	4	3	2	1	0
		BRLE	—	—	—	—	—	—	WAITE
Initial value	:	0	0	0	0	1	0	0	0
R/W	:	R/W	R/W	—	R/W	R/W	R/W	R/W	R/W

BCRL is an 8-bit readable/writable register that performs selection of the external bus-released state protocol, and enabling or disabling of $\overline{\text{WAIT}}$ pin input.

BCRL is initialized to H'08 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

Bit 7—Bus Release Enable (BRLE): Enables or disables external bus release.

Bit 7

BRLE	Description
0	External bus release is disabled. $\overline{\text{BREQ}}$ and $\overline{\text{BACK}}$ can be used as I/O ports. (Initial value)
1	External bus release is enabled.

Bit 6—Reserved: Only 0 should be written to this bit.

Bit 5—Reserved: This bit cannot be modified and is always read as 0.

Bit 4—Reserved: Only 0 should be written to this bit.

Bit 3—Reserved: Only 1 should be written to this bit.

Bits 2 and 1—Reserved: Only 0 should be written to these bits.

Bit 0—WAIT Pin Enable (WAITE): Selects enabling or disabling of wait input by the $\overline{\text{WAIT}}$ pin.

Bit 0

WAITE	Description
0	Wait input by $\overline{\text{WAIT}}$ pin disabled. $\overline{\text{WAIT}}$ pin can be used as I/O port. (Initial value)
1	Wait input by $\overline{\text{WAIT}}$ pin enabled

6.2.6 Pin Function Control Register (PFCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	AE3	AE2	AE1	AE0
Modes 4 and 5									
Initial value	:	0	0	0	0	1	1	0	1
Modes 6 and 7									
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFCR is an 8-bit readable/writable register that performs address output control in external expanded mode.

PFCR is initialized to H'0D (modes 4 and 5) or H'00 (modes 6 and 7) by a power-on reset and in hardware standby mode. It retains its previous state in a manual reset and in software standby mode.

Bits 7 to 4—Reserved: Only 0 should be written to these bits.

Bits 3 to 0—Address Output Enable 3 to 0 (AE3 to AE0): These bits select enabling or disabling of address outputs A8 to A23 in ROMless expanded mode and modes with ROM. When a pin is enabled for address output, the address is output regardless of the corresponding DDR setting. When a pin is disabled for address output, it becomes an output port when the corresponding DDR bit is set to 1.

Bit 3	Bit 2	Bit 1	Bit 0	
AE3	**AE2**	**AE1**	**AE0**	**Description**
0	0	0	0	A8 to A23 output disabled (Initial value*1)
			1	A8 output enabled; A9 to A23 output disabled
		1	0	A8, A9 output enabled; A10 to A23 output disabled
			1	A8 to A10 output enabled; A11 to A23 output disabled
	1	0	0	A8 to A11 output enabled; A12 to A23 output disabled
			1	A8 to A12 output enabled; A13 to A23 output disabled
		1	0	A8 to A13 output enabled; A14 to A23 output disabled
			1	A8 to A14 output enabled; A15 to A23 output disabled
1	0	0	0	A8 to A15 output enabled; A16 to A23 output disabled
			1	A8 to A16 output enabled; A17 to A23 output disabled
		1	0	A8 to A17 output enabled; A18 to A23 output disabled
			1	A8 to A18 output enabled; A19 to A23 output disabled
	1	0	0	A8 to A19 output enabled; A20 to A23 output disabled
			1	A8 to A20 output enabled; A21 to A23 output disabled (Initial value*2)
		1	0	A8 to A21 output enabled; A22, A23 output disabled
			1	A8 to A23 output enabled

Notes: *1 In expanded mode with ROM, bits AE3 to AE0 are initialized to B'0000.
In expanded mode with ROM, address pins A0 to A7 are made address outputs by setting the corresponding DDR bits to 1.

*2 In ROMless expanded mode, bits AE3 to AE0 are initialized to B'1101.
In ROMless expanded mode, address pins A0 to A7 are always made address output.

6.3 Overview of Bus Control

6.3.1 Area Divisions

In advanced mode, the bus controller partitions the 16 Mbytes address space into eight areas, 0 to 7, in 2-Mbyte units, and performs bus control for external space in area units. In normal mode*, it controls a 64-kbyte address space comprising part of area 0 (not available in the H8S/2214). Figure 6-2 shows an outline of the memory map.

Chip select signals ($\overline{CS0}$ to $\overline{CS7}$) can be output for each area.

Note: * Not available in the H8S/2214.

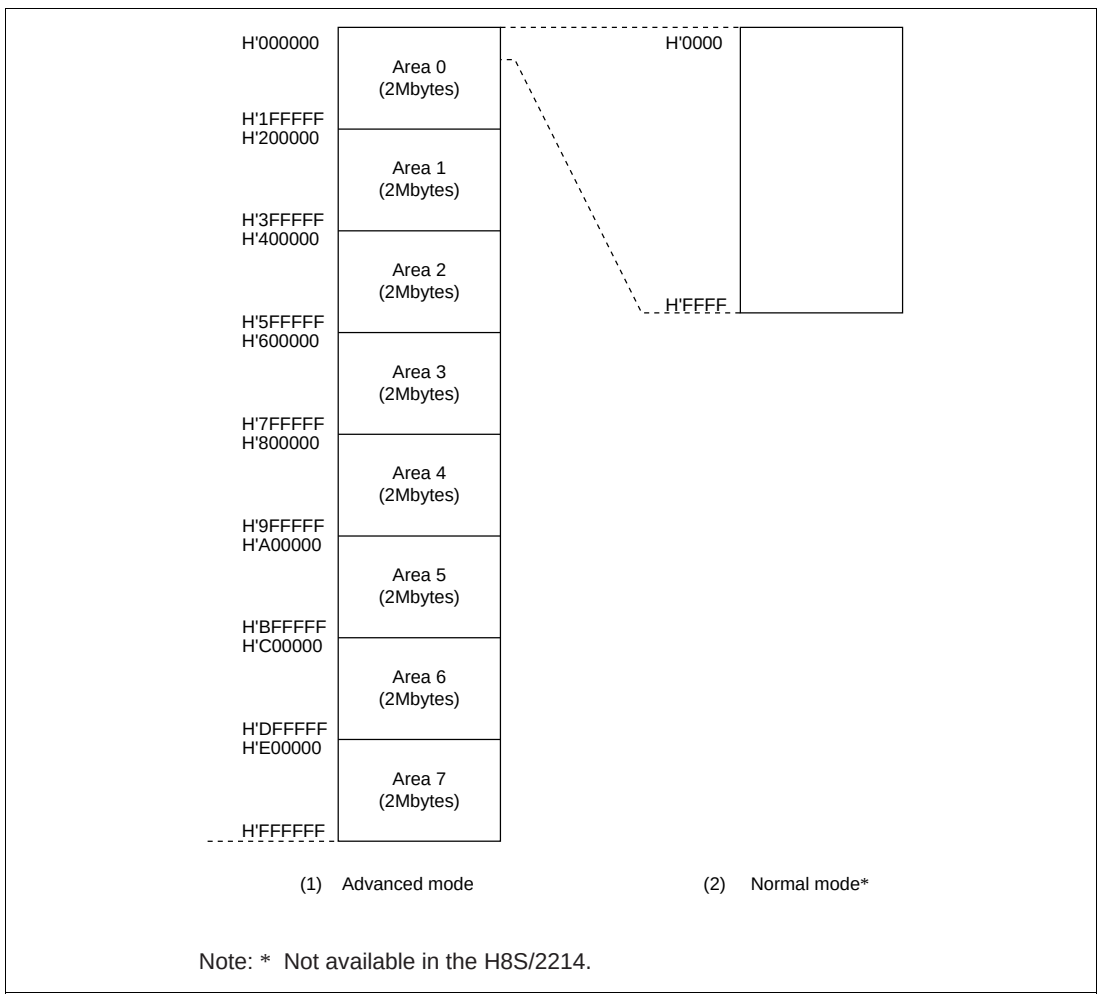


Figure 6-2 Overview of Area Divisions

6.3.2 Bus Specifications

The external space bus specifications consist of three elements: bus width, number of access states, and number of program wait 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.

(1) 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 a16-bit access space.

If all areas are designated for 8-bit access, 8-bit bus mode is set; if any area is designated for 16-bit access, 16-bit bus mode is set. When the burst ROM interface is designated, 16-bit bus mode is always set.

(2) 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 burst ROM interface, the number of access states may be determined without regard to ASTCR.

When 2-state access space is designated, wait insertion is disabled.

(3) 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 WCRH and WCRL. From 0 to 3 program wait states can be selected.

Table 6-3 shows the bus specifications for each basic bus interface area.

Table 6-3 Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WCRH, WCRL		Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn1	Wn0	Bus Width	Access States	Program Wait States
0	0	—	—	16	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3
1	0	—	—	8	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3

6.3.3 Memory Interfaces

The H8S/2214 memory interfaces comprise a basic bus interface that allows direct connection of ROM, SRAM, and so on, and a burst ROM interface (for area 0 only) that allows direct connection of burst ROM.

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

6.3.4 Interface Specifications for Each Area

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. The bus specifications described here cover basic items only, and the sections on each memory interface (6.4 and 6.5) should be referred to for further details.

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

Areas 1 to 6: In external expansion mode, all of areas 1 to 6 is external space.

When area 1 to 6 external space is accessed, the $\overline{CS1}$ to $\overline{CS6}$ pin signals respectively can be output.

Only the basic bus interface can be used for areas 1 to 6.

Area 7: Area 7 includes the on-chip RAM, external module expansion function space, and internal l/O registers. In external expansion mode, the space excluding the on-chip RAM, external module expansion function space, and internal l/O registers, is external space. The on-chip RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space becomes external space.

When the P75MSOE bit in the external module connection output pin select register (OPINSEL) is set to 1, the external module expansion function is enabled and the signal is output for addresses H'FFFF40 to H'FFFF5F. When the P75MSOE bit is cleared to 0, the external module expansion function is disabled and the corresponding addresses are 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.

6.3.5 Chip Select Signals

The H8S/2214 can output chip select signals ($\overline{CS0}$ to $\overline{CS7}$) to areas 0 to 7, the signal being driven low when the corresponding external space area is accessed.

Figure 6-3 shows an example of $\overline{CSn}$ (n = 0 to 7) output timing.

Enabling or disabling of the $\overline{CSn}$ signal is performed by setting the data direction register (DDR) for the port corresponding to the particular $\overline{CSn}$ pin.

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

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

For details, see section 9, I/O Ports.

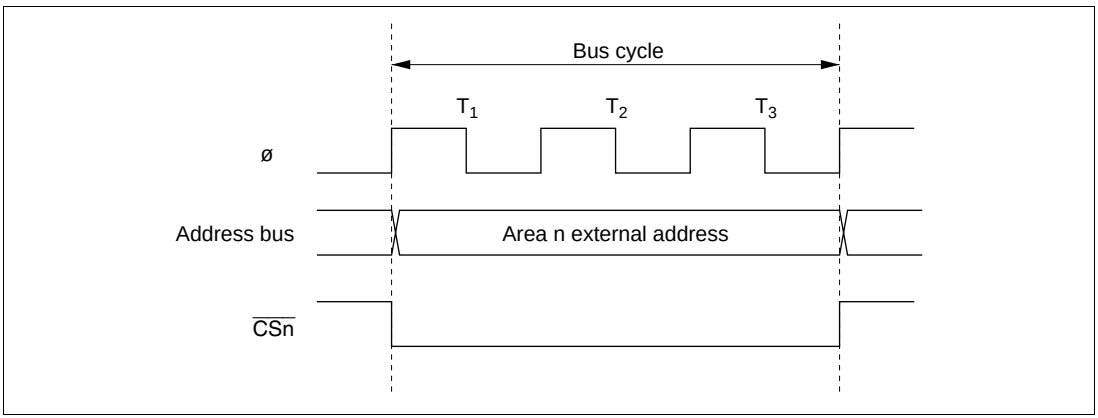


Figure 6-3 $\overline{CSn}$ Signal Output Timing (n = 0 to 7)

6.4 Basic Bus Interface

6.4.1 Overview

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

The bus specifications can be selected with ABWCR, ASTCR, WCRH, and WCRL (see table 6-3).

6.4.2 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 6-4 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 transfer instruction is performed as two byte accesses, and a longword transfer instruction, as four byte accesses.

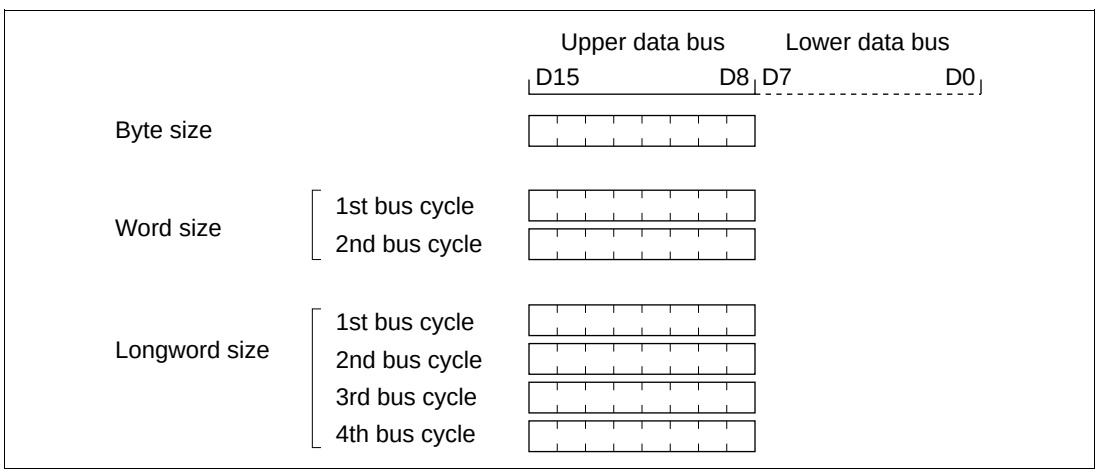


Figure 6-4 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-Bit Access Space: Figure 6-5 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 transfer instruction is executed as two word transfer instructions.

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.

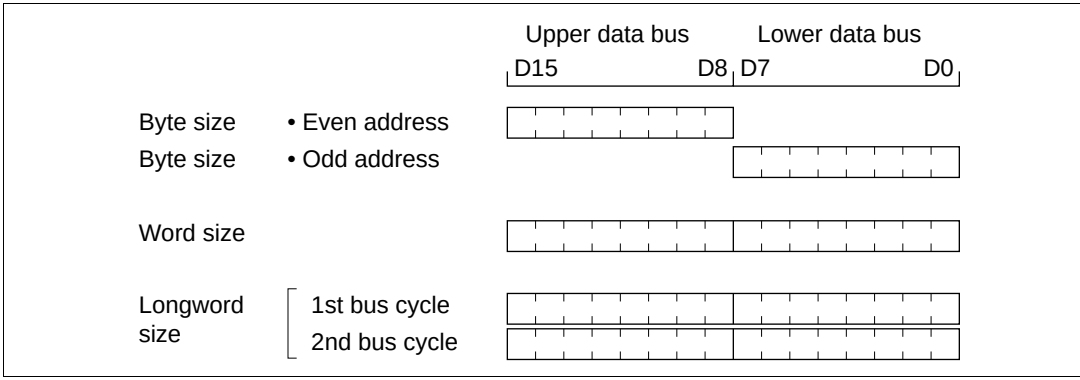


Figure 6-5 Access Sizes and Data Alignment Control (16-Bit Access Space)

6.4.3 Valid Strobes

Table 6-4 shows the data buses used and valid strobes for the access spaces.

In a read, the $\overline{RD}$ signal is valid without discrimination between the upper and lower halves 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 6-4 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

Note: Hi-Z: High impedance.
Invalid: Input state; input value is ignored.

6.4.4 Basic Timing

8-Bit 2-State Access Space: Figure 6-6 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.

Wait states cannot be inserted.

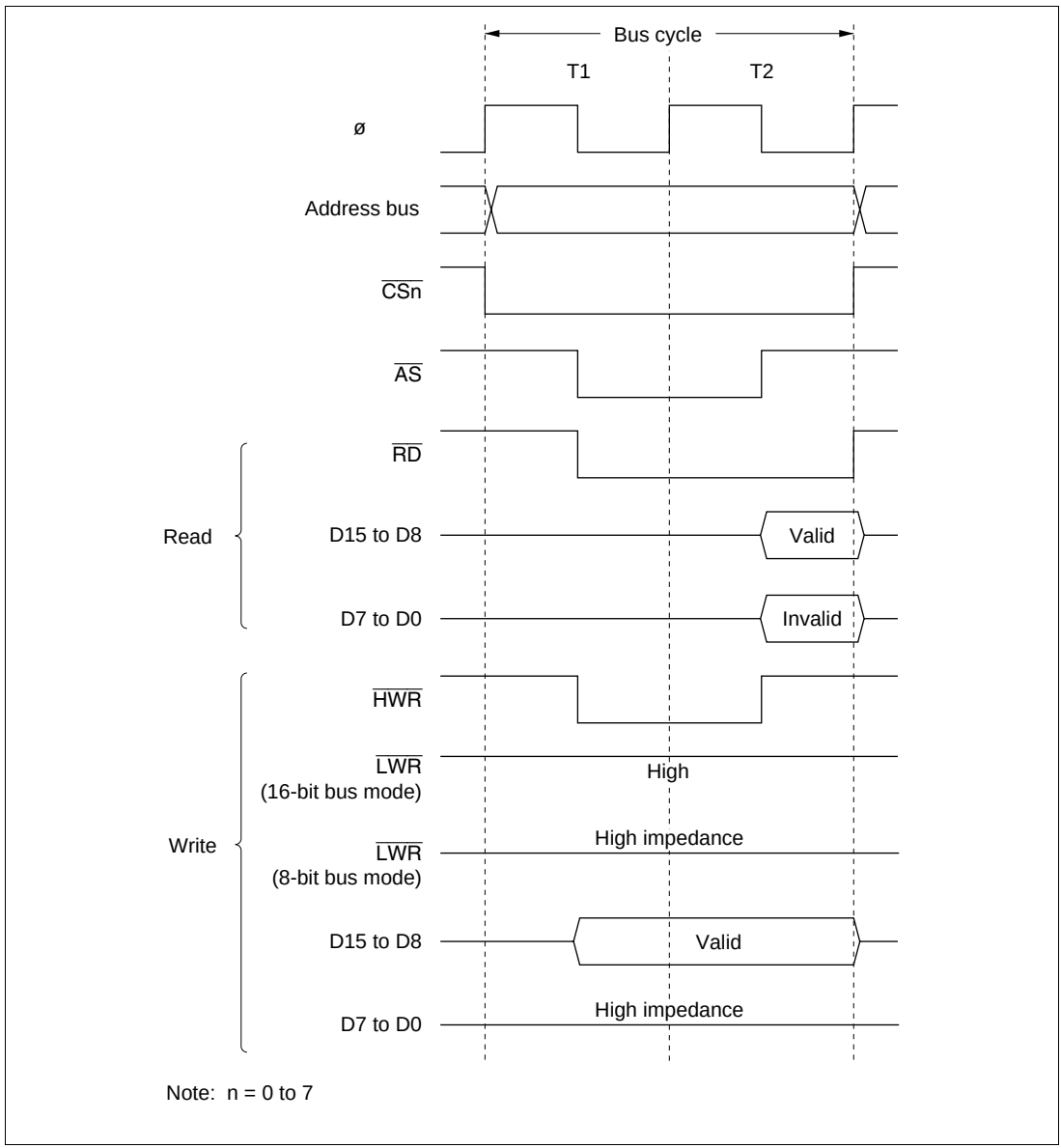


Figure 6-6 Bus Timing for 8-Bit 2-State Access Space

8-Bit 3-State Access Space: Figure 6-7 shows the bus timing for an 8-bit 3-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used.

Wait states can be inserted.

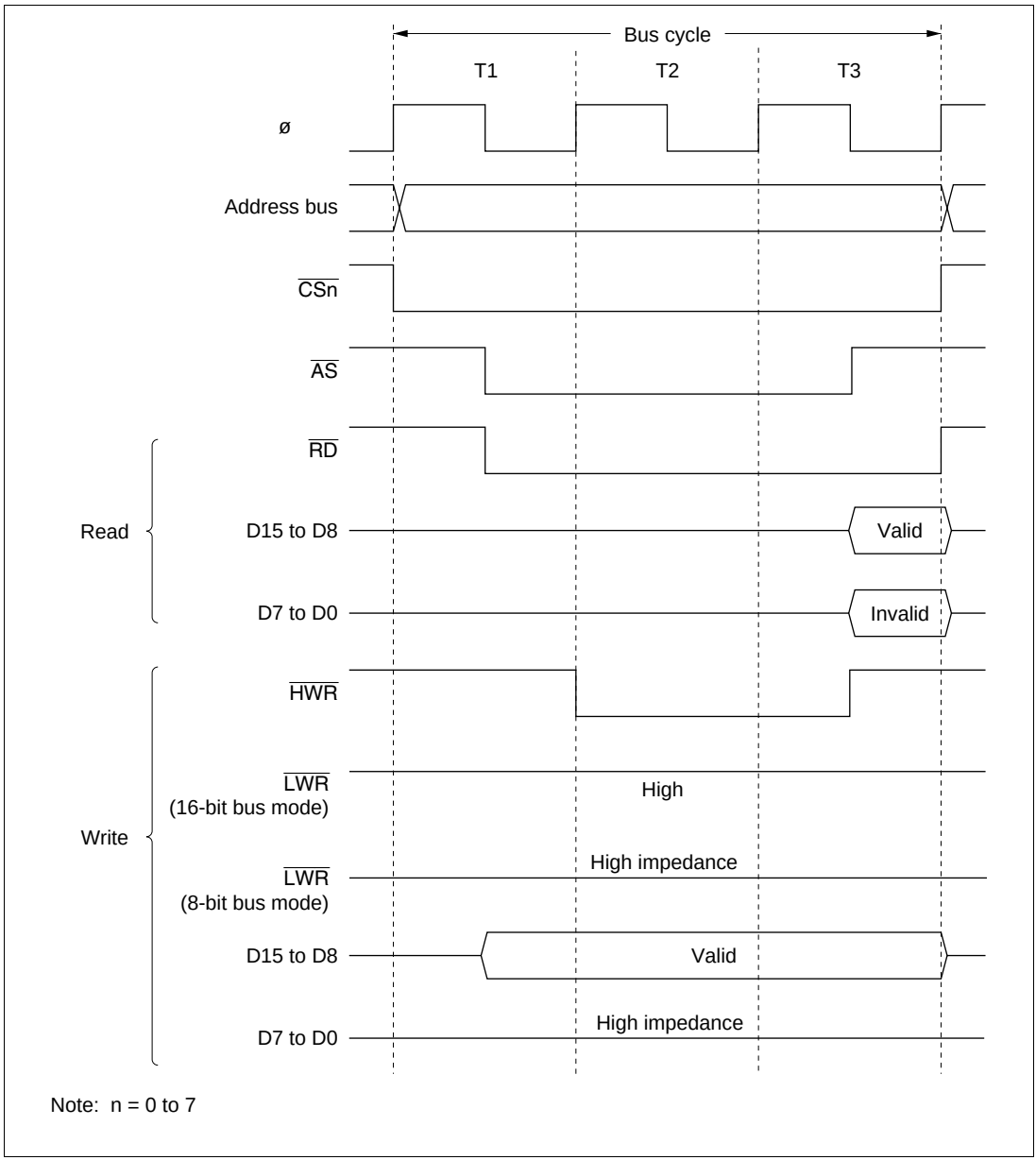


Figure 6-7 Bus Timing for 8-Bit 3-State Access Space

16-Bit 2-State Access Space: Figures 6-8 to 6-10 show bus timings for a 16-bit 2-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for the even address, and the lower half (D7 to D0) for the odd address.

Wait states cannot be inserted.

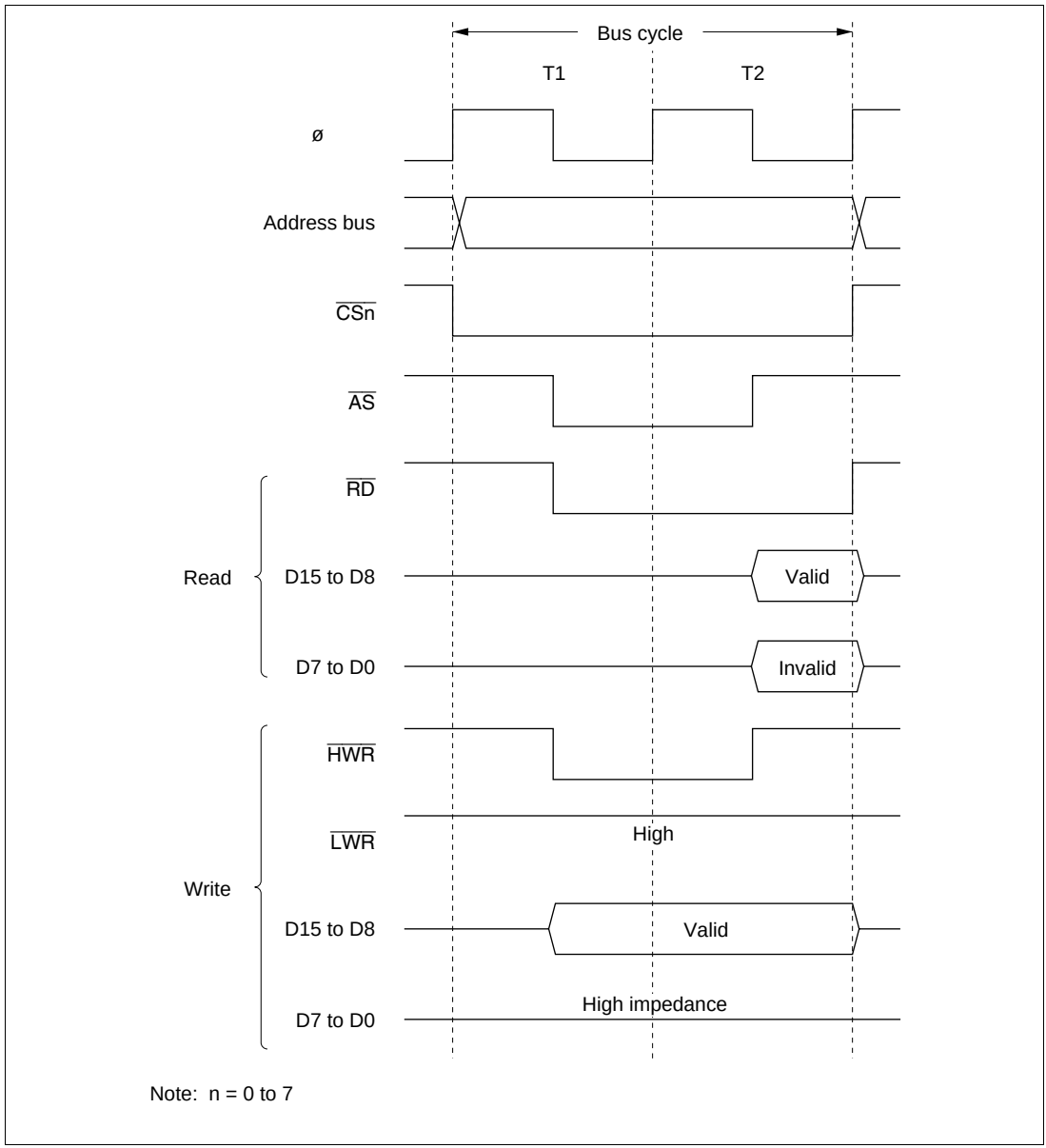


Figure 6-8 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byte Access)

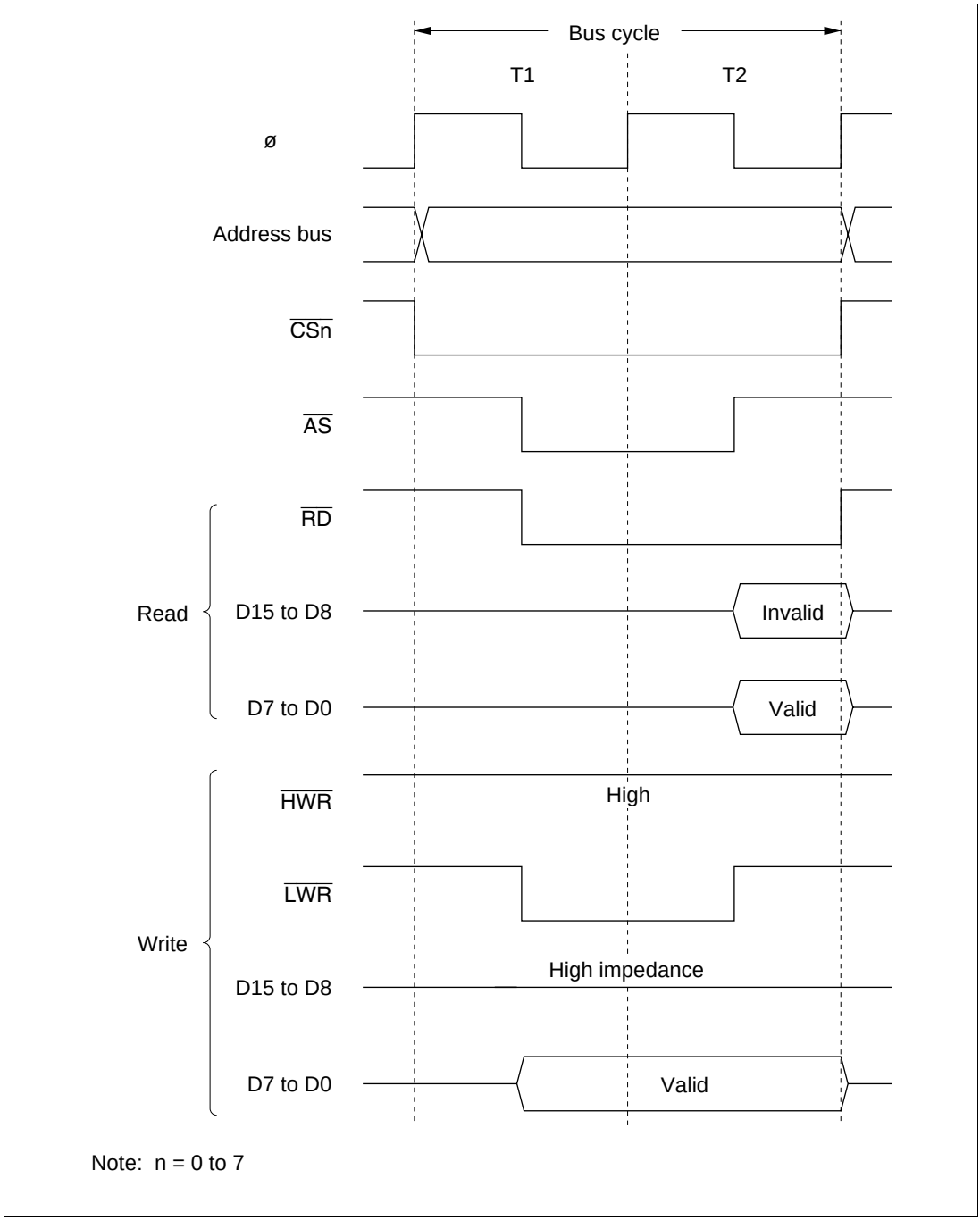


Figure 6-9 Bus Timing for 16-Bit 2-State Access Space (2) (Odd Address Byte Access)

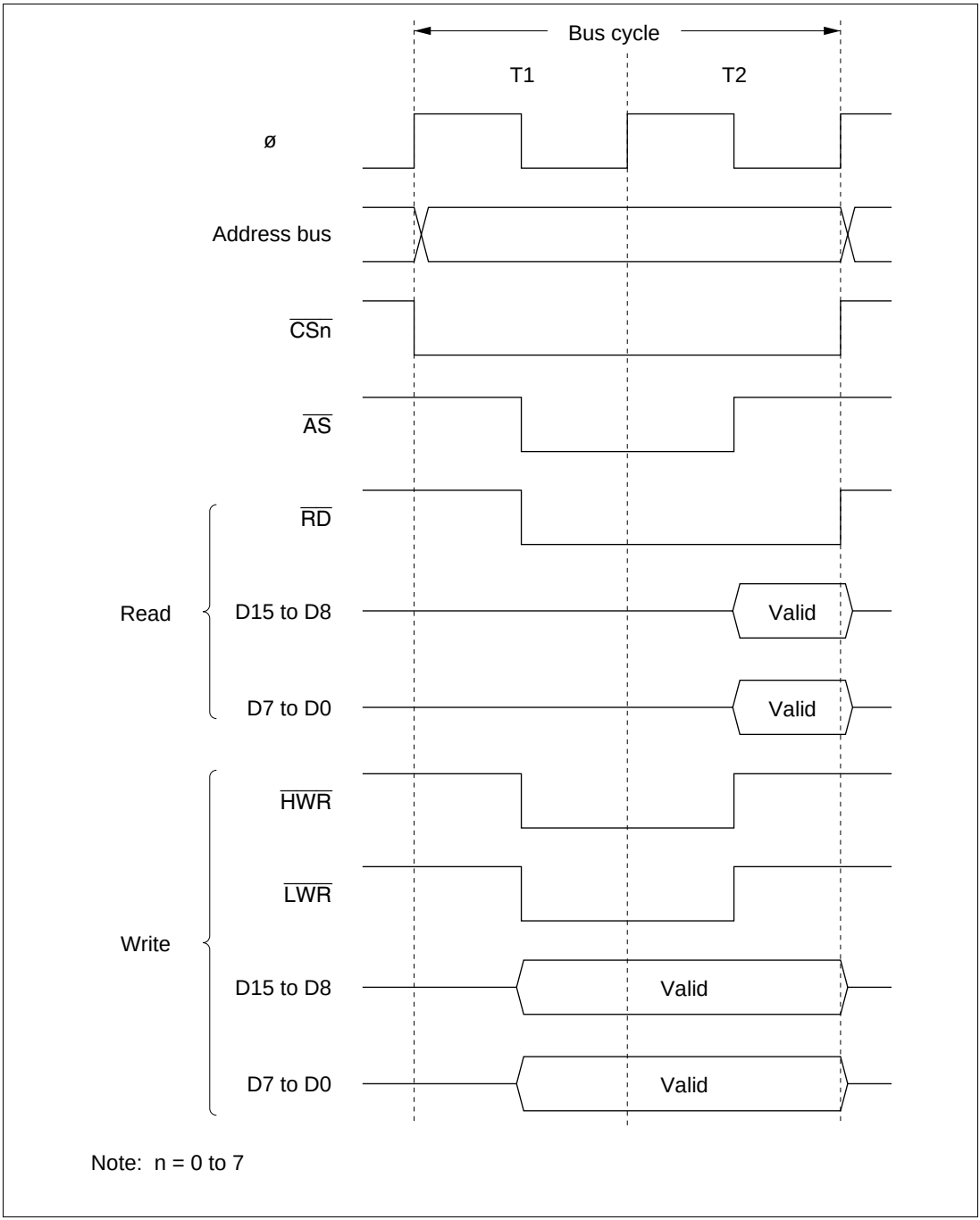


Figure 6-10 Bus Timing for 16-Bit 2-State Access Space (3) (Word Access)

16-Bit 3-State Access Space: Figures 6-11 to 6-13 show bus timings for a 16-bit 3-state access space. When a 16-bit access space is accessed , the upper half (D15 to D8) of the data bus is used for the even address, and the lower half (D7 to D0) for the odd address.

Wait states can be inserted.

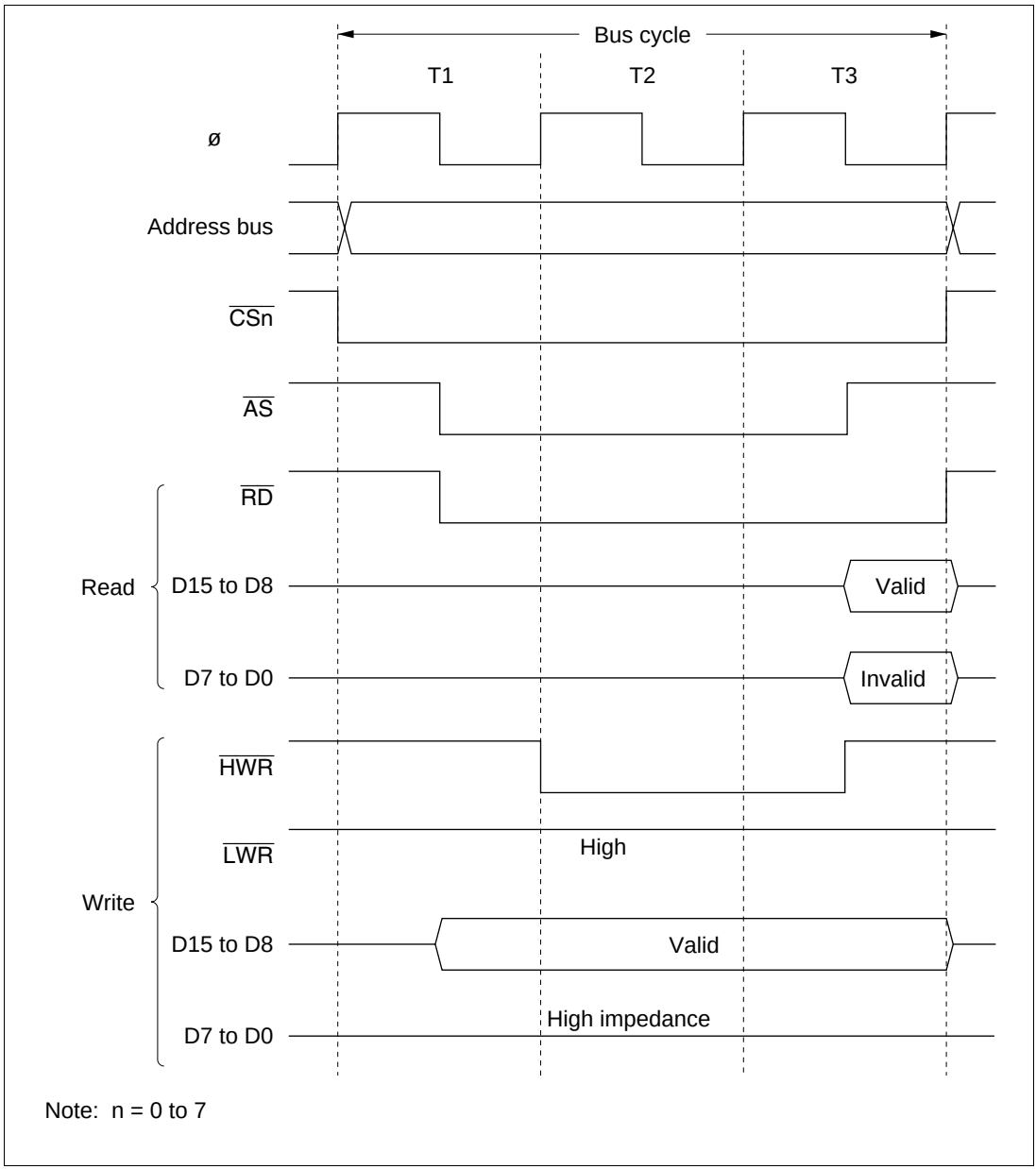


Figure 6-11 Bus Timing for 16-Bit 3-State Access Space (1) (Even Address Byte Access)

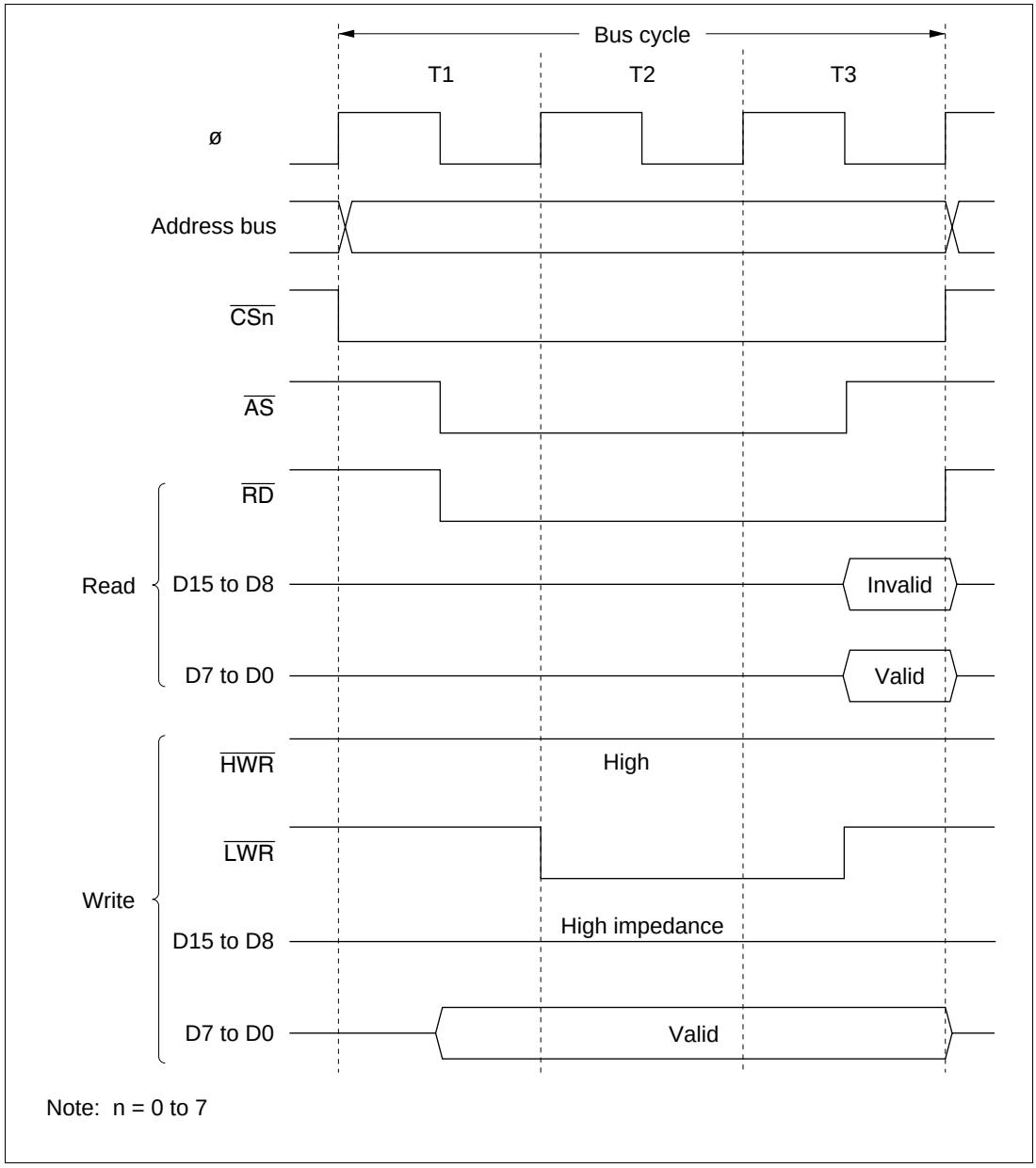


Figure 6-12 Bus Timing for 16-Bit 3-State Access Space (2) (Odd Address Byte Access)

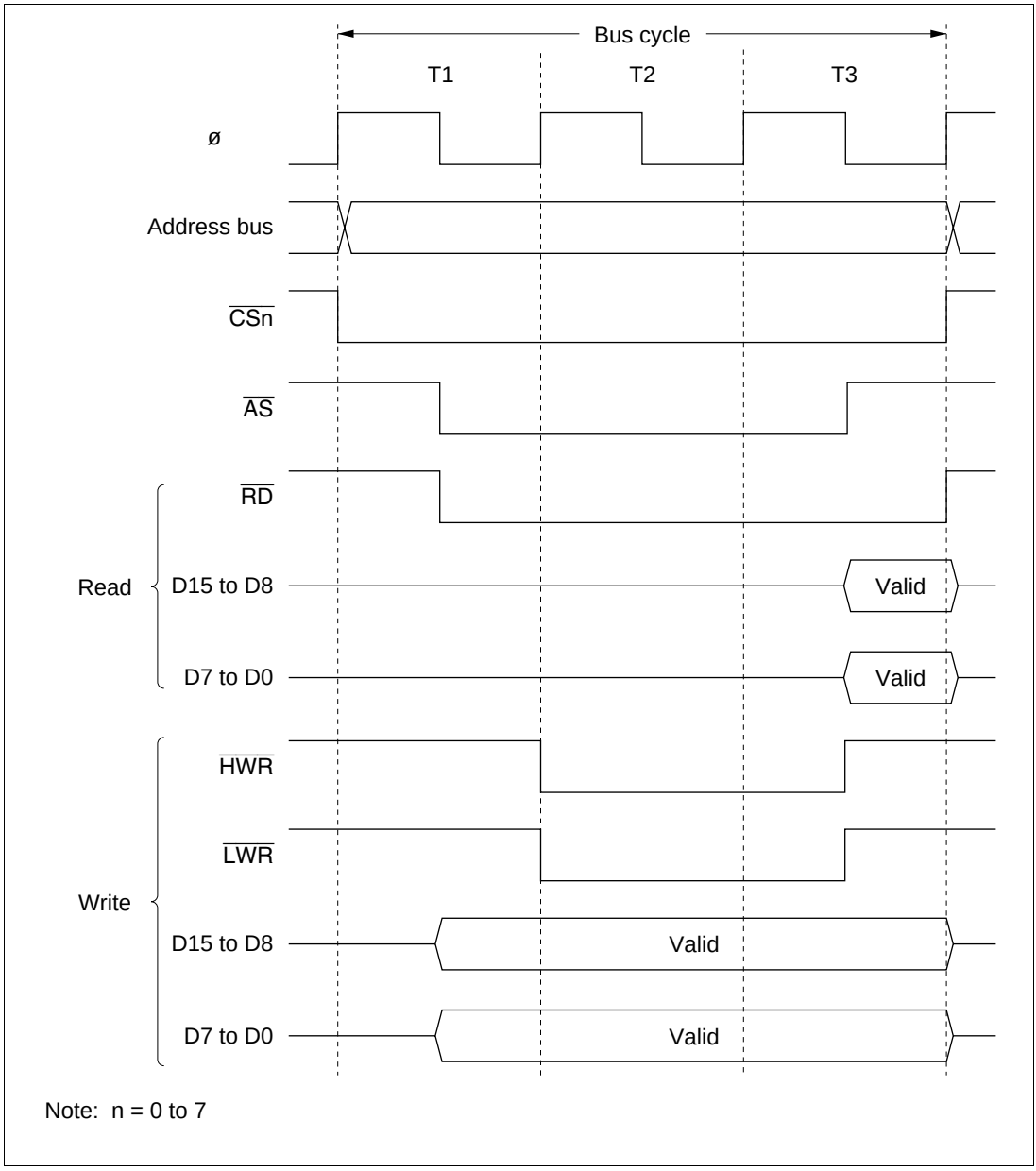


Figure 6-13 Bus Timing for 16-Bit 3-State Access Space (3) (Word Access)

6.4.5 Wait Control

When accessing external space, the H8S/2214 can extend the bus cycle by inserting one or more wait states (Tw). 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 3 wait states can be inserted automatically between the T2 state and T3 state on an individual area basis in 3-state access space, according to the settings of WCRH and WCRL.

Pin Wait Insertion

Setting the WAITE bit in BCRH to 1 enables wait insertion by means of the $\overline{\text{WAIT}}$ pin. When external space is accessed in this state, program wait insertion is first carried out according to the settings in WCRH and WCRL. Then , if the $\overline{\text{WAIT}}$ pin is low at the falling edge of ø in the last T2 or Tw state, a Tw state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, Tw states are inserted until it goes high.

This is useful when inserting four or more Tw states, or when changing the number of Tw states for different external devices.

The WAITE bit setting applies to all areas.

Figure 6-14 shows an example of wait state insertion timing.

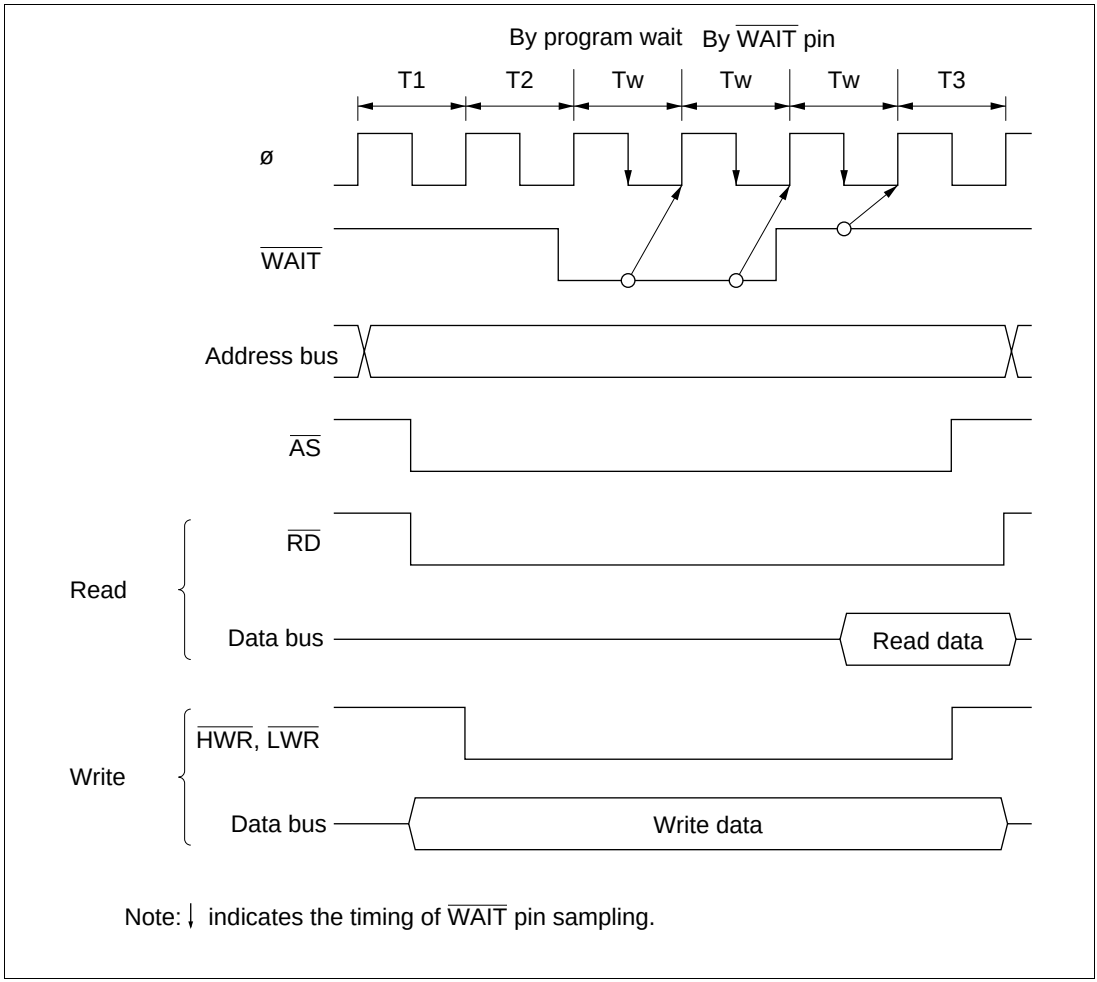


Figure 6-14 Example of Wait State Insertion Timing

The settings after a power-on reset are: 3-state access, 3 program wait state insertion, and $\overline{\text{WAIT}}$ input disabled. When a manual reset is performed, the contents of bus controller registers are retained, and the wait control settings remain the same as before the reset.

6.5 Burst ROM Interface

6.5.1 Overview

With the H8S/2214, external space area 0 can be designated as burst ROM space, and burst ROM interfacing can be performed. The burst ROM space interface enables 16-bit configuration ROM with burst access capability to be accessed at high speed.

Area 0 can be designated as burst ROM space by means of the BRSTRM bit in BCRH. Consecutive burst accesses of a maximum of 4 words or 8 words can be performed for CPU instruction fetches only. One or two states can be selected for burst access.

6.5.2 Basic Timing

The number of states in the initial cycle (full access) of the burst ROM interface is in accordance with the setting of the AST0 bit in ASTCR. Also, when the AST0 bit is set to 1, wait state insertion is possible. One or two states can be selected for the burst cycle, according to the setting of the BRSTS1 bit in BCRH. Wait states cannot be inserted. When area 0 is designated as burst ROM space, it becomes 16-bit access space regardless of the setting of the ABW0 bit in ABWCR.

When the BRSTS0 bit in BCRH is cleared to 0, burst access of up to 4 words is performed; when the BRSTS0 bit is set to 1, burst access of up to 8 words is performed.

The basic access timing for burst ROM space is shown in figures 6-15 and 6-16. The timing shown in figure 6-15 is for the case where the AST0 and BRSTS1 bits are both set to 1, and that in figure 6-16 is for the case where both these bits are cleared to 0.

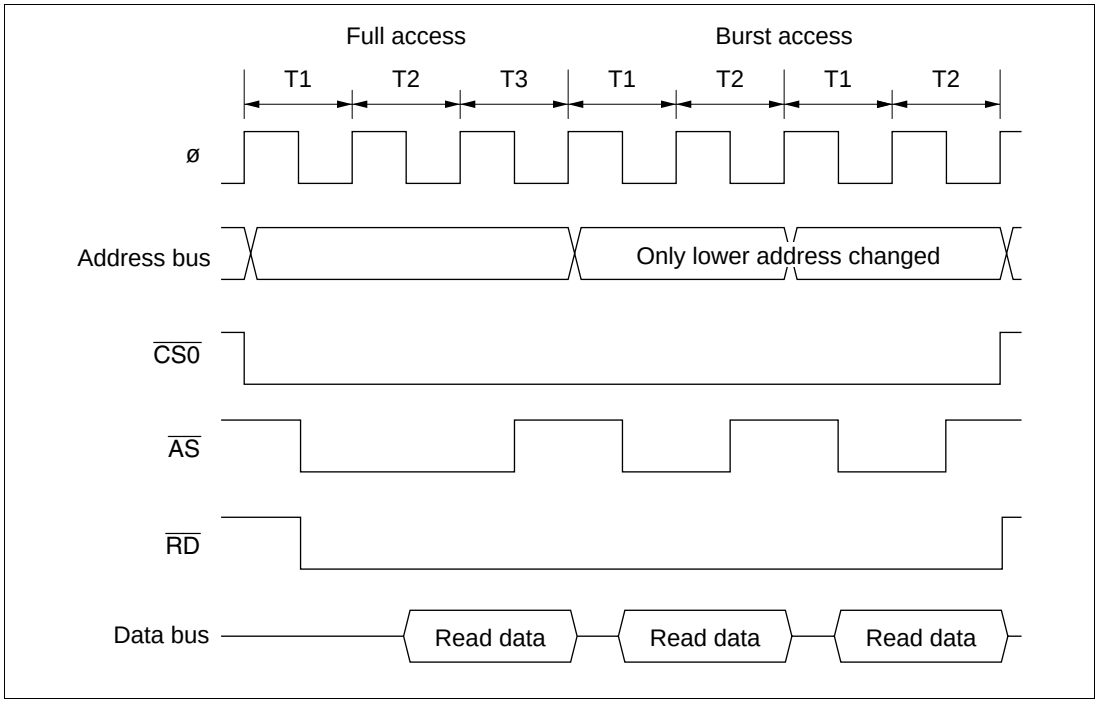


Figure 6-15 Example of Burst ROM Access Timing (When AST0 = BRSTS1 = 1)

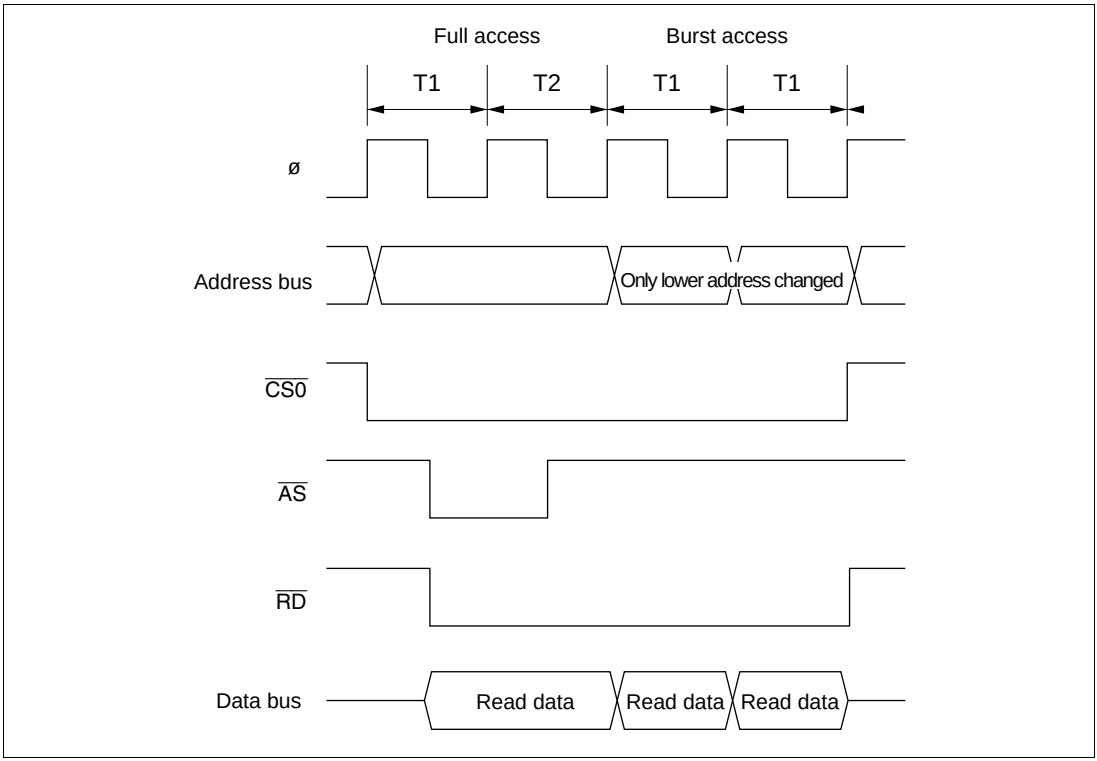


Figure 6-16 Example of Burst ROM Access Timing (When AST0 = BRSTS1 = 0)

6.5.3 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion using the $\overline{\text{WAIT}}$ pin can be used in the initial cycle (full access) of the burst ROM interface. See section 6.4.5, Wait Control.

Wait states cannot be inserted in a burst cycle.

6.6 Idle Cycle

6.6.1 Operation

When the H8S/2214 accesses external space , it can insert a 1-state idle cycle (T_I) between bus cycles in the following two cases: (1) when read accesses between different areas occur consecutively, and (2) when a write cycle occurs immediately after a read cycle. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, with a long output floating time, and high-speed memory, I/O interfaces, and so on.

(1) Consecutive Reads between Different Areas

If consecutive reads between different areas occur while the ICIS1 bit in BCRH is set to 1, an idle cycle is inserted at the start of the second read cycle.

Figure 6-17 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a read cycle from SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in 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.

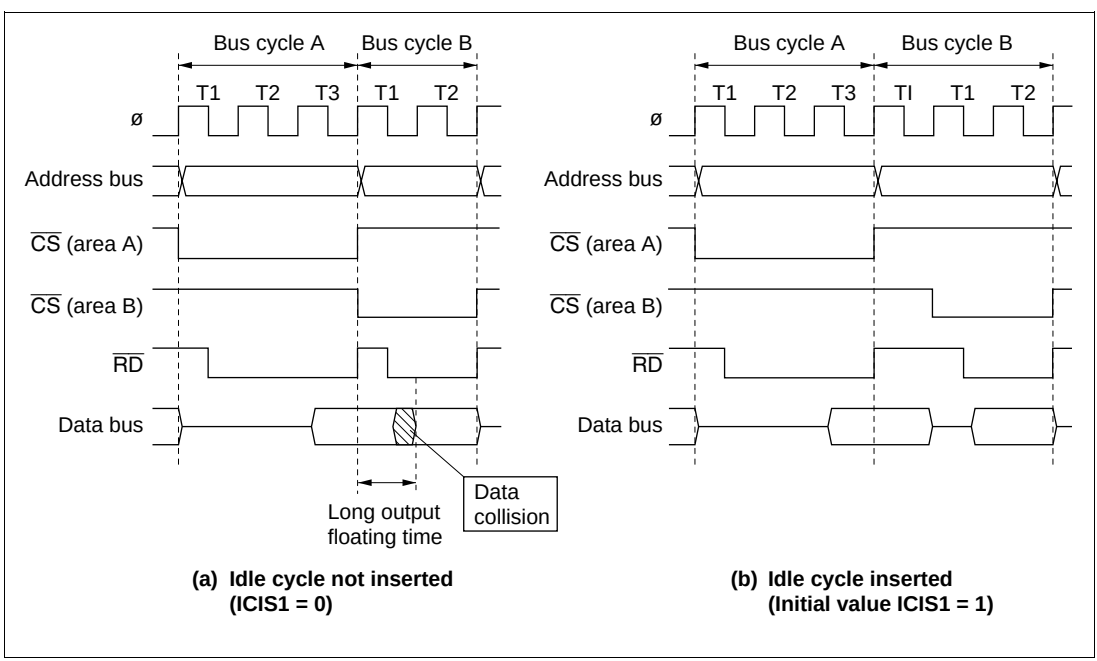


Figure 6-17 Example of Idle Cycle Operation (1)

(2) Write after Read

If an external write occurs after an external read while the ICIS0 bit in BCRH is set to 1, an idle cycle is inserted at the start of the write cycle.

Figure 6-18 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from 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 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.

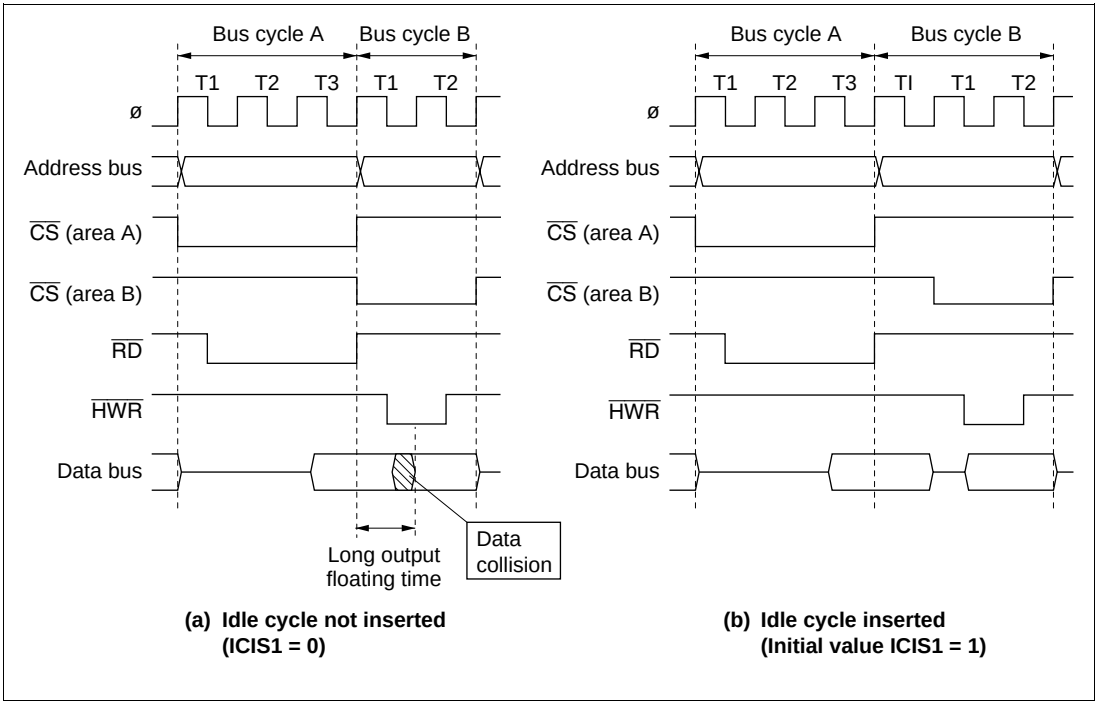


Figure 6-18 Example of Idle Cycle Operation (2)

(3) 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 6-19.

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.

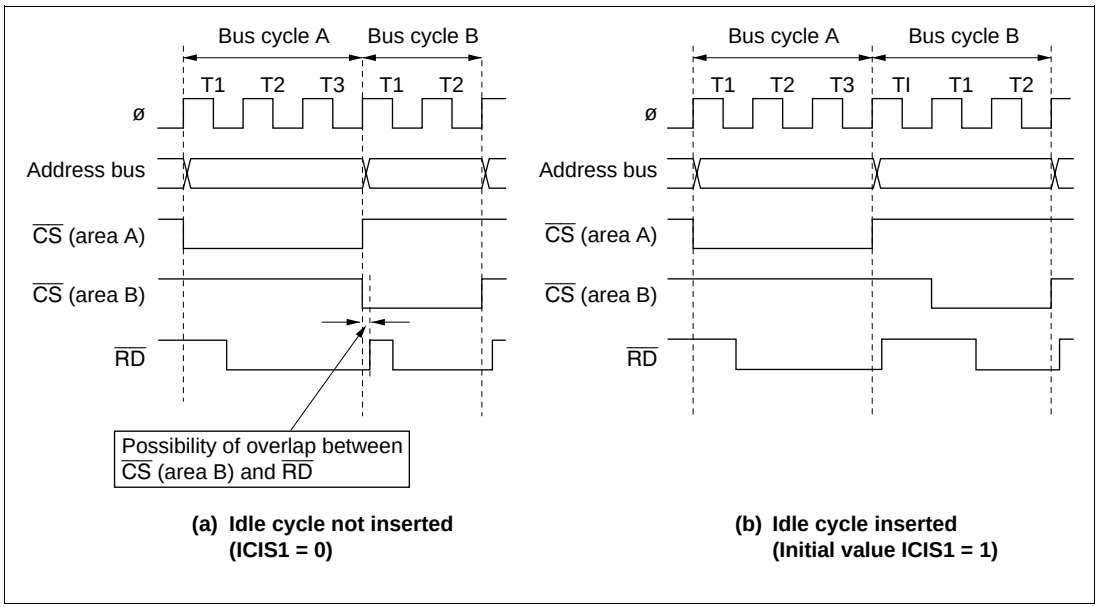


Figure 6-19 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

6.6.2 Pin States in Idle Cycle

Table 6-5 shows pin states in an idle cycle.

Table 6-5 Pin States in Idle Cycle

Pins	Pin State
A23 to A0	Contents of next bus cycle
D15 to D0	High impedance
$\overline{\text{CSn}}$	High
$\overline{\text{AS}}$	High
$\overline{\text{RD}}$	High
$\overline{\text{HWR}}$	High
$\overline{\text{LWR}}$	High
$\overline{\text{DACKn}}$	High

6.7 Bus Release

6.7.1 Overview

The H8S/2214 can release the external bus in response to a bus request from an external device. In the external bus released state, the internal bus master continues to operate as long as there is no external access.

6.7.2 Operation

In external expansion mode, the bus can be released to an external device by setting the BRLE bit in BCRL to 1. Driving the $\overline{\text{BREQ}}$ pin low issues an external bus request to the H8S/2214. 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, an internal bus master can perform accesses using the internal bus. When an internal bus master wants to make an external access, it temporarily defers activation of the bus cycle, and waits for the bus request from the external bus master to be dropped.

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.

In the event of simultaneous external bus release request and external access request generation, the order of priority is as follows:

(High) External bus release > Internal bus master external access (Low)

6.7.3 Pin States in External Bus Released State

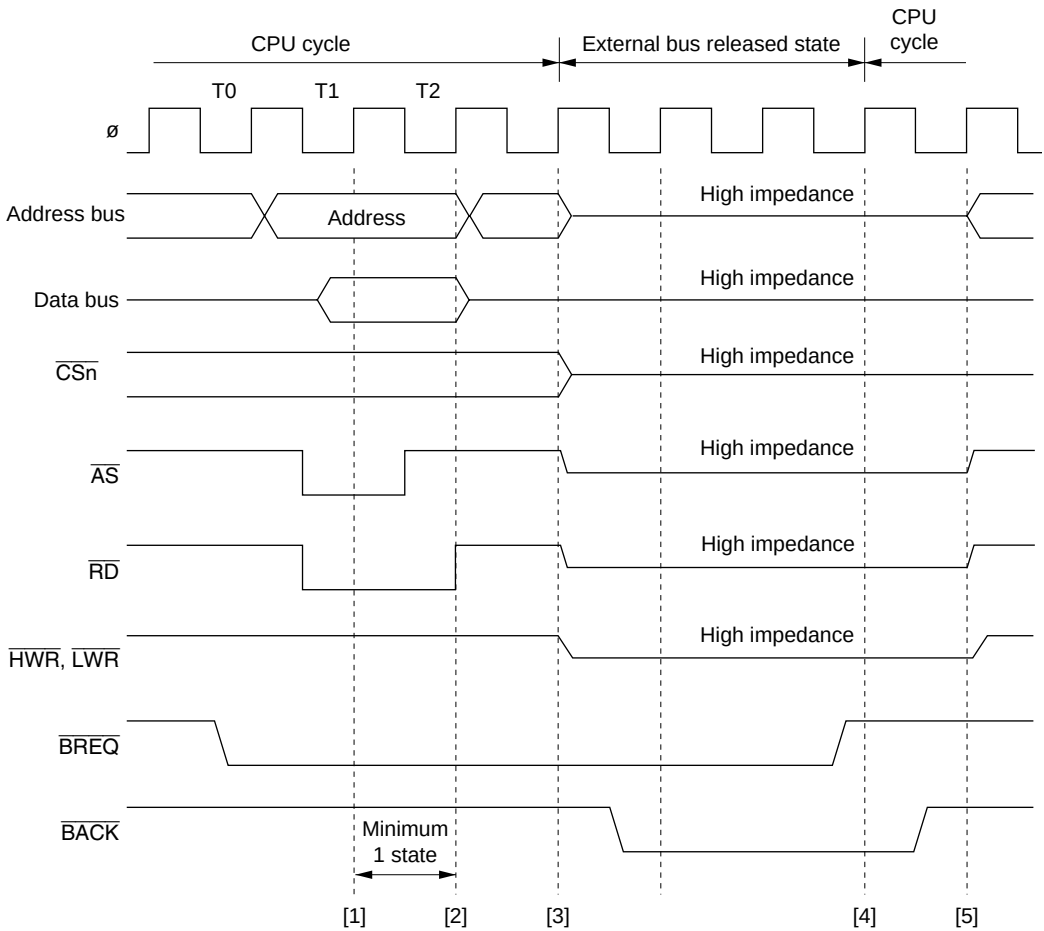
Table 6-6 shows pin states in the external bus released state.

Table 6-6 Pin States in Bus Released State

Pins	Pin State
A23 to A0	High impedance
D15 to D0	High impedance
$\overline{\text{CSn}}$	High impedance
$\overline{\text{AS}}$	High impedance
$\overline{\text{RD}}$	High impedance
$\overline{\text{HWR}}$	High impedance
$\overline{\text{LWR}}$	High impedance
$\overline{\text{DACKn}}$	High level

6.7.4 Transition Timing

Figure 6-20 shows the timing for transition to the bus-released state.



[1] Low level of $\overline{\text{BREQ}}$ pin is sampled at rise of T2 state.

[2] $\overline{\text{BACK}}$ pin is driven low at end of CPU read cycle, releasing bus to external bus master.

[3] $\overline{\text{BREQ}}$ pin state is still sampled in external bus released state.

[4] High level of $\overline{\text{BREQ}}$ pin is sampled.

[5] $\overline{\text{BACK}}$ pin is driven high, ending bus release cycle.

Note: n = 0 to 7

Figure 6-20 Bus-Released State Transition Timing

6.7.5 Usage Note

When MSTPCR is set to H'FFFFFF and a transition is made to sleep mode, the external bus release function halts. Therefore, MSTPCR should not be set to H'FFFFFF if the external bus release function is to be used in sleep mode.

6.8 Bus Arbitration

6.8.1 Overview

The H8S/2214 has a bus arbiter that arbitrates bus master operations.

There are two bus masters, the CPU, DMAC, and DTC, 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.

6.8.2 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 making the request. 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 an internal bus master, and external bus release, can be executed in parallel.

In the event of simultaneous external bus release request, and internal bus master external access request generation, the order of priority is as follows:

(High) External bus release > Internal bus master external access (Low)

6.8.3 Bus Transfer Timing

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 times 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 DMAC and DTC, 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 operations. See Appendix A-5, Bus States during Instruction Execution, for timings at which the bus is not transferred.
- If the CPU is in sleep mode, it transfers the bus 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.

6.8.4 External Bus Release Usage Note

External bus release can be performed on completion of an external bus cycle. The $\overline{CS}$ signal remains low until the end of the external bus cycle. Therefore, when external bus release is performed, the $\overline{CS}$ signal may change from the low level to the high-impedance state.

6.9 Resets and the Bus Controller

In a power-on reset, the H8S/2214, including the bus controller, enters the reset state at that point, and an executing bus cycle is discontinued.

In a manual reset, the bus controller's registers and internal state are maintained, and an executing external bus cycle is completed. In this case, $\overline{\text{WAIT}}$ input is ignored and write data is not guaranteed.

6.10 External Module Expansion Function

6.10.1 Overview

The H8S/2214 has an external module expansion function to provide for the addition of peripheral devices. Using this function to provide a combination of H8S/2214 and external modules makes it possible to implement a multichip system on the user board.

Figure 6-21 shows a block diagram.

Bus access states can be changed by means of a bus controller setting.

The $\overline{\text{EXMS}}$ signal is output to external modules for addresses H'FFFF40 to H'FFFF5F.

Priority and DTC activation can be specified for interrupts $\overline{\text{EXIRQ7}}$ to $\overline{\text{EXIRQ0}}$ in the same way as for the H8S/2214's on-chip supporting functions.

The DTC data transfer end signal for $\overline{\text{EXIRQ7}}$ to $\overline{\text{EXIRQ0}}$ interrupt input is output from $\overline{\text{EXDTCE}}$. Also, the inverse of the value of bit 0 in module stop control register B is output from EXMSTP.

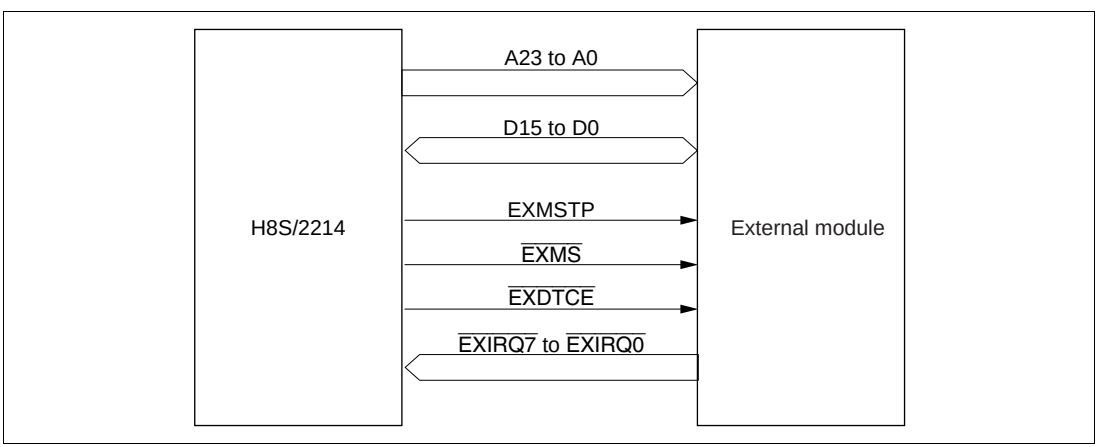


Figure 6-21 Multichip Block Diagram

6.10.2 Pin Configuration

Table 6-7 summarizes the pins of the external module expansion function.

Table 6-7 External Module Expansion Function Pins

Name	Symbol	I/O	Function
External expansion interrupt request 7 to 0	$\overline{\text{EXIRQ7}}$ to $\overline{\text{EXIRQ0}}$	Input	Input pins for interrupt requests from external modules
External expansion module select	$\overline{\text{EXMS}}$	Output	Select signal for external modules
External expansion DTC transfer end	$\overline{\text{EXDTCE}}$	Output	DTC transfer end signal for $\overline{\text{EXIRQ7}}$ to $\overline{\text{EXIRQ0}}$ interrupt input
External expansion module stop	EXMSTP	Output	Module stop signal for external modules

6.10.3 Register Configuration

Table 6-8 summarizes the registers of the bus controller.

Table 6-8 Bus Controller Registers

			Initial Value		
Name	Abbreviation	R/W	Power-On Reset	Manual Reset	Address*
Interrupt request input pin select register 0	IPINSEL0	R/W	H'00	Retained	H'FE4A
External module connection output pin select register	OPINSEL	R/W	B'-000----	Retained	H'FE4E
Module stop control register B	MSTPCRB	R/W	H'FF	H'FF	H'FDE9

Note: * Lower 16 bits of the address.

6.11 Register Descriptions

6.11.1 Interrupt Request Input Pin Select Register 0 (IPINSEL0)

Bit	:	7	6	5	4	3	2	1	0
		P36 IRQ7E	P47 IRQ6E	P46 IRQ5E	P44 IRQ4E	P43 IRQ3E	P42 IRQ2E	P41 IRQ1E	P40 IRQ0E
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IPINSEL0 is an 8-bit readable/writable register that selects which pins are to be used for interrupt request input signals ($\overline{\text{EXIRQ7}}$ to $\overline{\text{EXIRQ0}}$) from externally connected modules when operating as H8S/2214 modules. IPINSEL0 is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state in a manual reset and in software standby mode.

Bit 7—Enable of $\overline{\text{EXIRQ7}}$ Input from P36 (P36IRQ7E): Selects whether or not P36 is used as the $\overline{\text{EXIRQ7}}$ input pin.

Bit 7

P36IRQ7E	Description	
0	P36 is not used as $\overline{\text{EXIRQ7}}$ input	(Initial value)
1	P36 is used as $\overline{\text{EXIRQ7}}$ input	

Bit 6—Enable of $\overline{\text{EXIRQ6}}$ Input from P47 (P47IRQ6E): Selects whether or not P47 is used as the $\overline{\text{EXIRQ6}}$ input pin.

Bit 6

P47IRQ6E	Description	
0	P47 is not used as $\overline{\text{EXIRQ6}}$ input	(Initial value)
1	P47 is used as $\overline{\text{EXIRQ6}}$ input	

Bit 5—Enable of $\overline{\text{EXIRQ5}}$ Input from P46 (P46IRQ5E): Selects whether or not P46 is used as the $\overline{\text{EXIRQ5}}$ input pin.

Bit 5

P46IRQ5E	Description	
0	P46 is not used as $\overline{\text{EXIRQ5}}$ input	(Initial value)
1	P46 is used as $\overline{\text{EXIRQ5}}$ input	

Bit 4—Enable of $\overline{\text{EXIRQ4}}$ Input from P44 (P44IRQ4E): Selects whether or not P44 is used as the $\overline{\text{EXIRQ4}}$ input pin.

Bit 4

P44IRQ4E	Description	
0	P44 is not used as $\overline{\text{EXIRQ4}}$ input	(Initial value)
1	P44 is used as $\overline{\text{EXIRQ4}}$ input	

Bit 3—Enable of $\overline{\text{EXIRQ3}}$ Input from P43 (P43IRQ3E): Selects whether or not P43 is used as the $\overline{\text{EXIRQ3}}$ input pin.

Bit 3

P43IRQ3E	Description	
0	P43 is not used as $\overline{\text{EXIRQ3}}$ input	(Initial value)
1	P43 is used as $\overline{\text{EXIRQ3}}$ input	

Bit 2—Enable of $\overline{\text{EXIRQ2}}$ Input from P42 (P42IRQ2E): Selects whether or not P42 is used as the $\overline{\text{EXIRQ2}}$ input pin.

Bit 2

P42IRQ2E	Description	
0	P42 is not used as $\overline{\text{EXIRQ2}}$ input	(Initial value)
1	P42 is used as $\overline{\text{EXIRQ2}}$ input	

Bit 1—Enable of $\overline{\text{EXIRQ1}}$ Input from P41 (P41IRQ1E): Selects whether or not P41 is used as the $\overline{\text{EXIRQ1}}$ input pin.

Bit 1

P41IRQ1E	Description	
0	P41 is not used as $\overline{\text{EXIRQ1}}$ input	(Initial value)
1	P41 is used as $\overline{\text{EXIRQ1}}$ input	

Bit 0—Enable of $\overline{\text{EXIRQ0}}$ Input from P40 (P40IRQ0E): Selects whether or not P40 is used as the $\overline{\text{EXIRQ0}}$ input pin.

Bit 0

P40IRQ0E	Description	
0	P40 is not used as $\overline{\text{EXIRQ0}}$ input	(Initial value)
1	P40 is used as $\overline{\text{EXIRQ0}}$ input	

6.11.2 External Module Connection Output Pin Select Register (OPINSEL)

Bit	:	7	6	5	4	3	2	1	0
		—	P76 STPOE	P75 MSOE	P74 DTCOE	—	—	—	—
Initial value	:	Undefined	0	0	0	Undefined	Undefined	Undefined	Undefined
R/W	:	R/W	R/W	R/W	R/W	—	—	—	—

OPINSEL is an 8-bit readable/writable register that selects whether or not output signals (EXDTCEN, EXMSTP, EXMSN) to externally connected modules are output to pins P77 to P74 in H8S/2214 operation. OPINSEL bits 6 to 4 are initialized to 000 by a power-on reset and in hardware standby mode. They retain their previous states in a manual reset and in software standby mode.

Bit 7—Reserved: This bit will return an undefined value if read, and should only be written with 0.

Bit 6—Enable of EXMSTP Output to P76 (P76STPOE): Selects whether or not the EXMSTP module stop signal to external modules (corresponding to bit 0 in MSTPCRB) is output to P76.

Bit 6

P76STPOE	Description	
0	EXMSTP is not output to P76	(Initial value)
1	EXMSTP is output to P76	

Bit 5—Enable of $\overline{\text{EXMS}}$ Output to P75 (P75MSOE): Selects whether or not the $\overline{\text{EXMS}}$ module stop signal to external modules (corresponding to addresses H'FFFF40 to H'FFFF5F) is output to P75.

Bit 5

P75MSOE	Description	
0	$\overline{\text{EXMS}}$ is not output to P75	(Initial value)
1	$\overline{\text{EXMS}}$ is output to P75	

Bit 4—Enable of $\overline{\text{EXDTCE}}$ Output to P74 (P74DTCOE): Selects whether or not the $\overline{\text{EXDTCE}}$ signal, indicating that DTC transfer corresponding to EXIRQ0—F input is in progress, is output to P74. This signal is used, for example, when the DTC in the chip has been activated by an interrupt (EXIRQ0 to EXIRQF) from an external module, and the interrupt request is to be cleared automatically on the external module side by DTC transfer.

Bit 4

P74DTCOE	Description	
0	$\overline{\text{EXDTCE}}$ is not output to P74	(Initial value)
1	$\overline{\text{EXDTCE}}$ is output to P74	

Bits 3 to 0—Reserved: These bits will return an undefined value if read, and should only be written with 0.

6.11.3 Module Stop Control Register B (MSTPCRB)

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRB is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPB0 bit is set to 1, the external module expansion function stops operation at the end of the bus cycle, and enters module stop mode. For details, see section 17.5, Module Stop Mode.

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

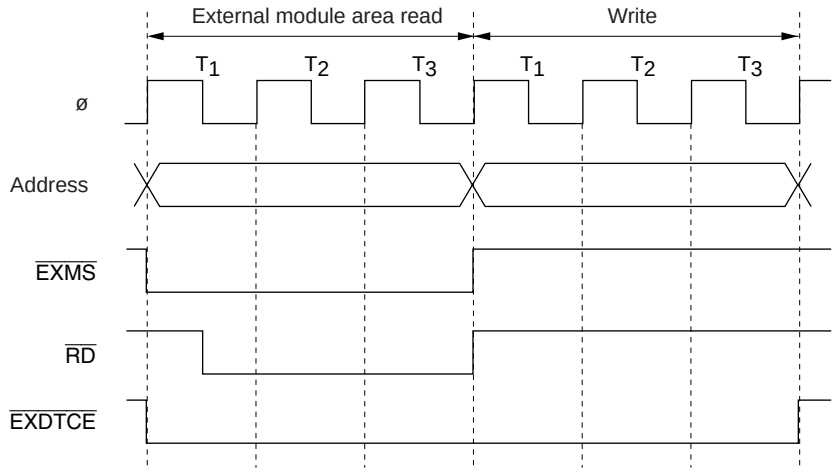
Bit 0—Module Stop (MSTPB0): Specifies the external module expansion function module stop mode.

Bit 0

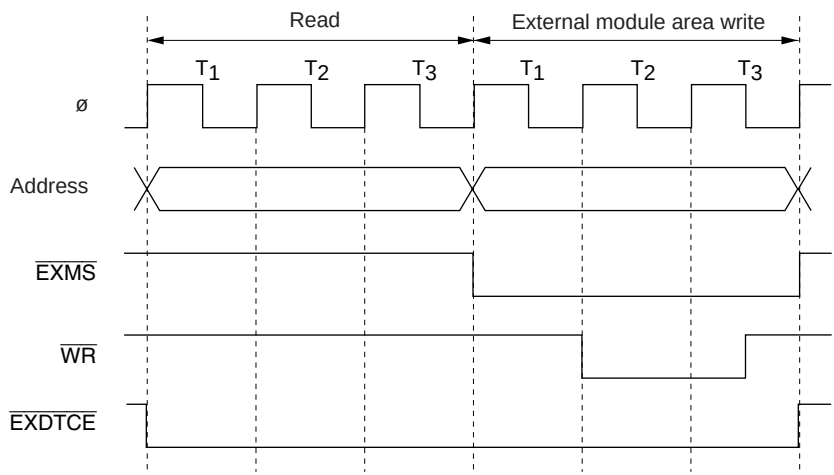
MSTPB0	Description	
0	External module expansion function module stop mode is cleared	
1	External module expansion function module stop mode is set	(Initial value)

6.12 Basic Timing

Figure 6-22 shows the timing of external module area (H'FFFF40 to H'FFFF5F) DTC data transfer using 3-state access.



(a) Timing of external module area read by DTC



(b) Timing of external module area write by DTC

Figure 6-22 Timing of External Module Area Access by DTC

Section 7 DMA Controller

7.1 Overview

The H8S/2214 has a built-in DMA controller (DMAC) which can carry out data transfer on up to 4 channels.

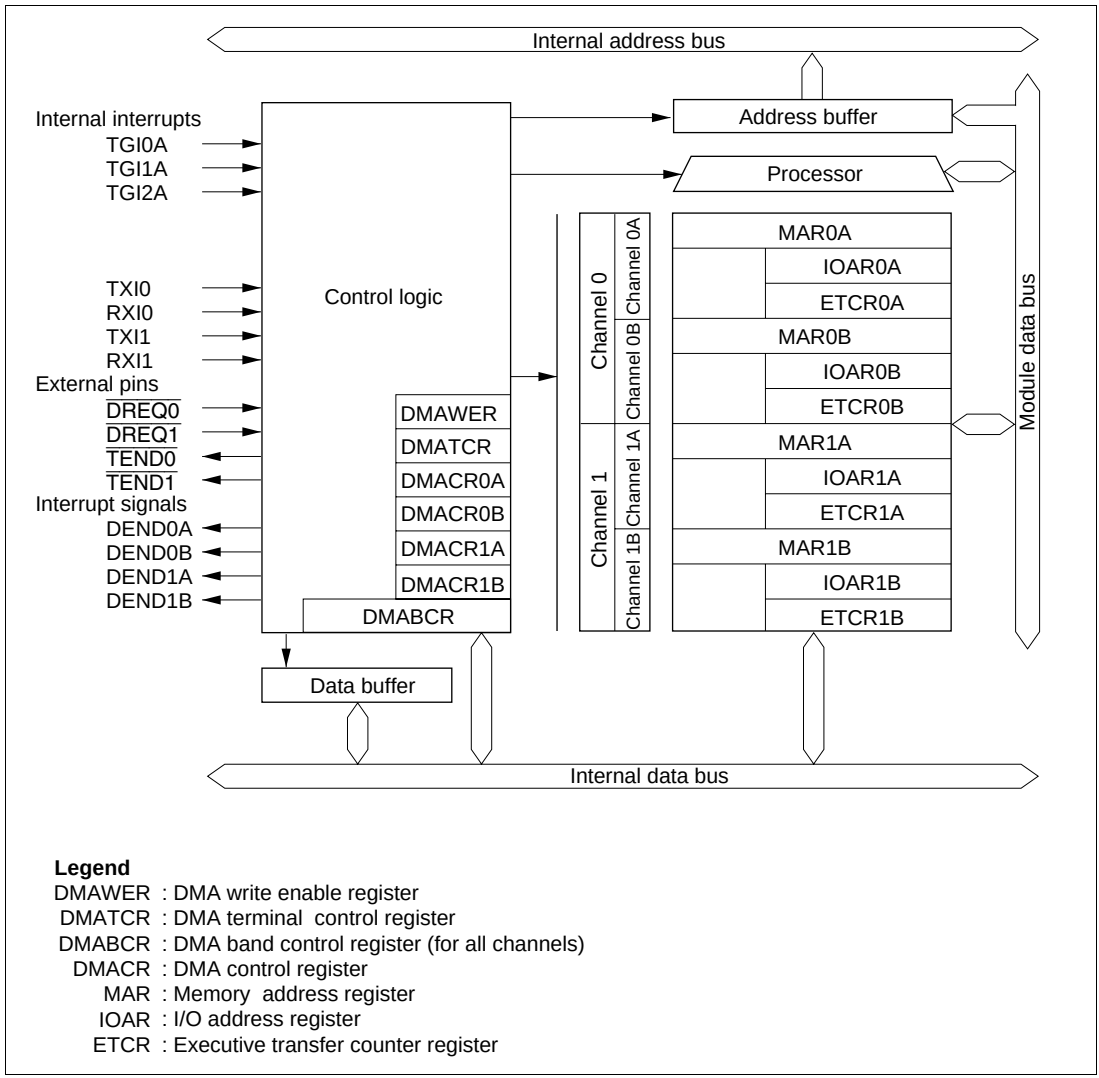
7.1.1 Features

The features of the DMAC are listed below.

- Choice of short address mode or full address mode
  Short address mode
  — Maximum of 4 channels can be used
  — Choice of dual address mode
  — In dual address mode, one of the two addresses, transfer source and transfer destination, is specified as 24 bits and the other as16 bits
  — Choice of sequential mode, idle mode, or repeat mode for dual address mode
  Full address mode
  — Maximum of 2 channels can be used
  — Transfer source and transfer destination address 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)
  — Three 16-bit timer-pulse unit (TPU) compare match/input capture interrupts
  — Serial communication interface (SCI0, SCI1) transmission complete interrupt, reception complete interrupt
  — External request
  — Auto-request
- Module stop mode can be set
  — The initial setting enables DMAC registers to be accessed. DMAC operation is halted by setting module stop mode

7.1.2 Block Diagram

A block diagram of the DMAC is shown in figure 7-1.



Legend

DMAWER : DMA write enable register
DMATCR : DMA terminal control register
DMABCR : DMA band control register (for all channels)
DMACR : DMA control register
MAR : Memory address register
IOAR : I/O address register
ETCR : Executive transfer counter register

Figure 7-1 Block Diagram of DMAC

7.1.3 Overview of Functions

Tables 7-1 and 7-2 summarize DMAC functions in short address mode and full address mode, respectively.

Table 7-1 Overview of DMAC Functions (Short Address Mode)

		Address Register Bit Length	
Transfer Mode	**Transfer Source**	**Source**	**Destination**
Dual address mode • Sequential mode — 1-byte or 1-word transfer executed for one transfer request — Memory address incremented/decremented by 1 or 2 — 1 to 65536 transfers • Idle mode — 1-byte or 1-word transfer executed for one transfer request — Memory address fixed — 1 to 65536 transfers • Repeat mode — 1-byte or 1-word transfer executed for one transfer request — Memory address incremented/ decremented by 1 or 2 — After specified number of transfers (1 to 256), initial state is restored and operation continues	• TPU channel 0 to 2 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • External request	24/16	16/24

Table 7-2 Overview of DMAC Functions (Full Address Mode)

Transfer Mode	Transfer Source	Address Register Bit Length	
		Source	Destination
• Normal mode Auto-request — Transfer request retained internally — Transfers continue for the specified number of times (1 to 65536) — Choice of burst or cycle steal transfer	• Auto-request	24	24
External request — 1-byte or 1-word transfer executed for one transfer request — 1 to 65536 transfers	• External request		
• Block transfer mode — Specified block size transfer executed for one transfer request — 1 to 65536 transfers — Either source or destination specifiable as block area — Block size: 1 to 256 bytes or words	• TPU channel 0 to 2 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • External request	24	24

7.1.4 Pin Configuration

Table 7-3 summarizes the DMAC pins.

In short address mode, external request transfer, and transfer end output are not performed for channel A.

When the $\overline{\text{DREQ}}$ pin is used, do not designate the corresponding port for output.

With regard to the $\overline{\text{TEND}}$ pins, whether or not the corresponding port is used as a $\overline{\text{TEND}}$ pin can be specified by means of a register setting.

Table 7-3 DMAC Pins

Channel	Pin Name	Symbol	I/O	Function
0	DMA request 0	$\overline{\text{DREQ0}}$	Input	DMAC channel 0 external request
	DMA transfer end 0	$\overline{\text{TEND0}}$	Output	DMAC channel 0 transfer end
1	DMA request 1	$\overline{\text{DREQ1}}$	Input	DMAC channel 1 external request
	DMA transfer end 1	$\overline{\text{TEND1}}$	Output	DMAC channel 1 transfer end

7.1.5 Register Configuration

Table 7-4 summarizes the DMAC registers.

Table 7-4 DMAC Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*	Bus Width
0	Memory address register 0A	MAR0A	R/W	Undefined	H'FEE0	16 bits
	I/O address register 0A	IOAR0A	R/W	Undefined	H'FEE4	16 bits
	Transfer count register 0A	ETCR0A	R/W	Undefined	H'FEE6	16 bits
	Memory address register 0B	MAR0B	R/W	Undefined	H'FEE8	16 bits
	I/O address register 0B	IOAR0B	R/W	Undefined	H'FEEC	16 bits
	Transfer count register 0B	ETCR0B	R/W	Undefined	H'FEEE	16 bits
1	Memory address register 1A	MAR1A	R/W	Undefined	H'FEF0	16 bits
	I/O address register 1A	IOAR1A	R/W	Undefined	H'FEF4	16 bits
	Transfer count register 1A	ETCR1A	R/W	Undefined	H'FEF6	16 bits
	Memory address register 1B	MAR1B	R/W	Undefined	H'FEF8	16 bits
	I/O address register 1B	IOAR1B	R/W	Undefined	H'FEFC	16 bits
	Transfer count register 1B	ETCR1B	R/W	Undefined	H'FEFE	16 bits
0, 1	DMA write enable register	DMAWER	R/W	H'00	H'FF60	8 bits
	DMA terminal control register	DMATCR	R/W	H'00	H'FF61	8 bits
	DMA control register 0A	DMACR0A	R/W	H'00	H'FF62	16 bits
	DMA control register 0B	DMACR0B	R/W	H'00	H'FF63	16 bits
	DMA control register 1A	DMACR1A	R/W	H'00	H'FF64	16 bits
	DMA control register 1B	DMACR1B	R/W	H'00	H'FF65	16 bits
	DMA band control register	DMABCR	R/W	H'0000	H'FF66	16 bits
	Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8	8 bits

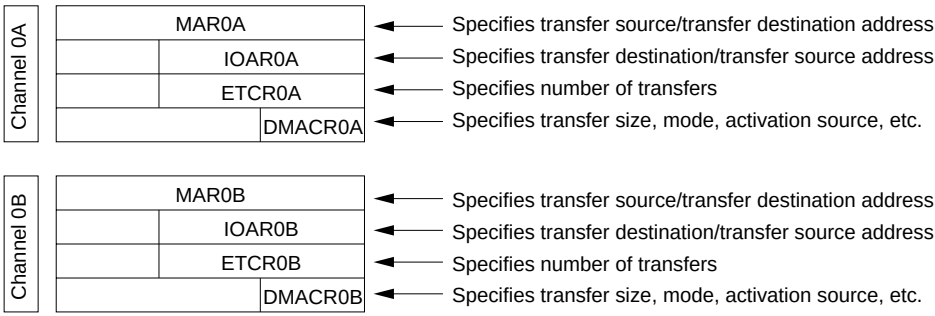
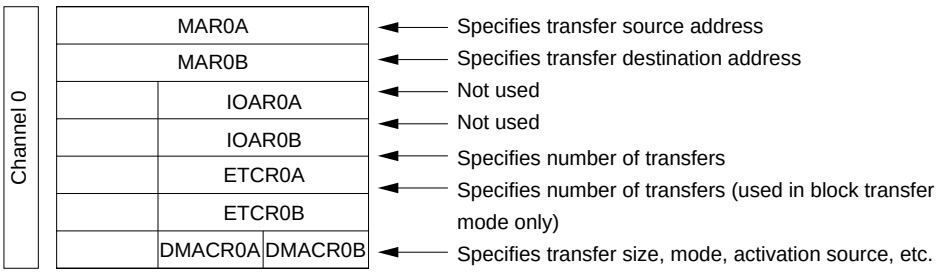
Note: * Lower 16 bits of the address.

7.2 Register Descriptions (1) (Short Address Mode)

Short address mode transfer can be performed for channels A and B independently.

Short address mode transfer is specified for each channel by clearing the FAE bit in DMABCR to 0, as shown in table 7-5. Short address mode or full address mode can be selected for channels 1 and 0 independently by means of bits FAE1 and FAE0.

Table 7-5 Short Address Mode and Full Address Mode (For 1 Channel: Example of Channel 0)

FAE0	Description
0	Short address mode specified (channels A and B operate independently) 
1	Full address mode specified (channels A and B operate in combination) 

7.2.1 Memory Address Registers (MAR)

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

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

*: Undefined

MAR is a 32-bit readable/writable register that specifies the transfer source address or destination address.

The upper 8 bits of MAR are reserved: they are always read as 0, and cannot be modified.

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. For details, see section 7.2.4, DMA Control Register (DMACR).

MAR is not initialized by a reset or in standby mode.

7.2.2 I/O Address Register (IOAR)

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

*: Undefined

IOAR is a 16-bit readable/writable register that specifies the lower 16 bits of the transfer source address or destination address. The upper 8 bits of the transfer address are automatically set to H'FF.

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 invalid in single address mode.

IOAR is not incremented or decremented each time a transfer is executed, so that the address specified by IOAR is fixed.

IOAR is not initialized by a reset or in standby mode.

7.2.3 Execute Transfer Count Register (ETCR)

ETCR is a 16-bit readable/writable register that specifies the number of transfers. The setting of this register is different for sequential mode and idle mode on the one hand, and for repeat mode on the other.

(1) Sequential Mode and Idle Mode

Transfer Counter

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

*: Undefined

In sequential mode and idle mode, ETCR functions as a 16-bit transfer counter (with a count range of 1 to 65536). ETCR is decremented by 1 each time a transfer is performed, and when the count reaches H'0000, the DTE bit in DMABCR is cleared, and transfer ends.

(2) Repeat Mode

Transfer Number Storage

Bit	:	15	14	13	12	11	10	9	8
ETCRH	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transfer Counter

Bit	:	7	6	5	4	3	2	1	0
ETCRL	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

In repeat mode, ETCR functions as transfer counter ETCRL (with a count range of 1 to 256) and transfer number storage register ETCRH. 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 DMABCR is not cleared, and so transfers can be performed repeatedly until the DTE bit is cleared by the user.

ETCR is not initialized by a reset or in standby mode.

7.2.4 DMA Control Register (DMACR)

Bit	:	7	6	5	4	3	2	1	0
DMACR	:	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DMACR is an 8-bit readable/writable register that controls the operation of each DMAC channel.

DMACR is initialized to H'00 by a reset, and in standby mode.

Bit 7—Data Transfer Size (DTSZ): Selects the size of data to be transferred at one time.

Bit 7		
DTSZ	**Description**	
0	Byte-size transfer	(Initial value)
1	Word-size transfer	

Bit 6—Data Transfer Increment/Decrement (DTID): Selects incrementing or decrementing of MAR every data transfer in sequential mode or repeat mode.

In idle mode, MAR is neither incremented nor decremented.

Bit 6		
DTID	**Description**	
0	MAR is incremented after a data transfer • When DTSZ = 0, MAR is incremented by 1 after a transfer • When DTSZ = 1, MAR is incremented by 2 after a transfer	(Initial value)
1	MAR is decremented after a data transfer • When DTSZ = 0, MAR is decremented by 1 after a transfer • When DTSZ = 1, MAR is decremented by 2 after a transfer	

Bit 5—Repeat Enable (RPE): Used in combination with the DTIE bit in DMABCR to select the mode (sequential, idle, or repeat) in which transfer is to be performed.

Bit 5	DMABCR		
RPE	**DTIE**	**Description**	
0	0	Transfer in sequential mode (no transfer end interrupt)	(Initial value)
	1	Transfer in sequential mode (with transfer end interrupt)	
1	0	Transfer in repeat mode (no transfer end interrupt)	
	1	Transfer in idle mode (with transfer end interrupt)	

For details of operation in sequential, idle, and repeat mode, see section 7.5.2, Sequential Mode, section 7.5.3, Idle Mode, and section 7.5.4, Repeat Mode.

Bit 4—Data Transfer Direction (DTDIR): To specify the data transfer direction (source or destination).

Bit 4		
DTDIR	**Description**	
0	Transfer with MAR as source address and IOAR as destination address	Initial value)
1	Transfer with IOAR as source address and MAR as destination address	

Bits 3 to 0—Data Transfer Factor (DTF3 to DTF0): These bits select the data transfer factor (activation source). There are some differences in activation sources for channel A and for channel B.

Channel A

Bit 3	Bit 2	Bit 1	Bit 0	
DTF3	**DTF2**	**DTF1**	**DTF0**	**Description**
0	0	0	0	— (Initial value)
			1	—
		1	0	—
			1	—
	1	0	0	Activated by SCI channel 0 transmission complete interrupt
			1	Activated by SCI channel 0 reception complete interrupt
		1	0	Activated by SCI channel 1 transmission complete interrupt
			1	Activated by SCI channel 1 reception complete interrupt
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt
			1	Activated by TPU channel 1 compare match/input capture A interrupt
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt
			1	—
	1	0	0	—
			1	—
		1	0	—
			1	—

Channel B

Bit 3	Bit 2	Bit 1	Bit 0	
DTF3	**DTF2**	**DTF1**	**DTF0**	**Description**
0	0	0	0	— (Initial value)
			1	—
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input*
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	0	Activated by SCI channel 0 transmission complete interrupt
			1	Activated by SCI channel 0 reception complete interrupt
		1	0	Activated by SCI channel 1 transmission complete interrupt
			1	Activated by SCI channel 1 reception complete interrupt
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt
			1	Activated by TPU channel 1 compare match/input capture A interrupt
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt
			1	—
	1	0	0	—
			1	—
		1	0	—
			1	—

Note: * Detected as a low level in the first transfer after transfer is enabled.

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 7.5.13, DMAC Multi-Channel Operation.

7.2.5 DMA Band Control Register (DMABCR)

Bit	:	15	14	13	12	11	10	9	8
DMABCRH	:	FAE1	FAE0	—	—	DTA1B	DTA1A	DTA0B	DTA0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	7	6	5	4	3	2	1	0
DMABCRL	:	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DMABCR is a 16-bit readable/writable register that controls the operation of each DMAC channel.

DMABCR is initialized to H'0000 by a reset, and in standby mode.

Bit 15—Full Address Enable 1 (FAE1): Specifies whether channel 1 is to be used in short address mode or full address mode.

In short address mode, channels 1A and 1B are used as independent channels.

Bit 15

FAE1	Description	
0	Short address mode	(Initial value)
1	Full address mode	

Bit 14—Full Address Enable 0 (FAE0): Specifies whether channel 0 is to be used in short address mode or full address mode.

In short address mode, channels 0A and 0B are used as independent channels.

Bit 14

FAE0	Description	
0	Short address mode	(Initial value)
1	Full address mode	

Bit 13 and 12—Reserved: This bit is reserved and only 0 can be written to, writing 1 causes a malfunction error.

Bits 11 to 8—Data Transfer Acknowledge (DTA): These bits enable or disable clearing, when DMA transfer is performed, of the internal interrupt source selected by the data transfer factor setting.

When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting is cleared automatically by DMA transfer. When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting does not issue an interrupt request to the CPU or DTC.

When DTE = 1 and DTA = 0, the internal interrupt source selected by the data transfer factor setting 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 selected by the data transfer factor setting issues an interrupt request to the CPU or DTC regardless of the DTA bit setting.

Bit 11—Data Transfer Acknowledge 1B (DTA1B): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 1B data transfer factor setting.

Bit 11

DTA1B	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bit 10—Data Transfer Acknowledge 1A (DTA1A): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 1A data transfer factor setting.

Bit 10

DTA1A	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bit 9—Data Transfer Acknowledge 0B (DTA0B): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 0B data transfer factor setting.

Bit 9

DTA0B	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bit 8—Data Transfer Acknowledge 0A (DTA0A): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 0A data transfer factor setting.

Bit 8

DTA0A	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bits 7 to 4—Data Transfer Enable (DTE): When DTE = 0, data transfer is disabled and the activation source selected by the data transfer factor setting is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. 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.

The conditions for the DTE bit being cleared to 0 are as follows:

- 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 abort the transfer, or for a similar reason

When DTE = 1, data transfer is enabled and the DMAC waits for a request by the activation source selected by the data transfer factor setting. When a request is issued by the activation source, DMA transfer is executed.

The condition for the DTE bit being set to 1 is as follows:

- When 1 is written to the DTE bit after the DTE bit is read as 0

Bit 7—Data Transfer Enable 1B (DTE1B): Enables or disables data transfer on channel 1B.

Bit 7		
DTE1B	**Description**	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 6—Data Transfer Enable 1A (DTE1A): Enables or disables data transfer on channel 1A.

Bit 6		
DTE1A	**Description**	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 5—Data Transfer Enable 0B (DTE0B): Enables or disables data transfer on channel 0B.

Bit 5		
DTE0B	**Description**	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 4—Data Transfer Enable 0A (DTE0A): Enables or disables data transfer on channel 0A.

Bit 4		
DTE0A	**Description**	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bits 3 to 0—Data Transfer End Interrupt Enable (DTIE): 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.

Bit 3—Data Transfer Interrupt Enable 1B (DTIE1B): Enables or disables the channel 1B transfer end interrupt.

Bit 3

DTIE1B	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 2—Data Transfer Interrupt Enable 1A (DTIE1A): Enables or disables the channel 1A transfer end interrupt.

Bit 2

DTIE1A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 1—Data Transfer Interrupt Enable 0B (DTIE0B): Enables or disables the channel 0B transfer end interrupt.

Bit 1

DTIE0B	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 0—Data Transfer Interrupt Enable 0A (DTIE0A): Enables or disables the channel 0A transfer end interrupt.

Bit 0

DTIE0A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

7.3 Register Descriptions (2) (Full Address Mode)

Full address mode transfer is performed with channels A and B together. For details of full address mode setting, see table 7-5.

7.3.1 Memory Address Register (MAR)

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

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

*: Undefined

MAR is a 32-bit readable/writable register; MARA functions as the transfer source address register, and MARB as the destination address register.

MAR is composed 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.

MAR is incremented or decremented each time a byte or word transfer is executed, so that the source or destination memory address can be updated automatically. For details, see section 7.3.4, DMA Control Register (DMACR).

MAR is not initialized by a reset or in standby mode.

7.3.2 I/O Address Register (IOAR)

IOAR is not used in full address transfer.

7.3.3 Execute Transfer Count Register (ETCR)

ETCR is a 16-bit readable/writable register that specifies the number of transfers. The function of this register is different in normal mode and in block transfer mode.

ETCR is not initialized by a reset or in standby mode.

(1) Normal Mode

ETCRA

Transfer Counter

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

*: Undefined

In normal mode, ETCRA functions as a 16-bit transfer counter. ETCRA is decremented by 1 each time a transfer is performed, and transfer ends when the count reaches H'0000. ETCRB is not used at this time.

ETCRB

ETCRB is not used in normal mode.

(2) Block Transfer Mode

ETCRA

Holds block size

Bit	:	15	14	13	12	11	10	9	8
ETCRAH	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Block size counter

Bit	:	7	6	5	4	3	2	1	0
ETCRAL	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

ETCRB

Block Transfer Counter

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

In block transfer mode, ETCRAL functions as an 8-bit block size counter and ETCRAH holds the block size. ETCRAL is decremented 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.

ETCRB functions in block transfer mode, 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.

7.3.4 DMA Control Register (DMACR)

DMACR is a 16-bit readable/writable register that controls the operation of each DMAC channel. In full address mode, DMACRA and DMACRB have different functions.

DMACR is initialized to H'0000 by a reset, and in standby mode.

DMACRA

Bit	:	15	14	13	12	11	10	9	8
DMACRA	:	DTSZ	SAID	SAIDE	BLKDIR	BLKE	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DMACRB

Bit	:	7	6	5	4	3	2	1	0
DMACRB	:	—	DAID	DAIDE	—	DTF3	DTF2	DTF1	DTF0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 15—Data Transfer Size (DTSZ): Selects the size of data to be transferred at one time.

Bit 15		
DTSZ	**Description**	
0	Byte-size transfer	(Initial value)
1	Word-size transfer	

Bit 14—Source Address Increment/Decrement (SAID)

Bit 13—Source Address Increment/Decrement Enable (SAIDE): These bits specify whether source address register MARA is to be incremented, decremented, or left unchanged, when data transfer is performed.

Bit 14	Bit 13		
SAID	**SAIDE**	**Description**	
0	0	MARA is fixed	(Initial value)
	1	MARA is incremented after a data transfer • When DTSZ = 0, MARA is incremented by 1 after a transfer • When DTSZ = 1, MARA is incremented by 2 after a transfer	
1	0	MARA is fixed	
	1	MARA is decremented after a data transfer • When DTSZ = 0, MARA is decremented by 1 after a transfer • When DTSZ = 1, MARA is decremented by 2 after a transfer	

Bit 12—Block Direction (BLKDIR)

Bit 11—Block Enable (BLKE): These bits specify whether normal mode or block transfer mode is to be used. If block transfer mode is specified, the BLKDIR bit specifies whether the source side or the destination side is to be the block area.

Bit 12	Bit 11		
BLKDIR	**BLKE**	**Description**	
0	0	Transfer in normal mode	(Initial value)
	1	Transfer in block transfer mode, destination side is block area	
1	0	Transfer in normal mode	
	1	Transfer in block transfer mode, source side is block area	

For operation in normal mode and block transfer mode, see section 7.5, Operation.

Bits 10 to 7—Reserved: Can be read or written to.

Bit 6—Destination Address Increment/Decrement (DAID)

Bit 5—Destination Address Increment/Decrement Enable (DAIDE): These bits specify whether destination address register MARB is to be incremented, decremented, or left unchanged, when data transfer is performed.

Bit 6	Bit 5		
DAID	**DAIDE**	**Description**	
0	0	MARB is fixed	(Initial value)
	1	MARB is incremented after a data transfer • When DTSZ = 0, MARB is incremented by 1 after a transfer • When DTSZ = 1, MARB is incremented by 2 after a transfer	
1	0	MARB is fixed	
	1	MARB is decremented after a data transfer • When DTSZ = 0, MARB is decremented by 1 after a transfer • When DTSZ = 1, MARB is decremented by 2 after a transfer	

Bit 4—Reserved: Can be read or written to.

Bits 3 to 0—Data Transfer Factor (DTF3 to DTF0): These bits select the data transfer factor (activation source). The factors that can be specified differ between normal mode and block transfer mode.

- Normal Mode

Bit 3	Bit 2	Bit 1	Bit 0		
DTF3	**DTF2**	**DTF1**	**DTF0**	**Description**	
0	0	0	0	—	(Initial value)
			1	—	
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input	
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input	
	1	0	*	—	
		1	0	Auto-request (cycle steal)	
			1	Auto-request (burst)	
1	*	*	*	—	

*: Don't care

- Block Transfer Mode

Bit 3	Bit	Bit 1	Bit 0	
DTF3	DTF2	DTF1	DTF0	Description
0	0	0	0	— (Initial value)
			1	—
		1	0	Activated by $\overline{\text{DREQ}}$ pin falling edge input*
			1	Activated by $\overline{\text{DREQ}}$ pin low-level input
	1	0	0	Activated by SCI channel 0 transmission complete interrupt
			1	Activated by SCI channel 0 reception complete interrupt
		1	0	Activated by SCI channel 1 transmission complete interrupt
			1	Activated by SCI channel 1 reception complete interrupt
1	0	0	0	Activated by TPU channel 0 compare match/input capture A interrupt
			1	Activated by TPU channel 1 compare match/input capture A interrupt
		1	0	Activated by TPU channel 2 compare match/input capture A interrupt
			1	—
	1	0	0	—
			1	—
		1	0	—
			1	—

Note: * Detected as a low level in the first transfer after transfer is enabled.

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 7.5.13, DMAC Multi-Channel Operation.

7.3.5 DMA Band Control Register (DMABCR)

Bit	:	15	14	13	12	11	10	9	8
DMABCRH	:	FAE1	FAE0	—	—	DTA1B	DTA1A	DTA0B	DTA0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	:	7	6	5	4	3	2	1	0
DMABCRL	:	DTME1	DTE1	DTME0	DTE0	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DMABCR is a 16-bit readable/writable register that controls the operation of each DMAC channel.

DMABCR is initialized to H'0000 by a reset, and in standby mode.

Bit 15—Full Address Enable 1 (FAE1): Specifies whether channel 1 is to be used in short address mode or full address mode.

In full address mode, channels 1A and 1B are used together as a single channel.

Bit 15

FAE1	Description	
0	Short address mode	(Initial value)
1	Full address mode	

Bit 14—Full Address Enable 0 (FAE0): Specifies whether channel 0 is to be used in short address mode or full address mode.

In full address mode, channels 0A and 0B are used together as a single channel.

Bit 14

FAE0	Description	
0	Short address mode	(Initial value)
1	Full address mode	

Bits 13 and 12—Reserved: This bit is reserved and only 0 can be written to, writing 1 causes a malfunction error.

Bits 11 and 9—Data Transfer Acknowledge (DTA): These bits enable or disable clearing, when DMA transfer is performed, of the internal interrupt source selected by the data transfer factor setting.

When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting is cleared automatically by DMA transfer. When DTE = 1 and DTA = 1, the internal interrupt source selected by the data transfer factor setting does not issue an interrupt request to the CPU or DTC.

When the DTE = 1 and the DTA = 0, the internal interrupt source selected by the data transfer factor setting 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 the DTE = 0, the internal interrupt source selected by the data transfer factor setting issues an interrupt request to the CPU or DTC regardless of the DTA bit setting.

The state of the DTME bit does not affect the above operations.

Bit 11—Data Transfer Acknowledge 1 (DTA1B): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 1 data transfer factor setting.

Bit 11

DTA1B	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bit 9—Data Transfer Acknowledge 0 (DTA0B): Enables or disables clearing, when DMA transfer is performed, of the internal interrupt source selected by the channel 0 data transfer factor setting.

Bit 9

DTA0B	Description
0	Clearing of selected internal interrupt source at time of DMA transfer is disabled (Initial value)
1	Clearing of selected internal interrupt source at time of DMA transfer is enabled

Bits 10 and 8—Reserved (DTA1A, DTA0A): Reserved bits in full address mode. Read and write possible.

Bits 7 and 5—Data Transfer Master Enable (DTME): Together with the DTE bit, these bits control enabling or disabling of data transfer on the relevant channel. When both the DTME bit and the DTE bit are set to 1, transfer is enabled for the channel.

If the relevant channel is in the middle of a burst mode transfer when an NMI interrupt is generated, the DTME bit is cleared, the transfer is interrupted, and bus mastership passes to the CPU. When the DTME bit is subsequently set to 1 again, the interrupted transfer is resumed. In block transfer mode, however, the DTME bit is not cleared by an NMI interrupt, and transfer is not interrupted.

The conditions for the DTME bit being cleared to 0 are as follows:

- When initialization is performed
- When NMI is input in burst mode
- When 0 is written to the DTME bit

The condition for DTME being set to 1 is as follows:

- When 1 is written to DTME after DTME is read as 0

Bit 7—Data Transfer Master Enable 1 (DTME1): Enables or disables data transfer on channel 1.

Bit 7

DTME1	Description
0	Data transfer disabled. In burst mode, cleared to 0 by an NMI interrupt (Initial value)
1	Data transfer enabled

Bit 5—Data Transfer Master Enable 0 (DTME0): Enables or disables data transfer on channel 0.

Bit 5

DTME0	Description
0	Data transfer disabled. In normal mode, cleared to 0 by an NMI interrupt (Initial value)
1	Data transfer enabled

Bits 6 and 4—Data Transfer Enable (DTE): When DTE = 0, data transfer is disabled and the activation source selected by the data transfer factor setting is ignored. If the activation source is an internal interrupt, an interrupt request is issued to the CPU or DTC. 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.

The conditions for the DTE bit being cleared to 0 are as follows:

- When initialization is performed
- When the specified number of transfers have been completed
- When 0 is written to the DTE bit to forcibly abort the transfer, or for a similar reason

When DTE = 1 and DTME = 1, data transfer is enabled and the DMAC waits for a request by the activation source selected by the data transfer factor setting. When a request is issued by the activation source, DMA transfer is executed.

The condition for the DTE bit being set to 1 is as follows:

- When 1 is written to the DTE bit after the DTE bit is read as 0

Bit 6—Data Transfer Enable 1 (DTE1): Enables or disables data transfer on channel 1.

Bit 6

DTE1	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bit 4—Data Transfer Enable 0 (DTE0): Enables or disables data transfer on channel 0.

Bit 4

DTE0	Description	
0	Data transfer disabled	(Initial value)
1	Data transfer enabled	

Bits 3 and 1—Data Transfer Interrupt Enable B (DTIEB): These bits enable or disable an interrupt to the CPU or DTC when transfer is interrupted. If the DTIEB bit is set to 1 when DTME = 0, 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 DTIEB bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the DTME bit to 1.

Bit 3—Data Transfer Interrupt Enable 1B (DTIE1B): Enables or disables the channel 1 transfer break interrupt.

Bit 3

DTIE1B	Description	
0	Transfer break interrupt disabled	(Initial value)
1	Transfer break interrupt enabled	

Bit 1—Data Transfer Interrupt Enable 0B (DTIE0B): Enables or disables the channel 0 transfer break interrupt.

Bit 1

DTIE0B	Description	
0	Transfer break interrupt disabled	(Initial value)
1	Transfer break interrupt enabled	

Bits 2 and 0—Data Transfer End Interrupt Enable A (DTIEA): These bits enable or disable an interrupt to the CPU or DTC when transfer ends. If DTIEA 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 DTIEA 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.

Bit 2—Data Transfer Interrupt Enable 1A (DTIE1A): Enables or disables the channel 1 transfer end interrupt.

Bit 2

DTIE1A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 0—Data Transfer Interrupt Enable 0A (DTIE0A): Enables or disables the channel 0 transfer end interrupt.

Bit 0

DTIE0A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

7.4 Register Descriptions (3)

7.4.1 DMA Write Enable Register (DMAWER)

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 reactivate the DTC. DMAWER applies restrictions so that only specific bits of DMACR for the specific channel and also DMATCR and DMABCR can be changed to prevent inadvertent changes being made to registers other than those for the channel concerned. The restrictions applied by DMAWER are valid for the DTC.

Figure 7-2 shows the transfer areas for activating the DTC with a channel 0A transfer end interrupt, and reactivating channel 0A. The address register and count register area is re-set by the first DTC transfer, then the control register area is re-set by the second DTC chain transfer.

When re-setting the control register area, perform masking by setting bits in DMAWER to prevent modification of the contents of the other channels.

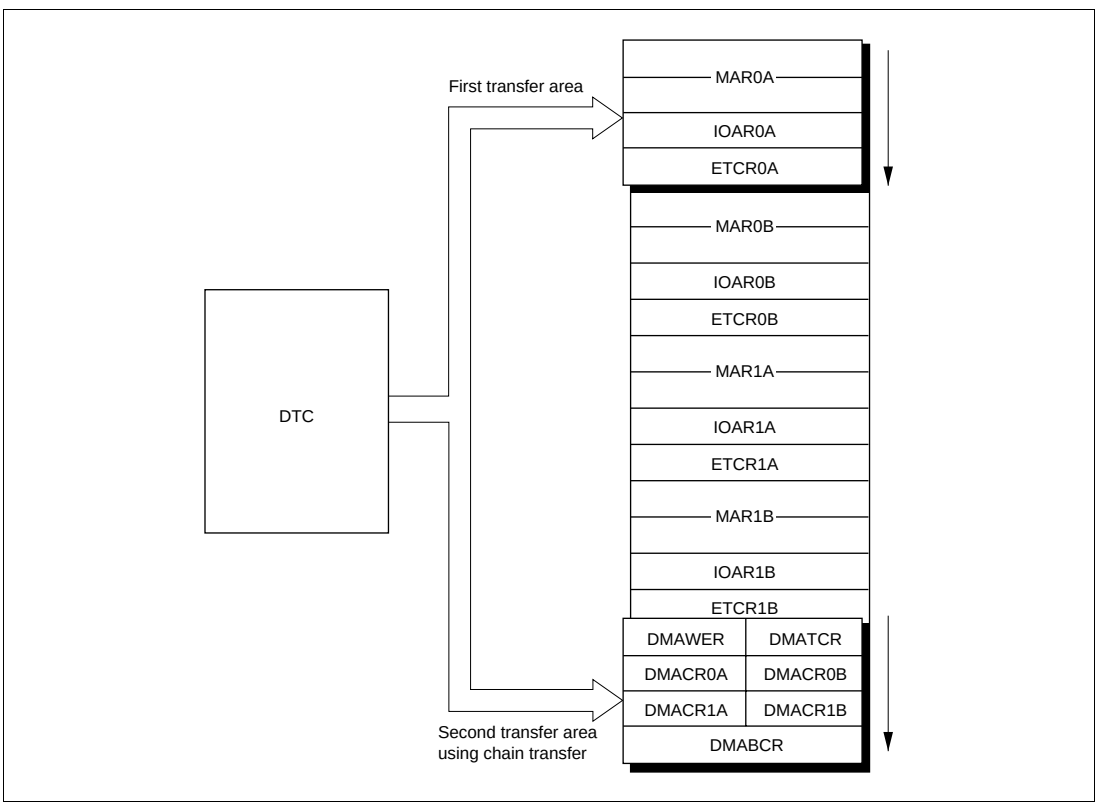


Figure 7-2 Areas for Register Re-Setting by DTC (Example: Channel 0A)

Bit	:	7	6	5	4	3	2	1	0
DMAWER	:	—	—	—	—	WE1B	WE1A	WE0B	WE0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

DMAWER is an 8-bit readable/writable register that controls enabling or disabling of writes to the DMACR, DMABCR, and DMATCR by the DTC.

DMAWER is initialized to H'00 by a reset, and in standby mode.

Bits 7 to 4—Reserved: Read-only bits, always read as 0.

Bit 3—Write Enable 1B (WE1B): Enables or disables writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR by the DTC.

Bit 3

WE1B	Description
0	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR are disabled (Initial value)
1	Writes to all bits in DMACR1B, bits 11, 7, and 3 in DMABCR, and bit 5 in DMATCR are enabled

Bit 2—Write Enable 1A (WE1A): Enables or disables writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR by the DTC.

Bit 2

WE1A	Description
0	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are disabled (Initial value)
1	Writes to all bits in DMACR1A, and bits 10, 6, and 2 in DMABCR are enabled

Bit 1—Write Enable 0B (WE0B): Enables or disables writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR.

Bit 1

WE0B	Description
0	Writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR are disabled (Initial value)
1	Writes to all bits in DMACR0B, bits 9, 5, and 1 in DMABCR, and bit 4 in DMATCR are enabled

Bit 0—Write Enable 0A (WE0A): Enables or disables writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR.

Bit 0

WE0A	Description
0	Writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR are disabled (Initial value)
1	Writes to all bits in DMACR0A, and bits 8, 4, and 0 in DMABCR are enabled

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 are always write-enabled regardless of the DMAWER settings. When modifying these registers, the channel for which the modification is to be made should be halted.

7.4.2 DMA Terminal Control Register (DMATCR)

Bit	:	7	6	5	4	3	2	1	0
DMATCR	:	—	—	TEE1	TEE0	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	—	—	—	—

DMATCR is an 8-bit readable/writable register that controls enabling or disabling of DMAC transfer end pin output. A port can be set for output automatically, and a transfer end signal output, by setting the appropriate bit.

DMATCR is initialized to H'00 by a reset, and in standby mode.

Bits 7 and 6—Reserved: Read-only bits, always read as 0.

Bit 5—Transfer End Enable 1 (TEE1): Enables or disables transfer end pin 1 ($\overline{\text{TEND1}}$) output.

Bit 5

TEE1	Description
0	$\overline{\text{TEND1}}$ pin output disabled (Initial value)
1	$\overline{\text{TEND1}}$ pin output enabled

Bit 4—Transfer End Enable 0 (TEE0): Enables or disables transfer end pin 0 ($\overline{\text{TEND0}}$) output.

Bit 4	
TEE0	**Description**
0	$\overline{\text{TEND0}}$ pin output disabled (Initial value)
1	$\overline{\text{TEND0}}$ pin output enabled

The $\overline{\text{TEND}}$ pins are assigned only to channel B in short address mode.

The transfer end signal indicates the transfer cycle in which the transfer counter reached 0, regardless of the transfer source. An exception is block transfer mode, in which the transfer end signal indicates the transfer cycle in which the block counter reached 0.

Bits 3 to 0—Reserved: Read-only bits, always read as 0.

7.4.3 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRA is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPA7 bit in MSTPCR is set to 1, the DMAC operation stops at the end of the bus cycle and a transition is made to module stop mode. For details, see section 17.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Bit 7—Module Stop (MSTP7): Specifies the DMAC module stop mode.

Bits 7	
MSTPA7	**Description**
0	DMAC module stop mode cleared (Initial value)
1	DMAC module stop mode set

7.5 Operation

7.5.1 Transfer Modes

Table 7-6 lists the DMAC modes.

Table 7-6 DMAC Transfer Modes

Transfer Mode			Transfer Source	Remarks
Short address mode	Dual address mode	(1) Sequential mode (2) Idle mode (3) Repeat mode	• TPU channel 0 to 2 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • External request	• Up to 4 channels can operate independently • External request applies to channel B only
Full address mode		(4) Normal mode	• External request • Auto-request	• Max. 2-channel operation, combining channels A and B • With auto-request, burst mode transfer or cycle steal transfer can be selected
		(5) Block transfer mode	• TPU channel 0 to 2 compare match/input capture A interrupt • SCI transmission complete interrupt • SCI reception complete interrupt • External request	

Operation in each mode is summarized below.

(1) Sequential mode

In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. An interrupt request can be sent to the CPU or DTC when the specified number of transfers have been completed. One address is specified as 24 bits, and the other as 16 bits. The transfer direction is programmable.

(2) Idle mode

In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. An interrupt request can be sent to the CPU or DTC when the specified number of transfers have been completed. One address is specified as 24 bits, and the other as 16 bits. The transfer source address and transfer destination address are fixed. The transfer direction is programmable.

(3) Repeat mode

In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. When the specified number of transfers have been completed, the addresses and transfer counter are restored to their original settings, and operation is continued. No interrupt request is sent to the CPU or DTC. One address is specified as 24 bits, and the other as 16 bits. The transfer direction is programmable.

(4) Normal mode

- Auto-request

  By means of register settings only, the DMAC is activated, and transfer continues until the specified number of transfers have been completed. An interrupt request can be sent to the CPU or DTC when transfer is completed. Both addresses are specified as 24 bits.

  — Cycle steal mode: The bus is released to another bus master every byte or word transfer.

  — Burst mode: The bus is held and transfer continued until the specified number of transfers have been completed.

- External request

  In response to a single transfer request, the specified number of transfers are carried out, one byte or one word at a time. An interrupt request can be sent to the CPU or DTC when the specified number of transfers have been completed. Both addresses are specified as 24 bits.

(5) Block transfer mode

In response to a single transfer request, a block transfer of the specified block size is carried out. This is repeated the specified number of times, once each time there is a transfer request. At the end of each single block transfer, one address is restored to its original setting. An interrupt request can be sent to the CPU or DTC when the specified number of block transfers have been completed. Both addresses are specified as 24 bits.

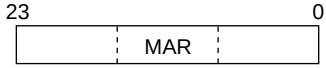
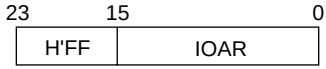
7.5.2 Sequential Mode

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 7-7 summarizes register functions in sequential mode.

Table 7-7 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/ decremented 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

Legend

MAR : Memory address register
IOAR : I/O address register
ETCR : Transfer count register
DTDIR : Data transfer direction bit

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.

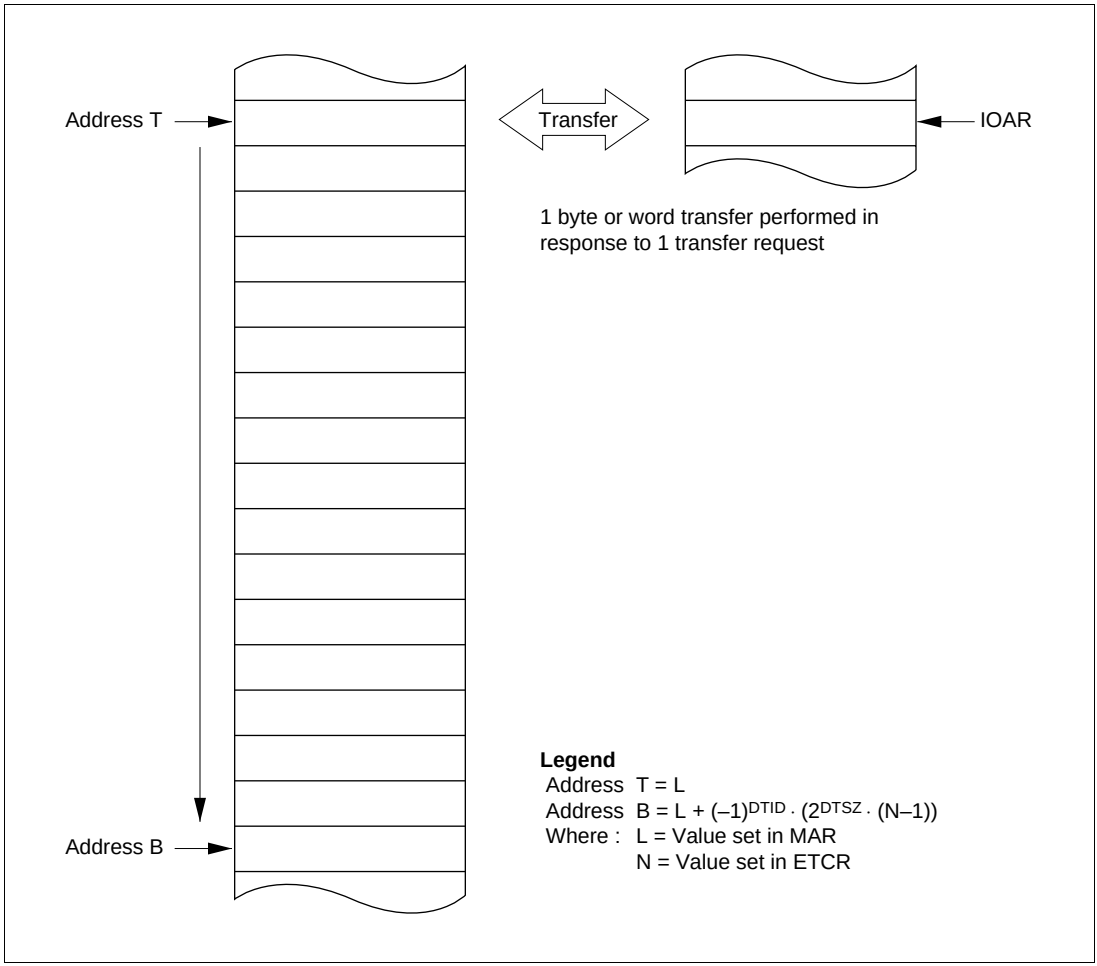


Figure 7-3 Operation in Sequential Mode

The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and 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 external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 2 compare match/input capture A interrupts. External requests can be set for channel B only.

Figure 7-4 shows an example of the setting procedure for sequential mode.

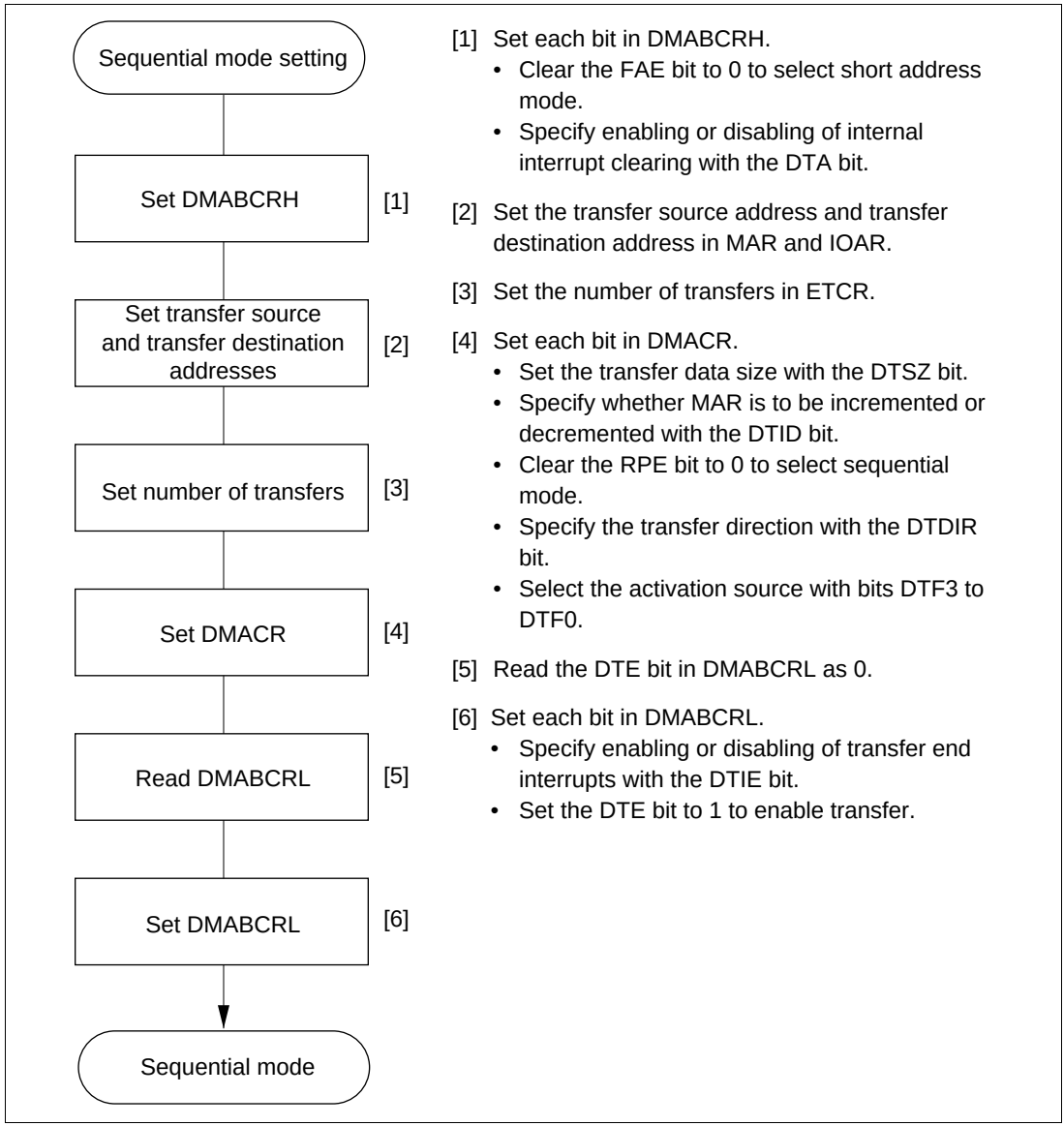


Figure 7-4 Example of Sequential Mode Setting Procedure

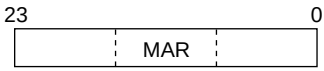
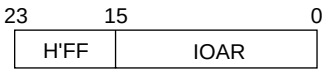
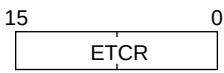
7.5.3 Idle Mode

Idle mode can be specified by setting the RPE bit and DTIE bit in DMACR 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 7-8 summarizes register functions in idle mode.

Table 7-8 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

Legend

MAR : Memory address register
IOAR : I/O address register
ETCR : Transfer count register
DTDIR : Data transfer direction bit

MAR specifies the start address of the transfer source or transfer destination as 24 bits. MAR is neither incremented nor decremented 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 7-5 illustrates operation in idle mode.

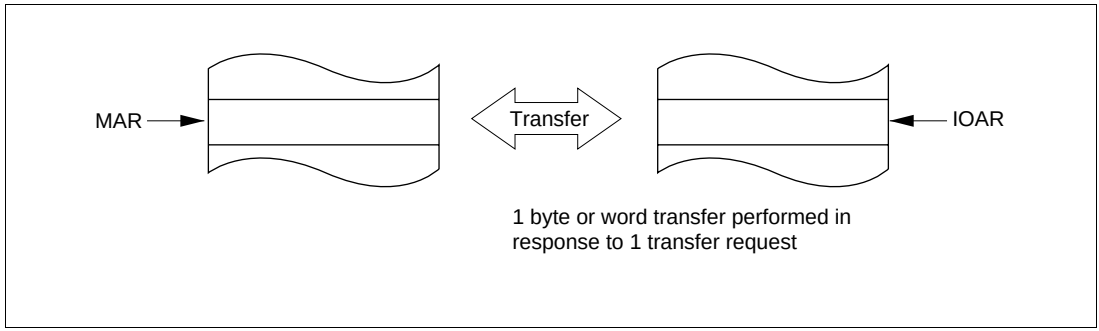


Figure 7-5 Operation in Idle Mode

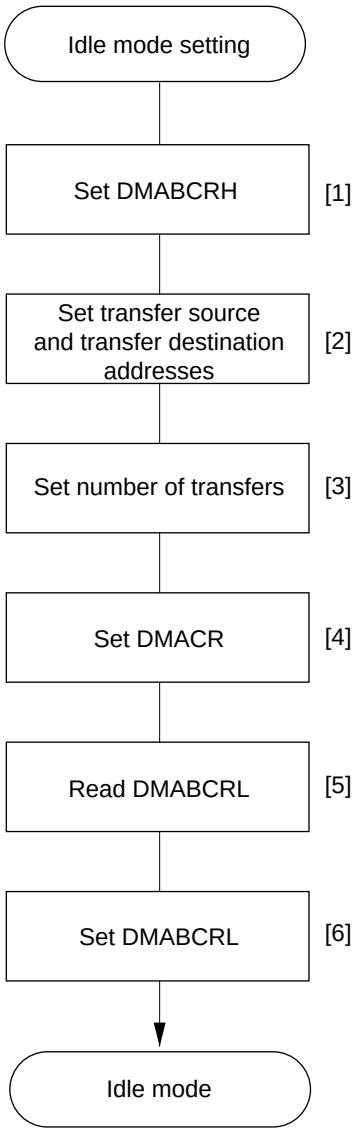
The number of transfers is specified as 16 bits in ETCR. ETCR is decremented by 1 each time a transfer is executed, and when its value reaches H'0000, the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this time, an interrupt request is sent to the CPU or DTC.

The maximum number of transfers, when H'0000 is set in ETCR, is 65,536.

Transfer requests (activation sources) consist of external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 2 compare match/input capture A interrupts. External requests can be set for channel B only.

When the DMAC is used in single address mode, only channel B can be set.

Figure 7-6 shows an example of the setting procedure for idle mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in ETCR.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Set the DTIE bit to 1.
- Set the DTE bit to 1 to enable transfer.

Figure 7-6 Example of Idle Mode Setting Procedure

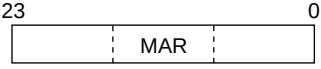
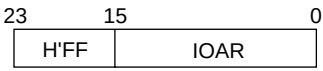
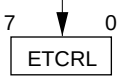
7.5.4 Repeat Mode

Repeat mode can be specified by setting the RPE bit in DMACR to 1, and clearing the DTIE bit 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 ETCR. 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 7-9 summarizes register functions in repeat mode.

Table 7-9 Register Functions in Repeat Mode

Register	Function: DTDIR = 0	Function: DTDIR = 1	Initial Setting	Operation
	Source address register	Destination address register	Start address of transfer destination or transfer source	Incremented/ decremented 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

Legend

MAR : Memory address register
IOAR : I/O address register
ETCR : Transfer count register
DTDIR : Data transfer direction bit

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.

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 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 restored in accordance with the values of the DTSZ and DTID bits in DMACR. The MAR restoration operation is as shown below.

$$MAR = MAR - (-1)^{DTID} \cdot 2^{DTSZ} \cdot ETCRH$$

The same value should be set in ETCRH and ETCRL.

In repeat mode, operation continues until the DTE bit is cleared. To end the transfer operation, therefore, you should clear the DTE bit 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.

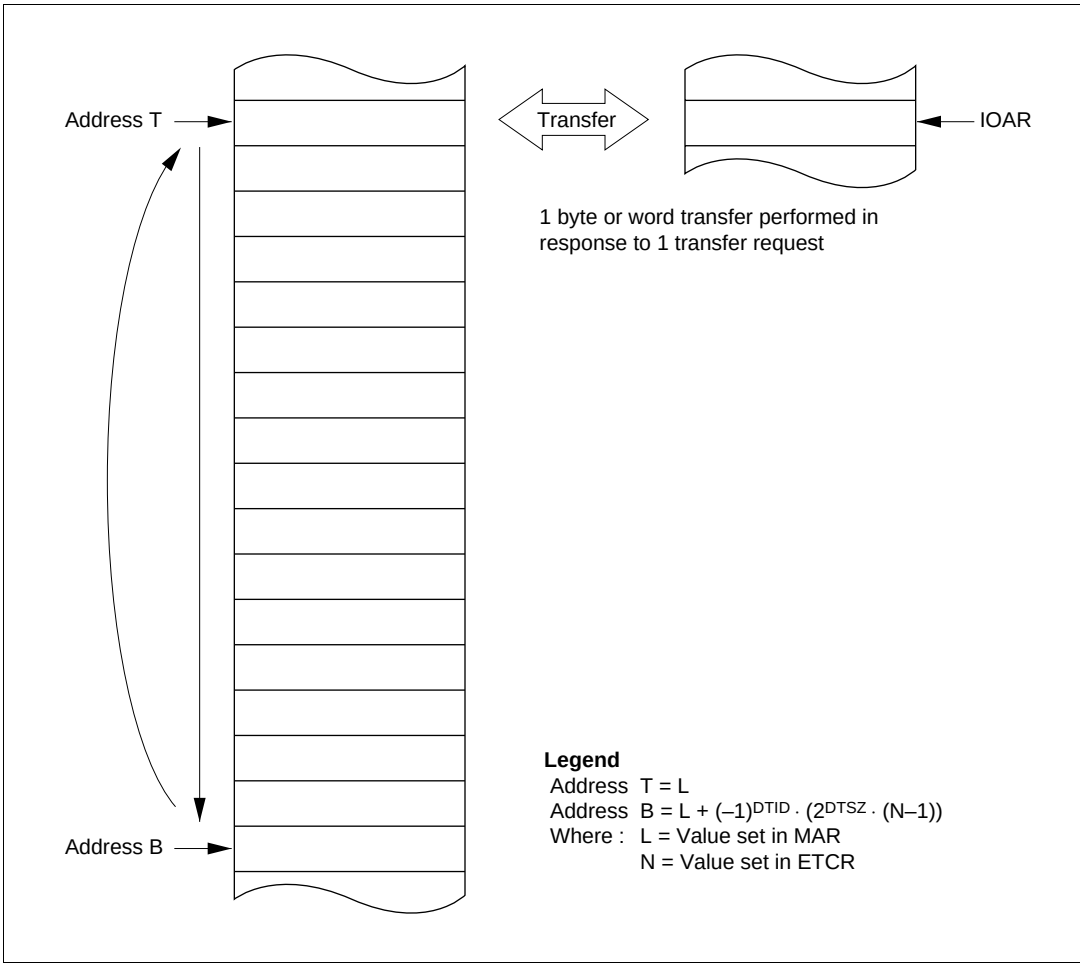
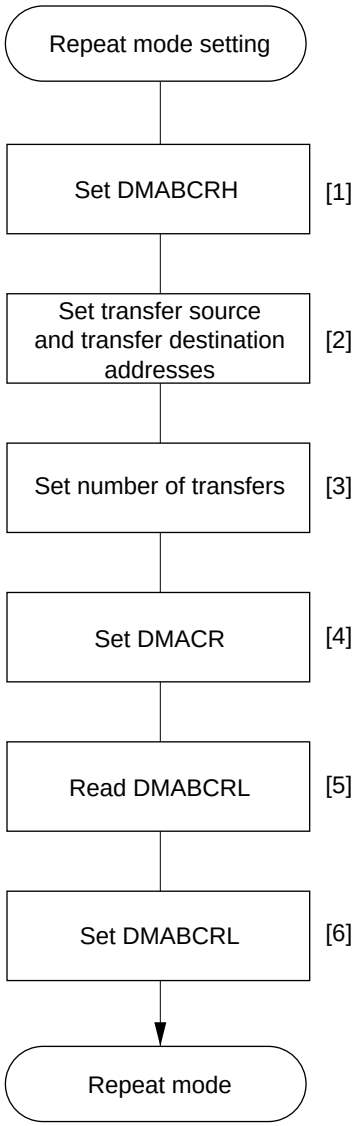


Figure 7-7 Operation in Repeat mode

Transfer requests (activation sources) consist of external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 2 compare match/input capture A interrupts. External requests can be set for channel B only.

Figure 7-8 shows an example of the setting procedure for repeat mode.



[1] Set each bit in DMABCRH.
- Clear the FAE bit to 0 to select short address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address and transfer destination address in MAR and IOAR.

[3] Set the number of transfers in both ETCRH and ETCRL.

[4] Set each bit in DMACR.
- Set the transfer data size with the DTSZ bit.
- Specify whether MAR is to be incremented or decremented with the DTID bit.
- Set the RPE bit to 1.
- Specify the transfer direction with the DTDIR bit.
- Select the activation source with bits DTF3 to DTF0.

[5] Read the DTE bit in DMABCRL as 0.

[6] Set each bit in DMABCRL.
- Clear the DTIE bit to 0.
- Set the DTE bit to 1 to enable transfer.

Figure 7-8 Example of Repeat Mode Setting Procedure

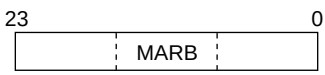
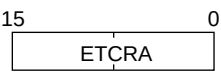
7.5.5 Normal Mode

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 DMABCR to 1 and clearing the BLKE bit in DMACRA to 0.

In normal 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 ETCRA. The transfer source is specified by MARA, and the transfer destination by MARB.

Table 7-10 summarizes register functions in normal mode.

Table 7-10 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	Decremented every transfer; transfer ends when count reaches H'0000

Legend

MARA : Memory address register A
MARB : Memory address register B
ETCRA : Transfer count register A

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 each time a transfer is performed, and when its value reaches H'0000 the DTE bit is cleared and 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 ETCRA, is 65,536.

Figure 7-9 illustrates operation in normal mode.

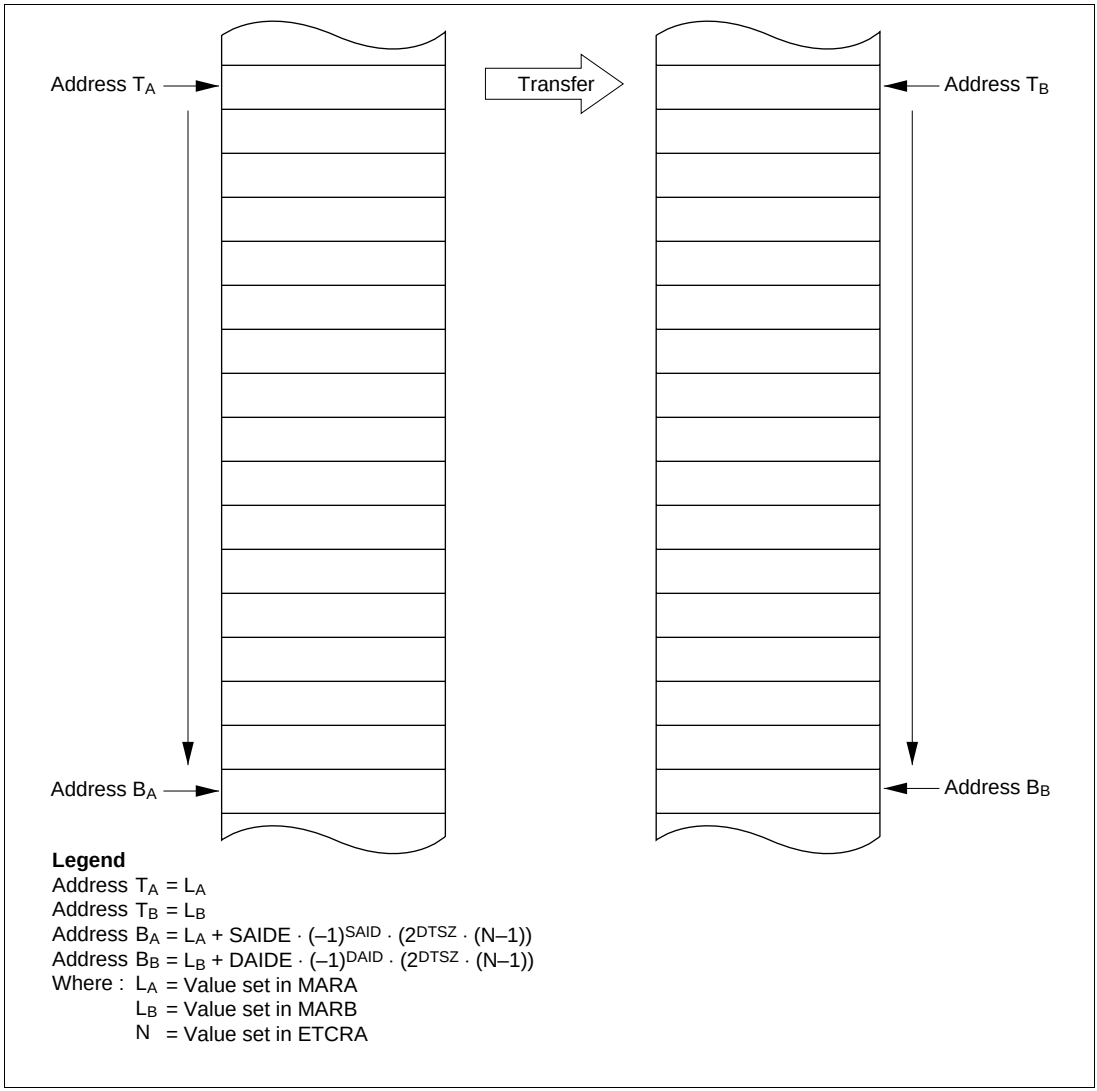


Figure 7-9 Operation in Normal Mode

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.

For setting details, see section 7.3.4, DMA Controller Register (DMACR).

Figure 7-10 shows an example of the setting procedure for normal mode.

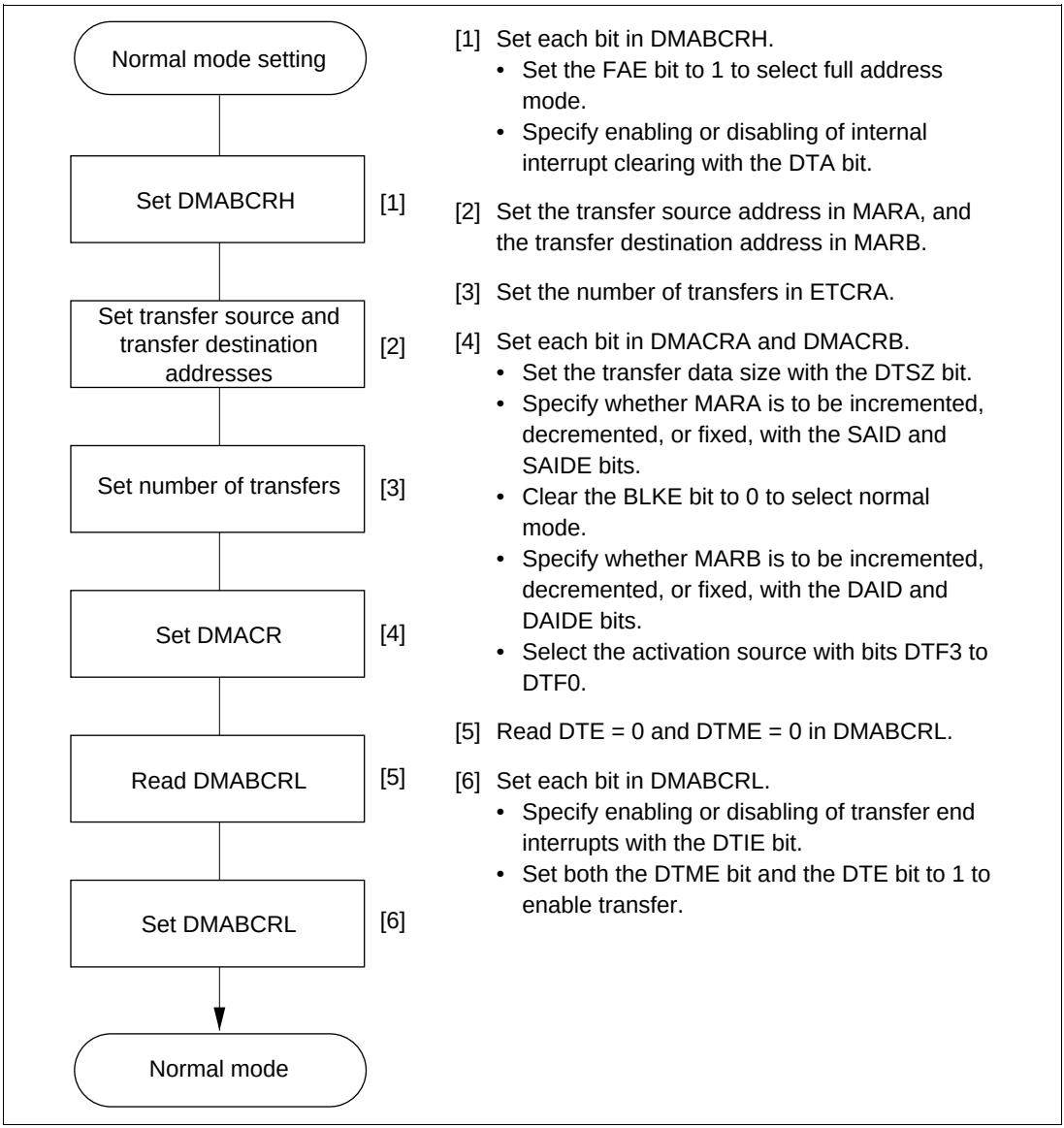


Figure 7-10 Example of Normal Mode Setting Procedure

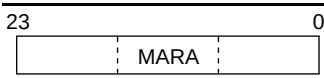
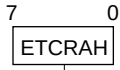
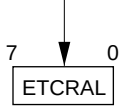
7.5.6 Block Transfer Mode

In block transfer mode, transfer is performed with channels A and B used in combination. Block transfer mode can be specified by setting the FAE bit in DMABCR and the BLKE bit in DMACRA to 1.

In block transfer mode, a transfer of the specified block size is carried out in response to a single transfer request, and this is executed the specified number of times. 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 7-11 summarizes register functions in block transfer mode.

Table 7-11 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	Decremented every transfer; ETCRH value copied when count reaches H'00
	Block transfer counter	Number of block transfers	Decremented every block transfer; transfer ends when count reaches H'0000

Legend

MARA : Memory address register A
MARB : Memory address register B
ETCRA : Transfer count register A
ETCRB : Transfer count register B

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 7-11 illustrates operation in block transfer mode when MARB is designated as a block area.

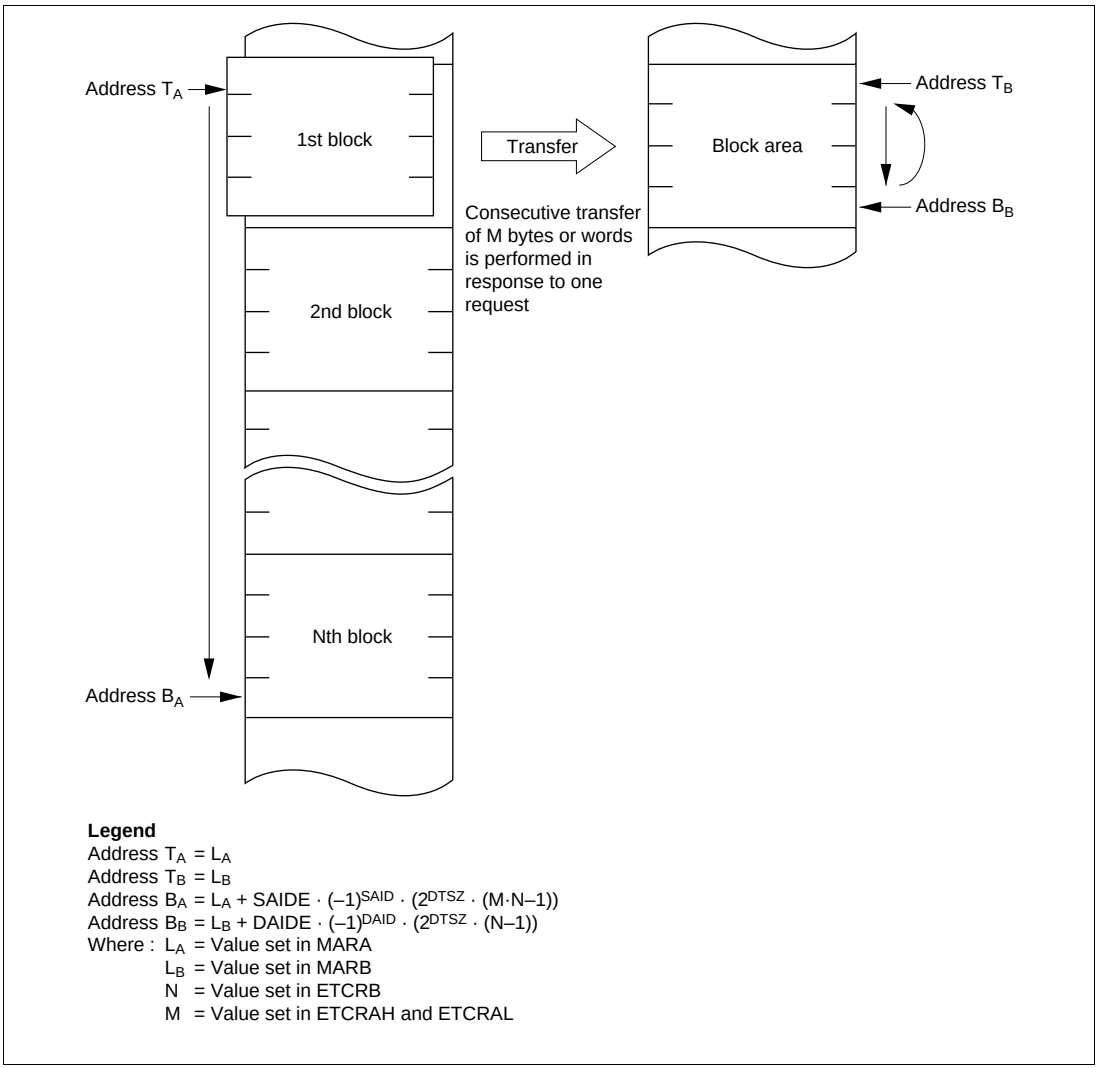


Figure 7-11 Operation in Block Transfer Mode (BLKDIR = 0)

Figure 7-12 illustrates operation in block transfer mode when MARA is designated as a block area.

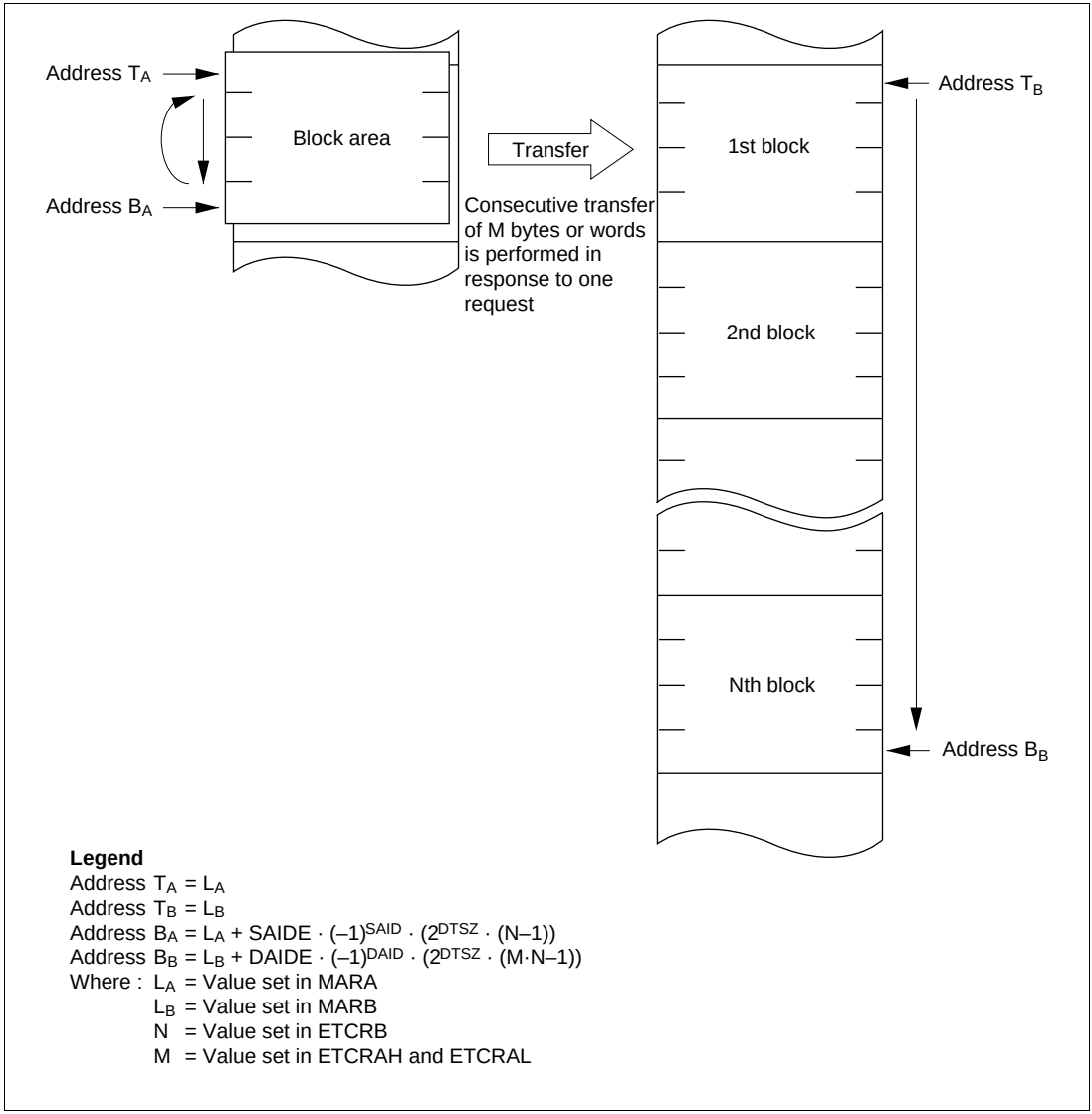


Figure 7-12 Operation in Block Transfer Mode (BLKDIR = 1)

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.

ETCRB is decremented by 1 every block transfer, and when the count reaches H'0000 the DTE bit is cleared and transfer ends. If the DTIE bit is set to 1 at this point, an interrupt request is sent to the CPU or DTC.

Figure 7-13 shows the operation flow in block transfer mode.

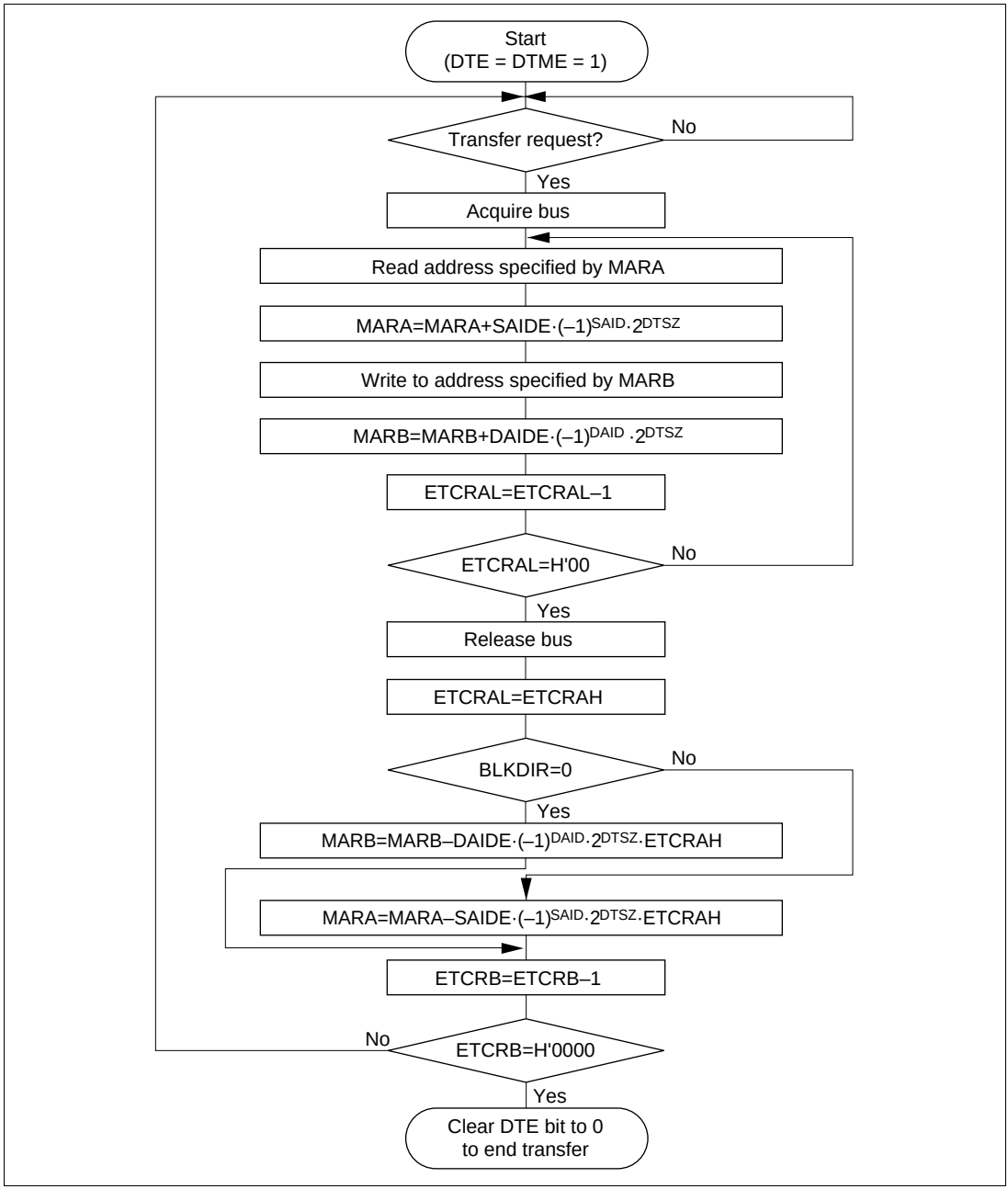
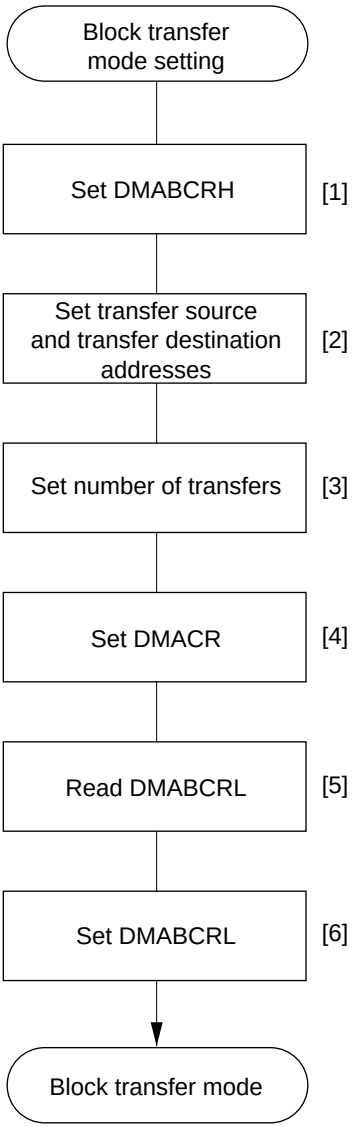


Figure 7-13 Operation Flow in Block Transfer Mode

Transfer requests (activation sources) consist of external requests, SCI transmission complete and reception complete interrupts, and TPU channel 0 to 2 compare match/input capture A interrupts.

For details, see section 7.3.4, DMA Control Register (DMACR).

Figure 7-14 shows an example of the setting procedure for block transfer mode.



[1] Set each bit in DMABCRH.
- Set the FAE bit to 1 to select full address mode.
- Specify enabling or disabling of internal interrupt clearing with the DTA bit.

[2] Set the transfer source address in MARA, and the transfer destination address in MARB.

[3] Set the block size in both ETCRAH and ETCRAL. Set the number of transfers in ETCRB.

[4] Set each bit in DMACRA and DMACRB.
- Set the transfer data size with the DTSZ bit.
- Specify whether MARA is to be incremented, decremented, or fixed, with the SAID and SAIDE bits.
- Set the BLKE bit to 1 to select block transfer mode.
- Specify whether the transfer source or the transfer destination is a block area with the BLKDIR bit.
- Specify whether MARB is to be incremented, decremented, or fixed, with the DAID and DAIDE bits.
- Select the activation source with bits DTF3 to DTF0.

[5] Read DTE = 0 and DTME = 0 in DMABCRL.

[6] Set each bit in DMABCRL.
- Specify enabling or disabling of transfer end interrupts to the CPU with the DTIE bit.
- Set both the DTME bit and the DTE bit to 1 to enable transfer.

Figure 7-14 Example of Block Transfer Mode Setting Procedure

7.5.7 DMAC Activation Sources

DMAC activation sources consist of internal interrupts, external requests, and auto-requests. The activation sources that can be specified depend on the transfer mode and the channel, as shown in table 7-12.

Table 7-12 DMAC Activation Sources

		Short Address Mode		Full Address Mode	
Activation Source		**Channels 0A and 1A**	**Channels 0B and 1B**	**Normal Mode**	**Block Transfer Mode**
Internal Interrupts	TXI0	○	○	X	○
	RXI0	○	○	X	○
	TXI1	○	○	X	○
	RXI1	○	○	X	○
	TGI0A	○	○	X	○
	TGI1A	○	○	X	○
	TGI2A	○	○	X	○
External Requests	$\overline{\text{DREQ}}$ pin falling edge input	X	○	○	○
	$\overline{\text{DREQ}}$ pin low-level input	X	○	○	○
Auto-request		X	X	○	X

Legend

○: Can be specified

X : Cannot be specified

Activation by Internal Interrupt: An interrupt request selected as a DMAC activation source can be sent simultaneously to the CPU and DTC. For details, see section 5, Interrupt Controller.

With activation by an internal interrupt, the DMAC accepts the request independently of the interrupt controller. Consequently, interrupt controller priority settings are not accepted.

If the DMAC is activated by a CPU interrupt source or an interrupt source that is not used as a DTC activation source (DTA = 1), the interrupt source flag is cleared automatically by the DMA transfer. With TXI and RXI interrupts, however, the interrupt source flag is not cleared unless the prescribed 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 first. Transfer requests for other channels are held pending in the DMAC, and activation is carried out in order of priority.

When DTE = 0, such as after completion of a transfer, a request from the selected activation source is not sent to the DMAC, regardless of the DTA bit. 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.

Activation by External Request: If an external request ($\overline{\text{DREQ}}$ pin) is specified as an 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 (short address mode or full address mode) is described below.

When edge sensing is selected, a 1-byte or 1-word transfer is executed each time a high-to-low transition is detected on the $\overline{\text{DREQ}}$ pin. The next transfer may not be performed if the next edge is input before transfer is completed.

When level sensing is selected, the DMAC stands by for a transfer request while the $\overline{\text{DREQ}}$ pin is held high. While the $\overline{\text{DREQ}}$ pin is held low, transfers continue in succession, with the bus being released each time a byte or word is transferred. If the $\overline{\text{DREQ}}$ pin goes high in the middle of a transfer, the transfer is interrupted and the DMAC stands by for a transfer request.

Activation by Auto-Request: Auto-request activation is performed 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 usually alternate.

In burst mode, the DMAC keeps possession of the bus until the end of the transfer, and transfer is performed continuously.

7.5.8 Basic DMAC Bus Cycles

An example of the basic DMAC bus cycle timing is shown in figure 7-15. 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 with CPU cycles, DMA cycles conform to the bus controller settings.

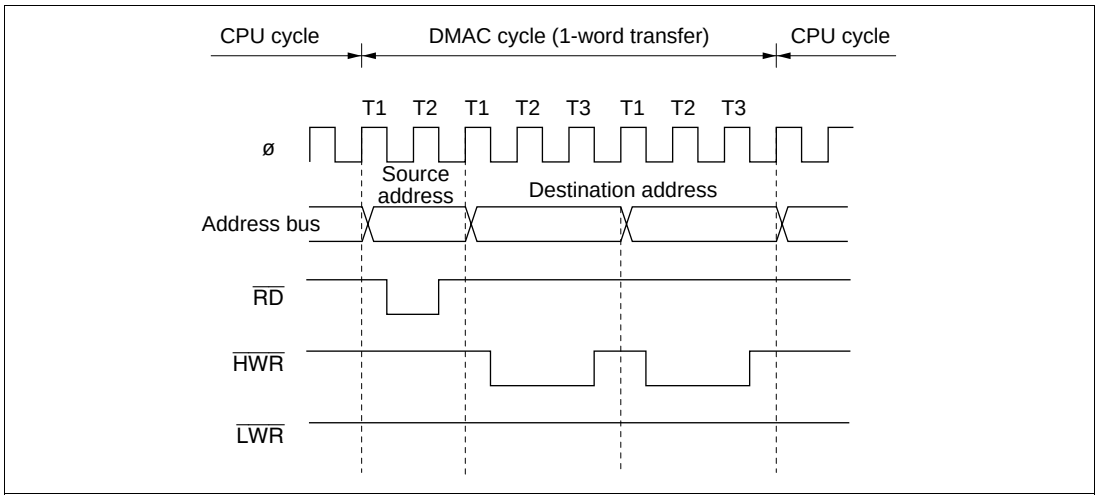


Figure 7-15 Example of DMA Transfer Bus Timing

The address is not output to the external address bus in an access to on-chip memory or an internal I/O register.

7.5.9 DMAC Bus Cycles (Dual Address Mode)

Short Address Mode: Figure 7-16 shows a transfer example in which $\overline{\text{TEND}}$ 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.

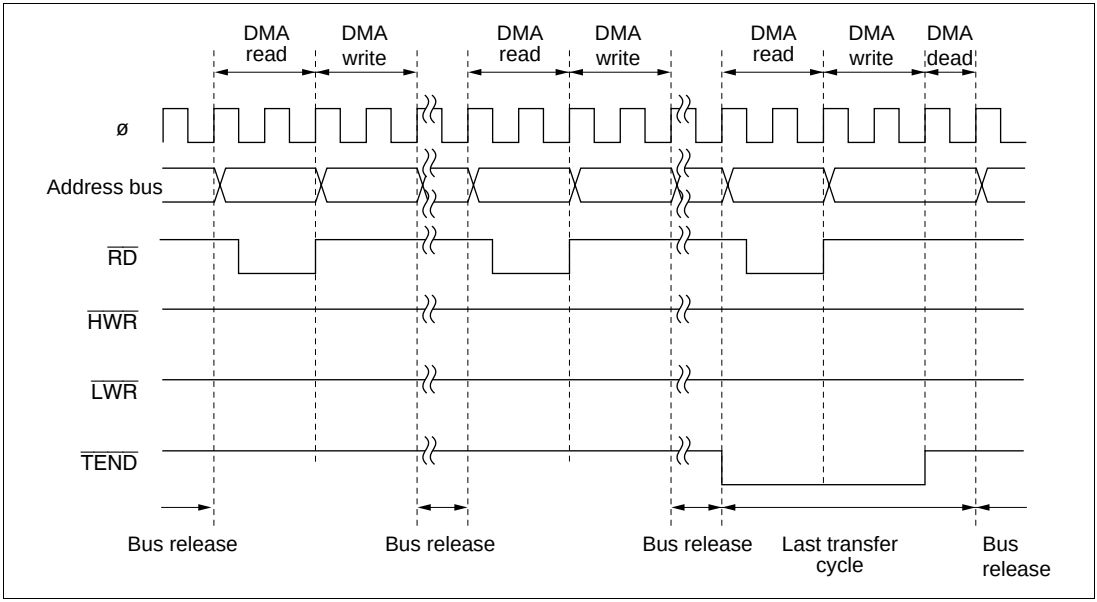


Figure 7-16 Example of Short Address Mode Transfer

A one-byte or one-word transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released one or more bus cycles are inserted 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 cycle in which the transfer counter reaches 0.

Full Address Mode (Cycle Steal Mode): Figure 7-17 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.

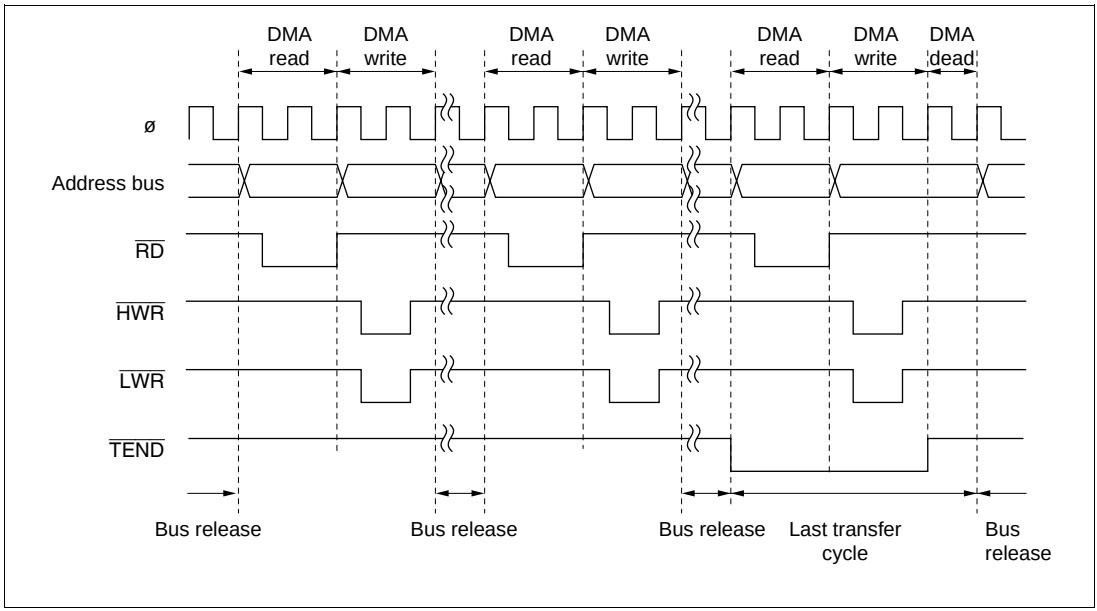


Figure 7-17 Example of Full Address Mode (Cycle Steal) Transfer

A one-byte or one-word transfer is performed, and after the transfer the bus is released. While the bus is released one bus cycle is inserted 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 7-18 shows a transfer example in which $\overline{\text{TEND}}$ 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.

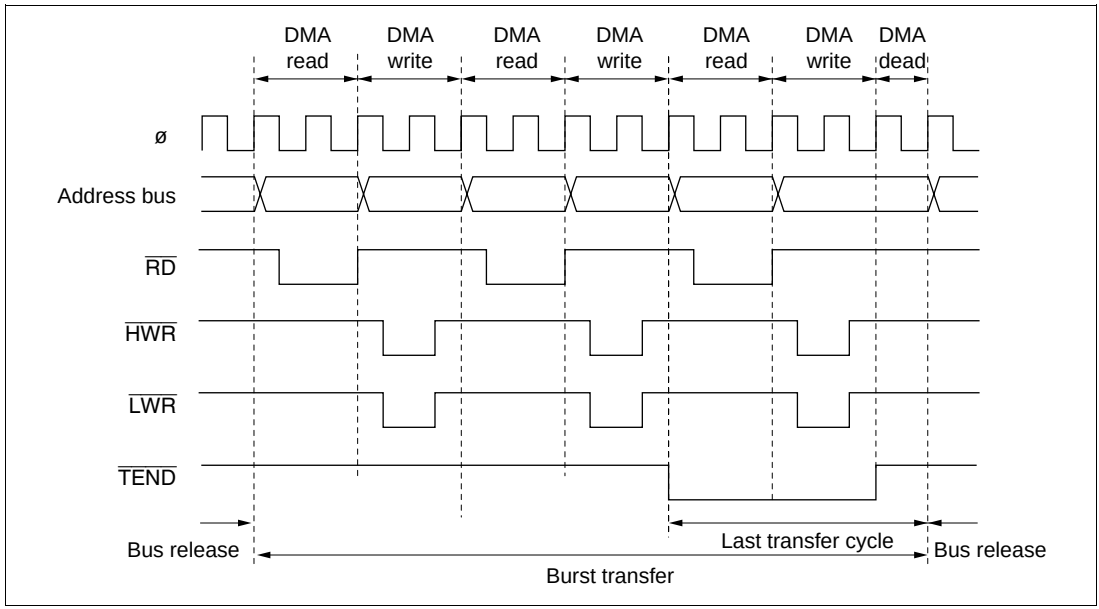


Figure 7-18 Example of Full Address Mode (Burst Mode) Transfer

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 is generated while a channel designated for burst transfer is in the transfer enabled state, the DTME bit 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 7-19 shows a transfer example in which $\overline{\text{TEND}}$ 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.

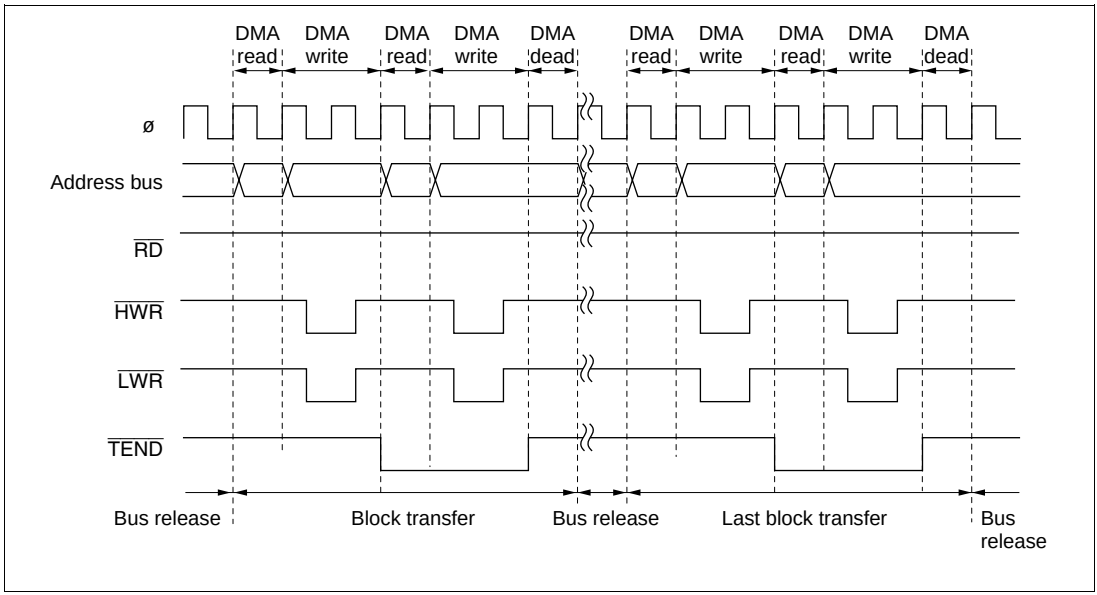


Figure 7-19 Example of Full Address Mode (Block Transfer Mode) Transfer

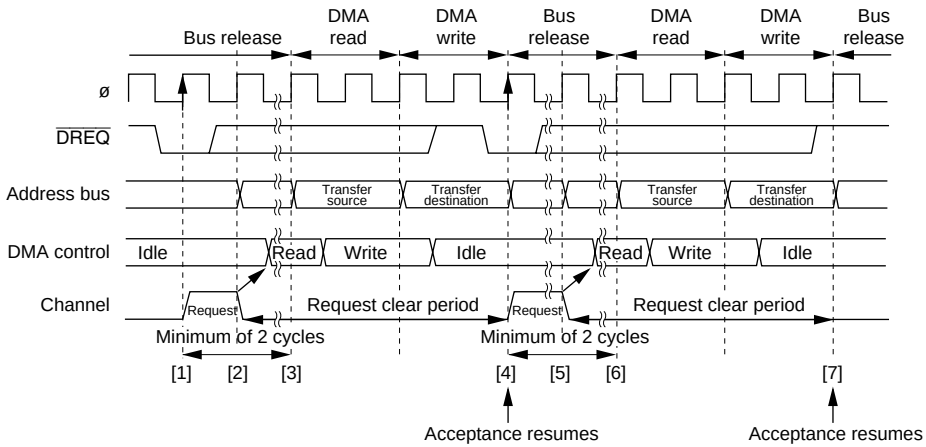
A one-block transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released, one or more bus cycles are inserted by the CPU or DTC.

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.

One block is transmitted without interruption. NMI generation does not affect block transfer operation.

$\overline{\text{DREQ}}$ Pin Falling Edge Activation Timing: Set the DTA bit for the channel for which the $\overline{\text{DREQ}}$ pin is selected to 1.

Figure 7-20 shows an example of $\overline{\text{DREQ}}$ pin falling edge activated normal mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, 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 ø 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 ø, and the request is held.)

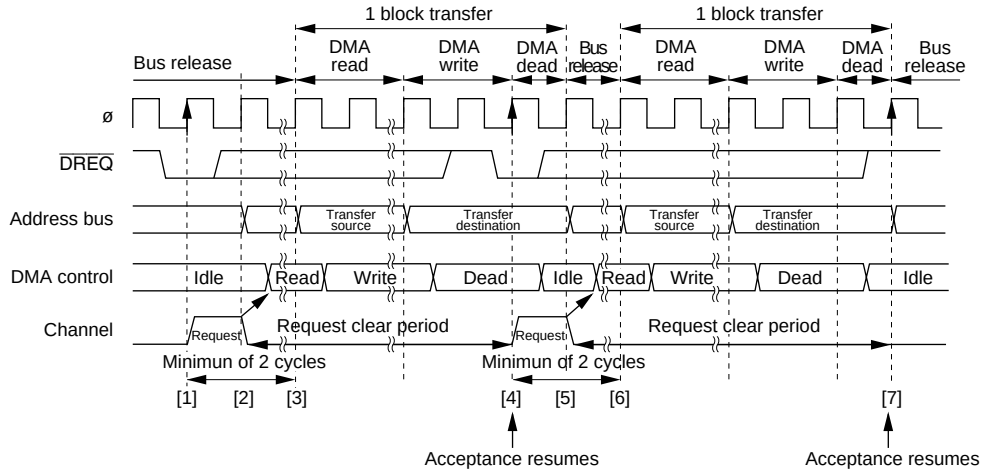
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7-20 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 ø 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 7-21 shows an example of $\overline{\text{DREQ}}$ pin falling edge activated block transfer mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, 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 ø 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 ø, and the request is held.)

Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

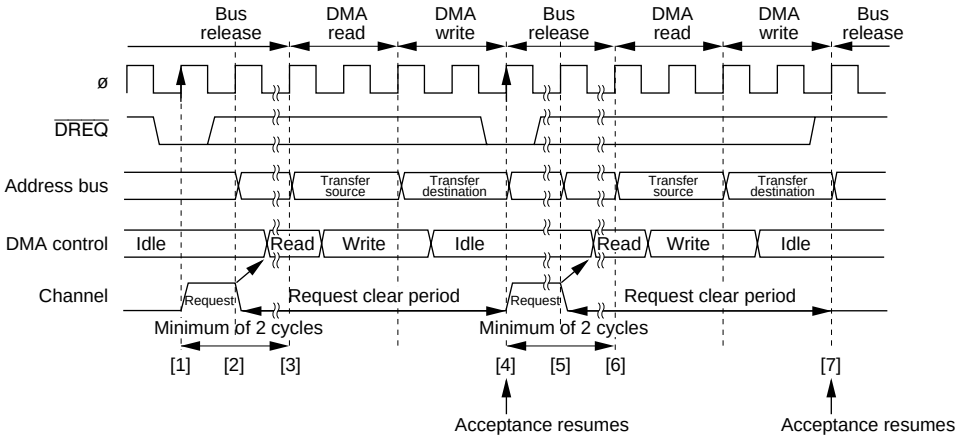
Figure 7-21 Example of $\overline{\text{DREQ}}$ Pin Falling Edge Activated Block Transfer Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ø 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.

$\overline{\text{DREQ}}$ Level Activation Timing (Normal Mode): Set the DTA bit for the channel for which the $\overline{\text{DREQ}}$ pin is selected to 1.

Figure 7-22 shows an example of $\overline{\text{DREQ}}$ level activated normal mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, 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 ø, and the request is held.)

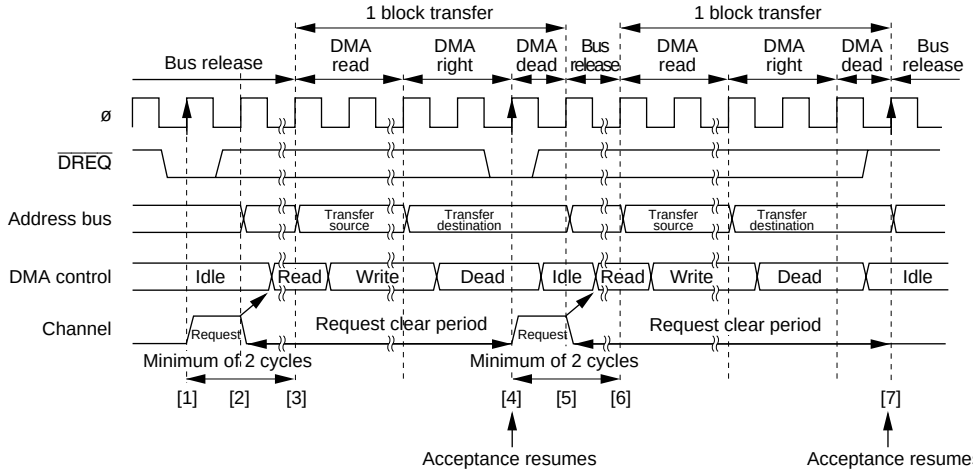
Note: In write data buffer mode, bus breaks from [2] to [7] may be hidden, and not visible.

Figure 7-22 Example of $\overline{\text{DREQ}}$ Level Activated Normal Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ø 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 7-23 shows an example of $\overline{\text{DREQ}}$ level activated block transfer mode transfer.



[1] Acceptance after transfer enabling; the $\overline{\text{DREQ}}$ pin low level is sampled on the rising edge of ø, 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 ø, 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 $\overline{\text{DREQ}}$ Level Activated Block Transfer Mode Transfer

$\overline{\text{DREQ}}$ pin sampling is performed every cycle, with the rising edge of the next ø 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 DMAC Multi-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1, and channel A > channel B. Table 7-13 summarizes the priority order for DMAC channels.

Table 7-13 DMAC Channel Priority Order

Short Address Mode	Full Address Mode	Priority
Channel 0A	Channel 0	High
Channel 0B		
Channel 1A	Channel 1	
Channel 1B		Low

If transfer requests are issued simultaneously for more than one channel, or if a transfer request for another channel is issued during a transfer, when the bus is released the DMAC selects the highest-priority channel from among those issuing a request according to the priority order shown in table 7-13.

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 7-24 shows a transfer example in which transfer requests are issued simultaneously for channels 0A, 0B, and 1.

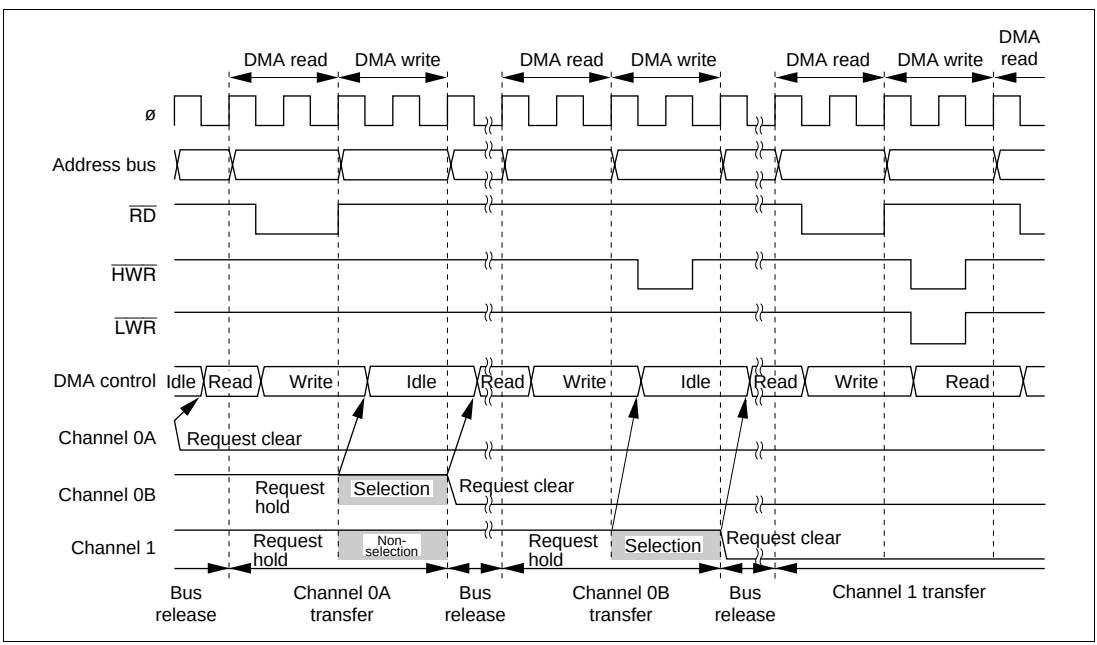


Figure 7-24 Example of Multi-Channel Transfer

7.5.11 Relation between the DMAC, External Bus Requests, and the DTC

There can be no break between a DMA cycle read and a DMA cycle write. This means that an external bus release cycle, or DTC cycle is not generated between the external read and external write in a DMA cycle.

In the case of successive read and write cycles, such as in burst transfer or block transfer, an external bus released state may be inserted after a write cycle. Since the DTC has a lower priority than the DMAC, the DTC does not operate until the DMAC releases the bus.

When DMA cycle reads or writes are accesses to on-chip memory or internal I/O registers, these DMA cycles can be executed at the same time as external bus release. However, simultaneous operation may not be possible when a write buffer is used.

7.5.12 NMI Interrupts and DMAC

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 the 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 7-25 shows the procedure for continuing transfer when it has been interrupted by an NMI interrupt on a channel designated for burst mode transfer.

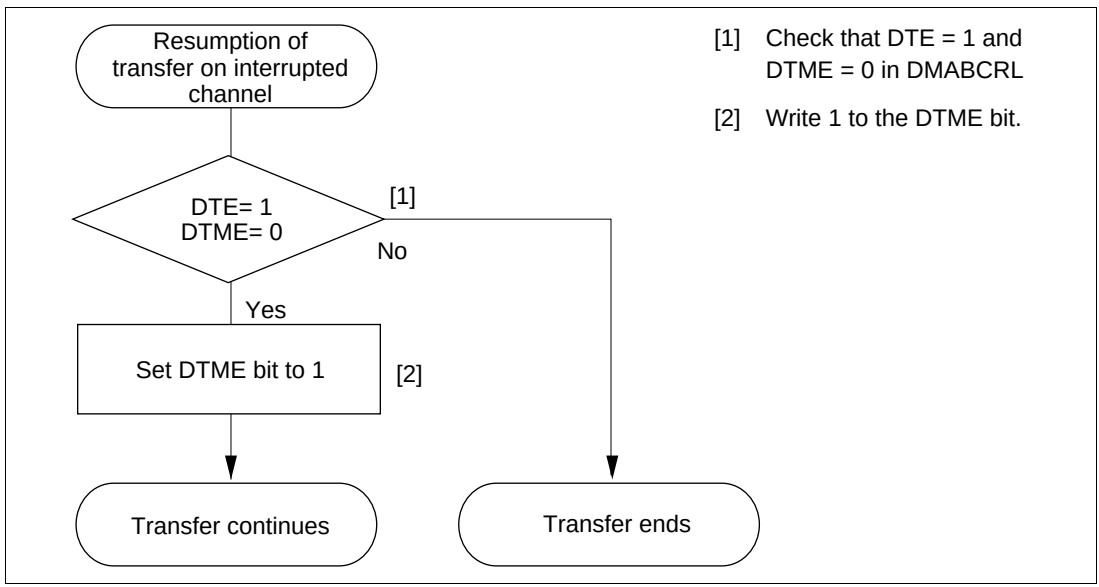


Figure 7-25 Example of Procedure for Continuing Transfer on Channel Interrupted by NMI Interrupt

7.5.13 Forced Termination of DMAC Operation

If the DTE bit for the channel currently operating is cleared to 0, 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.

Figure 7-26 shows the procedure for forcibly terminating DMAC operation by software.

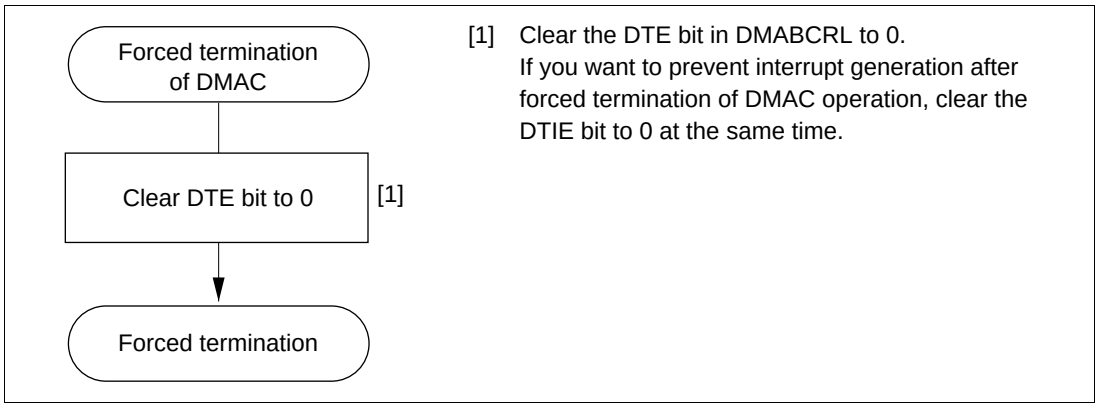


Figure 7-26 Example of Procedure for Forcibly Terminating DMAC Operation

7.5.14 Clearing Full Address Mode

Figure 7-27 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.

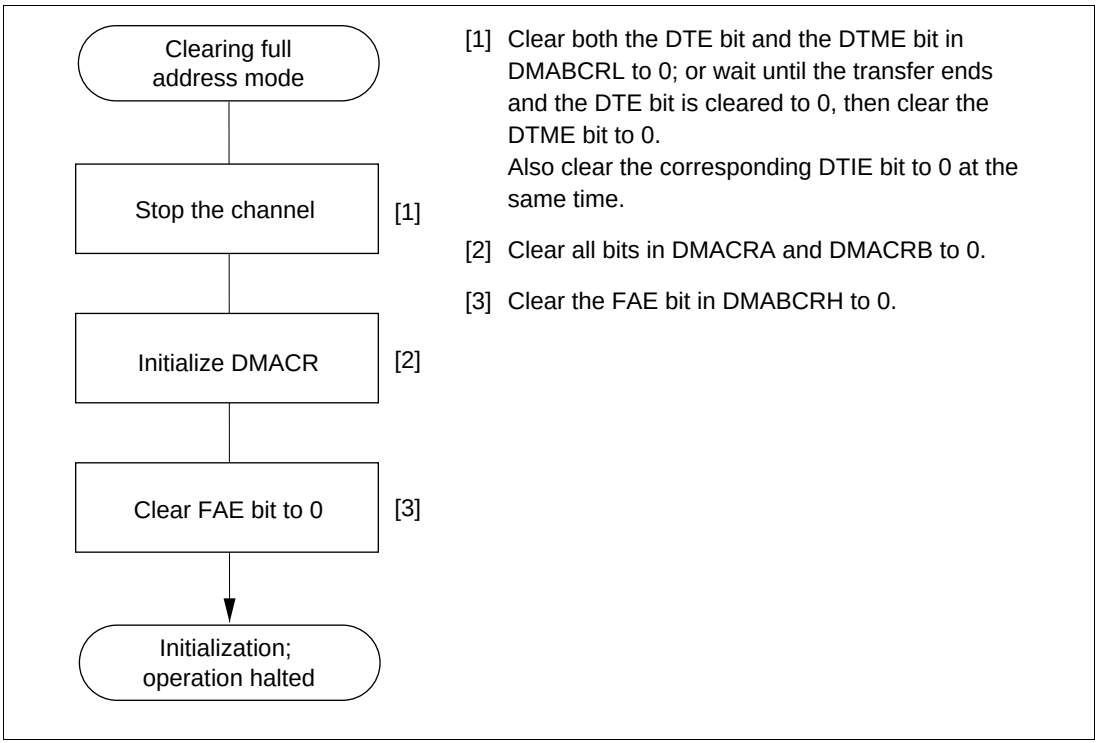


Figure 7-27 Example of Procedure for Clearing Full Address Mode

7.6 Interrupts

The sources of interrupts generated by the DMAC are transfer end and transfer break. Table 7-14 shows the interrupt sources and their priority order.

Table 7-14 Interrupt Source Priority Order

Interrupt Name	Interrupt Source		Interrupt Priority Order
	Short Address Mode	Full Address Mode	
DEND0A	Interrupt due to end of transfer on channel 0A	Interrupt due to end of transfer on channel 0	High
DEND0B	Interrupt due to end of transfer on channel 0B	Interrupt due to break in transfer on channel 0	
DEND1A	Interrupt due to end of transfer on channel 1A	Interrupt due to end of transfer on channel 1	
DEND1B	Interrupt due to end of transfer on channel 1B	Interrupt due to break in transfer on channel 1	Low

Enabling or disabling of each interrupt source is set by means of the DTIE bit for the corresponding channel in DMABCR, and interrupts from each source are sent to the interrupt controller independently.

The relative priority of transfer end interrupts on each channel is decided by the interrupt controller, as shown in table 7-14.

Figure 7-28 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 DTE bit is cleared to 0.

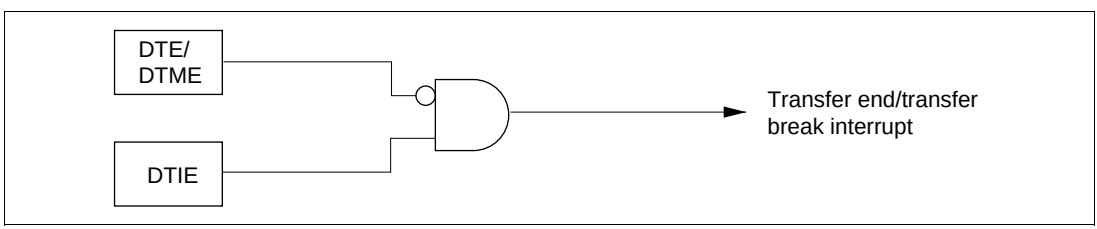


Figure 7-28 Block Diagram of Transfer End/Transfer Break Interrupt

In full address mode, a transfer break interrupt is generated when the DTME bit is cleared to o while DTIEB bit is set to 1.

In both short address mode and full address mode, DMABCR should be set so as to prevent the occurrence of a combination that constitutes a condition for interrupt generation during setting.

7.7 Usage Notes

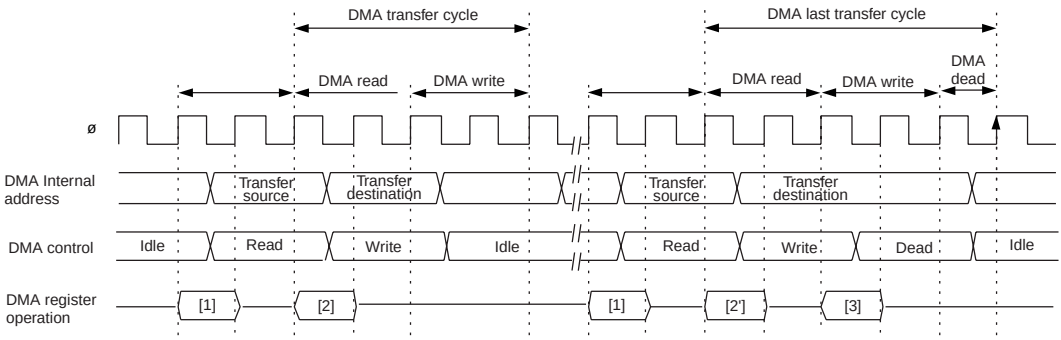
DMAC Register Access during Operation: Except for forced termination, the operating (including transfer waiting state) channel setting should not be changed. The operating channel setting should only be changed when transfer is disabled.

Also, the DMAC register should not be written to in a DMA transfer.

DMAC register reads during operation (including the transfer waiting state) are described below.

(a) DMAC control starts one cycle before the bus cycle, with output of the internal address. Consequently, MAR is updated in the bus cycle before DMAC transfer.

Figure 7-29 shows an example of the update timing for DMAC registers in dual address transfer mode.

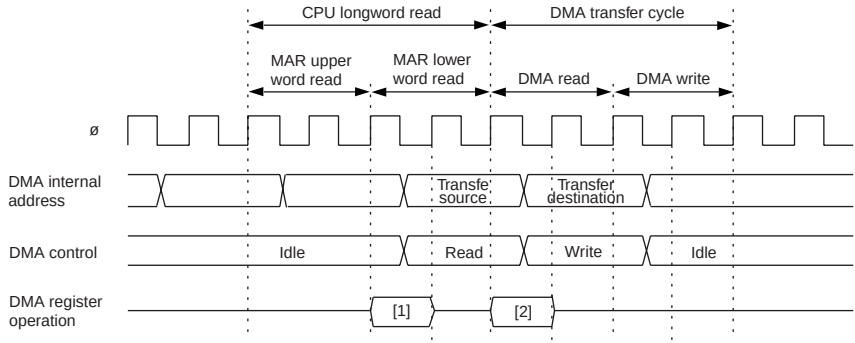


[1] Transfer source address register MAR operation (incremented/decremented/fixed)
Transfer counter ETCR operation (decremented)
Block size counter ETCR operation (decremented in block transfer mode)
[2] Transfer destination address register MAR operation (incremented/decremented/fixed)
[2'] Transfer destination address register MAR operation (incremented/decremented/fixed)
Block transfer counter ETCR operation (decremented, in last transfer cycle of a block in block transfer mode)
[3] Transfer address register MAR restore operation (in block or repeat transfer mode)
Transfer counter ETCR restore (in repeat transfer mode
Block size counter ETCR restore (in block transfer mode)

Notes: 1. In single address transfer mode, the update timing is the same as [1].
2. The MAR operation is post-incrementing/decrementing of the DMA internal address value.

Figure 7-29 DMAC Register Update Timing

(b) If a DMAC transfer cycle occurs immediately after a DMAC register read cycle, the DMAC register is read as shown in figure 7-30.



Note: The lower word of MAR is the updated value after the operation in [1].

Figure 7-30 Contention between DMAC Register Update and CPU Read

Module Stop: When the MSTPA7 bit in MSTPCR is set to 1, the DMAC clock stops, and the module stop state is entered. However, 1 cannot be written to the MSTPA7 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/suspend interrupt (DTE = 0 and DTIE = 1)
- $\overline{\text{TEND}}$ pin enable (TEE = 1)

Medium-Speed Mode: When the DTA bit is 0, internal interrupt signals specified as DMAC transfer sources are edge-detected.

In medium-speed mode, the DMAC operates on a medium-speed clock, while on-chip supporting modules operate on a high-speed clock. Consequently, if the period in which the relevant interrupt source is cleared by the CPU, DTC, or another DMAC channel, and the next interrupt is generated, is less than one state with respect to the DMAC clock (bus master clock), edge detection may not be possible and the interrupt may be ignored.

Also, in medium-speed mode, $\overline{\text{DREQ}}$ pin sampling is performed on the rising edge of the medium-speed clock.

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 by detection of a low level.

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 execution of the DMABCRL write to enable transfer.

When the DMAC is activated, take any necessary steps to prevent an internal interrupt or $\overline{\text{DREQ}}$ pin low level remaining from the end of the previous transfer, etc.

Internal Interrupt after End of Transfer: When the DTE bit is cleared to 0 by the end of transfer or an abort, the selected internal interrupt request will be sent to the CPU or DTC even if DTA is set to 1.

Also, if internal DMAC activation has already been initiated when operation is aborted, the transfer is executed but flag clearing is not performed for the selected internal interrupt even if DTA is set to 1.

An internal interrupt request following the end of transfer or an abort should be handled by the CPU as necessary.

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 a 1 to them.

Section 8 Data Transfer Controller (DTC)

8.1 Overview

The H8S/2214 includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

8.1.1 Features

The features of the DTC are:

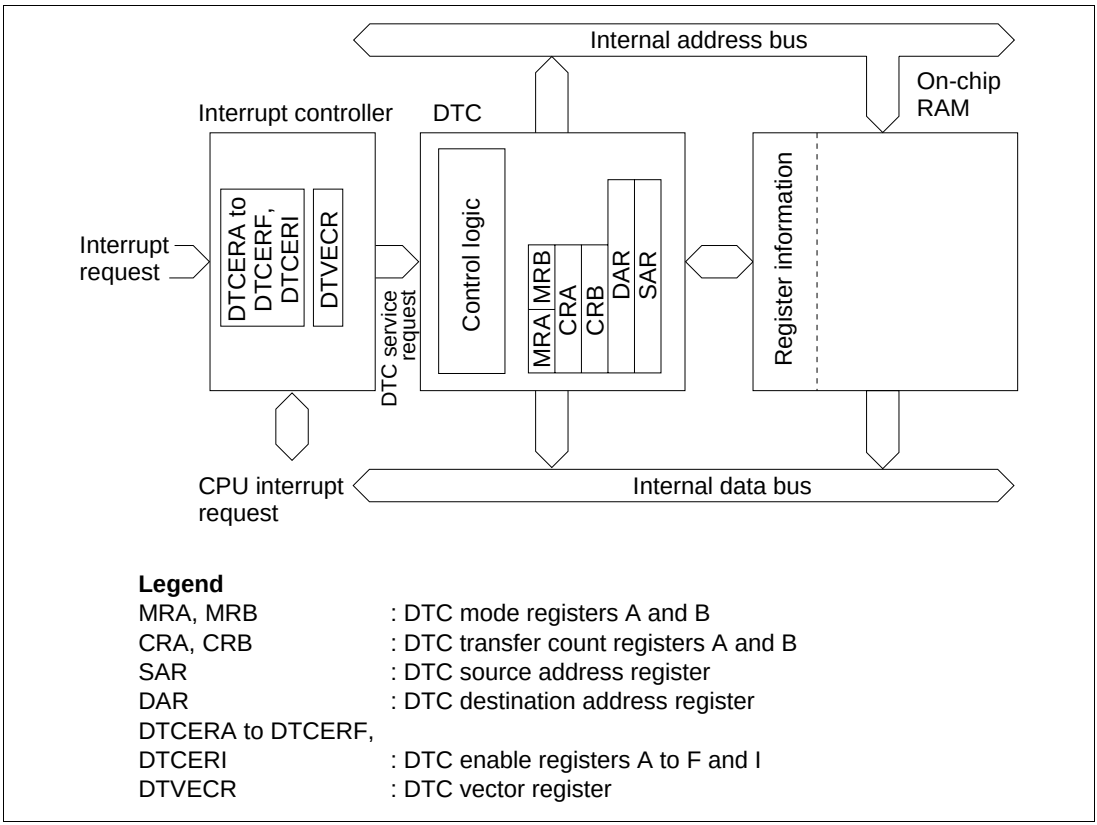
- Transfer possible over any number of channels
  - — Transfer information is stored in memory
  - — One activation source can trigger a number of data transfers (chain transfer)
- Wide range of transfer modes
  - — Normal, repeat, and block transfer modes available
  - — Incrementing, decrementing, and fixing of source and destination addresses can be selected
- Direct specification of 16-Mbyte address space possible
  - — 24-bit transfer source and destination addresses can be specified
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
  - — An interrupt request can be issued to the CPU after one data transfer ends
  - — An interrupt request can be issued to the CPU after the specified data transfers have completely ended
- Activation by software is possible
- Module stop mode can be set
  - — The initial setting enables DTC registers to be accessed. DTC operation is halted by setting module stop mode.

8.1.2 Block Diagram

Figure 8-1 shows a block diagram of the DTC.

The DTC's register information is stored in the on-chip RAM*. 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.

Note: * When the DTC is used, the RAME bit in SYSCR must be set to 1.



Legend

MRA, MRB	: DTC mode registers A and B
CRA, CRB	: DTC transfer count registers A and B
SAR	: DTC source address register
DAR	: DTC destination address register
DTCERA to DTCERF, DTCERI	: DTC enable registers A to F and I
DTVECR	: DTC vector register

Figure 8-1 Block Diagram of DTC

8.1.3 Register Configuration

Table 8-1 summarizes the DTC registers.

Table 8-1 DTC Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
DTC mode register A	MRA	—*[2]	Undefined	—*[3]
DTC mode register B	MRB	—*[2]	Undefined	—*[3]
DTC source address register	SAR	—*[2]	Undefined	—*[3]
DTC destination address register	DAR	—*[2]	Undefined	—*[3]
DTC transfer count register A	CRA	—*[2]	Undefined	—*[3]
DTC transfer count register B	CRB	—*[2]	Undefined	—*[3]
DTC enable registers	DTCER	R/W	H'00	H'FF16 to H'FE1B, H'FE1E
DTC vector register	DTVECR	R/W	H'00	H'FE1F
Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address.

*2 Registers within the DTC cannot be read or written to directly.

*3 Register information is located in on-chip RAM addresses H'EBC0 to H'EFBF. It cannot be located in external memory space. When the DTC is used, do not clear the RAME bit in SYSCR to 0.

8.2 Register Descriptions

8.2.1 DTC Mode Register A (MRA)

Bit	:	7	6	5	4	3	2	1	0
		SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz
Initial value	:	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W	:	—	—	—	—	—	—	—	—

MRA is an 8-bit register that controls the DTC operating mode.

Bits 7 and 6—Source Address Mode 1 and 0 (SM1, SM0): These bits specify whether SAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 7	Bit 6	
SM1	**SM0**	**Description**
0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bits 5 and 4—Destination Address Mode 1 and 0 (DM1, DM0): These bits specify whether DAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 5	Bit 4	
DM1	**DM0**	**Description**
0	—	DAR is fixed
1	0	DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bits 3 and 2—DTC Mode (MD1, MD0): These bits specify the DTC transfer mode.

Bit 3	Bit 2	
MD1	**MD0**	**Description**
0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

Bit 1—DTC Transfer Mode Select (DTS): 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.

Bit 1	
DTS	**Description**
0	Destination side is repeat area or block area
1	Source side is repeat area or block area

Bit 0—DTC Data Transfer Size (Sz): Specifies the size of data to be transferred.

Bit 0	
Sz	**Description**
0	Byte-size transfer
1	Word-size transfer

8.2.2 DTC Mode Register B (MRB)

Bit	:	7	6	5	4	3	2	1	0
		CHNE	DISEL	—	—	—	—	—	—
Initial value:		Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W	:	—	—	—	—	—	—	—	—

MRB is an 8-bit register that controls the DTC operating mode.

Bit 7—DTC Chain Transfer Enable (CHNE): Specifies chain transfer. With chain transfer, a number of data transfers can be performed consecutively in response to a single transfer request.

In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the interrupt source flag, and clearing of DTCER is not performed.

Bit 7

CHNE	Description
0	End of DTC data transfer (activation waiting state is entered)
1	DTC chain transfer (new register information is read, then data is transferred)

Bit 6—DTC Interrupt Select (DISEL): Specifies whether interrupt requests to the CPU are disabled or enabled after a data transfer.

Bit 6

DISEL	Description
0	After a data transfer ends, the CPU interrupt is disabled unless the transfer counter is 0 (the DTC clears the interrupt source flag of the activating interrupt to 0)
1	After a data transfer ends, the CPU interrupt is enabled (the DTC does not clear the interrupt source flag of the activating interrupt to 0)

Bits 5 to 0—Reserved: These bits have no effect on DTC operation in the H8S/2214, and should always be written with 0.

8.2.3 DTC Source Address Register (SAR)

Bit	:	23	22	21	20	19	---	4	3	2	1	0

Initial value:		Undefined	Undefined	Undefined	Undefined	Undefined	---	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	:	—	—	—	—	—	---	—	—	—	—	—

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)

Bit	:	23	22	21	20	19	---	4	3	2	1	0

Initial value	:	Undefined	Undefined	Undefined	Undefined	Undefined	---	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	:	—	—	—	—	—	---	—	—	—	—	—

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.

8.2.5 DTC Transfer Count Register A (CRA)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:		Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		← CRAH →								← CRAL →							

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 65536). 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. This operation is repeated.

8.2.6 DTC Transfer Count Register B (CRB)

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

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 65536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

8.2.7 DTC Enable Registers (DTCER)

Bit	:	7	6	5	4	3	2	1	0
		DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The DTC enable registers comprise seven 8-bit readable/writable registers, DTCERA to DTCERF and DTCERI, with bits corresponding to the interrupt sources that can control enabling and disabling of DTC activation. These bits enable or disable DTC service for the corresponding interrupt sources.

The DTC enable registers are initialized to H'00 by a reset and in hardware standby mode.

Bit n—DTC Activation Enable (DTCEn)

Bit n DTCEn	Description	
0	DTC activation by this interrupt is disabled [Clearing conditions] • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended	(Initial value)
1	DTC activation by this interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified number of transfers have not ended	

(n = 7 to 0)

A DTCE bit can be set for each interrupt source that can activate the DTC. The correspondence between interrupt sources and DTCE bits is shown in table 8-4, together with the vector number generated for each interrupt controller.

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 by writing data after executing a dummy read on the relevant register.

8.2.8 DTC Vector Register (DTVECR)

Bit	:	7	6	5	4	3	2	1	0
		SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/(W)*1	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Notes: *1 Only 1 can be written to the SWDTE bit.
*2 Bits DTVEC6 to DTVEC0 can be written to when SWDTE = 0.

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

DTVECR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—DTC Software Activation Enable (SWDTE): Enables or disables DTC activation by software.

Bit 7

SWDTE	Description
0	DTC software activation is disabled (Initial value) [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 is written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU
1	DTC software activation is enabled [Holding conditions] • When the DISEL bit is 1 and data transfer has ended • When the specified number of transfers have ended • During data transfer due to software activation

Bits 6 to 0—DTC Software Activation Vectors 6 to 0 (DTVEC6 to DTVEC0): These bits specify a vector number for DTC software activation.

The vector address is expressed as H'0400 + ((vector number) << 1). <<1 indicates a one-bit left-shift. For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420.

8.2.9 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRA is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPA6 bit in MSTPCRA is set to 1, the DTC operation stops at the end of the bus cycle and a transition is made to module stop mode. However, 1 cannot be written in the MSTPA6 bit while the DTC is operating. For details, see section 17.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 6—Module Stop (MSTPA6): Specifies the DTC module stop mode.

Bit 6

MSTPA6	Description	
0	DTC module stop mode cleared	(Initial value)
1	DTC module stop mode set	

8.3 Operation

8.3.1 Overview

When activated, the DTC reads register information that is already stored in memory and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to memory. Pre-storage of register information in memory makes it possible to transfer data over any required number of channels. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation.

Figure 8-2 shows a flowchart of DTC operation.

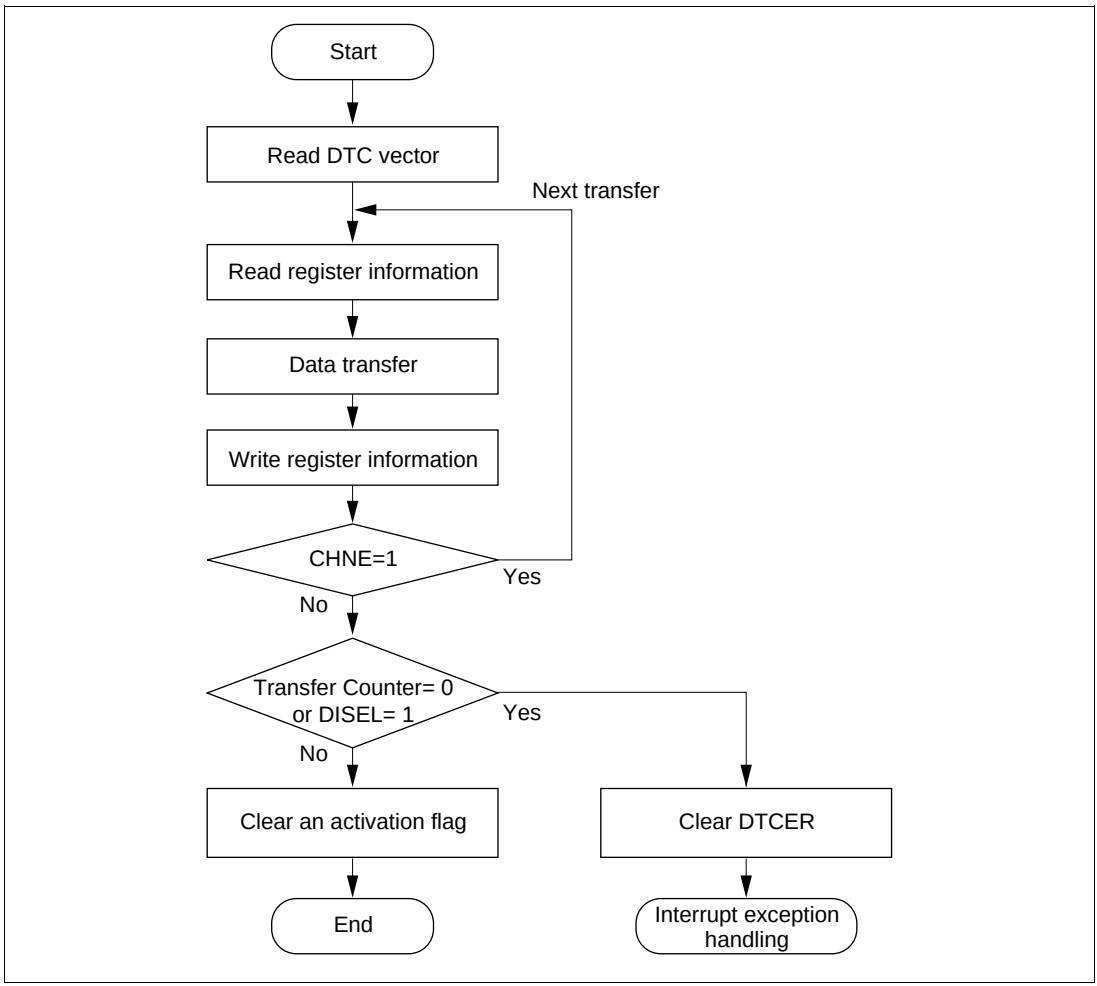


Figure 8-2 Flowchart of DTC Operation

The DTC transfer mode can be normal mode, repeat mode, or block transfer mode.

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.

Table 8-2 outlines the functions of the DTC.

Table 8-2 DTC Functions

		Address Registers	
Transfer Mode	**Activation Source**	**Transfer Source**	**Transfer Destination**
• Normal mode — One transfer request transfers one byte or one word — Memory addresses are incremented or decremented by 1 or 2 — Up to 65,536 transfers possible • Repeat mode — One transfer request transfers one byte or one word — Memory addresses are incremented or decremented by 1 or 2 — After the specified number of transfers (1 to 256), the initial state resumes and operation continues • Block transfer mode — One transfer request transfers a block of the specified size — Block size is from 1 to 256 bytes or words — Up to 65,536 transfers possible — A block area can be designated at either the source or destination	• IRQ • TPU TGI • 8-bit timer CMI • SCI TXI or RXI • A/D converter ADI • Software	24 bits	24 bits

8.3.2 Activation Sources

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. An interrupt becomes a DTC activation source when the corresponding bit is set to 1, and a CPU interrupt source when the bit is cleared to 0.

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. Table 8-3 shows activation source and DTCER clearance. The activation source flag, in the case of RXI0, for example, is the RDRF flag of SCI0.

Table 8-3 Activation Source and DTCER Clearance

Activation Source	When the DISEL Bit Is 0 and the Specified Number of Transfers Have Not Ended	When the DISEL Bit Is 1, or when the Specified Number of Transfers Have Ended
Software activation	The SWDTE bit is cleared to 0	The SWDTE bit remains set to 1 An interrupt is issued to the CPU
Interrupt activation	The corresponding DTCER bit remains set to 1 The activation source flag is cleared to 0	The corresponding DTCER bit is cleared to 0 The activation source flag remains set to 1 A request is issued to the CPU for the activation source interrupt

Figure 8-3 shows a block diagram of activation source control. For details see section 5, Interrupt Controller.

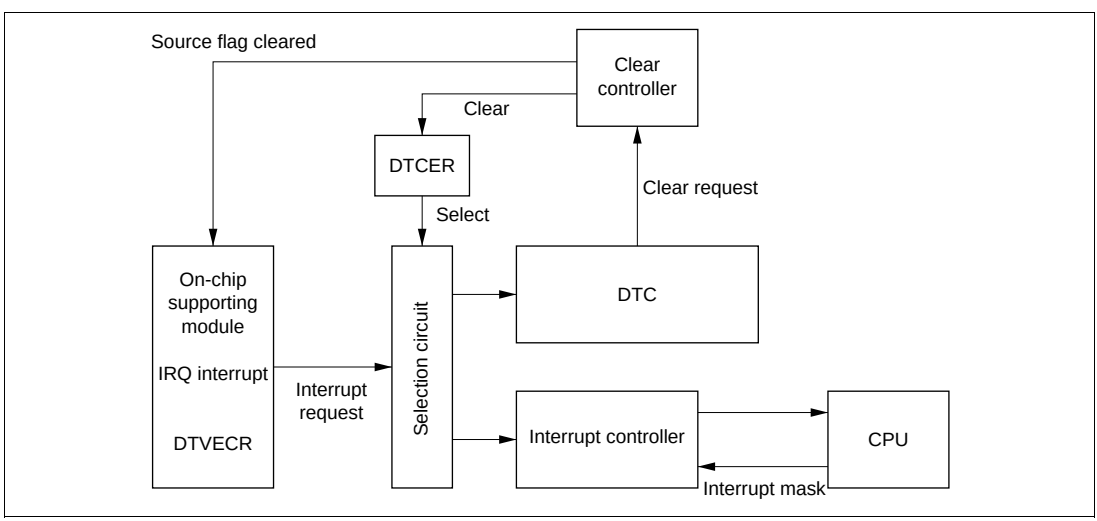


Figure 8-3 Block Diagram of DTC Activation Source Control

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.

8.3.3 DTC Vector Table

Figure 8-4 shows the correspondence between DTC vector addresses and register information.

Table 8-4 shows the correspondence between activation and vector addresses. When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] << 1) (where << 1 indicates a 1-bit left shift). For example, if DTVECR is H'10, the vector address is H'0420.

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. The register information can be placed at predetermined addresses in the on-chip RAM. The start address of the register information should be an integral multiple of four.

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 address in the on-chip RAM.

Note: * Not available in the H8S/2214.

Table 8-4 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE*	Priority
Write to DTVECR	Software	DTVECR	H'0400+ (DTVECR [6:0] <<1)	—	High
IRQ0	External pin	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	
TGI0A (GR0A compare match/ input capture)	TPU channel 0	32	H'0440	DTCEB5	
TGI0B (GR0B compare match/ input capture)		33	H'0442	DTCEB4	
TGI0C (GR0C compare match/ input capture)		34	H'0444	DTCEB3	
TGI0D (GR0D compare match/ input capture)		35	H'0446	DTCEB2	
TGI1A (GR1A compare match/ input capture)	TPU channel 1	40	H'0450	DTCEB1	
TGI1B (GR1B compare match/ input capture)		41	H'0452	DTCEB0	
TGI2A (GR2A compare match/ input capture)	TPU channel 2	44	H'0458	DTCEC7	
TGI2B (GR2B compare match/ input capture)		45	H'045A	DTCEC6	Low

Note: * DTCE bits with no corresponding interrupt are reserved, and should be written with 0.

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE	Priority
DEND0A (channel 0/channel 0A transfer end)	DMAC	72	H'0490	DTCEE7	High
DEND0B (channel 0B transfer end)		73	H'0492	DTCEE6	
DEND1A (channel 1/channel 1A transfer end)		74	H'0494	DTCEE5	
DEND1B (channel 1B transfer end)		75	H'0496	DTCEE4	
RXI0 (reception complete 0)	SCI channel 0	81	H'04A2	DTCEE3	
TXI0 (transmit data empty 0)		82	H'04A4	DTCEE2	
RXI1 (reception complete 1)	SCI channel 1	85	H'04AA	DTCEE1	
TXI1 (transmit data empty 1)		86	H'04AC	DTCEE0	
RXI2 (reception complete 2)	SCI channel 2	89	H'04B2	DTCEF7	
TXI2 (transmit data empty 2)		90	H'04B4	DTCEF6	
EXIRQ0	External module	104	H'04D0	DTCEG7	
EXIRQ1		105	H'04D2	DTCEG6	
EXIRQ2		106	H'04D4	DTCEG5	
EXIRQ3		107	H'04D6	DTCEG4	
EXIRQ4		108	H'04D8	DTCEG3	
EXIRQ5		109	H'04DA	DTCEG2	
EXIRQ6		110	H'04DC	DTCEG1	
EXIRQ7		111	H'04DE	DTCEG0	Low

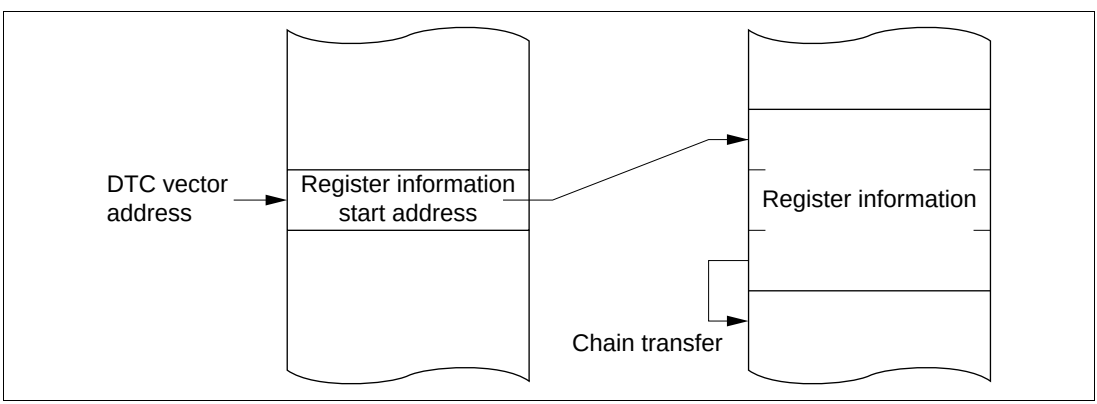


Figure 8-4 Correspondence between DTC Vector Address and Register Information

8.3.4 Location of Register Information in Address Space

Figure 8-5 shows how the register information should be located in the address space.

Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information (contents of the vector address). In the case of chain transfer, register information should be located in consecutive areas.

Locate the register information in the on-chip RAM (addresses: H'FFEBC0 to H'FFEFBF).

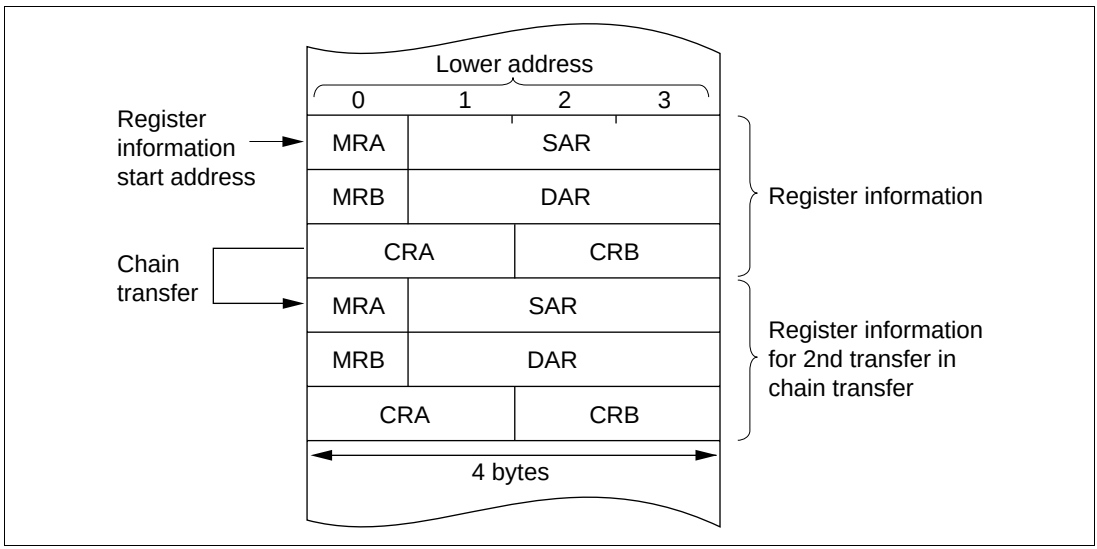


Figure 8-5 Location of Register Information in Address Space

8.3.5 Normal Mode

In normal mode, one operation transfers one byte or one word of data.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt can be requested.

Table 8-5 lists the register information in normal mode and figure 8-6 shows memory mapping in normal mode.

Table 8-5 Register Information 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

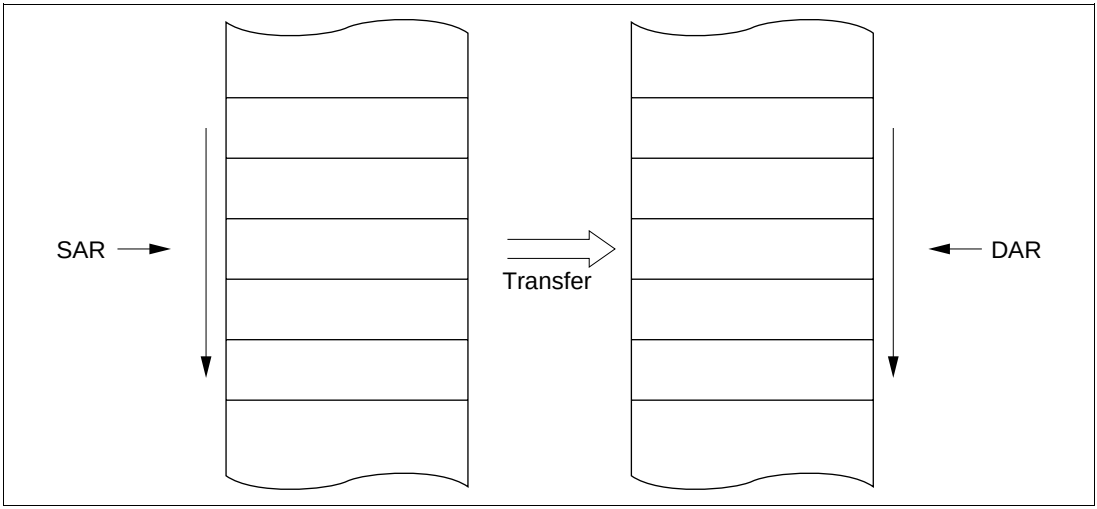


Figure 8-6 Memory Mapping in Normal Mode

8.3.6 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data.

From 1 to 256 transfers can be specified. Once the specified number of transfers have 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 8-6 lists the register information in repeat mode and figure 8-7 shows memory mapping in repeat mode.

Table 8-6 Register Information in Repeat 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 number of transfers
DTC transfer count register AL	CRAL	Designates transfer count
DTC transfer count register B	CRB	Not used

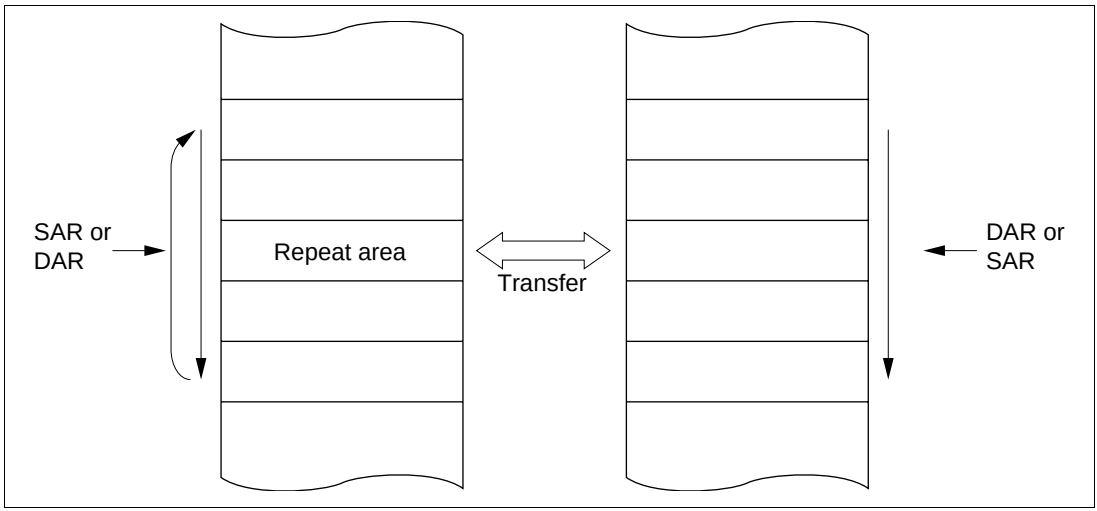


Figure 8-7 Memory Mapping in Repeat Mode

8.3.7 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.

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 have ended, a CPU interrupt is requested.

Table 8-7 lists the register information in block transfer mode and figure 8-8 shows memory mapping in block transfer mode.

Table 8-7 Register Information 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	Transfer count

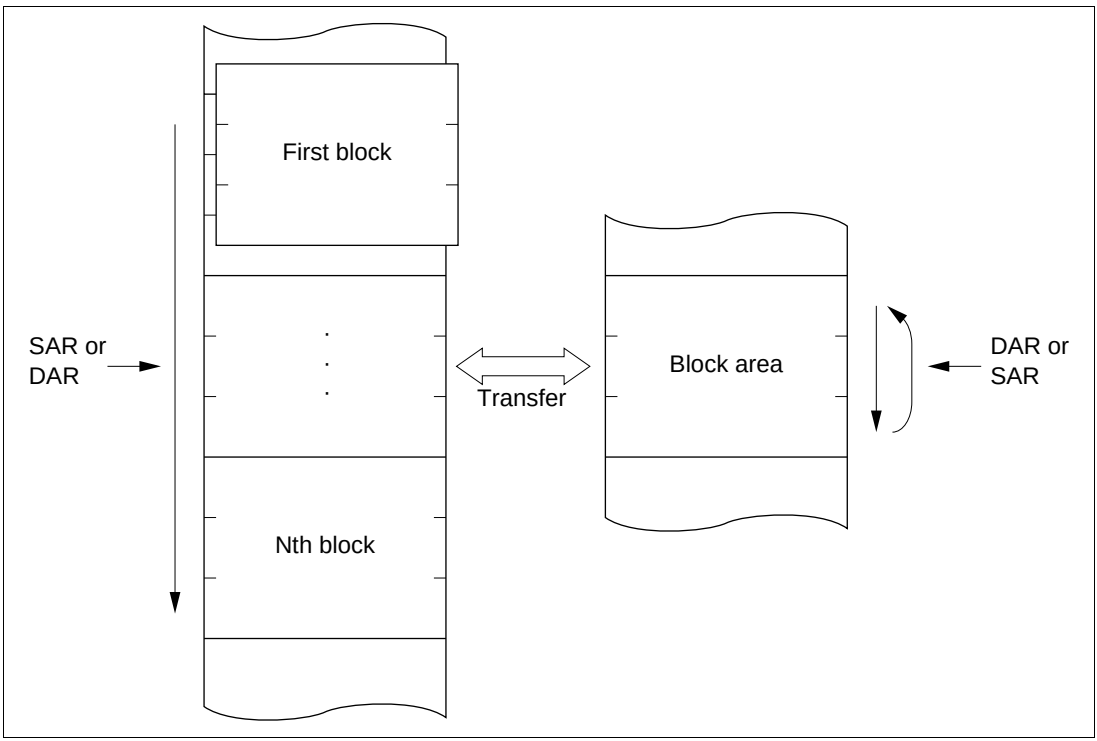


Figure 8-8 Memory Mapping in Block Transfer Mode

8.3.8 Chain Transfer

Setting the CHNE bit to 1 enables a number of data transfers to be performed consectutively 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 memory map for chain transfer.

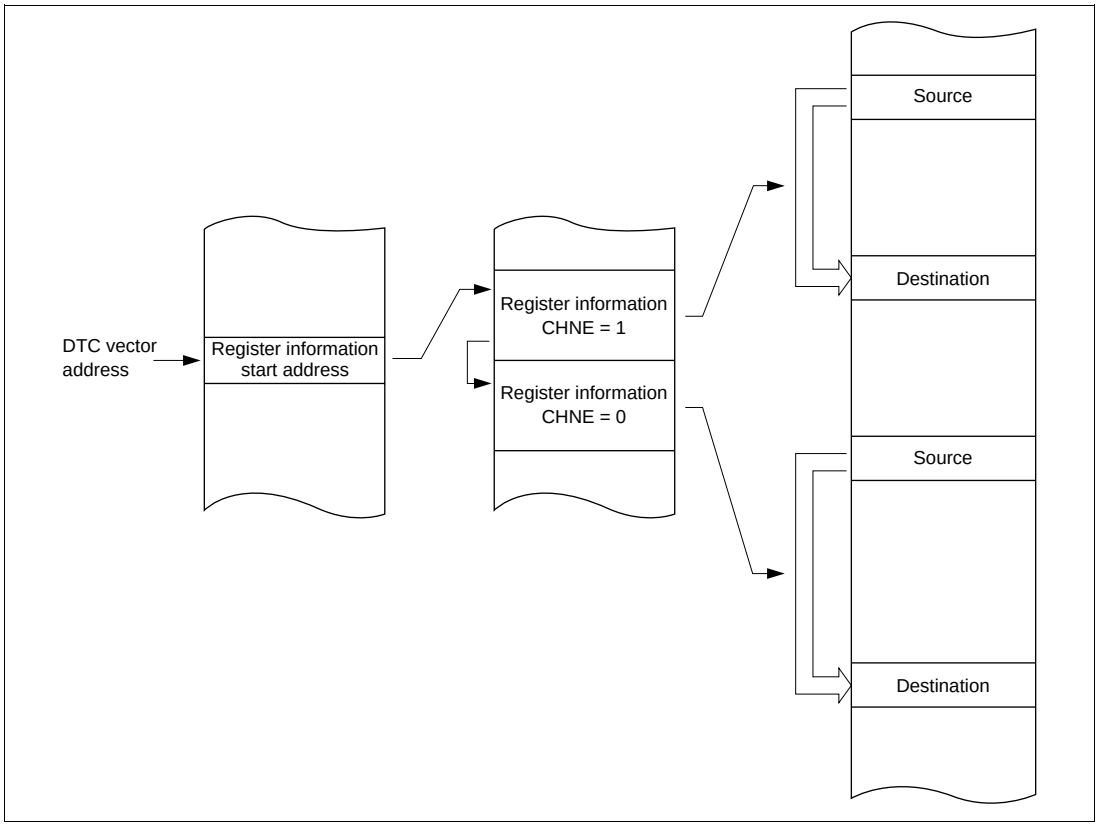


Figure 8-9 Chain Transfer Memory Map

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.

8.3.9 Operation Timing

Figures 8-10 to 8-12 show an example of DTC operation timing.

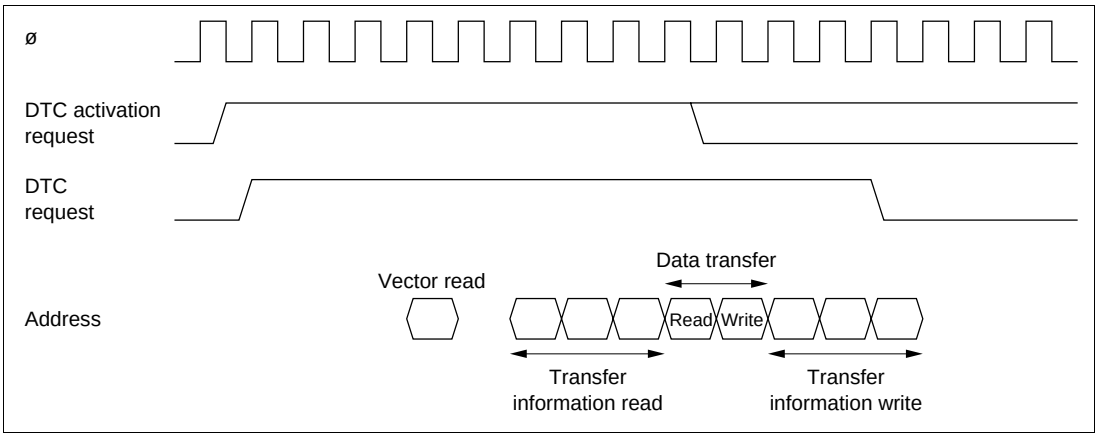


Figure 8-10 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

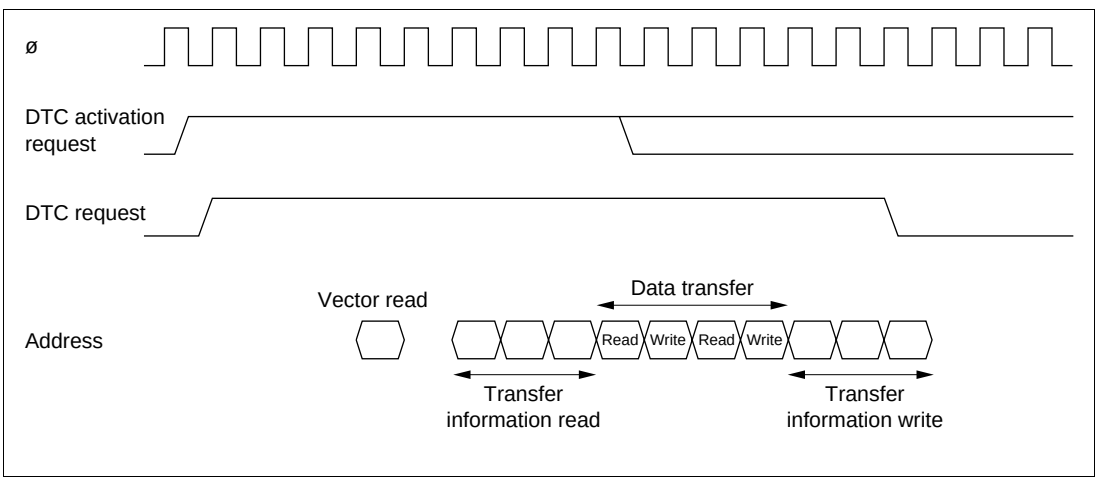


Figure 8-11 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

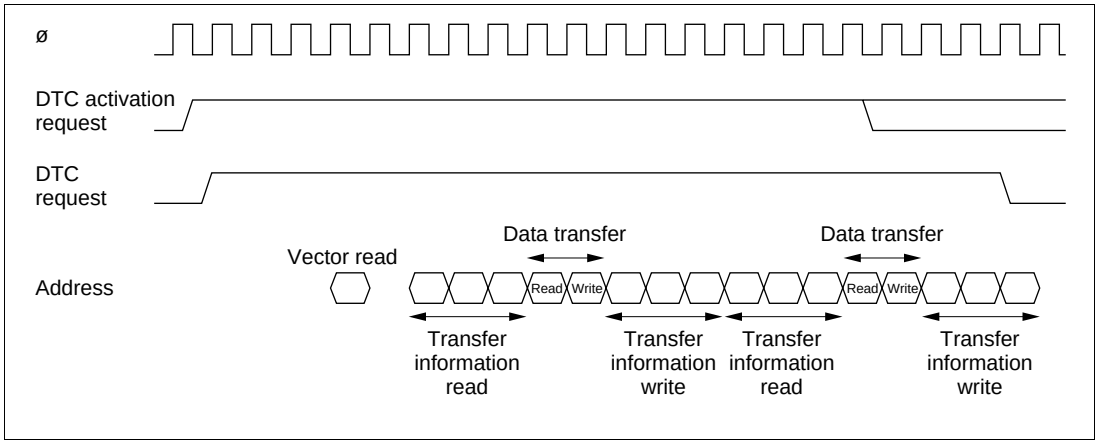


Figure 8-12 DTC Operation Timing (Example of Chain Transfer)

8.3.10 Number of DTC Execution States

Table 8-8 lists execution statuses for a single DTC data transfer, and table 8-9 shows the number of states required for each execution status.

Table 8-8 DTC Execution Statuses

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operations M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

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

Table 8-9 Number of States Required for Each Execution Status

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 + 2 m	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 + 2 m	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 + 2 m	2	3 + m
	Internal operation	S_M	1							

m: Number of wait states in external device access

The number of execution states is calculated from the formula below. Note that Σ means the sum of all transfers activated by one activation event (the number in which the CHNE bit is set to 1, plus 1).

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.

8.3.11 Procedures for Using DTC

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 DTCER 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.

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.

8.3.12 Examples of Use of the DTC

(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.

(2) Software Activation

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.

8.4 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 have 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.

8.5 Usage Notes

Module Stop: When the MSTPA6 bit in MSTPCRA is set to 1, the DTC clock stops, and the DTC enters the module stop state. However, 1 cannot be written in the MSTPA6 bit while the DTC is operating.

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.

DMAC Transfer End Interrupt: When DTC transfer is activated by a DMAC transfer end interrupt, the DMAC's DTE bit is not subject to DTC control, regardless of the transfer counter and DISEL bit, and the write data has priority. Consequently, an interrupt request is not sent to the CPU when the DTC transfer counter reaches 0.

DTCE Bit Setting: For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are masked, multiple activation sources can be set at one time by writing data after executing a dummy read on the relevant register.

Section 9 I/O Ports

9.1 Overview

The H8S/2214 has ten I/O ports (ports 1, 3, 7, and A to G), and two input-only ports (ports 4 and 9).

Table 9-1 summarizes the port functions. The pins of each port also have other functions.

Each port includes a data direction register (DDR) that controls input/output (not provided for the input-only ports), a data register (DR) that stores output data, and a port register (PORT) used to read the pin states.

Ports A to E have a built-in MOS input pull-up function, and in addition to DR and DDR, have a MOS input pull-up control register (PCR) to control the on/off status of the MOS input pull-ups.

Ports 3 and A include an open-drain control register (ODR) that controls the on/off status of the output buffer PMOS.

All the ports can drive a single TTL load and 30 pF capacitive load.

The $\overline{\text{IRQ}}$ pins and external expansion interrupt input pins are Schmitt-triggered inputs.

Block diagrams of each port are shown in Appendix C, I/O Port Block Diagrams.

Table 9-1 H8S/2214 Port Functions

Port	Description	Pins	Modes 4 and 5	Mode 6	Mode 7
Port 1	• 8-bit I/O port • Schmitt-triggered input (IRQ1, IRQ0)	P17/TIOCB2/TCLKD P16/TIOCA2/IRQ1 P15/TIOCB1/TCLKC P14/TIOCA1/IRQ0	8-bit I/O port also functioning as TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, TIOCB2), and external interrupt input (IRQ0, IRQ1)		
		P13/TIOCD0/TCLKB/A23 P12/TIOCC0/TCLKA/A22 P11/TIOCB0/A21 P10/TIOCA0/A20	8-bit I/O port also functioning as DMAC output pins (DACK0, DACK1), TPU I/O pins (TCLKA, TCLKB, TIOCA0, TIOCB0, TIOCC0, TIOCD0), and address output (A20 to A23)		
Port 3	• 7-bit I/O port • Open-drain output capability • Schmitt-triggered input (IRQ5, IRQ4, EXIRQ7)	P36/EXIRQ7 P35/SCK1/IRQ5 P34/RxD1 P33/TxD1 P32/SCK0/IRQ4 P31/RxD0 P30/TxD0	7-bit I/O port also functioning as SCI (channel 0 and 1) I/O pins (TxD0, RxD0, SCK0, TxD1, RxD1, SCK1) and interrupt input (IRQ4, IRQ5), and external extended interrupt input (EXIRQ7)		
Port 4	• 8-bit input port • Schmitt-triggered input (EXIRQ6, to EXIRQ0)	P47/EXIRQ6 P46/EXIRQ5 P45 P44/EXIRQ4 P43/EXIRQ3 P42/EXIRQ2 P41/EXIRQ1 P40/EXIRQ0	8-bit input port also functioning as external extended interrupt input pins (EXIRQ6 to EXIRQ0)		

Port	Description	Pins	Modes 4 and 5	Mode 6	Mode 7
Port 7	• 8-bit I/O port	P77 P76/EXMSTP P75/$\overline{\text{EXMS}}$ P74/$\overline{\text{MRES}}$/$\overline{\text{EXDTCE}}$ P73/$\overline{\text{TEND1}}$/$\overline{\text{CS7}}$ P72/$\overline{\text{TEND0}}$/$\overline{\text{CS6}}$ P71/$\overline{\text{DREQ1}}$/$\overline{\text{CS5}}$ P70/$\overline{\text{DREQ0}}$/$\overline{\text{CS4}}$	8-bit I/O port also functioning as DMAC I/O pins ($\overline{\text{DREQ0}}$, $\overline{\text{TEND0}}$, $\overline{\text{DREQ1}}$, $\overline{\text{TEND1}}$), bus control output pins ($\overline{\text{CS4}}$ to $\overline{\text{CS7}}$), the manual reset input pin ($\overline{\text{MRES}}$), and external module output pins (EXMSTP, $\overline{\text{EXMS}}$, $\overline{\text{EXDTCE}}$)		8-bit I/O port also functioning as DMAC I/O pins ($\overline{\text{DREQ0}}$, $\overline{\text{TEND0}}$, $\overline{\text{DREQ1}}$, $\overline{\text{TEND1}}$), the manual reset input pins ($\overline{\text{MRES}}$), and external module output pin (EXMSTP, $\overline{\text{EXMS}}$, $\overline{\text{EXDTCE}}$)
Port 9	• 1-bit input port	P96/DA0	1-bit input port also functioning as D/A analog output pin (D/A0)		
Port A	• 4-bit I/O port • Built-in MOS input pull-up • Open-drain output capability	PA3/A19/SCK2 PA2/A18/RxD2 PA1/A17/TxD2 PA0/A16	I/O port also functioning as SCI (channel 2) I/O pins (TxD2, RxD2, SCK2) and address output (A16 to A19)		I/O port also functioning as SCI (channel 2) I/O pins (TxD2, RxD2, SCK2)
Port B	• 8-bit I/O port • Built-in MOS input pull-up	PB7/A15 PB6/A14 PB5/A13 PB4/A12 PB3/A11 PB2/A10 PB1/A9 PB0/A8	I/O port also functioning as address output (A8 to A15)		I/O port
Port C	• 8-bit I/O port • Built-in MOS input pull-up	PC7/A7–PC0/A0	Address output (A0 to A7)	When DDR = 0: Input port When DDR = 1: Address output	I/O port
Port D	• 8-bit I/O port • Built-in MOS input pull-up	PD7/D15–PD0/D8	Data bus input/output		I/O port
Port E	• 8-bit I/O port • Built-in MOS input pull-up	PE7/D7–PE0/D0	8-bit bus mode: I/O port 16-bit bus mode: Data bus input/output		I/O port

Port	Description	Pins	Modes 4 and 5	Mode 6	Mode 7
Port F	• 8-bit I/O port • Schmitt-triggered input ($\overline{IRQ3}$, $\overline{IRQ2}$)	PF7/ø	When DDR = 0: Input port When DDR = 1 (after reset): ø output		When DDR = 0 (after reset): Input port When DDR = 1: ø output
		PF6/$\overline{AS}$ PF5/$\overline{RD}$ PF4/$\overline{HWR}$	$\overline{AS}$, $\overline{RD}$, $\overline{HWR}$ output		I/O port
		PF3/$\overline{LWR}$/$\overline{IRQ3}$	16-bit bus mode: $\overline{LWR}$ output 8-bit bus mode: I/O port also functioning as interrupt input pin ($\overline{IRQ3}$)		I/O port also functioning as interrupt input pin ($\overline{IRQ3}$)
		PF2/$\overline{WAIT}$	When WAITE = 0 (after reset): I/O port When WAITE = 1: $\overline{WAIT}$ input		I/O port
		PF1/$\overline{BACK}$	When BRLE = 0 (after reset): I/O port When BRLE = 1: $\overline{BACK}$ output		I/O port
		PF0/$\overline{BREQ}$/$\overline{IRQ2}$	When BRLE = 0 (after reset): I/O port also functioning as interrupt input pin ($\overline{IRQ2}$) When BRLE = 1: $\overline{BREQ}$ input also functioning as interrupt input pin ($\overline{IRQ2}$)		I/O port also functioning as interrupt input pin ($\overline{IRQ2}$)
Port G	• 5-bit I/O port • Schmitt-triggered input ($\overline{IRQ7}$, $\overline{IRQ6}$)	PG4/$\overline{CS0}$	When DDR = 0*1: Input port When DDR = 1*2: $\overline{CS0}$ output		I/O port also functioning as interrupt input pins ($\overline{IRQ6}$, $\overline{IRQ7}$)
		PG3/$\overline{CS1}$ PG2/$\overline{CS2}$ PG1/$\overline{CS3}$/$\overline{IRQ7}$	When DDR = 0 (after reset): Input port also functioning as interrupt input pin ($\overline{IRQ7}$) When DDR = 1: Interrupt input pin ($\overline{IRQ7}$) also functions as $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$ output		
		PG0/$\overline{IRQ6}$			

Notes: *1 After a mode 6 reset

*2 After a mode 4 or 5 reset

9.2 Port 1

9.2.1 Overview

Port 1 is an 8-bit I/O port. Port 1 pins also function as TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, and TIOCB2), external interrupt pins ($\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$), and address bus output pins (A23 to A20). Port 1 pin functions depend on the operating mode.

The interrupt input pins ($\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$) are Schmitt-triggered inputs.

Figure 9-1 shows the port 1 pin configuration.

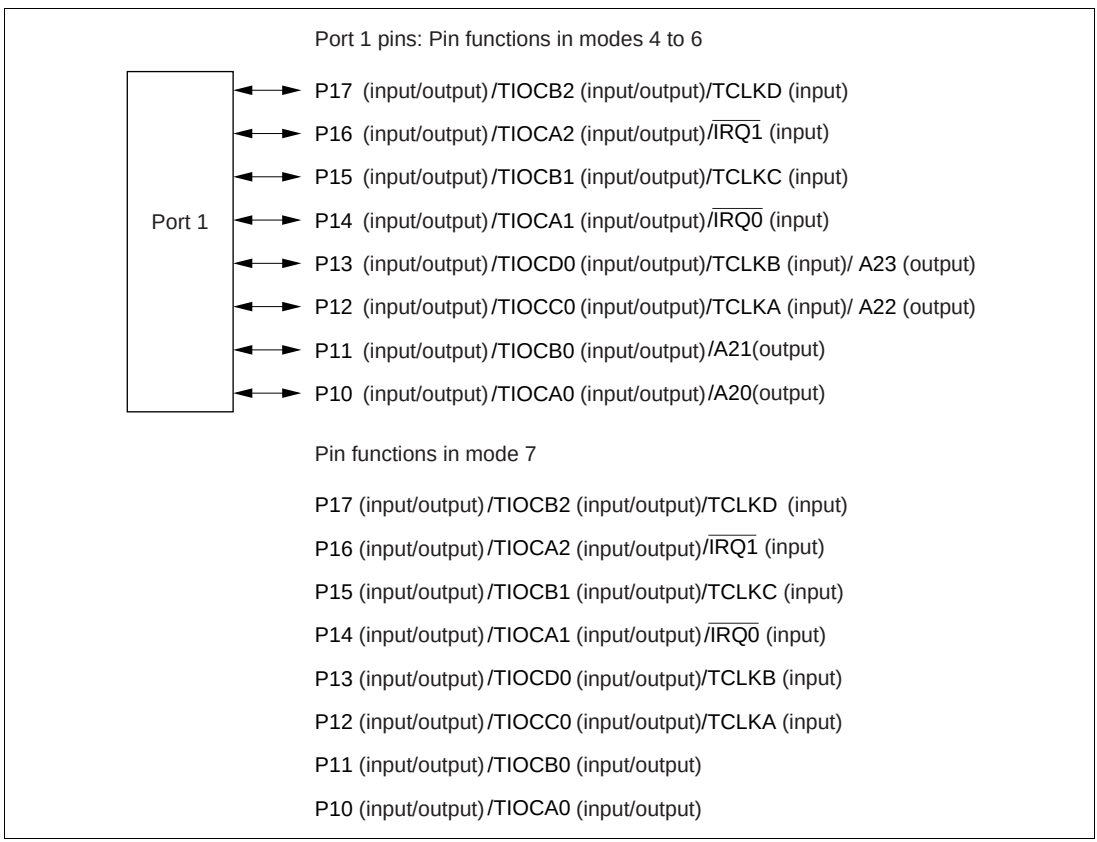


Figure 9-1 Port 1 Pin Functions

9.2.2 Register Configuration

Table 9-2 shows the port 1 register configuration.

Table 9-2 Port 1 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 1 data direction register	P1DDR	W	H'00	H'FE30
Port 1 data register	P1DR	R/W	H'00	H'FF00
Port 1 register	PORT1	R	Undefined	H'FFB0

Note: * Lower 16 bits of the address.

(1) Port 1 Data Direction Register (P1DDR)

Bit	:	7	6	5	4	3	2	1	0
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 1. P1DDR cannot be read; if it is, an undefined value will be read.

P1DDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode. As the TPU is initialized by a manual reset, the pin states in this case are determined by the P1DDR and P1DR specifications.

The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

(a) Modes 4, 5, and 6

If address output is enabled by the setting of bits AE3 to AE0 in PFCR, pins P13 to P10 are address outputs. Pins P17 to P14, and pins P13 to P10 when address output is disabled, are output ports when the corresponding P1DDR bits are set to 1, and input ports when the corresponding P1DDR bits are cleared to 0.

(b) Mode 7

Setting a P1DDR bit to 1 makes the corresponding port 1 pin an output port, while clearing the bit to 0 makes the pin an input port.

(2) Port 1 Data Register (P1DR)

Bit	:	7	6	5	4	3	2	1	0
		P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1DR is an 8-bit readable/writable register that stores output data for the port 1 pins (P17 to P10).

P1DR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(3) Port 1 Register (PORT1)

Bit	:	7	6	5	4	3	2	1	0
		P17	P16	P15	P14	P13	P12	P11	P10
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins P17 to P10.

PORT1 is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 1 pins (P17 to P10) must always be performed on P1DR.

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.

After a power-on reset and in hardware standby mode, PORT1 contents are determined by the pin states, as P1DDR and P1DR are initialized. PORT1 retains its previous state after a manual reset and in software standby mode.

9.2.3 Pin Functions

Port 1 pins also function as TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, and TIOCB2), external interrupt input pins ($\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$), and address output pins (A23 to A20). Port 1 pin functions are shown in table 9-3.

Table 9-3 Port 1 Pin Functions

Pin | **Pin Functions and Selection Method**

P17/ TIOCB2/ TCLKD

The pin function is switched as shown below according to the combination of the TPU channel 2 settings (bits MD3 to MD0 in TMDR2, bits IOB3 to IOB0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bits TPSC2 to TPSC0 in TCR0 and TCR5, and bit P17DDR.

TPU channel 2 settings	(1) in table below	(2) in table below	
P17DDR	—	0	1
Pin function	TIOCB2 output	P17 input	P17 output
		TIOCB2 input*1	
	TCLKD input*2		

Notes: *1 TIOCB2 input when MD3 to MD0 = B'0000 or B'01xx and IOB3 = 1.

*2 TCLKD input when the setting for either TCR0 or TCR5 is: TPSC2 to TPSC0 = B'111.

Also, 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	—

x: Don't care

Pin	Pin Functions and Selection Method
P16/ TIOCA2/ $\overline{\text{IRQ1}}$	The pin function is switched as shown below according to the combination of the TPU channel 2 settings (bits MD3 to MD0 in TMDR2, bits IOA3 to IOA0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2) and bit P16DDR.

TPU channel 2 settings	(1) in table below	(2) in table below	
P16DDR	—	0	1
Pin function	TIOCA2 output	P16 input	P16 output
		TIOCA2 input*1	
	$\overline{\text{IRQ1}}$ input*2		

TPU channel 2 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 mode 1 output*3	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCA2 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 = 1.

*2 When used as an external interrupt pin, do not use for another function.

*3 Output is disabled for TIOCB2.

Pin	Pin Functions and Selection Method
P15/ TIOCB1/ TCLKC	The pin function is switched as shown below according to the combination of the TPU channel 1 settings (bits MD3 to MD0 in TMDR1, bits IOB3 to IOB0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bits TPSC2 to TPSC0 in TCR0, TCR2, TCR4, and TCR5, and bit P15DDR.

TPU channel 1 settings	(1) in table below	(2) in table below	
P15DDR	—	0	1
Pin function	TIOCB1 output	P15 input	P15 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 TCR0 or TCR2 is: TPSC2 to TPSC0 = B'110, or the setting for either TCR4 or TCR5 is: TPSC2 to TPSC0 = B'101.

Also, TCLKC input when channels 2 and 4 are set to phase counting mode.

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	—

x: Don’t care

Pin	Pin Functions and Selection Method

P14/ TIOCA1/ $\overline{\text{IRQ0}}$

The pin function is switched as shown below according to the combination of the TPU channel 1 settings (bits MD3 to MD0 in TMDR1, bits IOA3 to IOA0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1) and bit P14DDR.

TPU channel 1 settings	(1) in table below	(2) in table below	
P14DDR	—	0	1
Pin function	TIOCA1 output	P14 input	P14 output
		TIOCA1 input*1	
	$\overline{\text{IRQ0}}$ input*2		

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 mode 1 output*3	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCA1 input when MD3 to MD0 = B'0000 or B'01xx and IOA3 to IOA0= B'10xx.

*2 When used as an external interrupt pin, do not use for another function

*3 Output is disabled for TIOCB1.

Pin	Pin Functions and Selection Method

P13/ TIOCD0/ TCLKB/ A23

The pin function is switched as shown below according to the combination of the operating mode, the TPU channel 0 settings (bits MD3 to MD0 in TMDR0, bits IOD3 to IOD0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR2, bits AE3 to AE0 in PFCR, and bit P13DDR.

<table>
<tr><td>Operating mode</td><td colspan="3">Modes 4, 5, 6</td><td></td><td colspan="3">Mode 7</td></tr>
<tr><td>AE3 to AE0</td><td colspan="3">Other than B'1111</td><td>B'1111</td><td colspan="3">—</td></tr>
<tr><td>TPU channel 0 settings</td><td>(1)
in table below</td><td colspan="2">(2)
in table below</td><td>—</td><td>(1)
in table below</td><td colspan="2">(2)
in table below</td></tr>
<tr><td>P13DDR</td><td>—</td><td>0</td><td>1</td><td>—</td><td>—</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="2">TIOCD0
output</td><td>P13
input</td><td>P13
output</td><td>—</td><td rowspan="2">TIOCD0
output</td><td>P13
input</td><td>P13
output</td></tr>
<tr><td colspan="2">TIOCD0
input*1</td><td>—</td><td colspan="2">TIOCD0
input*1</td></tr>
<tr><td colspan="3">TCLKB input*2</td><td>A23
output</td><td colspan="3">TCLKB input*2</td></tr>
</table>

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 TCR0 to TCR2 is: TPSC2 to TPSC0 = B'101.

Also, TCLKB input when channels 1 and 5 are set to phase counting mode.

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 to CCLR0	—	—	—	—	Other than B'110	B'110
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don’t care

Pin	Pin Functions and Selection Method

P12/ TIOCC0/ TCLKA/ A22

The pin function is switched as shown below according to the combination of the operating mode, the TPU channel 0 settings (bits MD3 to MD0 in TMDR0, bits IOC3 to IOC0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR5, bits AE3 to AE0 in PFCR, and bit P12DDR.

<table>
<tr><td>Operating mode</td><td colspan="3">Modes 4, 5, 6</td><td>Mode 7</td><td colspan="2"></td></tr>
<tr><td>AE3 to AE0</td><td colspan="2">Other than B'1111</td><td>B'1111</td><td colspan="3">—</td></tr>
<tr><td>TPU channel 0 settings</td><td>(1) in table below</td><td colspan="2">(2) in table below</td><td>—</td><td>(1) in table below</td><td colspan="2">(2) in table below</td></tr>
<tr><td>P12DDR</td><td>—</td><td>0</td><td>1</td><td>—</td><td>—</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="2">TIOCC0 output</td><td>P12 input</td><td>P12 output</td><td>—</td><td rowspan="2">TIOCC0 output</td><td>P12 input</td><td>P12 output</td></tr>
<tr><td colspan="2">TIOCC0 input*1</td><td>—</td><td colspan="2">TIOCC0 input*1</td></tr>
<tr><td colspan="3">TCLKA input*2</td><td>A22 output</td><td colspan="3">TCLKA input*2</td></tr>
</table>

<table>
<tr><td>TPU channel 0 settings</td><td>(2)</td><td colspan="2">(1)</td><td>(2)</td><td>(1)</td><td colspan="2">(1)</td><td>(2)</td></tr>
<tr><td>MD3 to MD0</td><td colspan="2">B'0000</td><td>B'001x</td><td>B'0010</td><td colspan="3">B'0011</td></tr>
<tr><td>IOC3 to IOC0</td><td>B'0000
B'0100
B'1xxx</td><td colspan="2">B'0001 to B'0011
B'0101 to B'0111</td><td>B'xx00</td><td>Other than B'xx00</td><td colspan="3">Other than B'xx00</td></tr>
<tr><td>CCLR2 to CCLR0</td><td>—</td><td colspan="2">—</td><td>—</td><td>—</td><td colspan="2">Other than B'101</td><td>B'101</td></tr>
<tr><td>Output function</td><td>—</td><td colspan="2">Output compare output</td><td>—</td><td>PWM mode 1 output*3</td><td colspan="2">PWM mode 2 output</td><td>—</td></tr>
</table>

x: Don't care

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 TCR0 to TCR5 is: TPSC2 to TPSC0 = B'100.

Also, TCLKA input when channels 1 and 5 are set to phase counting mode.

*3 Output is disabled for TIOCD0.

When BFA = 1 or BFB = 1 in TMDR0, output is disabled and the settings in (2) apply.

Pin	Pin Functions and Selection Method

P11/ TIOCB0/ A21

The pin function is switched as shown below according to the combination of the operating mode, the TPU channel 0 settings (bits MD3 to MD0 in TMDR0 and bits IOB3 to IOB0 in TIOR0H), bits AE3 to AE0 in PFCR, and bit P11DDR.

Operating mode	Modes 4, 5, 6			
AE3 to AE0	B'0000 to B'1101			B'1110 to B'1111
TPU channel0 settings	(1) in table below	(2) in table below		—
P11DDR	—	0	1	—
Pin function	TIOCB0 output	P11 input	P11 output	A21 output
		TIOCB0 input*1		

Operating mode	Mode 7		
AE3 to AE0	—		
TPU channel0 settings	(1) in table below	(2) in table below	
P11DDR	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output
		TIOCB0 input*1	

Note: *1 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 to CCLR0	—	—	—	—	Other than B'010	B'010
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

Pin	Pin Functions and Selection Method

P10/ TIOCA0/ A20

The pin function is switched as shown below according to the combination of the operating mode, the TPU channel 0 settings (bits MD3 to MD0 in TMDR0, bits IOA3 to IOA0 in TIOR0H, and bits CCLR2 to CCLR0 in TCR0), bits AE3 to AE0 in PFCR, and bit P10DDR.

Operating mode	Modes 4, 5, 6			
AE3 to AE0	B'0000 to B'1100			B'1101 to B'1111
TPU channel0 settings	(1) in table below	(2) in table below		—
P10DDR	—	0	1	—
Pin function	TIOCA0 output	P10 input	P10 output	A20 output
		TIOCA0 input*1		

Operating mode	Mode 7		
AE3 to AE0	—		
TPU channel0 settings	(1) in table below	(2) in table below	
P10DDR	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output
		TIOCA0 input*1	

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 to CCLR0	—	—	—	—	Other than B'001	B'001
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCA0 input when MD3 to MD0 = B'0000 and IOA3 to IOA0 = B'10xx.

*2 Output is disabled for TIOCB0.

9.3 Port 3

9.3.1 Overview

Port 3 is a 7-bit I/O port. Port 3 pins also function as SCI I/O pins (TxD0, RxD0, SCK0, TxD1, RxD1, and SCK1), external interrupt input pins ($\overline{\text{IRQ4}}$ and $\overline{\text{IRQ5}}$), and an external expansion interrupt input pin ($\overline{\text{EXIRQ7}}$). Port 3 pin functions are the same in all operating modes.

The interrupt input pins ($\overline{\text{IRQ4}}$ and $\overline{\text{IRQ5}}$) and the external expansion interrupt input pin ($\overline{\text{EXIRQ7}}$) are Schmitt-triggered inputs.

Figure 9-2 shows the port 3 pin configuration.

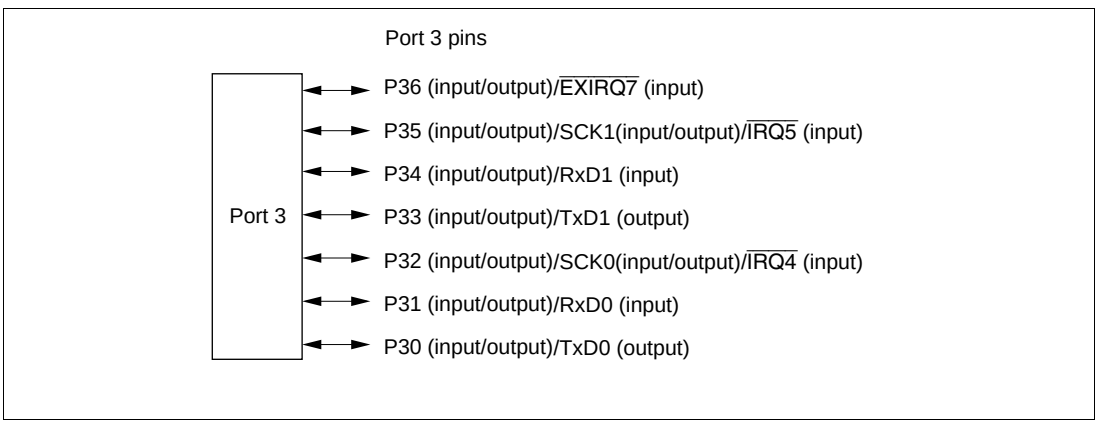


Figure 9-2 Port 3 Pin Functions

9.3.2 Register Configuration

Table 9-4 shows the port 3 register configuration.

Table 9-4 Port 3 Registers

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port 3 data direction register	P3DDR	W	H'00	H'FE32
Port 3 data register	P3DR	R/W	H'00	H'FF02
Port 3 register	PORT3	R	H'00	H'FFB2
Port 3 open-drain control register	P3ODR	R/W	H'00	H'FE46
Interrupt request input pin select register 0	IPINTSEL0	R/W	H'00	H'FE4A

Notes: *1 Lower 16 bits of the address.
*2 Value of bits 6 to 0.

(1) Port 3 Data Direction Register (P3DDR)

Bit	:	7	6	5	4	3	2	1	0
		—	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	:	Undefined	0	0	0	0	0	0	0
R/W	:	—	W	W	W	W	W	W	W

P3DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 3. P3DDR cannot be read; if it is, an undefined value will be returned. Bit 7 is reserved; this bit cannot be modified and will return an undefined value if read.

Setting a P3DDR bit to 1 makes the corresponding port 3 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P3DDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode. As the SCI is initialized by a manual reset, the pin states in this case are determined by the P3DDR and P3DR specifications.

(2) Port 3 Data Register (P3DR)

Bit	:	7	6	5	4	3	2	1	0
		—	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
Initial value	:	Undefined	0	0	0	0	0	0	0
R/W	:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3DR is an 8-bit readable/writable register that stores output data for the port 3 pins (P36 to P30). Bit 7 is reserved; this bit cannot be modified and will return an undefined value if read.

P3DR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(3) Port 3 Register (PORT3)

Bit	:	7	6	5	4	3	2	1	0
		—	P36	P35	P34	P33	P32	P31	P30
Initial value	:	Undefined	—*	—*	—*	—*	—*	—*	—*
R/W	:	—	R	R	R	R	R	R	R

Note: * Determined by the state of pins P36 to P30.

PORT3 is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 3 pins (P36 to P30) must always be performed on P3DR. Bit 7 is reserved; this bit cannot be modified and will return an undefined value if read.

If a port 3 read is performed while P3DDR bits are set to 1, the P3DR values are read. If a port 3 read is performed while P3DDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORT3 contents are determined by the pin states, as P3DDR and P3DR are initialized. PORT3 retains its previous state after a manual reset and in software standby mode.

(4) Port 3 Open-Drain Control Register (P3ODR)

Bit	:	7	6	5	4	3	2	1	0
		—	P36ODR	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR
Initial value	:	Undefined	0	0	0	0	0	0	0
R/W	:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3ODR is an 8-bit readable/writable register that controls the PMOS on/off status for each port 3 pin (P36 to P30). Bit 7 is reserved; this bit cannot be modified and will return an undefined value if read.

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.

P3ODR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(5) Interrupt Request Input Pin Select Register 0 (IPINSEL0)

Bit	:	7	6	5	4	3	2	1	0
		P36 IRQ7E	P47 IRQ6E	P46 IRQ5E	P44 IRQ4E	P43 IRQ3E	P42 IRQ2E	P41 IRQ1E	P40 IRQ0E
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IPINSEL0 is an 8-bit readable/writable register that selects which pins are to be used for interrupt request input signals ($\overline{\text{EXIRQ0}}$ to $\overline{\text{EXIRQ7}}$) from externally connected modules. IPINSEL0 is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state in a manual reset and in software standby mode.

Bit 7—Enable of $\overline{\text{EXIRQ7}}$ Input from P36 (P36IRQ7E): Selects whether or not P36 is used as the $\overline{\text{EXIRQ7}}$ input pin.

Bit 7

P36IRQ7E	Description	
0	P36 is not used as $\overline{\text{EXIRQ7}}$ input	(Initial value)
1	P36 is used as $\overline{\text{EXIRQ7}}$ input	

Bit 6—Enable of $\overline{\text{EXIRQ6}}$ Input from P47 (P47IRQ6E): Selects whether or not P47 is used as the $\overline{\text{EXIRQ6}}$ input pin.

Bit 6

P47IRQ6E	Description	
0	P47 is not used as $\overline{\text{EXIRQ6}}$ input	(Initial value)
1	P47 is used as $\overline{\text{EXIRQ6}}$ input	

Bit 5— Enable of $\overline{\text{EXIRQ5}}$ Input from P46 (P46IRQ5E): Selects whether or not P46 is used as the $\overline{\text{EXIRQ5}}$ input pin.

Bit 5

P46IRQ5E	Description	
0	P46 is not used as $\overline{\text{EXIRQ5}}$ input	(Initial value)
1	P46 is used as $\overline{\text{EXIRQ5}}$ input	

Bit 4—Enable of $\overline{\text{EXIRQ4}}$ Input from P44 (P44IRQ4E): Selects whether or not P44 is used as the $\overline{\text{EXIRQ4}}$ input pin.

Bit 4

P44IRQ4E	Description	
0	P44 is not used as $\overline{\text{EXIRQ4}}$ input	(Initial value)
1	P44 is used as $\overline{\text{EXIRQ4}}$ input	

Bit 3—Enable of $\overline{\text{EXIRQ3}}$ Input from P43 (P43IRQ3E): Selects whether or not P43 is used as the $\overline{\text{EXIRQ3}}$ input pin.

Bit 3

P43IRQ3E	Description	
0	P43 is not used as $\overline{\text{EXIRQ3}}$ input	(Initial value)
1	P43 is used as $\overline{\text{EXIRQ3}}$ input	

Bit 2—Enable of $\overline{\text{EXIRQ2}}$ Input from P42 (P42IRQ2E): Selects whether or not P42 is used as the $\overline{\text{EXIRQ2}}$ input pin.

Bit 2

P42IRQ2E	Description	
0	P42 is not used as $\overline{\text{EXIRQ2}}$ input	(Initial value)
1	P42 is used as $\overline{\text{EXIRQ2}}$ input	

Bit 1—Enable of $\overline{\text{EXIRQ1}}$ Input from P41 (P41IRQ1E): Selects whether or not P41 is used as the $\overline{\text{EXIRQ1}}$ input pin.

Bit 1

P41IRQ1E	Description	
0	P41 is not used as $\overline{\text{EXIRQ1}}$ input	(Initial value)
1	P41 is used as $\overline{\text{EXIRQ1}}$ input	

Bit 0—Enable of $\overline{\text{EXIRQ0}}$ Input from P40 (P40IRQ0E): Selects whether or not P40 is used as the $\overline{\text{EXIRQ0}}$ input pin.

Bit 0

P40IRQ0E	Description	
0	P40 is not used as $\overline{\text{EXIRQ0}}$ input	(Initial value)
1	P40 is used as $\overline{\text{EXIRQ0}}$ input	

9.3.3 Pin Functions

Port 3 pins also function as SCI I/O pins (TxD0, RxD0, SCK0, TxD1, RxD1, and SCK1), interrupt input pins ($\overline{\text{IRQ4}}$ and $\overline{\text{IRQ5}}$), and an external expansion interrupt input pin ($\overline{\text{EXIRQ7}}$). Port 3 pin functions are shown in table 9-5.

Table 9-5 Port 3 Pin Functions

Pin — **Pin Functions and Selection Method**

P36

The pin function is switched as shown below according to the combinations of the P36DDR bit and bit P36IRQ7E in IPINSELQ.

P36IRQ7E	0		1
P36DDR	0	1	—
Pin function	P36 input	P36 output*	$\overline{\text{EXIRQ7}}$ input

Note: * NMOS open-drain output when P36ODR = 1.

P35/SCK1/$\overline{\text{IRQ5}}$

The pin function is switched as shown below according to the combination of bit C/$\overline{\text{A}}$ in SMR of SCI1, bits CKE0 and CKE1 in SCR, and bit P35DDR.

CKE1	0				1
C/$\overline{\text{A}}$	0			1	—
CKE0	0		1	—	—
P35DDR	0	1	—	—	—
Pin function	P35 input	P35 output*[1]	SCK1 output*[1]	SCK1 output*[1]	SCK1 input
	$\overline{\text{IRQ5}}$ input*[2]				

Notes: *1 NMOS open-drain output when P35ODR = 1.

*2 When used as an external interrupt pin, do not use for another function.

P34/RxD1

The pin function is switched as shown below according to the combination of bit RE in SCR of SCI1 and bit P34DDR.

RE	0		1
P34DDR	0	1	—
Pin function	P34 input	P34 output*	RxD1 input

Note: * NMOS open-drain output when P34ODR = 1.

Pin	Pin Functions and Selection Method

P33/TxD1 The pin function is switched as shown below according to the combination of bit TE in SCR of SCI1 and bit P33DDR.

TE	0		1
P33DDR	0	1	—
Pin function	P33 input	P33 output*	TxD1 output*

Note: * NMOS open-drain output when P33ODR = 1.

P32/SCK0/$\overline{\text{IRQ4}}$ The pin function is switched as shown below according to the combination of bit C/$\overline{\text{A}}$ in SMR of SCI0, bits CKE0 and CKE1 in SCR, and bit P32DDR.

CKE1	0				1
C/$\overline{\text{A}}$	0			1	—
CKE0	0		1	—	—
P32DDR	0	1	—	—	—
Pin function	P32 input	P32 output*[1]	SCK0 output*[1]	SCK0 output*[1]	SCK0 input
	$\overline{\text{IRQ4}}$ input*[2]				

Notes: *1 NMOS open-drain output when P32ODR = 1.

*2 When used as an external interrupt pin, do not use for another function.

P31/RxD0 The pin function is switched as shown below according to the combination of bit RE in SCR of SCI0 and bit P31DDR.

RE	0		1
P31DDR	0	1	—
Pin function	P31 input	P31 output*	RxD0 input

Note: * NMOS open-drain output when P31ODR = 1.

P30/TxD0 The pin function is switched as shown below according to the combination of bit TE in SCR of SCI0 and bit P30DDR.

TE	0		1
P30DDR	0	1	—
Pin function	P30 input	P30 output*	TxD0 output*

Note: * NMOS open-drain output when P30ODR = 1.

9.4 Port 4

9.4.1 Overview

Port 4 is an 8-bit input-only port. Port 4 pins also function as external expansion interrupt input pins ($\overline{\text{EXIRQ6}}$ to $\overline{\text{EXIRQ0}}$). Port 4 pin functions are the same in all operating modes. Figure 9-3 shows the port 4 pin configuration.

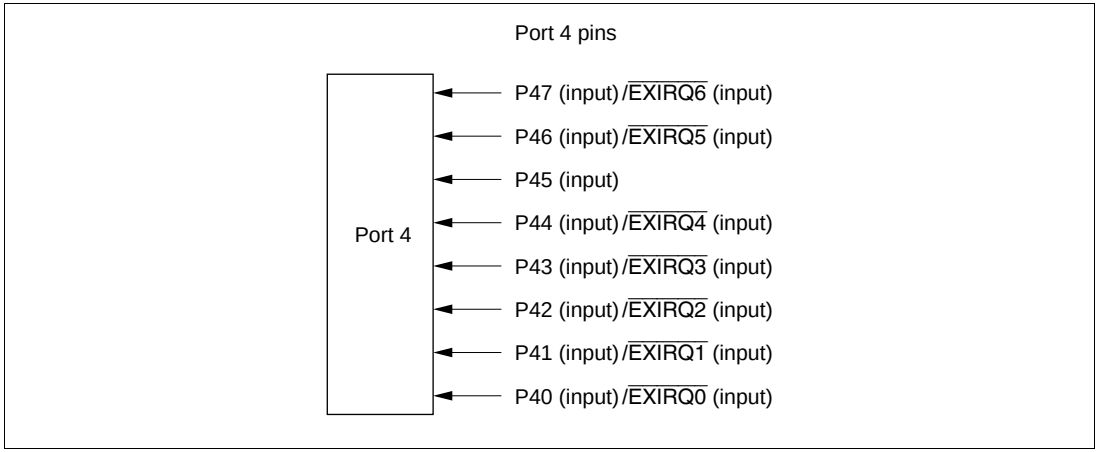


Figure 9-3 Port 4 Pin Functions

9.4.2 Register Configuration

Table 9-6 shows the port 4 register configuration. Port 4 is an input-only register, and does not have a data direction register or data register.

Table 9-6 Port 4 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 4 register	PORT4	R	Undefined	H'FFB3
Interrupt request input pin select register 0	IPINSEL0	R/W	H'00	H'FE4A

Note: * Lower 16 bits of the address.

(1) Port 4 Register (PORT4)

Bit	:	7	6	5	4	3	2	1	0
		P47	P46	P45	P44	P43	P42	P41	P40
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins P47 to P40.

PORT4 is an 8-bit read-only register. The pin states are always read when a port 4 read is performed. This register cannot be written to.

(2) Interrupt Request Input Pin Select Register 0 (IPINSEL0)

Bit	:	7	6	5	4	3	2	1	0
		P36 IRQ7E	P47 IRQ6E	P46 IRQ5E	P44 IRQ4E	P43 IRQ3E	P42 IRQ2E	P41 IRQ1E	P40 IRQ0E
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IPINSEL0 is an 8-bit readable/writable register that selects which pins are to be used for interrupt request input signals ($\overline{\text{EXIRQ0}}$ to $\overline{\text{EXIRQ7}}$) from externally connected modules. IPINSEL0 is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state in a manual reset and in software standby mode.

Bit 7—Enable of $\overline{\text{EXIRQ7}}$ Input from P36 (P36IRQ7E): Selects whether or not P36 is used as the $\overline{\text{EXIRQ7}}$ input pin.

Bit 7

P36IRQ7E	Description	
0	P36 is not used as $\overline{\text{EXIRQ7}}$ input	(Initial value)
1	P36 is used as $\overline{\text{EXIRQ7}}$ input	

Bit 6—Enable of $\overline{\text{EXIRQ6}}$ Input from P47 (P47IRQ6E): Selects whether or not P47 is used as the $\overline{\text{EXIRQ6}}$ input pin.

Bit 6

P47IRQ6E	Description	
0	P47 is not used as $\overline{\text{EXIRQ6}}$ input	(Initial value)
1	P47 is used as $\overline{\text{EXIRQ6}}$ input	

Bit 5— Enable of $\overline{\text{EXIRQ5}}$ Input from P46 (P46IRQ5E): Selects whether or not P46 is used as the $\overline{\text{EXIRQ5}}$ input pin.

Bit 5

P46IRQ5E	Description	
0	P46 is not used as $\overline{\text{EXIRQ5}}$ input	(Initial value)
1	P46 is used as $\overline{\text{EXIRQ5}}$ input	

Bit 4—Enable of $\overline{\text{EXIRQ4}}$ Input from P44 (P44IRQ4E): Selects whether or not P44 is used as the $\overline{\text{EXIRQ4}}$ input pin.

Bit 4

P44IRQ4E	Description	
0	P44 is not used as $\overline{\text{EXIRQ4}}$ input	(Initial value)
1	P44 is used as $\overline{\text{EXIRQ4}}$ input	

Bit 3—Enable of $\overline{\text{EXIRQ3}}$ Input from P43 (P43IRQ3E): Selects whether or not P43 is used as the $\overline{\text{EXIRQ3}}$ input pin.

Bit 3

P43IRQ3E	Description	
0	P43 is not used as $\overline{\text{EXIRQ3}}$ input	(Initial value)
1	P43 is used as $\overline{\text{EXIRQ3}}$ input	

Bit 2—Enable of $\overline{\text{EXIRQ2}}$ Input from P42 (P42IRQ2E): Selects whether or not P42 is used as the $\overline{\text{EXIRQ2}}$ input pin.

Bit 2

P42IRQ2E	Description	
0	P42 is not used as $\overline{\text{EXIRQ2}}$ input	(Initial value)
1	P42 is used as $\overline{\text{EXIRQ2}}$ input	

Bit 1—Enable of $\overline{\text{EXIRQ1}}$ Input from P41 (P41IRQ1E): Selects whether or not P41 is used as the $\overline{\text{EXIRQ1}}$ input pin.

Bit 1

P41IRQ1E	Description	
0	P41 is not used as $\overline{\text{EXIRQ1}}$ input	(Initial value)
1	P41 is used as $\overline{\text{EXIRQ1}}$ input	

Bit 0—Enable of $\overline{\text{EXIRQ0}}$ Input from P40 (P40IRQ0E): Selects whether or not P40 is used as the $\overline{\text{EXIRQ0}}$ input pin.

Bit 0

P40IRQ0E	Description	
0	P40 is not used as $\overline{\text{EXIRQ0}}$ input	(Initial value)
1	P40 is used as $\overline{\text{EXIRQ0}}$ input	

9.4.3 Pin Functions

Port 4 pins also function as external expansion interrupt input pins ($\overline{\text{EXIRQ6}}$ to $\overline{\text{EXIRQ0}}$).

9.5 Port 7

9.5.1 Overview

Port 7 is an 8-bit I/O port. Port 7 pins also function as DMAC input pins ($\overline{\text{DREQ0}}$, $\overline{\text{TEND0}}$, $\overline{\text{DREQ1}}$, and $\overline{\text{TEND1}}$), bus control output pins ($\overline{\text{CS4}}$ to $\overline{\text{CS7}}$), external module output pins (EXMSTP, $\overline{\text{EXMS}}$, and $\overline{\text{EXDTCE}}$), and the manual reset input pin ($\overline{\text{MRES}}$). The functions of pins P77 to P74 are the same in all operating mode, but the functions of pins P73 to P70 depend on the operating mode.

Figure 9-4 shows the port 7 pin configuration.

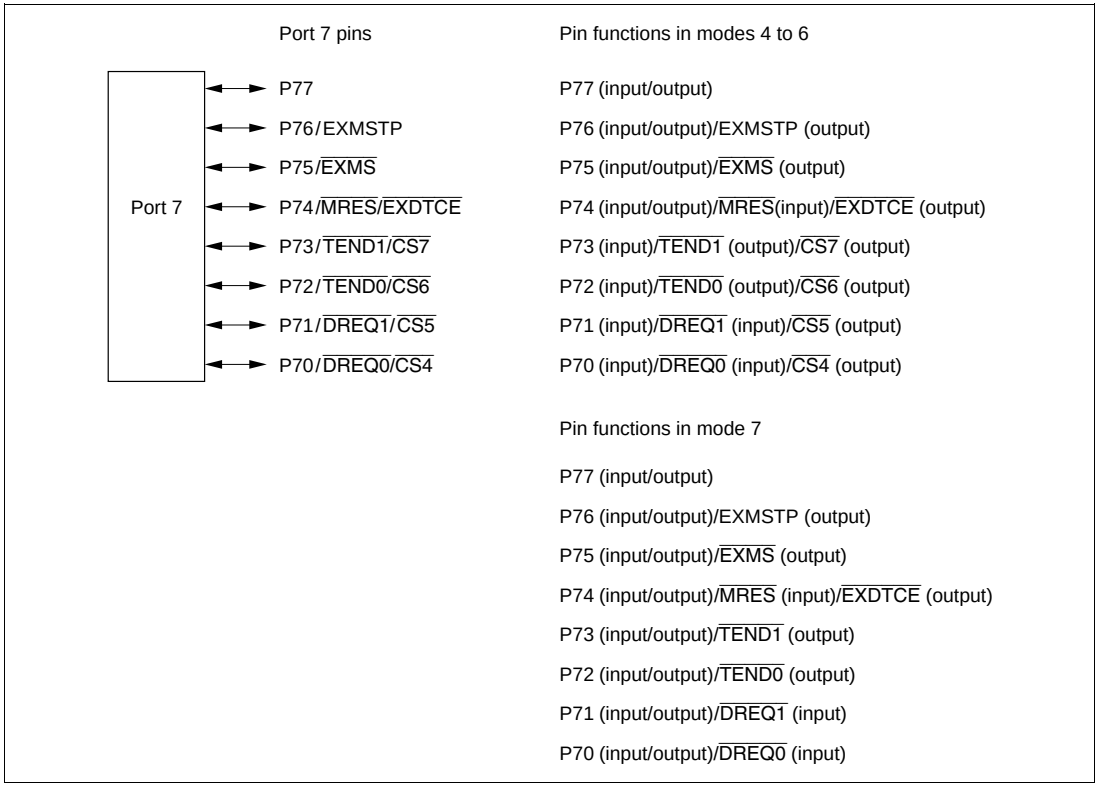


Figure 9-4 Port 7 Pin Functions

9.5.2 Register Configuration

Table 9-7 shows the port 7 register configuration.

Table 9-7 Port 7 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 7 data direction register	P7DDR	W	H'00	H'FE36
Port 7 data register	P7DR	R/W	H'00	H'FF06
Port 7 register	PORT7	R	Undefined	H'FFB6
External module connection output pin select register	OPINSEL	R/W	b'-000----	H'FE4E

Note: * Lower 16 bits of the address.

(1) Port 7 Data Direction Register (P7DDR)

Bit	:	7	6	5	4	3	2	1	0
		P77DDR	P76DDR	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P7DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 7. P7DDR cannot be read; if it is, an undefined value will be read.

Setting a P7DDR bit to 1 makes the corresponding port 7 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P7DDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode. As the 8-bit timer and SCI are initialized by a manual reset, the pin states in this case are determined by the P7DDR and P7DR specifications.

(2) Port 7 Data Register (P7DR)

Bit	:	7	6	5	4	3	2	1	0
		P77DR	P76DR	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P7DR is an 8-bit readable/writable register that stores output data for the port 7 pins (P77 to P70).

P7DR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(3) Port 7 Register (PORT7)

Bit	:	7	6	5	4	3	2	1	0
		P77	P76	P75	P74	P73	P72	P71	P70
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins P77 to P70.

PORT7 is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 7 pins (P77 to P70) must always be performed on P7DR.

If a port 7 read is performed while P7DDR bits are set to 1, the P7DR values are read. If a port 7 read is performed while P7DDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORT7 contents are determined by the pin states, as P7DDR and P7DR are initialized. PORT7 retains its previous state after a manual reset and in software standby mode.

(4) External Module Connection Output Pin Select Register (OPINSEL)

Bit	:	7	6	5	4	3	2	1	0
		—	P76 STPOE	P75 MSOE	P74 DTCOE	—	—	—	—
Initial value	:	Undefined	0	0	0	Undefined	Undefined	Undefined	Undefined
R/W	:	—	R/W	R/W	R/W	—	—	—	—

OPINSEL is an 8-bit readable/writable register that selects whether or not output signals ($\overline{\text{EXDTCE}}$, EXMSTP, $\overline{\text{EXMS}}$) to externally connected modules are output to pins P76 to P74. OPINSEL bits 6 to 4 are initialized to 000 by a power-on reset and in hardware standby mode. They retain their previous states in a manual reset and in software standby mode.

Bit 7—Reserved: This bit will return an undefined value if read, and should only be written with 0.

Bit 6—Enable of EXMSTP Output to P76 (P76STPOE): Selects whether or not the EXMSTP module stop signal to external modules (bit 0 in MSTPCRB) is output to P76.

Bit 6

P76STPOE	Description	
0	EXMSTP is not output to P76	(Initial value)
1	EXMSTP is output to P76	

Bit 5—Enable of $\overline{\text{EXMS}}$ Output to P75 (P75MSOE): Selects whether or not the $\overline{\text{EXMS}}$ module stop signal to external modules (corresponding to addresses H'FFFF40 to H'FFFF5F) is output to P75.

Bit 5

P75MSOE	Description	
0	$\overline{\text{EXMS}}$ is not output to P75	(Initial value)
1	$\overline{\text{EXMS}}$ is output to P75	

Bit 4—Enable of $\overline{\text{EXDTCE}}$ Output to P74 (P74DTCOE): Selects whether or not the $\overline{\text{EXDTCE}}$ signal, indicating that DTC transfer corresponding to EXIRQ0 to EXIRQF input is in progress, is output to P74. This signal is used, for example, when the DTC in the chip has been activated by an interrupt (EXIRQ0 to EXIRQF) from an external module, and the interrupt request is to be cleared automatically on the external module side by DTC transfer.

Bit 4

P74DTCOE	Description	
0	$\overline{\text{EXDTCE}}$ is not output to P74	(Initial value)
1	$\overline{\text{EXDTCE}}$ is output to P74	

Bits 3 to 0—Reserved: These bits will return an undefined value if read, and should only be written with 0.

9.5.3 Pin Functions

Port 7 pins also function as DMAC I/O pins ($\overline{\text{DREQ0}}$, $\overline{\text{TEND0}}$, $\overline{\text{DREQ1}}$, and $\overline{\text{TEND1}}$), bus control output pins ($\overline{\text{CS4}}$ to $\overline{\text{CS7}}$), external module output pins (EXMSTP, $\overline{\text{EXMS}}$, and $\overline{\text{EXDTCE}}$), and the manual reset input pin ($\overline{\text{MRES}}$). Port 7 pin functions are shown in table 9-8.

Table 9-8 Port 7 Pin Functions

Pin — **Pin Functions and Selection Method**

P77 — The pin function is switched as shown below according to the setting of bit P77DDR.

P77DDR	0	1
Pin function	P77 input	P77 output

P76/EXMSTP — The pin function is switched as shown below according to the combination of bit P76STPOE in OPINSEL and bit P76DDR.

P76STPOE	0		1
P76DDR	0	1	—
Pin function	P76 input	P76 output	EXMSTP output

P75/$\overline{\text{EXMS}}$ — The pin function is switched as shown below according to the combination of bit P75MSOE in OPINSEL and bit P75DDR.

P75MSOE	0		1
P75DDR	0	1	—
Pin function	P75 input	P75 output	$\overline{\text{EXMS}}$ output

P74/$\overline{\text{MRES}}$/$\overline{\text{EXDTCE}}$ — The pin function is switched as shown below according to the combination of bit MRESE in SYSCR and bit P74DDR.

P74DTCOE	0			1
MRESE	0		1	—
P74DDR	0	1	0	—
Pin function	P74 input	P74 output	$\overline{\text{MRES}}$ input	$\overline{\text{EXDTCE}}$ output

Pin	Pin Functions and Selection Method

P73/$\overline{\text{TEND1}}$/$\overline{\text{CS7}}$ The pin function is switched as shown below according to the combination of the operating mode, bit TEE1 in DMATCR of the DMAC, and bit P73DDR.

Operating mode	Modes 4, 5, 6			Mode 7		
TEE1	0		1	0		1
P73DDR	0	1	—	0	1	—
Pin function	P73 input	$\overline{\text{CS7}}$ output	$\overline{\text{TEND1}}$ output	P73 input	P73 output	$\overline{\text{TEND1}}$ output

P72/$\overline{\text{TEND0}}$/$\overline{\text{CS6}}$ The pin function is switched as shown below according to the combination of the operating mode, bit TEE0 in DMATCR of the DMAC, and bit P72DDR.

Operating mode	Modes 4, 5, 6			Mode 7		
TEE0	0		1	0		1
P72DDR	0	1	—	0	1	—
Pin function	P72 input	$\overline{\text{CS6}}$ output	$\overline{\text{TEND0}}$ output	P72 input	P72 output	$\overline{\text{TEND0}}$ output

P71/$\overline{\text{DREQ1}}$/$\overline{\text{CS5}}$ The pin function is switched as shown below according to the combination of the operating mode and bit P71DDR.

Operating mode	Modes 4, 5, 6		Mode 7	
P71DDR	0	1	0	1
Pin function	P71 input	$\overline{\text{CS5}}$ output	P71 input	P71 output
	$\overline{\text{DREQ1}}$ input			

P70/$\overline{\text{DREQ0}}$/$\overline{\text{CS4}}$ The pin function is switched as shown below according to the combination of the operating mode and bit P70DDR.

Operating mode	Modes 4, 5, 6		Mode 7	
P70DDR	0	1	0	1
Pin function	P70 input	$\overline{\text{CS4}}$ output	P70 input	P70 output
	$\overline{\text{DREQ0}}$ input			

9.6 Port 9

9.6.1 Overview

Port 9 is a 1-bit input-only port. Port 9 pins also function as D/A converter analog output pin (DA0). Port 9 pin functions are the same in all operating modes. Figure 9-5 shows the port 9 pin configuration.

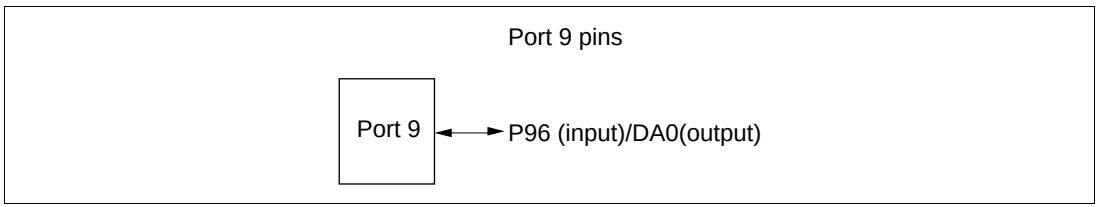


Figure 9-5 Port 9 Pin Functions

9.6.2 Register Configuration

Table 9-9 shows the port 9 register configuration. Port 9 is an input-only register, and does not have a data direction register or data register.

Table 9-9 Port 9 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 9 register	PORT9	R	Undefined	H'FFB8

Note: * Lower 16 bits of the address.

(1) Port 9 Register (PORT9)

Bit	:	7	6	5	4	3	2	1	0
		—	P96	—	—	—	—	—	—
Initial value	:	—	—*	—	—	—	—	—	—
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pin P96.

PORT9 is an 8-bit read-only register. The pin states are always read when a port 9 read is performed. This register cannot be written to. Bits 7 and 5 to 0 are reserved, and will return an undefined value if read.

9.6.3 Pin Functions

Port 9 pins also function as D/A converter analog output pin (DA0).

9.7 Port A

9.7.1 Overview

Port A is an 8-bit I/O port. Port A pins also function as address bus outputs and SCI2 I/O pins (SCK2, RxD2, and TxD2). The pin functions depend on the operating mode.

Port A has a built-in MOS input pull-up function that can be controlled by software.

Figure 9-6 shows the port A pin configuration.

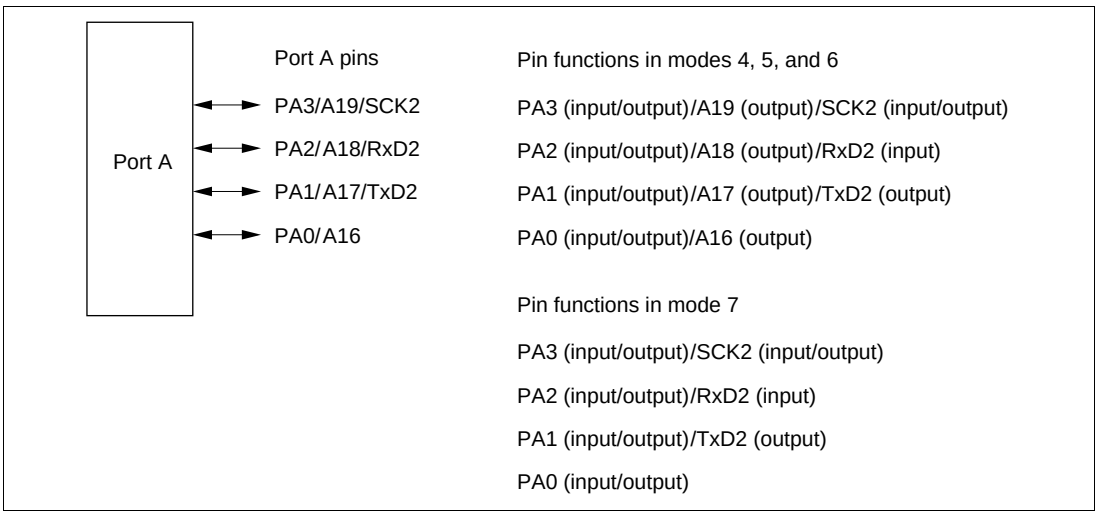


Figure 9-6 Port A Pin Functions

9.7.2 Register Configuration

Table 9-10 shows the port A register configuration.

Table 9-10 Port A Registers

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port A data direction register	PADDR	W	H'0	H'FE39
Port A data register	PADR	R/W	H'0	H'FF09
Port A register	PORTA	R	Undefined	H'FFB9
Port A MOS pull-up control register	PAPCR	R/W	H'0	H'FE40
Port A open-drain control register	PAODR	R/W	H'0	H'FE47

Notes: *1 Lower 16 bits of the address.
*2 Value of bits 3 to 0.

(1) Port A Data Direction Register (PADDR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	W	W	W	W

PADDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port A. PADDR cannot be read; if it is, an undefined value will be read.

Bits 7 to 4 are reserved; these bits cannot be modified and will return an undefined value if read.

PADDR is initialized to H'0 (bits 3 to 0) by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

(a) Modes 4, 5, and 6

If address output is enabled by the setting of bits AE3 to AE0 in PFCR, the corresponding port A pins are address outputs.

When address output is disabled, setting a PADDR bit to 1 makes the corresponding port A pin an output port, while clearing the bit to 0 makes the pin an input port.

(b) Mode 7

Setting a PADDR bit to 1 makes the corresponding port A pin an output port, while clearing the bit to 0 makes the pin an input port.

(2) Port A Data Register (PADR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

PADR is an 8-bit readable/writable register that stores output data for the port A pins (PA3 to PA0).

Bits 7 to 4 are reserved; these bits cannot be modified and will return an undefined value if read.

PADR is initialized to H'0 (bits 3 to 0) by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(3) Port A Register (PORTA)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3	PA2	PA1	PA0
Initial value	:	Undefined	Undefined	Undefined	Undefined	—*	—*	—*	—*
R/W	:	—	—	—	—	R	R	R	R

Note: * Determined by the state of pins PA3 to PA0.

PORTA is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port A pins (PA3 to PA0) must always be performed on PADR.

Bits 7 to 4 are reserved; these bits cannot be modified and will return an undefined value if read.

If a port A read is performed while PADDR bits are set to 1, the PADR values are read. If a port A read is performed while PADDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTA contents are determined by the pin states, as PADDR and PADR are initialized. PORTA retains its previous state after a manual reset and in software standby mode.

(4) Port A MOS Pull-Up Control Register (PAPCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

PAPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port A on a bit-by-bit basis.

Bits 7 to 4 are reserved; these bits cannot be modified and will return an undefined value if read.

PAPCR is valid for port input and SCI input pins. When a PADDR bit is cleared to 0 (input port setting), setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PAPCR is initialized to H'0 (bits 3 to 0) by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(5) Port A Open-Drain Control Register (PAODR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

PAODR is an 8-bit readable/writable register that controls the PMOS on/off status for each port A pin (PA3 to PA0).

Bits 7 to 4 are reserved; these bits cannot be modified and will return an undefined value if read.

PAODR is valid for port output and SCI output pins.

Setting a PAODR bit to 1 makes the corresponding port A pin an NMOS open-drain output pin, while clearing the bit to 0 makes the pin a CMOS output pin.

PAODR is initialized to H'0 (bits 3 to 0) by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

9.7.3 Pin Functions

Port A pins also function as SCI2 I/O pins (TxD2, RxD2, and SCK2) and address output pins (A19 to A16). Port A pin functions are shown in table 9-11.

Table 9-11 Port A Pin Functions

Pin	Pin Functions and Selection Method
PA3/A19/ SCK2	The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, SCI channel 2 settings, and bit PA3DDR.

Operating mode	Modes 4 to 6					
AE3 to AE0	11xx	Other than 11xx				
CKE1	—	0				1
C/A	—	0			1	—
CKE0	—	0		1	—	—
PA3DDR	—	0	1	—	—	—
Pin function	A19 output	PA3 input	PA3 output*	SCK2 output*	SCK2 output*	SCK2 input

Operating mode	Mode 7				
AE3 to AE0	—				
CKE1	0				1
C/A	0			1	—
CKE0	0		1	—	—
PA3DDR	0	1	—	—	—
Pin function	PA3 input	PA3 output*	SCK2 output*	SCK2 output*	SCK2 input

Note: * NMOS open-drain output when PA3ODR = 1 in PAODR.

PA2/A18/ RxD2 — The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, SCI channel 2 settings, and bit PA2DDR.

Operating mode	Modes 4 to 6				Mode 7		
AE3 to AE0	1011 or 11xx	Other than (1011 or 11xx)			—		
RE	—	0		1	0		1
PA2DDR	—	0	1	—	0	1	—
Pin function	A18 output	PA2 input	PA2 output*	RxD2 input	PA2 input	PA2 output*	RxD2 input

Note: * NMOS open-drain output when PA2ODR = 1 in PAODR.

Pin	Pin Functions and Selection Method
PA1/A17/ TxD2	The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, SCI channel 2 settings, and bit PA1DDR.

Operating mode	Modes 4 to 6				Mode 7		
AE3 to AE0	101x or 11xx	Other than (101x or 11xx)			—		
TE	—	0		1	0		1
PA1DDR	—	0	1	—	0	1	—
Pin function	A17 output	PA1 input	PA1 output*	TxD2 output*	PA1 input	PA1 output*	TxD2 output*

Note: * NMOS open-drain output when PA1ODR = 1 in PAODR.

PA0/A16 The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, and bit PA0DDR.

Operating mode	Modes 4 to 6			Mode 7	
AE3 to AE0	Other than (0xxx or 1000)	0xxx or 1000		—	
PA1DDR	—	0	1	0	1
Pin function	A16 output	PA0 input	PA0 output*	PA0 input	PA0 output*

Note: * NMOS open-drain output when PA0ODR = 1 in PAODR.

9.7.4 MOS Input Pull-Up Function

Port A has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be specified as on or off for individual bits.

With port input and SCI input pins, when a PADDR bit is cleared to 0, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a power-on reset and in hardware standby mode. The previous state is retained after a manual reset and in software standby mode.

Table 9-12 summarizes the MOS input pull-up states.

Table 9-12 MOS Input Pull-Up States (Port A)

Pins	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output, port output, SCI output	OFF	OFF	OFF	OFF	OFF
Port input, SCI input	OFF	OFF	ON/OFF	ON/OFF	ON/OFF

Legend:

OFF: MOS input pull-up is always off.

ON/OFF: On when PADDR = 0 and PAPCR = 1; otherwise off.

9.8 Port B

9.8.1 Overview

Port B is an 8-bit I/O port. Port B pins also function as address bus outputs. The pin functions depend on the operating mode.

Port B has a built-in MOS input pull-up function that can be controlled by software.

Figure 9-7 shows the port B pin configuration.

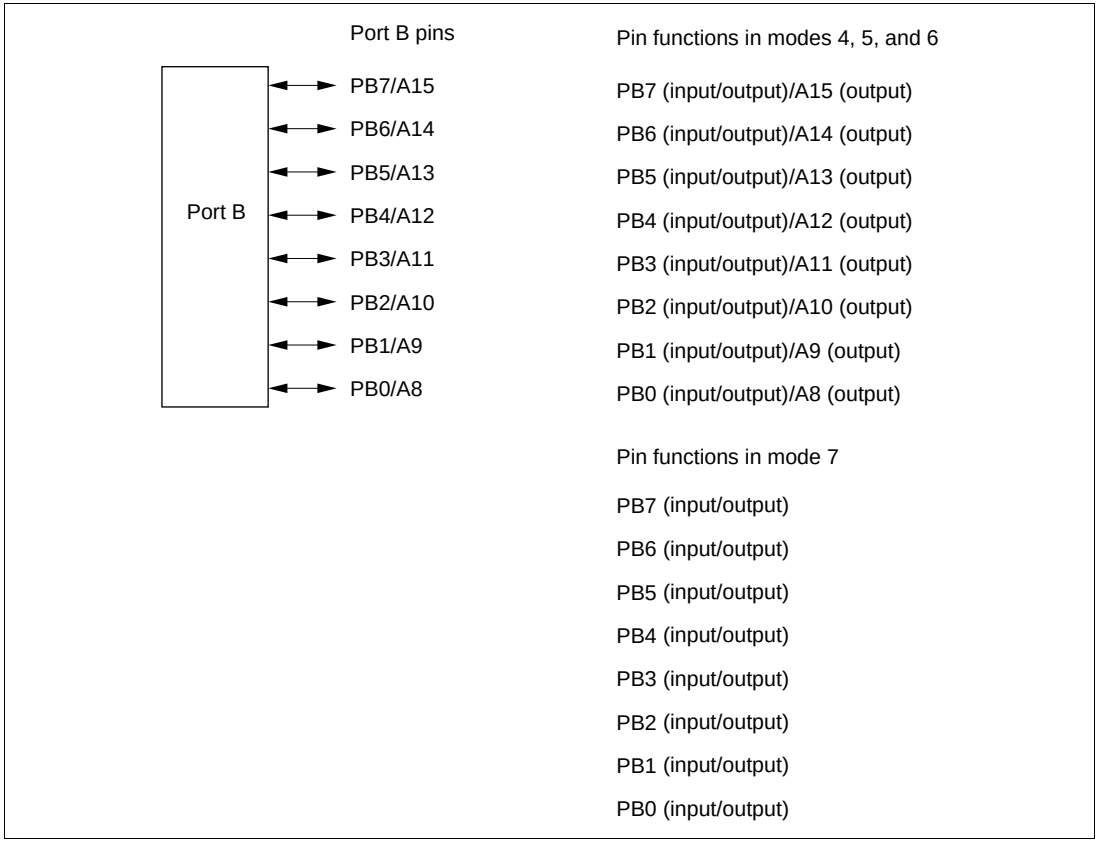


Figure 9-7 Port B Pin Functions

9.8.2 Register Configuration

Table 9-13 shows the port B register configuration.

Table 9-13 Port B Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port B data direction register	PBDDR	W	H'00	H'FE3A
Port B data register	PBDR	R/W	H'00	H'FF0A
Port B register	PORTB	R	Undefined	H'FFBA
Port B MOS pull-up control register	PBPCR	R/W	H'00	H'FE41

Note: * Lower 16 bits of the address.

(1) Port B Data Direction Register (PBDDR)

Bit	:	7	6	5	4	3	2	1	0
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PBDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port B. PBDDR cannot be read; if it is, an undefined value will be read.

PBDDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

(a) Modes 4, 5, and 6

If address output is enabled by the setting of bits AE3 to AE0 in PFCR, the corresponding port B pins are address outputs.

When address output is disabled, setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

(b) Mode 7

Setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

(2) Port B Data Register (PBDR)

Bit	:	7	6	5	4	3	2	1	0
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PBDR is an 8-bit readable/writable register that stores output data for the port B pins (PB7 to PB0).

PBDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(3) Port B Register (PORTB)

Bit	:	7	6	5	4	3	2	1	0
		PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins PB7 to PB0.

PORTB is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port B pins (PB7 to PB0) must always be performed on PBDR.

If a port B read is performed 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.

After a power-on reset and in hardware standby mode, PORTB contents are determined by the pin states, as PBDDR and PBDR are initialized. PORTB retains its previous state after a manual reset and in software standby mode.

(4) Port B MOS Pull-Up Control Register (PBPCR)

Bit	:	7	6	5	4	3	2	1	0
		PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PBPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port B on a bit-by-bit basis.

PBPCR is valid for port input and TPU input pins.

When a PBDDR bit is cleared to 0 (input port setting), setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PBPCR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

9.8.3 Pin Functions

Port B pins also function as address output pins (A15 to A8). Port B pin functions are shown in table 9-14.

Table 9-14 Port B Pin Functions

<table>
<tr><th>Pin</th><th>Pin Functions and Selection Method</th></tr>
<tr><td>PB7/A15</td><td>The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, and bit PB7DDR.
<table>
<tr><td>Operating mode</td><td colspan="3">Modes 4 to 6</td></tr>
<tr><td>AE3 to AE0 in PFCR</td><td>B'1xxx</td><td colspan="2">Other than B'1xxx</td></tr>
<tr><td>PB7DDR</td><td>—</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>A15 output</td><td>PB7 input</td><td>PB7 output</td></tr>
</table>
<table>
<tr><td>Operating mode</td><td colspan="2">Mode 7</td></tr>
<tr><td>AE3 to AE0 in PFCR</td><td colspan="2">—</td></tr>
<tr><td>PB7DDR</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>PB7 input</td><td>PB7 output</td></tr>
</table>
</td></tr>
</table>

Pin	Pin Functions and Selection Method

PB6/A14 The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, and bit PB6DDR.

Operating mode	Modes 4 to 6		
AE3 to AE0 in PFCR	B'0111 or B'1xxx	Other than (B'0111 or B'1xxx)	
PB6DDR	—	0	1
Pin function	A14 output	PB6 input	PB6 output

Operating mode	Mode 7	
AE3 to AE0 in PFCR	—	
PB6DDR	0	1
Pin function	PB6 input	PB6 output

PB5/A13 The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, and bit PB5DDR.

Operating mode	Modes 4 to 6		
AE3 to AE0 in PFCR	B'011x or B'1xxx	Other than (B'011x or B'1xxx)	
PB5DDR	—	0	1
Pin function	A13 output	PB5 input	PB5 output

Operating mode	Mode 7	
AE3 to AE0 in PFCR	—	
PB5DDR	0	1
Pin function	PB5 input	PB5 output

Pin	Pin Functions and Selection Method
PB4/A12	The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, and bit PB4DDR.

Operating mode	Modes 4 to 6		
AE3 to AE0 in PFCR	B'0100 or B'00xx		Other than (B'0100 or B'00xx)
PB5DDR	0	1	—
Pin function	PB4 input	PB4 output	A12 output

Operating mode	Mode 7	
AE3 to AE0 in PFCR	—	
PB4DDR	0	1
Pin function	PB4 input	PB4 output

PB3/A11 The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, and bit PB3DDR.

Operating mode	Modes 4 to 6		
AE3 to AE0 in PFCR	B'00xx		Other than B'00xx
PB3DDR	0	1	—
Pin function	PB3 input	PB3 output	A11 output

Operating mode	Mode 7	
AE3 to AE0 in PFCR	—	
PB3DDR	0	1
Pin function	PB3 input	PB3 output

Pin	Pin Functions and Selection Method

PB2/A10 The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, and bit PB2DDR.

Operating mode	Modes 4 to 6		
AE3 to AE0 in PFCR	B'0010 or B'000x		Other than B'0010 or B'000x
PB2DDR	0	1	—
Pin function	PB2 input	PB2 output	A10 output

Operating mode	Mode 7	
AE3 to AE0 in PFCR	—	
PB2DDR	0	1
Pin function	PB2 input	PB2 output

PB1/A9 The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, and bit PB1DDR.

Operating mode	Modes 4 to 6		
AE3 to AE0 in PFCR	B'000x		Other than B'000x
PB1DDR	0	1	—
Pin function	PB1 input	PB1 output	A9 output

Operating mode	Mode 7	
AE3 to AE0 in PFCR	—	
PB1DDR	0	1
Pin function	PB1 input	PB1 output

Pin	Pin Functions and Selection Method
PB0/A8	The pin function is switched as shown below according to the combination of the operating mode, PFCR setting, bit PB1DDR.

Operating mode	Modes 4 to 6		
AE3 to AE0 in PFCR	B'0000		Other than B'0000
P30DDR	0	1	—
Pin function	PB0 input	PB0 output	A8 output

Operating mode	Mode 7	
AE3 to AE0 in PFCR	—	
PB0DDR	0	1
Pin function	PB0 input	PB0 output

9.8.4 MOS Input Pull-Up Function

Port B has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be specified as on or off for individual bits.

With port input pins, 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.

The MOS input pull-up function is in the off state after a power-on reset and in hardware standby mode. The previous state is retained after a manual reset and in software standby mode.

Table 9-15 summarizes the MOS input pull-up states.

Table 9-15 MOS Input Pull-Up States (Port B)

Pins	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output, port output	OFF	OFF	OFF	OFF	OFF
Port input	OFF	OFF	ON/OFF	ON/OFF	ON/OFF

Legend:

OFF: MOS input pull-up is always off.

ON/OFF: On when PBDDR = 0 and PBPCR = 1; otherwise off.

9.9 Port C

9.9.1 Overview

Port C is an 8-bit I/O port. Port C pins also function as address bus outputs. The pin functions depend on the operating mode.

Port C has a built-in MOS input pull-up function that can be controlled by software.

Figure 9-8 shows the port C pin configuration.

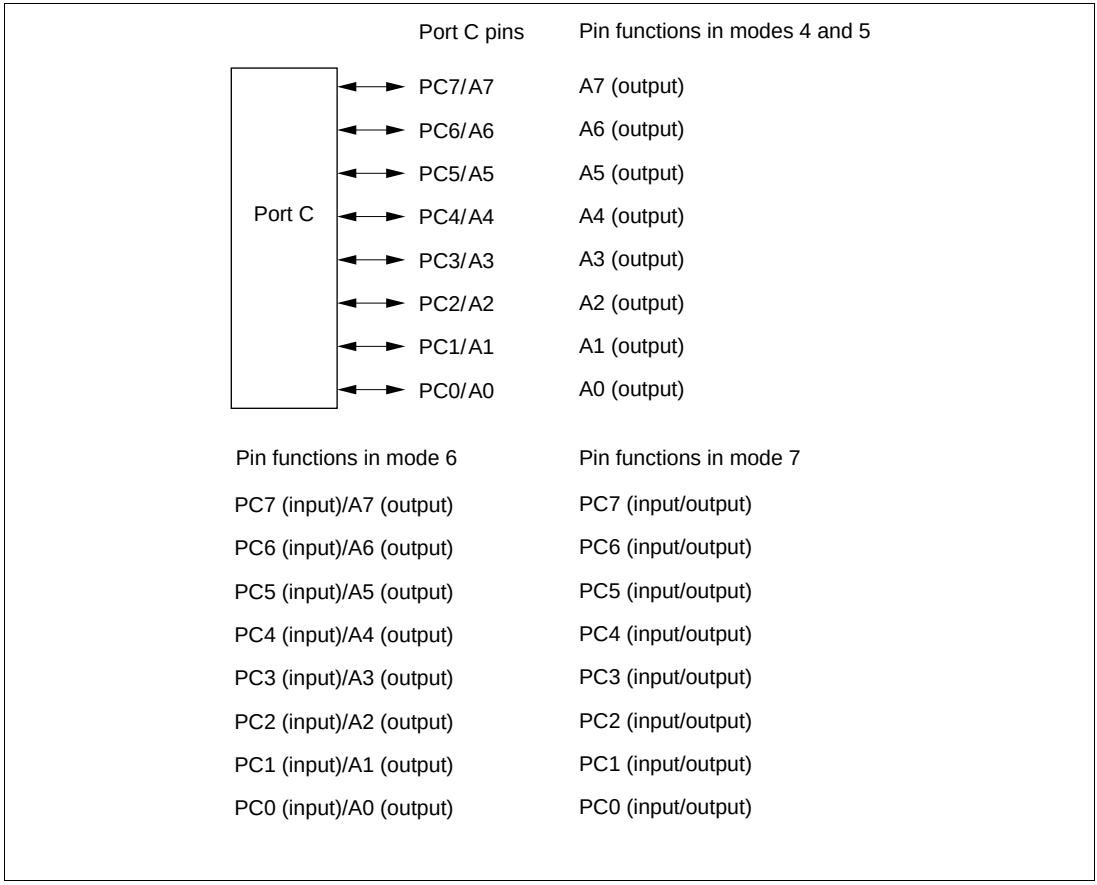


Figure 9-8 Port C Pin Functions

9.9.2 Register Configuration

Table 9-16 shows the port C register configuration.

Table 9-16 Port C Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port C data direction register	PCDDR	W	H'00	H'FE3B
Port C data register	PCDR	R/W	H'00	H'FF0B
Port C register	PORTC	R	Undefined	H'FFBB
Port C MOS pull-up control register	PCPCR	R/W	H'00	H'FE42

Note: * Lower 16 bits of the address.

(1) Port C Data Direction Register (PCDDR)

Bit	:	7	6	5	4	3	2	1	0
		PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PCDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port C. PCDDR cannot be read; if it is, an undefined value will be read.

PCDDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

(a) Modes 4 and 5

Port C pins are address outputs regardless of the PCDDR settings.

(b) Mode 6

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.

(c) Mode 7

Setting a PCDDR bit to 1 makes the corresponding port C pin an output port, while clearing the bit to 0 makes the pin an input port.

(2) Port C Data Register (PCDR)

Bit	:	7	6	5	4	3	2	1	0
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCDR is an 8-bit readable/writable register that stores output data for the port C pins (PC7 to PC0).

PCDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(3) Port C Register (PORTC)

Bit	:	7	6	5	4	3	2	1	0
		PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins PC7 to PC0.

PORTC is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port C pins (PC7 to PC0) must always be performed on PCDR.

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.

After a power-on reset and in hardware standby mode, PORTC contents are determined by the pin states, as PCDDR and PCDR are initialized. PORTC retains its previous state after a manual reset and in software standby mode.

(4) Port C MOS Pull-Up Control Register (PCPCR)

Bit	:	7	6	5	4	3	2	1	0
		PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port C on a bit-by-bit basis.

PCPCR is valid for port input (modes 6 and 7).

When a PCDDR bit is cleared to 0 (input port setting), setting the corresponding PCPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PCPCR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

9.9.3 Pin Functions in Each Mode

(1) Modes 4 and 5

In modes 4 and 5, port C pins function as address outputs automatically. Port C pin functions in modes 4 and 5 are shown in figure 9-9.

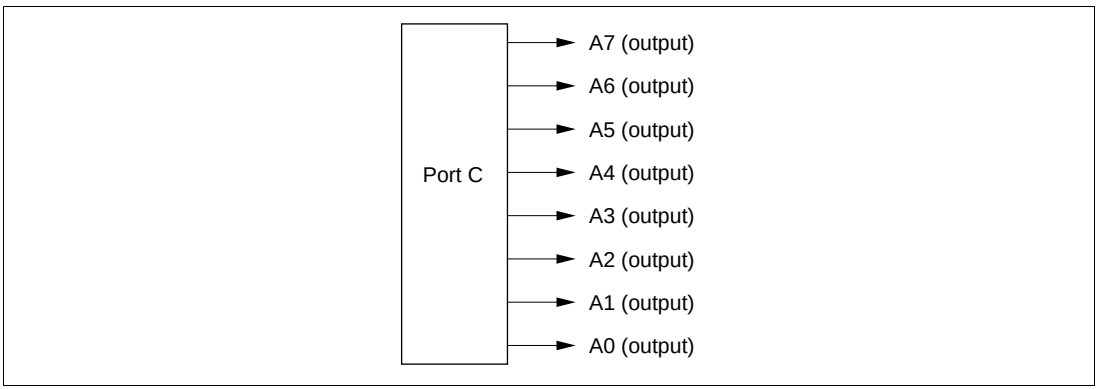


Figure 9-9 Port C Pin Functions (Modes 4 and 5)

(2) Mode 6

In mode 6, port C pins function as address outputs or input ports, and input or output can be specified bit by bit. 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.

Port C pin functions in mode 6 are shown in figure 9-10.

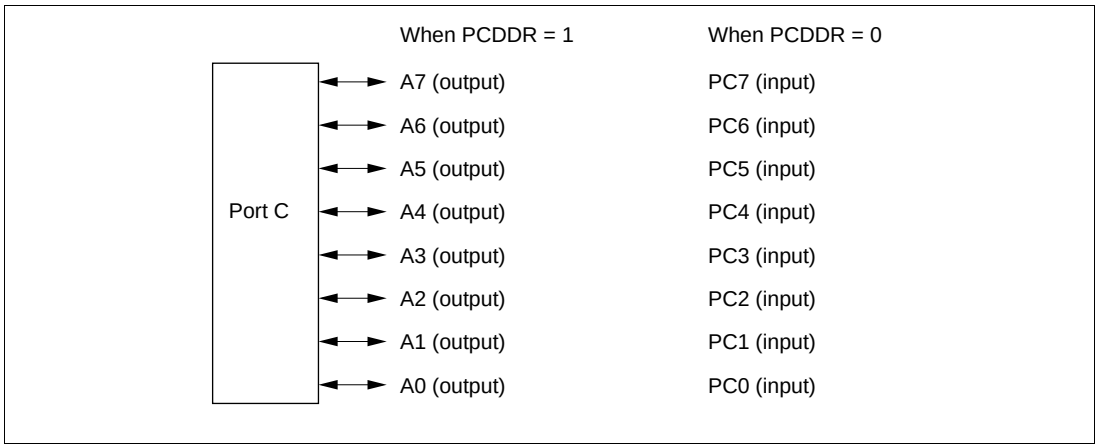


Figure 9-10 Port C Pin Functions (Mode 6)

(3) Mode 7

In mode 7, port C functions as an I/O port, and input or output can be specified bit by bit. Setting a PCDDR bit to 1 makes the corresponding port C pin an output port, while clearing the bit to 0 makes the pin an input port.

Port C pin functions in mode 7 are shown in figure 9-11.

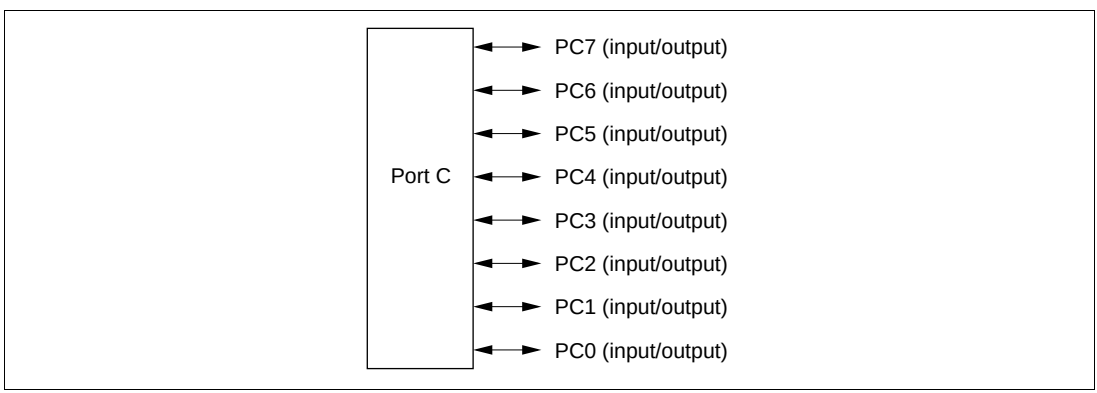


Figure 9-11 Port C Pin Functions (Mode 7)

9.9.4 MOS Input Pull-Up Function

Port C has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be used in modes 6 and 7, and can be specified as on or off for individual bits.

With the port input pin function (modes 6 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.

The MOS input pull-up function is in the off state after a power-on reset and in hardware standby mode. The previous state is retained after a manual reset and in software standby mode.

Table 9-17 summarizes the MOS input pull-up states.

Table 9-17 MOS Input Pull-Up States (Port C)

Pins	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Address output (modes 4 and 5), port output (modes 6 and 7)	OFF	OFF	OFF	OFF	OFF
Port input (modes 6 and 7)	OFF	OFF	ON/OFF	ON/OFF	ON/OFF

Legend:

OFF: MOS input pull-up is always off.

ON/OFF: On when PCDDR = 0 and PCPCR = 1; otherwise off.

9.10 Port D

9.10.1 Overview

Port D is an 8-bit I/O port. Port D pins also function as data bus input/output pins. The pin functions depend on the operating mode.

Port D has a built-in MOS input pull-up function that can be controlled by software.

Figure 9-12 shows the port D pin configuration.

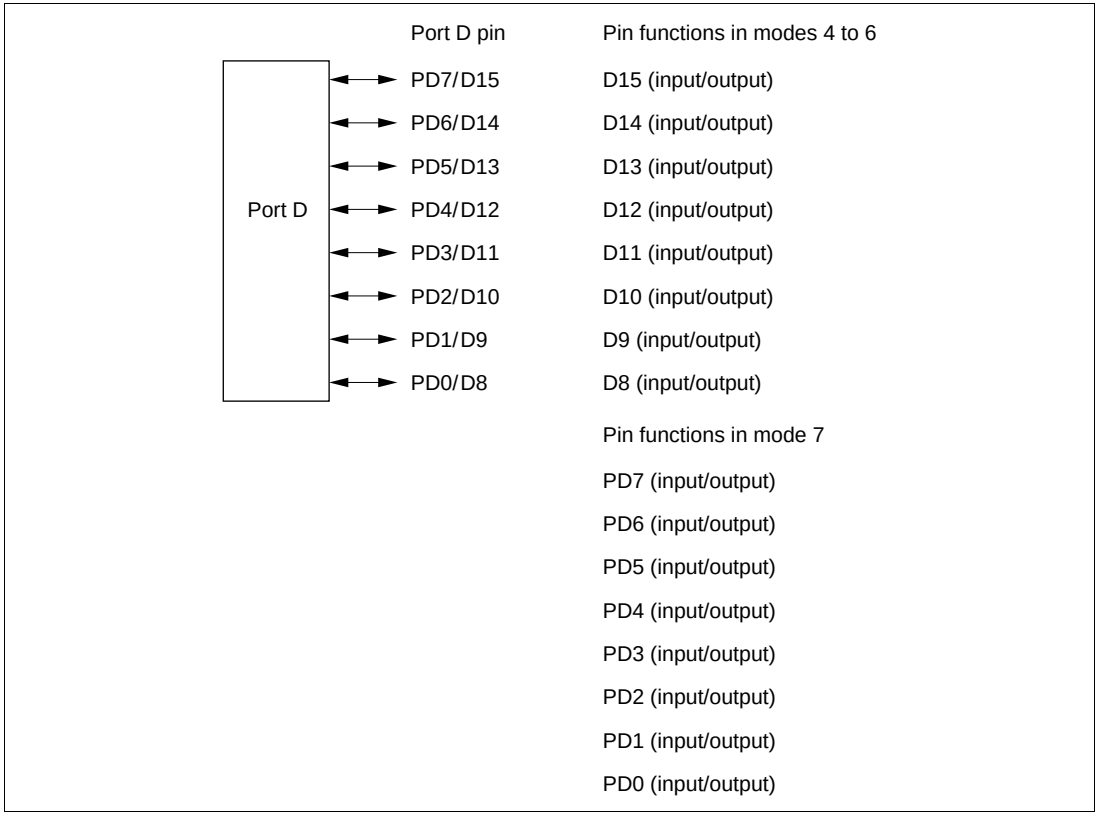


Figure 9-12 Port D Pin Functions

9.10.2 Register Configuration

Table 9-18 shows the port D register configuration.

Table 9-18 Port D Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port D data direction register	PDDDR	W	H'00	H'FE3C
Port D data register	PDDR	R/W	H'00	H'FF0C
Port D register	PORTD	R	Undefined	H'FFBC
Port D MOS pull-up control register	PDPCR	R/W	H'00	H'FE43

Note: * Lower 16 bits of the address.

(1) Port D Data Direction Register (PDDDR)

Bit	:	7	6	5	4	3	2	1	0
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PDDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port D. PDDDR cannot be read; if it is, an undefined value will be read.

PDDDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(a) Modes 4 to 6

The input/output direction settings in PDDDR are ignored, and port D pins automatically function as data input/output pins.

(b) Mode 7

Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, while clearing the bit to 0 makes the pin an input port.

(2) Port D Data Register (PDDR)

Bit	:	7	6	5	4	3	2	1	0
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PDDR is an 8-bit readable/writable register that stores output data for the port D pins (PD7 to PD0).

PDDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(3) Port D Register (PORTD)

Bit	:	7	6	5	4	3	2	1	0
		PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins PD7 to PD0.

PORTD is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port D pins (PD7 to PD0) must always be performed on PDDR.

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.

After a power-on reset and in hardware standby mode, PORTD contents are determined by the pin states, as PDDDR and PDDR are initialized. PORTD retains its previous state after a manual reset and in software standby mode.

(4) Port D MOS Pull-Up Control Register (PDPCR)

Bit	:	7	6	5	4	3	2	1	0
		PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port D on a bit-by-bit basis.

PDPCR is valid for port input pins (mode 7). When a PDDDR bit is cleared to 0 (input port setting), setting the corresponding PDPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PDPCR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

9.10.3 Pin Functions in Each Mode

(1) Modes 4 to 6

In modes 4 to 6, port D pins function as data input/output pins automatically. Port D pin functions in modes 4 to 6 are shown in figure 9-13.

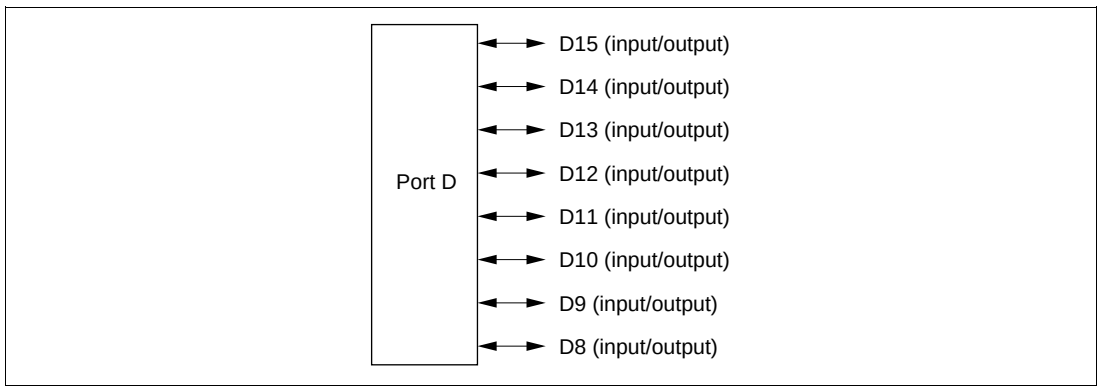


Figure 9-13 Port D Pin Functions (Modes 4 to 6)

(2) Mode 7

In mode 7, port D functions as an I/O port, and input or output can be specified bit by bit. Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, while clearing the bit to 0 makes the pin an input port.

Port D pin functions in mode 7 are shown in figure 9-14.

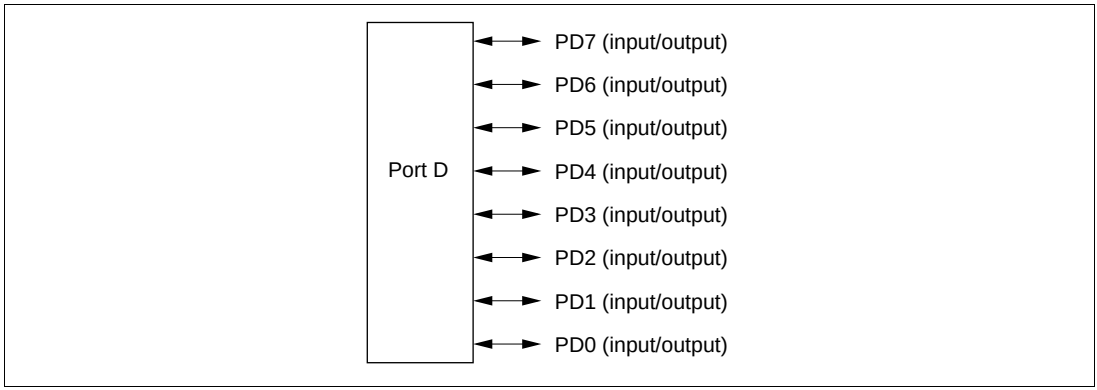


Figure 9-14 Port D Pin Functions (Mode 7)

9.10.4 MOS Input Pull-Up Function

Port D has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be used in mode 7, and can be specified as on or off for individual bits.

With the port input pin function (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.

The MOS input pull-up function is in the off state after a power-on reset and in hardware standby mode. The previous state is retained after a manual reset and in software standby mode.

Table 9-19 summarizes the MOS input pull-up states.

Table 9-19 MOS Input Pull-Up States (Port D)

Pins	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Data input/output (modes 4 to 6), port output (mode 7)	OFF	OFF	OFF	OFF	OFF
Port input (mode 7)	OFF	OFF	ON/OFF	ON/OFF	ON/OFF

Legend:
OFF: MOS input pull-up is always off.
ON/OFF: On when PDDDR = 0 and PDPCR = 1; otherwise off.

9.11 Port E

9.11.1 Overview

Port E is an 8-bit I/O port. Port E pins also function as data bus input/output pins. The pin functions depend on the operating mode and on whether 8-bit or 16-bit bus mode is used.

Port E has a built-in MOS input pull-up function that can be controlled by software.

Figure 9-15 shows the port E pin configuration.

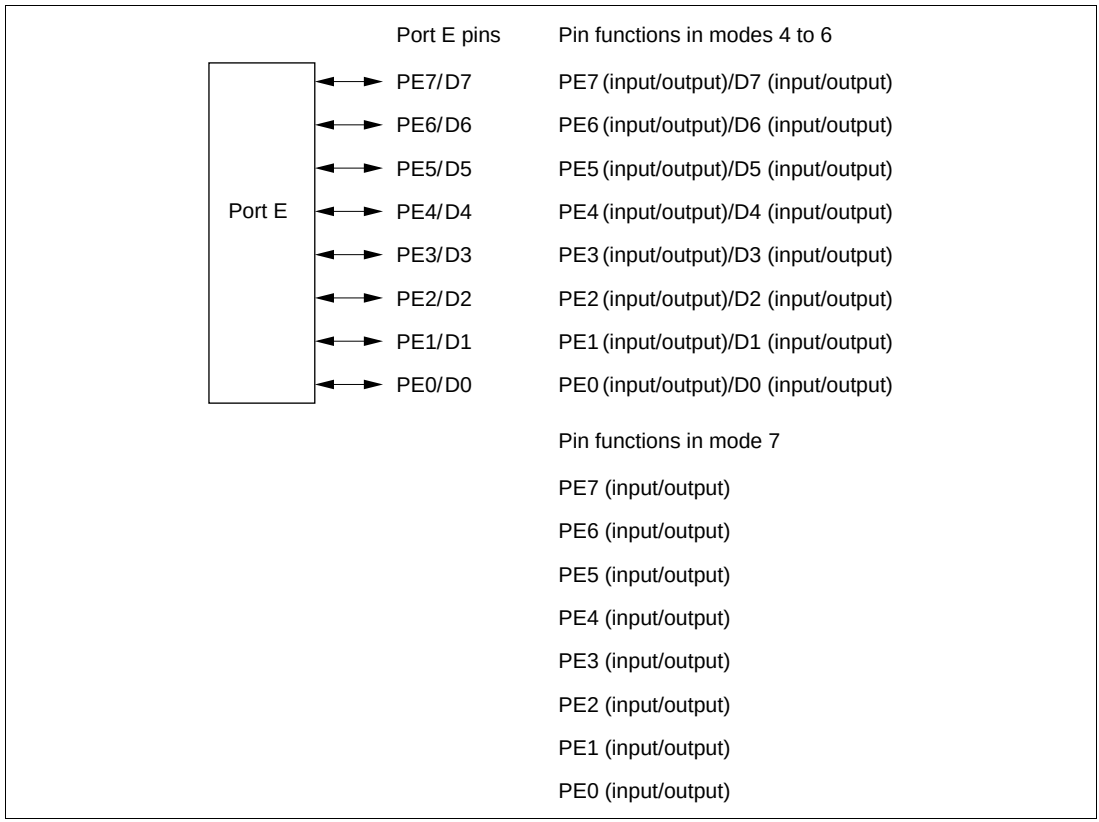


Figure 9-15 Port E Pin Functions

9.11.2 Register Configuration

Table 9-20 shows the port E register configuration.

Table 9-20 Port E Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port E data direction register	PEDDR	W	H'00	H'FE3D
Port E data register	PEDR	R/W	H'00	H'FF0D
Port E register	PORTE	R	Undefined	H'FFBD
Port E MOS pull-up control register	PEPCR	R/W	H'00	H'FE44

Note: * Lower 16 bits of the address.

(1) Port E Data Direction Register (PEDDR)

Bit	:	7	6	5	4	3	2	1	0
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PEDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port E. PEDDR cannot be read; if it is, an undefined value will be read.

PEDDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(a) Modes 4 to 6

When 8-bit bus mode is selected, port E functions as an I/O port. Setting a PEDDR 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, the input/output direction settings in PEDDR are ignored, and port E pins automatically function as data input/output pins.

For details of the 8-bit and 16-bit bus modes, see section 6, Bus Controller.

(b) Mode 7

Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

(2) Port E Data Register (PEDR)

Bit	:	7	6	5	4	3	2	1	0
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEDR is an 8-bit readable/writable register that stores output data for the port E pins (PE7 to PE0).

PEDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(3) Port E Register (PORTE)

Bit	:	7	6	5	4	3	2	1	0
		PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins PE7 to PE0.

PORTE is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port E pins (PE7 to PE0) must always be performed on PEDR.

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.

After a power-on reset and in hardware standby mode, PORTE contents are determined by the pin states, as PEDDR and PEDR are initialized. PORTE retains its previous state after a manual reset and in software standby mode.

(4) Port E MOS Pull-Up Control Register (PEPCR)

Bit	:	7	6	5	4	3	2	1	0
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port E on a bit-by-bit basis.

PEPCR is valid for port input pins (modes 4 to 6 in 8-bit bus mode, or mode 7).

When a PEDDR bit is cleared to 0 (input port setting), setting the corresponding PEPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PEPCR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

9.11.3 Pin Functions in Each Mode

(1) Modes 4 to 6

In modes 4 to 6, if 8-bit access space is designated and 8-bit bus mode is selected, port E functions as an I/O port. Setting a PEDDR 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, the input/output direction settings in PEDDR are ignored, and port E pins function as data input/output pins.

Port E pin functions in modes 4 to 6 are shown in figure 9-16.

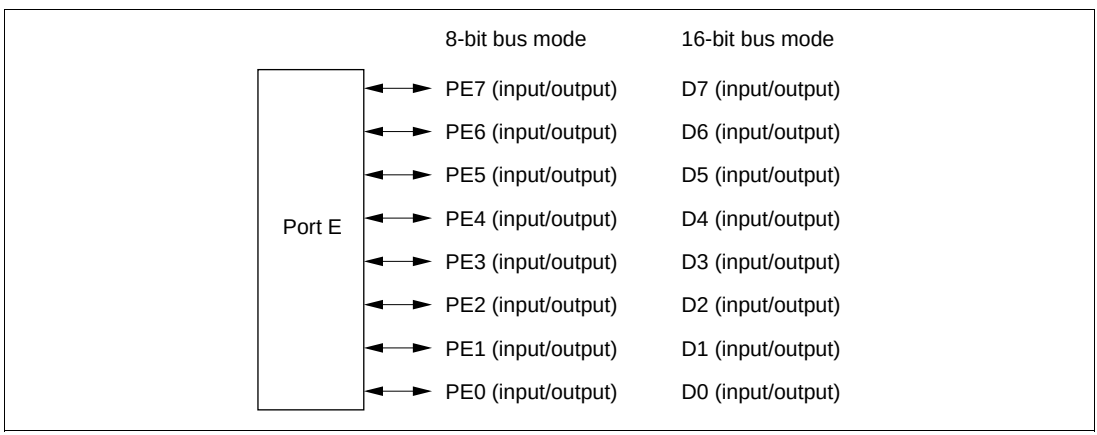


Figure 9-16 Port E Pin Functions (Modes 4 to 6)

(2) Mode 7

In mode 7, port E functions as an I/O port, and input or output can be specified bit by bit. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

Port E pin functions in mode 7 are shown in figure 9-17.

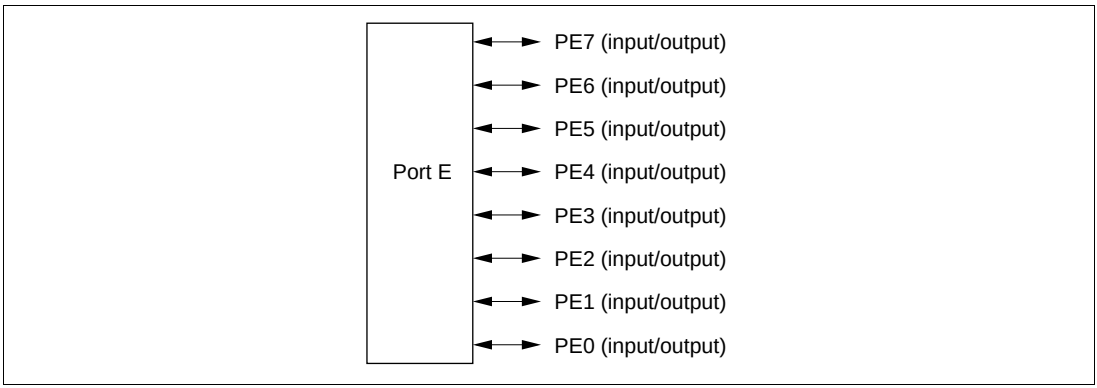


Figure 9-17 Port E Pin Functions (Mode 7)

9.11.4 MOS Input Pull-Up Function

Port E has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be used in modes 4 to 6 in 8-bit bus mode, or in mode 7, and can be specified as on or off for individual bits.

With the port input pin function (modes 4 to 6 in 8-bit bus mode, or mode 7), 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.

The MOS input pull-up function is in the off state after a power-on reset and in hardware standby mode. The previous state is retained after a manual reset and in software standby mode.

Table 9-21 summarizes the MOS input pull-up states.

Table 9-21 MOS Input Pull-Up States (Port E)

Pins	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
Data input/output (modes 4 to 6 with 16-bit bus), port output (modes 4 to 6 with 8-bit bus, mode 7)	OFF	OFF	OFF	OFF	OFF
Port input (modes 4 to 6 with 8-bit bus, mode 7)	OFF	OFF	ON/OFF	ON/OFF	ON/OFF

Legend:

OFF: MOS input pull-up is always off.

ON/OFF: On when PEDDR = 0 and PEPCR = 1; otherwise off.

9.12 Port F

9.12.1 Overview

Port F is an 8-bit I/O port. Port F pins also function as external interrupt input pins ($\overline{\text{IRQ2}}$ and $\overline{\text{IRQ3}}$), bus control signal I/O pins ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, $\overline{\text{LWR}}$, $\overline{\text{WAIT}}$, $\overline{\text{BREQ}}$, and $\overline{\text{BACK}}$), and the system clock (ø) output pin.

The interrupt input pins ($\overline{\text{IRQ2}}$ and $\overline{\text{IRQ3}}$) are Schmitt-triggered inputs.

Figure 9-18 shows the port F pin configuration.

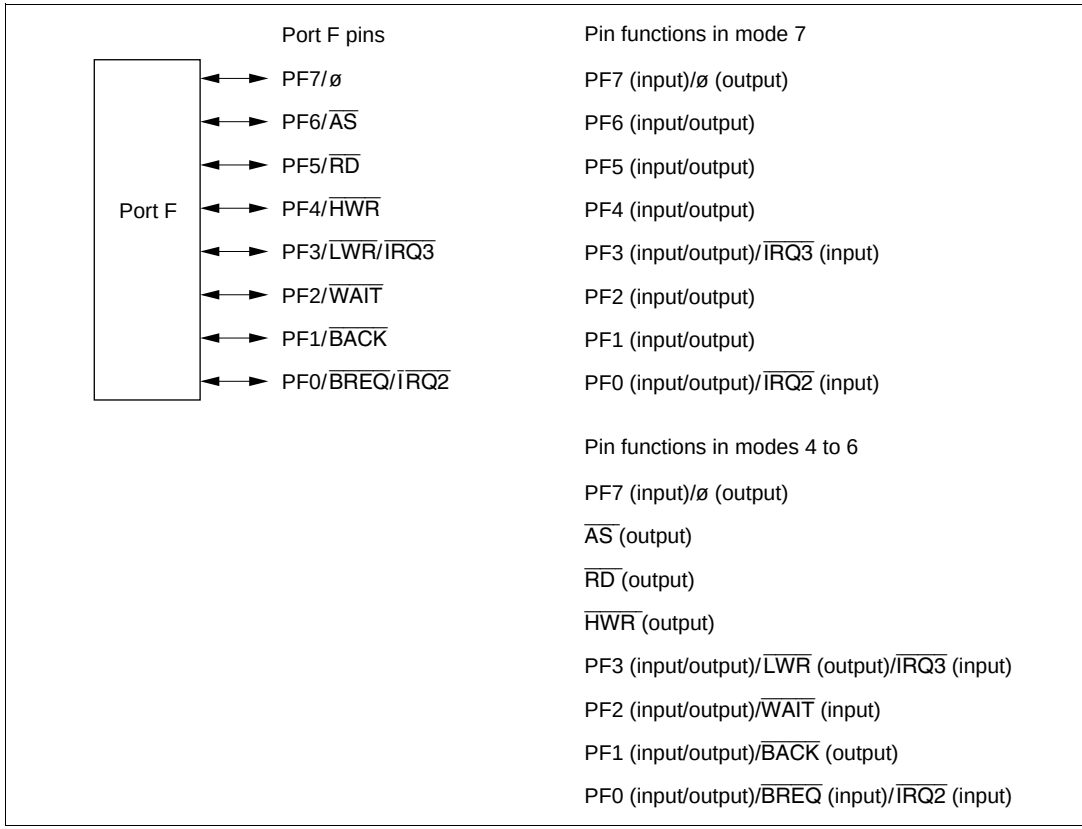


Figure 9-18 Port F Pin Functions

9.12.2 Register Configuration

Table 9-22 shows the port F register configuration.

Table 9-22 Port F Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
Port F data direction register	PFDDR	W	H'80/H'00*[2]	H'FE3E
Port F data register	PFDR	R/W	H'00	H'FF0E
Port F register	PORTF	R	Undefined	H'FFBE

Notes: *1 Lower 16 bits of the address.
*2 Initial value depends on the mode. Initialized to H'80 in modes 4 to 6, and to H'00 in mode 7.

(1) Port F Data Direction Register (PFDDR)

Bit	:	7	6	5	4	3	2	1	0
		PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR
Modes 4 to 6	:								
Initial value	:	1	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W
Mode 7	:								
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PFDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port F. PFDDR cannot be read; if it is, an undefined value will be read..

PFDDR is initialized to H'80 (modes 4 to 6) or H'00 (mode 7) by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode. The OPE bit in SBYCR is used to select whether the bus control output pins retain their output state or become high-impedance when a transition is made to software standby mode.

(a) Modes 4 to 6

Pin PF7 functions as the ø output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0.

The input/output direction specification in PFDDR is ignored for pins PF6 to PF3, which are automatically designated as bus control outputs ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$).

Pins PF2 to PF0 are made bus control input/output pins ($\overline{WAIT}$, $\overline{BACK}$, and $\overline{BREQ}$) by bus controller settings. Otherwise, setting a PFDDR bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.

(b) Mode 7

Setting a PFDDR bit to 1 makes the corresponding port F pin PF6 to PF0 an output port, or in the case of pin PF7, the ø output pin. Clearing the bit to 0 makes the pin an input port.

(2) Port F Data Register (PFDR)

Bit	:	7	6	5	4	3	2	1	0
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFDR is an 8-bit readable/writable register that stores output data for the port F pins (PF7 to PF0).

PFDR is initialized to H'00 by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(3) Port F Register (PORTF)

Bit	:	7	6	5	4	3	2	1	0
		PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins PF7 to PF0.

PORTF is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port F pins (PF7 to PF0) must always be performed on PFDR.

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.

After a power-on reset and in hardware standby mode, PORTF contents are determined by the pin states, as PFDDR and PFDR are initialized. PORTF retains its previous state after a manual reset and in software standby mode.

9.12.3 Pin Functions

Port F pins also function as external interrupt input pins ($\overline{\text{IRQ2}}$ and $\overline{\text{IRQ3}}$), bus control signal I/O pins ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, $\overline{\text{LWR}}$, $\overline{\text{WAIT}}$, $\overline{\text{BREQ}}$, and $\overline{\text{BACK}}$), and the system clock (ø) output pin. The pin functions differ between modes 4 to 6 and mode 7. Port F pin functions are shown in table 9-23.

Table 9-23 Port F Pin Functions

Pin	Pin Functions and Selection Method
PF7/ø	The pin function is switched as shown below according to bit PF7DDR.

PF7DDR	0	1
Pin function	PF7 input	ø output

PF6/$\overline{AS}$	The pin function is switched as shown below according to the operating mode and bit PF6DDR.

Operating mode	Modes 4 to 6	Mode 7	
PF6DDR	—	0	1
Pin function	$\overline{AS}$ output	PF6 input	PF6 output

PF5/$\overline{RD}$	The pin function is switched as shown below according to the operating mode and bit PF5DDR.

Operating mode	Modes 4 to 6	Mode 7	
PF5DDR	—	0	1
Pin function	$\overline{RD}$ output	PF5 input	PF5 output

PF4/$\overline{HWR}$	The pin function is switched as shown below according to the operating mode and bit PF4DDR.

Operating mode	Modes 4 to 6	Mode 7	
PF4DDR	—	0	1
Pin function	$\overline{HWR}$ output	PF4 input	PF4 output

PF3/$\overline{LWR}$/ $\overline{IRQ3}$	The pin function is switched as shown below according to the operating mode, the bus mode, and bit PF3DDR.

Operating mode	Modes 4 to 6			Mode 7	
Bus mode	16-bit bus mode	8-bit bus mode		—	
PF3DDR	—	0	1	0	1
Pin function	$\overline{LWR}$ output	PF3 input	PF3 output	PF3 input	PF3 output
		$\overline{IRQ3}$ input*			

Note: *When used as an external interrupt input pin, do not use as an I/O pin for another function.

Pin	Pin Functions and Selection Method

PF2/$\overline{\text{WAIT}}$ The pin function is switched as shown below according to the operating mode, bit WAITE, and bit PF2DDR.

Operating mode	Modes 4 to 6			Mode 7	
WAITE	0		1	—	
PF2DDR	0	1	—	0	1
Pin function	PF2 input	PF2 output	$\overline{\text{WAIT}}$ input	PF2 input	PF2 output

PF1/$\overline{\text{BACK}}$/ BUZZ The pin function is switched as shown below according to the operating mode, bit BRLE, bit BUZZE in PFCR, and bit PF1DDR.

Operating mode	Modes 4 to 6			Mode 7	
BRLE	0		1	—	
PF1DDR	0	1	—	0	1
Pin function	PF1 input	PF1 output	$\overline{\text{BACK}}$ output	PF1 input	PF1 output

PF0/$\overline{\text{BREQ}}$/ $\overline{\text{IRQ2}}$ The pin function is switched as shown below according to the operating mode, bit BRLE, and bit PF0DDR.

Operating mode	Modes 4 to 6			Mode 7	
BRLE	0		1	—	
PF0DDR	0	1	—	0	1
Pin function	PF0 input	PF0 output	$\overline{\text{BREQ}}$ input	PF0 input	PF0 output
	$\overline{\text{IRQ2}}$ input*				

Note: * When used as an external interrupt input pin, do not use as an I/O pin for another function.

9.13 Port G

9.13.1 Overview

Port G is a 5-bit I/O port. Port G pins also function as external interrupt input pins ($\overline{IRQ6}$ and $\overline{IRQ7}$) and bus control signal output pins ($\overline{CS0}$ to $\overline{CS3}$).

The interrupt input pins ($\overline{IRQ6}$ and $\overline{IRQ7}$) are Schmitt-triggered inputs.

Figure 9-19 shows the port G pin configuration.

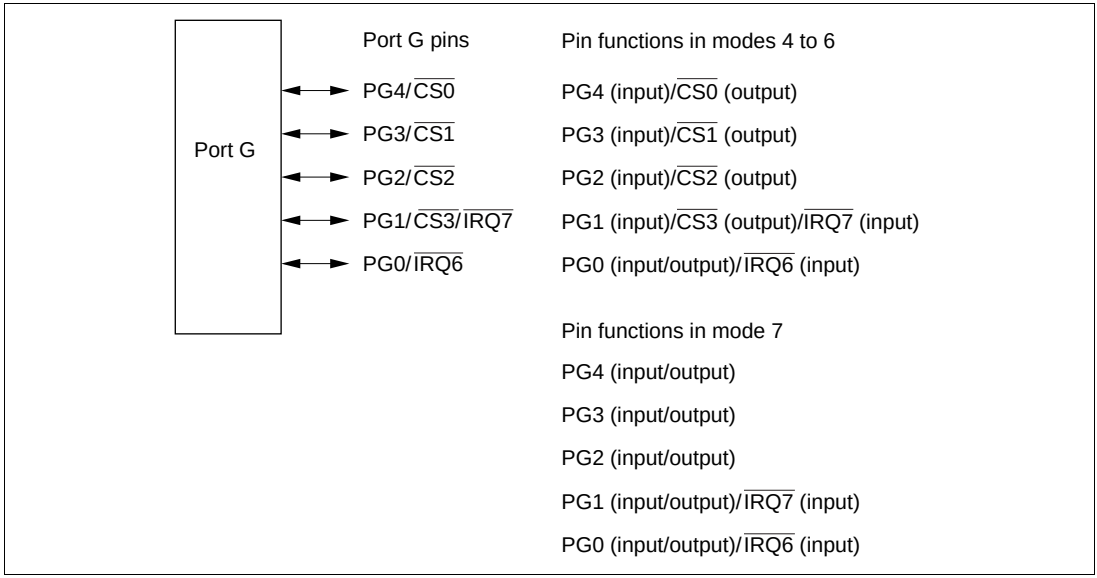


Figure 9-19 Port G Pin Functions

9.13.2 Register Configuration

Table 9-24 shows the port G register configuration.

Table 9-24 Port G Registers

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port G data direction register	PGDDR	W	H'10/H'00*3	H'FE3F
Port G data register	PGDR	R/W	H'00	H'FF0F
Port G register	PORTG	R	Undefined	H'FFBF

Notes: *1 Lower 16 bits of the address.
*2 Value of bits 4 to 0.
*3 Initial value depends on the mode. Initialized to H'10 in modes 4 and 5, and to H'00 in modes 6 and 7.

(1) Port G Data Direction Register (PGDDR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR
Modes 4 and 5	:								
Initial value	:	Undefined	Undefined	Undefined	1	0	0	0	0
R/W	:	—	—	—	W	W	W	W	W
Modes 6 and 7	:								
Initial value	:	Undefined	Undefined	Undefined	0	0	0	0	0
R/W	:	—	—	—	W	W	W	W	W

PGDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port G. PGDDR cannot be read. Also, bits 7 to 5 are reserved, and will return an undefined value if read.

Bit PG4DDR is initialized to 1 (modes 4 and 5) or 0 (modes 6 and 7) by a power-on reset and in hardware standby mode. PGDDR retains its previous state after a manual reset and in software standby mode. The OPE bit in SBYCR is used to select whether the bus control output pins retain their output state or become high-impedance when a transition is made to software standby mode.

(a) Modes 4 to 6

Pins PG4 to PG1 function as bus control signal output pins ($\overline{CS0}$ to $\overline{CS3}$) when the corresponding PGDDR bits are set to 1, and as input ports when the bits are cleared to 0.

Pin PG0 functions as an output port when the corresponding PGDDR bit is set to 1, and as an input port when the bit is cleared to 0.

(b) Mode 7

Setting a PGDDR bit to 1 makes the corresponding pin an output port, while clearing the bit to 0 makes the pin an input port.

(2) Port G Data Register (PGDR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
Initial value	:	Undefined	Undefined	Undefined	0	0	0	0	0
R/W	:	—	—	—	R/W	R/W	R/W	R/W	R/W

PGDR is an 8-bit readable/writable register that stores output data for the port G pins (PG4 to PG0).

Bits 7 to 5 are reserved; these bits cannot be modified and will return an undefined value if read.

PGDR is initialized to H'00 (bits 4 to 0) by a power-on reset and in hardware standby mode. It retains its previous state after a manual reset and in software standby mode.

(3) Port G Register (PORTG)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4	PG3	PG2	PG1	PG0
Initial value	:	Undefined	Undefined	Undefined	—*	—*	—*	—*	—*
R/W	:	—	—	—	R	R	R	R	R

Note: * Determined by the state of pins PG4 to PG0.

PORTG is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port G pins (PG4 to PG0) must always be performed on PGDR.

Bits 7 to 5 are reserved; these bits cannot be modified and will return an undefined value if read.

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.

After a power-on reset and in hardware standby mode, PORTG contents are determined by the pin states, as PGDDR and PGDR are initialized. PORTG retains its previous state after a manual reset and in software standby mode.

9.13.3 Pin Functions

Port G pins also function as external interrupt input pins ($\overline{\text{IRQ6}}$ and $\overline{\text{IRQ7}}$) and bus control signal output pins ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$). The pin functions differ between modes 4 to 6 and mode 7. Port G pin functions are shown in table 9-25.

Table 9-25 Port G Pin Functions

Pin — **Pin Functions and Selection Method**

PG4/$\overline{\text{CS0}}$ — The pin function is switched as shown below according to the operating mode and bit PG4DDR.

Operating mode	Modes 4 to 6		Mode 7	
PG4DDR	0	1	0	1
Pin function	PG4 input	$\overline{\text{CS0}}$ output	PG4 input	PG4 output

PG3/$\overline{\text{CS1}}$ — The pin function is switched as shown below according to the operating mode and bit PG3DDR.

Operating mode	Modes 4 to 6		Mode 7	
PG3DDR	0	1	0	1
Pin function	PG3 input	$\overline{\text{CS1}}$ output	PG3 input	PG3 output

PG2/$\overline{\text{CS2}}$ — The pin function is switched as shown below according to the operating mode and bit PG2DDR.

Operating mode	Modes 4 to 6		Mode 7	
PG2DDR	0	1	0	1
Pin function	PG2 input	$\overline{\text{CS2}}$ output	PG2 input	PG2 output

PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$ — The pin function is switched as shown below according to the operating mode and bit PG1DDR.

Operating mode	Modes 4 to 6		Mode 7	
PG1DDR	0	1	0	1
Pin function	PG1 input	$\overline{\text{CS3}}$ output	PG1 input	PG1 output
	$\overline{\text{IRQ7}}$ input*			

Note: * When used as an external interrupt input pin, do not use as an I/O pin for another function.

Pin	Pin Functions and Selection Method
PG0/$\overline{\text{IRQ6}}$	The pin function is switched as shown below according to bit PG0DDR.

PG0DDR	0	1
Pin function	PG0 input	PG0 output
	$\overline{\text{IRQ6}}$ input*	

Note: * When used as an external interrupt input pin, do not use as an I/O pin for another function.

Section 10 16-Bit Timer Pulse Unit (TPU)

10.1 Overview

The H8S/2214 has an on-chip 16-bit timer pulse unit (TPU) comprising three 16-bit timer channels.

10.1.1 Features

- Can input/output a maximum of 8 pulses
  - A total of 8 timer general registers (TGRs) are provided (four each for channel 0, and two each for channels 1and 2), each of which can be set independently as an output compare/input capture register
  - TGRC and TGRD for channel 0 can also be used as buffer registers
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output at compare match: Selection of 0, 1, or toggle output
  - Input capture function: Selection of rising edge, falling edge, or both edge detection
  - Counter clear operation: Counter clearing possible by compare match or input capture
  - Synchronous operation: 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
  - PWM mode: Any PWM output duty can be set
    Maximum of 7-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channel 0
  - Input capture register double-buffering possible
  - Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 1and 2
  - Two-phase encoder pulse up/down-count possible
- SCI0 baud rate clock generation by channels 1 and 2
  - An SCI0 baud rate clock can be generated using an AND circuit for TIOCA1 output and TIOCA2 output
- Fast access via internal 16-bit bus
  - Fast access is possible via a 16-bit bus interface

- 13 interrupt sources
  - For channel 0, four compare match/input capture dual-function interrupts and one overflow interrupt can be requested independently
  - For channels 1 and 2 two compare match/input capture dual-function interrupts, one overflow interrupt, and one underflow interrupt can be requested independently
- Automatic transfer of register data
  - Block transfer, 1-word data transfer, and 1-byte data transfer possible by data transfer controller (DTC) and DMA controller (DMAC) activation
- Module stop mode can be set
  - As the initial setting, TPU operation is halted. Register access is enabled by exiting module stop mode.

Table 10-1 lists the functions of the TPU.

Table 10-1 TPU Functions

Item		Channel 0	Channel 1	Channel 2
Count clock		ø/1 ø/4 ø/16 ø/64 TCLKA TCLKB TCLKC TCLKD	ø/1 ø/4 ø/16 ø/64 ø/256 TCLKA TCLKB	ø/1 ø/4 ø/16 ø/64 ø/1024 TCLKA TCLKB TCLKC
General registers		TGR0A TGR0B	TGR1A TGR1B	TGR2A TGR2B
General registers/ buffer registers		TGR0C TGR0D	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	○	○	○
	1 output	○	○	○
	Toggle output	○	○	○
Input capture function		○	○	○
Synchronous operation		○	○	○
PWM mode		○	○	○
Phase counting mode		—	○	○
Buffer operation		○	—	—
DTC activation		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
DMAC activation		TGR0A compare match or input capture	TGR1A compare match or input capture	TGR2A compare match or input capture

Legend

○ : Possible

— : Not possible

Item	Channel 0	Channel 1	Channel 2
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow

10.1.2 Block Diagram

Figure 10-1 shows a block diagram of the TPU.

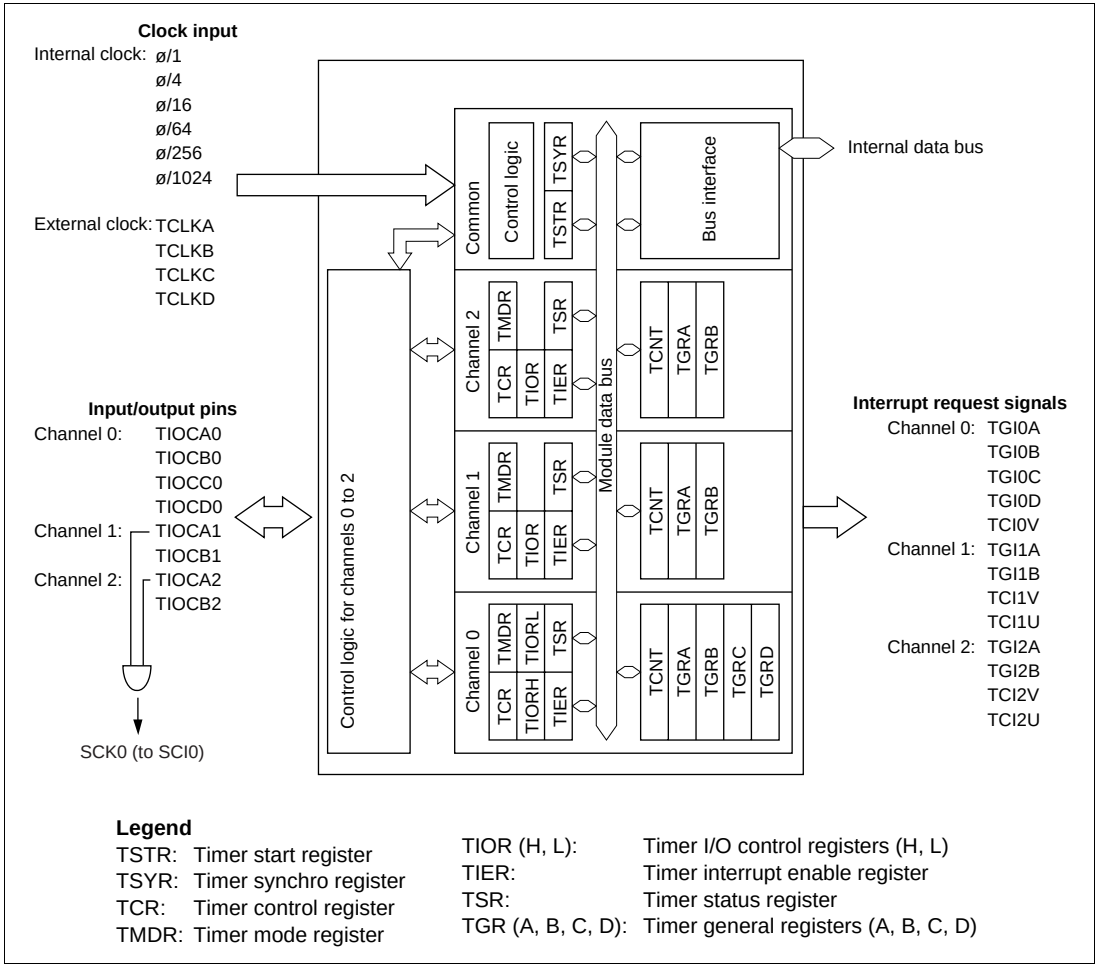


Figure 10-1 Block Diagram of H8S/2214 TPU

10.1.3 Pin Configuration

Table 10-2 summarizes the TPU pins.

Table 10-2 TPU Pins

Channel	Name	Symbol	I/O	Function
All	Clock input A	TCLKA	Input	External clock A input pin (Channel 1 phase counting mode A phase input)
	Clock input B	TCLKB	Input	External clock B input pin (Channel 1 phase counting mode B phase input)
	Clock input C	TCLKC	Input	External clock C input pin (Channel 2 phase counting mode A phase input)
	Clock input D	TCLKD	Input	External clock D input pin (Channel 2 phase counting mode B phase input)
0	Input capture/out compare match A0	TIOCA0	I/O	TGR0A input capture input/output compare output/PWM output pin
	Input capture/out compare match B0	TIOCB0	I/O	TGR0B input capture input/output compare output/PWM output pin
	Input capture/out compare match C0	TIOCC0	I/O	TGR0C input capture input/output compare output/PWM output pin
	Input capture/out compare match D0	TIOCD0	I/O	TGR0D input capture input/output compare output/PWM output pin
1	Input capture/out compare match A1	TIOCA1	I/O	TGR1A input capture input/output compare output/PWM output pin
	Input capture/out compare match B1	TIOCB1	I/O	TGR1B input capture input/output compare output/PWM output pin
2	Input capture/out compare match A2	TIOCA2	I/O	TGR2A input capture input/output compare output/PWM output pin
	Input capture/out compare match B2	TIOCB2	I/O	TGR2B input capture input/output compare output/PWM output pin

10.1.4 Register Configuration

Table 10-3 summarizes the TPU registers.

Table 10-3 TPU Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
0	Timer control register 0	TCR0	R/W	H'00	H'FF10
	Timer mode register 0	TMDR0	R/W	H'C0	H'FF11
	Timer I/O control register 0H	TIOR0H	R/W	H'00	H'FF12
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H'FF13
	Timer interrupt enable register 0	TIER0	R/W	H'40	H'FF14
	Timer status register 0	TSR0	R/(W)*[2]	H'C0	H'FF15
	Timer counter 0	TCNT0	R/W	H'0000	H'FF16
	Timer general register 0A	TGR0A	R/W	H'FFFF	H'FF18
	Timer general register 0B	TGR0B	R/W	H'FFFF	H'FF1A
	Timer general register 0C	TGR0C	R/W	H'FFFF	H'FF1C
	Timer general register 0D	TGR0D	R/W	H'FFFF	H'FF1E
1	Timer control register 1	TCR1	R/W	H'00	H'FF20
	Timer mode register 1	TMDR1	R/W	H'C0	H'FF21
	Timer I/O control register 1	TIOR1	R/W	H'00	H'FF22
	Timer interrupt enable register 1	TIER1	R/W	H'40	H'FF24
	Timer status register 1	TSR1	R/(W)*[2]	H'C0	H'FF25
	Timer counter 1	TCNT1	R/W	H'0000	H'FF26
	Timer general register 1A	TGR1A	R/W	H'FFFF	H'FF28
	Timer general register 1B	TGR1B	R/W	H'FFFF	H'FF2A
2	Timer control register 2	TCR2	R/W	H'00	H'FF30
	Timer mode register 2	TMDR2	R/W	H'C0	H'FF31
	Timer I/O control register 2	TIOR2	R/W	H'00	H'FF32
	Timer interrupt enable register 2	TIER2	R/W	H'40	H'FF34
	Timer status register 2	TSR2	R/(W)*[2]	H'C0	H'FF35
	Timer counter 2	TCNT2	R/W	H'0000	H'FF36
	Timer general register 2A	TGR2A	R/W	H'FFFF	H'FF38
	Timer general register 2B	TGR2B	R/W	H'FFFF	H'FF3A

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
All	Timer start register	TSTR	R/W	H'00	H'FEB0
	Timer synchro register	TSYR	R/W	H'00	H'FEB1
	Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address.
*2 Can only be written with 0 for flag clearing.

10.2 Register Descriptions

10.2.1 Timer Control Register (TCR)

Channel 0: TCR0

Bit	:	7	6	5	4	3	2	1	0
		CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 1: TCR1
Channel 2: TCR2

Bit	:	7	6	5	4	3	2	1	0
		—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The TCR registers are 8-bit registers that control the TCNT channels. The TPU has three TCR registers, one for each of channels 0 to 2. The TCR registers are initialized to H'00 by a reset, and in hardware standby mode.

Bits 7 to 5—Counter Clear 2 to 0 (CCLR2 to CCLR0): These bits select the TCNT counter clearing source.

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture*2
		1	0	TCNT cleared by TGRD compare match/input capture*2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operatio*1

Channel	Bit 7 Reserved*3	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*1

Notes: *1 Synchronous operation 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.

*3 Bit 7 is reserved in channels 1 and 2. It is always read as 0 and cannot be modified.

Bits 4 and 3—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. ø/4 both edges = ø/2 rising edge). If phase counting mode is used on channels 1 and 2, this setting is ignored and the phase counting mode setting has priority.

Bit 4	Bit 3		
CKEG1	**CKEG0**	**Description**	
0	0	Count at rising edge	(Initial value)
	1	Count at falling edge	
1	—	Count at both edges	

Note: Internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Bits 2 to 0—Time Prescaler 2 to 0 (TPSC2 to TPSC0): These bits select the TCNT counter clock. The clock source can be selected independently for each channel. Table 10-4 shows the clock sources that can be set for each channel.

Table 10-4 TPU Clock Sources

Channel	Internal Clock							External Clock			
Channel	**ø/1**	**ø/4**	**ø/16**	**ø/64**	**ø/256**	**ø/1024**	**ø/4096**	**TCLKA**	**TCLKB**	**TCLKC**	**TCLKD**
0	○	○	○	○				○	○	○	○
1	○	○	○	○	○			○	○		
2	○	○	○	○		○		○	○	○	

Legend

○ : Setting

Blank : No setting

	Bit 2	Bit 1	Bit 0		
Channel	**TPSC2**	**TPSC1**	**TPSC0**	**Description**	
0	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/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	

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
1	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/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 ø/256	
			1	Setting prohibited	

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 ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/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 ø/1024	

Note: This setting is ignored when channel 2 is in phase counting mode.

10.2.2 Timer Mode Register (TMDR)

Channel 0: TMDR0

Bit	:	7	6	5	4	3	2	1	0
		—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Channel 1: TMDR1

Channel 2: TMDR2

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode for each channel. The TPU has three TMDR registers, one for each channel. The TMDR registers are initialized to H'C0 by a reset, and in hardware standby mode.

Bits 7 and 6—Reserved: Read-only bits, always read as 1.

Bit 5—Buffer Operation B (BFB): 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 and 2 which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5

BFB	Description	
0	TGRB operates normally	(Initial value)
1	TGRB and TGRD used together for buffer operation	

Bit 4—Buffer Operation A (BFA): 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 and 2 which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified.

Bit 4

BFA	Description	
0	TGRA operates normally	(Initial value)
1	TGRA and TGRC used together for buffer operation	

Bits 3 to 0—Modes 3 to 0 (MD3 to MD0): These bits are used to set the timer operating mode.

Bit 3	Bit 2	Bit 1	Bit 0		
MD3*[1]	MD2*[2]	MD1	MD0	Description	
0	0	0	0	Normal operation	(Initial value)
			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	*	*	*	—	

*: Don't care

Notes: *1 MD3 is a reserved bit. In a write, it should always be written with 0.

*2 Phase counting mode cannot be set for channel 0. In this case, 0 should always be written to MD2.

10.2.3 Timer I/O Control Register (TIOR)

Channel 0: TIOR0H

Channel 1: TIOR1

Channel 2: TIOR2

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 0: TIOR0L

Bit	:	7	6	5	4	3	2	1	0
		IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

The TIOR registers are 8-bit registers that control the TGR registers. The TPU has four TIOR registers, two each for channel 0, and one each for channels 1 and 2. The TIOR registers are initialized to H'00 by a reset, and in hardware standby mode.

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.

Bits 7 to 4— I/O Control B3 to B0 (IOB3 to IOB0)
I/O Control D3 to D0 (IOD3 to IOD0):

Bits IOB3 to IOB0 specify the function of TGRB.
Bits IOD3 to IOD0 specify the function of TGRD.

	Bit 7	Bit 6	Bit 5	Bit 4			
Channel	**IOB3**	**IOB2**	**IOB1**	**IOB0**	**Description**		
0	0	0	0	0	TGR0B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Setting prohibited	

*: Don't care

Channel	Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	Description		
0	0	0	0	0	TGR0D is output compare register*1	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0D is input capture register*1	Capture input source is TIOCD0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Setting prohibited	

*: Don't care

Note: *1 When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
1	0	0	0	0	TGR1B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Setting prohibited	

*: Don't care

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
2	0	0	0	0	TGR2B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don’t care

Bits 3 to 0— I/O Control A3 to A0 (IOA3 to IOA0)
I/O Control C3 to C0 (IOC3 to IOC0):

IOA3 to IOA0 specify the function of TGRA.
IOC3 to IOC0 specify the function of TGRC.

	Bit 3	Bit 2	Bit 1	Bit 0			
Channel	**IOA3**	**IOA2**	**IOA1**	**IOA0**	**Description**		
0	0	0	0	0	TGR0A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Setting prohibited	

*: Don’t care

Channel	Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	Description		
0	0	0	0	0	TGR0C is output compare register*1	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0C is input capture register*1	Capture input source is TIOCC0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Setting prohibited	

*: Don't care

Note: *1 When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
1	0	0	0	0	TGR1A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Setting prohibited	

*: Don’t care

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
2	0	0	0	0	TGR2A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don’t care

10.2.4 Timer Interrupt Enable Register (TIER)

Channel 0: TIER0

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
R/W	:	R/W	—	—	R/W	R/W	R/W	R/W	R/W

Channel 1: TIER1

Channel 2: TIER2

Bit	:	7	6	5	4	3	2	1	0
		—	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	—	—	R/W	R/W

The TIER registers are 8-bit registers that control enabling or disabling of interrupt requests for each channel. The TPU has three TIER registers, one for each channel. The TIER registers are initialized to H'40 by a reset, and in hardware standby mode.

Bit 7—Reserved: Only 0 should be written to this bit.

Bit 6—Reserved: Read-only bit, always read as 1.

Bit 5—Underflow Interrupt Enable (TCIEU): Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1 and 2.

In channel 0, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5

TCIEU	Description	
0	Interrupt requests (TCIU) by TCFU disabled	(Initial value)
1	Interrupt requests (TCIU) by TCFU enabled	

Bit 4—Overflow Interrupt Enable (TCIEV): Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1.

Bit 4

TCIEV	Description	
0	Interrupt requests (TCIV) by TCFV disabled	(Initial value)
1	Interrupt requests (TCIV) by TCFV enabled	

Bit 3—TGR Interrupt Enable D (TGIED): Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channel 0.

In channels 1 and 2, bit 3 is reserved. It is always read as 0 and cannot be modified.

Bit 3

TGIED	Description	
0	Interrupt requests (TGID) by TGFD bit disabled	(Initial value)
1	Interrupt requests (TGID) by TGFD bit enabled	

Bit 2—TGR Interrupt Enable C (TGIEC): Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channel 0.

In channels 1 and 2, bit 2 is reserved. It is always read as 0 and cannot be modified.

Bit 2

TGIEC	Description	
0	Interrupt requests (TGIC) by TGFC bit disabled	(Initial value)
1	Interrupt requests (TGIC) by TGFC bit enabled	

Bit 1—TGR Interrupt Enable B (TGIEB): Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1.

Bit 1

TGIEB	Description	
0	Interrupt requests (TGIB) by TGFB bit disabled	(Initial value)
1	Interrupt requests (TGIB) by TGFB bit enabled	

Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1.

Bit 0

TGIEA	Description	
0	Interrupt requests (TGIA) by TGFA bit disabled	(Initial value)
1	Interrupt requests (TGIA) by TGFA bit enabled	

10.2.5 Timer Status Register (TSR)

Channel 0: TSR0

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Channel 1: TSR1

Channel 2: TSR2

Bit	:	7	6	5	4	3	2	1	0
		TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Note: * Can only be written with 0 for flag clearing.

The TSR registers are 8-bit registers that indicate the status of each channel. The TPU has three TSR registers, one for each channel. The TSR registers are initialized to H'C0 by a reset, and in hardware standby mode.

Bit 7—Count Direction Flag (TCFD): Status flag that shows the direction in which TCNT counts in channels 1 and 2.

In channel 0, bit 7 is reserved. It is always read as 1 and cannot be modified.

Bit 7

TCFD	Description	
0	TCNT counts down	
1	TCNT counts up	(Initial value)

Bit 6—Reserved: Read-only bit, always read as 1 and cannot be modified.

Bit 5—Underflow Flag (TCFU): Status flag that indicates that TCNT underflow has occurred when channels 1and 2 are set to phase counting mode.

In channel 0, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5

TCFU	Description	
0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1	(Initial value)
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)	

Bit 4—Overflow Flag (TCFV): Status flag that indicates that TCNT overflow has occurred.

Bit 4

TCFV	Description	
0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1	(Initial value)
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)	

Bit 3—Input Capture/Output Compare Flag D (TGFD): Status flag that indicates the occurrence of TGRD input capture or compare match in channel 0.

In channels 1 and 2, bit 3 is reserved. It is always read as 0 and cannot be modified.

Bit 3

TGFD	Description	
0	[Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1	(Initial value)
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register	

Bit 2—Input Capture/Output Compare Flag C (TGFC): Status flag that indicates the occurrence of TGRC input capture or compare match in channel 0.

In channels 1and 2, bit 2 is reserved. It is always read as 0 and cannot be modified.

Bit 2

TGFC	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIC interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register

Bit 1—Input Capture/Output Compare Flag B (TGFB): Status flag that indicates the occurrence of TGRB input capture or compare match.

Bit 1

TGFB	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Bit 0—Input Capture/Output Compare Flag A (TGFA): Status flag that indicates the occurrence of TGRA input capture or compare match.

Bit 0

TGFA	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

10.2.6 Timer Counter (TCNT)

Channel 0: TCNT0 (up-counter)

Channel 1: TCNT1 (up/down-counter*)

Channel 2: TCNT2 (up/down-counter*)

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

Note : * These counters can be used as up/down-counters only in phase counting mode. In other cases they function as up-counters.

The TCNT registers are 16-bit counters. The TPU has three TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset, and in hardware standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

10.2.7 Timer General Register (TGR)

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

The TGR registers are 16-bit registers with a dual function as output compare and input capture registers. The TPU has eight TGR registers, four each for channel 0 and two each for channels 1 and 2. TGRC and TGRD for channel 0 can also be designated for operation as buffer registers*. The TGR registers are initialized to H'FFFF by a reset, and in hardware standby mode.

The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

Note: * TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

10.2.8 Timer Start Register (TSTR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	CST2	CST1	CST0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	—	R/W	R/W	R/W

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 2. TSTR is initialized to H'00 by a reset, and in hardware standby mode.

TCNT counter operation must be halted before setting the operating mode in TMDR, or setting the TCNT count clock in TCR.

Bits 7 to 3—Reserved: Should always be written with 0.

Bits 2 to 0—Counter Start 2 to 0 (CST2 to CST0): These bits select operation or stoppage for TCNT.

Bit n

CSTn	Description	
0	TCNTn count operation is stopped	(Initial value)
1	TCNTn performs count operation	

(n = 2 to 0)

Note: 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.

10.2.9 Timer Synchro Register (TSYR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	SYNC2	SYNC1	SYNC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	—	R/W	R/W	R/W

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channels 0 to 2 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

TSYR is initialized to H'00 by a reset, and in hardware standby mode.

Bits 7 to 3—Reserved: Should always be written with 0.

Bits 2 to 0—Timer Synchro 2 to 0 (SYNC2 to SYNC0): These bits select whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, synchronous presetting of multiple channels*[1], and synchronous clearing through counter clearing on another channel*[2] are possible.

Notes: *1 To set synchronous operation, the SYNC bits for at least two channels must be set to 1.

*2 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.

Bit n

SYNCn	Description
0	TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels) (Initial value)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

(n = 2 to 0)

10.2.10 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRA is a 16-bit readable/writable register that performs module stop mode control.

When the MSTPA5 bit in MSTPCR is set to 1, TPU operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 20.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 5—Module Stop (MSTPA5): Specifies the TPU module stop mode.

Bit 5

MSTPA5	Description
0	TPU module stop mode cleared
1	TPU module stop mode set (Initial value)

10.3 Interface to Bus Master

10.3.1 16-Bit Registers

TCNT and TGR are 16-bit registers. As the data bus to the bus master is 16 bits wide, these registers can be read and written to in 16-bit units.

These registers cannot be read or written to in 8-bit units; 16-bit access must always be used.

An example of 16-bit register access operation is shown in figure 10-2.

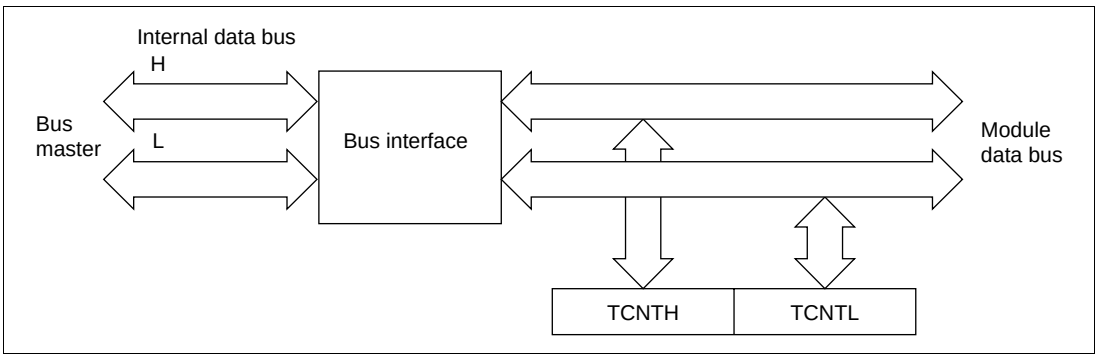


Figure 10-2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)]

10.3.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits wide, these registers can be read and written to in 16-bit units. They can also be read and written to in 8-bit units.

Examples of 8-bit register access operation are shown in figures 10-3, 10-4, and 10-5.

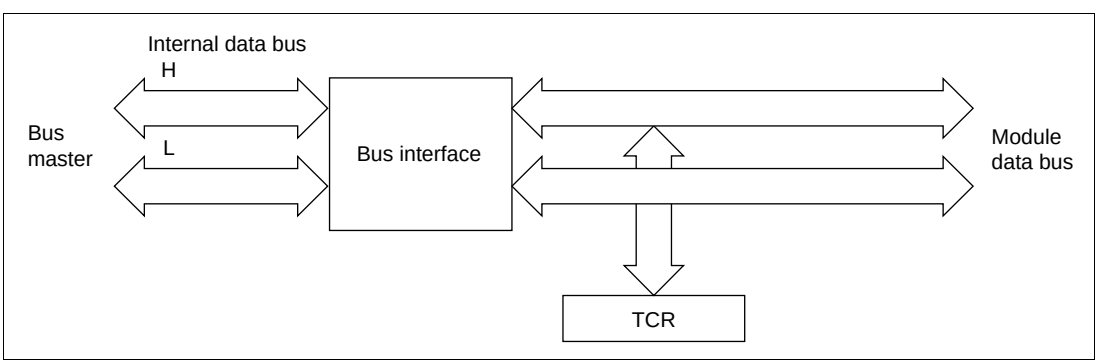


Figure 10-3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)]

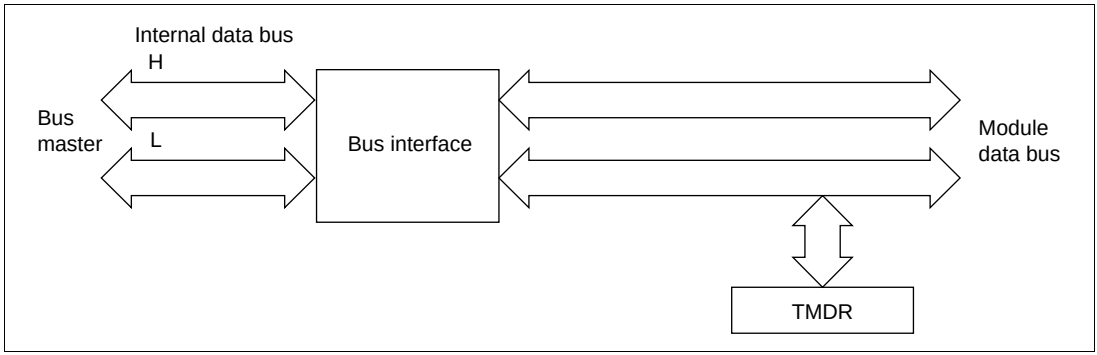


Figure 10-4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)]

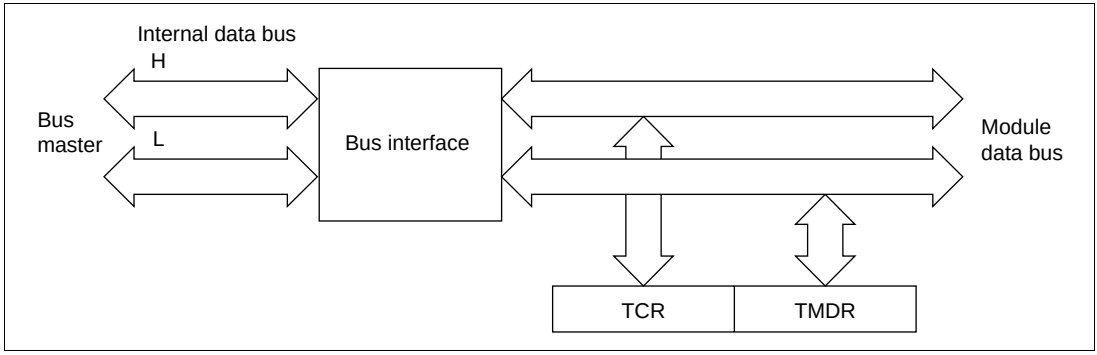


Figure 10-5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)]

10.4 Operation

10.4.1 Overview

Operation in each mode is outlined below.

Normal Operation: Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Synchronous Operation: When synchronous operation is designated for a channel, TCNT for that channel performs synchronous presetting. That is, when TCNT for a channel designated for synchronous operation is rewritten, the TCNT counters for the other channels are also rewritten at the same time. Synchronous clearing of the TCNT counters is also possible by setting the timer synchronization bits in TSYR for channels designated for synchronous operation.

Buffer Operation

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the relevant channel is transferred to TGR.

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transfer to TGR and the value previously held in TGR is transferred to the buffer register.

PWM Mode: In this mode, a PWM waveform is output. The output level can be set by means of TIOR. A PWM waveform with a duty of between 0% and 100% can be output, according to the setting of each TGR register.

Phase Counting Mode: In this mode, TCNT is incremented or decremented by detecting the phases of two clocks input from the external clock input pins in channels 1 and 2. When phase counting mode is set, the corresponding TCLK pin functions as the clock pin, and TCNT performs up- or down-counting.

This can be used for two-phase encoder pulse input.

10.4.2 Basic Functions

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.

- Example of count operation setting procedure
  Figure 10-6 shows an example of the count operation setting procedure.

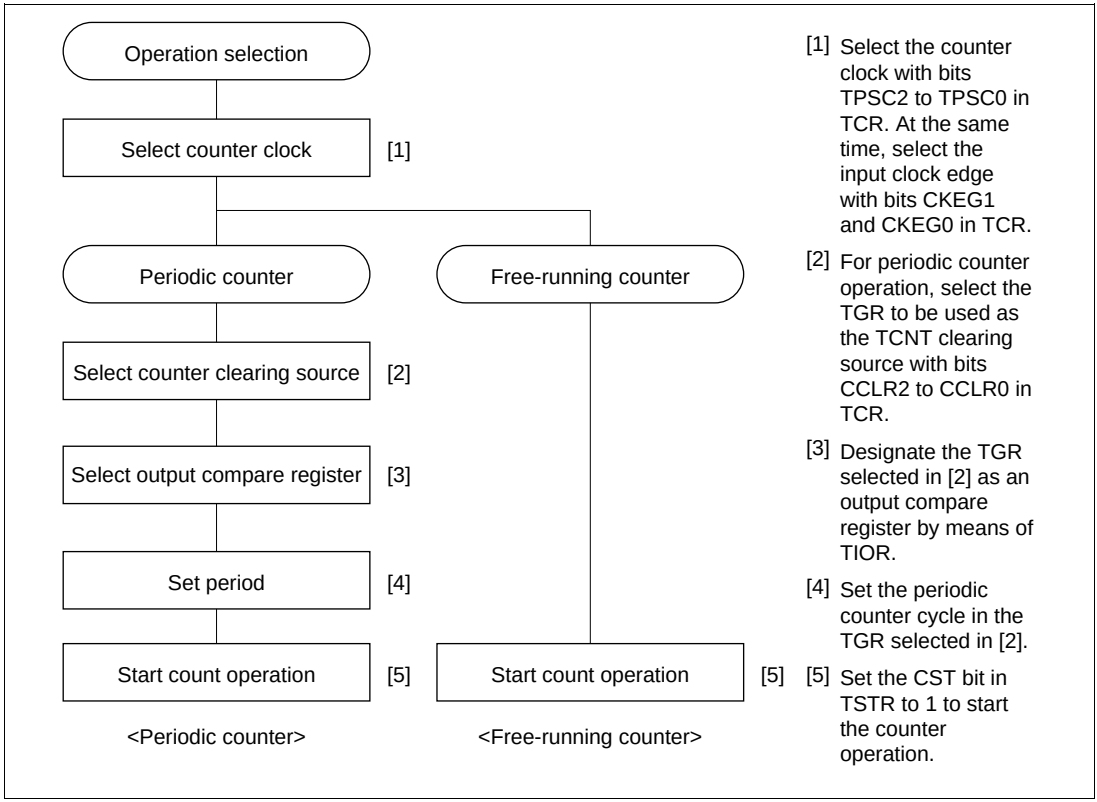


Figure 10-6 Example of Counter Operation Setting Procedure

- Free-running count operation and periodic count operation

  Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

  Figure 10-7 illustrates free-running counter operation.

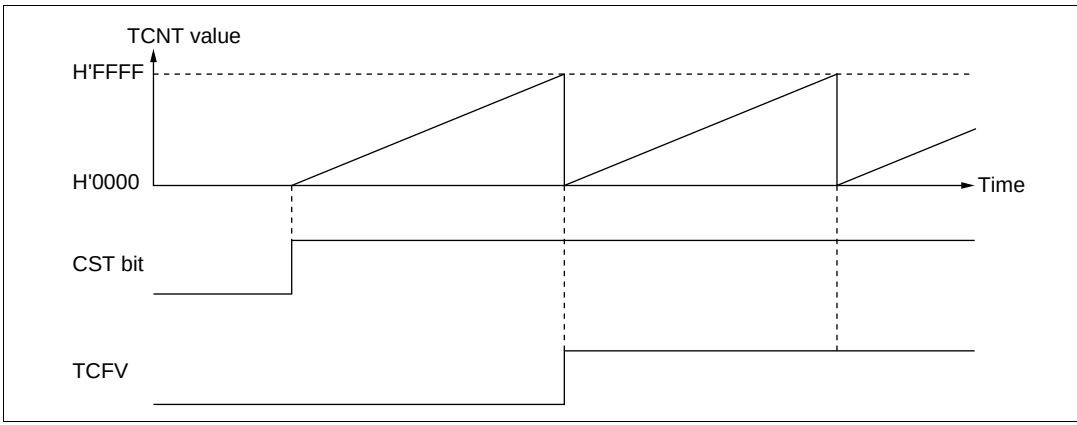


Figure 10-7 Free-Running Counter Operation

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 up-count operation as 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 10-8 illustrates periodic counter operation.

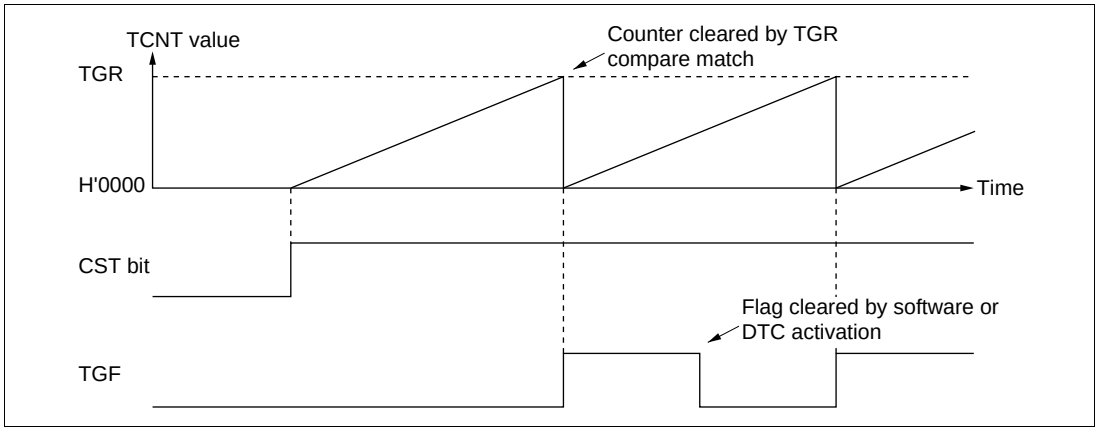


Figure 10-8 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

- Example of setting procedure for waveform output by compare match

  Figure 10-9 shows an example of the setting procedure for waveform output by compare match.

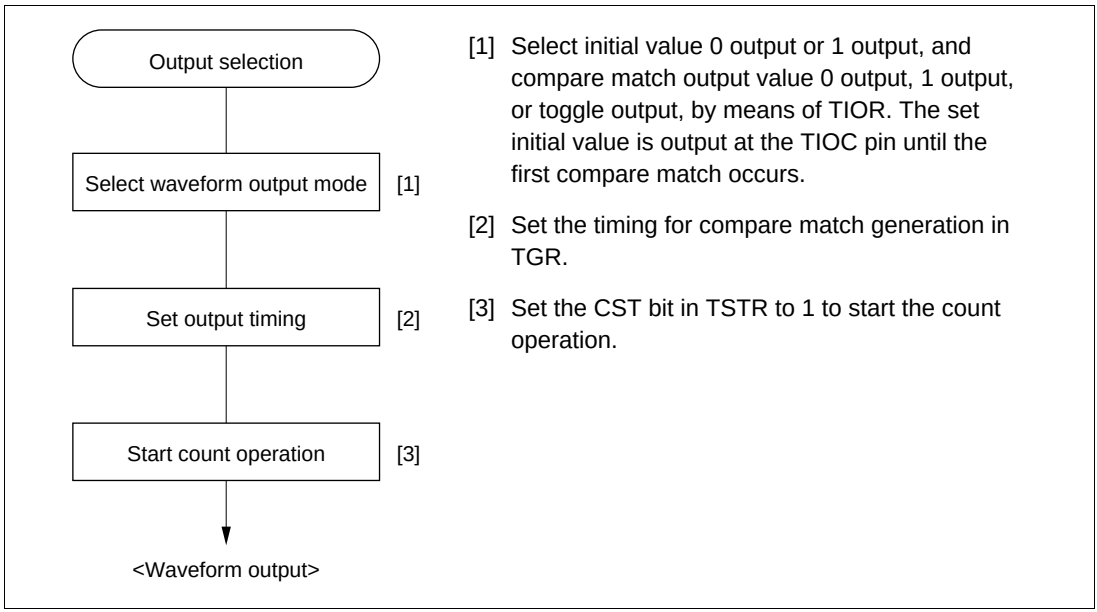


Figure 10-9 Example Of Setting Procedure For Waveform Output By Compare Match

- Examples of waveform output operation

  Figure 10-10 shows an example of 0 output/1 output.

  In this example TCNT has been designated as a free-running counter, and settings have been made 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 coincide, the pin level does not change.

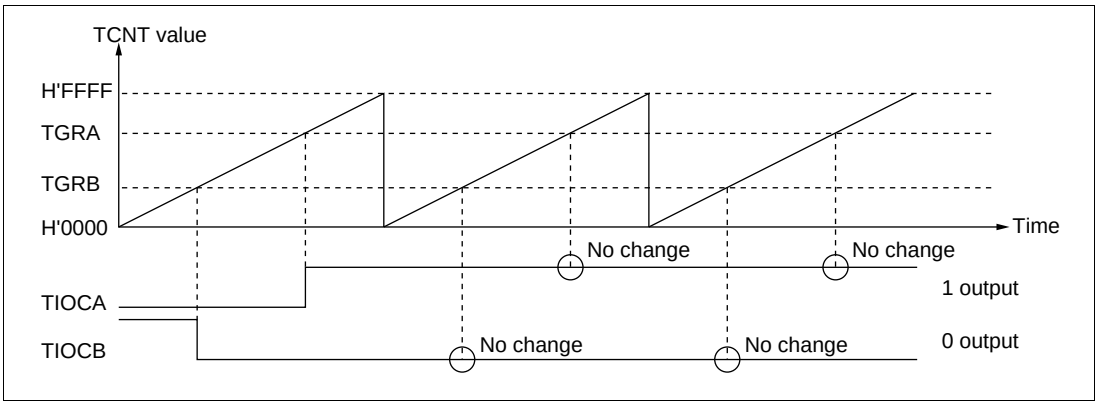


Figure 10-10 Example of 0 Output/1 Output Operation

Figure 10-11 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.

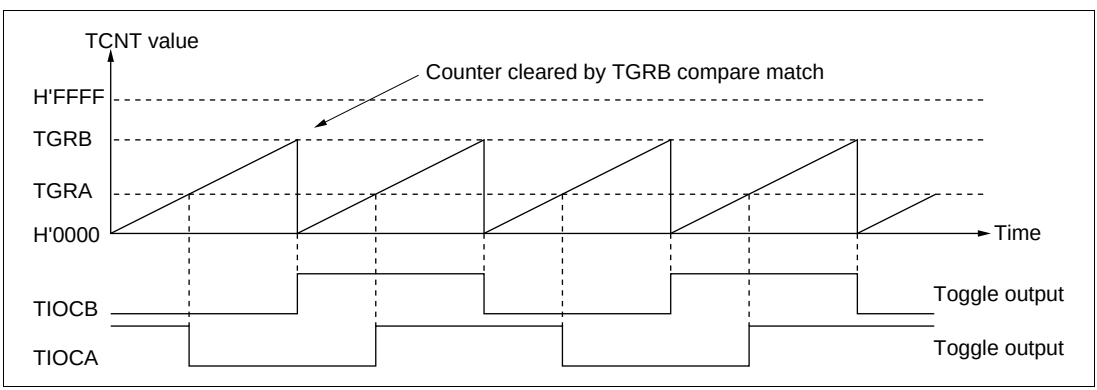


Figure 10-11 Example of Toggle Output Operation

Input Capture Function: The TCNT value can be transferred to TGR on detection of the TIOC pin input edge. Rising edge, falling edge, or both edges can be selected as the detected edge.

- Example of input capture operation setting procedure
  Figure 10-12 shows an example of the input capture operation setting procedure.

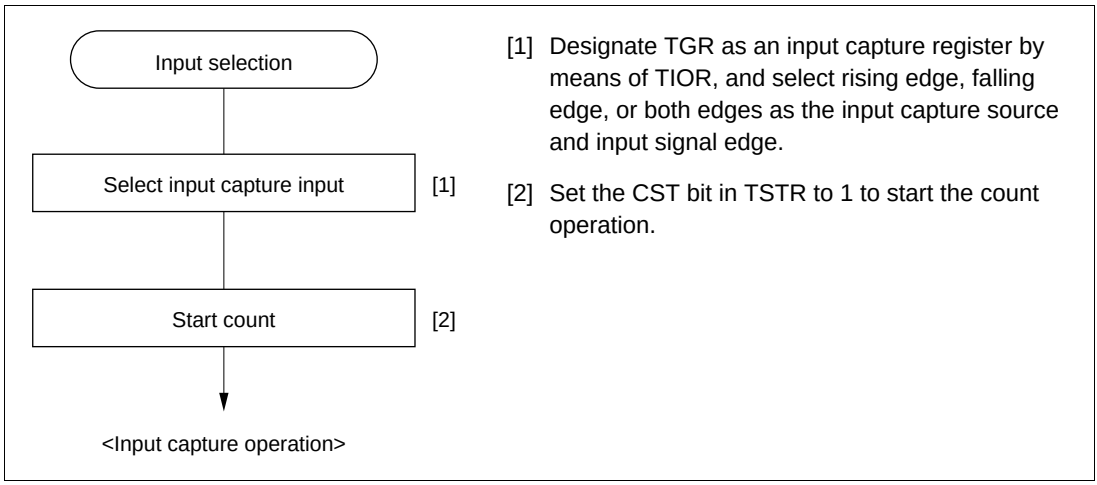


Figure 10-12 Example of Input Capture Operation Setting Procedure

- Example of input capture operation

  Figure 10-13 shows an example of input capture operation.

  In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

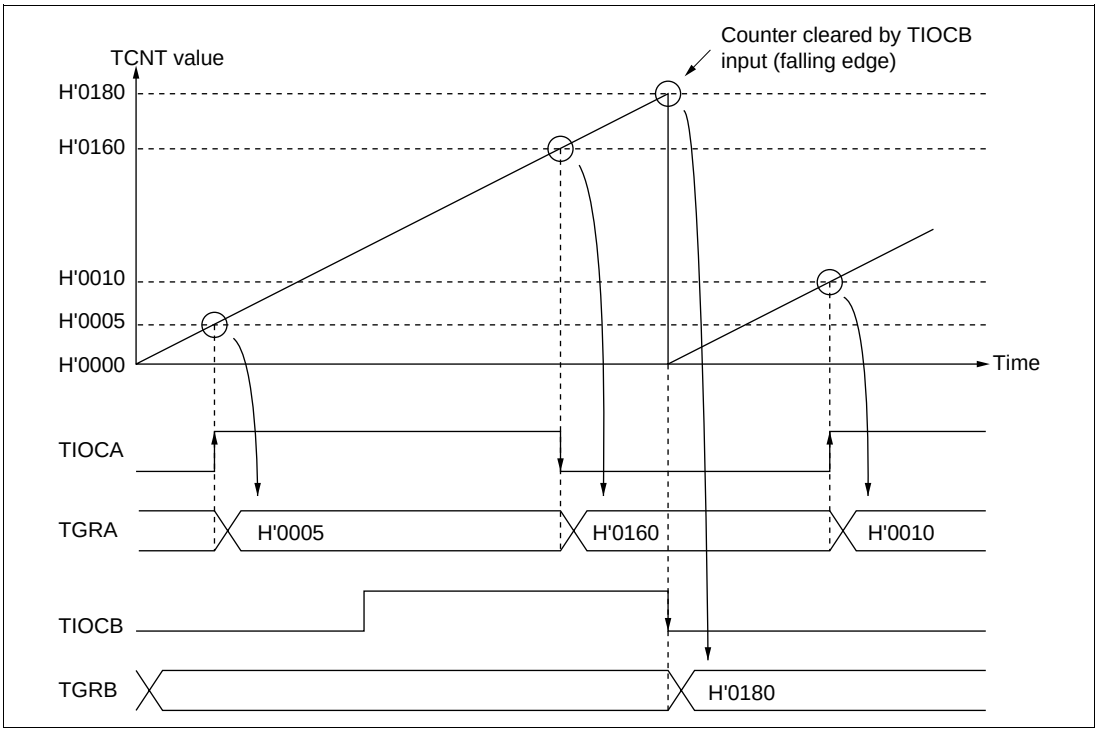


Figure 10-13 Example of Input Capture Operation

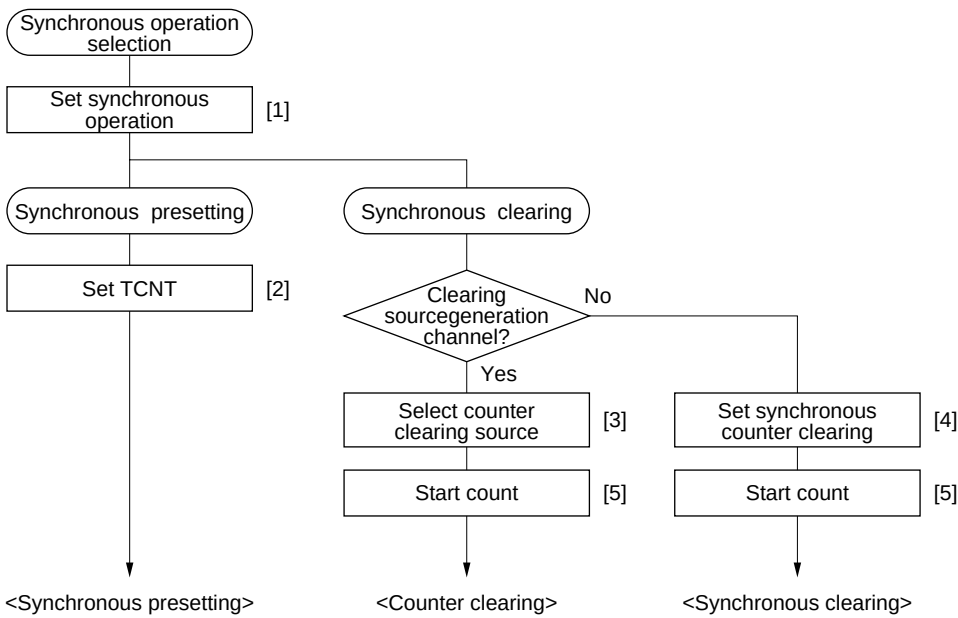
10.4.3 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 2 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure: Figure 10-14 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.

[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10-14 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 10-15 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGR0B compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGR0B compare match, is performed for channel 0 to 2 TCNT counters, and the data set in TGR0B is used as the PWM cycle.

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

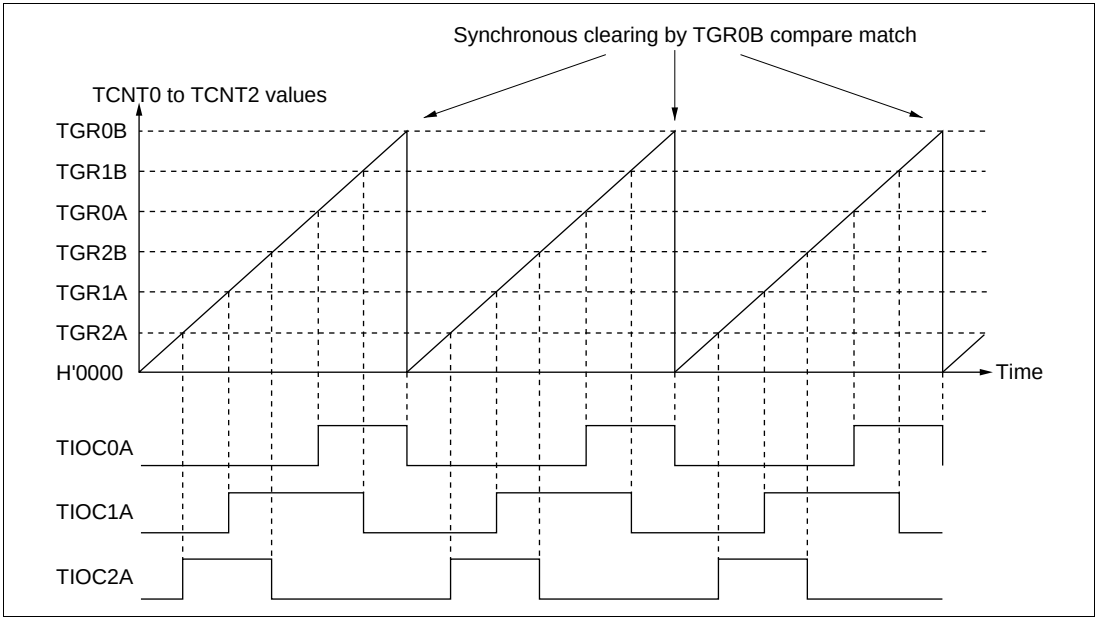


Figure 10-15 Example of Synchronous Operation

10.4.4 Buffer Operation

Buffer operation, provided for channels 0 and 3, enables TGRC and TGRD to be used as buffer registers.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Table 10-5 shows the register combinations used in buffer operation.

Table 10-5 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGR0A	TGR0C
	TGR0B	TGR0D

- 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 10-16.

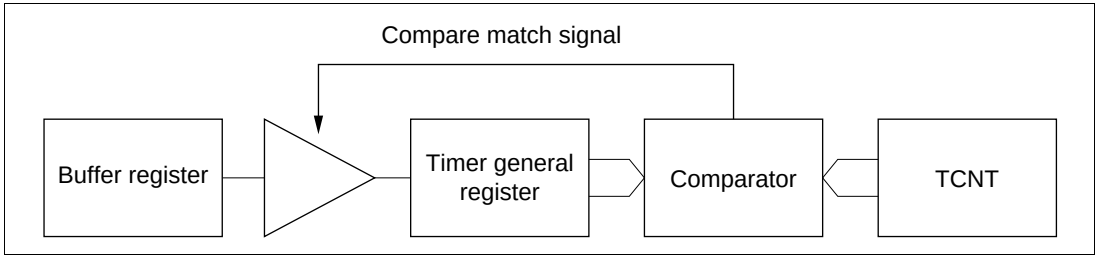


Figure 10-16 Compare Match Buffer Operation

- 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 10-17.

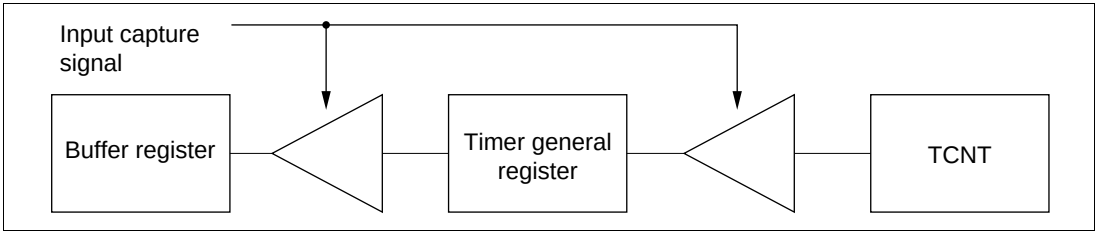


Figure 10-17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 10-18 shows an example of the buffer operation setting procedure.

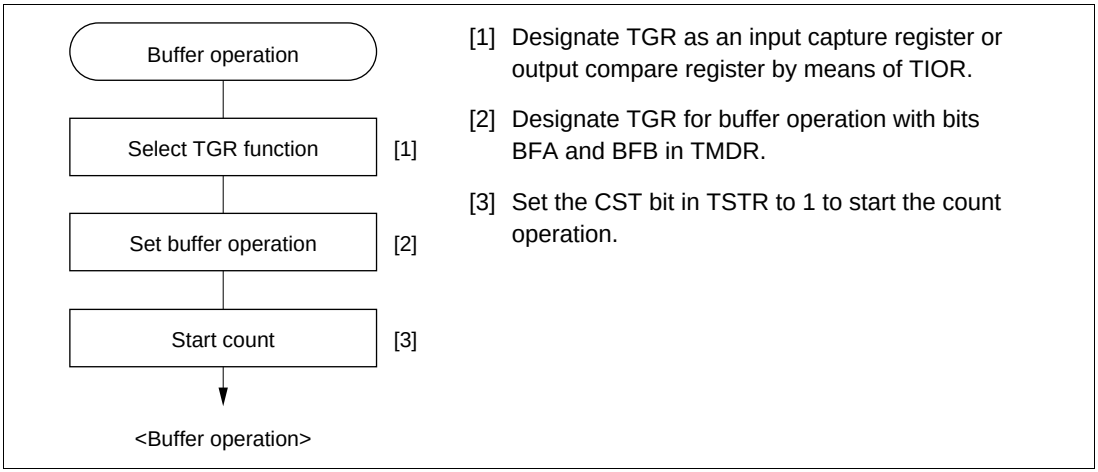


Figure 10-18 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation

- When TGR is an output compare register

  Figure 10-19 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 of PWM modes, see section 10.4.5, PWM Modes.

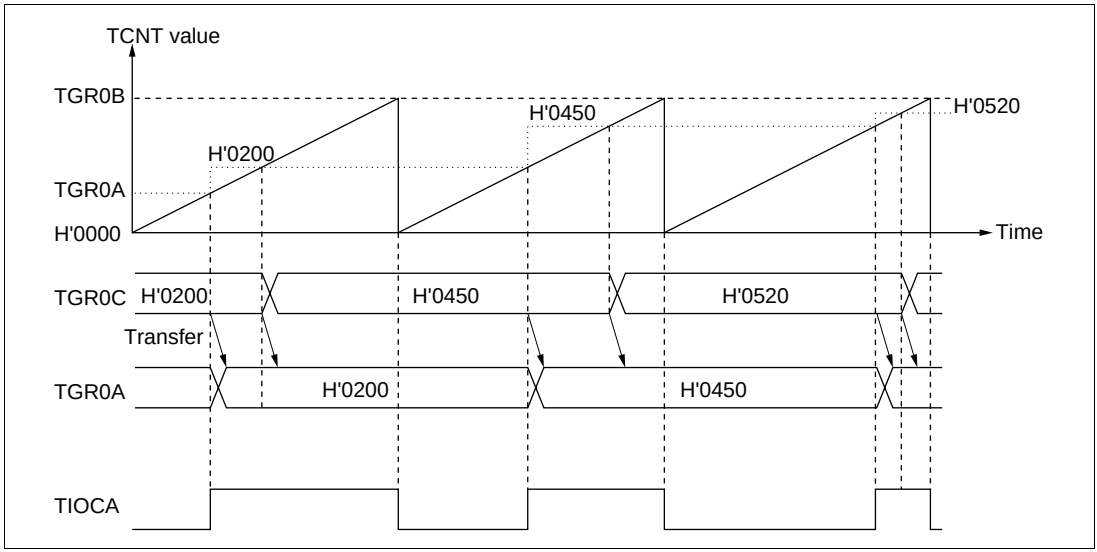


Figure 10-19 Example of Buffer Operation (1)

- When TGR is an input capture register

  Figure 10-20 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

  Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

  As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

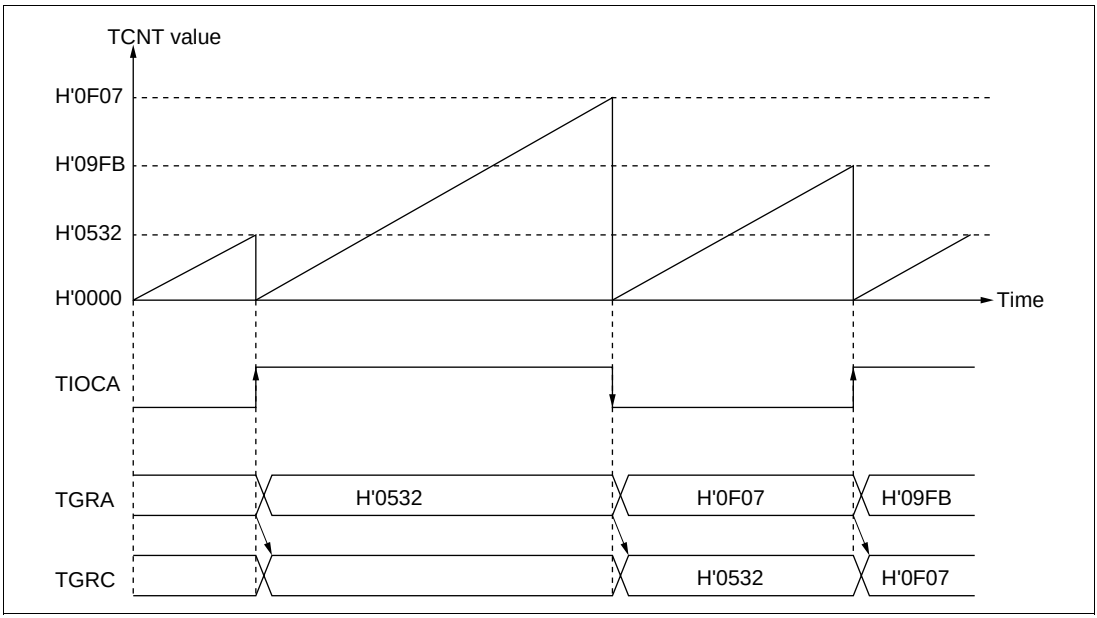


Figure 10-20 Example of Buffer Operation (2)

10.4.5 PWM Modes

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.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 4-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 7-phase PWM output is possible by combined use with synchronous operation.

The correspondence between PWM output pins and registers is shown in table 10-6.

Table 10-6 PWM Output Registers and Output Pins

Channel	Registers	Output Pins	
		PWM Mode 1	PWM Mode 2
0	TGR0A	TIOCA0	TIOCA0
	TGR0B		TIOCB0
	TGR0C	TIOCC0	TIOCC0
	TGR0D		TIOCD0
1	TGR1A	TIOCA1	TIOCA1
	TGR1B		TIOCB1
2	TGR2A	TIOCA2	TIOCA2
	TGR2B		TIOCB2

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

Example of PWM Mode Setting Procedure: Figure 10-21 shows an example of the PWM mode setting procedure.

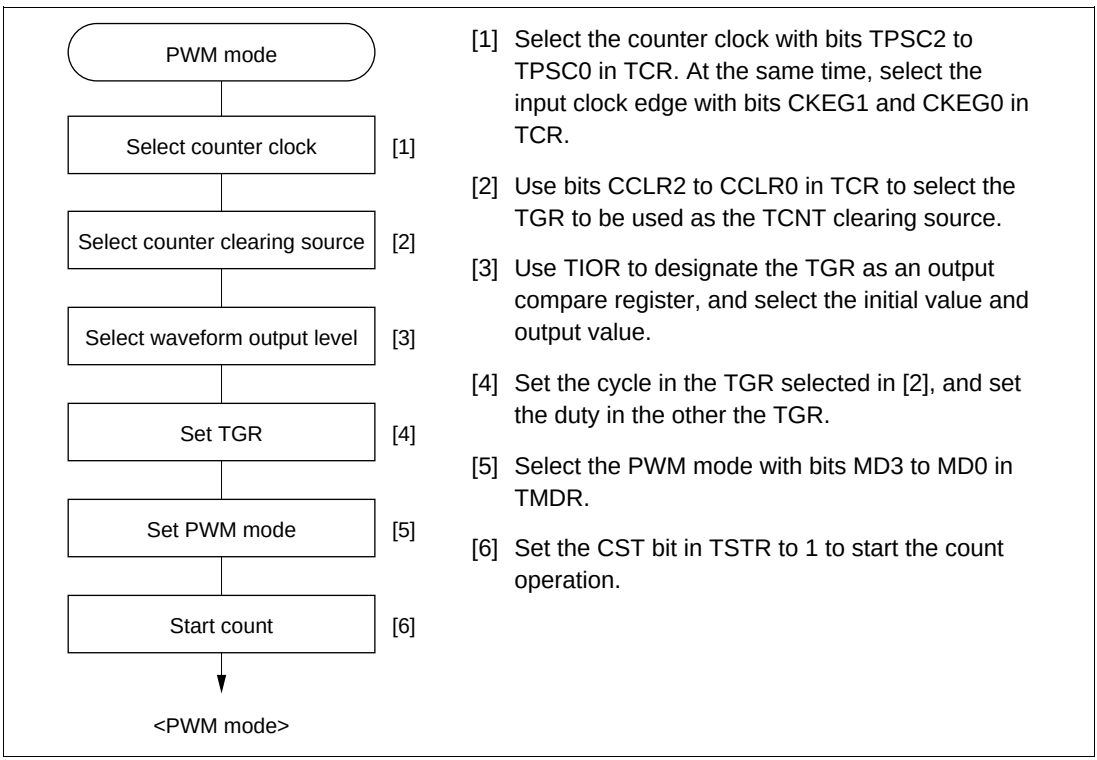


Figure 10-21 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 10-22 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in TGRB registers as the duty.

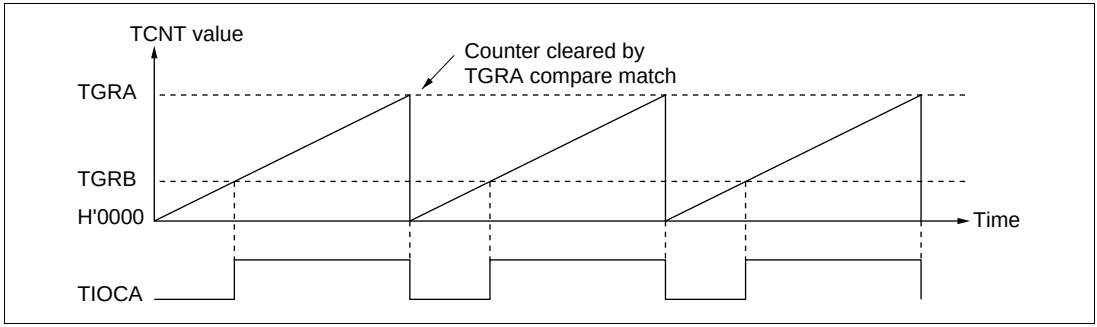


Figure 10-22 Example of PWM Mode Operation (1)

Figure 10-23 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGR1B 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 (TGR0A to TGR0D, TGR1A), to output a 5-phase PWM waveform.

In this case, the value set in TGR1B is used as the cycle, and the values set in the other TGRs as the duty.

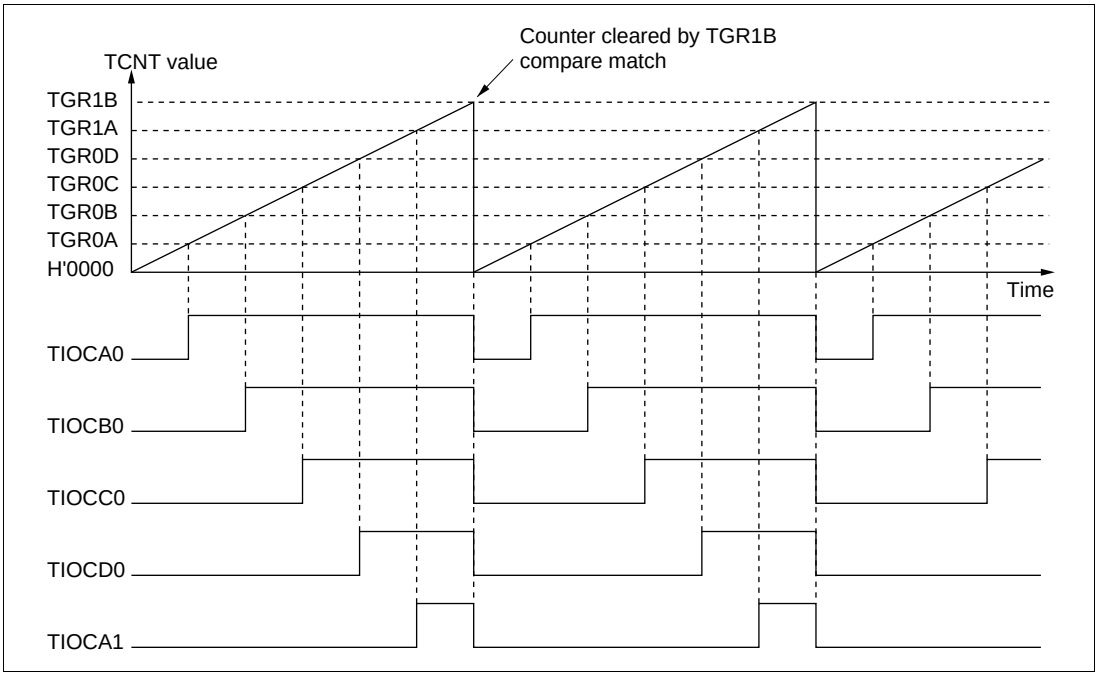


Figure 10-23 Example of PWM Mode Operation (2)

Figure 10-24 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

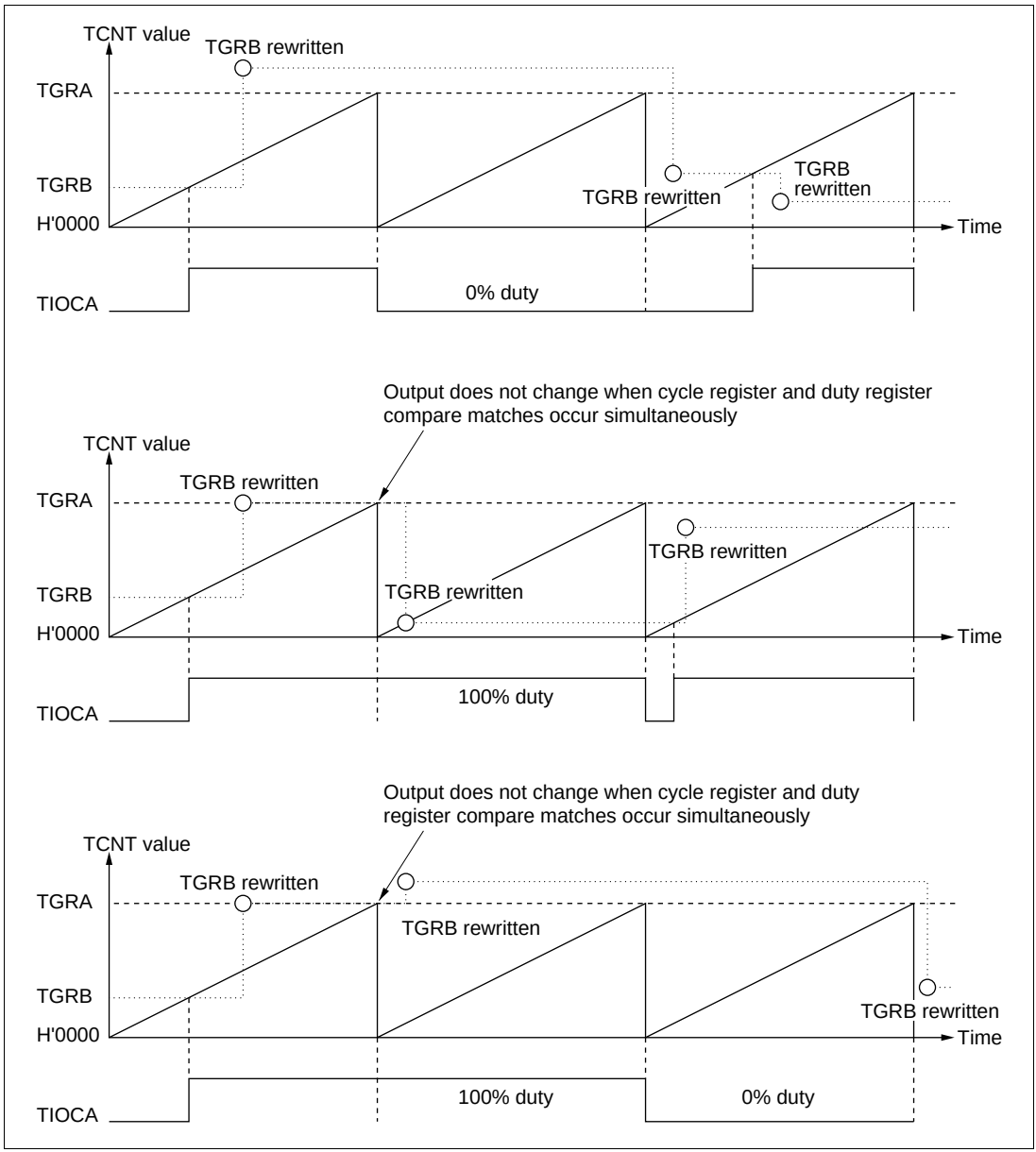


Figure 10-24 Example of PWM Mode Operation (3)

10.4.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1and 2.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits 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.

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 10-7 shows the correspondence between external clock pins and channels.

Table 10-7 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 is set to phase counting mode	TCLKA	TCLKB
When channel 2 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 10-25 shows an example of the phase counting mode setting procedure.

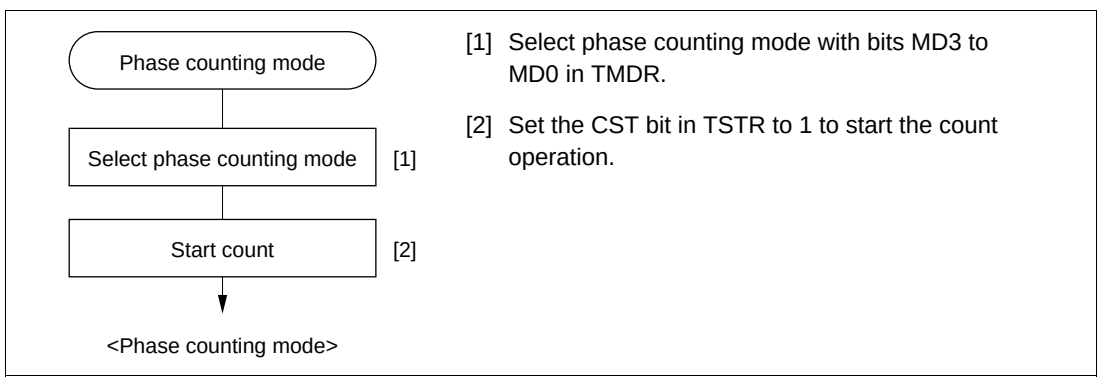


Figure 10-25 Example of Phase Counting Mode Setting Procedure

Examples of Phase Counting Mode Operation: In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

- Phase counting mode 1

  Figure 10-26 shows an example of phase counting mode 1 operation, and table 10-8 summarizes the TCNT up/down-count conditions.

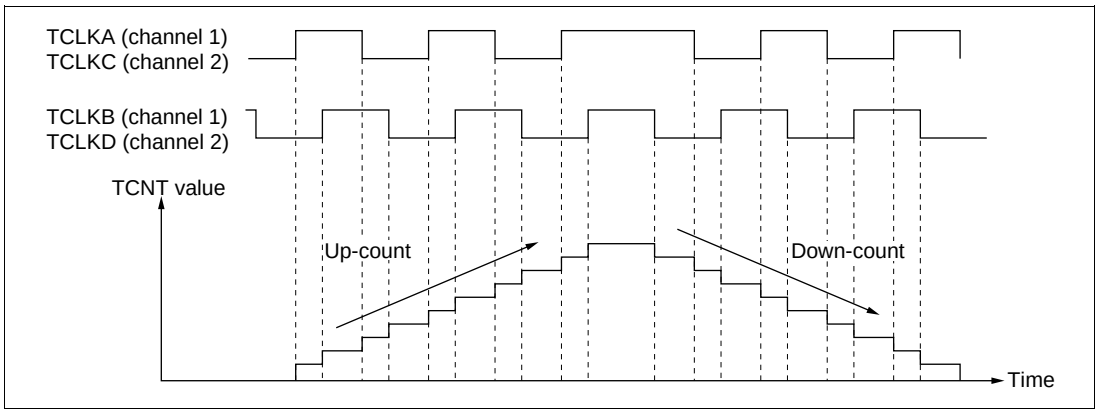


Figure 10-26 Example of Phase Counting Mode 1 Operation

Table 10-8 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	⌐↑	Up-count
Low level	⌐↓	
⌐↑	Low level	
⌐↓	High level	
High level	⌐↓	Down-count
Low level	⌐↑	
⌐↑	High level	
⌐↓	Low level	

Legend

⌐↑ : Rising edge

⌐↓ : Falling edge

- Phase counting mode 2

  Figure 10-27 shows an example of phase counting mode 2 operation, and table 10-9 summarizes the TCNT up/down-count conditions.

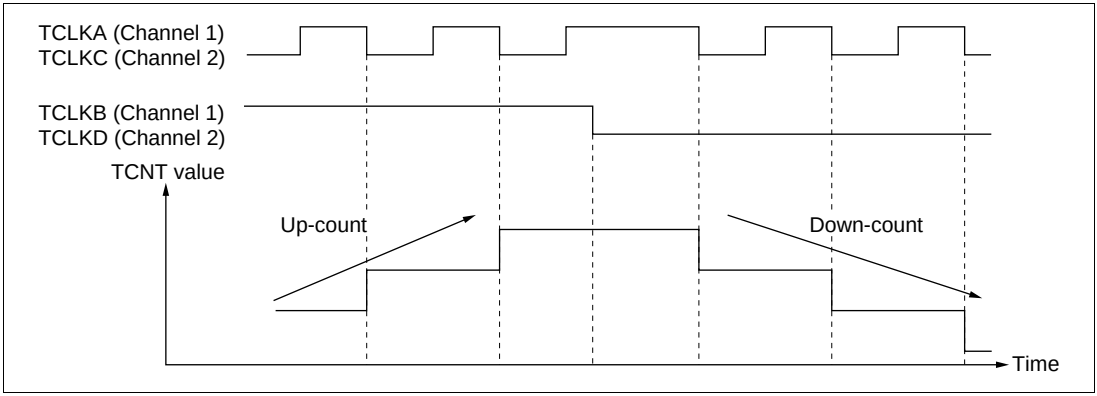


Figure 10-27 Example of Phase Counting Mode 2 Operation

Table 10-9 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

Legend

↑ : Rising edge

↓ : Falling edge

- Phase counting mode 3

  Figure 10-28 shows an example of phase counting mode 3 operation, and table 10-10 summarizes the TCNT up/down-count conditions.

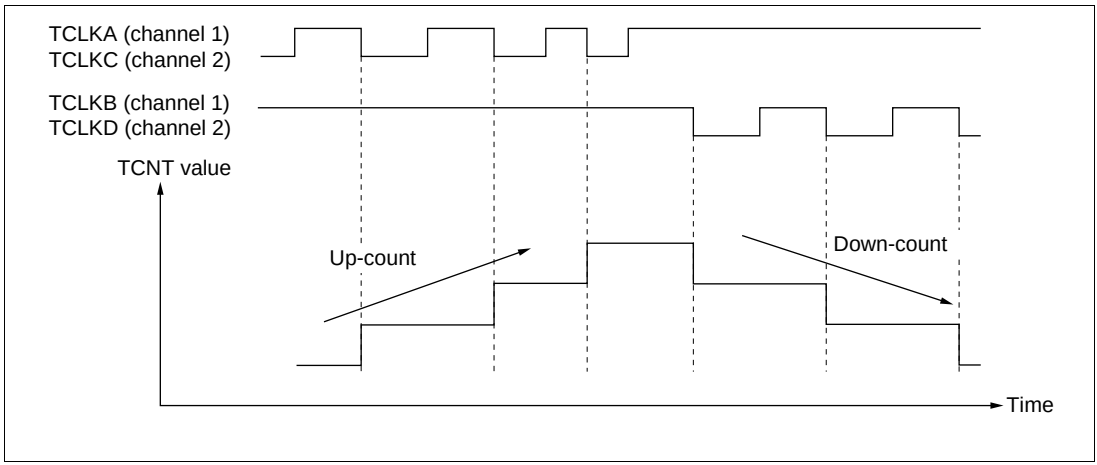


Figure 10-28 Example of Phase Counting Mode 3 Operation

Table 10-10 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Down-count
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Don't care

Legend

↑ : Rising edge

↓ : Falling edge

- Phase counting mode 4

  Figure 10-29 shows an example of phase counting mode 4 operation, and table 10-11 summarizes the TCNT up/down-count conditions.

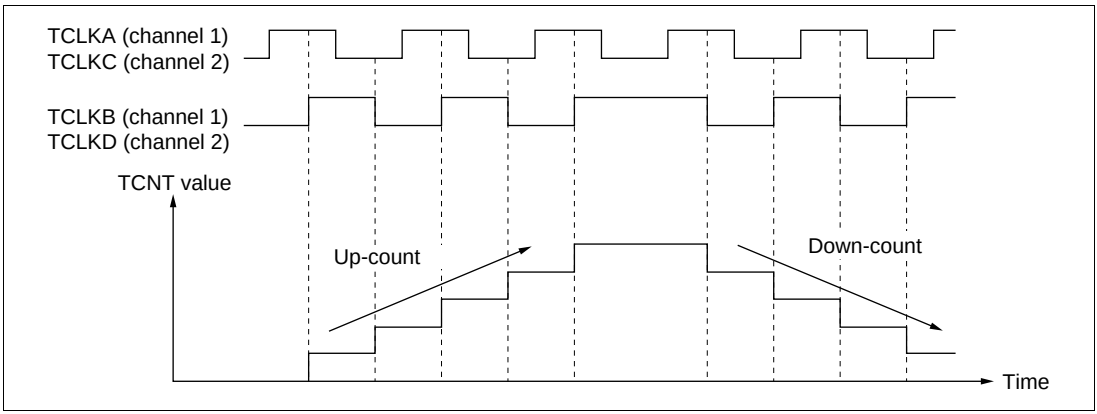


Figure 10-29 Example of Phase Counting Mode 4 Operation

Table 10-11 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channel 1) TCLKC (Channel 2)	TCLKB (Channel 1) TCLKD (Channel 2)	Operation
High level	⌐↑	Up-count
Low level	⌐↓	
⌐↑	Low level	Don't care
⌐↓	High level	
High level	⌐↓	Down-count
Low level	⌐↑	
⌐↑	High level	Don't care
⌐↓	Low level	

Legend

⌐↑ : Rising edge

⌐↓ : Falling edge

10.5 Interrupts

10.5.1 Interrupt Sources and Priorities

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/disabled 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 10-12 lists interrupt sources and DMA controller (DMAC) and data transfer controller (DTC) activation.

Table 10-12 Interrupt Sources and DMA Controller (DMAC) and Data Transfer (DTC) Activation

Channel	Interrupt Source	Description	DMAC Activation	DTC Activation	Priority
0	TGI0A	TGR0A input capture/compare match	Possible	Possible	High
	TGI0B	TGR0B input capture/compare match	Not possible	Possible	
	TGI0C	TGR0C input capture/compare match	Not possible	Possible	
	TGI0D	TGR0D input capture/compare match	Not possible	Possible	
	TCI0V	TCNT0 overflow	Not possible	Not possible	
1	TGI1A	TGR1A input capture/compare match	Possible	Possible	
	TGI1B	TGR1B input capture/compare match	Not possible	Possible	
	TCI1V	TCNT1 overflow	Not possible	Not possible	
	TCI1U	TCNT1 underflow	Not possible	Not possible	
2	TGI2A	TGR2A input capture/compare match	Possible	Possible	
	TGI2B	TGR2B input capture/compare match	Not possible	Possible	
	TCI2V	TCNT2 overflow	Not possible	Not possible	
	TCI2U	TCNT2 underflow	Not possible	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Input Capture/Compare Match Interrupt: An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has eight input capture/compare match interrupts, four for channel 0, and two each for channels 1 and 2.

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 three 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 two overflow interrupts, one each for channels 1 and 2.

10.5.2 DTC Activation

DTC Activation: The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 8, Data Transfer Controller (DTC).

A total of eight TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channel 0, and two each for channels 1 and 2.

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).

With the TPU, a total of three TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

10.6 Operation Timing

10.6.1 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.

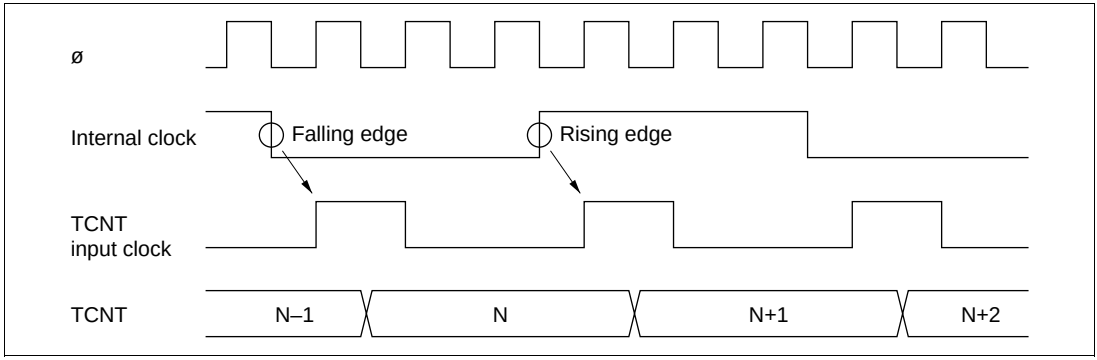


Figure 10-30 Count Timing in Internal Clock Operation

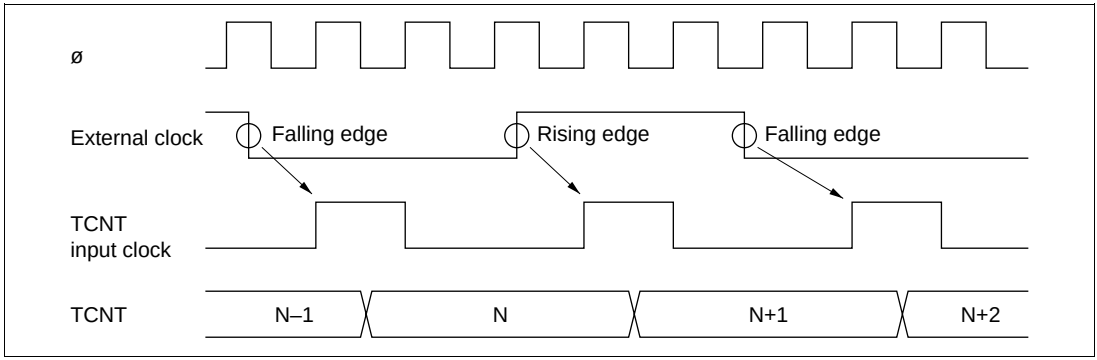


Figure 10-31 Count Timing in External Clock Operation

Output Compare Output Timing: A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin (TIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 10-32 shows output compare output timing.

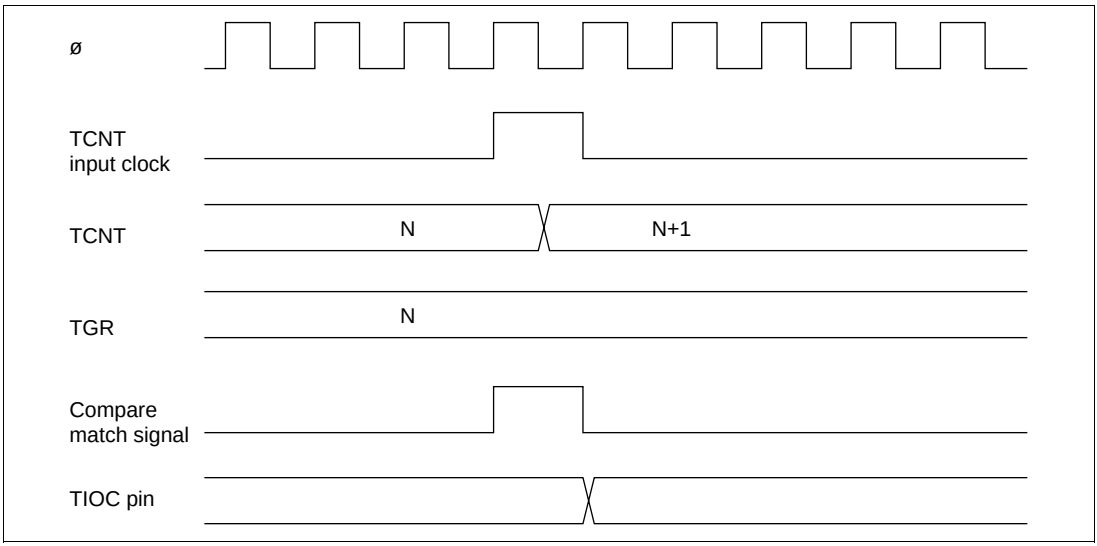


Figure 10-32 Output Compare Output Timing

Input Capture Signal Timing: Figure 10-33 shows input capture signal timing.

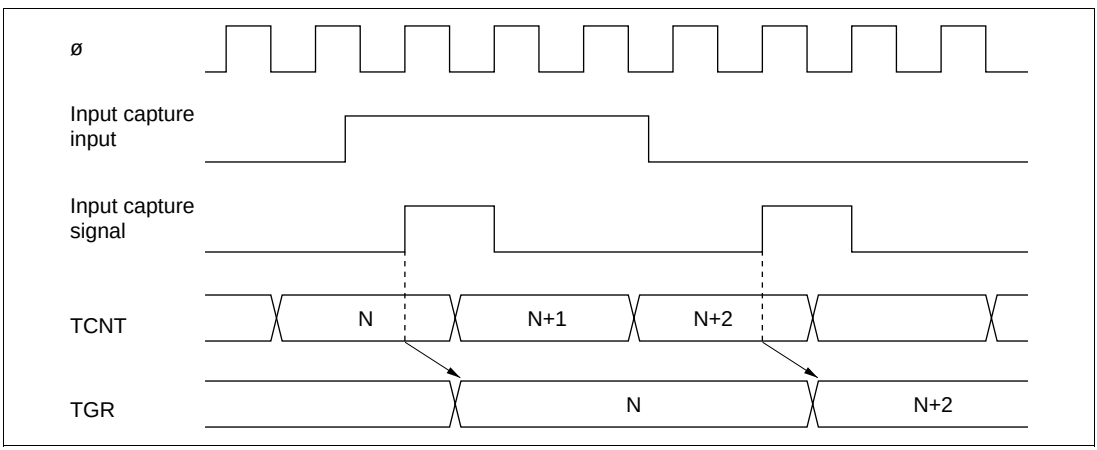


Figure 10-33 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 10-34 shows the timing when counter clearing by compare match occurrence is specified, and figure 10-35 shows the timing when counter clearing by input capture occurrence is specified.

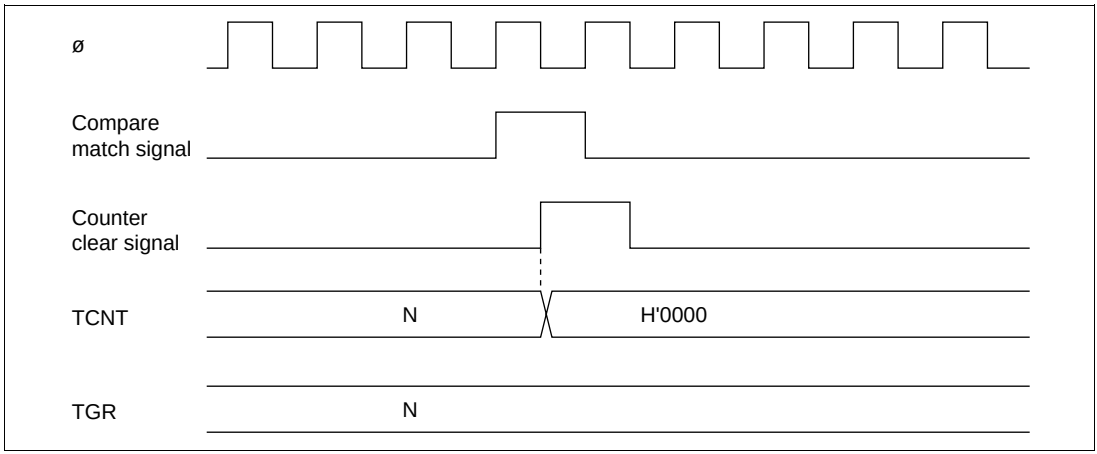


Figure 10-34 Counter Clear Timing (Compare Match)

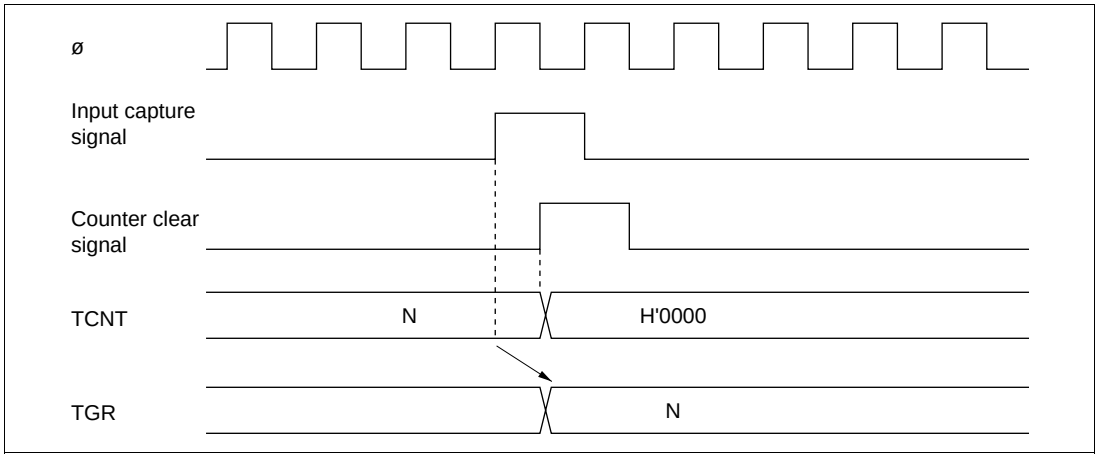


Figure 10-35 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 10-36 and 10-37 show the timing in buffer operation.

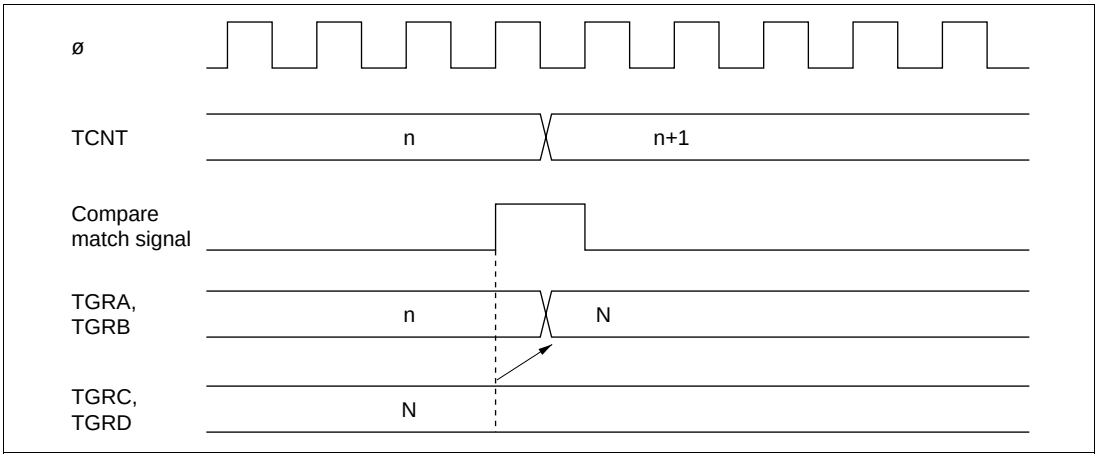


Figure 10-36 Buffer Operation Timing (Compare Match)

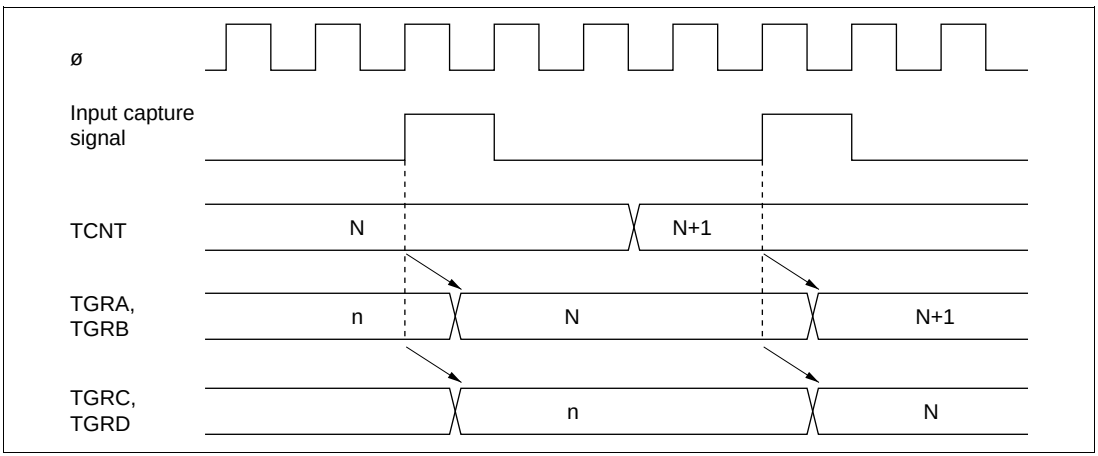


Figure 10-37 Buffer Operation Timing (Input Capture)

10.6.2 Interrupt Signal Timing

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 TGI interrupt request signal timing.

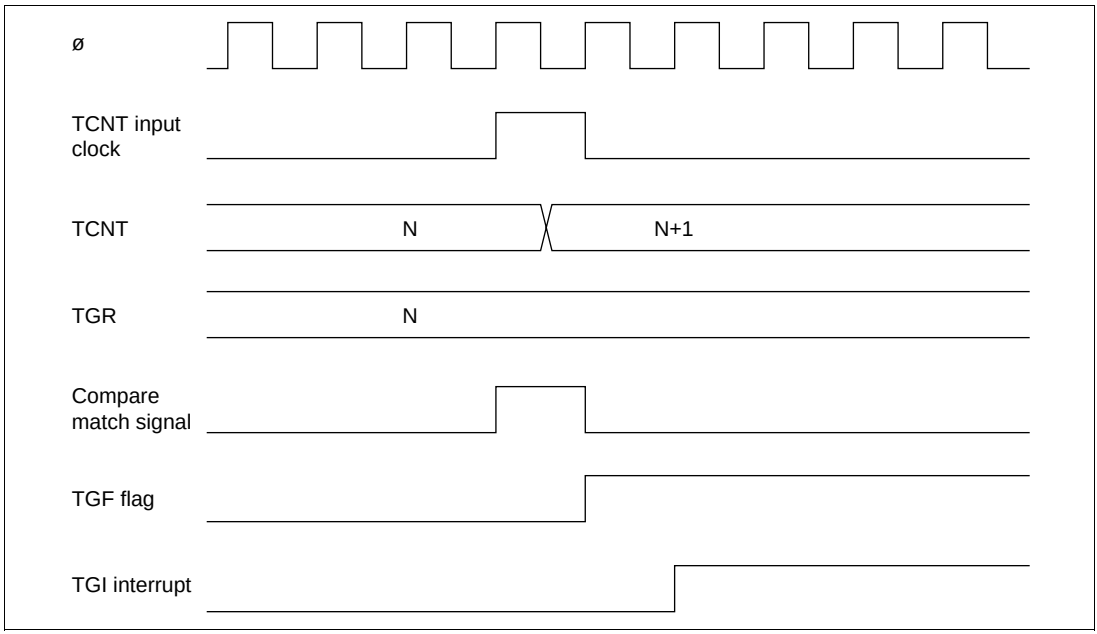


Figure 10-38 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 10-39 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and TGI interrupt request signal timing.

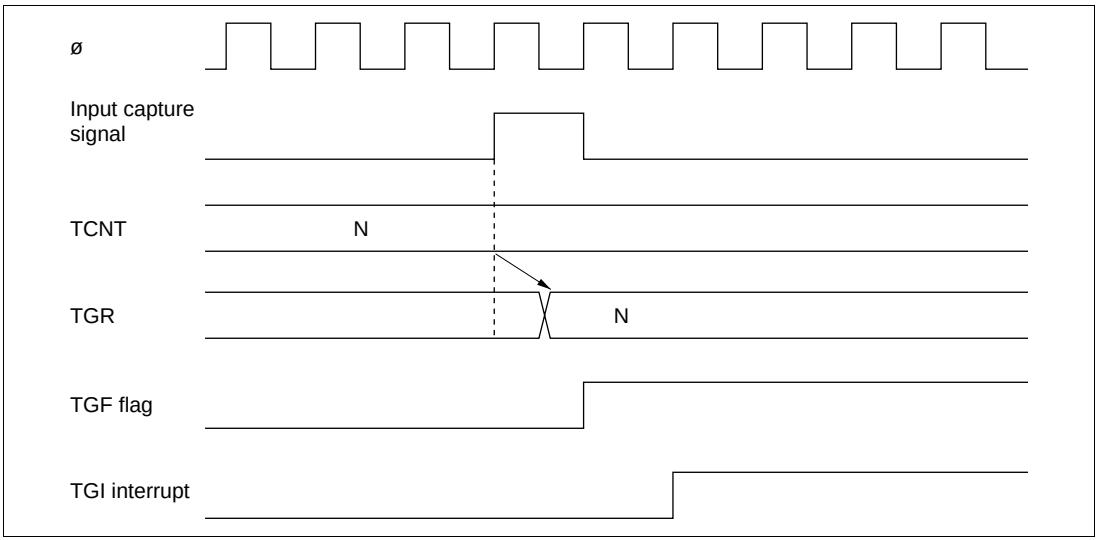


Figure 10-39 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 10-40 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and TCIV interrupt request signal timing.

Figure 10-41 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and TCIU interrupt request signal timing.

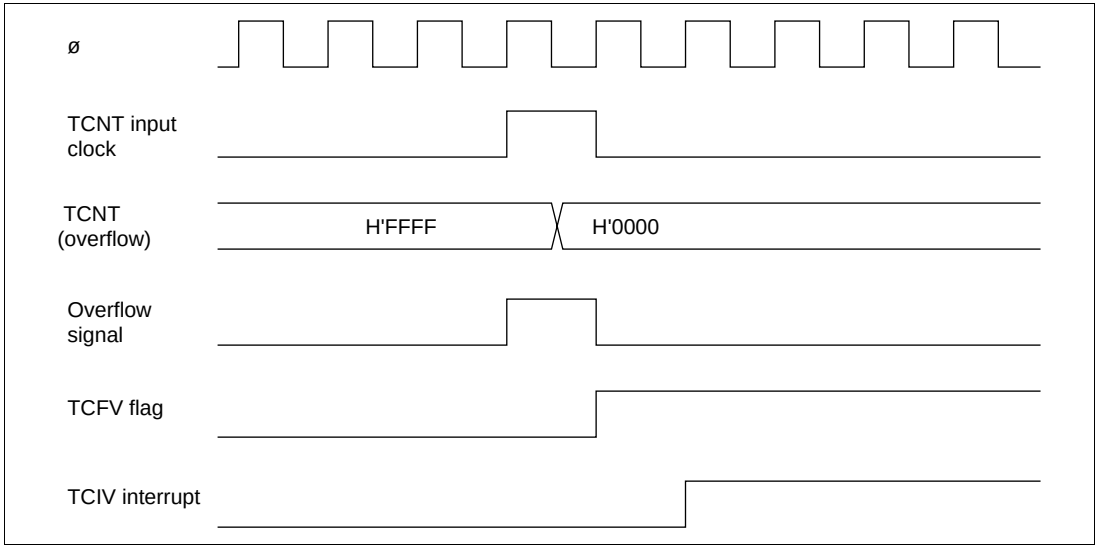


Figure 10-40 TCIV Interrupt Setting Timing

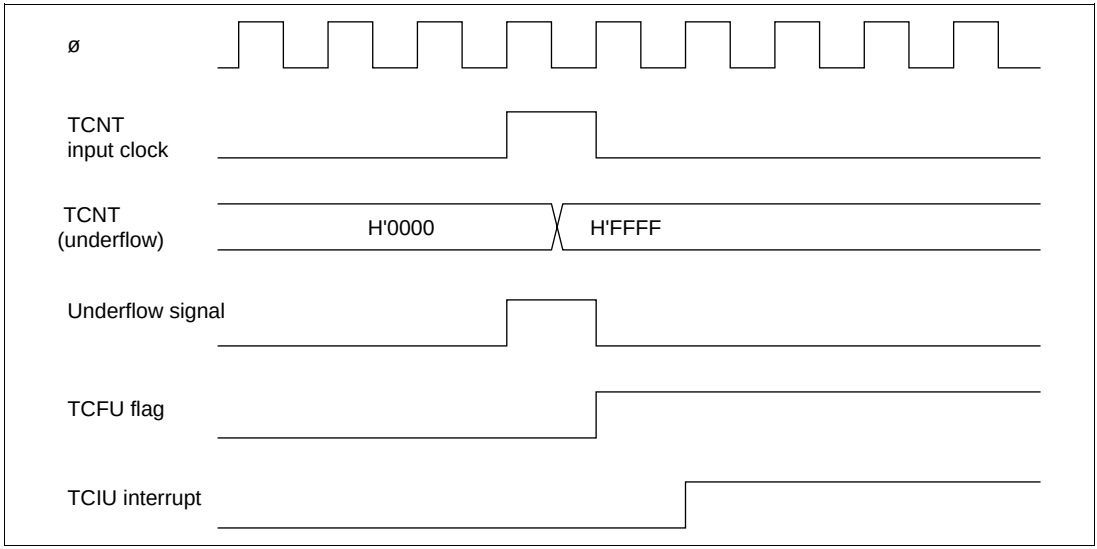


Figure 10-41 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC or DMAC is activated, the flag is cleared automatically. Figure 10-42 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.

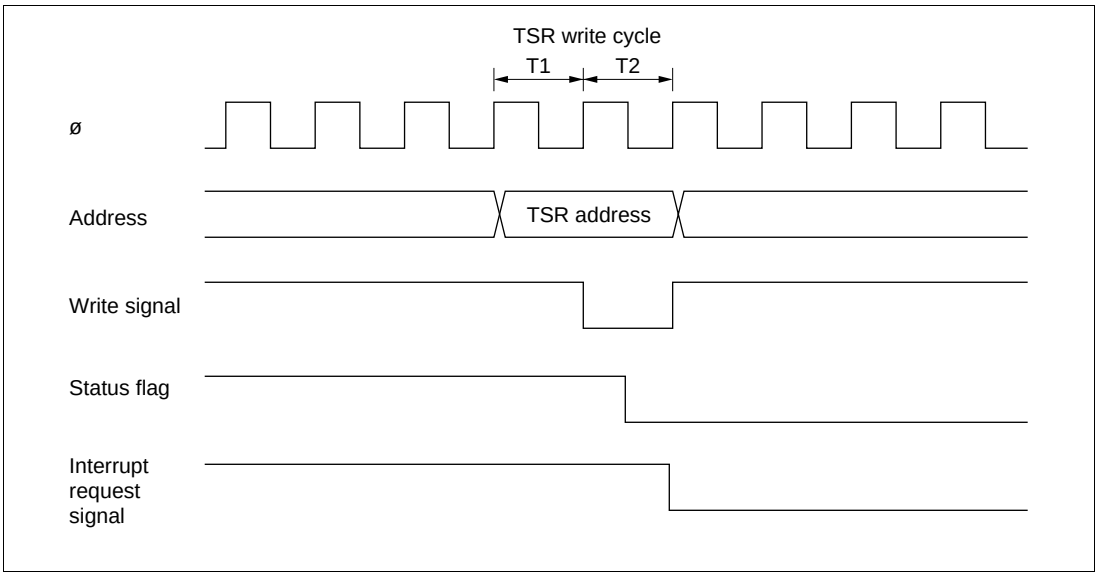


Figure 10-42 Timing for Status Flag Clearing by CPU

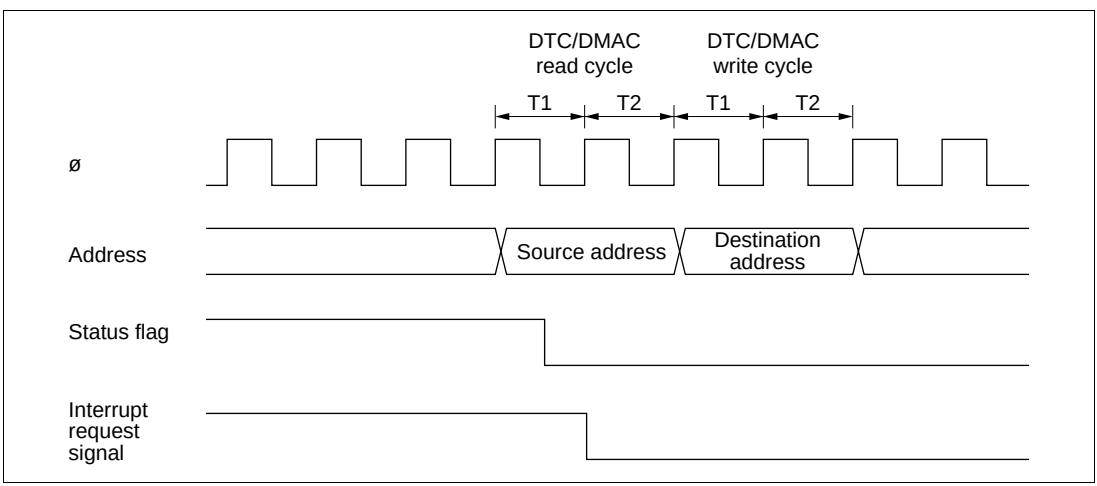


Figure 10-43 Timing for Status Flag Clearing by DTC/DMAC Activation

10.7 Usage Notes

Note that the kinds of operation and contention described below occur during TPU operation.

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.

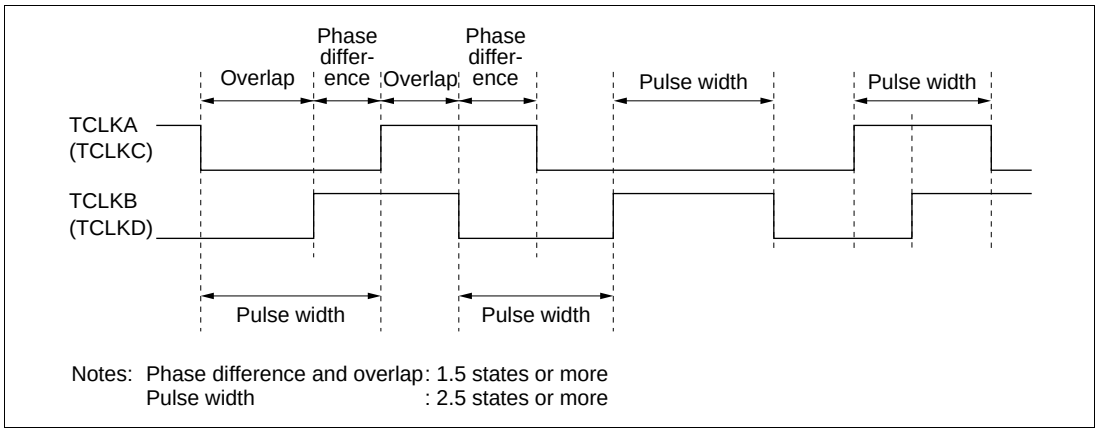


Figure 10-44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

Caution on Period Setting: 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:

$$f = \frac{ø}{(N + 1)}$$

Where f : Counter frequency
ø : Operating frequency
N : TGR set value

Contention between TCNT Write and Clear Operations: If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 10-45 shows the timing in this case.

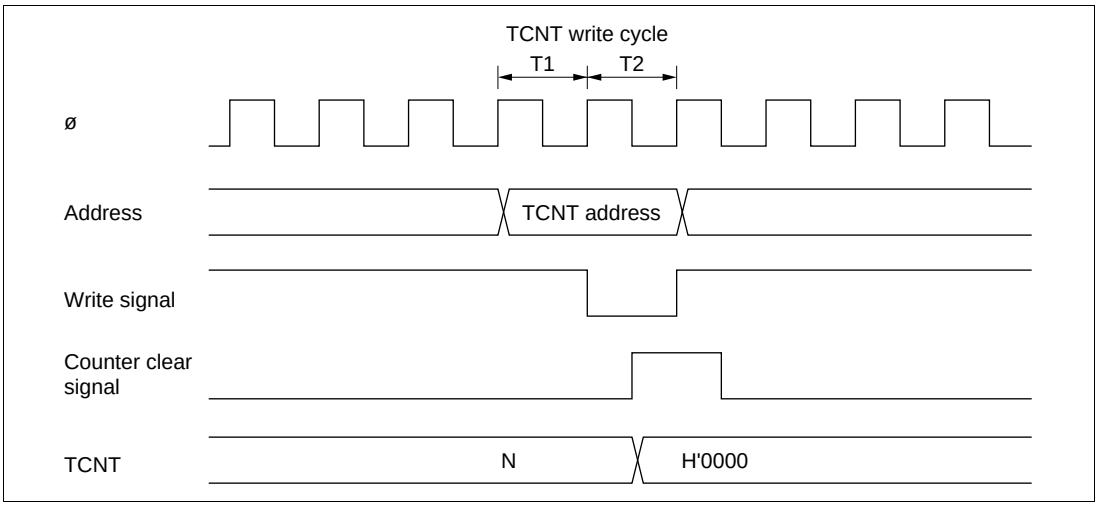


Figure 10-45 Contention between TCNT Write and Clear Operations

Contention between TCNT Write and Increment Operations: If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 10-46 shows the timing in this case.

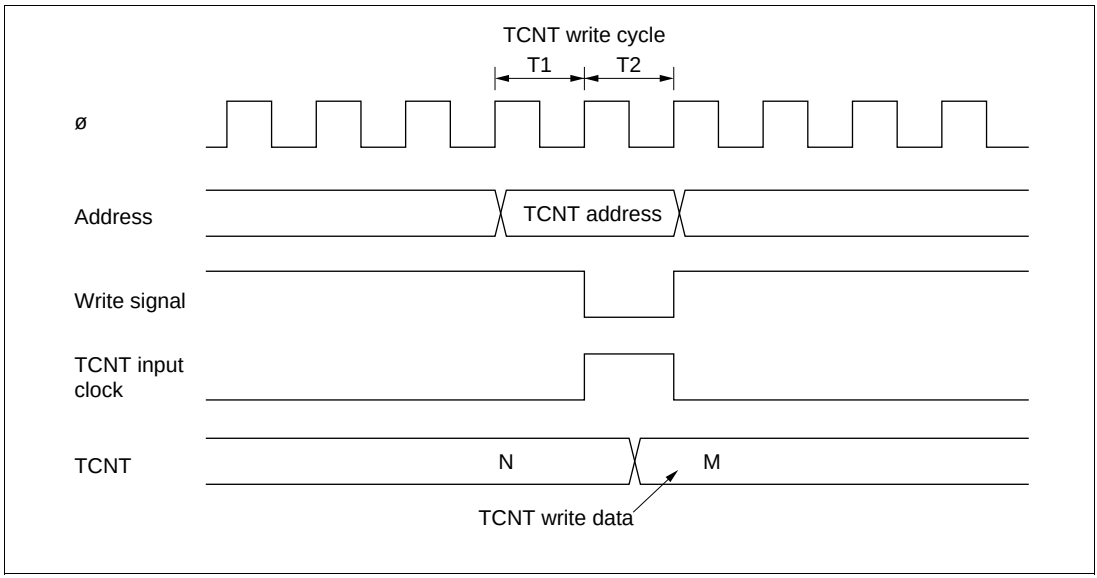


Figure 10-46 Contention between TCNT Write and Increment Operations

Contention between TGR Write and Compare Match: If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes precedence and the compare match signal is inhibited. A compare match does not occur even if the same value as before is written.

Figure 10-47 shows the timing in this case.

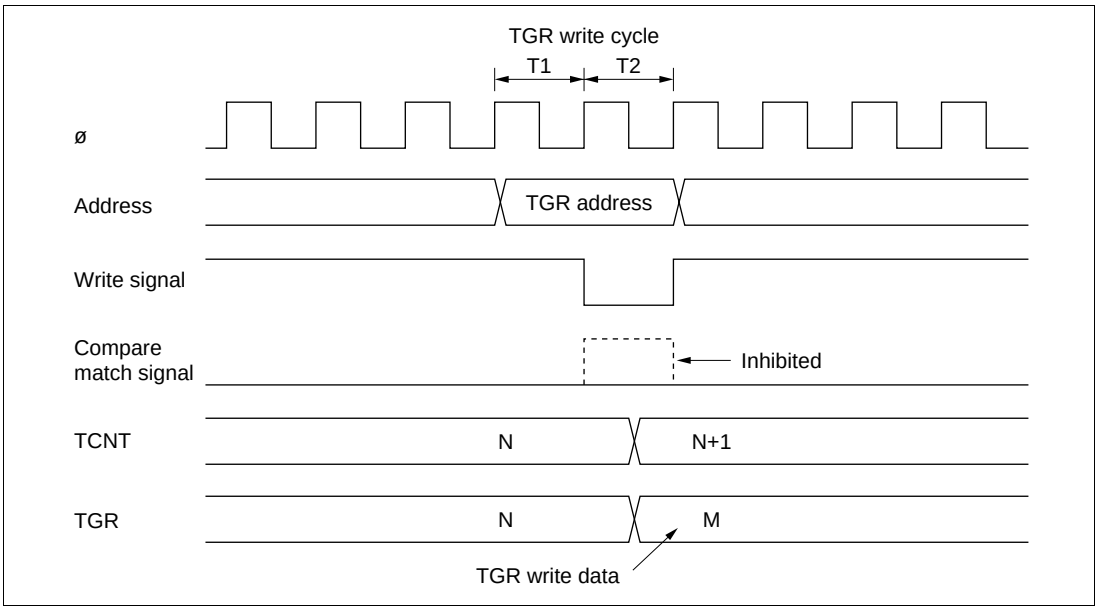


Figure 10-47 Contention between TGR Write and Compare Match

Contention between Buffer Register Write and Compare Match: If a compare match occurs in the T2 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.

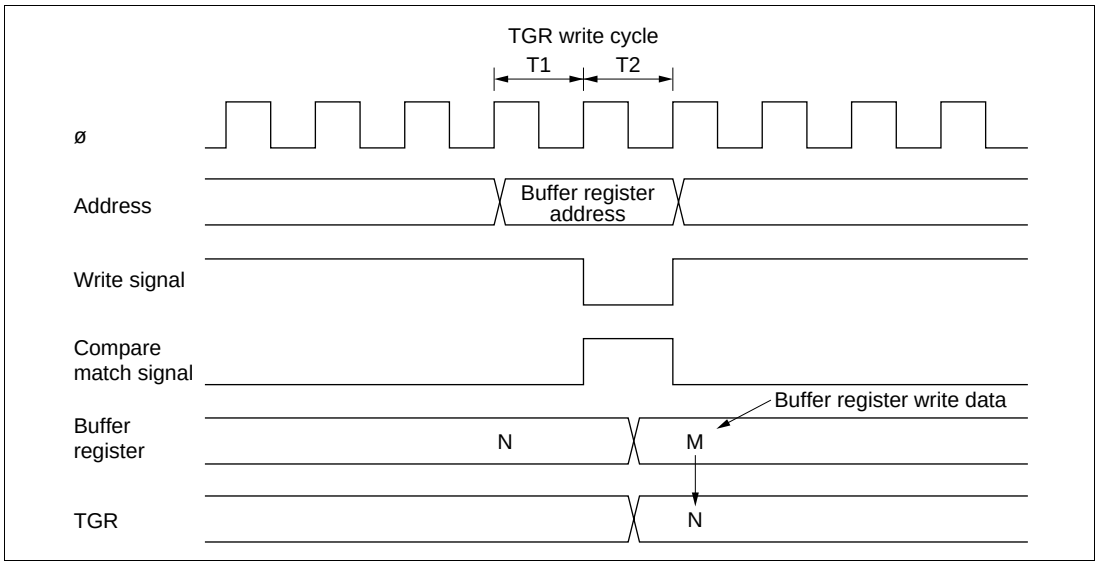


Figure 10-48 Contention between Buffer Register Write and Compare Match

Contention between TGR Read and Input Capture: If the input capture signal is generated in the T1 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.

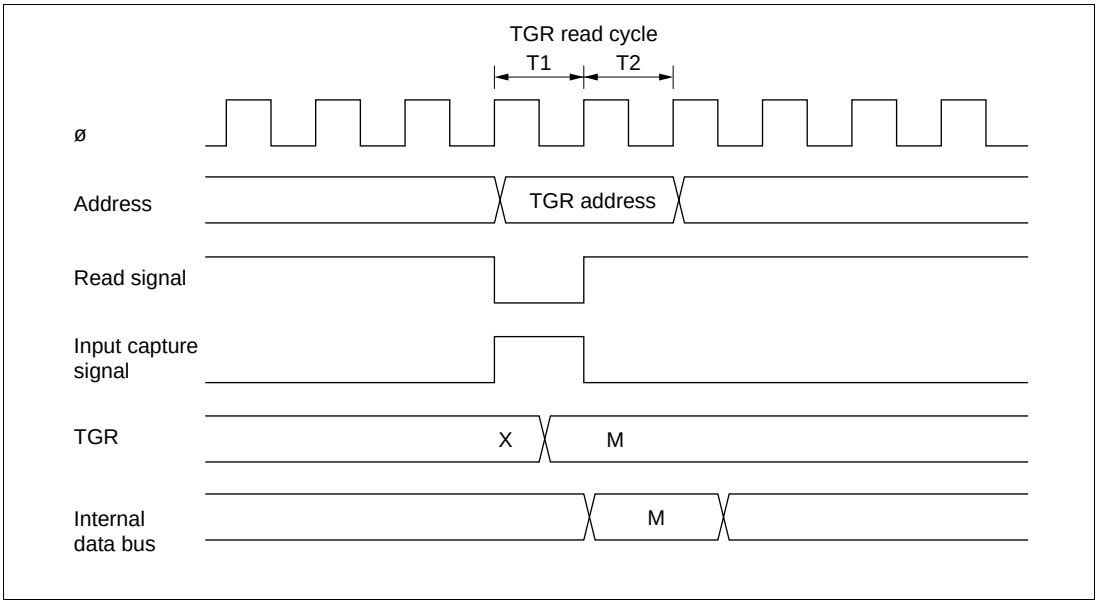


Figure 10-49 Contention between TGR Read and Input Capture

Contention between TGR Write and Input Capture: If the input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 10-50 shows the timing in this case.

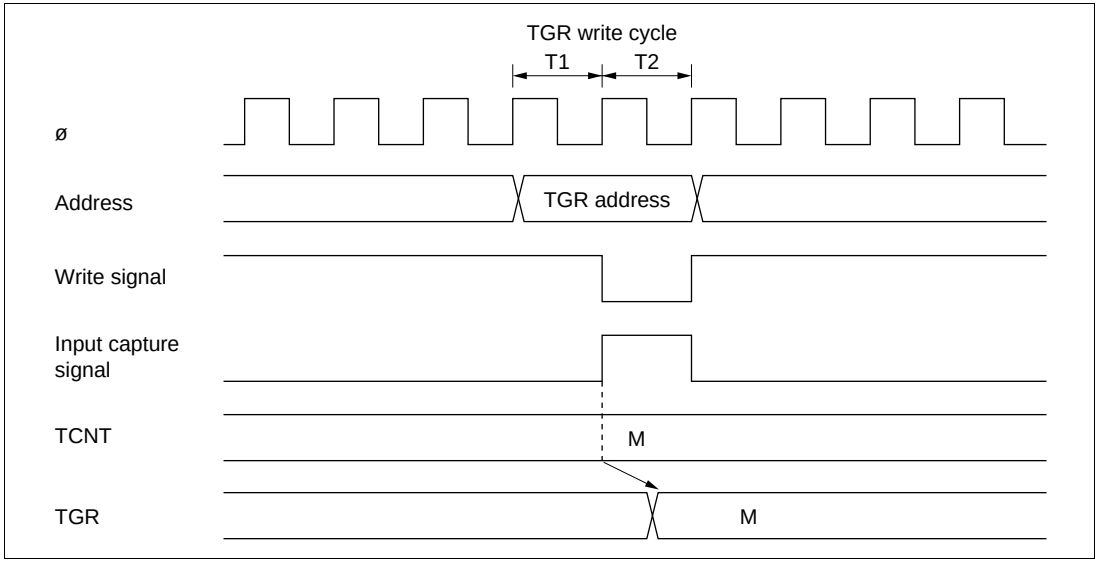


Figure 10-50 Contention between TGR Write and Input Capture

Contention between Buffer Register Write and Input Capture: If the input capture signal is generated in the T2 state of a buffer 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.

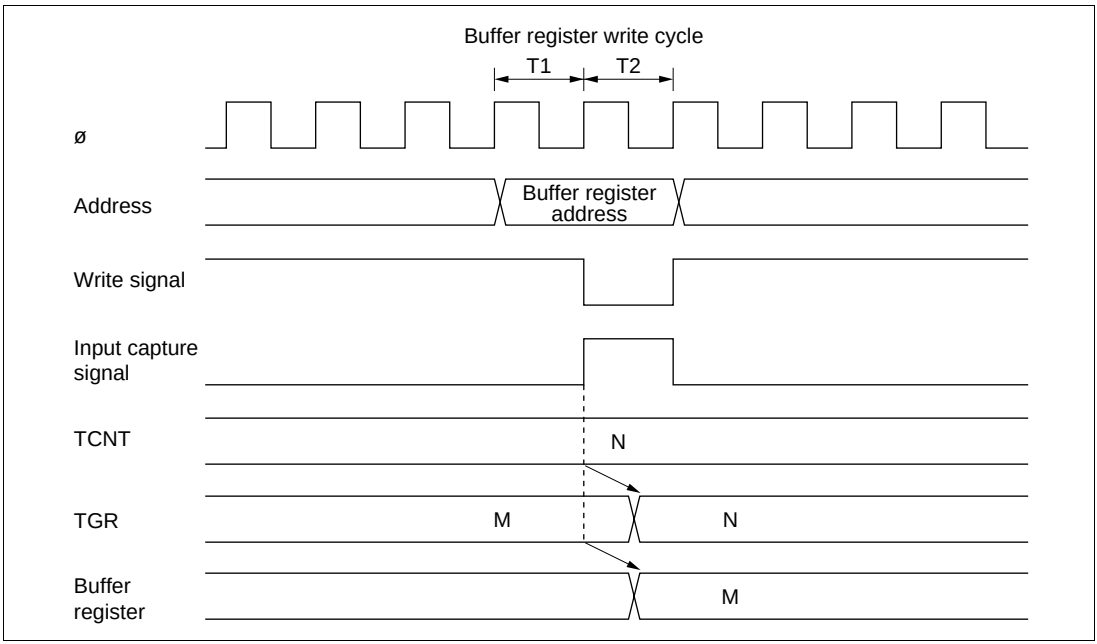


Figure 10-51 Contention between Buffer Register Write and Input Capture

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.

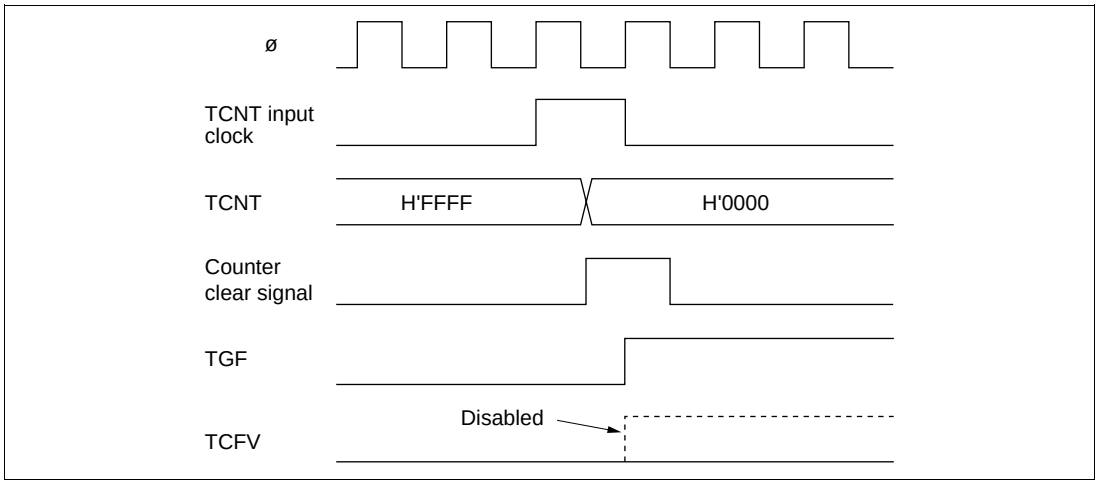


Figure 10-52 Contention between Overflow and Counter Clearing

Contention between TCNT Write and Overflow/Underflow: If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set .

Figure 10-53 shows the operation timing when there is contention between TCNT write and overflow.

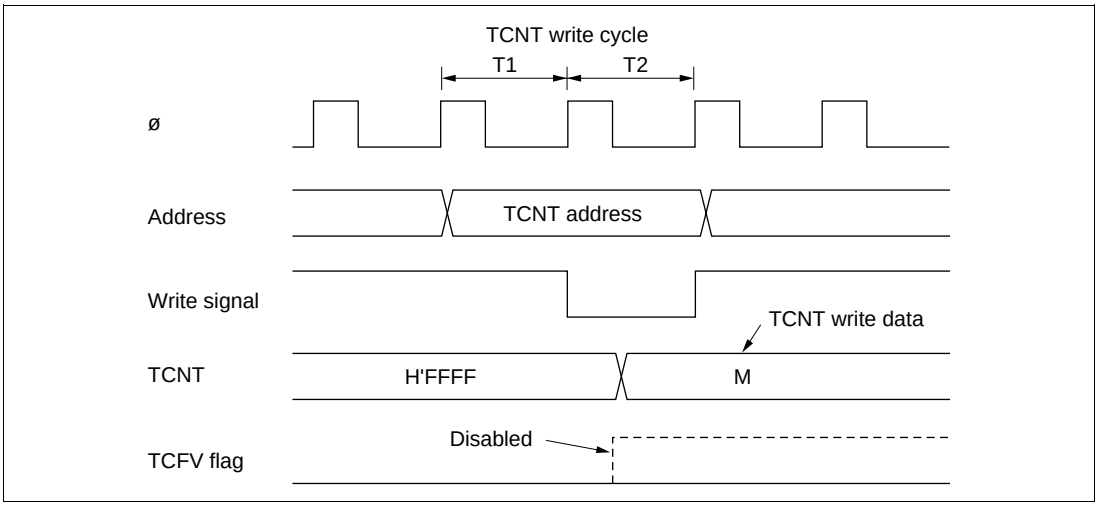


Figure 10-53 Contention between TCNT Write and Overflow

Multiplexing of I/O Pins: In the H8S/2214, 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.

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, DTC activation source, or DMAC activation source. Interrupts should therefore be disabled before entering module stop mode.

Section 11 Watchdog Timer (WDT)

11.1 Overview

The H8S/2214 has an on-chip watchdog timer/watch timer with one channel. The watchdog timer can generate an internal interrupt or an internal reset signal if a system crash prevents the CPU from writing to the counter, allowing it to overflow.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer mode, an interval timer interrupt is generated each time the counter overflows.

11.1.1 Features

WDT features are listed below.

- Switchable between watchdog timer mode and interval timer mode
- Internal reset or internal interrupt generated when watchdog timer mode
  Choice of whether or not an internal reset (power-on reset or manual reset selectable) is effected when the counter overflows
- Interrupt generation in interval timer mode
  — An interval timer interrupt is generated when the counter overflows
- Choice of 8 counter input clocks
  — Maximum WDT interval: system clock period × 131072 × 256

11.1.2 Block Diagram

Figure 11-1 shows block diagrams of WDT.

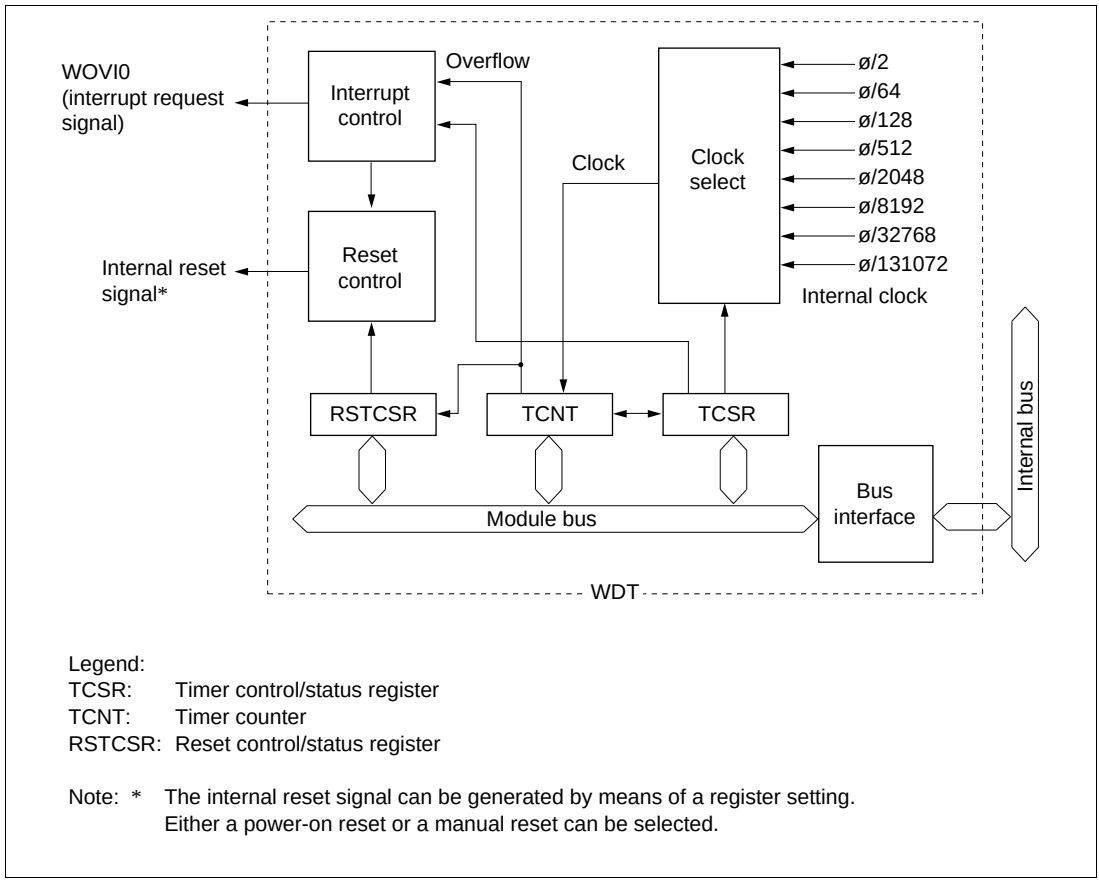


Figure 11-1 Block Diagram of WDT

11.1.3 Register Configuration

The WDT has three registers, as summarized in table 11-1. These registers control clock selection, WDT mode switching, the reset signal, etc.

Table 11-1 WDT Registers

				Address*1	
Name	**Abbreviation**	**R/W**	**Initial Value**	**Write*2**	**Read**
Timer control/status register	TCSR0	R/(W)*3	H'00	H'FF74	H'FF74
Timer counter	TCNT0	R/W	H'00	H'FF74	H'FF75
Reset control/status register	RSTCSR0	R/(W)*3	H'1F	H'FF76	H'FF77

Notes: *1 Lower 16 bits of the address.
*2 For details of write operations, see section 11.2.4, Notes on Register Access.
*3 Only 0 can be written in bit 7, to clear the flag.

11.2 Register Descriptions

11.2.1 Timer Counter (TCNT)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT is an 8-bit readable/writable* up-counter.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from the internal clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (changes from H'FF to H'00), the OVF flag in TCSR is set to 1.

TCNT is initialized to H'00 by a reset, in hardware standby mode, or when the TME bit is cleared to 0. It is not initialized in software standby mode.

Note: * TCNT is write-protected by a password to prevent accidental overwriting. For details see section 11.2.4, Notes on Register Access.

11.2.2 Timer Control/Status Register (TCSR)

Bit	:	7	6	5	4	3	2	1	0
		OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	:	0	0	0	1	1	0	0	0
R/W	:	R/(W)*	R/W	R/W	—	—	R/W	R/W	R/W

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

TCSR is an 8-bit readable/writable* register. Its functions include selecting the clock source to be input to TCNT, and the timer mode.

TCSR0 is initialized to H'18 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Note: * TCSR is write-protected by a password to prevent accidental overwriting. For details see section 11.2.4, Notes on Register Access.

Bit 7—Overflow Flag (OVF): A status flag that indicates that TCNT has overflowed from H'FF to H'00.

Bit 7

OVF	Description	
0	[Clearing condition]	(Initial value)
	*Read TCSR when OVF = 1, then write 0 in OVFA	
1	[Setting condition]	
	When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.	

Note: * When the interval timer interrupt is disabled and OVF is polled, read the state of OVF = 1 twice or more.

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether the WDT is used as a watchdog timer or interval timer. If WDT is used in watchdog timer mode, it can generate a reset when TCNT overflows. If WDT is used in interval timer mode, it generates a WOVI interrupt request to the CPU when TCNT overflows.

Bit 6

WT/$\overline{IT}$	Description	
0	Interval timer mode: Interval timer interrupt (WOVI) request is sent to CPU when TCNT overflows	(Initial value)
1	Watchdog timer mode: Internal reset can be selected when TCNT overflows*	

Note: * For details of the case where TCNT overflows in watchdog timer mode, see section 11.2.3, Reset Control/Status Register (RSTCSR).

Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted.

Bit 5

TME	Description	
0	TCNT is initialized to H'00 and count operation is halted	(Initial value)
1	TCNT counts	

WDT0 TCSR bits 4 and 3—Reserved: These bits cannot be modified and are always read as 1.

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select an internal clock source, obtained by dividing the system clock (ø) for input to TCNT.

Bit 2	Bit 1	Bit 0	Description	
CKS2	CKS1	CKS0	Clock	Overflow Period* (when ø = 10 MHz)
0	0	0	ø/2 (Initial value)	51.2 µs
		1	ø/64	1.6 ms
	1	0	ø/128	3.2 ms
		1	ø/512	13.2 ms
1	0	0	ø/2048	52.4 ms
		1	ø/8192	209.8 ms
	1	0	ø/32768	838.8 ms
		1	ø/131072	3.36 s

Note: * The overflow period is the time from when TCNT starts counting up from H'00 until overflow occurs.

11.2.3 Reset Control/Status Register (RSTCSR)

Bit	:	7	6	5	4	3	2	1	0
		WOVF	RSTE	RSTS	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
R/W	:	R/(W)*	R/W	R/W	—	—	—	—	—

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

RSTCSR is an 8-bit readable/writable* register that 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 $\overline{\text{RES}}$ pin, but not by the internal reset signal caused by a WDT overflow.

Note: * RSTCSR is write-protected by a password to prevent accidental overwriting. For details see section 11.2.4, Notes on Register Access.

Bit 7—Watchdog Overflow Flag (WOVF): Indicates that TCNT has overflowed (from H'FF to H'00) during watchdog timer operation. This bit is not set in interval timer mode.

Bit 7

WOVF	Description	
0	[Clearing condition]	(Initial value)
	Cleared by reading RSTCSR when WOVF = 1, then writing 0 to WOVF	
1	[Setting condition]	
	When TCNT overflows (from H'FF to H'00) in watchdog timer mode	

Bit 6—Reset Enable (RSTE): Specifies whether or not an internal reset signal is generated if TCNT overflows in watchdog timer mode.

Bit 6

RSTE	Description	
0	No internal reset when TCNT overflows*	(Initial value)
1	Internal reset is generated when TCNT overflows	

Note: * The chip is not reset internally, but TCNT and TCSR in WDT0 are reset.

Bit 5—Reset Select (RSTS): Selects the type of internal reset generated if TCNT overflows in watchdog timer mode.

For details of the types of resets, see section 4, Exception Handling.

Bit 5

RSTS	Description	
0	Power-on reset	(Initial value)
1	Manual reset	

Bits 4 to 0—Reserved: These bits cannot be modified and are always read as 1.

11.2.4 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 and TCSR: These registers must be written to by a word transfer instruction. They cannot be written to with byte transfer instructions.

Figure 11-2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. For a write to TCNT, the upper byte of the written word must contain H'5A and the lower byte must contain the write data. For a write to TCSR, the upper byte of the written word must contain H'A5 and the lower byte must contain the write data. This transfers the write data from the lower byte to TCNT or TCSR.

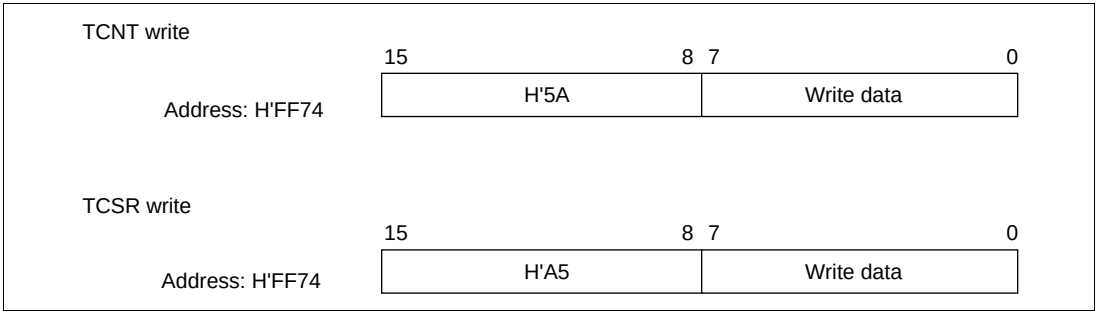


Figure 11-2 Format of Data Written to TCNT and TCSR (Example of WDT0)

Writing to RSTCSR: RSTCSR must be written to by a word transfer to address H'FF76. It cannot be written to with byte instructions.

Figure 11-3 shows the format of data written to RSTCSR. The method of writing 0 to the WOVF bit differs from that for writing to the RSTE and RSTS bits.

To write 0 to the WOVF bit, the upper byte of the written word must contain H'A5 and the lower byte must contain H'00. This clears the WOVF bit to 0, but has no effect on the RSTE and RSTS bits. To write to the RSTE and RSTS bits, the upper byte must contain H'5A and the lower byte must contain the write data. This writes the values in bits 6 and 5 of the lower byte into the RSTE and RSTS bits, but has no effect on the WOVF bit.

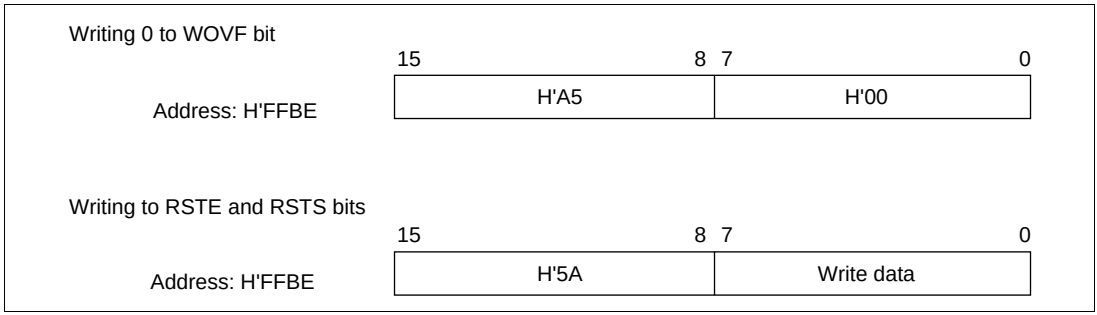


Figure 11-3 Format of Data Written to RSTCSR (Example of WDT0)

Reading TCNT, TCSR, and RSTCSR: These registers are read in the same way as other registers. The read addresses are H'FF74 for TCSR, H'FF75 for TCNT, and H'FF77 for RSTCSR.

11.3 Operation

11.3.1 Watchdog Timer Operation

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ and TME bits in TCSR to 1. Software must prevent TCNT overflows by rewriting the TCNT value (normally by writing H'00) before overflow occurs. This ensures that TCNT does not overflow while the system is operating normally.

In this way, TCNT will not overflow while the system is operating normally, but if TCNT is not rewritten and overflows because of a system crash or other error, in the case of WDT, if the RSTE bit in RSTCSR is set to 1 beforehand, a signal is generated that effects an internal chip reset. Either a power-on reset or a manual reset can be selected with the RSTS bit in RSTCSR. The internal reset signal is output for 518 states. This is illustrated in figure 11-4.

If a reset caused by an input signal from the $\overline{RES}$ pin and a reset caused by WDT overflow occur simultaneously, the $\overline{RES}$ pin reset has priority, and the WOVF bit in RSTCSR is cleared to 0.

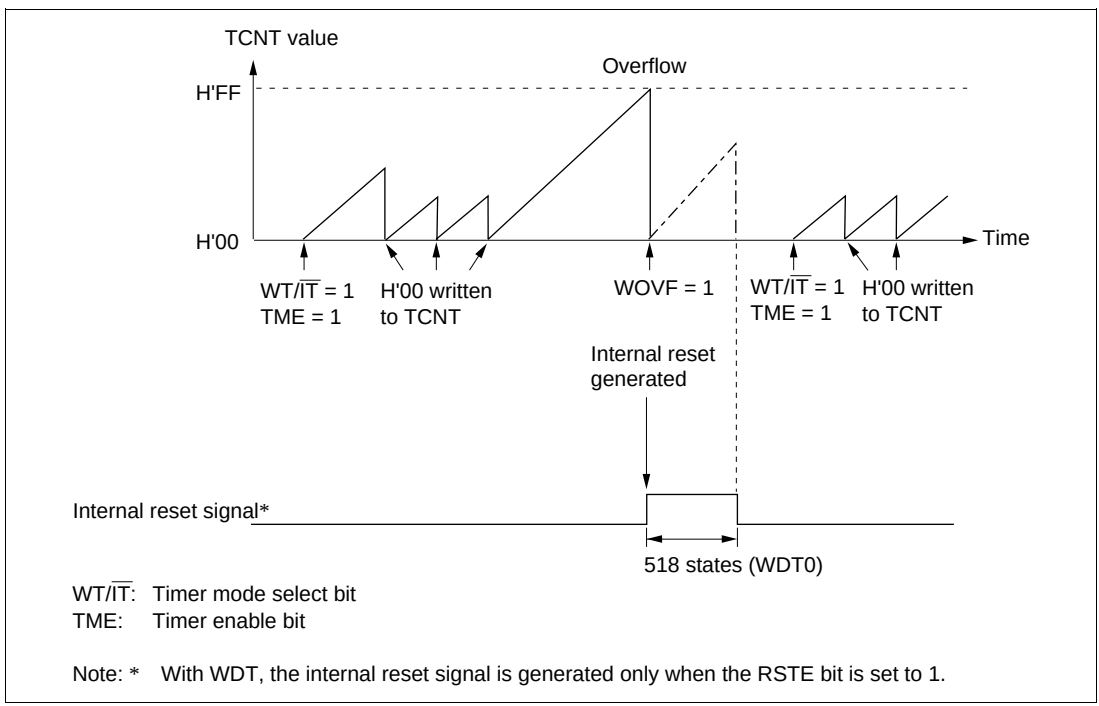


Figure 11-4 Operation in Watchdog Timer Mode

11.3.2 Interval Timer Operation

To use the WDT as an interval timer, clear the WT/$\overline{IT}$ bit in TCSR to 0 and set the TME bit to 1. An interval timer interrupt (WOVI) is generated each time TCNT overflows, provided that the WDT is operating as an interval timer, as shown in figure 11-5. This function can be used to generate interrupt requests at regular intervals.

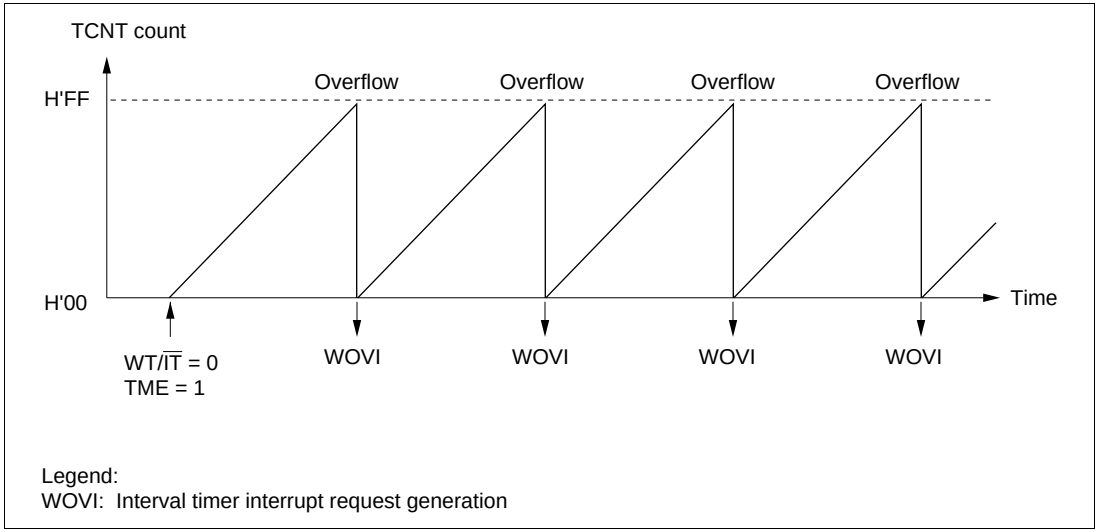


Figure 11-5 Operation in Interval Timer Mode

11.3.3 Timing of Setting of Overflow Flag (OVF)

The OVF flag is set to 1 if TCNT overflows during interval timer operation. At the same time, an interval timer interrupt (WOVI) is requested. This timing is shown in figure11-6.

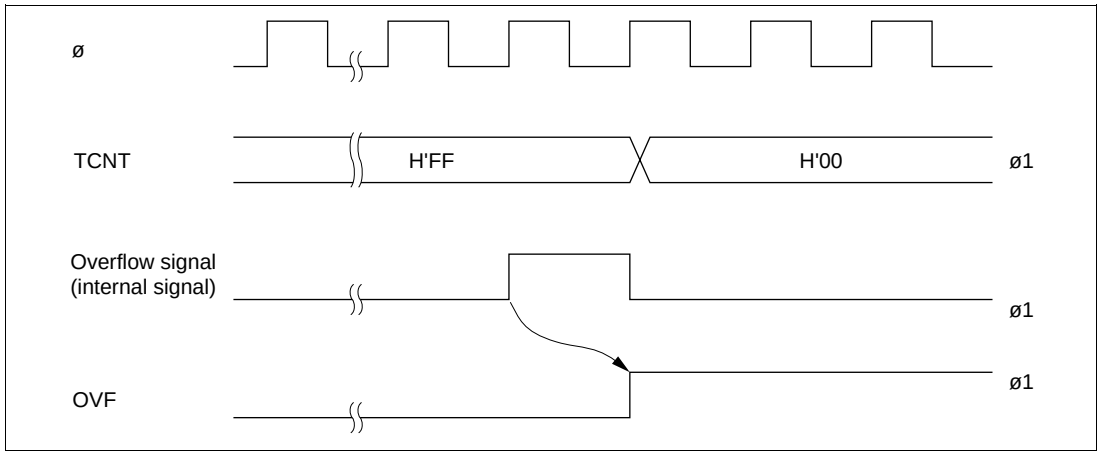


Figure 11-6 Timing of OVF Setting

11.3.4 Timing of Setting of Watchdog Timer Overflow Flag (WOVF)

With WDT, the WOVF bit in RSTCSR is set to 1 if TCNT overflows in watchdog timer mode. If TCNT overflows while the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated for the entire chip. This timing is illustrated in figure 11-7.

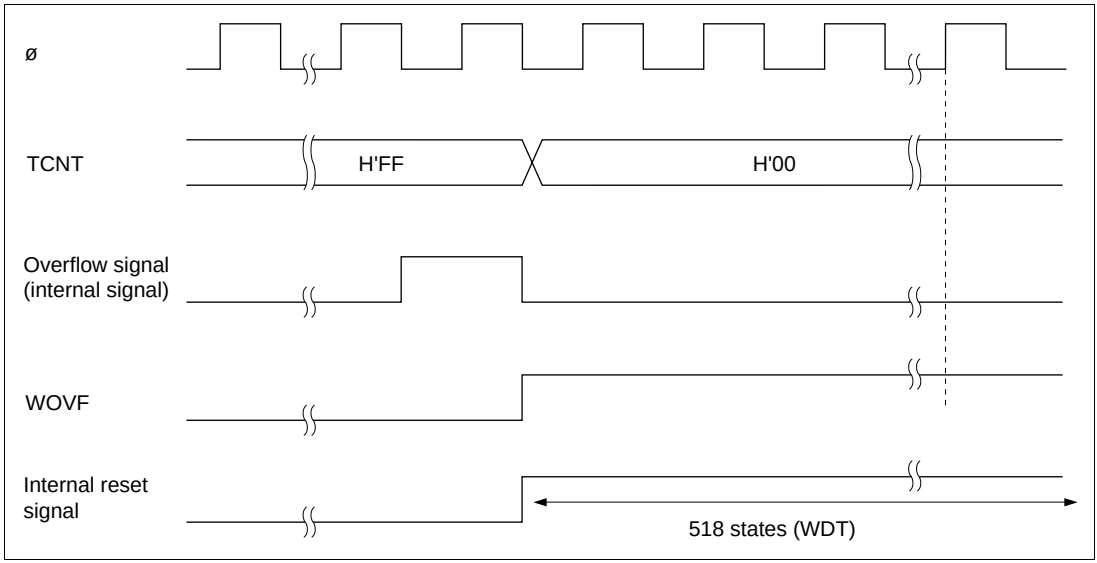


Figure 11-7 Timing of WOVF Setting

11.4 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.

11.5 Usage Notes

11.5.1 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 11-8 shows this operation.

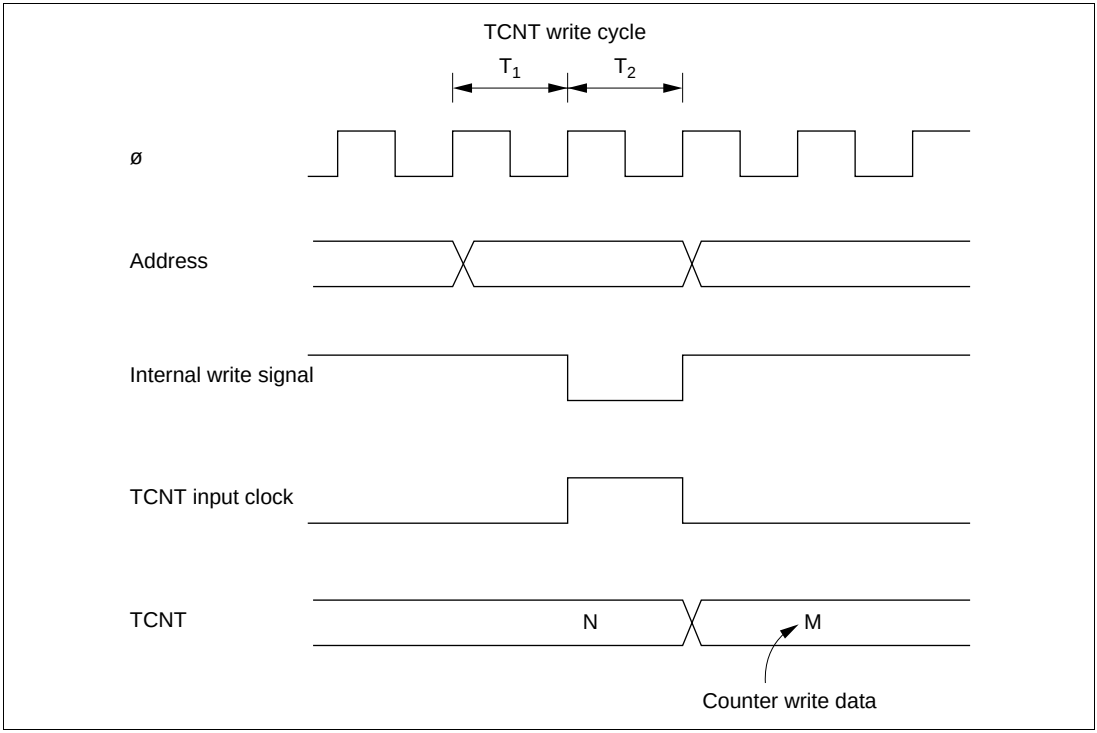


Figure 11-8 Contention between TCNT Write and Increment

11.5.2 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.

11.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, or vice versa, 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.

11.5.4 Internal Reset in Watchdog Timer Mode

If the RSTE bit is cleared to 0 in watchdog timer mode, the chip will not be reset internally if TCNT overflows, but TCNT and TCSR in WDT will be reset.

TCNT, TCSR, and RSTCR cannot be written to for a 132-state interval after overflow occurs, and a read of the WOVF flag is not recognized during this time. It is therefore necessary to wait for 132 states after overflow occurs before writing 0 to the WOVF flag to clear it.

Section 12 Serial Communication Interface (SCI)

12.1 Overview

The H8S/2214 is equipped with mutually independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. A function is also provided for serial communication between processors (multiprocessor communication function).

SCI0 allows a choice of 720 kbps, 460.784 kbps, or 115.192 kbps at 16 MHz operation.

12.1.1 Features

SCI features are listed below.

- Choice of asynchronous or clocked synchronous serial communication mode

  Asynchronous mode

  — Serial data communication executed using asynchronous system in which synchronization is achieved character by character

    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 multiprocessor communication function is provided that enables serial data communication with a number of processors

  — Choice of 12 serial data transfer formats

    Data length : 7 or 8 bits

    Stop bit length : 1 or 2 bits

    Parity : Even, odd, or none

    Multiprocessor bit : 1 or 0

  — 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

  — Average transfer rate generator (SCI0): 720 kbps, 460.784 kbps, or 115.192 kbps can be selected at 16 MHz

  — A transfer rate clock can be input from the TPU (SCI0)

  Clocked Synchronous mode

  — Serial data communication synchronized with a clock

Serial data communication can be carried out with other chips that have a synchronous communication function

— One serial data transfer format

Data length : 8 bits

— Receive error detection : Overrun errors detected

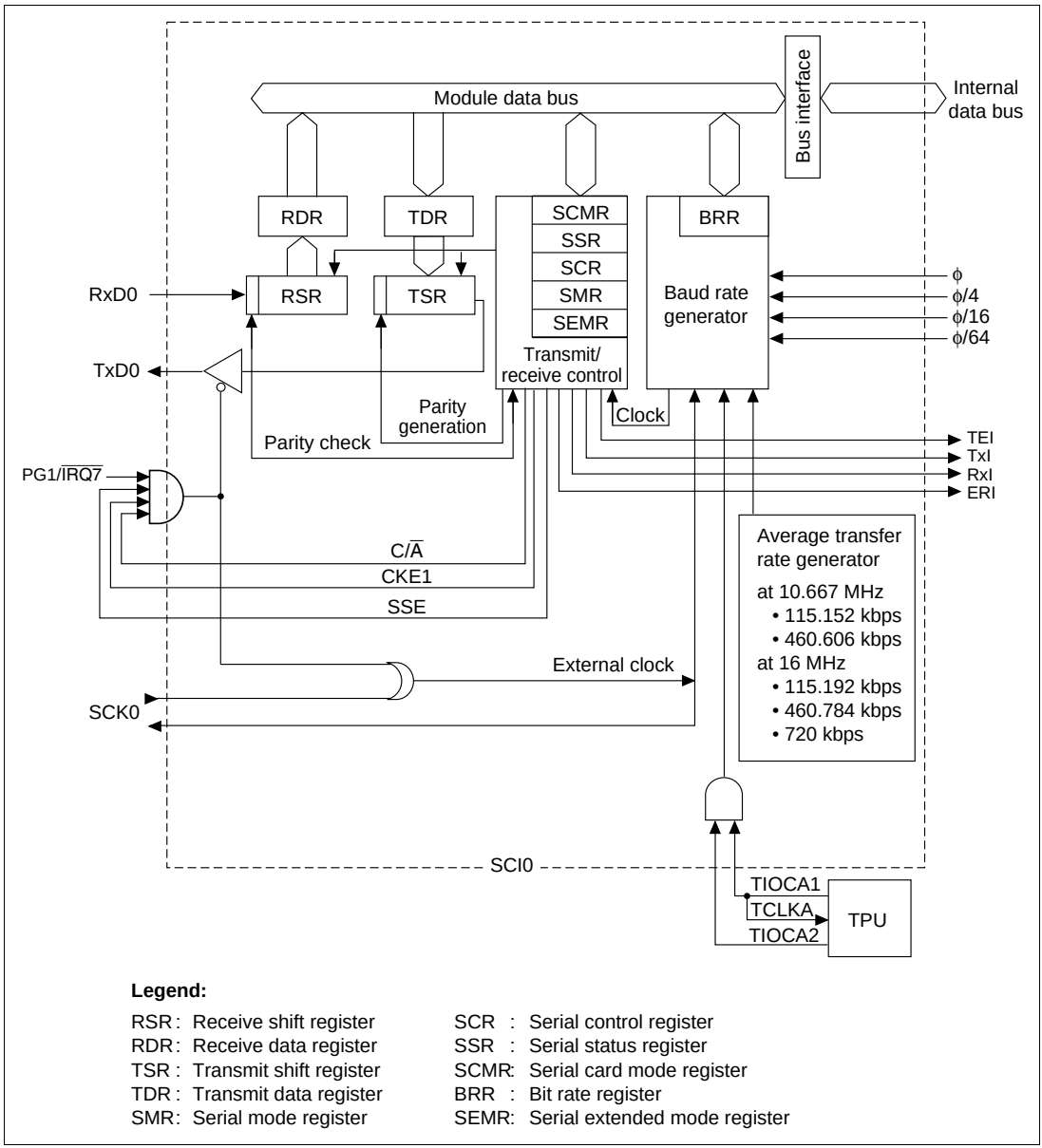
— SCI select function (SCI0: TxD0 = high-impedance and SCK0 = fixed high-level input can be selected when $\overline{\text{IRQ7}}$ = 1)

- 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
- Choice of LSB-first or MSB-first transfer
  - — Can be selected regardless of the communication mode* (except in the case of asynchronous mode 7-bit data)

    Note: * Descriptions in this section refer to LSB-first transfer.
- On-chip baud rate generator allows any bit rate to be selected
- Choice of serial clock source: internal clock from baud rate generator or external clock from SCK pin
- Four interrupt sources
  - — Four interrupt sources — transmit-data-empty, transmit-end, receive-data-full, and receive error — that can issue requests independently
  - — The transmit-data-empty interrupt and receive data full interrupts can activate the data transfer controller (DTC) or DMA controller (DMAC) to execute data transfer
- Module stop mode can be set
  - — As the initial setting, SCI operation is halted. Register access is enabled by exiting module stop mode.

12.1.2 Block Diagram

Figures 12-1 and 12-2 show block diagrams of the SCI.



Legend:

RSR : Receive shift register	SCR : Serial control register
RDR : Receive data register	SSR : Serial status register
TSR : Transmit shift register	SCMR: Serial card mode register
TDR : Transmit data register	BRR : Bit rate register
SMR : Serial mode register	SEMR: Serial extended mode register

Figure 12-1 Block Diagram of SCI0

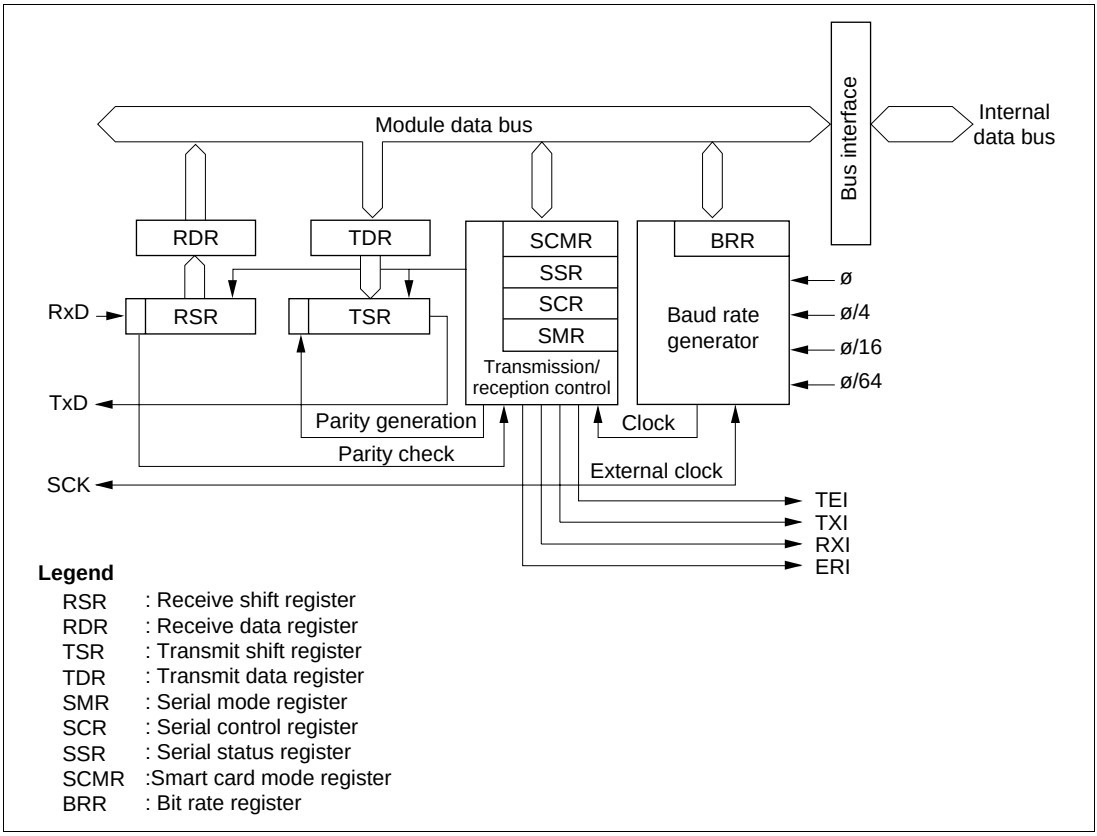


Figure 12-2 Block Diagram of SCI1 and SCI2

12.1.3 Pin Configuration

Table 12-1 shows the serial pins for each SCI channel.

Table 12-1 SCI Pins

Channel	Pin Name	Symbol	I/O	Function
0	Serial clock pin 0	SCK0	I/O	SCI0 clock input/output
	Receive data pin 0	RxD0	Input	SCI0 receive data input
	Transmit data pin 0	TxD0	Output	SCI0 transmit data output
1	Serial clock pin 1	SCK1	I/O	SCI1 clock input/output
	Receive data pin 1	RxD1	Input	SCI1 receive data input
	Transmit data pin 1	TxD1	Output	SCI1 transmit data output
2	Serial clock pin 2	SCK2	I/O	SCI2 clock input/output
	Receive data pin 2	RxD2	Input	SCI2 receive data input
	Transmit data pin 2	TxD2	Output	SCI2 transmit data output

Note: Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

12.1.4 Register Configuration

The SCI has the internal registers shown in table 12-2. These registers are used to specify asynchronous mode or clocked synchronous mode, the data format , and the bit rate, and to control transmitter/receiver.

Table 12-2 SCI Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
0	Serial mode register 0	SMR0	R/W	H'00	H'FF78
	Bit rate register 0	BRR0	R/W	H'FF	H'FF79
	Serial control register 0	SCR0	R/W	H'00	H'FF7A
	Transmit data register 0	TDR0	R/W	H'FF	H'FF7B
	Serial status register 0	SSR0	R/(W)*[2]	H'84	H'FF7C
	Receive data register 0	RDR0	R	H'00	H'FF7D
	Smart card mode register 0	SCMR0	R/W	H'F2	H'FF7E
	Serial expansion mode register 0	SEMR0	R/W	H'00	H'FDF8
1	Serial mode register 1	SMR1	R/W	H'00	H'FF80
	Bit rate register 1	BRR1	R/W	H'FF	H'FF81
	Serial control register 1	SCR1	R/W	H'00	H'FF82
	Transmit data register 1	TDR1	R/W	H'FF	H'FF83
	Serial status register 1	SSR1	R/(W)*[2]	H'84	H'FF84
	Receive data register 1	RDR1	R	H'00	H'FF85
	Smart card mode register 1	SCMR1	R/W	H'F2	H'FF86
2	Serial mode register 2	SMR2	R/W	H'00	H'FF88
	Bit rate register 2	BRR2	R/W	H'FF	H'FF89
	Serial control register 2	SCR2	R/W	H'00	H'FF8A
	Transmit data register 2	TDR2	R/W	H'FF	H'FF8B
	Serial status register 2	SSR2	R/(W)*[2]	H'84	H'FF8C
	Receive data register 2	RDR2	R	H'00	H'FF8D
	Smart card mode register 2	SCMR2	R/W	H'F2	H'FF8E
All	Module stop control register B	MSTPCRB	R/W	H'FF	H'FDE9

Notes: *1 Lower 16 bits of the address.
*2 Can only be written with 0 for flag clearing.

12.2 Register Descriptions

12.2.1 Receive Shift Register (RSR)

Bit	:	7	6	5	4	3	2	1	0
R/W	:	—	—	—	—	—	—	—	—

RSR is a register used to receive serial data.

The SCI sets serial data input from the RxD pin in RSR in the order received, starting with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to RDR automatically.

RSR cannot be directly read or written to by the CPU.

12.2.2 Receive Data Register (RDR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R

RDR is a register that stores received serial 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, and completes the receive operation. 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.

RDR is a read-only register, and cannot be written to by the CPU.

RDR is initialized to H'00 by a reset, in standby mode, watch mode, subactive mode, and subsleep mode or module stop mode.

12.2.3 Transmit Shift Register (TSR)

Bit	:	7	6	5	4	3	2	1	0
R/W	:	—	—	—	—	—	—	—	—

TSR is a register used to transmit 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 with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from TDR to TSR, and transmission started, automatically. However, data transfer from TDR to TSR is not performed if the TDRE bit in SSR is set to 1.

TSR cannot be directly read or written to by the CPU.

12.2.4 Transmit Data Register (TDR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TDR is an 8-bit register that stores data for serial transmission.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts serial transmission. Continuous serial transmission can be carried out by writing the next transmit data to TDR during serial transmission of the data in TSR.

TDR can be read or written to by the CPU at all times.

TDR is initialized to H'FF by a reset, in standby mode, watch mode, subactive mode, and subsleep mode or module stop mode.

12.2.5 Serial Mode Register (SMR)

Bit	:	7	6	5	4	3	2	1	0
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SMR is an 8-bit register used to set the SCI's serial transfer format and select the baud rate generator clock source.

SMR can be read or written to by the CPU at all times.

SMR is initialized to H'00 by a reset and in hardware standby mode. It retains its previous state in module stop mode, software standby mode, watch mode, subactive mode, and subsleep mode.

Bit 7—Communication Mode (C/$\overline{A}$): Selects asynchronous mode or clocked synchronous mode as the SCI operating mode.

Bit 7

C/$\overline{A}$	Description	
0	Asynchronous mode	(Initial value)
1	Clocked synchronous mode	

Bit 6—Character Length (CHR): Selects 7 or 8 bits as the data length in asynchronous mode. In clocked synchronous mode, a fixed data length of 8 bits is used regardless of the CHR setting.

Bit 6

CHR	Description	
0	8-bit data	(Initial value)
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it is not possible to choose between LSB-first or MSB-first transfer.

Bit 5—Parity Enable (PE): In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking in reception. In clocked synchronous mode with a multiprocessor format, parity bit addition and checking is not performed, regardless of the PE bit setting.

Bit 5

PE	Description	
0	Parity bit addition and checking disabled	(Initial value)
1	Parity bit addition and checking enabled*	

Note:* When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

Bit 4—Parity Mode (O/$\overline{E}$): Selects either even or odd parity for use in parity addition and checking.

The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking, in asynchronous mode. The O/$\overline{E}$ bit setting is invalid in clocked synchronous mode, when parity addition and checking is disabled in asynchronous mode, and when a multiprocessor format is used.

Bit 4

O/$\overline{E}$	Description	
0	Even parity*[1]	(Initial value)
1	Odd parity*[2]	

Notes: *1 When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is even.

*2 When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.

Bit 3—Stop Bit Length (STOP): Selects 1 or 2 bits as the stop bit length in asynchronous mode. The STOP bits setting is only valid in asynchronous mode. If clocked synchronous mode is set the STOP bit setting is invalid since stop bits are not added.

Bit 3	
STOP	**Description**
0	1 stop bit: In transmission, a single 1 bit (stop bit) is added to the end of a transmit character before it is sent. (Initial value)
1	2 stop bits: In transmission, two 1 bits (stop bits) are added to the end of a transmit character before it is sent.

In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next transmit character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, the PE bit and O/$\overline{E}$ bit parity settings are invalid. The MP bit setting is only valid in asynchronous mode; it is invalid in clocked synchronous mode.

For details of the multiprocessor communication function, see section 12.3.3, Multiprocessor Communication Function.

Bit 2	
MP	**Description**
0	Multiprocessor function disabled (Initial value)
1	Multiprocessor format selected

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source for the baud rate generator. The clock source can be selected from ø, ø/4, ø/16, and ø/64, according to the setting of bits CKS1 and CKS0.

For the relation between the clock source, the bit rate register setting, and the baud rate, see section 12.2.8, Bit Rate Register.

Bit 1	Bit 0		
CKS1	**CKS0**	**Description**	
0	0	ø clock	(Initial value)
	1	ø/4 clock	
1	0	ø/16 clock	
	1	ø/64 clock	

12.2.6 Serial Control Register (SCR)

Bit	:	7	6	5	4	3	2	1	0
		TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR is a register that performs enabling or disabling of SCI transfer operations, serial clock output in asynchronous mode, and interrupt requests, and selection of the serial clock source.

SCR can be read or written to by the CPU at all times.

SCR is initialized to H'00 by a reset and in hardware standby mode. It retains its previous state in module stop mode, software standby mode, watch mode, subactive mode, and subsleep mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit data empty interrupt (TXI) request generation when serial transmit data is transferred from TDR to TSR and the TDRE flag in SSR is set to 1.

Bit 7

TIE	Description
0	Transmit data empty interrupt (TXI) requests disabled* (Initial value)
1	Transmit data empty interrupt (TXI) requests enabled

Note: * 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.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables receive data full interrupt (RXI) request and receive error interrupt (ERI) request generation when serial receive data is transferred from RSR to RDR and the RDRF flag in SSR is set to 1.

Bit 6

RIE	Description
0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled* (Initial value)
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Note: * 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 clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of serial transmission by the SCI.

Bit 5

TE	Description	
0	Transmission disabled*1	(Initial value)
1	Transmission enabled*2	

Notes: *1 The TDRE flag in SSR is fixed at 1.

*2 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.

Bit 4—Receive Enable (RE): Enables or disables the start of serial reception by the SCI.

Bit 4

RE	Description	
0	Reception disabled*1	(Initial value)
1	Reception enabled*2	

Notes: *1 Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.

*2 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.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE bit setting is only valid in asynchronous mode when the MP bit in SMR is set to 1.

The MPIE bit setting is invalid in clocked synchronous mode or when the MP bit is cleared to 0.

Bit 3

MPIE	Description	
0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received	(Initial value)
1	Multiprocessor interrupts enabled* Receive interrupt (RXI) requests, receive error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.	

Note: * When receive data including MPB = 0 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.

Bit 2—Transmit End Interrupt Enable (TEIE): Enables or disables transmit end interrupt (TEI) request generation when there is no valid transmit data in TDR in MSB data transmission.

Bit 2

TEIE	Description	
0	Transmit end interrupt (TEI) request disabled*	(Initial value)
1	Transmit end interrupt (TEI) request enabled*	

Note: * 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 clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the SCI clock source and enable or disable clock output from the SCK pin. The combination of the CKE1 and CKE0 bits determines whether the SCK pin functions as an I/O port, the serial clock output pin, or the serial clock input pin.

The setting of the CKE0 bit, however, is only valid for internal clock operation (CKE1 = 0) in asynchronous mode. The CKE0 bit setting is invalid in clocked synchronous mode, and in the case of external clock operation (CKE1 = 1). Note that the SCI's operating mode must be decided using SMR after setting the CKE1 and CKE0 bits.

For details of clock source selection, see table 12.9 in section 12.3, Operation.

Bit 1	Bit 0		
CKE1	**CKE0**	**Description**	
0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port*1
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*2
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*3
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*3
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input

Notes: *1 Initial value
*2 Outputs a clock of the same frequency as the bit rate.
*3 Inputs a clock with a frequency 16 times the bit rate.

12.2.7 Serial Status Register (SSR)

Bit :	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value :	1	0	0	0	0	1	0	0
R/W :	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

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

SSR is an 8-bit register containing status flags that indicate the operating status of the SCI, and multiprocessor bits.

SSR can be read or written to by the CPU at all times. However, 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER. Also note that in order to clear these flags they must be read as 1 beforehand. The TEND flag and MPB flag are read-only flags and cannot be modified.

SSR is initialized to H'84 by a reset, in standby mode, watch mode, subactive mode, and subsleep mode or module stop mode.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that data has been transferred from TDR to TSR and the next serial data can be written to TDR.

Bit 7

TDRE	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Bit 6—Receive Data Register Full (RDRF): Indicates that the received data is stored in RDR.

Bit 6

RDRF	Description
0	[Clearing conditions] (Initial value) • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and reads data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Note: RDR and the RDRF flag are not affected and retain their previous values when an error is detected during reception or when the RE bit in SCR is cleared to 0.
If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that an overrun error occurred during reception, causing abnormal termination.

Bit 5

ORER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1*2

Notes: *1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
*2 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.

Bit 4—Framing Error (FER): Indicates that a framing error occurred during reception in asynchronous mode, causing abnormal termination.

Bit 4

FER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data when reception ends, and the stop bit is 0 *2

Notes: *1 The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*2 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.

Bit 3—Parity Error (PER): Indicates that a parity error occurred during reception using parity addition in asynchronous mode, causing abnormal termination.

Bit 3

PER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR*2

Notes: *1 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*2 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.

Bit 2—Transmit End (TEND): Indicates that there is no valid data in TDR when the last bit of the transmit character is sent, and transmission has been ended.

The TEND flag is read-only and cannot be modified.

Bit 2

TEND	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Bit 1—Multiprocessor Bit (MPB): When reception is performed using multiprocessor format in asynchronous mode, MPB stores the multiprocessor bit in the receive data.

MPB is a read-only bit, and cannot be modified.

Bit 1

MPB	Description
0	[Clearing condition] (Initial value)* When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Note: * Retains its previous state when the RE bit in SCR is cleared to 0 with multiprocessor format.

Bit 0—Multiprocessor Bit Transfer (MPBT): When transmission is performed using multiprocessor format in asynchronous mode, MPBT stores the multiprocessor bit to be added to the transmit data.

The MPBT bit setting is invalid when multiprocessor format is not used, when not transmitting, and in clocked synchronous mode.

Bit 0

MPBT	Description
0	Data with a 0 multiprocessor bit is transmitted (Initial value)
1	Data with a 1 multiprocessor bit is transmitted

12.2.8 Bit Rate Register (BRR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BRR is an 8-bit register that sets the serial transfer bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SMR.

BRR can be read or written to by the CPU at all times.

BRR is initialized to H'FF by a reset and in hardware standby mode. It retains its previous state in module stop mode, software standby mode, watch mode, subactive mode, and subsleep mode.

As baud rate generator control is performed independently for each channel, different values can be set for each channel.

Table 12-3 shows sample BRR settings in asynchronous mode, and table 12-4 shows sample BRR settings in clocked synchronous mode.

Table 12-3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	ø = 2 MHz			ø = 2.097152 MHz			ø = 2.4576 MHz			ø = 3 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19	–2.34
9600	—	—	—	0	6	–2.48	0	7	0.00	0	9	–2.34
19200	—	—	—	—	—	—	0	3	0.00	0	4	–2.34
31250	0	1	0.00	—	—	—	—	—	—	0	2	0.00
38400	—	—	—	—	—	—	0	1	0.00	—	—	—

	ø = 3.6864 MHz			ø = 4 MHz			ø = 4.9152 MHz			ø = 5 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	—	—	—	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4	0.00
38400	0	2	0.00	—	—	—	0	3	0.00	0	3	1.73

	ø = 6 MHz			ø = 6.144 MHz			ø = 7.3728 MHz			ø = 8 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	—	—	—	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	—	—	—

	ø = 9.8304 MHz			ø = 10 MHz			ø = 12 MHz			ø = 12.288 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bit/s)	ø = 14 MHz			ø = 14.7456 MHz			ø = 16 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03
150	2	181	0.13	2	191	0.00	2	207	0.16
300	2	90	0.13	2	95	0.00	2	103	0.16
600	1	181	0.13	1	191	0.00	1	207	0.16
1200	1	90	0.13	1	95	0.00	1	103	0.16
2400	0	181	0.13	0	191	0.00	0	207	0.16
4800	0	90	0.13	0	95	0.00	0	103	0.16
9600	0	45	–0.93	0	47	0.00	0	51	0.16
19200	0	22	–0.93	0	23	0.00	0	25	0.16
31250	0	13	0.00	0	14	–1.70	0	15	0.00
38400	—	—	—	0	11	0.00	0	12	0.16

Table 12-4 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate	ø = 2 MHz		ø = 4 MHz		ø = 6 MHz		ø = 8 MHz		ø = 10 MHz		ø = 16 MHz	
(bit/s)	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—								
250	2	124	2	249			3	124	—	—	3	249
500	1	249	2	124			2	249	—	—	3	124
1 k	1	124	1	249			2	124	—	—	2	249
2.5 k	0	199	1	99	1	149	1	199	1	249	2	99
5 k	0	99	0	199	1	74	1	99	1	124	1	199
10 k	0	49	0	99	0	149	0	199	0	249	1	99
25 k	0	19	0	39	0	59	0	79	0	99	0	159
50 k	0	9	0	19	0	29	0	39	0	49	0	79
100 k	0	4	0	9	0	14	0	19	0	24	0	39
250 k	0	1	0	3	0	5	0	7	0	9	0	15
500 k	0	0*	0	1	0	2	0	3	0	4	0	7
1 M			0	0*			0	1			0	3
2.5 M									0	0*		
5 M												

Note: As far as possible, the setting should be made so that the error is no more than 1%.

Legend

Blank : Cannot be set.

— : Can be set, but there will be a degree of error.

* : Continuous transfer is not possible.

The BRR setting is found from the following formulas.

Asynchronous mode:

$$N = \frac{ø}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clocked synchronous mode:

$$N = \frac{ø}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ø: Operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ø	0	0
1	ø/4	0	1
2	ø/16	1	0
3	ø/64	1	1

The bit rate error in asynchronous mode is found from the following formula:

$$\text{Error (\%)} = \{ \frac{ø \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \} \times 100$$

Table 12-5 shows the maximum bit rate for each frequency in asynchronous mode. Tables 12-6 and 12-7 show the maximum bit rates with external clock input.

When the ABCS bit in SCI0's serial expansion mode register 0 (SEMR0) is set to 1 in asynchronous mode, the maximum bit rates are twice those shown in tables 12-5 and 12-6.

Table 12-5 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ø (MHz)	Maximum Bit Rate (bit/s)	n	N
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
5	156250	0	0
6	187500	0	0
6.144	192000	0	0
7.3728	230400	0	0
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

Table 12-6 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
5	1.2500	78125
6	1.5000	93750
6.144	1.5360	96000
7.3728	1.8432	115200
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

Table 12-7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)

ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
2	0.3333	333333.3
4	0.6667	666666.7
6	1.0000	1000000.0
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

12.2.9 Smart Card Mode Register (SCMR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	—
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

SCMR selects LSB-first or MSB-first by means of bit SDIR. Except in the case of asynchronous mode 7-bit data, LSB-first or MSB-first can be selected regardless of the serial communication mode

SCMR is initialized to H'F2 by a reset and in hardware standby mode. It retains its previous state in module stop mode, software standby mode, watch mode, subactive mode, and subsleep mode.

Bits 7 to 4—Reserved: Read-only bits, always read as 1.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

This bit is valid when 8-bit data is used as the transmit/receive format.

Bit 3

SDIR	Description	
0	TDR contents are transmitted LSB-first	(Initial value)
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

Bit 2—Smart Card Data Invert (SINV): Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit(s): parity bit inversion requires inversion of the O/$\overline{E}$ bit in SMR.

Bit 2

SINV	Description	
0	TDR contents are transmitted without modification Receive data is stored in RDR without modification	(Initial value)
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form	

Bit 1—Reserved: Read-only bit, always read as 1.

Bit 0—Reserved: This bit can be read or written to, but only 0 should be written.

12.2.10 Serial Extended Mode Register 0 (SEMR0)

Bit	:	7	6	5	4	3	2	1	0
		SSE	—	—	—	ABCS	ACS2	ACS1	ACS0
Initial value	:	0	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	R/W	—	—	—	R/W	R/W	R/W	R/W

SEMR0 is an 8-bit register that extends the functions of SCI0.

SEMR0 enables selection of the SCI0 select function in synchronous mode, base clock setting in asynchronous mode, and also clock source selection and automatic transfer rate setting.

SEMR0 is initialized to H'00 by a reset and in hardware standby mode. It retains its previous state in module stop mode and software standby mode.

Bit 7—SCI0 Select Enable (SSE): Allows selection of the SCI0 select function when an external clock is input in synchronous mode. When the SCI0 select function is enabled, if 1 is input to the PG1/IRQ7 pin, TxD0 output goes to the high-impedance state, SCK0 input is fixed high inside the chip, and SCI0 data transmission/reception is halted.

The SSE setting is valid when external clock input is used (CKE1 = 1 in SCR) in synchronous mode (C/$\overline{A}$ = 1 in SMR). When an internal clock is selected (CKE1 = 0 in SCR) in synchronous mode, or when the chip is in asynchronous mode (C/$\overline{A}$ = 0 in SMR), the SCI0 select function is disabled even if SSE is set to 1.

Bit 7

SSE	Description	
0	SCI0 select function disabled	(Initial value)
1	SCI0 select function enabled When PG1/IRQ7 pin input = 1, TxD0 output goes to high-impedance state and SCK0 clock input is fixed high	

Bits 6 to 4—Reserved: Write 0 to these bits.

Bit 3—Asynchronous Base Clock Select (ABCS): Selects the 1-bit-interval base clock in asynchronous mode.

The ABCS setting is valid in asynchronous mode (C/$\overline{A}$ = 0 in SMR). It is invalid in synchronous mode (C/$\overline{A}$ = 1 in SMR).

Bit 3

ABCS	Description	
0	SCI0 operates on base clock with frequency of 16 times transfer rate	(Initial value)
1	SCI0 operates on base clock with frequency of 8 times transfer rate	

Bits 2 to 0—Asynchronous Clock Source Select 2 to 0 (ACS2 to ACS0): These bits select the clock source in asynchronous mode.

When an average transfer rate is selected, the base clock is set automatically regardless of the ABCS value. Note that average transfer rates are not supported for operating frequencies other than 10.667 MHz and 16 MHz.

The setting in bits ACS2 to ACS0 is valid when external clock input is used (CKE1 = 1 in SCR) in asynchronous mode (C/$\overline{A}$ = 0 in SMR). The setting in ACS2 to ACS0 is invalid when an internal clock is selected (CKE1 = 0 in SCR) in asynchronous mode, or when the chip is in synchronous mode (C/$\overline{A}$ = 1 in SMR).

Bit 2	Bit 1	Bit 0	
ACS2	**ACS1**	**ACS0**	**Description**
0	0	0	External clock input (Initial value)
		1	115.152 kbps average transfer rate (for ø = 10.667 MHz only) is selected (SCI0 operates on base clock with frequency of 16 times transfer rate)
	1	0	460.606 kbps average transfer rate (for ø = 10.667 MHz only) is selected (SCI0 operates on base clock with frequency of 8 times transfer rate)
		1	Reserved
1	0	0	TPU clock input (AND of TIOCA1 and TIOCA2)
		1	115.196 kbps average transfer rate (for ø = 16 MHz only) is selected (SCI0 operates on base clock with frequency of 16 times transfer rate)
	1	0	460.784 kbps average transfer rate (for ø = 16 MHz only) is selected (SCI0 operates on base clock with frequency of 16 times transfer rate)
		1	720 kbps average transfer rate (for ø = 16 MHz only) is selected (SCI0 operates on base clock with frequency of 8 times transfer rate)

Figures 12-3 and 12-4 show examples of the internal base clock when an average transfer rate is selected.

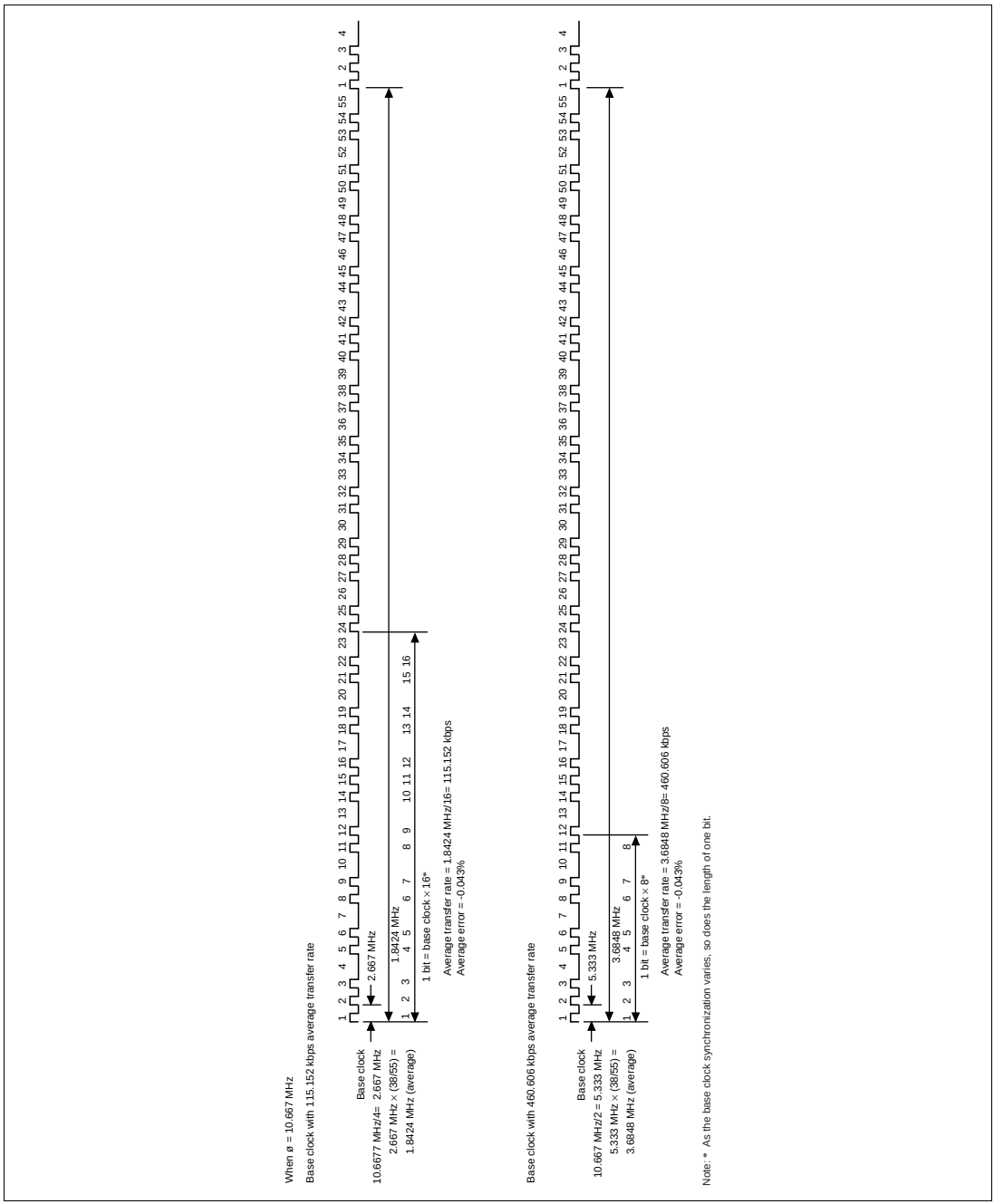


Figure 12-3 Examples of Base Clock when Average Transfer Rate is Selected (1)

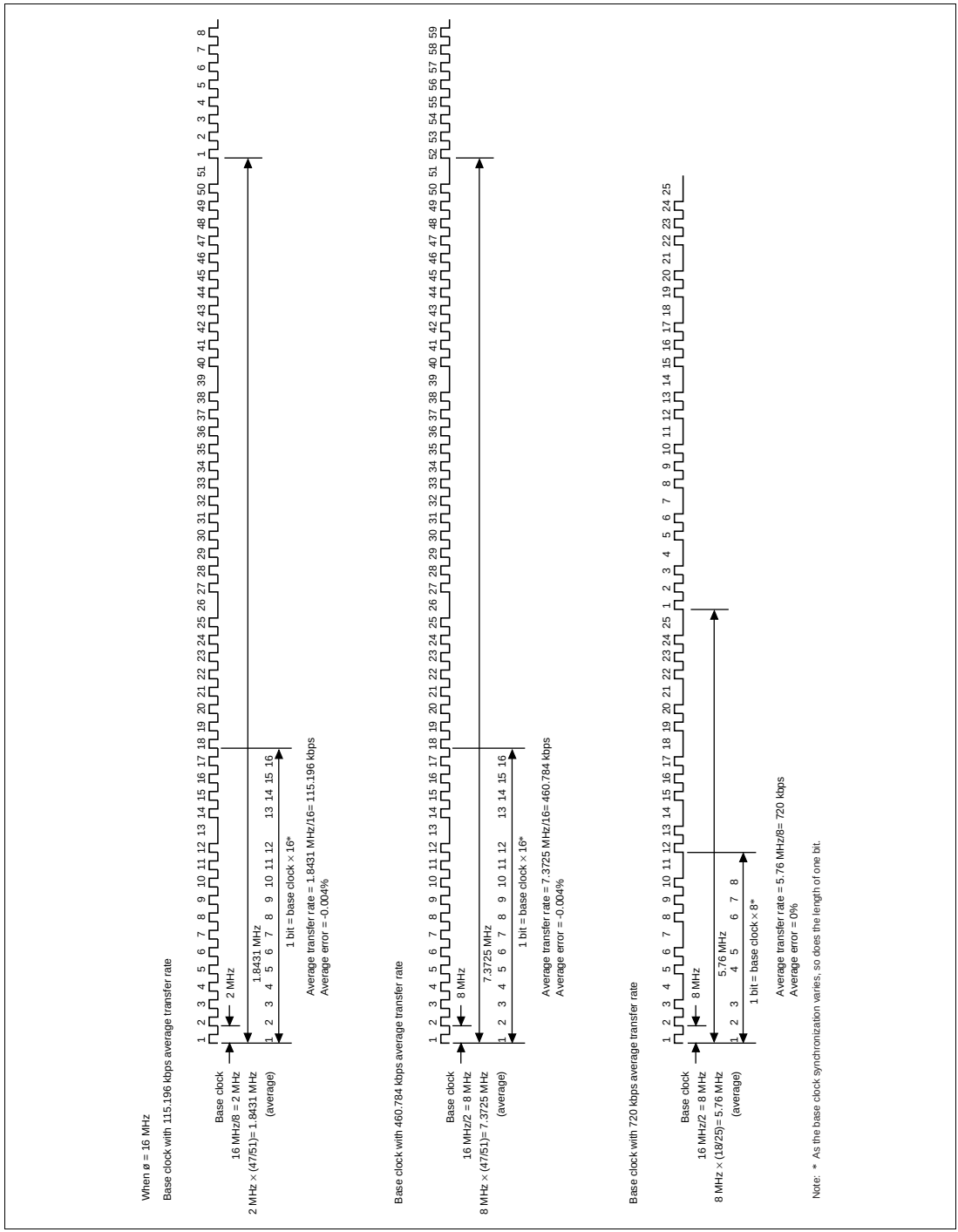


Figure 12-4 Examples of Base Clock when Average Transfer Rate is Selected (2)

12.2.11 Module Stop Control Register B (MSTPCRB)

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRB is an 8-bit readable/writable register that performs module stop mode control.

When one of bits MSTPB7 to MSTPB5 is set to 1, SCI0, SCI1, or SCI2 respectively, stops operation at the end of the bus cycle, and enters module stop mode. For details, see section 17.5, Module Stop Mode.

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

Bit 7—Module Stop (MSTPB7): Specifies the SCI0 module stop mode.

Bit 7

MSTPB7	Description	
0	SCI0 module stop mode is cleared	
1	SCI0 module stop mode is set	(Initial value)

Bit 6—Module Stop (MSTPB6): Specifies the SCI1 module stop mode.

Bit 6

MSTPB6	Description	
0	SCI1 module stop mode is cleared	
1	SCI1 module stop mode is set	(Initial value)

Bit 5—Module Stop (MSTPB5): Specifies the SCI2 module stop mode.

Bit 5

MSTPB5	Description	
0	SCI2 module stop mode is cleared	
1	SCI2 module stop mode is set	(Initial value)

12.3 Operation

12.3.1 Overview

The SCI can carry out serial communication in two modes: asynchronous mode in which synchronization is achieved character by character, and clocked synchronous mode in which synchronization is achieved with clock pulses.

Selection of asynchronous or clocked synchronous mode and the transmission format is made using SMR as shown in table 12-8. The SCI clock is determined by a combination of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 12-9.

Asynchronous Mode

- Data length: Choice of 7 or 8 bits
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bits (the combination of these parameters determines the transfer format and character length)
- Detection of framing, parity, and overrun errors, and breaks, during reception
- Choice of internal or external clock as SCI clock source
  - — When internal clock is selected:

    The SCI operates on the baud rate generator clock and a clock with the same frequency as the bit rate can be output
  - — When external clock is selected:

    A clock with a frequency of 16 times the bit rate must be input (the on-chip baud rate generator is not used)

Clocked Synchronous Mode

- Transfer format: Fixed 8-bit data
- Detection of overrun errors during reception
- Choice of internal or external clock as SCI clock source
  - — When internal clock is selected:

    The SCI operates on the baud rate generator clock and a serial clock is output off-chip
  - — When external clock is selected:

    The on-chip baud rate generator is not used, and the SCI operates on the input serial clock

Table 12-8 SMR Settings and Serial Transfer Format Selection

SMR Settings						SCI Transfer Format			
Bit 7	Bit 6	Bit 2	Bit 5	Bit 3		Data	Multi Processor	Parity	Stop Bit
C/$\overline{A}$	CHR	MP	PE	STOP	Mode	Length	Bit	Bit	Length
0	0	0	0	0	Asynchronous mode	8-bit data	No	No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	1		0	0		7-bit data		No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Yes	No	1 bit
			—	1					2 bits
	1		—	0		7-bit data			1 bit
			—	1					2 bits
1	—	—	—	—	Clocked synchronous mode	8-bit data	No		None

Table 12-9 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Setting			SCI Transmit/Receive Clock	
Bit 7	Bit 1	Bit 0		Clock	
C/$\overline{A}$	CKE1	CKE0	Mode	Source	SCK Pin Function
0	0	0	Asynchronous mode	Internal	SCI does not use SCK pin
		1			Outputs clock with same frequency as bit rate
	1	0		External	Inputs clock with frequency of 16 times the bit rate
		1			
1	0	0	Clocked synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

Table 12-10 SMR0, SCR0, SEMR0 Settings and SCI Clock Source Selection (SCI0 Only)

SMR0	SCR0 Setting		SEMR0 Setting				SCI Transmit/Receive Clock	
Bit 7	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0			
C/$\overline{A}$	CKE1	CKE0	ACS2	ACS1	ACS0	Mode	Clock Source	SCK Pin Function
0	0	0	*	*	*	Asynchronous mode	Internal	SCI does not use SCK pin
		1						Outputs clock with some frequency as bit rate
	1	*	0	0	0		External	Inputs clock with frequency of 16 or 8 times the bit rate
					1		Average transfer rate generator (115.152 kbps at 10.667 MHz)	SCI does not use SCK pin
				1	0		Average transfer rate generator (460.606 kbps at 10.667 MHz)	SCI does not use SCK pin
					1		—	—
			1	0	0		TPU (AND of T10CA1 and T10CA2)	SCI does not use SCK pin
					1		Average transfer rate generator (115.196 kbps at 16 MHz)	SCI does not use SCK pin
				1	0		Average transfer rate generator (460.784 kbps at 16 MHz)	SCI does not use SCK pin
					1		Average transfer rate generator (720 kbps at 16 MHz)	SCI does not use SCK pin

SMR0	SCR0 Setting		SEMR0 Setting				SCI Transmit/Receive Clock	
Bit 7	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0			
C/$\overline{A}$	CKE1	CKE0	ACS2	ACS1	ACS0	Mode	Clock Source	SCK Pin Function
1	0	0	*	*	*	Clocked synchronous mode	Internal	Outputs serial clock
		1						
	1	0					External	Input serial clock
		1						

12.3.2 Operation in Asynchronous Mode

In asynchronous mode, characters are sent or received, each preceded by a start bit indicating the start of communication and stop bits indicating the end of communication. Serial communication is thus carried out with synchronization established on a character-by-character basis.

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 12-5 shows the general format for asynchronous serial communication.

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.

One serial communication character consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level).

In asynchronous mode, the SCI performs synchronization at the falling edge of the start bit in reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16 times the length of one bit, so that the transfer data is latched at the center of each bit.

When the ABCS bit in SEMR0 is set to 1, SCI0 samples the data on the 4th pulse of a clock with a frequency of 8 times the length of one bit.

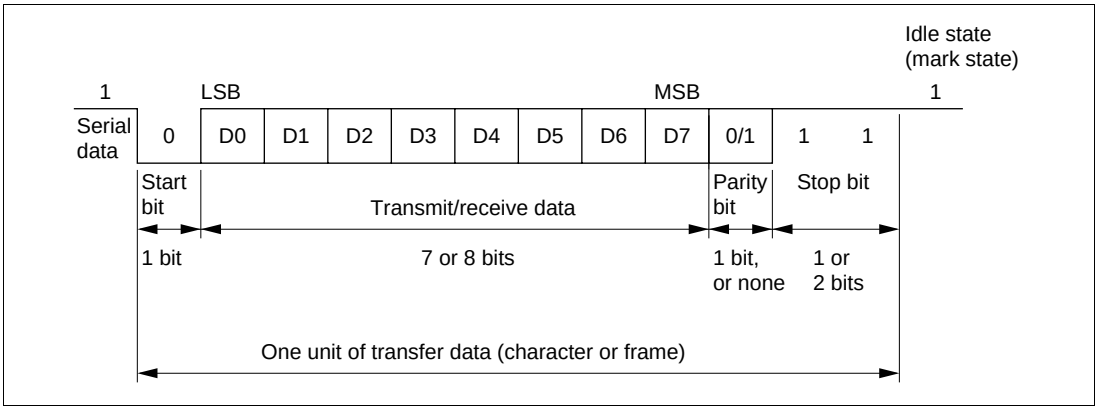


Figure 12-5 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

Data Transfer Format: Table 12-11 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.

Table 12-11 Serial Transfer Formats (Asynchronous Mode)

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

Clock: 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/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 12-9.

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 12-6.

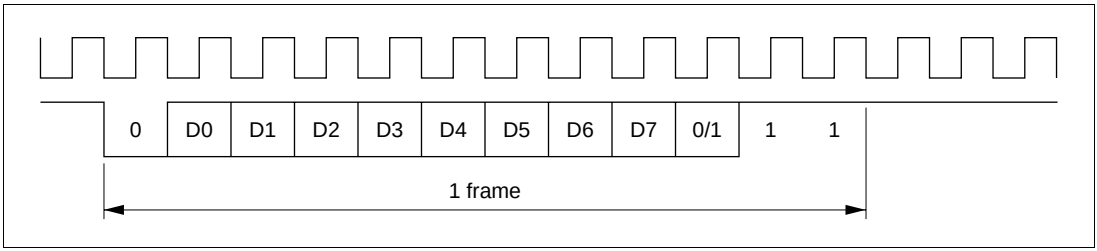


Figure 12-6 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

Data Transfer Operations:

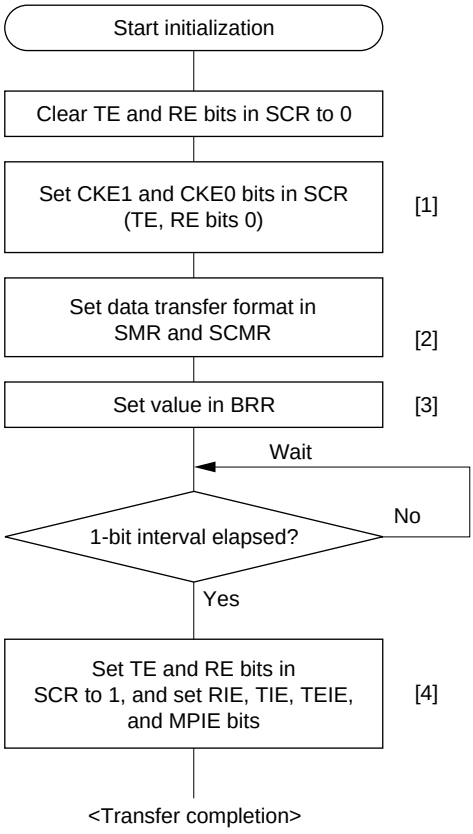
- SCI initialization (asynchronous mode)

  Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

  When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. 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.

  When an external clock is used the clock should not be stopped during operation, including initialization, since operation is uncertain.

Figure 12-7 shows a sample SCI initialization flowchart.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.

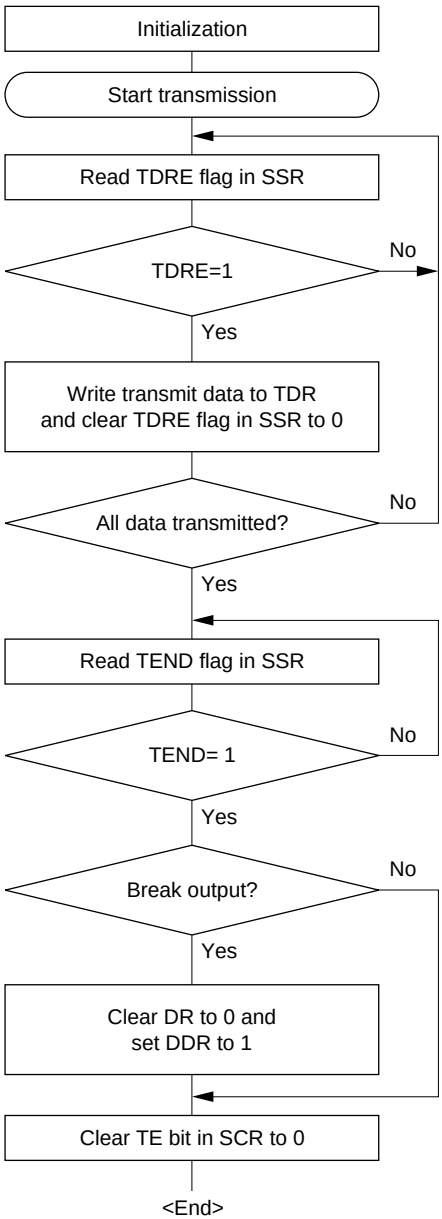
Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 12-7 Sample SCI Initialization Flowchart

- Serial data transmission (asynchronous mode)

  Figure 12-8 shows a sample flowchart for serial transmission.

  The following procedure should be used for serial data transmission.



[1] SCI initialization:
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 DTC is activated by a transmit data empty interrupt (TXI) request, and date 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.

Figure 12-8 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

[1] The SCI monitors the TDRE flag in SSR, and if 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 is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.

The serial transmit data is sent from the TxD pin in the following order.

[a] Start bit:

One 0-bit is output.

[b] Transmit data:

8-bit or 7-bit data is output in LSB-first order.

[c] Parity bit or multiprocessor bit:

One parity bit (even or odd parity), or one multiprocessor bit is output.

A format in which neither a parity bit nor a multiprocessor bit is output can also be selected.

[d] Stop bit(s):

One or two 1-bits (stop bits) are output.

[e] Mark state:

1 is output continuously until the start bit that starts the next transmission is sent.

[3] The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is cleared to 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

If the TDRE flag is set to 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 continuously. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 12-9 shows an example of the operation for transmission in asynchronous mode.

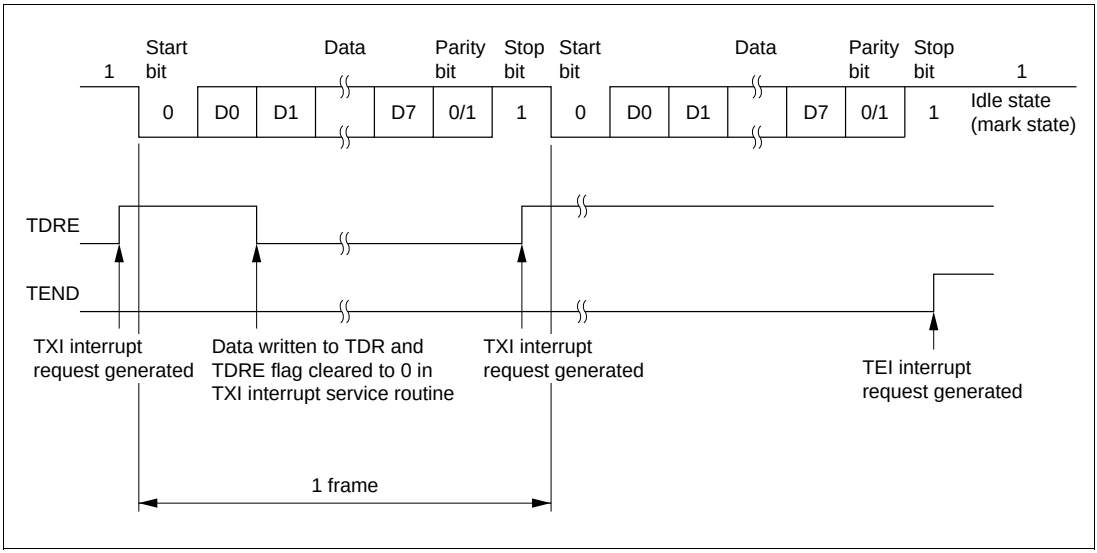
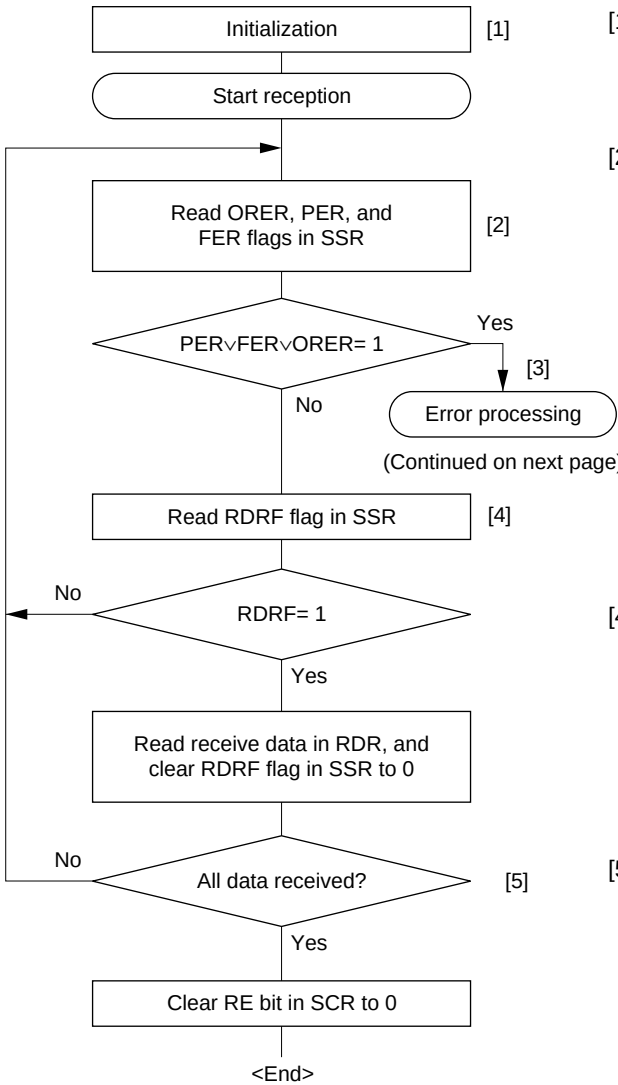


Figure 12-9 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

- Serial data reception (asynchronous mode)

  Figures 12-10 and 12-11 show a sample flowchart for serial reception.

  The following procedure should be used for serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing 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 DTC is activated by an RXI interrupt and the RDR value is read.

Figure 12-10 Sample Serial Reception Data Flowchart (1)

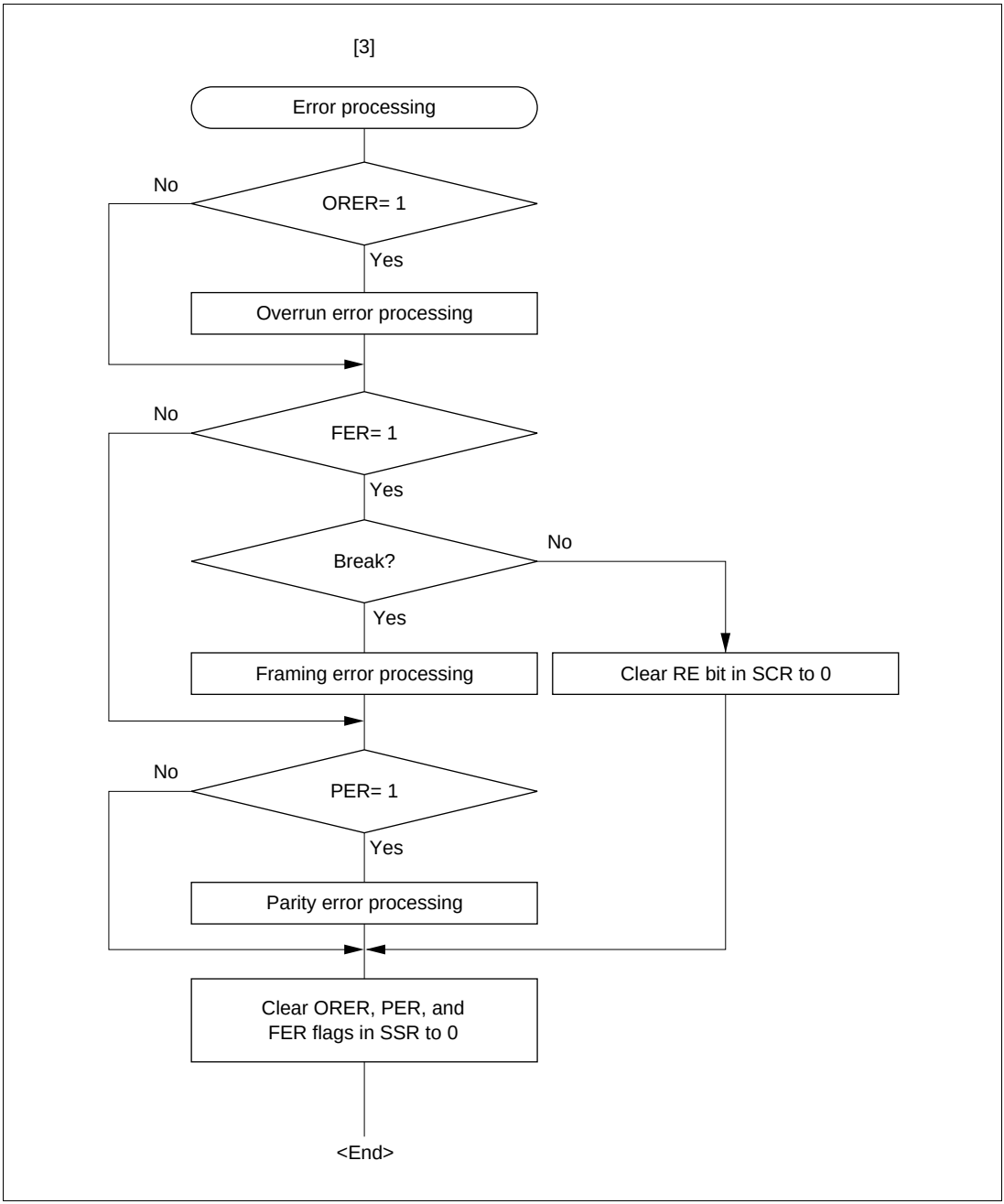


Figure 12-11 Sample Serial Reception Data Flowchart (2)

In serial reception, the SCI operates as described below.

[1] The SCI monitors the transmission line, and if a 0 stop bit is detected, performs internal synchronization and starts reception.

[2] The received data is stored in RSR in LSB-to-MSB order.

[3] The parity bit and stop bit are received.

After receiving these bits, the SCI carries out the following checks.

[a] Parity check:

The SCI checks whether the number of 1 bits in the receive data agrees with the parity (even or odd) set in the O/$\overline{E}$ bit in SMR.

[b] Stop bit check:

The SCI checks whether the stop bit is 1.

If there are two stop bits, only the first is checked.

[c] Status check:

The SCI checks whether the RDRF flag is 0, indicating that the receive data can be transferred from RSR to RDR.

If all the above checks are passed, the RDRF flag is set to 1, and the receive data is stored in RDR.

If a receive error* is detected in the error check, the operation is as shown in table 12-11.

Note: * Subsequent receive operations cannot be performed when a receive error has occurred. Also note that the RDRF flag is not set to 1 in reception, and so the error flags must be cleared to 0.

[4] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data full interrupt (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER, PER, or FER flag changes to 1, a receive error interrupt (ERI) request is generated.

Table 12-12 Receive Errors and Conditions for Occurrence

Receive Error	Abbreviation	Occurrence Condition	Data Transfer
Overrun error	ORER	When the next data reception is completed while the RDRF flag in SSR is set to 1	Receive data is not transferred from RSR to RDR.
Framing error	FER	When the stop bit is 0	Receive data is transferred from RSR to RDR.
Parity error	PER	When the received data differs from the parity (even or odd) set in SMR	Receive data is transferred from RSR to RDR.

Figure 12-12 shows an example of the operation for reception in asynchronous mode.

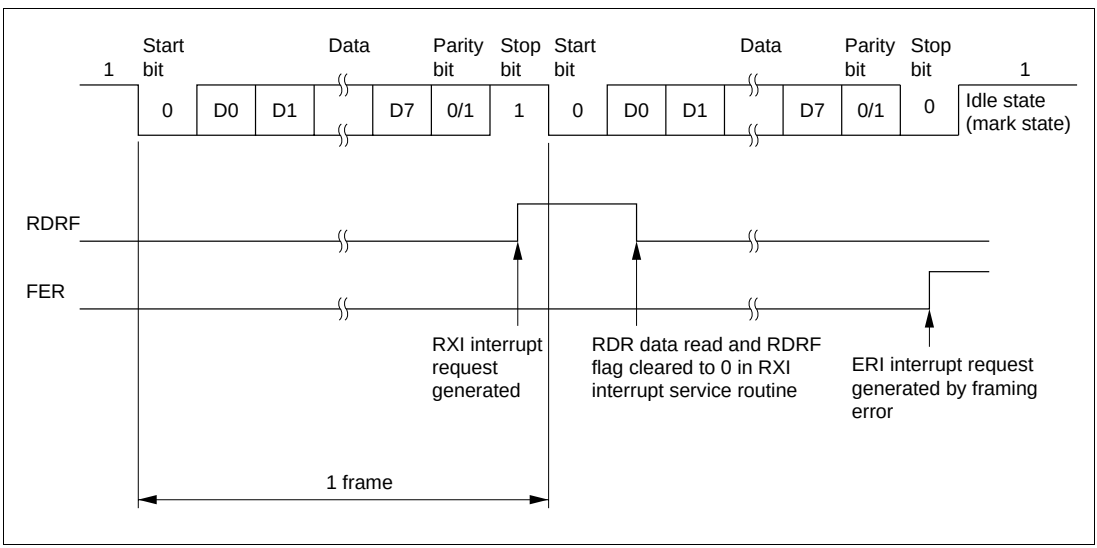


Figure 12-12 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

12.3.3 Multiprocessor Communication Function

The multiprocessor communication function performs serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data, in asynchronous mode. Use of this function enables data transfer to be performed among a number of processors sharing transmission lines.

When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code.

The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station , and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle.

The transmitting station first sends the ID of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added.

The receiving station skips the data until data with a 1 multiprocessor bit is sent.

When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip the data until data with a 1 multiprocessor bit is again received. In this way, data communication is carried out among a number of processors.

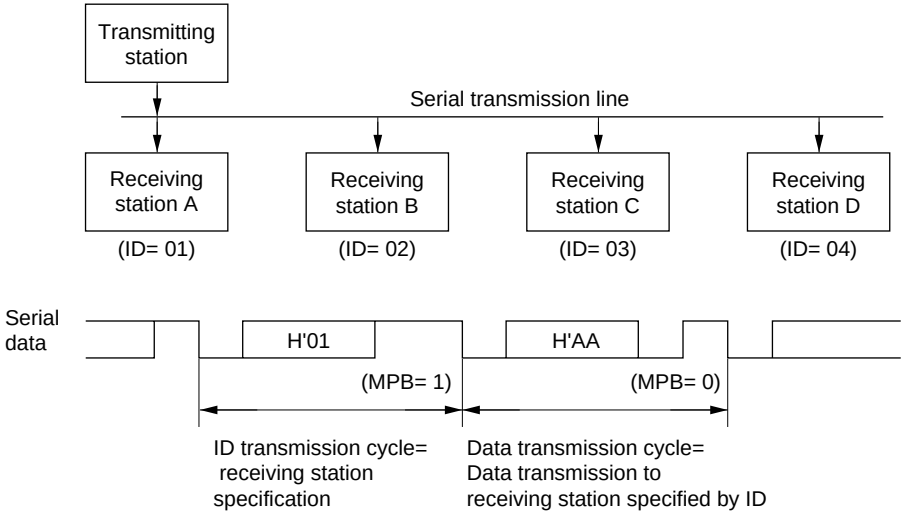
Figure 12-13 shows an example of inter-processor communication using the multiprocessor format.

Data Transfer Format: There are four data transfer formats.

When the multiprocessor format is specified, the parity bit specification is invalid.

For details, see table 12-11.

Clock: See the section on asynchronous mode.

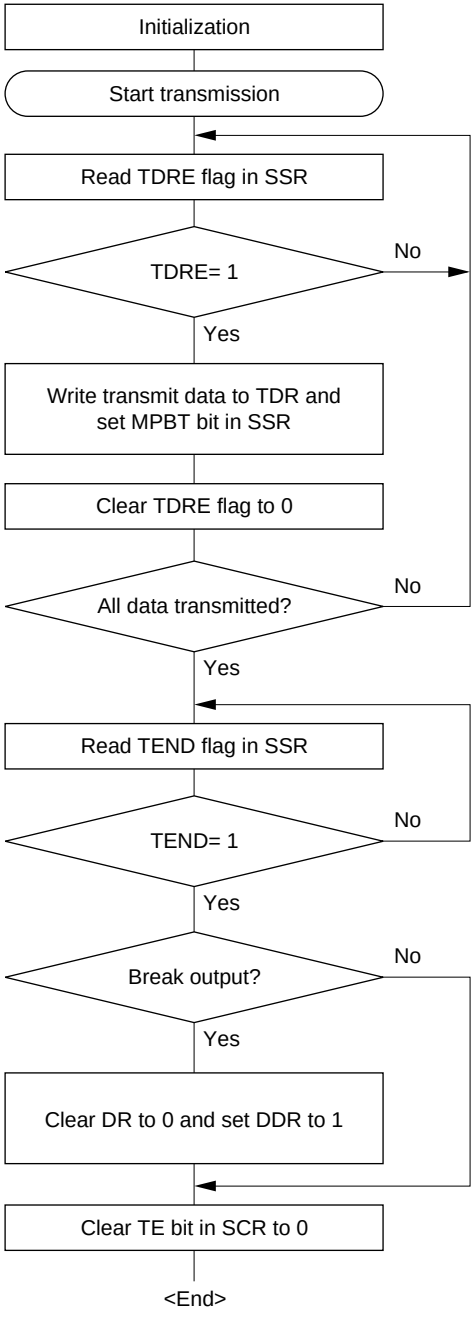


Legend
MPB: Multiprocessor bit

Figure 12-13 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

Data Transfer Operations:

- Multiprocessor serial data transmission
  Figure 12-14 shows a sample flowchart for multiprocessor serial data transmission.
  The following procedure should be used for multiprocessor serial data transmission.



[1] SCI initialization:
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. Set the MPBT bit in SSR to 0 or 1.
Finally, clear the TDRE flag to 0.

[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 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 the port DDR to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 12-14 Sample Multiprocessor Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

[1] The SCI monitors the TDRE flag in SSR, and if 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 transmit data empty interrupt (TXI) is generated.

The serial transmit data is sent from the TxD pin in the following order.

[a] Start bit:

One 0-bit is output.

[b] Transmit data:

8-bit or 7-bit data is output in LSB-first order.

[c] Multiprocessor bit

One multiprocessor bit (MPBT value) is output.

[d] Stop bit(s):

One or two 1-bits (stop bits) are output.

[e] Mark state:

1 is output continuously until the start bit that starts the next transmission is sent.

[3] The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

If the TDRE flag is set to 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 continuously. If the TEIE bit in SCR is set to 1 at this time, a transmission end interrupt (TEI) request is generated.

Figure 12-15 shows an example of SCI operation for transmission using the multiprocessor format.

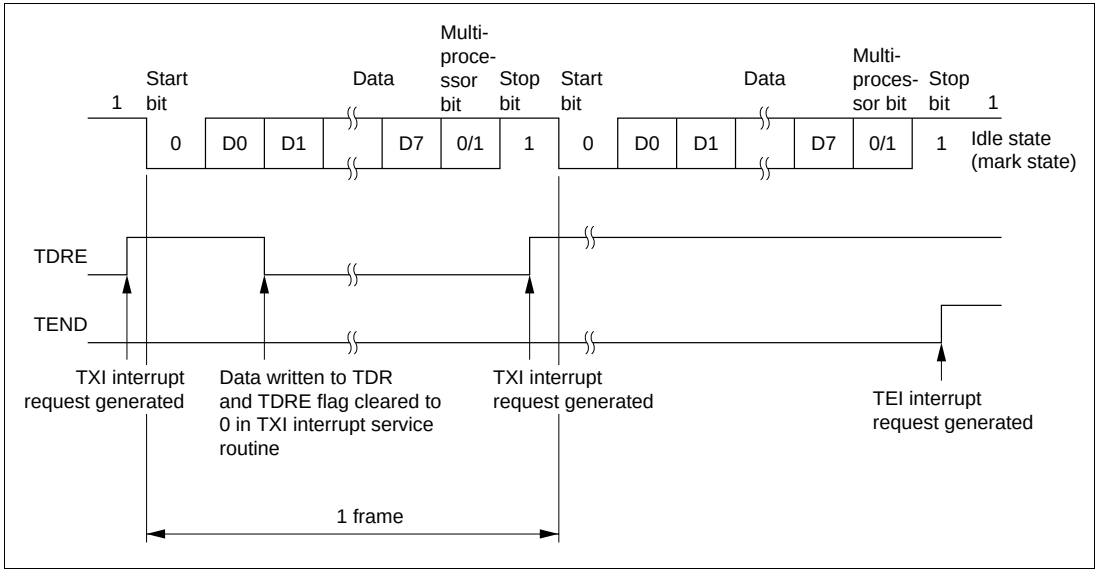
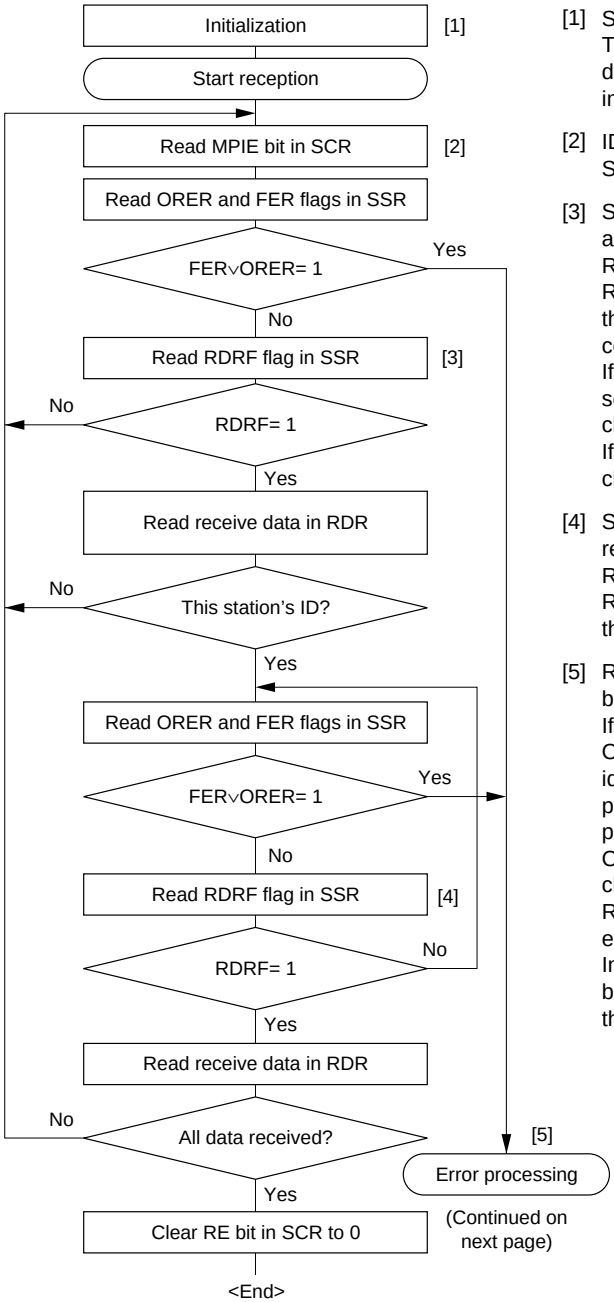


Figure 12-15 Example of SCI Operation in Transmission (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

- Multiprocessor serial data reception

  Figures 12-16 and 12-17 show a sample flowchart for multiprocessor serial reception.

  The following procedure should be used for multiprocessor serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1.

[3] SCI status check, ID reception and comparison:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error processing and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER and FER flags are all cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 12-16 Sample Multiprocessor Serial Reception Flowchart (1)

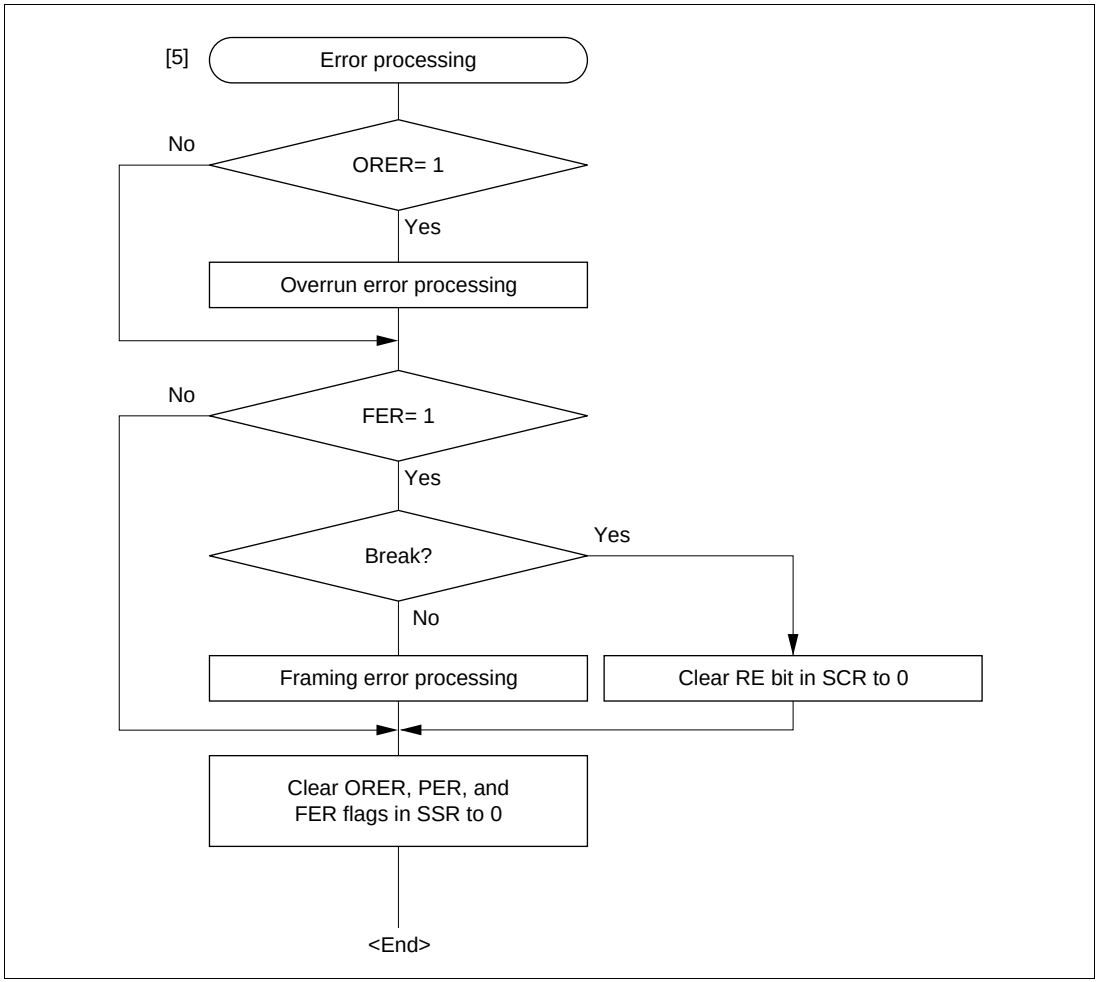


Figure 12-17 Sample Multiprocessor Serial Reception Flowchart (2)

Figure 12-18 shows an example of SCI operation for multiprocessor format reception.

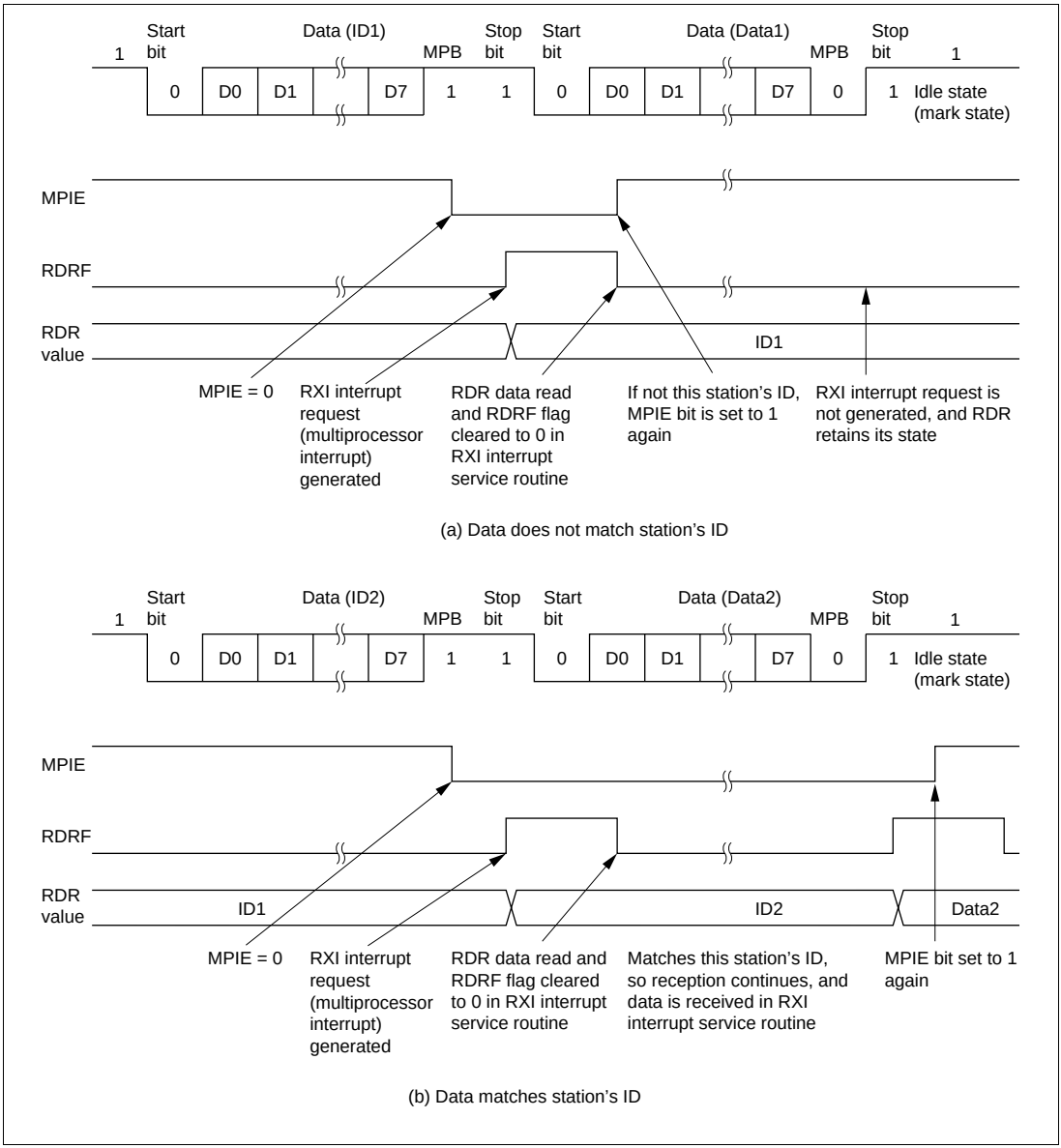


Figure 12-18 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

12.3.4 Operation in Clocked Synchronous Mode

In clocked synchronous mode, data is transmitted or received in synchronization with clock pulses, making it suitable for high-speed serial communication.

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 12-19 shows the general format for clocked synchronous serial communication.

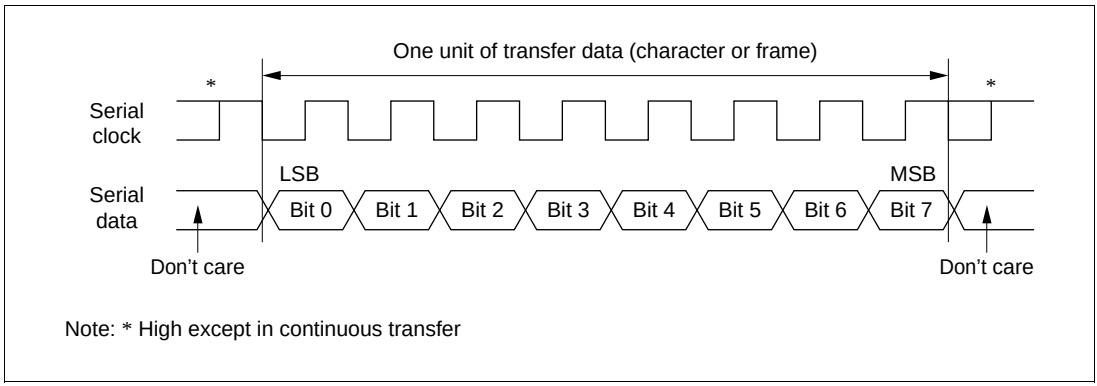


Figure 12-19 Data Format in Synchronous Communication

In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. Data confirmation is guaranteed at the rising edge of the serial clock.

In clocked serial communication, one character consists of data output starting with the LSB and ending with the MSB. After the MSB is output, the transmission line holds the MSB state.

In clocked synchronous mode, the SCI receives data in synchronization with the rising edge of the serial clock.

Data Transfer Format: A fixed 8-bit data format is used.

No parity or multiprocessor bits are added.

Clock: Either an internal clock generated by the on-chip baud rate generator or an external serial clock input at the SCK pin can be selected, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 12-9.

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. When only receive operations are performed, however, the serial clock is output until an overrun error occurs or the RE bit is cleared to 0. If you want to perform receive operations in units of one character, you should select an external clock as the clock source.

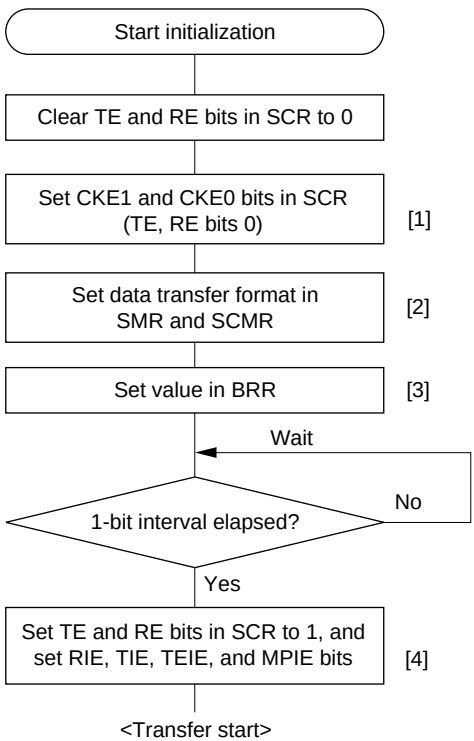
Data Transfer Operations:

- SCI initialization (clocked synchronous mode)

  Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

  When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. 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 12-20 shows a sample SCI initialization flowchart.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, TE and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used.

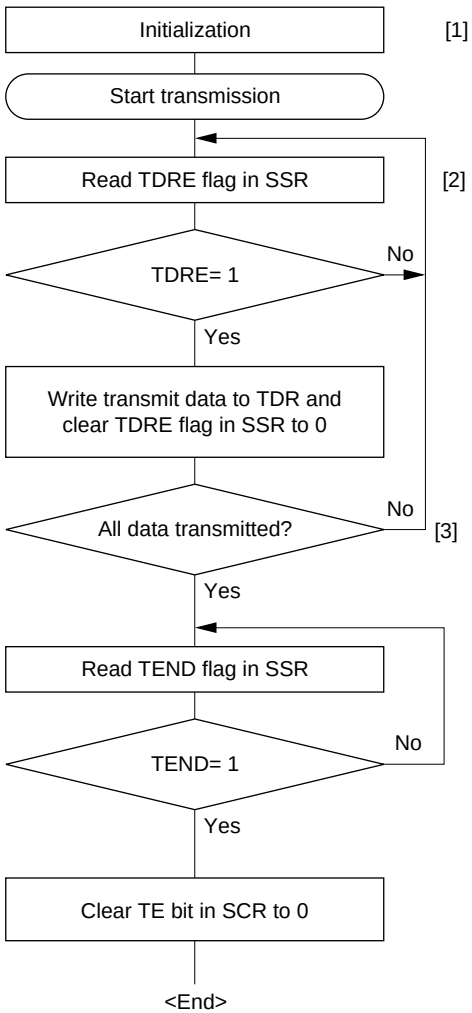
Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 12-20 Sample SCI Initialization Flowchart

- Serial data transmission (clocked synchronous mode)

  Figure 12-21 shows a sample flowchart for serial transmission.

  The following procedure should be used for serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[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, 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 DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR.

Figure 12-21 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

[1] The SCI monitors the TDRE flag in SSR, and if 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 transmit data empty interrupt (TXI) is generated.

When clock output mode has been set, the SCI outputs 8 serial clock pulses. When use of an external clock has been specified, data is output synchronized with the input clock.

The serial transmit data is sent from the TxD pin starting with the LSB (bit 0) and ending with the MSB (bit 7).

[3] The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).

If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the MSB (bit 7) is sent, and the TxD pin maintains its state.

If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

[4] After completion of serial transmission, the SCK pin is fixed.

Figure 12-22 shows an example of SCI operation in transmission.

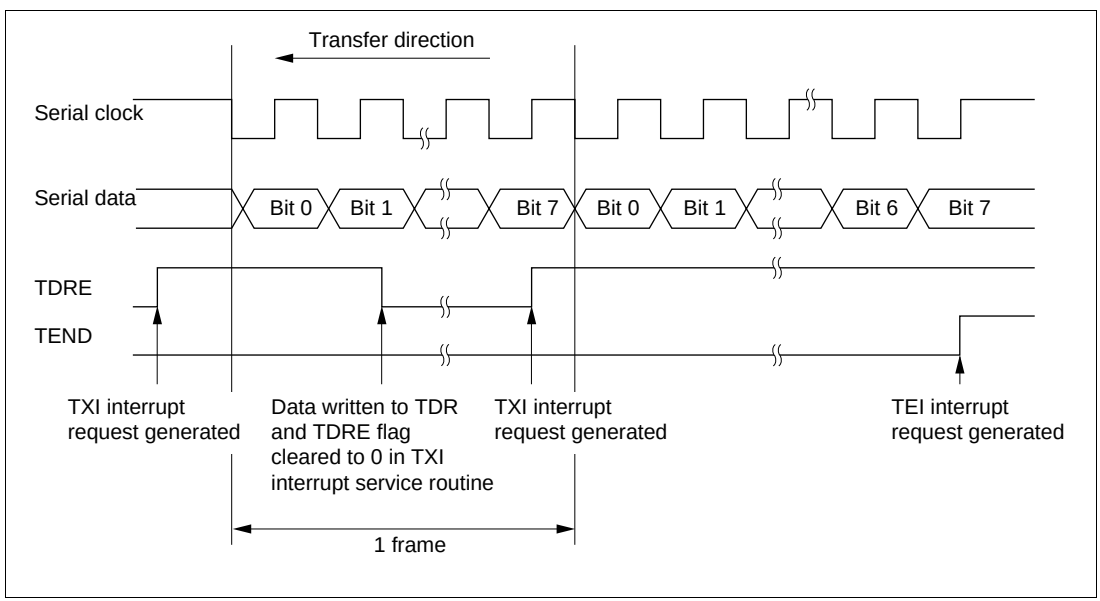


Figure 12-22 Example of SCI Operation in Transmission

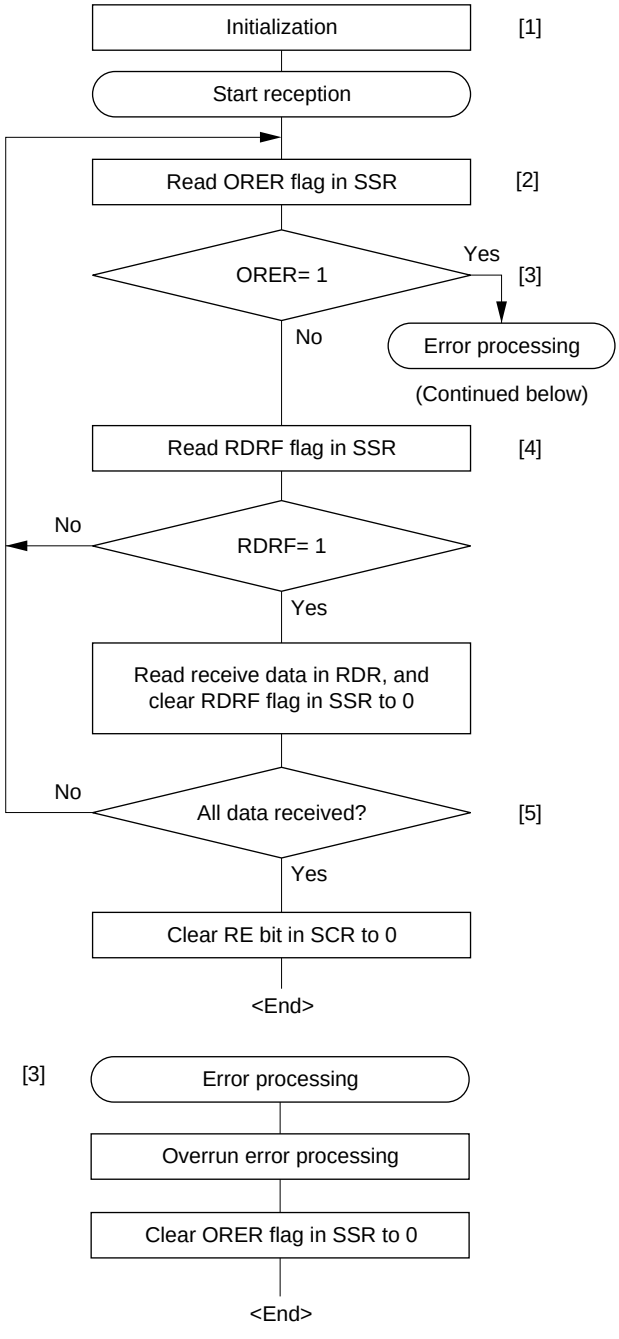
- Serial data reception (clocked synchronous mode)

  Figure 12-23 shows a sample flowchart for serial reception.

  The following procedure should be used for serial data reception.

  When changing the operating mode from asynchronous to clocked synchronous, be sure to check that the ORER, PER, and FER flags are all cleared to 0.

  The RDRF flag will not be set if the FER or PER flag is set to 1, and neither transmit nor receive operations will be possible.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR , and after performing the appropriate error processing, 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 DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Figure 12-23 Sample Serial Reception Flowchart

In serial reception, the SCI operates as described below.

[1] The SCI performs internal initialization in synchronization with serial clock input or output.

[2] The received data is stored in RSR in LSB-to-MSB order.

After reception, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from RSR to RDR.

If this check is passed, the RDRF flag is set to 1, and the receive data is stored in RDR. If a receive error is detected in the error check, the operation is as shown in table 12-11.

Neither transmit nor receive operations can be performed subsequently when a receive error has been found in the error check.

[3] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data full interrupt (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER flag changes to 1, a receive error interrupt (ERI) request is generated.

Figure 12-24 shows an example of SCI operation in reception.

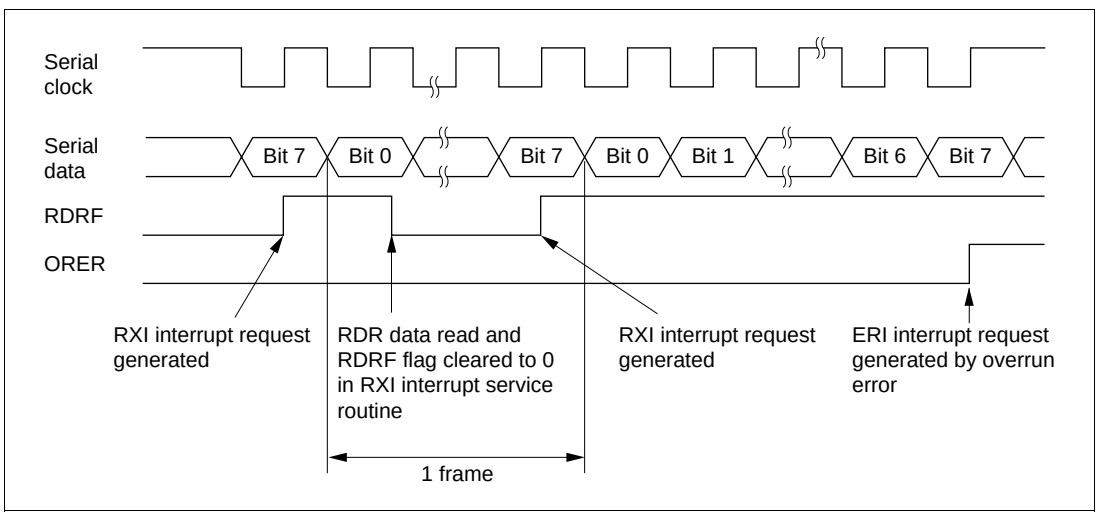
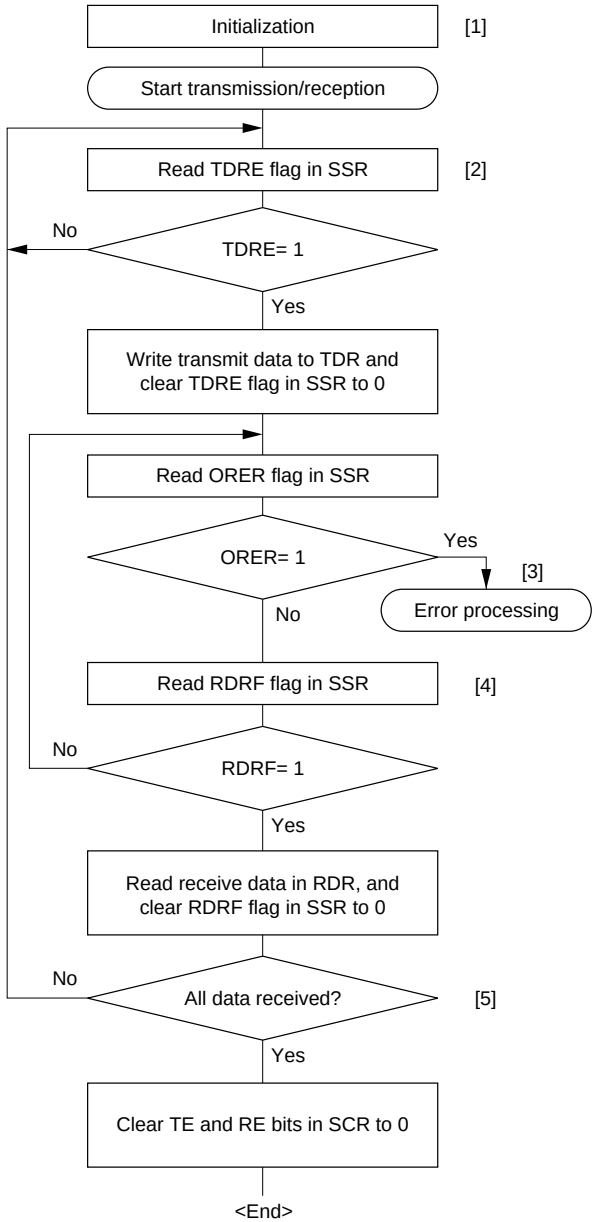


Figure 12-24 Example of SCI Operation in Reception

- Simultaneous serial data transmission and reception (clocked synchronous mode)

  Figure 12-25 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.



[1] SCI initialization:
The TxD pin is designated as the 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 processing:
If a receive error occurs, read the ORER flag in SSR , and after performing the appropriate error processing, 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 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 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 bit and RE bit to 0, then set both these bits to 1 simultaneously.

Figure 12-25 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

12.4 SCI Interrupts

The SCI has four interrupt sources: the transmit-end interrupt (TEI) request, receive-error interrupt (ERI) request, receive-data-full interrupt (RXI) request, and transmit-data-empty interrupt (TXI) request. Table 12-13 shows the interrupt sources and their relative priorities. Individual interrupt sources can be enabled or disabled with the TIE, RIE, and TEIE bits in the SCR. Each kind of interrupt request is sent to the interrupt controller independently.

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 DMAC or DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer is performed by the DMAC or DTC. The DMAC or DTC cannot be activated by a TEI interrupt request.

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 can activate the DMAC or DTC to perform data transfer. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DMAC or DTC. The DMAC or DTC cannot be activated by an ERI interrupt request.

Table 12-13 SCI Interrupt Sources

Channel	Interrupt Source	Description	DMAC Activation	DTC Activation	Priority*
0	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	High
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	
1	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	
2	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	Not possible	
	RXI	Interrupt due to receive data full state (RDRF)	Possible	Not possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	Not possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Not possible	Low

Note: * This table shows the initial state immediately after a reset. Relative priorities among channels can be changed by means of the interrupt controller.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1. The TEND flag is cleared at the same time as the TDRE flag. Consequently, if a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt may have priority for acceptance, with the result that the TDRE and TEND flags are cleared. Note that the TEI interrupt will not be accepted in this case.

12.5 Usage Notes

The following points should be noted when using the SCI.

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.

Operation when Multiple Receive Errors Occur Simultaneously

If a number of receive errors occur at the same time, the state of the status flags in SSR is as shown in table 12-14. If there is an overrun error, data is not transferred from RSR to RDR, and the receive data is lost.

Table 12-14 State of SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	
RDRF	ORER	FER	PER	RSR to RDR	Receive Error Status
1	1	0	0	X	Overrun error
0	0	1	0	○	Framing error
0	0	0	1	○	Parity error
1	1	1	0	X	Overrun error + framing error
1	1	0	1	X	Overrun error + parity error
0	0	1	1	○	Framing error + parity error
1	1	1	1	X	Overrun error + framing error + parity error

Notes: ○: Receive data is transferred from RSR to RDR.
X: Receive data is not transferred from RSR to RDR.

Break Detection and Processing (Asynchronous Mode Only): When framing error (FER) 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 parity error flag (PER) 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.

Sending a Break (Asynchronous Mode Only): The TxD pin has a dual function as an I/O port whose direction (input or output) is determined by DR and DDR. This can be used to send a break.

Between serial transmission initialization and setting of the TE bit to 1, the mark state is replaced by the value of DR (the pin does not function as the TxD pin until the TE bit is set to 1). Consequently, DDR and DR for the port corresponding to the TxD pin are first set to 1.

To send a break during serial transmission, first clear DR to 0, then clear the TE bit to 0.

When the TE bit 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.

Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only): 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.

Receive Data Sampling Timing and Reception Margin in Asynchronous Mode: In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the transfer 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 internally at the rising edge of the 8th pulse of the basic clock. This is illustrated in figure 12-26.

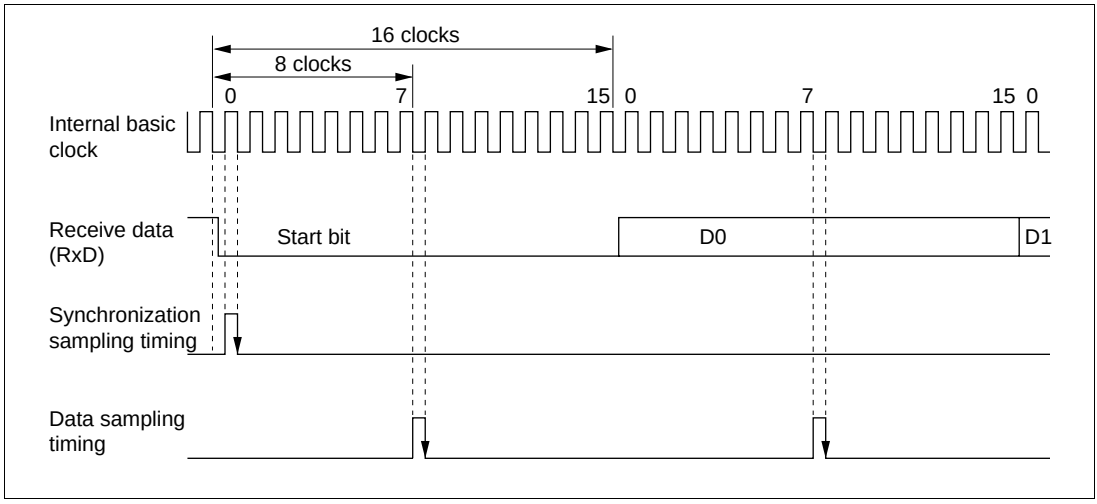


Figure 12-26 Receive Data Sampling Timing in Asynchronous Mode

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 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 of 46.875% is given by formula (2) below.

When D = 0.5 and F = 0,

$$M = \left(0.5 - \frac{1}{2 \times 16}\right) \times 100\%$$

$$= 46.875\%$$

... Formula (2)

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

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 5 ø clock cycles after TDR is updated by the DMAC or DTC. Misoperation may occur if the transmit clock is input within 4 ø clocks after TDR is updated. (Figure 12-27)
- When RDR is read by the DMAC or DTC, be sure to set the activation source to the relevant SCI reception end interrupt (RXI).

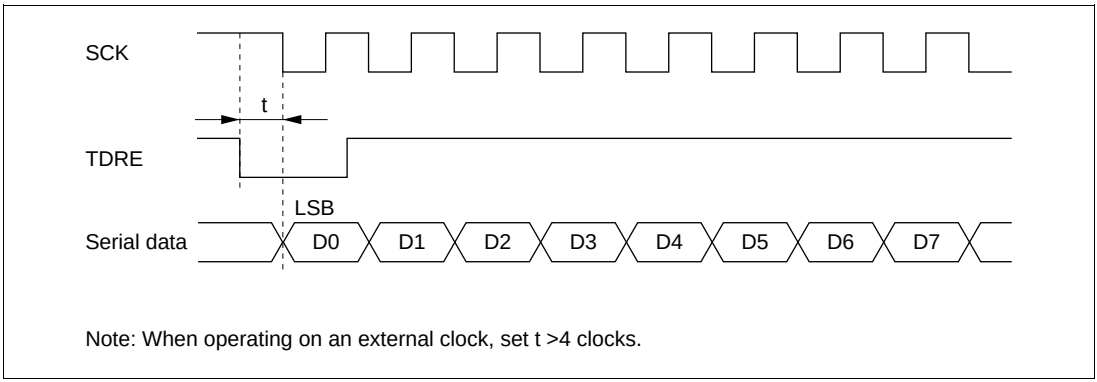


Figure 12-27 Example of Clocked Synchronous Transmission by DTC

Operation in Case of Mode Transition

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making a module stop mode, software standby mode, or subsleep mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode, software standby mode, or subsleep mode depend on the port settings, and becomes 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 12-28 shows a sample flowchart for mode transition during transmission. Port pin states are shown in figures 12-29 and 12-30.

  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, software standby mode, or subsleep 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, software standby mode, watch mode, subactive mode, or subsleep mode transition. RSR, RDR, and SSR are reset. If a transition is made without stopping operation, 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 12-31 shows a sample flowchart for mode transition during reception.

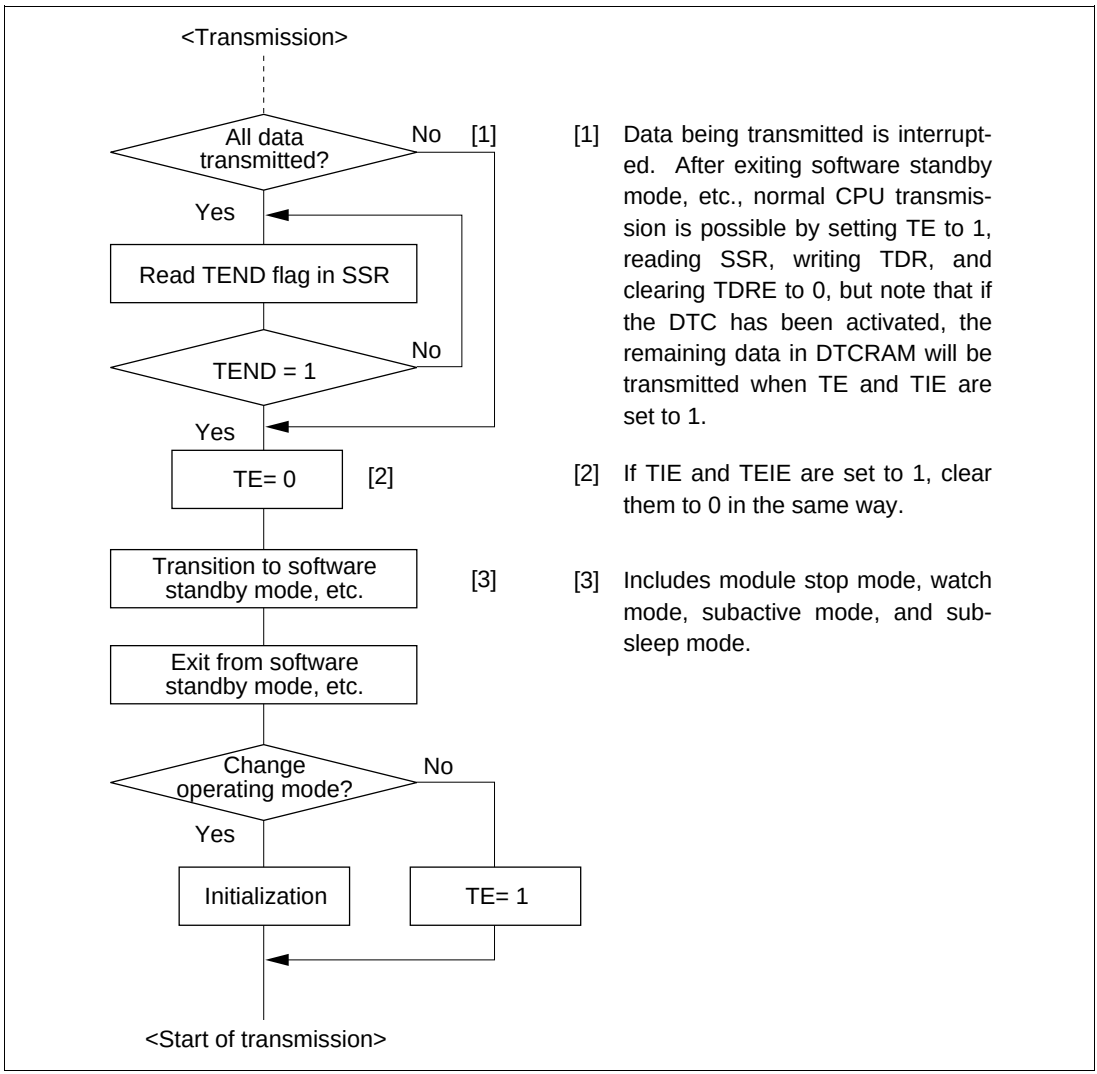


Figure 12-28 Sample Flowchart for Mode Transition during Transmission

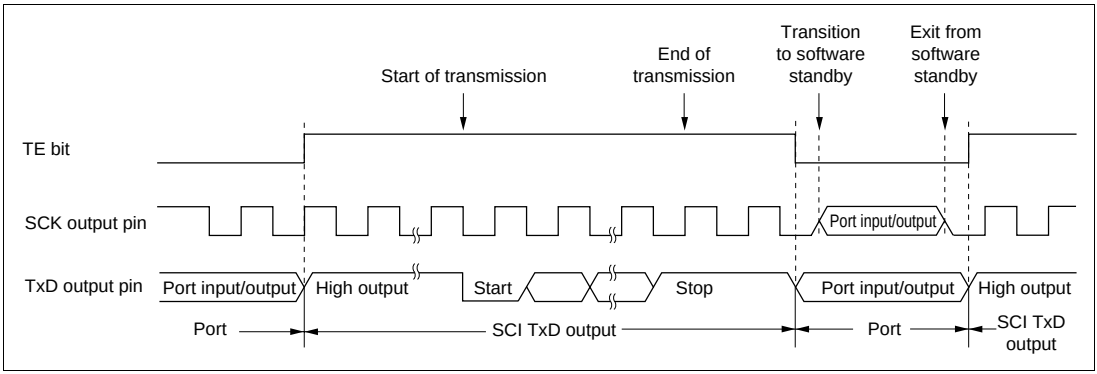


Figure 12-29 Asynchronous Transmission Using Internal Clock

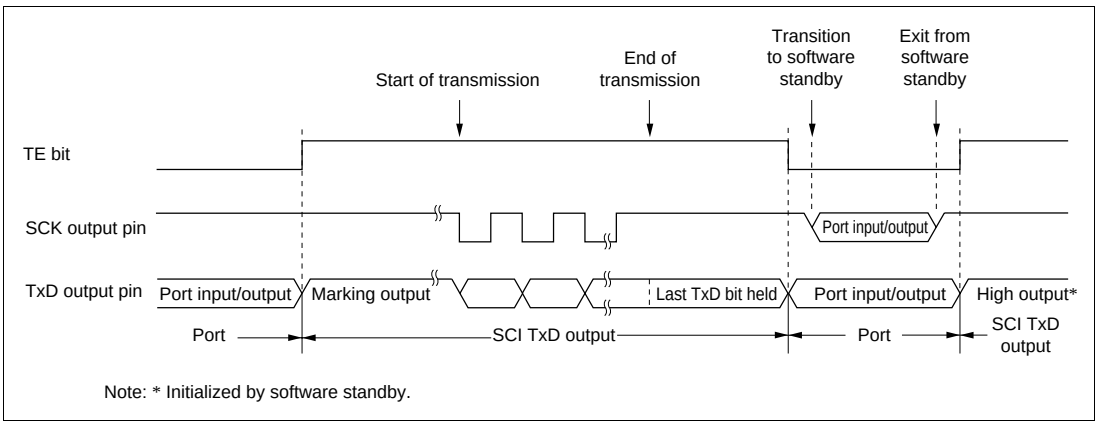


Figure 12-30 Synchronous Transmission Using Internal Clock

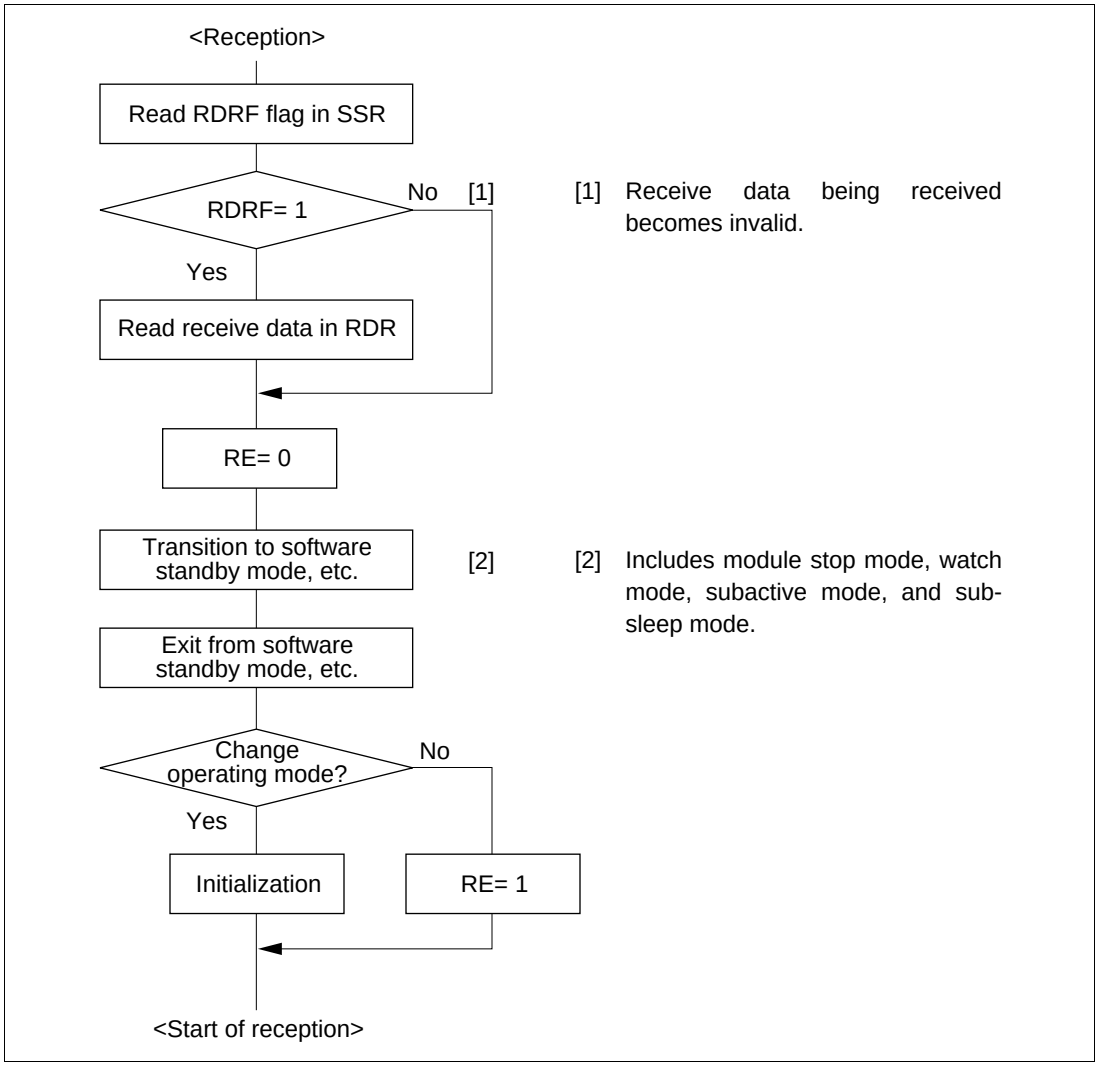


Figure 12-31 Sample Flowchart for Mode Transition during Reception

Switching from SCK Pin Function to Port Pin Function:

- Problem in Operation: When switching the SCK pin function to the output port function (high-level output) by making the following settings while DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1 (synchronous mode), low-level output occurs for one half-cycle.

1. End of serial data transmission
2. TE bit = 0
3. C/$\overline{A}$ bit = 0 ... switchover to port output
4. Occurrence of low-level output (see figure 12-32)

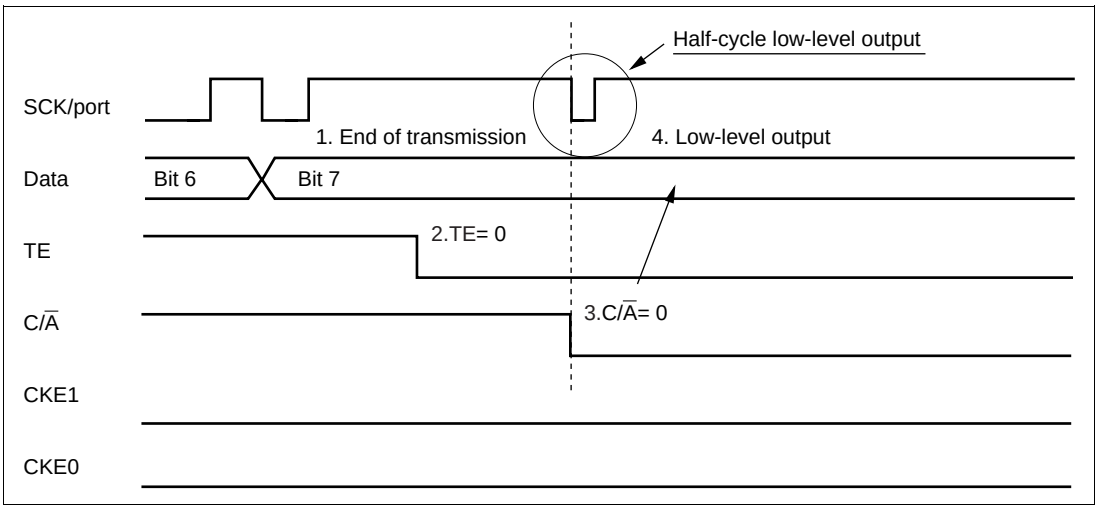


Figure 12-32 Operation when Switching from SCK Pin Function to Port Pin Function

- Sample Procedure for Avoiding Low-Level Output: As this sample procedure temporarily places the SCK pin in the input state, the SCK/port pin should be pulled up beforehand with an external circuit.

  With DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, make the following settings in the order shown.

1. End of serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 ... switchover to port output
5. CKE1 bit = 0

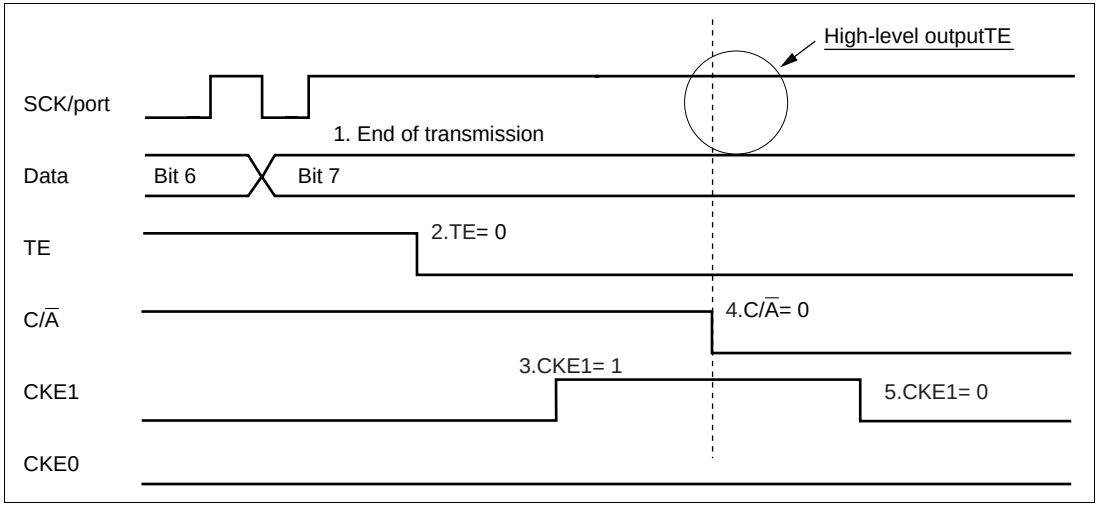


Figure 12-33 Operation when Switching from SCK Pin Function to Port Pin Function (Example of Preventing Low-Level Output)

Section 13 D/A Converter

13.1 Overview

The H8S/2214 includes a one-channel D/A converter.

13.1.1 Features

D/A converter features are listed below

- 8-bit resolution
- One output channel
- Maximum conversion time of 10 μs (with 20 pF load)
- Output voltage of 0 V to Vref
- D/A output hold function in software standby mode
- Module stop mode can be set
  - As the initial setting, D/A converter operation is halted. Register access is enabled by exiting module stop mode.

13.1.2 Block Diagram

Figure 13-1 shows a block diagram of the D/A converter.

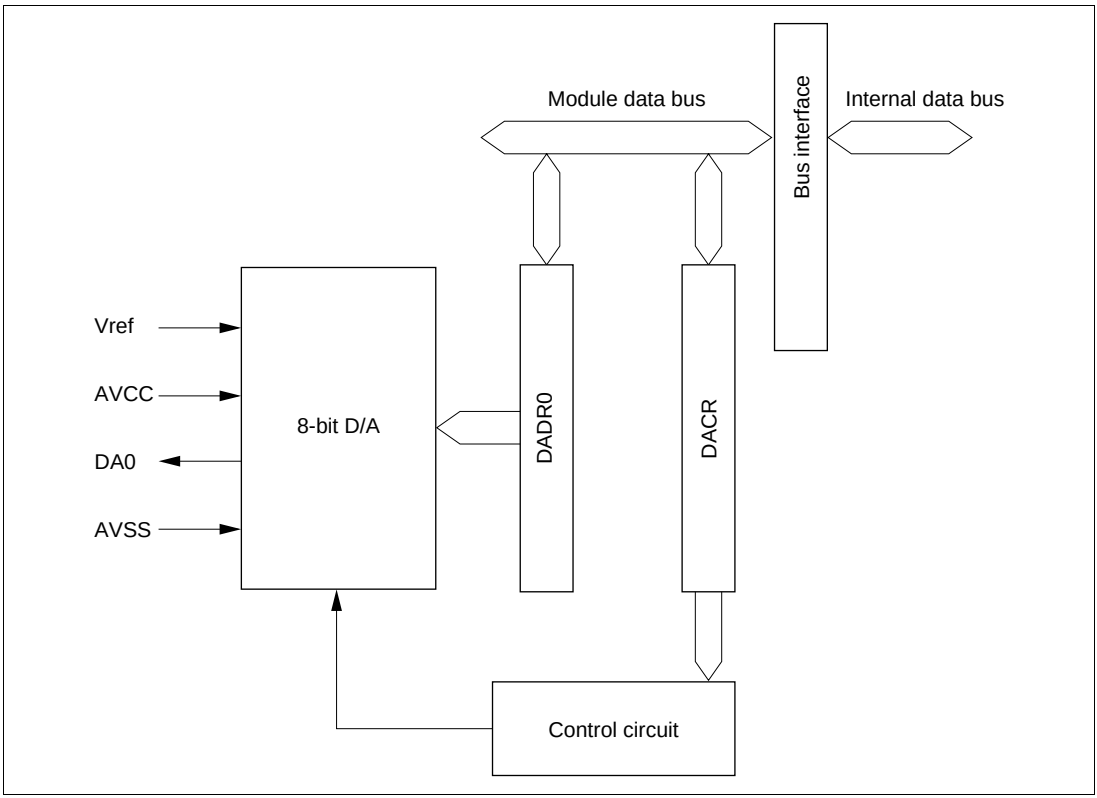


Figure 13-1 Block Diagram of D/A Converter

13.1.3 Pin Configuration

Table 13-1 summarizes the input and output pins of the D/A converter.

Table 13-1 Pin Configuration

Pin Name	Symbol	I/O	Function
Analog power pin	AVCC	Input	Analog power source
Analog ground pin	AVSS	Input	Analog ground and reference voltage
Analog output pin 0	DA0	Output	Channel 0 analog output
Reference voltage pin	Vref	Input	Analog reference voltage

13.1.4 Register Configuration

Table 13-2 summarizes the registers of the D/A converter.

Table 13-2 D/A Converter Registers

Name	Abbreviation	R/W	Initial Value	Address*
D/A data register 0	DADR0	R/W	H'00	H'FDAC
D/A control register	DACR	R/W	H'1F	H'FDAE
Module stop control register C	MSTPCRC	R/W	H'FF	H'FDEA

Note:* Lower 16 bits of the address.

13.2 Register Descriptions

13.2.1 D/A Data Register 0 (DADR0)

Bit	:	7	6	5	4	3	2	1	0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

D/A data register 0 (DADR0) is an 8-bit readable/writable registers that stores data for conversion.

Whenever output is enabled, the value in the D/A data register is converted and output from the analog output pin.

DADR0 is initialized to H'00 by a reset and in hardware standby mode.

13.2.2 D/A Control Register (DACR)

Bit	:	7	6	5	4	3	2	1	0
		—	DAOE0	—	—	—	—	—	—
Initial value:		0	0	0	1	1	1	1	1
R/W	:	R/W	R/W	R/W	—	—	—	—	—

DACR is an 8-bit readable/writable register that controls the operation of the D/A converter.

DACR is initialized to H'1F by a reset and in hardware standby mode.

Bit 7—Reserved: Only 0 should be written to this bit.

Bit 6—D/A Output Enable 0 (DAOE0): Controls D/A conversion and analog output.

Bit 6

DAOE0	Description	
0	Analog output DA0 is disabled	(Initial value)
1	Channel 0 D/A conversion is enabled; analog output DA0 is enabled	

Bit 5—Reserved: Only 0 should be written to this bit.

If the H8S/2214 enters software standby mode when D/A conversion is enabled, the D/A output is held and the analog power current is the same as during D/A conversion. When it is necessary to reduce the analog power current in software standby mode, clear the DAOE0 bit to 0 to disable D/A output.

Bits 4 to 0—Reserved: Read-only bits, always read as 1.

13.2.3 Module Stop Control Register C (MSTPCRC)

Bit	:	7	6	5	4	3	2	1	0
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPC5 bit in MSTPCR is set to 1, D/A converter operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 17.5, Module Stop Mode.

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

Bit 5—Module Stop (MSTPC5): Specifies the D/A converter module stop mode.

Bit 5		
MSTPC5	**Description**	
0	D/A converter module stop mode cleared	
1	D/A converter module stop mode set	(Initial value)

13.3 Operation

D/A conversion is performed continuously while enabled by DACR. If either DADR0 is written to, the new data is immediately converted. The conversion result is output by setting the corresponding DAOE0 bit to 1.

The operation example described in this section concerns D/A conversion on channel 0. Figure 13-2 shows the timing of this operation.

[1] Write the conversion data to DADR0.

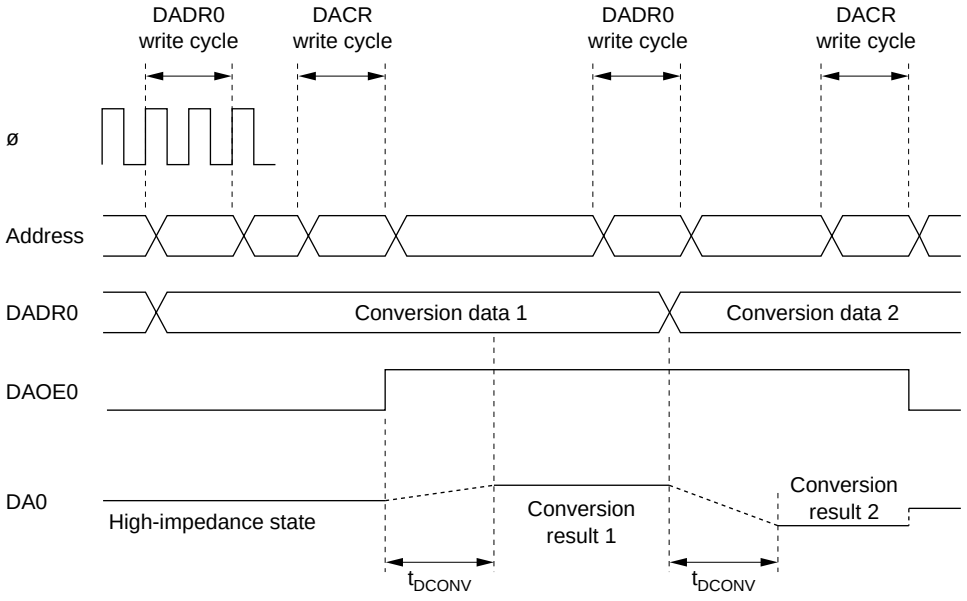
[2] Set the DAOE0 bit in DACR to 1. D/A conversion is started and the DA0 pin becomes an output pin. The conversion result is output after the conversion time has elapsed. The output value is expressed by the following formula:

$$\frac{\text{DADR contents}}{256} \times \text{Vref}$$

The conversion results are output continuously until DADR0 is written to again or the DAOE0 bit is cleared to 0.

[3] If DADR0 is written to again, the new data is immediately converted. The new conversion result is output after the conversion time has elapsed.

[4] If the DAOE0 bit is cleared to 0, the DA0 pin becomes an input pin.



Legend

t_{DCONV}: D/A conversion time

Figure 13-2 Example of D/A Converter Operation

Section 14 RAM

14.1 Overview

The H8S/2214 has 12 kbytes of 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. This makes it possible to perform fast word data transfer.

The on-chip RAM can be enabled or disabled by means of the RAM enable bit (RAME) in the system control register (SYSCR).

14.1.1 Block Diagram

Figure 14-1 shows a block diagram of the on-chip RAM.

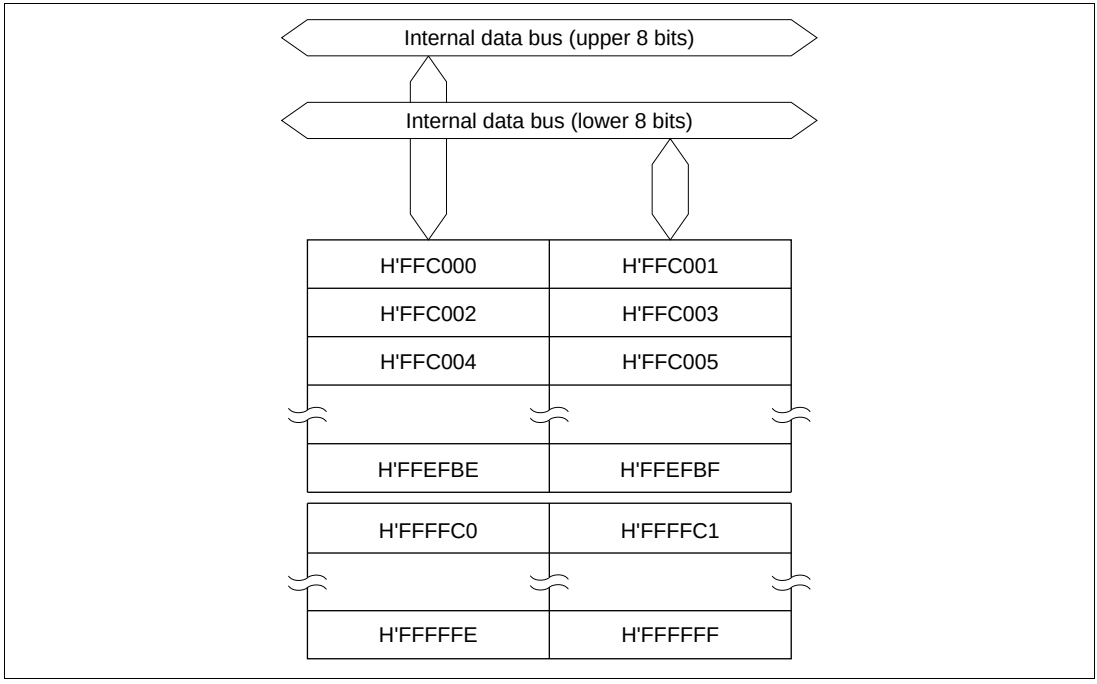


Figure 14-1 Block Diagram of RAM

14.1.2 Register Configuration

The on-chip RAM is controlled by SYSCR. Table 14-1 shows the address and initial value of SYSCR.

Table 14-1 RAM Register

Name	Abbreviation	R/W	Initial Value	Address*
System control register	SYSCR	R/W	H'01	H'FDE5

Note: * Lower 16 bits of the address.

14.2 Register Descriptions

14.2.1 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	INTM1	INTM0	NMIEG	MRESE	—	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	—	—	R/W	R/W	R/W	R/W	—	R/W

The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. For details of other bits in SYSCR, see section 3.2.2, System Control Register (SYSCR).

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized when the reset state is released. It is not initialized in software standby mode.

Bit 0		
RAME	**Description**	
0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

14.3 Operation

When the RAME bit is set to 1, accesses to addresses H'FFC000 to H'FFEFBF and H'FFFFC0 to H'FFFFFF in the H8S/2214 is directed to the on-chip RAM. When the RAME bit is cleared to 0, the off-chip address space is accessed.

Since the on-chip RAM is connected to the CPU by an internal 16-bit data bus, it can be written to and read in byte or word units. Each type of access can be performed in one state.

Even addresses use the upper 8 bits, and odd addresses use the lower 8 bits. Word data must start at an even address.

14.4 Usage Note

DTC register information can be located in addresses H'FFEBC0 to H'FFEFBF. When the DTC is used, the RAME bit must not be cleared to 0.

Section 15 ROM

15.1 Overview

The H8S/2214 has 128 kbytes of on-chip ROM (flash memory or mask ROM). The ROM is connected to the CPU by a 16-bit data bus. The CPU accesses both byte data and word data in one state, making possible rapid instruction fetches and high-speed processing.

The on-chip ROM is enabled or disabled by setting the mode pins (MD2, MD1, and MD0).

The flash memory versions can be erased and programmed on-board as well as with a PROM programmer.

15.1.1 Block Diagram

Figure 15-1 shows a block diagram of the on-chip ROM.

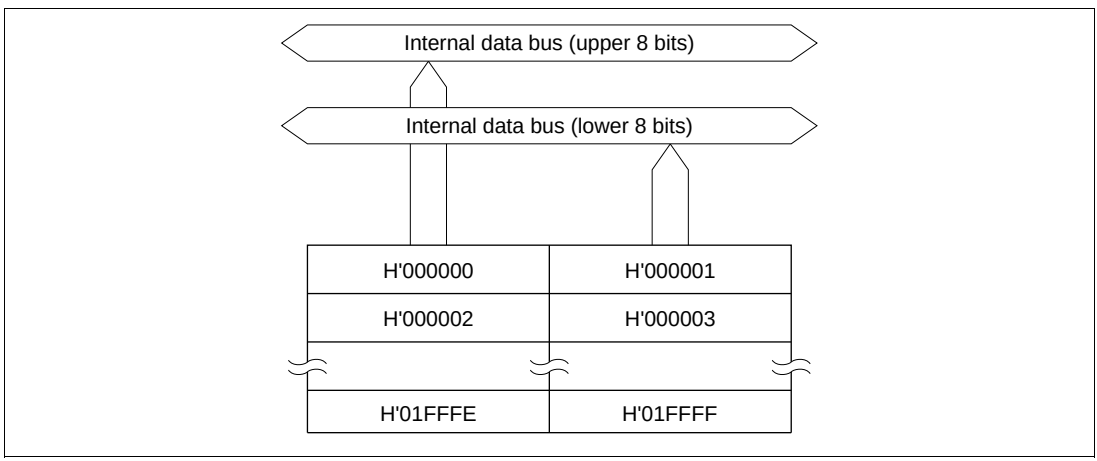


Figure 15-1 Block Diagram of ROM

15.1.2 Register Configuration

The H8S/2214's on-chip ROM is controlled by the mode pins. The register configuration is shown in table 15-1.

Table 15-1 ROM Register

Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R/W	Undefined	H'FDE7

Note: * Lower 16 bits of the address.

15.2 Register Descriptions

15.2.1 Mode Control Register (MDCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	:	1	0	0	0	0	—*	—*	—*
R/W	:	—	—	—	—	—	R	R	R

Note: * Determined by pins MD2 to MD0.

MDCR is an 8-bit read-only register that indicates the current operating mode of the H8S/2214.

Bit 7—Reserved: Read-only bit, always read as 1.

Bits 6 to 3—Reserved: Read-only bits, always read as 0.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to pins MD2 to MD0. MDS2 to MDS0 are read-only bits, and 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 power-on reset, but are retained after a manual reset.

15.3 Operation

The on-chip ROM is connected to the CPU by a 16-bit data bus, and both byte and word data can be accessed in one state. Even addresses are connected to the upper 8 bits, and odd addresses to the lower 8 bits. Word data must start at an even address.

The on-chip ROM is enabled and disabled by setting the mode pins (MD2, MD1, and MD0). These settings are shown in table 15-2.

Table 15-2 Operating Modes and ROM Area (F-ZTAT version and Mask ROM version)

		Mode Pin				
Operating Mode		**FWE**	**MD2**	**MD1**	**MD0**	**On-Chip ROM**
Mode 0	—	0	0	0	0	—
Mode 1					1	
Mode 2				1	0	
Mode 3					1	
Mode 4	Advanced expanded mode with on-chip ROM disabled		1	0	0	Disabled
Mode 5	Advanced expanded mode with on-chip ROM disabled				1	
Mode 6	Advanced expanded mode with on-chip ROM enabled			1	0	Enabled (128 kbytes)*1
Mode 7	Advanced single-chip mode				1	Enabled (128 kbytes)*1
Mode 8	—	1	0	0	0	—
Mode 9					1	
Mode 10	Boot mode (advanced expanded mode with on-chip ROM enabled)*1			1	0	Enabled (128 kbytes)*2
Mode 11	Boot mode (advanced single-chip mode)*2				1	Enabled (128 kbytes)*2
Mode 12	—		1	0	0	—
Mode 13					1	
Mode 14	User program mode (advanced expanded mode with on-chip ROM enabled)*1			1	0	Enabled (128 kbytes)*1
Mode 15	User program mode (advanced single-chip mode)*2				1	Enabled (128 kbytes)*1

Notes: *1 Apart from the fact that flash memory can be erased and programmed, operation is the same as in advanced expanded mode with on-chip ROM enabled.

*2 Apart from the fact that flash memory can be erased and programmed, operation is the same as in advanced single-chip mode.

15.4 Overview of Flash Memory

15.4.1 Features

The H8S/2214 has 128 kbytes of on-chip flash memory. The features of the flash memory are summarized below.

- Four flash memory operating modes
  - Program mode
  - Erase mode
  - Program-verify mode
  - Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Block erase (in single-block units) can be performed. To erase multiple blocks, each block must be erased in turn. In block erasing, 1-kbyte, 8-kbyte, 16-kbyte, 28-kbyte, and 32-kbyte block units can be set as required.
- Programming/erase times

  The flash memory programming time is T.B.D ms (typ.) for simultaneous 128-byte programming, equivalent to T.B.D µs (typ.) per byte, and the erase time is T.B.D ms (typ.).
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - Boot mode
  - User program mode
- Automatic bit rate adjustment

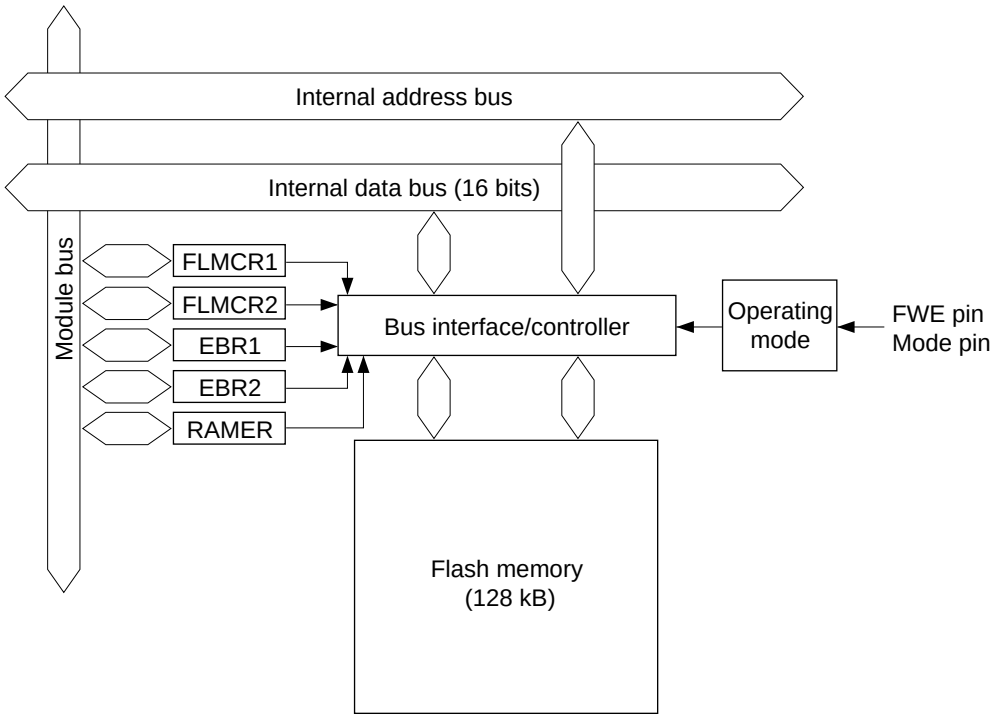
  With data transfer in boot mode, the LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation in RAM

  Flash memory programming can be emulated in real time by overlapping a part of RAM onto flash memory.
- Protect modes

  There are two protect modes, hardware and software, which allow protected status to be designated for flash memory program/erase/verify operations.
- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

15.4.2 Block Diagram



Legend
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2
RAMER: RAM emulation register

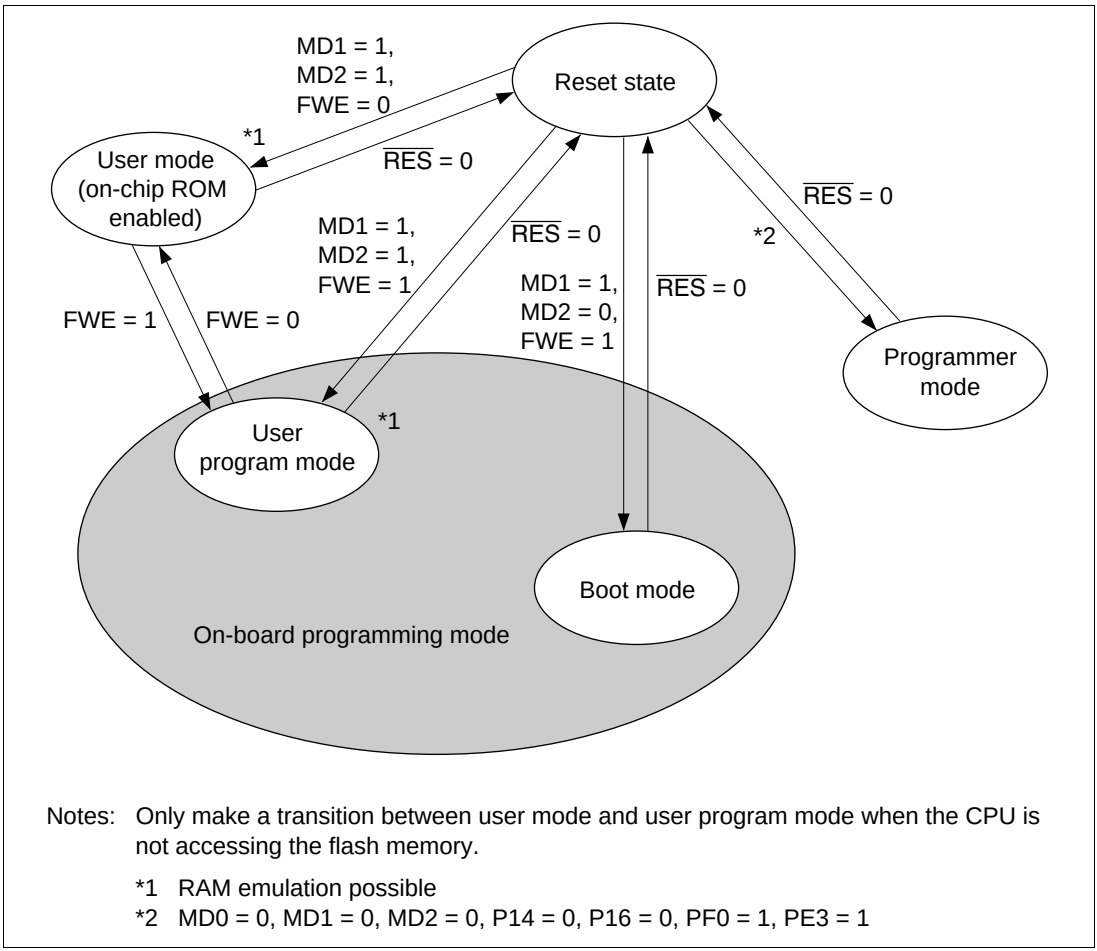
Note: These registers are for use exclusively by the flash memory version. Reads to the corresponding addresses in the mask ROM version will return an undefined value, and writes to these addresses are invalid.

Figure 15-2 Block Diagram of Flash Memory

15.4.3 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, the microcomputer enters an operating mode as shown in figure 15-3. Transitions between user mode and user program mode should only be made when the CPU is not accessing the flash memory.

The boot, user program and programmer modes are provided as modes to write and erase the flash memory.



Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

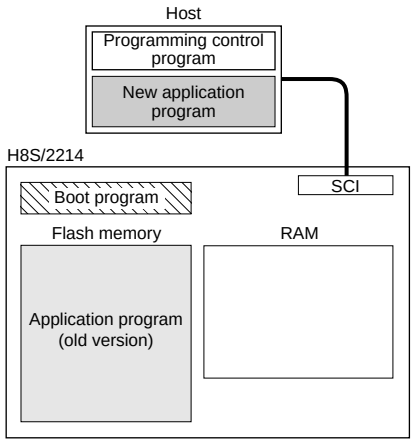
*1 RAM emulation possible
*2 MD0 = 0, MD1 = 0, MD2 = 0, P14 = 0, P16 = 0, PF0 = 1, PE3 = 1

Figure 15-3 Flash Memory State Transitions

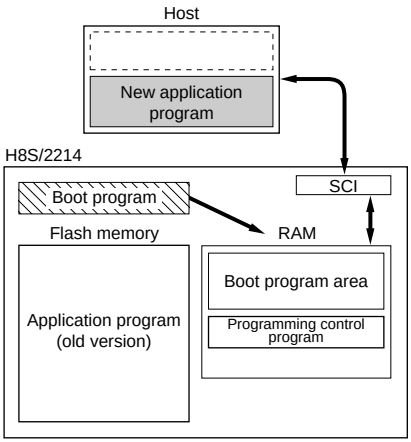
15.4.4 On-Board Programming Modes

Boot Mode

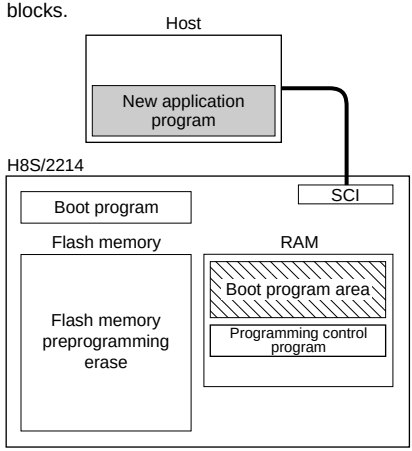
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



2. Programming control program transfer
   When boot mode is entered, the boot program in the H8S/2214 (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, total 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.

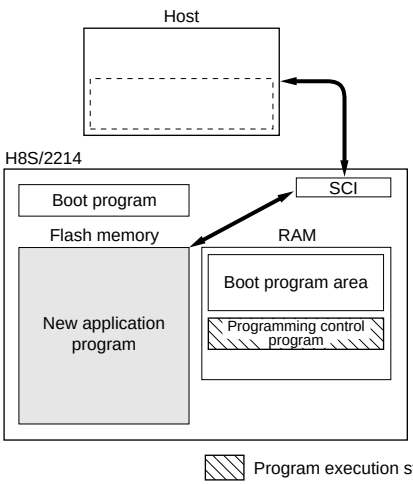
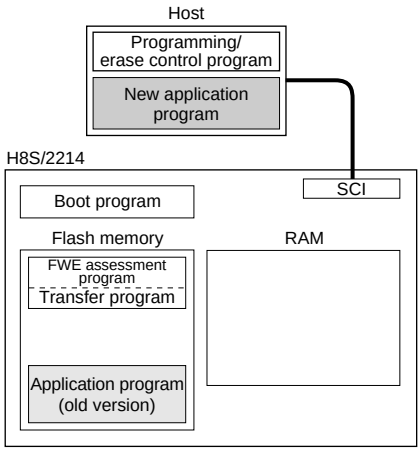
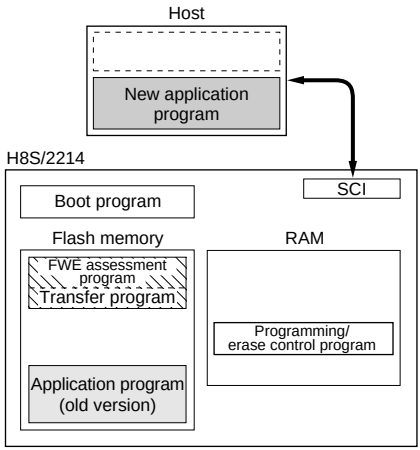


Figure 15-4 Boot Mode

1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.

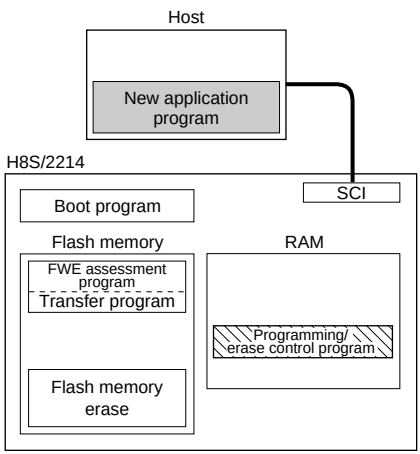
2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.

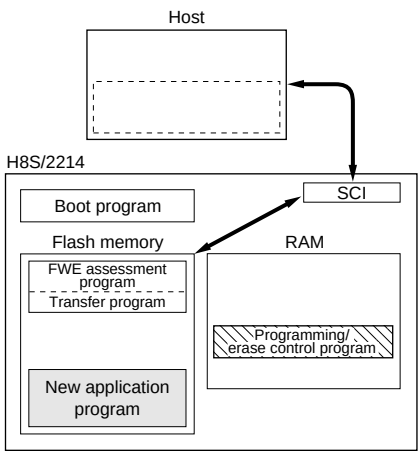




3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.

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 15-5 User Program Mode

15.4.5 Flash Memory Emulation in RAM

Emulation should be performed in user mode or user program mode. When the emulation block set in RAMER is accessed while the emulation function is being executed, data written in the overlap RAM is read.

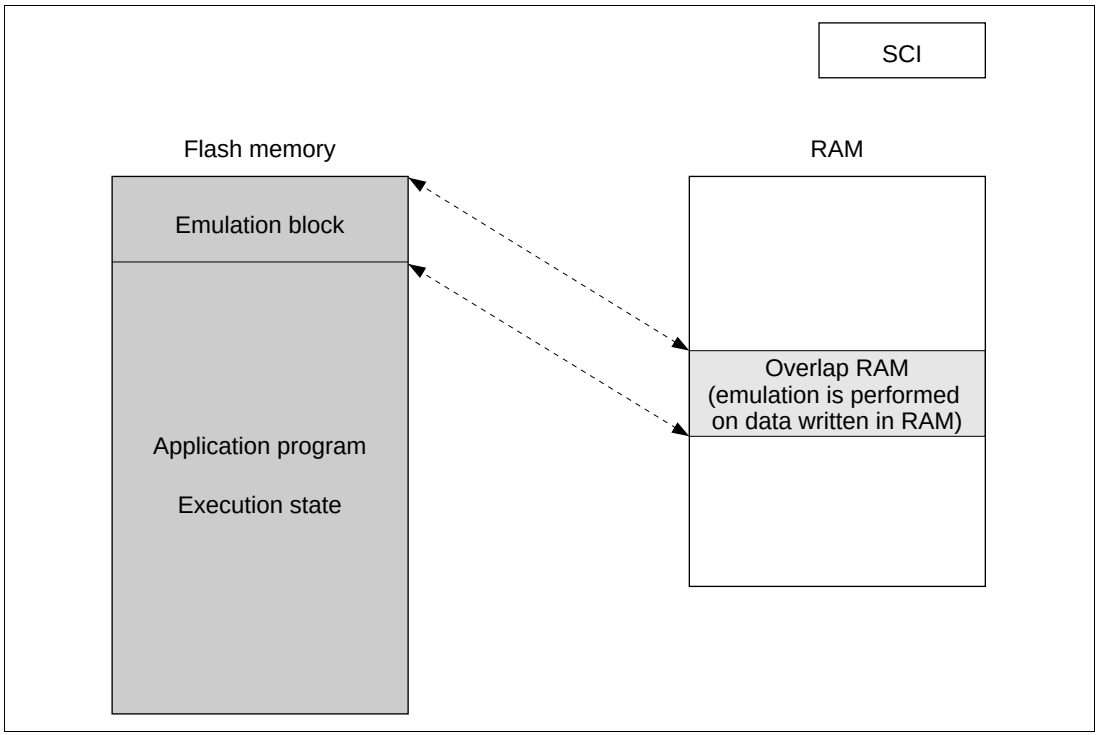


Figure 15-6 Reading Overlap RAM Data in User Mode or User Program Mode

When overlap RAM data is confirmed, the RAMS bit is cleared, RAM overlap is released, and writes should actually be performed to the flash memory.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

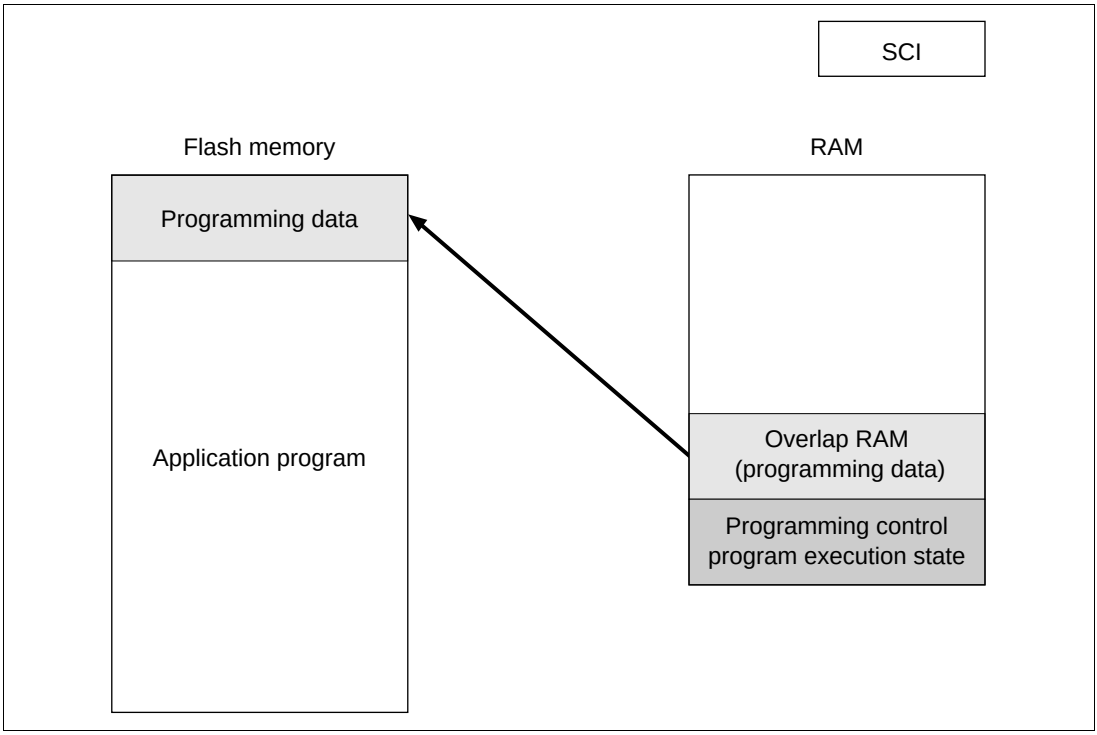


Figure 15-7 Writing Overlap RAM Data in User Program Mode

15.4.6 Differences between Boot Mode and User Program Mode

Table 15-3 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	(2)	(1) (2) (3)

(1) Erase/erase-verify

(2) Program/program-verify

(3) Emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

15.4.7 Block Divisions

The flash memory is divided into two 32-kbyte blocks, one 28-kbyte block, one 16-kbyte block, two 8-kbyte blocks, and four 1-kbyte blocks.

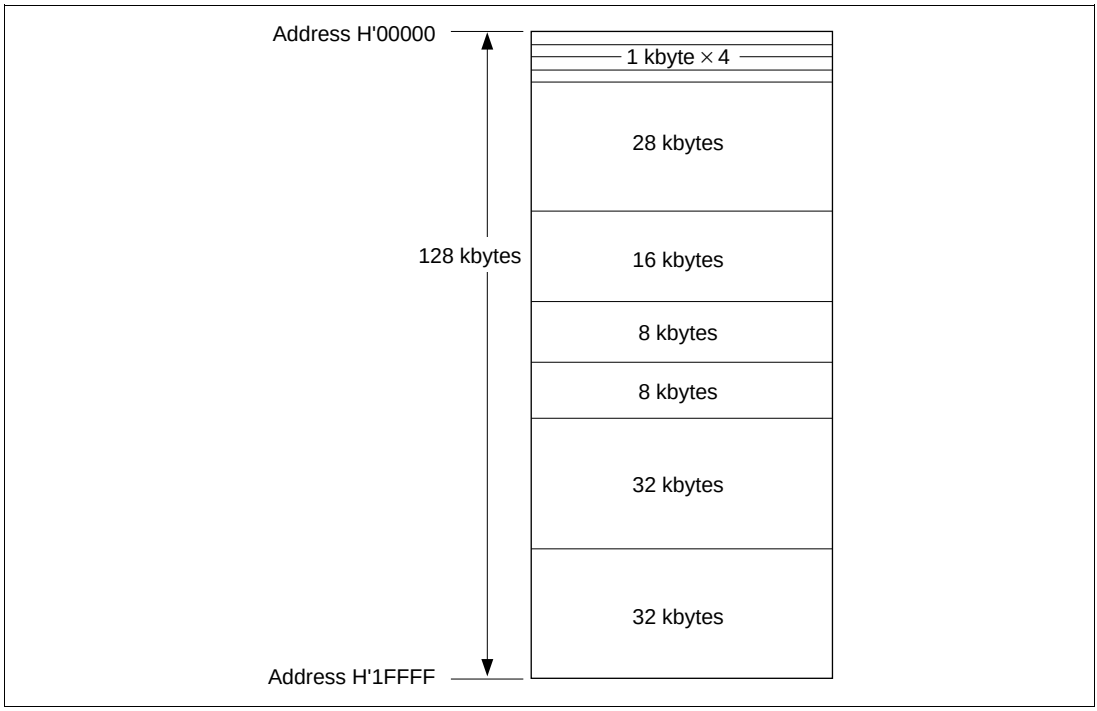


Figure 15-8 Flash Memory Blocks

15.5 Pin Configuration

The flash memory is controlled by means of the pins shown in table 15-4.

Table 15-4 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash write enable	FWE	Input	Flash program/erase protection by hardware
Mode 2	MD2	Input	Sets LSI operating mode
Mode 1	MD1	Input	Sets LSI operating mode
Mode 0	MD0	Input	Sets LSI operating mode
Port F0	PF0	Input	Sets LSI operating mode when MD2 = MD1 = MD0 = 0
Port 16	P16	Input	Sets LSI operating mode when MD2 = MD1 = MD0 = 0
Port 14	P14	Input	Sets LSI operating mode when MD2 = MD1 = MD0 = 0
Transmit data	TxD2	Output	Serial transmit data output
Receive data	RxD2	Input	Serial receive data input

15.6 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 15-5. In order to access these registers, the FLSHE bit in SCRX must be set to 1 (except for RAMER, SCRX).

Table 15-5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address*1
Flash memory control register 1	FLMCR1*5	R/W*2	H'00*3	H'FFA8
Flash memory control register 2	FLMCR2*5	R*2	H'00	H'FFA9
Erase block register 1	EBR1*5	R/W*2	H'00*4	H'FFAA
Erase block register 2	EBR2*5	R/W*2	H'00*4	H'FFAB
RAM emulation register	RAMER*5	R/W	H'00	H'FEDB
Serial control register X	SCRX	R/W	H'00	H'FDB4

Notes: *1 Lower 16 bits of the address.

*2 To access these registers, set the FLSHE bit to 1 in serial control register X. Even if FLSHE is set to 1, if the chip is in a mode in which the on-chip flash memory is disabled, a read will return H'00 and writes are invalid. Writes are also invalid when the FWE bit in FLMCR1 is not set to 1.

*3 When a high level is input to the FWE pin, the initial value is H'80.

*4 When a low level is input to the FWE pin, or if a high level is input and the SWE1 bit in FLMCR1 is not set, these registers are initialized to H'00.

*5 FLMCR1, FLMCR2, EBR1, EBR2, and RAMER are 8-bit registers.

Only byte access can be used on these registers, with the access requiring two states. These registers are for use exclusively by the flash memory version. Reads to the corresponding addresses in the mask ROM version will return an undefined value, and writes to these addresses are invalid.

15.7 Register Descriptions

15.7.1 Flash Memory Control Register 1 (FLMCR1)

Bit:	7	6	5	4	3	2	1	0
	FWE	SWE1	ESU1	PSU1	EV1	PV1	E1	P1
Initial value:	—*	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the state of the FWE pin.

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode for addresses H'00000 to H'1FFFF is entered by setting SWE1 bit to 1 when FWE = 1, then setting the PV1 or EV1 bit. Program mode for addresses H'00000 to H'1FFFF is entered by setting SWE1 bit to 1 when FWE = 1, then setting the PSU1 bit, and finally setting the P1 bit. Erase mode for addresses H'00000 to H'1FFFF is entered by setting SWE1 bit to 1 when FWE = 1, then setting the ESU1 bit, and finally setting the E1 bit. FLMCR1 is initialized by a power-on reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writes are enabled only in the following cases: Writes to bit SWE1 of FLMCR1 enabled when FWE = 1, to bits ESU1, PSU1, EV1, and PV1 when FWE = 1 and SWE1 = 1, to bit E1 when FWE = 1, SWE1 = 1 and ESU1 = 1, and to bit P1 when FWE = 1, SWE1 = 1, and PSU1 = 1.

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memory programming/erasing.

Bit 7

FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Bit 6—Software Write Enable Bit 1 (SWE1): Enables or disables flash memory programming and erasing. Set this bit when setting bits 5 to 0, bits 7 to 0 of EBR1, and bits 3 to 0 of EBR2.

Bit 6

SWE1	Description	
0	Writes disabled	(Initial value)
1	Writes enabled [Setting condition] When FWE = 1	

Bit 5—Erase Setup Bit 1 (ESU1): Prepares for a transition to erase mode. Set this bit to 1 before setting the E1 bit in FLMCR1 to 1. Do not set the SWE1, PSU1, EV1, PV1, E1, or P1 bit at the same time.

Bit 5

ESU1	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When FWE = 1 and SWE1 = 1	

Bit 4—Program Setup Bit 1 (PSU1): Prepares for a transition to program mode. Set this bit to 1 before setting the P1 bit in FLMCR1 to 1. Do not set the SWE1, ESU1, EV1, PV1, E1, or P1 bit at the same time.

Bit 4

PSU1	Description	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When FWE = 1 and SWE1 = 1	

Bit 3—Erase-Verify 1 (EV1): Selects erase-verify mode transition or clearing. Do not set the SWE1, ESU1, PSU1, PV1, E1, or P1 bit at the same time.

Bit 3

EV1	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE1 = 1	

Bit 2—Program-Verify 1 (PV1): Selects program-verify mode transition or clearing. Do not set the SWE1, ESU1, PSU1, EV1, E1, or P1 bit at the same time.

Bit 2

PV1	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE1 = 1	

Bit 1—Erase 1 (E1): Selects erase mode transition or clearing. Do not set the SWE1, ESU1, PSU1, EV1, PV1, or P1 bit at the same time.

Bit 1

E1	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE1 = 1, and ESU1 = 1	

Bit 0—Program 1 (P1): Selects program mode transition or clearing. Do not set the SWE1, PSU1, ESU1, EV1, PV1, or E1 bit at the same time.

Bit 0

P1	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE1 = 1, and PSU1 = 1	

15.7.2 Flash Memory Control Register 2 (FLMCR2)

Bit:	7	6	5	4	3	2	1	0
	FLER	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

Note: FLMCR2 is a read-only register, and should not be written to.

FLMCR2 is an 8-bit register used for flash memory operating mode control. FLMCR2 is initialized to H'00 by a power-on reset, and in hardware standby mode and software standby mode. When on-chip flash memory is disabled, a read will return H'00.

Bit 7—Flash Memory Error (FLER): 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.

Bit 7

FLER	Description	
0	Flash memory is operating normally Flash memory program/erase protection (error protection) is disabled [Clearing condition] Power-on reset or hardware standby mode	(Initial value)
1	An error has occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enabled [Setting condition] See 15.10.3 Error Protection	

Bits 6 to 0—Reserved: These bits always read 0.

15.7.3 Erase Block Register 1 (EBR1)

Bit:	7	6	5	4	3	2	1	0
	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EBR1 is an 8-bit register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a power-on reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE1 bit in FLMCR1 is not set. When a bit in EBR1 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit, as this will cause all the bits in both EBR1 and EBR2 to be automatically cleared to 0. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 15-5.

15.7.4 Erase Block Register 2 (EBR2)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	EB9	EB8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EBR2 is initialized to H'00 by a power-on reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin. Bit 0 will be initialized to 0 if bit SWE1 of FLMCR1 is not set, even though a high level is input to pin FWE. When a bit in EBR2 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit, as this will cause all the bits in both EBR1 and EBR2 to be automatically cleared to 0. Bits 7 to 2 are reserved and must only be written with 0. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 15-6.

Table 15-6 Flash Memory Erase Blocks

Block (Size)	Addresses
EB0 (1 kbyte)	H'000000–H'0003FF
EB1 (1 kbyte)	H'000400–H'0007FF
EB2 (1 kbyte)	H'000800–H'000BFF
EB3 (1 kbyte)	H'000C00–H'000FFF
EB4 (28 kbytes)	H'001000–H'007FFF
EB5 (16 kbytes)	H'008000–H'00BFFF
EB6 (8 kbytes)	H'00C000–H'00DFFF
EB7 (8 kbytes)	H'00E000–H'00FFFF
EB8 (32 kbytes)	H'010000–H'017FFF
EB9 (32 kbytes)	H'018000–H'01FFFF

15.7.5 RAM Emulation Register (RAMER)

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

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER initialized to H'00 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode. RAMER settings should be made in user mode or user program mode.

Flash memory area divisions are shown in table 15-7. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bits 7 to 5—Reserved: These bits always read 0.

Bit 4—Reserved: Only 0 may be written to these bits.

Bit 3—RAM Select (RAMS): Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, all flash memory block are program/erase-protected.

Bit 3

RAMS	Description	
0	Emulation not selected	(Initial value)
	Program/erase-protection of all flash memory blocks is disabled	
1	Emulation selected	
	Program/erase-protection of all flash memory blocks is enabled	

Bit 2—Reserved: Only 0 should be written to this bit.

Bits 1 and 0—Flash Memory Area Selection: These bits are used together with bit 3 to select the flash memory area to be overlapped with RAM. (See table 15-7.)

Table 15-7 Flash Memory Area Divisions

Addresses	Block Name	RAMS	RAM1	RAM0
H'FFD000–H'FFD3FF	RAM area 1 kbyte	0	*	*
H'000000–H'0003FF	EB0 (1 kbyte)	1	0	0
H'000400–H'0007FF	EB1 (1 kbyte)	1	0	1
H'000800–H'000BFF	EB2 (1 kbyte)	1	1	0
H'000C00–H'000FFF	EB3 (1 kbyte)	1	1	1

*: Don't care

15.7.6 Serial Control Register X (SCRX)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	FLSHE	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCRX is an 8-bit readable/writable register that performs register access control, and on-chip flash memory control (including the F-ZTAT version).

SCRX is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 to 4—Reserved: Only 0 should be written to these bits.

Bit 3—Flash Memory Control Register Enable (FLSHE): Controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2). Setting the FLSHE bit to 1 enables read/write access to the flash memory control registers. If FLSHE is cleared to 0, the flash memory control registers are deselected. In this case, the flash memory control register contents are retained. When the FLSHE bit is set to 1, the flash memory control registers can be read and written to. When FLSHE is cleared to 0, the flash memory control registers are deselected. In this case, the contents of the flash memory control registers are retained.

Bit 3

FLSHE	Description	
0	Flash control registers deselected in area H'FFFFA8 to H'FFFFAC	(Initial value)
1	Flash control registers selected in area H'FFFFA8 to H'FFFFAC	

Bits 2 to 0—Reserved: Only 0 should be written to these bits.

15.8 On-Board Programming Modes

When pins are set to on-board programming mode and a reset-start is executed, a transition is made to the on-board programming state in which program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 15-8. For a diagram of the transitions to the various flash memory modes, see figure 15-3.

Table 15-8 Setting On-Board Programming Modes

Mode		FWE	MD2	MD1	MD0
Boot mode	Expanded mode	1	0	1	0
	Single-chip mode		0	1	1
User program mode	Expanded mode	1	1	1	0
	Single-chip mode		1	1	1

15.8.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The SCI channel to be used is set to asynchronous mode.

When a reset-start is executed after the H8S/2214's pins have been set to boot mode, the boot program built into the H8S/2214 is started and the programming control program prepared in the host is serially transmitted to the H8S/2214 via the SCI. In the H8S/2214, the programming control program received via the SCI is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address of the programming control program area and the programming control program execution state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 15-9, and the boot mode execution procedure in figure 15-10.

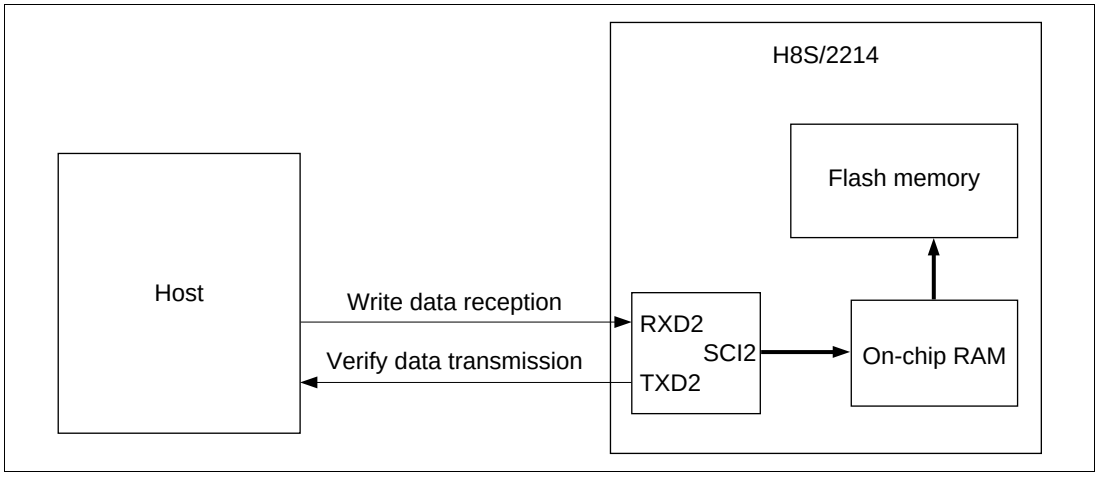
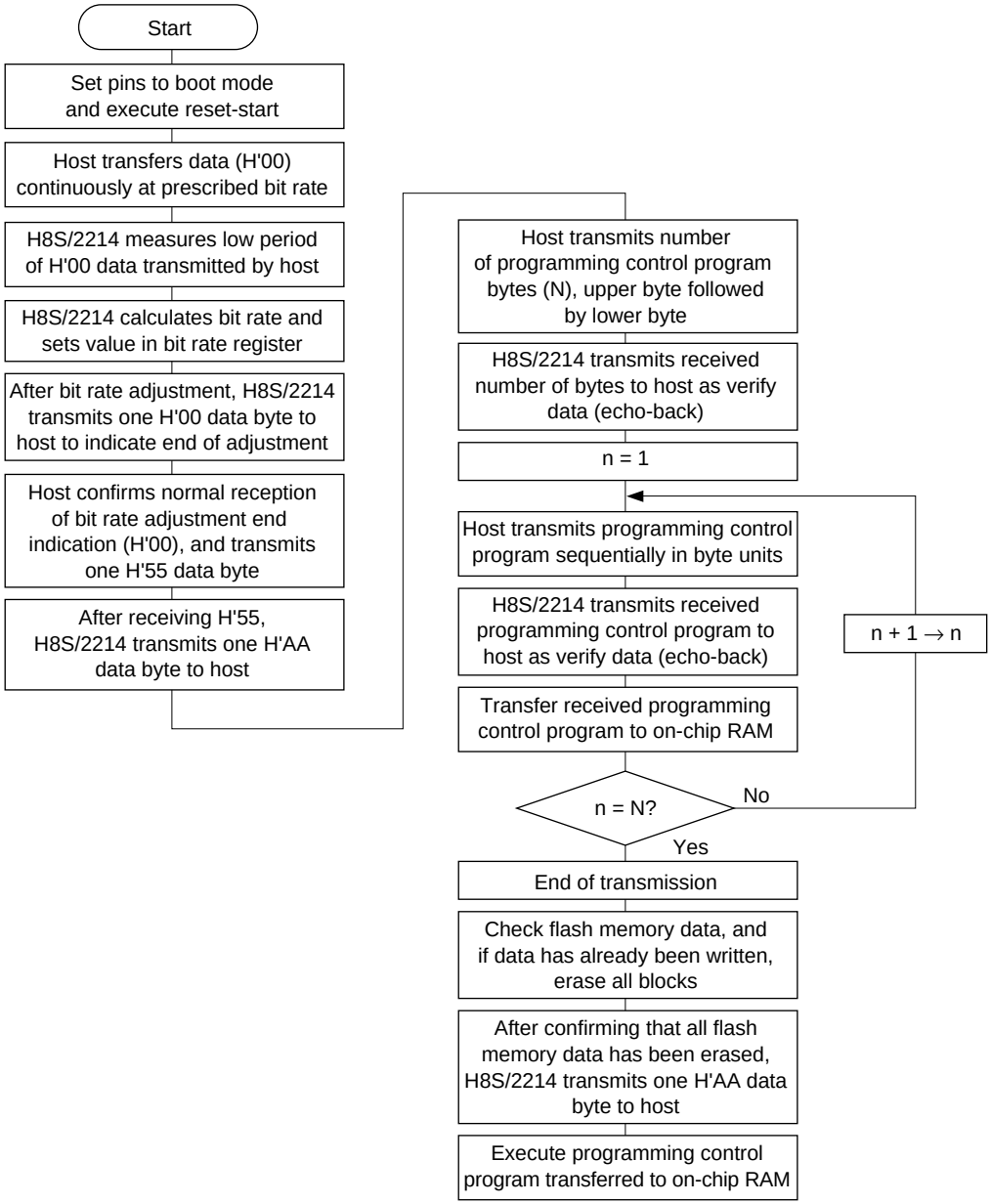


Figure 15-9 System Configuration in Boot Mode

If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error indication, and the erase operation and subsequent operations are halted. When a transition is made to boot mode, or from boot mode to another mode, mode switching must be carried out by means of $\overline{\text{RES}}$ input. The states of ports with multiplexed address functions and bus control output signals ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$) change during the switchover period (while a low level is being input at the $\overline{\text{RES}}$ pin), and therefore these pins should not be used for output signals during this period.



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operations are halted.

Figure 15-10 Boot Mode Execution Procedure

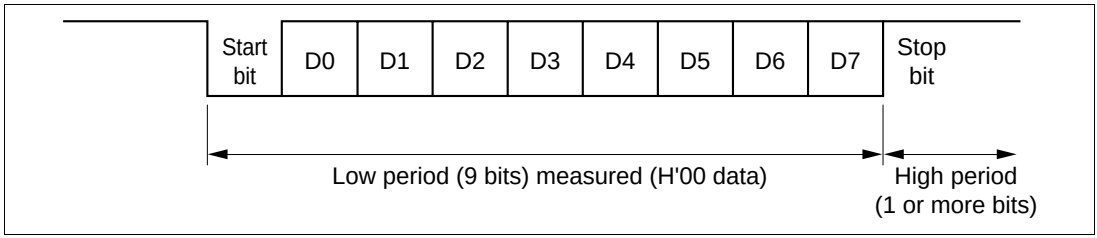


Figure 15-11 Automatic SCI Bit Rate Adjustment

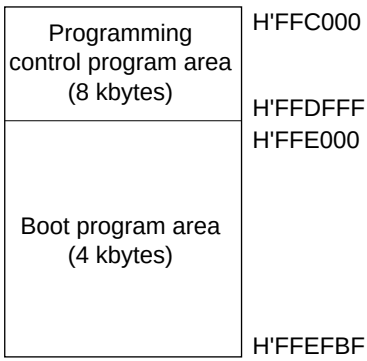
When boot mode is initiated, the H8S/2214 measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The H8S/2214 calculates the bit rate of the transmission from the host from the measured low period, and 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 H8S/2214. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the H8S/2214's system clock frequency, there will be a discrepancy between the bit rates of the host and the H8S/2214. Set the host transfer bit rate at 4800, 9600, or 19,200 bps to operate the SCI properly.

Table 15-9 shows host transfer bit rates and system clock frequencies for which automatic adjustment of the H8S/2214 bit rate is possible. The boot program should be executed within this system clock range.

Table 15-9 System Clock Frequencies for which Automatic Adjustment of H8S/2214 Bit Rate is Possible

Host Bit Rate	System Clock Frequency for Which Automatic Adjustment of LSI Bit Rate is Possible
4,800 bps	2 MHz to 16 MHz
9,600 bps	4 MHz to 16 MHz
19,200 bps	8 MHz to 16 MHz

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the RAM area is divided into an area used by the boot program and an area to which the programming control program is transferred via the SCI, as shown in figure 15-12. The boot program area cannot be used until the execution state in boot mode switches to the programming control program transferred from the host.



Note: The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note also that the boot program remains in this area of the on-chip RAM even after control branches to the programming control program.

Figure 15-12 RAM Areas in Boot Mode

Notes on Use of Boot Mode:

- When the chip comes out of reset in boot mode, it measures the low-level period of the input at the SCI's RxD2 pin. The reset should end with RxD2 high. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period of the RxD2 pin.
- In boot mode, if any data has been programmed into the flash memory (if all data is not 1), all flash memory blocks are erased. Boot mode is for use when user program mode is unavailable, such as the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.
- Interrupts cannot be used while the flash memory is being programmed or erased.
- The RxD2 and TxD2 pins should be pulled up on the board.
- Before branching to the programming control program (RAM area H'FFC000), the chip terminates transmit and receive operations by the on-chip SCI (channel 2) (by clearing the RE

and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. The transmit data output pin, TxD2, goes to the high-level output state (PA1DDR = 1, PA1DR = 1).

The contents of the CPU's internal general registers are undefined at this time, so these registers must be initialized immediately after branching to the programming control program. In particular, since the stack pointer (SP) is used implicitly in subroutine calls, etc., a stack area must be specified for use by the programming control program.

Initial settings must also be made for all other on-chip registers.

- Boot mode can be entered by making the pin settings shown in table 15-8 and executing a reset-start.

  Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the FWE pin and mode pins, and executing reset release*[1]. Boot mode can also be cleared by a WDT overflow reset.

  Do not change the mode pin input levels in boot mode, and do not drive the FWE pin low while the boot program is being executed or while flash memory is being programmed or erased*[2].

- If the mode pin input levels are changed (for example, from low to high) during a reset, the state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$) will change according to the change in the microcomputer's operating mode*[3].

  Therefore, care must be taken to make pin settings to prevent these pins from becoming output signal pins during a reset, or to prevent collision with signals outside the microcomputer.

Notes: *1 Mode pin and FWE pin input must satisfy the mode programming setup time (t_{MDS} = 200 ns) with respect to the reset release timing.

*2 For further information on FWE application and disconnection, see section 15.15, Flash Memory Programming and Erasing Precautions.

*3 See appendix D, Pin States.

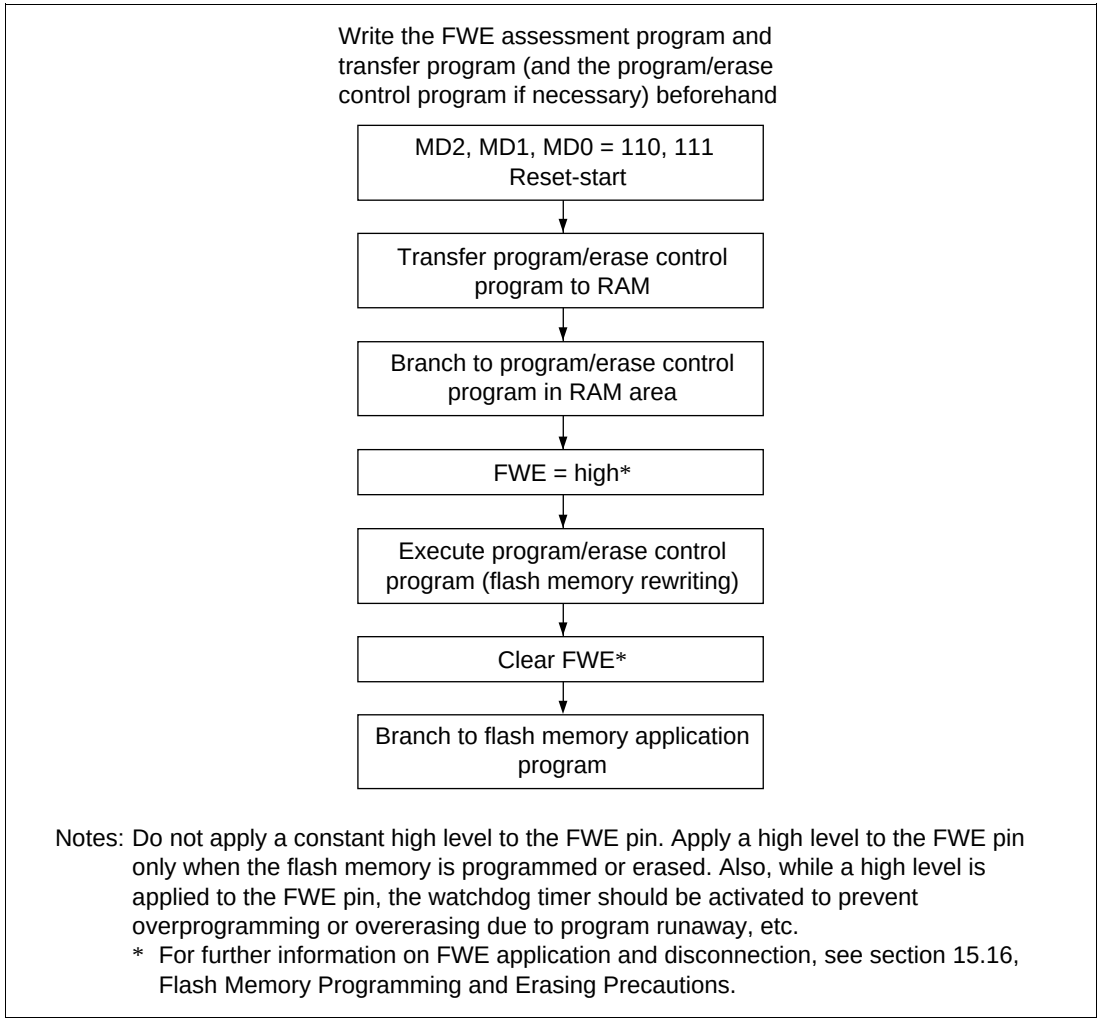
15.8.2 User Program Mode

When set to user program mode, the chip can program and erase its flash memory by executing a user program/erase control program. Therefore, on-board reprogramming of the on-chip flash memory can be carried out by providing on-board means of FWE control and supply of programming data, and storing a program/erase control program in part of the program area as necessary.

To select user program mode, select a mode that enables the on-chip flash memory (mode 6 or 7), and apply a high level to the FWE pin. In this mode, on-chip supporting modules other than flash memory operate as they normally would in modes 6 and 7.

The flash memory itself cannot be read while the SWE bit is set to 1 to perform programming or erasing, so the control program that performs programming and erasing should be run in on-chip RAM or external memory.

Figure 15-13 shows the procedure for executing the program/erase control program when transferred to on-chip RAM.



Notes: Do not apply a constant high level to the FWE pin. Apply a high level to the FWE pin only when the flash memory is programmed or erased. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

* For further information on FWE application and disconnection, see section 15.16, Flash Memory Programming and Erasing Precautions.

Figure 15-13 User Program Mode Execution Procedure

15.9 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes are made by setting the PSU1, ESU1, P1, E1, PV1, and EV1 bits in FLMCR1 for addresses H'000000 to H'01FFFF.

The flash memory cannot be read while it is being written or erased. Install the program to control flash memory programming and erasing (programming control program) in the on-chip RAM, in external memory, and execute the program from there.

Notes: 1. Operation is not guaranteed if bits SWE1, ESU1, PSU1, EV1, PV1, E1, and P1 of FLMCR1 are set/reset by a program in flash memory in the corresponding address areas.
2. When programming or erasing, set FWE to 1 (programming/erasing will not be executed if FWE = 0).
3. Programming should be performed in the erased state. Do not perform additional programming on previously programmed addresses.

15.9.1 Program Mode

Follow the procedure shown in the program/program-verify flowchart in figure 15-10 to write data or programs to flash memory. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 128 bytes at a time.

For the wait times (t_{sswe}, t_{spsu}, t_{sp10}, t_{sp30}, t_{sp200}, t_{cp}, t_{cpsu}, t_{spv}, t_{spvr}, t_{cpv}, t_{cswe}) after bits are set or cleared in flash memory control register 1 (FLMCR1) and the maximum number of programming operations (N), see section 18.6, Flash Memory Characteristics.

Following the elapse of t_{sswe} µs or more after the SWE1 bit is set to 1 in flash memory control register 1 (FLMCR1), 128-byte data is stored in the program data area and reprogram data area, and the 128-byte data in the program data area in RAM is written consecutively to the write addresses. The lower 8 bits of the first address written to must be H'00 or H'80. 128 consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.

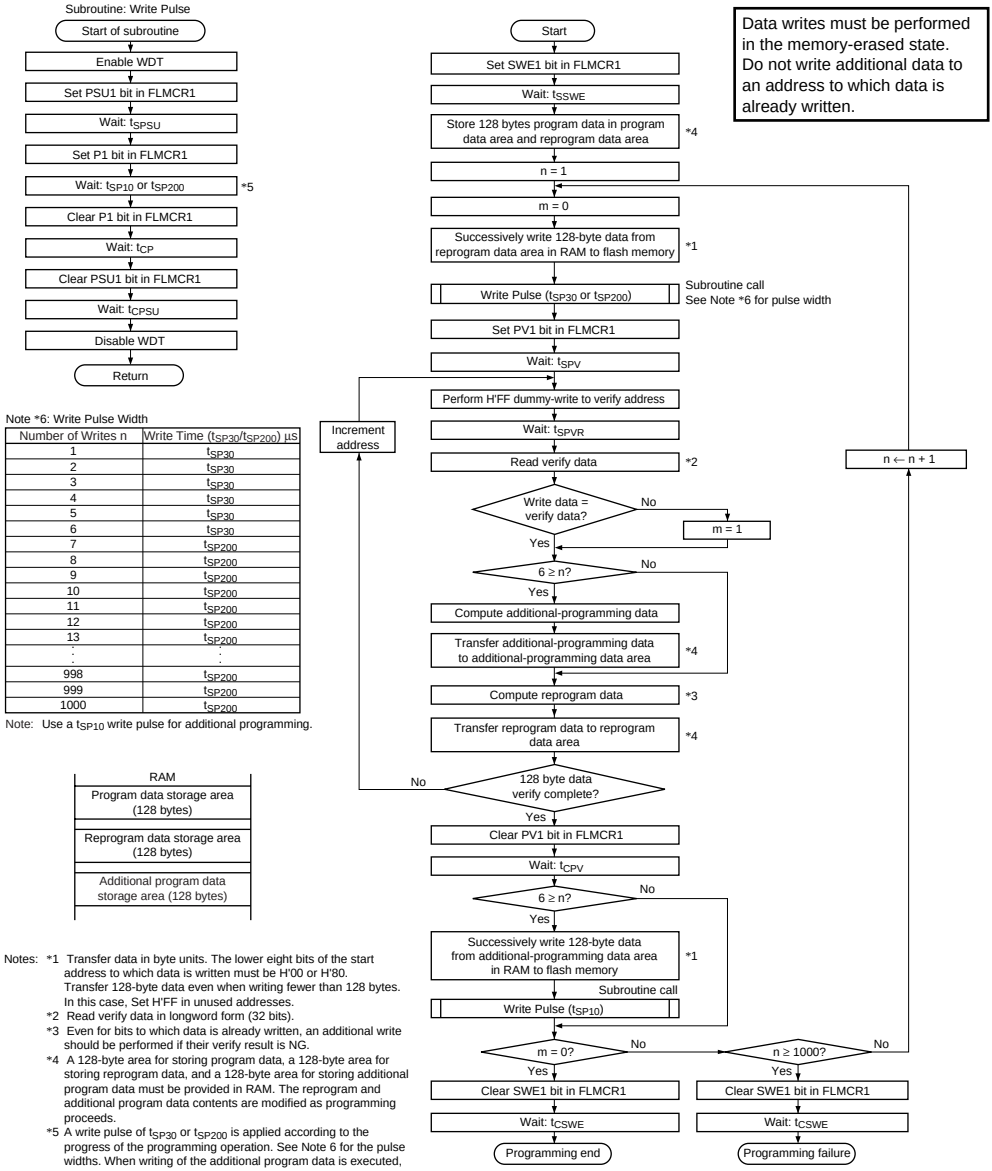
Next, the watchdog timer is set to prevent overprogramming in the event of program runaway, etc. Set a value greater than (t_{spsu} + t_{sp200} + t_{cp} + t_{cpsu}) µs as the WDT overflow period. After this, preparation for program mode (program setup) is carried out by setting the PSU1 bit in FLMCR1, and after the elapse of t_{spsu} µs or more, the operating mode is switched to program mode by

setting the P1 bit in FLMCR1. The time during which the P1 bit is set is the flash memory programming time. Set the programming time according to the table in the programming flowchart.

15.9.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of a given programming time, the programming mode is exited (the P1 bit in FLMCR1 is cleared, then the PSU1 bit is cleared at least t_{cp} µs later). The watchdog timer is cleared after the elapse of t_{cpsu} µs or more, and the operating mode is switched to program-verify mode by setting the PV1 bit in FLMCR1. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of t_{spv} µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least t_{spvr} µs after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 15-14) and transferred to the reprogram data area. After 128 bytes of data have been verified, exit program-verify mode, wait for at least t_{cpv} µs, then clear the SWE1 bit in FLMCR1 to 0. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. However, ensure that the program/program-verify sequence is not repeated more than (N) times on the same bits.



Note *6: Write Pulse Width

Number of Writes n	Write Time (t_{SP30}/t_{SP200}) µs
1	t_{SP30}
2	t_{SP30}
3	t_{SP30}
4	t_{SP30}
5	t_{SP30}
6	t_{SP30}
7	t_{SP200}
8	t_{SP200}
9	t_{SP200}
10	t_{SP200}
11	t_{SP200}
12	t_{SP200}
13	t_{SP200}
⋮	⋮
998	t_{SP200}
999	t_{SP200}
1000	t_{SP200}

Note: Use a t_{SP10} write pulse for additional programming.

Notes: *1 Transfer data in byte units. The lower eight bits of the start address to which data is written must be H'00 or H'80. Transfer 128-byte data even when writing fewer than 128 bytes. In this case, Set H'FF in unused addresses.
*2 Read verify data in longword form (32 bits).
*3 Even for bits to which data is already written, an additional write should be performed if their verify result is NG.
*4 A 128-byte area for storing program data, a 128-byte area for storing reprogram data, and a 128-byte area for storing additional program data must be provided in RAM. The reprogram and additional program data contents are modified as programming proceeds.
*5 A write pulse of t_{SP30} or t_{SP200} is applied according to the progress of the programming operation. See Note 6 for the pulse widths. When writing of the additional program data is executed, a t_{SP10} write pulse should be applied. Reprogram data X' means reprogram data when the pulse is applied.

Reprogram Data Computation Table

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming complete.
0	1	0	Programming is incomplete; reprogramming should be performed.
1	0	1	—
1	1	1	Left in the erased state.

Additional-Programming Data Computation Table

Reprogram Data (X')	Verify Data (V)	Additional-Programming Data (Y)	Comments
0	0	0	Additional programming executed
0	1	1	Additional programming not executed
1	0	1	Additional programming not executed
1	1	1	Additional programming not executed

Figure 15-14 Program/Program-Verify Flowchart

15.9.3 Erase Mode

Flash memory erasing should be performed block by block following the procedure shown in the erase/erase-verify flowchart shown in figure 15-15.

For the wait times (t_{sswe}, t_{sesu}, t_{se}, t_{ce}, t_{cesu}, t_{sev}, t_{sevr}, t_{cev}, t_{cswe}) after bits are set or cleared in flash memory control register 1 (FLMCR1) and the maximum number of erase operations (N), see section 18.6, Flash Memory Characteristics.

To perform data or program erasure, make a 1-bit setting for the flash memory area to be erased in erase block register 1 or 2 (EBR1 or EBR2) at least t_{sswe} µs after setting the SWE1 bit to 1 in flash memory control register 1 (FLMCR1). Next, set up the watchdog timer to prevent overerasing in the event of program runaway, etc. Set a value greater than ($t_{sesu} + t_{se} + t_{ce} + t_{cesu}$) µs as the WDT overflow period. After this, preparation for erase mode (erase setup) is carried out by setting the ESU1 bit in FLMCR1, and after the elapse of t_{sesu} µs or more, the operating mode is switched to erase mode by setting the E1 bit in FLMCR1. The time during which the E1 bit is set is the flash memory erase time. Ensure that the erase time does not exceed t_{se} ms.

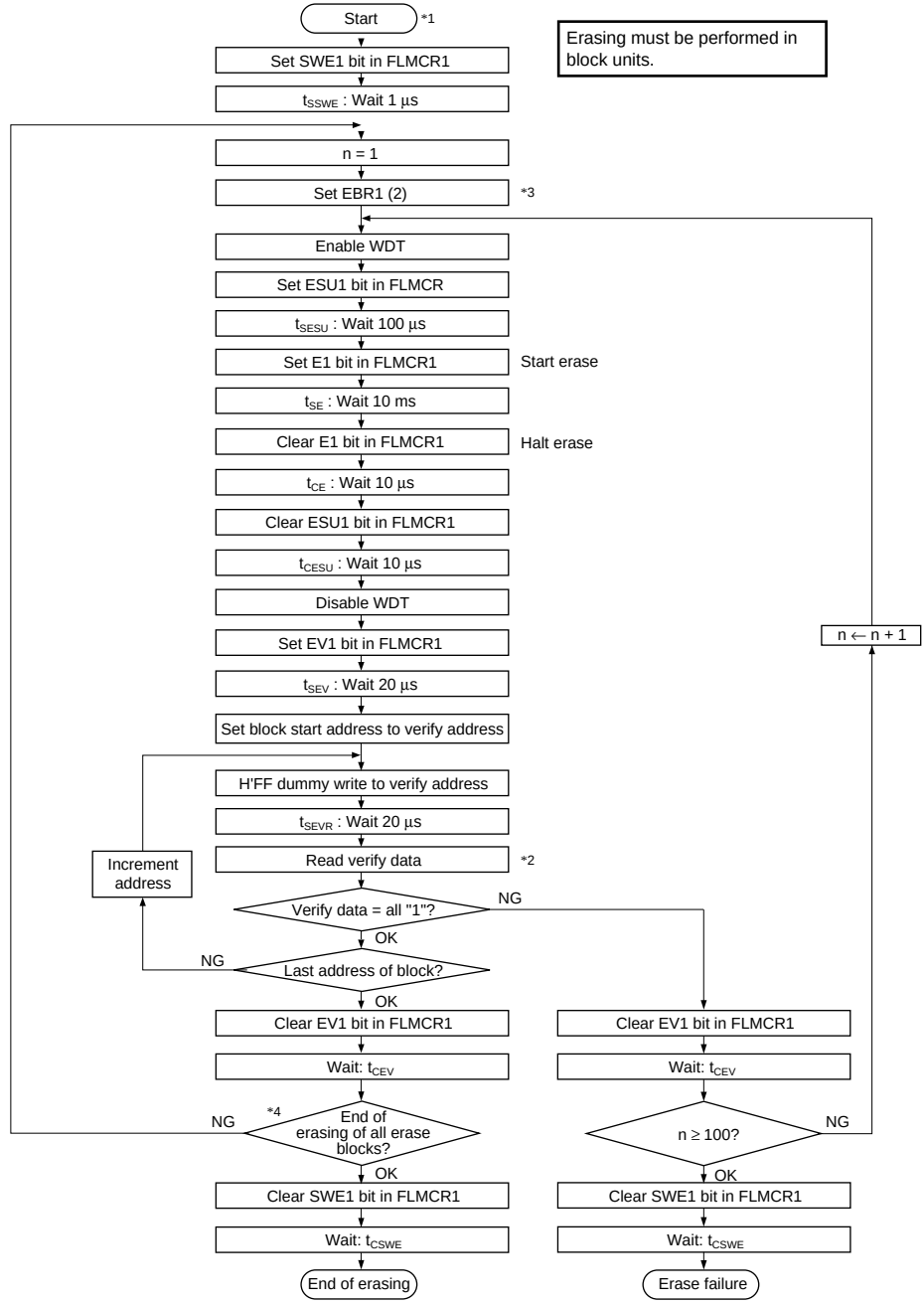
Note: With flash memory erasing, prewriting (setting all data in the memory to be erased to 0) is not necessary before starting the erase procedure.

15.9.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the erase time, erase mode is exited (the E1 bit in FLMCR1 is cleared to 0, then the ESU1 bit is cleared to 0 at least t_{ce} µs later), the watchdog timer is cleared after the elapse of t_{cesu} µs or more, and the operating mode is switched to erase-verify mode by setting the EV1 bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of t_{sev} µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least t_{sevr} µs after the dummy write before performing this read operation. If the read data has been erased (all 1), execute a dummy write to the next address, and perform an erase-verify. If the read data has not been erased, set erase mode again and repeat the erase/erase-verify sequence as before. However, ensure that the erase/erase-verify sequence is not repeated more than (N) times. When verification is completed, exit erase-verify mode, and wait for at least t_{cev} µs. If erasure has been completed on all the erase blocks, clear the SWE1 bit in FLMCR1. If there are any unerased blocks, make a 1-bit setting for the flash memory block to be erased, and repeat the erase/erase-verify sequence as before.



Notes: *1 Preprogramming (setting erase block data to all "0") is not necessary.
*2 Verify data is read in 32-bit (longword) units.
*3 Set only one bit in EBR1 (2). More than one bit cannot be set.
*4 Erasing is performed in block units. To erase a number of blocks, each block must be erased in turn.

Figure 15-15 Erase/Erase-Verify Flowchart

15.10 Protection

There are three kinds of flash memory program/erase protection: hardware protection, software protection, and error protection.

15.10.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2). The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained in the error-protected state. (See table 15-10.)

Table 15-10 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1, FLMCR2, (except bit FLER) EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Yes	Yes
Reset/standby protection	• In a power-on reset (including a WDT power-on reset) and in standby mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • 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 $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

15.10.2 Software Protection

Software protection can be implemented by setting the SWE1 bit in FLMCR1, erase block register 1 (EBR1), erase block register 2 (EBR2), and the RAMS bit in the RAM emulation register (RAMER). When software protection is in effect, setting the P1 or E1 bit in flash memory control register 1 (FLMCR1), does not cause a transition to program mode or erase mode. (See table 15-11.)

Table 15-11 Software Protection

Item	Description	Functions	
		Program	Erase
SWE bit protection	• Setting bit SWE1 in FLMCR1 to 0 will place area H'000000 to H'01FFFF in the program/erase-protected state. (Execute the program in the on-chip RAM, external memory)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) and erase block register 2 (EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

15.10.3 Error Protection

In error protection, an error is detected when H8S/2214 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 aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the H8S/2214 malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P1 or E1 bit. However, PV1 and EV1 bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

1. When the flash memory of the relevant address area is read during programming/erasing (including vector read and instruction fetch)
2. Immediately after exception handling (excluding a reset) during programming/erasing
3. When a SLEEP instruction (including software standby) is executed during programming/erasing
4. When the CPU releases the bus to the DTC during programming/erasing.

Error protection is released only by a power-on reset and in hardware standby mode.

Figure 15-16 shows the flash memory state transition diagram.

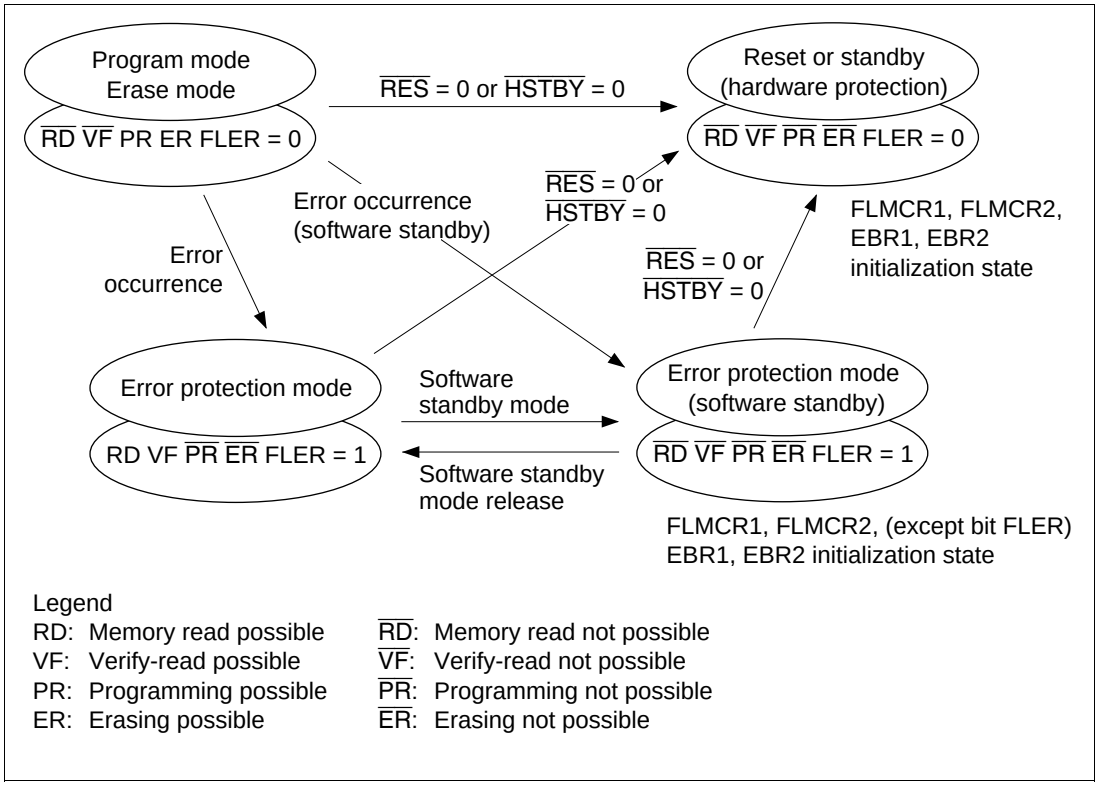


Figure 15-16 Flash Memory State Transitions

15.11 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses can be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 15-17 shows an example of emulation of real-time flash memory programming.

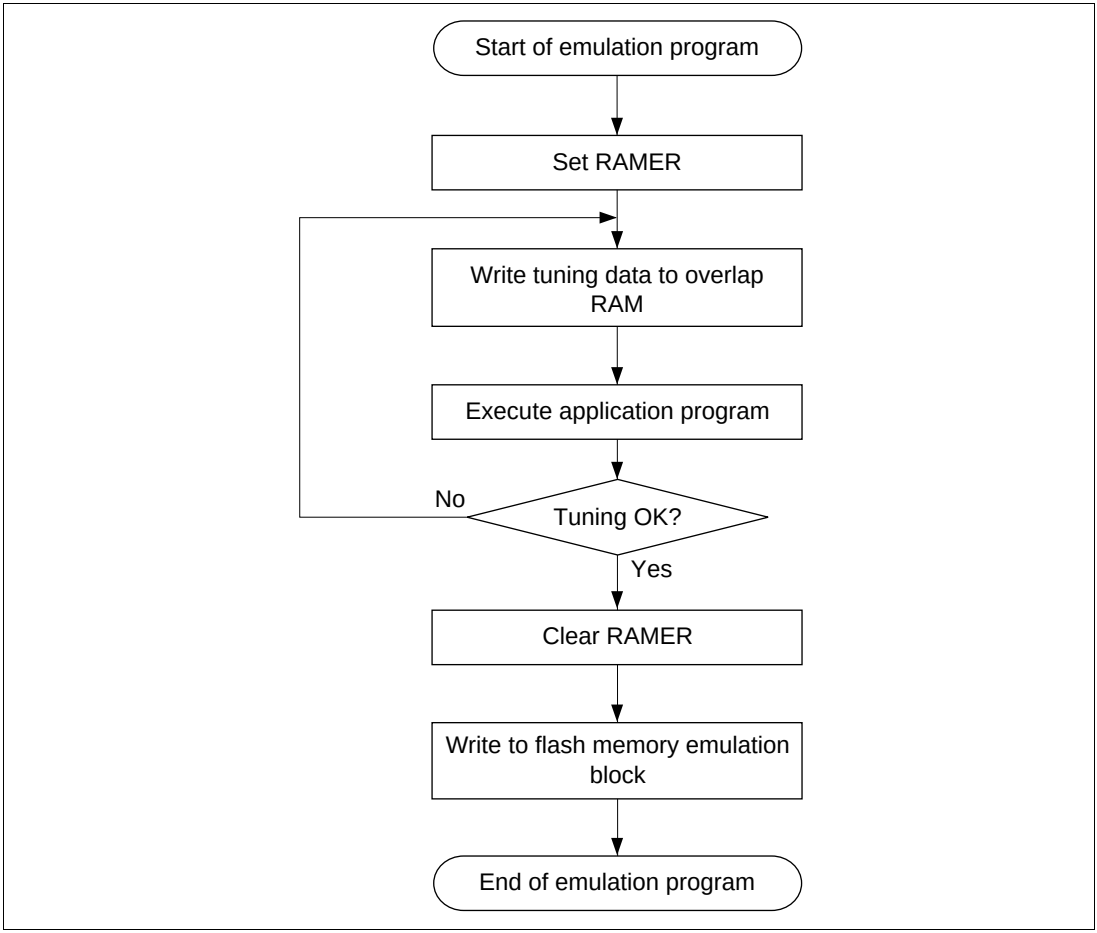


Figure 15-17 Flowchart for Flash Memory Emulation in RAM

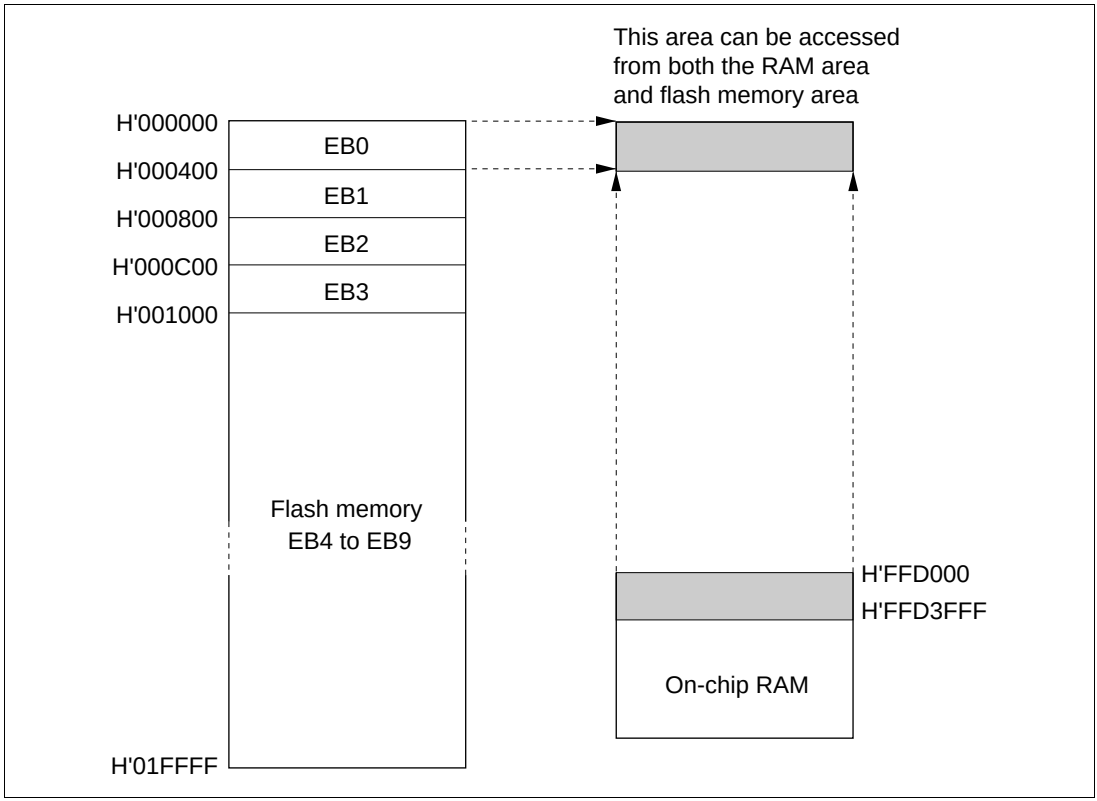


Figure 15-18 Example of RAM Overlap Operation

Example in which Flash Memory Block Area EB0 is Overlapped

1. Set bits RAMS, RAM1 to RAM0 in RAMER to 1, 0, 0, 0, to overlap part of RAM onto the area (EB0) for which real-time programming is required.
2. Real-time programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB0).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM1 to RAM0 (emulation protection). In this state, setting the P1 or E1 bit in flash memory control register 1 (FLMCR1), will not cause a transition to program mode or erase mode. When actually programming or erasing a flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

15.12 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI interrupt is disabled when flash memory is being programmed or erased (when the P1 or E1 bit is set in FLMCR1), and while the boot program is executing in boot mode*[1], to give priority to the program or erase operation. 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. In the interrupt exception handling sequence during programming or erasing, the vector would not be read correctly*[2], possibly resulting in MCU runaway.
3. If interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabling interrupt, as an exception to the general rule. However, this provision does not guarantee normal erasing and programming or MCU operation. All requests, including NMI interrupt, must therefore be restricted inside and outside the MCU when programming or erasing flash memory. NMI interrupt is also disabled in the error-protection state while the P1 or E1 bit remains set in FLMCR1.

Notes: *1 Interrupt requests must be disabled inside and outside the MCU until the programming control program has completed programming.

*2 The vector may not be read correctly in this case for the following two reasons:

- If flash memory is read while being programmed or erased (while the P1 or E1 bit is set in FLMCR1), correct read data will not be obtained (undetermined values will be returned).
- If the interrupt entry in the vector table has not been programmed yet, interrupt exception handling will not be executed correctly.

15.13 Flash Memory Programmer Mode

Programs and data can be written and erased in programmer mode as well as in the on-board programming modes. In programmer mode, flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

In programmer mode, set the mode pins to programmer mode (see table 15-12) and input a 12 MHz input clock.

Table 15-12 shows the pin settings for programmer mode. For the pin names in programmer mode, see section 1.3.2, Pin Functions in Each Operating Mode.

Table 15-12 Programmer Mode Pin Settings

Pin Names	Settings
Mode pins: MD2, MD1, MD0	Low level input to MD2, MD1, and MD0.
Mode setting pins: PF0, P16, P14, PF3	High level input to PF3, PF0, low level input to P16 and P14
FWE pin	High level input (in auto-program and auto-erase modes)
$\overline{\text{RES}}$ pin	Power-on reset circuit
XTAL, EXTAL pins	Oscillator circuit

15.13.1 Socket Adapter Pin Correspondence Diagram

Connect the socket adapter to the chip as shown in figure 15-20. This will enable conversion to a 40-pin arrangement. The on-chip ROM memory map is shown in figure 15-19, and the socket adapter pin correspondence diagram in figure 15-20.

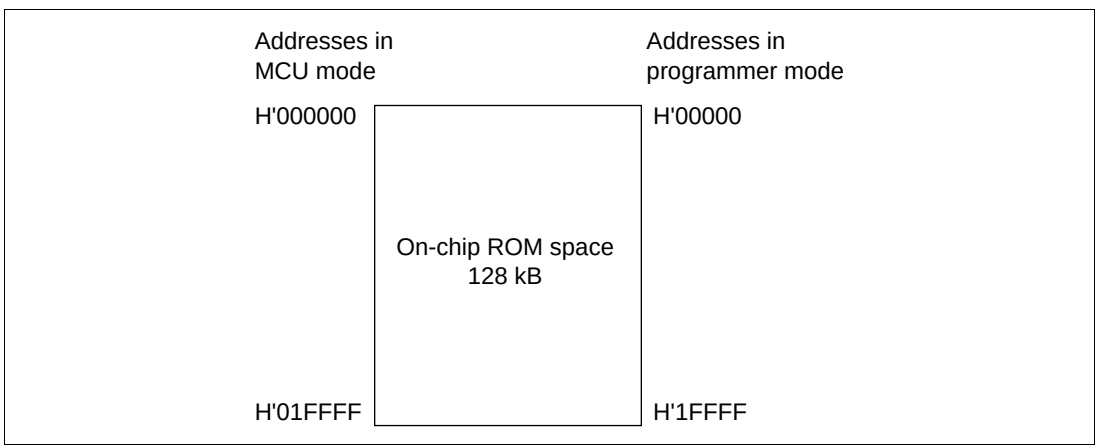


Figure 15-19 On-Chip ROM Memory Map

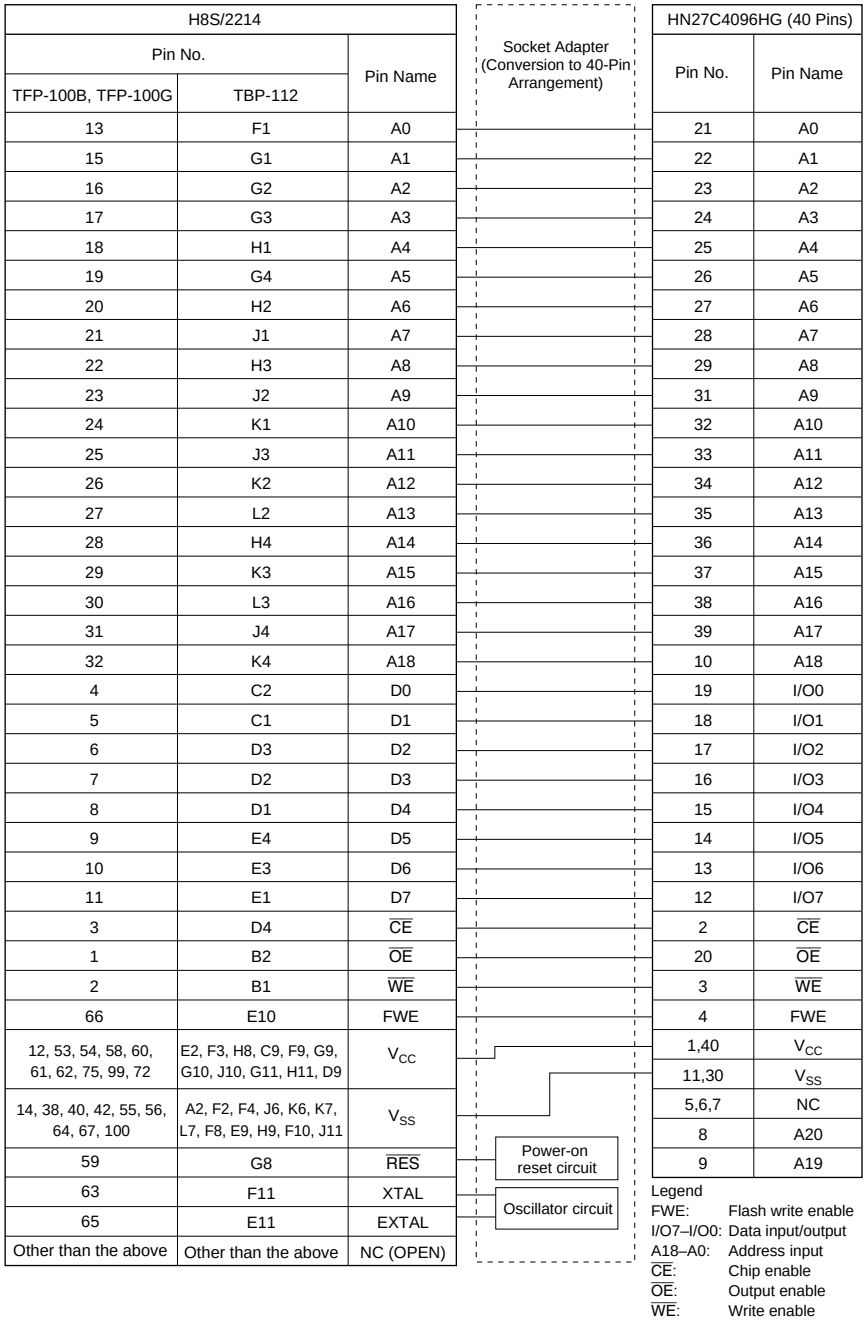


Figure 15-20 Socket Adapter Pin Correspondence Diagram

15.13.2 Programmer Mode Operation

Table 15-13 shows how the different operating modes are set when using programmer mode, and table 15-14 lists the commands used in programmer mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.
- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-programming.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the I/O6 signal. In status read mode, error information is output if an error occurs.

Table 15-13 Settings for Various Operating Modes In Programmer Mode

	Pin Names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**I/O7– I/O0**	**A18–A0**
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-z	X
Command write	H or L	L	H	L	Data input	*Ain
Chip disable	H or L	H	X	X	Hi-z	X

Notes: 1. Chip disable is not a standby state; internally, it is an operation state.
2. *Ain indicates that there is also address input in auto-program mode.
3. For command writes in auto-program and auto-erase modes, input a high level to the FWE pin.

Table 15-14 Programmer Mode Commands

Command Name	Number of Cycles	1st Cycle			2nd Cycle		
		Mode	Address	Data	Mode	Address	Data
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	WA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

Notes: 1. In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

15.13.3 Memory Read Mode

1. After completion of auto-program/auto-erase/status read operations, a transition is made to the command wait state. When reading memory contents, a transition to memory read mode must first be made with a command write, after which the memory contents are read.
2. In memory read mode, command writes can be performed in the same way as in the command wait state.
3. Once memory read mode has been entered, consecutive reads can be performed.
4. After powering on, memory read mode is entered.

Table 15-15 AC Characteristics in Transition to Memory Read Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

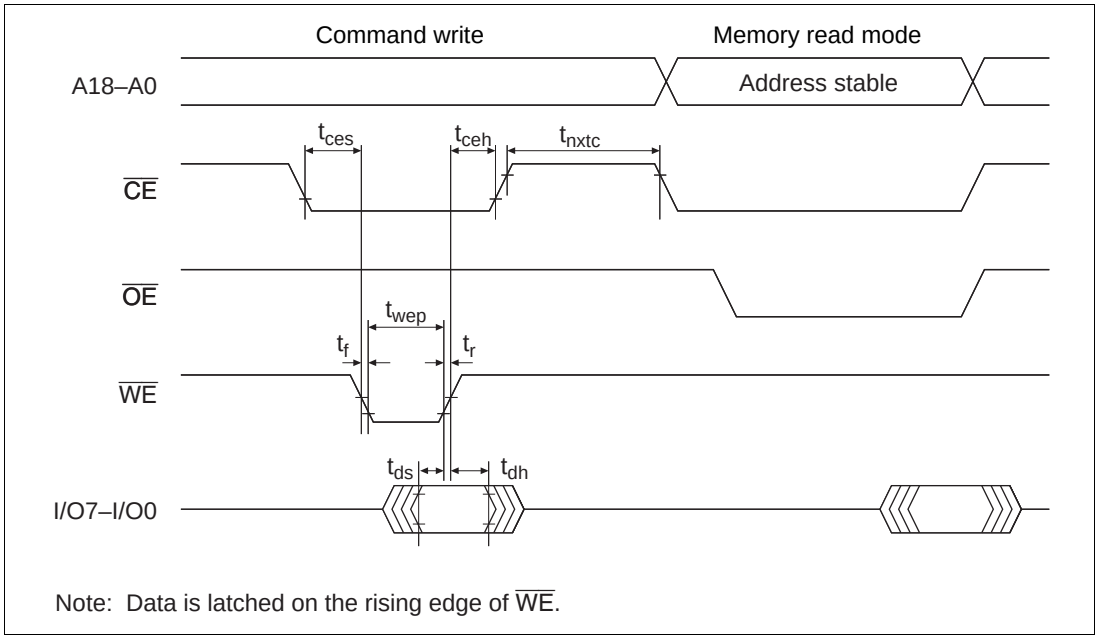
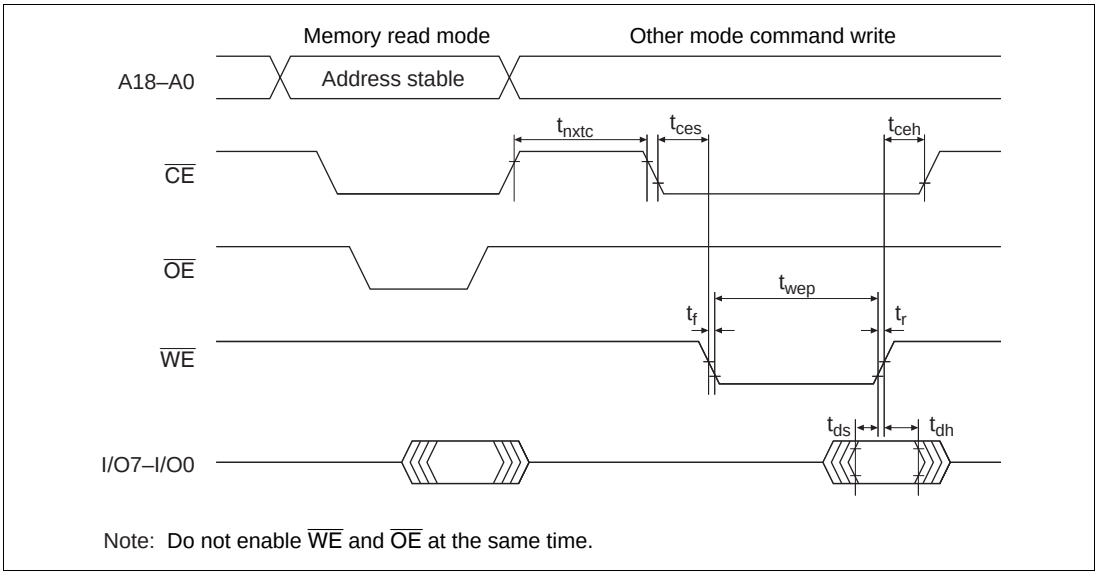


Figure 15-21 Timing Waveforms for Memory Read after Memory Write

Table 15-16 AC Characteristics in Transition from Memory Read Mode to Another Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



Note: Do not enable $\overline{WE}$ and $\overline{OE}$ at the same time.

Figure 15-22 Timing Waveforms in Transition from Memory Read Mode to Another Mode

Table 15-17 AC Characteristics in Memory Read Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Access time	t_{acc}		20	µs	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Output disable delay time	t_{df}		100	ns	
Data output hold time	t_{oh}	5		ns	

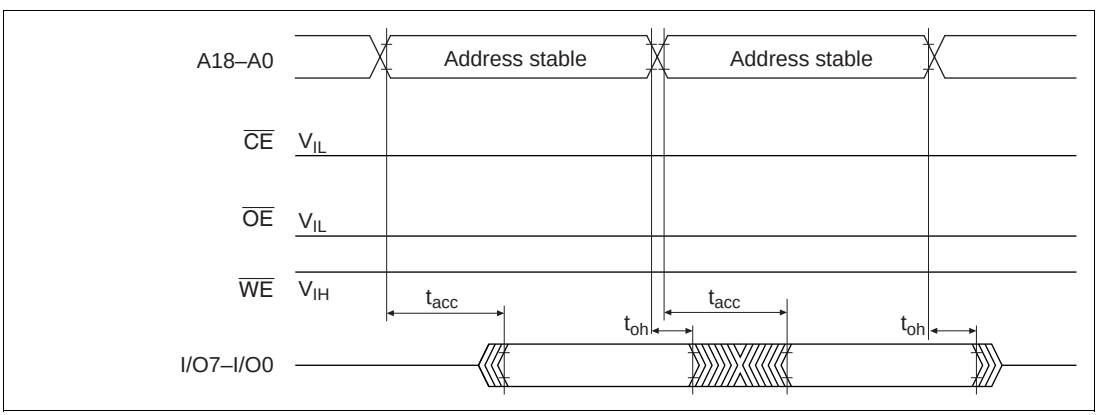


Figure 15-23 $\overline{CE}$ and $\overline{OE}$ Enable State Read Timing Waveforms

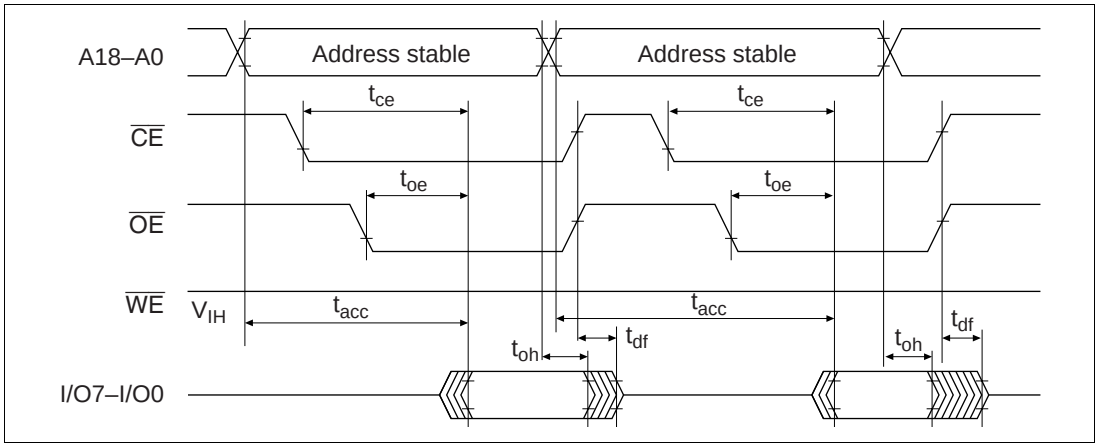


Figure 15-24 $\overline{CE}$ and $\overline{OE}$ Clock System Read Timing Waveforms

15.13.4 Auto-Program Mode

1. In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
2. A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. The lower 7 bits of the transfer address must be low. If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
4. Memory address transfer is performed in the second cycle (figure 15-25). Do not perform transfer after the third cycle.
5. Do not perform a command write during a programming operation.
6. Perform one auto-program operation for a 128-byte block for each address. Two or more additional programming operations cannot be performed on a previously programmed address block.
7. Confirm normal end of auto-programming by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-program operation end decision pin).
8. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 15-18 AC Characteristics in Auto-Program Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
Status polling start time	t_{wsts}	1		ms	
Status polling access time	t_{spa}		150	ns	
Address setup time	t_{as}	0		ns	
Address hold time	t_{ah}	60		ns	
Memory write time	t_{write}	1	3000	ms	
Write setup time	t_{pns}	100		ns	
Write end setup time	t_{pnh}	100		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

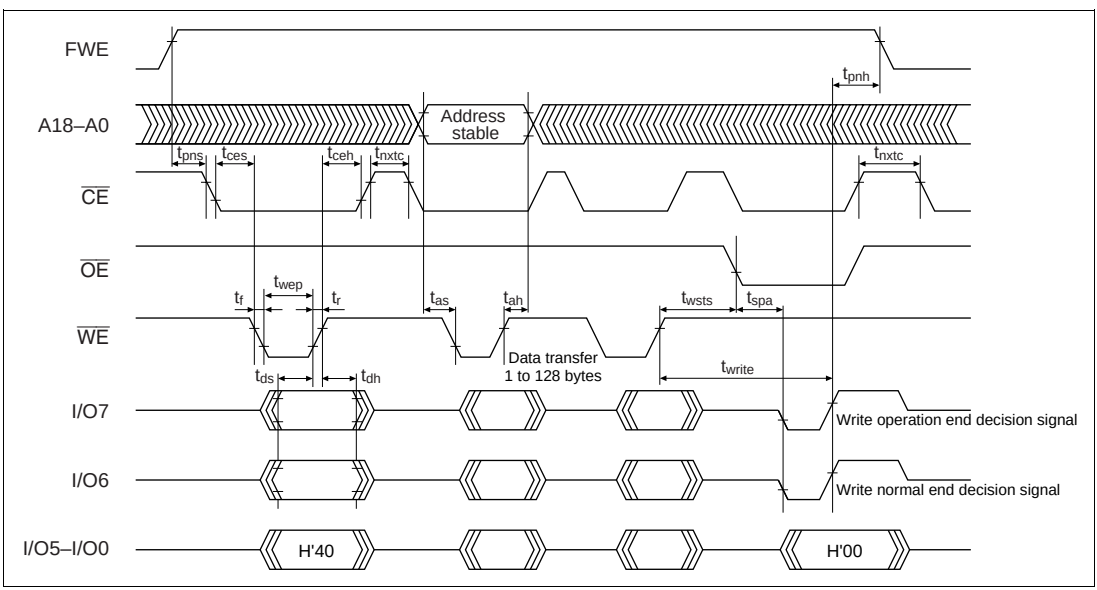


Figure 15-25 Auto-Program Mode Timing Waveforms

15.13.5 Auto-Erase Mode

1. Auto-erase mode supports only entire memory erasing.
2. Do not perform a command write during auto-erasing.
3. Confirm normal end of auto-erasing by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-erase operation end decision pin).
4. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 15-19 AC Characteristics in Auto-Erase Mode (Conditions: V_{CC} = 3.3 V ±3.0 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
Status polling start time	t_{ests}	1		ms	
Status polling access time	t_{spa}		150	ns	
Memory erase time	t_{erase}	100	40000	ms	
Erase setup time	t_{ens}	100		ns	
Erase end setup time	t_{enh}	100		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

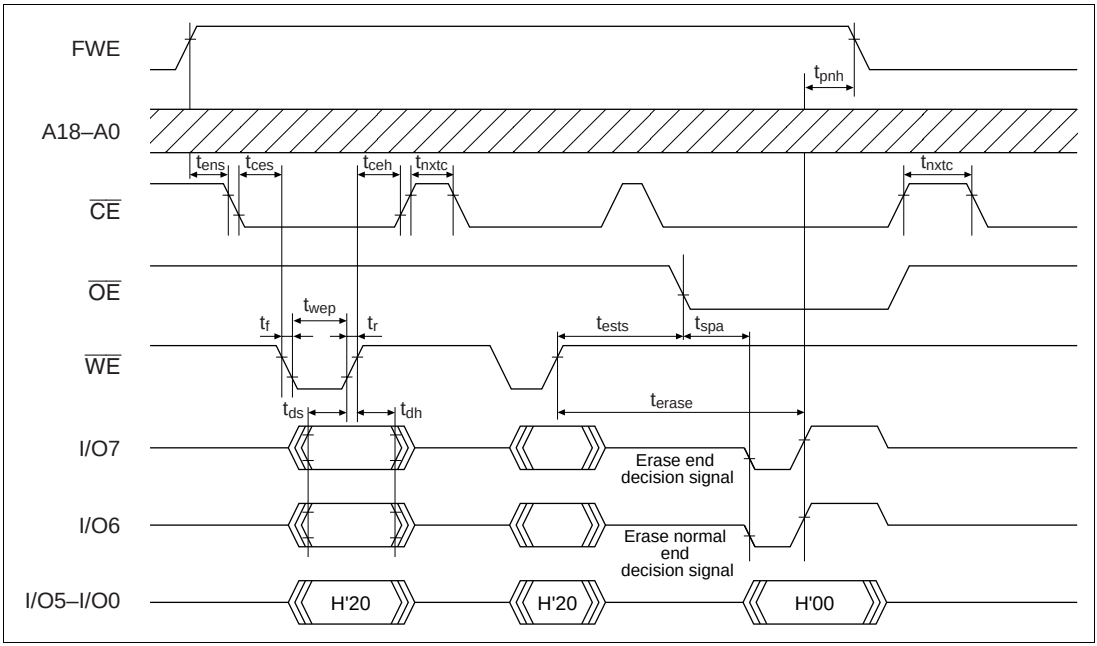


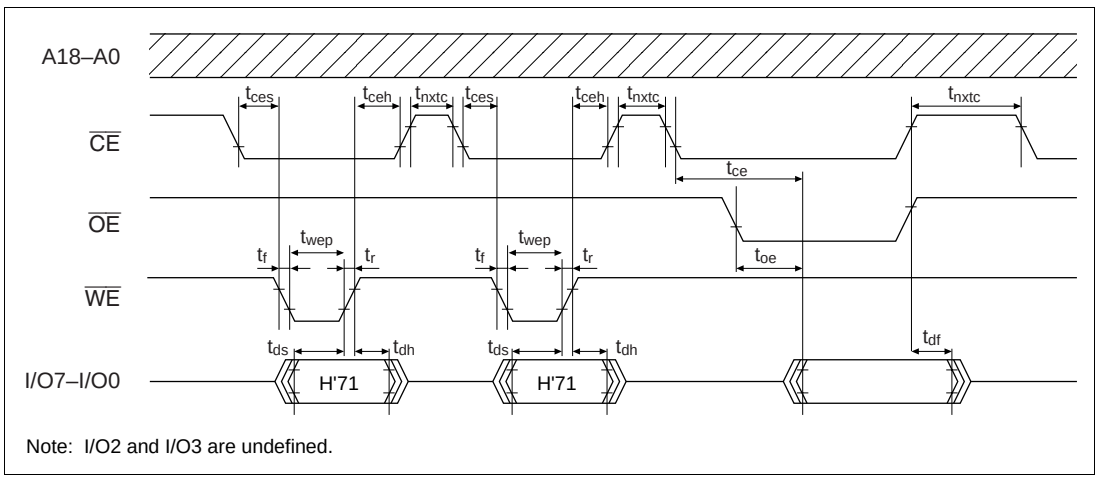
Figure 15-26 Auto-Erase Mode Timing Waveforms

15.13.6 Status Read Mode

1. Status read mode is provided to identify the kind of abnormal end. Use this mode when an abnormal end occurs in auto-program mode or auto-erase mode.
2. The return code is retained until a command write other than a status read mode command write is executed.

Table 15-20 AC Characteristics in Status Read Mode (Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Read time after command write	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Disable delay time	t_{df}		100	ns	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	



Note: I/O2 and I/O3 are undefined.

Figure 15-27 Status Read Mode Timing Waveforms

Table 15-21 Status Read Mode Return Commands

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end decision	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Program-ming error: 1 Otherwise: 0	Erasing error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address error: 1 Otherwise: 0

Note: I/O2 and I/O3 are undefined.

15.13.7 Status Polling

1. The I/O7 status polling flag indicates the operating status in auto-program/auto-erase mode.
2. The I/O6 status polling flag indicates a normal or abnormal end in auto-program/auto-erase mode.

Table 15-22 Status Polling Output Truth Table

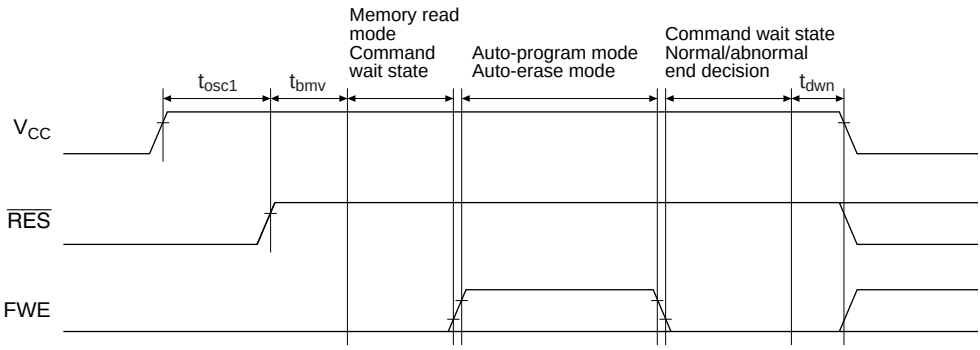
Pin Name	During Internal Operation	Abnormal End	—	Normal End
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O0–I/O5	0	0	0	0

15.13.8 Programmer Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the programmer mode setup period. After the programmer mode setup time, a transition is made to memory read mode.

Table 15-23 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit	Notes
Standby release (oscillation stabilization time)	t_{osc1}	30		ms	
Programmer mode setup time	t_{bmv}	10		ms	
V_{CC} hold time	t_{dwn}	0		ms	



Note: When using other than the automatic write mode and automatic erase mode, drive the FWE input pin low.

Figure 15-28 Oscillation Stabilization Time, Boot Program Transfer Time, and Power-Down Sequence

15.13.9 Notes on Memory Programming

1. When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
2. When performing programming using programmer mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Hitachi. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.

2. Auto-programming should be performed once only on the same address block. Additional programming cannot be performed on previously programmed address blocks.

15.14 Flash Memory and Power-Down States

In addition to its normal operating state, the flash memory has power-down states in which power consumption is reduced by halting part or all of the internal power supply circuitry.

There are three flash memory operating states:

(1) Normal operating mode: The flash memory can be read and written to.
(2) Standby mode: All flash memory circuits are halted, and the flash memory cannot be read or written to.

State (2) is flash memory power-down state. Table 15-24 shows the correspondence between the operating states of the LSI and the flash memory.

Table 15-24 Flash Memory Operating States

LSI Operating State	Flash Memory Operating State
High-speed mode	Normal mode (read/write)
Medium-speed mode	
Sleep mode	
Software standby mode	Standby mode
Hardware standby mode	

15.14.1 Note on Power-Down States

When the flash memory is in a power-down state, part or all of the internal power supply circuitry is halted. Therefore, a power supply circuit stabilization period must be provided when returning to normal operation. When the flash memory returns to its normal operating state from a power-down state, bits STS2 to STS0 in SBYCR must be set to provide a wait time of at least 100 µs (power supply stabilization time), even if an oscillation stabilization period is not necessary.

15.15 Flash Memory Programming and Erasing Precautions

Precautions concerning the use of on-board programming mode, the RAM emulation function, and PROM mode are summarized below.

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 Hitachi microcomputer device type with 256-kbyte on-chip flash memory (FZTAT256V3A).

Do not select the HN27C4096 setting for the PROM programmer, and only use the specified socket adapter. Failure to observe these points may result in damage to the device.

Powering on and off (See figures 15-29 to 15-31)

Do not apply a high level to the FWE pin until V_{CC} has stabilized. Also, drive the FWE pin low before turning off V_{CC}.

When applying or disconnecting V_{CC} power, fix the FWE 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.

FWE application/disconnection (See figures 15-29 to 15-31)

FWE application should be carried out when MCU operation is in a stable condition. If MCU operation is not stable, fix the FWE pin low and set the protection state.

The following points must be observed concerning FWE application and disconnection to prevent unintentional programming or erasing of flash memory:

- Apply FWE when the V_{CC} voltage has stabilized within its rated voltage range.
- In boot mode, apply and disconnect FWE during a reset.
- In user program mode, FWE can be switched between high and low level regardless of the reset state. FWE input can also be switched during execution of a program in flash memory.
- Do not apply FWE if program runaway has occurred.
- Disconnect FWE only when the SWE1, ESU1, PSU1, EV1, PV1, P1, and E bits in FLMCR1 are cleared.

  Make sure that the SWE1, ESU1, PSU1, EV1, PV1, P1, and E bits are not set by mistake when applying or disconnecting FWE.

Do not apply a constant high level to the FWE pin

Apply a high level to the FWE pin only when programming or erasing flash memory. A system configuration in which a high level is constantly applied to the FWE pin should be avoided. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

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 P1 or E1 bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

Do not set or clear the SWE1 bit during execution of a program in flash memory

Wait for at least 100 μs after clearing the SWE1 bit before executing a program or reading data in flash memory. When the SWE1 bit is set, data in flash memory can be rewritten, but access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE1 bit during programming, erasing, or verifying.

Similarly, when using emulation by RAM with a high level applied to the FWE pin, the SWE1 bit should be cleared before executing a program or reading data in flash memory. However, read/write accesses can be performed in the RAM area overlapping the flash memory space regardless of whether the SWE1 bit is set or cleared.

Do not use interrupts while flash memory is being programmed or erased

All interrupt requests, including NMI, should be disabled during FWE1 application to give priority to program/erase operations.

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.

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.

Do not touch the socket adapter or chip during programming

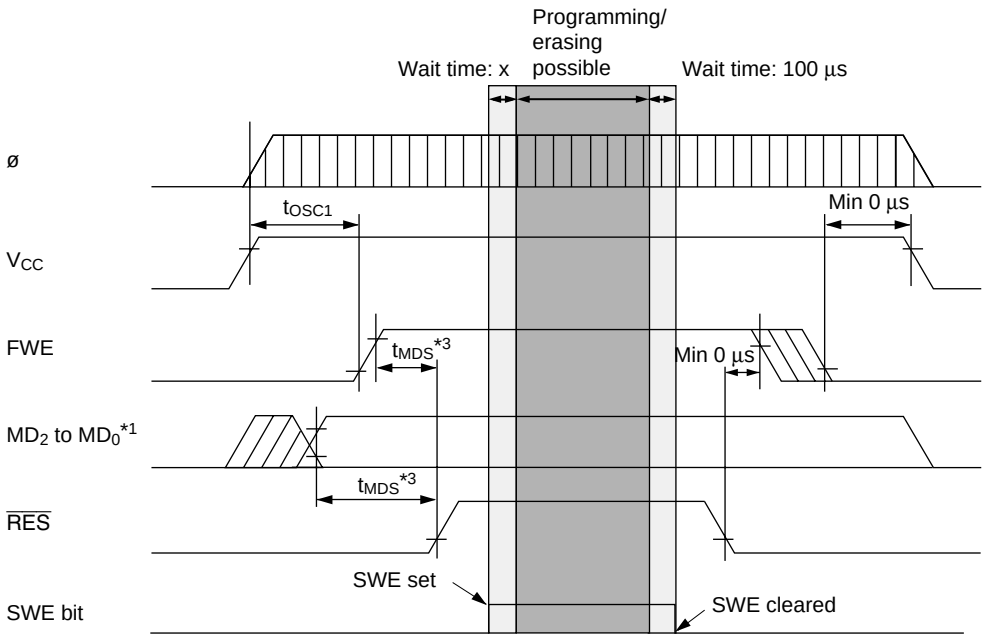
Touching either of these can cause contact faults and write errors.

The reset state must be entered after powering on

Apply the reset signal for at least 100 µs during the oscillation settling period.

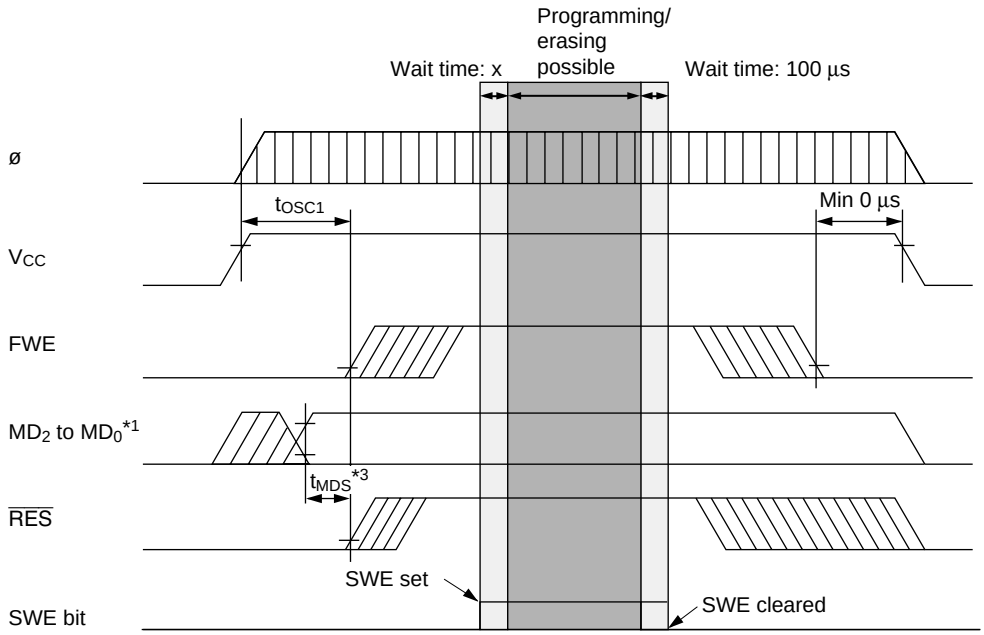
When a reset is applied during operation, this should be done while the SWE1 pin is low

Wait at least 100 µs after clearing the SWE1 bit before applying the reset.



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 18.6, Flash Memory Characteristics.
*3 Mode programming setup time t_{MDS} (min) = 200 ns

Figure 15-29 Power-On/Off Timing (Boot Mode)

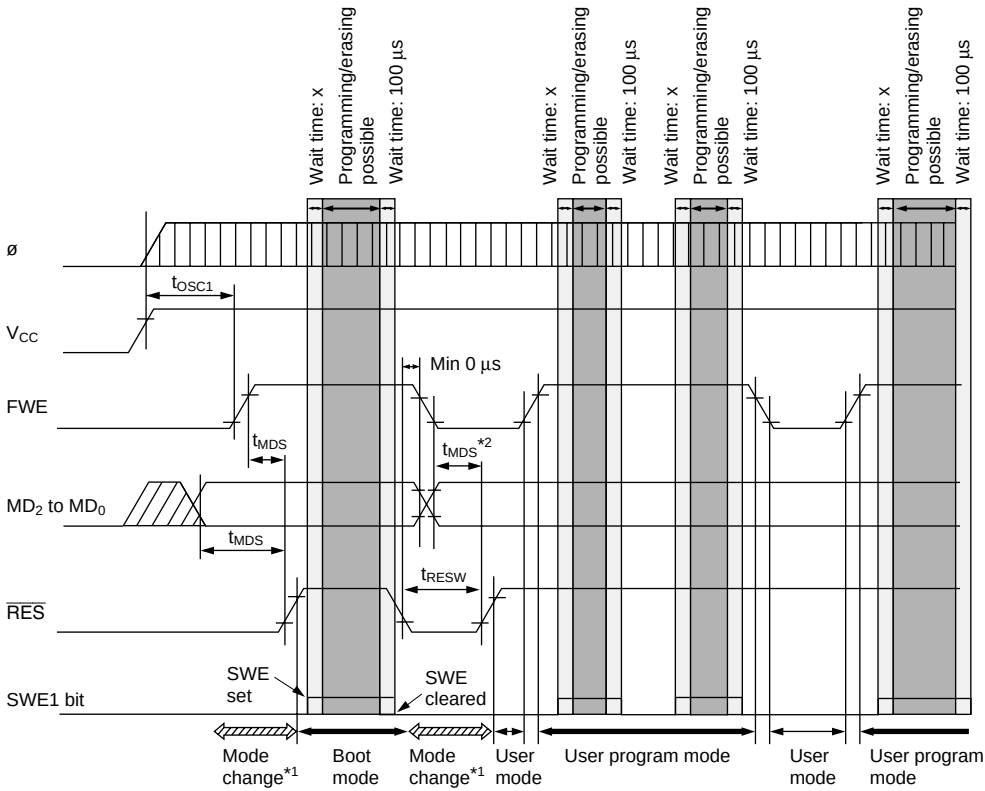


Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*2

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 18.6, Flash Memory Characteristics.
*3 Mode programming setup time t_{MDS} (min) = 200 ns

Figure 15-30 Power-On/Off Timing (User Program Mode)



Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*3

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 When entering boot mode or making a transition from boot mode to another mode, mode switching must be carried out by means of $\overline{RES}$ input. The state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{WR}$) will change during this switchover interval (the interval during which the $\overline{RES}$ pin input is low), and therefore these pins should not be used as output signals during this time.

*2 When making a transition from boot mode to another mode, a mode programming setup time t_{MDS} (min) of 200 ns is necessary with respect to $\overline{RES}$ clearance timing.

*3 See section 18.6, Flash Memory Characteristics.

Figure 15-31 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

15.16 Note on Switching from F-ZTAT Version to Mask ROM Version

The mask ROM version does not have the internal registers for flash memory control that are provided in the F-ZTAT version. Table 15-25 lists the registers that are present in the F-ZTAT version but not in the mask ROM version. If a register listed in table 15-25 is read in the mask ROM version, an undefined value will be returned. Therefore, if application software developed on the F-ZTAT version is switched to a mask ROM version product, it must be modified to ensure that the registers in table 15-25 have no effect.

Table 15-25 Registers Present in F-ZTAT Version but Absent in Mask ROM Version

Register	Abbreviation	Address
Flash memory control register 1	FLMCR1	H'FFA8
Flash memory control register 2	FLMCR2	H'FFA9
Erase block register 1	EBR1	H'FFAA
Erase block register 2	EBR2	H'FFAB
RAM emulation register	RAMER	H'FEDB
Serial control register X	SCRX	H'FDB4

Section 16 Clock Pulse Generator

16.1 Overview

The H8S/2214 has a built-in clock pulse generator (CPG) that generates the system clock (ø), the bus master clock, and internal clocks.

The clock pulse generator consists of a system clock oscillator, duty adjustment circuit, medium-speed clock divider, and bus master clock selection circuit.

16.1.1 Block Diagram

Figure 16-1 shows a block diagram of the clock pulse generator.

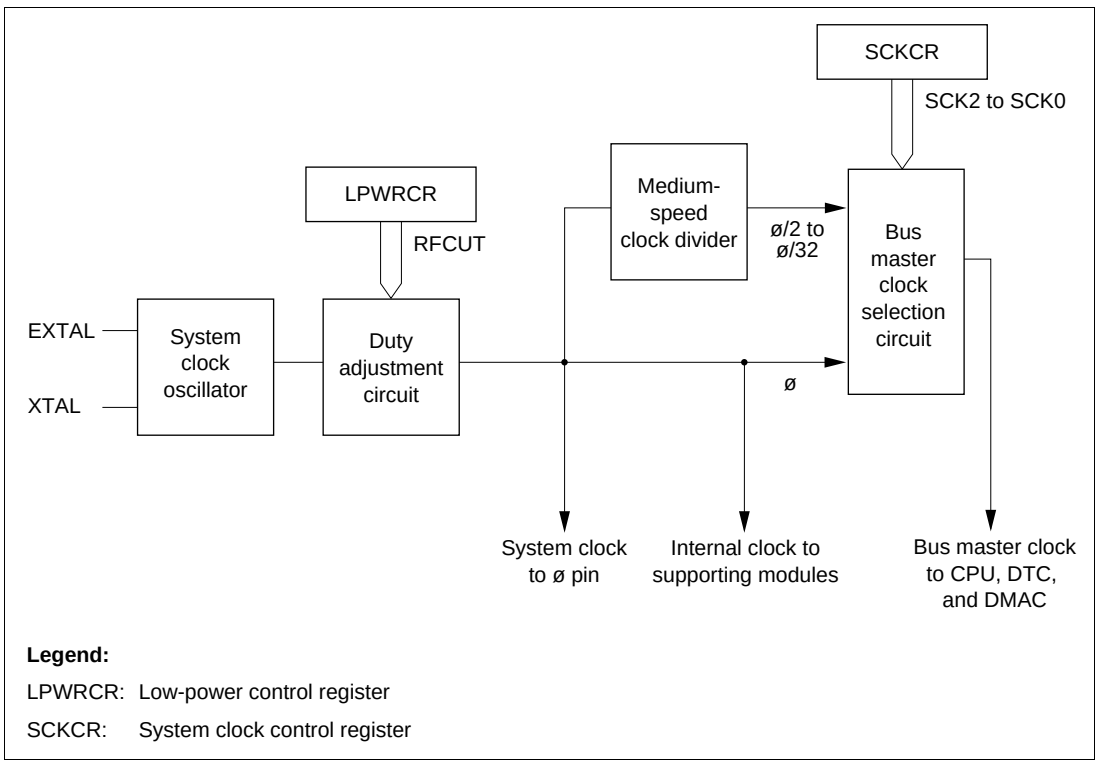


Figure 16-1 Block Diagram of Clock Pulse Generator

16.1.2 Register Configuration

The clock pulse generator is controlled by SCKCR and LPWRCR. Table 16-1 shows the register configuration.

Table 16-1 Clock Pulse Generator Register

Name	Abbreviation	R/W	Initial Value	Address*
System clock control register	SCKCR	R/W	H'00	H'FDE6
Low-power control register	LPWRCR	R/W	H'00	H'FDEC

Note:* Lower 16 bits of the address.

16.2 Register Descriptions

16.2.1 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	—	—	—	SCK2	SCK1	SCK0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	—	—	R/W	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ø clock output control and medium-speed mode control.

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

Bit 7—ø Clock Output Disable (PSTOP): Controls ø output.

	Description			
Bit 7 PSTOP	**High–Speed Mode Medium-Speed Mode**	**Sleep Mode**	**Software Standby Mode, Watch**	**Hardware Standby Mode**
0	ø output (initial value)	ø output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bits 6 and 3—Reserved: This bit can be read or written to, but only 0 should be written.

Bits 5 and 4—Reserved: Read-only bits, always read as 0.

Bits 2 to 0—System Clock Select 2 to 0 (SCK2 to SCK0): These bits select the bus master clock used in high-speed mode and medium-speed mode.

Bit 2	Bit 1	Bit 0		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master is in high-speed mode	(Initial value)
		1	Medium-speed clock is ø/2	
	1	0	Medium-speed clock is ø/4	
		1	Medium-speed clock is ø/8	
1	0	0	Medium-speed clock is ø/16	
		1	Medium-speed clock is ø/32	
	1	—	—	

16.2.2 Low-Power Control Register (LPWRCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	RFCUT	—	STC1	STC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LPWRCR is an 8-bit readable/writable register that performs power-down mode control. LPWRCR is initialized to H'00 by a power-on reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

Bits 7 to 4—Reserved: These bits can be read or written to, but only 0 should be written.

Bit 3—Built-in Feedback Resistor Control (RFCUT): Selects whether the oscillator's built-in feedback resistor and duty adjustment circuit are used with external clock input. Do not access this bit when a crystal oscillator is used.

After this bit is set when using external clock input, a transition should intially be made to software standby mode, watch mode, or subactive mode. Switching between use and non-use of the oscillator's built-in feedback resistor and duty adjustment circuit is performed when the transition is made to software standby mode, watch mode, or subactive mode.

Bit 3	
RFCUT	**Description**
0	System clock oscillator's built-in feedback resistor and duty adjustment circuit are used (Initial value)
1	System clock oscillator's built-in feedback resistor and duty adjustment circuit are not used

Bit 2—Reserved: This bit can be read or written to, but should only be written with 0.

Bits 1 and 0—Frequency Multiplication Factor (STC1, STC0): The STC bits specify the frequency multiplication factor of the PLL circuit incorporated into the evaluation chip. The specified frequency multiplication factor is valid after a transition to software standby mode, watch mode, or subactive mode.

With the H8S/2214, STC1 and STC0 must both be set to 1. After a reset, STC1 and STC0 are both cleared to 0, and so must be set to 1.

Bit 1	Bit 0	
STC1	**STC0**	**Description**
0	0	x1 (Initial value)
	1	x2 (Setting prohibited)
1	0	x4 (Setting prohibited)
	1	PLL is bypassed

16.3 System Clock Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

16.3.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 16-2. Select the damping resistance R_d according to table 16-2. An AT-cut parallel-resonance crystal should be used.

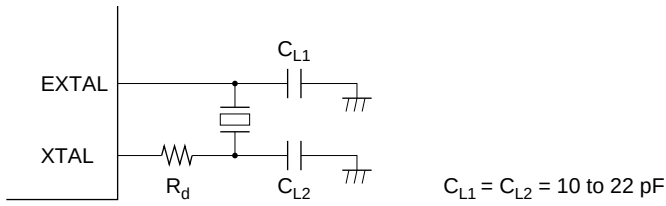


Figure 16-2 Connection of Crystal Resonator (Example)

Table 16-2 Damping Resistance Value

Frequency (MHz)	2	4	6	8	10	12	16
R_d (Ω)	1 k	500	300	200	100	0	0

Crystal Resonator: Figure 16-3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 16-3 and the same resonance frequency as the system clock (ø).

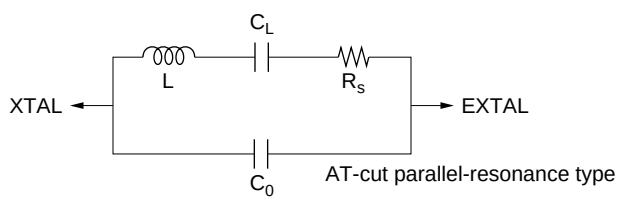


Figure 16-3 Crystal Resonator Equivalent Circuit

Table 16-3 Crystal Resonator Parameters

Frequency (MHz)	2	4	6	8	10	12	16
R_S max (Ω)	500	120	100	80	60	60	50
C_0 max (pF)	7	7	7	7	7	7	7

Note on Board Design: When a crystal resonator is connected, the following points should be noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 16-4.

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins.

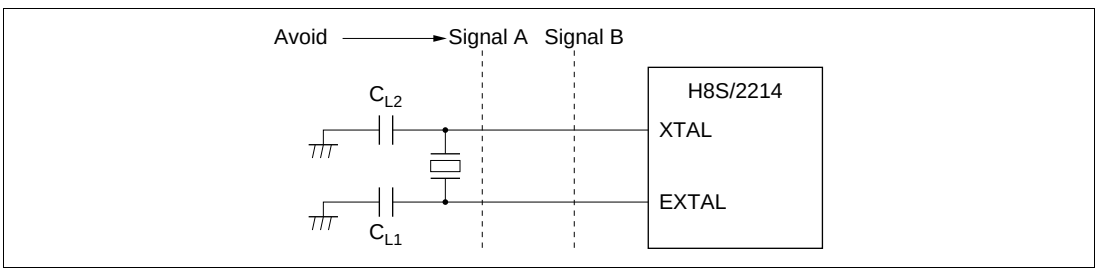


Figure 16-4 Example of Incorrect Board Design

16.3.2 External Clock Input

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 16-5. If the XTAL pin is left open, make sure that stray capacitance is no more than 10 pF.

In example (b), make sure that the external clock is held high in standby mode, subactive mode, subsleep mode, and watch mode.

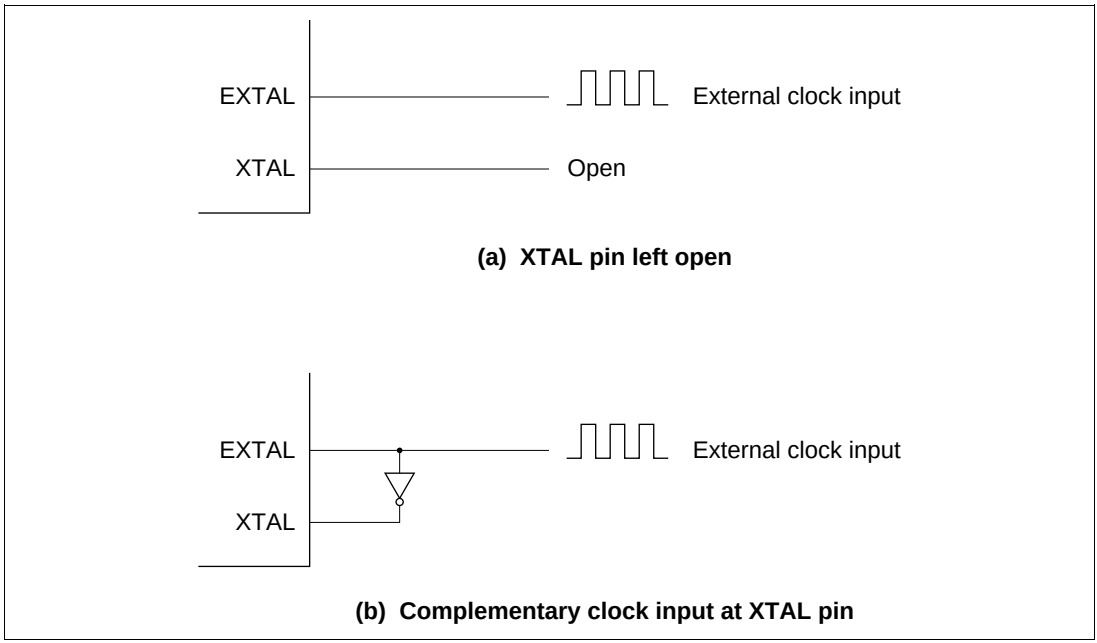


Figure 16-5 External Clock Input (Examples)

External Clock: The external clock signal should have the same frequency as the system clock (ø).

Table 16-4 and figure 16-6 show the input conditions for the external clock.

Table 16-4 External Clock Input Conditions

Item	Symbol	Min	Max	Unit	Test Conditions
External clock input lowpulse width	t_{EXL}	25	—	ns	Figure 16-6
External clock input high pulse width	t_{EXH}	25	—	ns	
External clock rise time	t_{EXr}	—	6.25	ns	
External clock fall time	t_{EXf}	—	6.25	ns	
Clock low pulse width level	t_{CL}	0.4	0.6	t_{cyc}	ø ≥ 5 MHz Figure 18-3
		80	—	ns	ø < 5 MHz
Clock high pulse width level	t_{CH}	0.4	0.6	t_{cyc}	ø ≥ 5 MHz
		80	—	ns	ø < 5 MHz

The external clock input conditions when the duty adjustment circuit is not used are shown in table 16-5 and figure 16-6. When the duty adjustment circuit is not used, the ø output waveform depends on the external clock input waveform, and so no restrictions apply.

Table 16-5 External Clock Input Conditions when the Duty Adjustment Circuit is not Used

Item	Symbol	Min	Max	Unit	Test Conditions
External clock input low pulse width	t_{EXL}	31.25	—	ns	Figure 16-6
External clock input high pulse width	t_{EXH}	31.25	—	ns	
External clock rise time	t_{EXr}	—	6.25	ns	
External clock fall time	t_{EXf}	—	6.25	ns	

Note: When duty adjustment circuit is not used, the maximum frequency decreases according to the input waveform. (Example: When $t_{EXL} = t_{EXH} = 50$ ns, and $t_{EXr} = t_{EXf} = 10$ ns, clock cycle time = 120 ns; therefore, maximum operating frequency = 8.3 MHz)

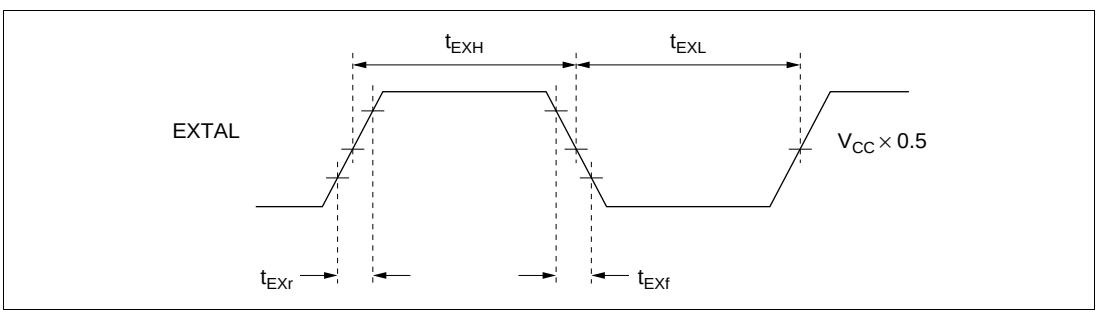


Figure 16-6 External Clock Input Timing

(3) Note on Switchover of External Clock

When two or more external clocks (e.g. 10 MHz and 2 MHz) are used as the system clock, switchover of the input clock should be carried out in software standby mode.

An example of an external clock switching circuit is shown in figure 16-7, and an example of the external clock switchover timing in figure 16-8.

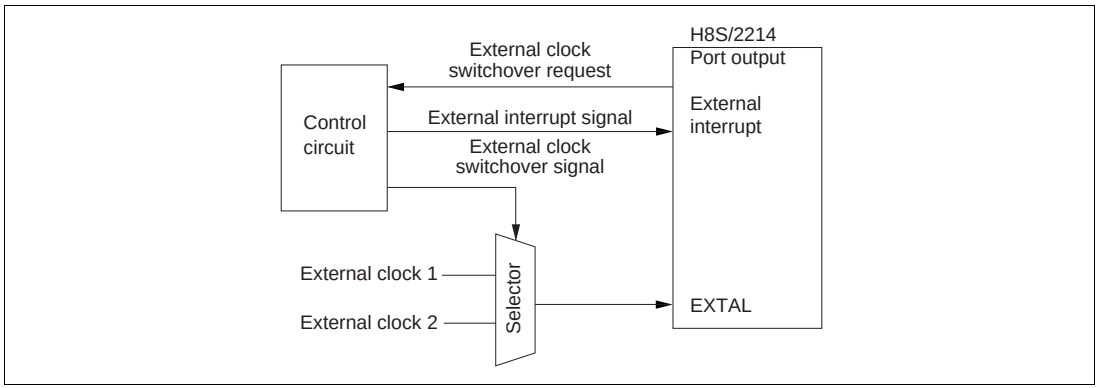


Figure 16-7 Example of External Clock Switching Circuit

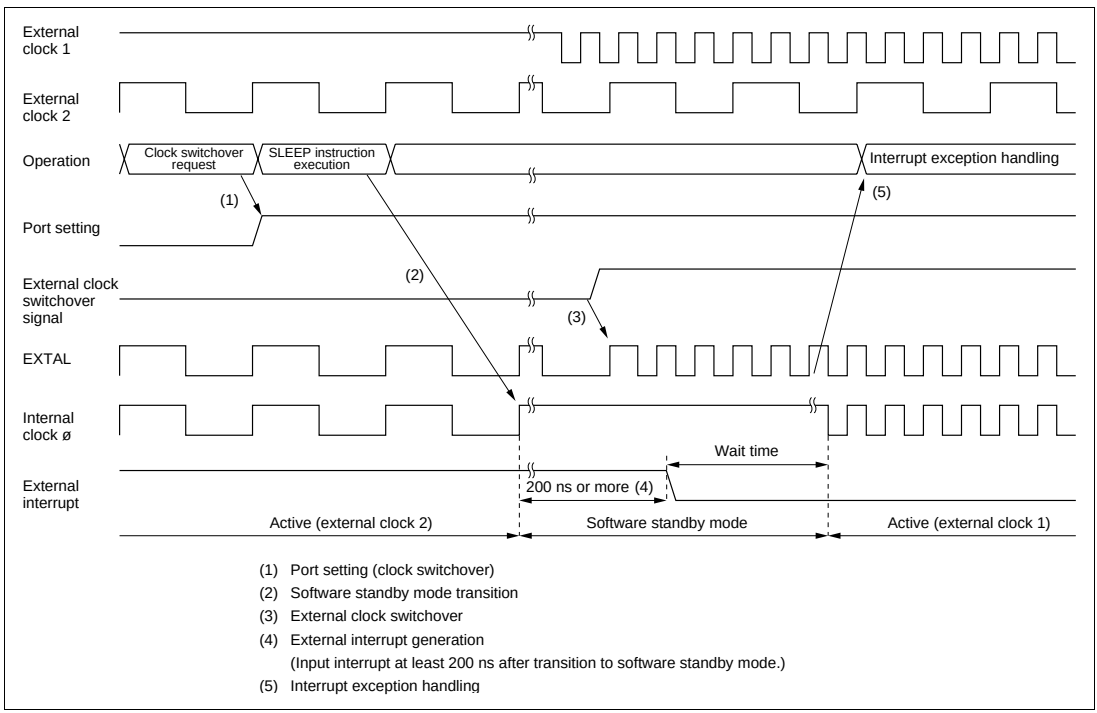


Figure 16-8 Example of External Clock Switchover Timing

16.4 Duty Adjustment Circuit

When the oscillator frequency is 5 MHz or higher, the duty adjustment circuit adjusts the duty cycle of the clock signal from the oscillator to generate the system clock (ø).

16.5 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate ø/2, ø/4, ø/8, ø/16, and ø/32.

16.6 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the system clock (ø) or one of the medium-speed clocks (ø/2, ø/4, or ø/8, ø/16, and ø/32) to be supplied to the bus master, according to the settings of the SCK2 to SCK0 bits in SCKCR.

16.7 Note on Crystal Resonator

Since various characteristics related to the crystal resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, for both the mask versions, and F-ZTAT versions, using the resonator connection examples shown in this section as a guide. As the resonator circuit ratings will depend on the floating capacitance of the resonator and the mounting circuit, the ratings should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

Section 17 Power-Down Modes

17.1 Overview

In addition to the normal program execution state, the H8S/2214 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 supporting modules, and so on.

The H8S/2214 operating modes are as follows:

(1) High-speed mode
(2) Medium-speed mode
(3) Sleep mode
(4) Module stop mode
(5) Software standby mode
(6) Hardware standby mode

Of these, (2) to (6) are power-down modes. Sleep mode is CPU mode, medium-speed mode is a CPU and bus master mode, and module stop mode is an on-chip supporting module mode (including bus masters other than the CPU). Certain combinations of these modes can be set.

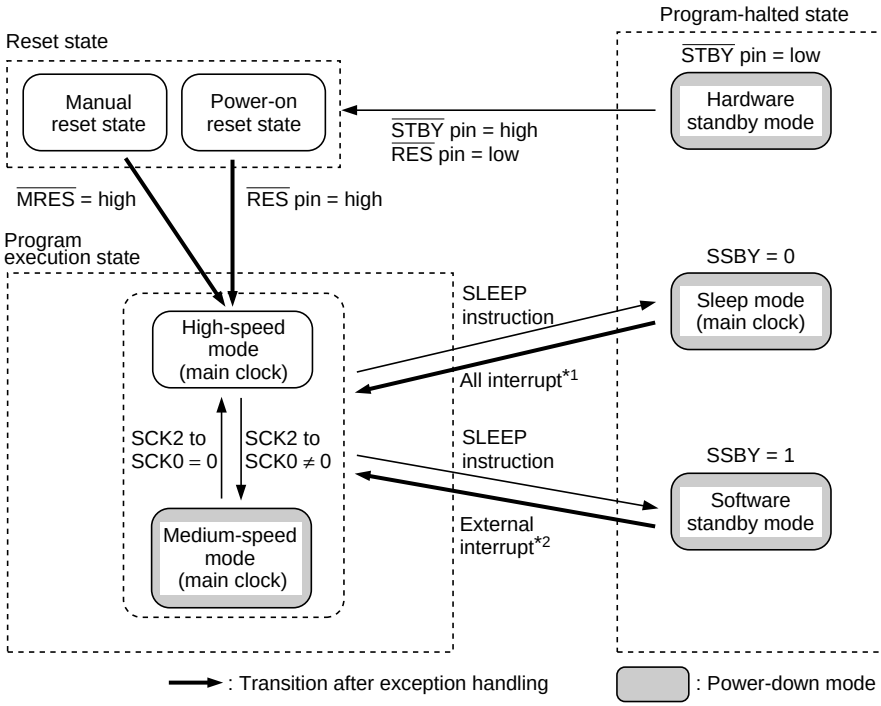
After a reset, the MCU is in high-speed mode.

Table 17-1 shows the internal chip states in each mode, and table 17-2 shows the conditions for transition to the various modes. Figure 17-1 shows a mode transition diagram.

Table 17-1 H8S/2214 Internal States in Each Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby
System clock oscillator		Functioning	Functioning	Functioning	Functioning	Halted	Halted
Subclock oscillator		Functioning	Functioning	Functioning	Functioning	Functioning/ Halted	Halted
CPU operation	Instructions	Functioning	Medium-speed	Halted	Functioning	Halted	Halted
	Registers			Retained		Retained	Undefined
RAM		Functioning	Functioning	Functioning (DTC)	Functioning	Retained	Retained
I/O		Functioning	Functioning	Functioning	Functioning	Retained	High impedance
External interrupts		Functioning	Functioning	Functioning	Functioning	Functioning	Halted
On-chip supporting module operation	DMAC	Functioning	Medium-speed	Functioning	Functioning/ halted (retained)	Halted (retained)	Halted (reset)
	DTC						
	WDT0		Functioning		Functioning		
	TPU				Functioning/ halted (retained)		
	SCI						
	D/A						

Note: “Halted (retained)” means that internal register values are retained. The internal state is operation suspended.
“Halted (reset)” means that internal register values and internal states are initialized.
In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).
: Operating state



Notes: *1 All interrupts
*2 NMI, IRQ0 to IRQ7

- When a transition is made between modes by means of an interrupt, transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
- From any state except hardware standby mode, a transition to the power-on reset state occurs whenever $\overline{RES}$ goes low. From any state except hardware standby mode and the power-on reset state, a transition to the manual reset state occurs whenever $\overline{MRES}$ goes low.
- From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.

Figure 17-1 Mode Transitions

17.1.1 Register Configuration

The power-down modes are controlled by the SBYCR, SCKCR, LPWRCR, TCSR (WDT1), and MSTPCR registers. Table 17-2 summarizes these registers.

Table 17-2 Power-Down Mode Registers

Name	Abbreviation	R/W	Initial Value	Address*
Standby control register	SBYCR	R/W	H'08	H'FDE4
System clock control register	SCKCR	R/W	H'00	H'FDE6
Module stop control register	MSTPCRA	R/W	H'3F	H'FDE8
	MSTPCRB	R/W	H'FF	H'FDE9
	MSTPCRC	R/W	H'FF	H'FDEA

Note: * Lower 16 bits of the address.

17.2 Register Descriptions

17.2.1 Standby Control Register (SBYCR)

Bit	:	7	6	5	4	3	2	1	0
		SSBY	STS2	STS1	STS0	OPE	—	—	—
Initial value	:	0	0	0	0	1	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	—	—	—

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

SBYCR is initialized to H'08 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Software Standby (SSBY): Specifies a transition to software standby mode. The SSBY setting is not changed by a mode transition due to an interrupt, etc.

Bit 7

SSBY	Description
0	Transition to sleep mode after execution of SLEEP instruction (Initial value)
	Transition to subsleep mode after execution of SLEEP instruction in subactive mode
1	Transition to software standby mode after execution of SLEEP instruction

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the time the MCU waits for the clock to stabilize when software standby mode is cleared and a transition is made to high-speed mode or medium-speed mode by means of a specific interrupt or instruction. With crystal oscillation, refer to table 17-4 and make a selection according to the operating frequency so that the standby time is at least 8 ms (the oscillation stabilization time). With an external clock, any selection can be made*.

Note: * In the F-ZTAT version, a 16-state standby time cannot be used with an external clock. Use 2048 states or more.

Bit 6	Bit 5	Bit 4		
STS2	**STS1**	**STS0**	**Description**	
0	0	0	Standby time = 8192 states	(Initial value)
		1	Standby time = 16384 states	
	1	0	Standby time = 32768 states	
		1	Standby time = 65536 states	
1	0	0	Standby time = 131072 states	
		1	Standby time = 262144 states	
	1	0	Standby time = 2048 states	
		1	Standby time = 16 states*	

Note: * Cannot be used in the F-ZTAT version.

Bit 2 to 0—Reserved: This bit cannot be modified and is always read as 0.

Bit 3—Output Port Enable (OPE): Specifies whether the address bus and bus control signals ($\overline{CS0}$ to $\overline{CS7}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$) retain their output state or go to the high-impedance state in software standby mode.

Bit 3	
OPE	**Description**
0	In software standby mode, address bus and bus control signals are high-impedance
1	In software standby mode, address bus and bus control signals retain their output state (Initial value)

17.2.2 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	—	—	—	SCK2	SCK1	SCK0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	—	—	R/W	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ø clock output control and medium-speed mode control.

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

Bit 7—ø Clock Output Disable (PSTOP): Controls ø output.

	Description			
Bit 7 PSTOP	**High–Speed Mode Medium-Speed Mode**	**Sleep Mode**	**Software Standby Mode, Watch**	**Hardware Standby Mode**
0	ø output (initial value)	ø output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bits 6 and 3—Reserved: These bits can be read or written to, but should only be written with 0.

Bits 5 and 4—Reserved: These bits cannot be modified and are always read as 0.

Bits 2 to 0—System Clock Select 2 to 0 (SCK2 to SCK0): These bits select the clock for the bus master in high-speed mode and medium-speed mode.

Bit 2	Bit 1	Bit 0		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master is in high-speed mode	(Initial value)
		1	Medium-speed clock is ø/2	
	1	0	Medium-speed clock is ø/4	
		1	Medium-speed clock is ø/8	
1	0	0	Medium-speed clock is ø/16	
		1	Medium-speed clock is ø/32	
	1	—	—	

17.2.3 Module Stop Control Register (MSTPCR)

MSTPCRA

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRB

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC

Bit	:	7	6	5	4	3	2	1	0
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRA, MSTPCRB, and MSTPCRC are 8-bit readable/writable registers that perform module stop mode control.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. MSTPCRB and MSTPCRC are initialized to H'FF. They are not initialized in software standby mode.

MSTPCRA, MSTPCRB, and MSTPCRC Bits 7 to 0—Module Stop (MSTPA7 to MSTPA0, MSTPB7 to MSTPB0, and MSTPC7 to MSTPC0): These bits specify module stop mode. See table 17-3 for the method of selecting on-chip supporting modules.

MSTPCRA, MSTPCRB, and MSTPCRC Bits 7 to 0

MSTPA7 to MSTPA0, MSTPB7 to MSTPB0, and MSTPC7 to MSTPC0	Description
0	Module stop mode is cleared (Initial value of MSTPA7, MSTPA6)
1	Module stop mode is set (Initial value of except MSTPA7 to MSTPA6)

17.3 Medium-Speed Mode

When the SCK2 to SCK0 bits in SCKCR are set to 1 in high-speed mode, the operating mode changes to medium-speed mode at the end of the bus cycle. In medium-speed mode, the CPU operates on the operating clock (ø/2, ø/4, ø/8, ø/16, or ø/32) specified by the SCK2 to SCK0 bits. The bus master other than the CPU (the DMAC and DTC) also operates in medium-speed mode. On-chip supporting modules other than the bus masters always operate on the high-speed clock (ø).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if ø/4 is selected as the operating clock, on-chip memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

If a SLEEP instruction is executed when the SSBY is cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, a transition is made to software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin and $\overline{\text{MRES}}$ pin is driven low, a transition is made to the reset state, and medium-speed mode is cleared. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 17-2 shows the timing for transition to and clearance of medium-speed mode.

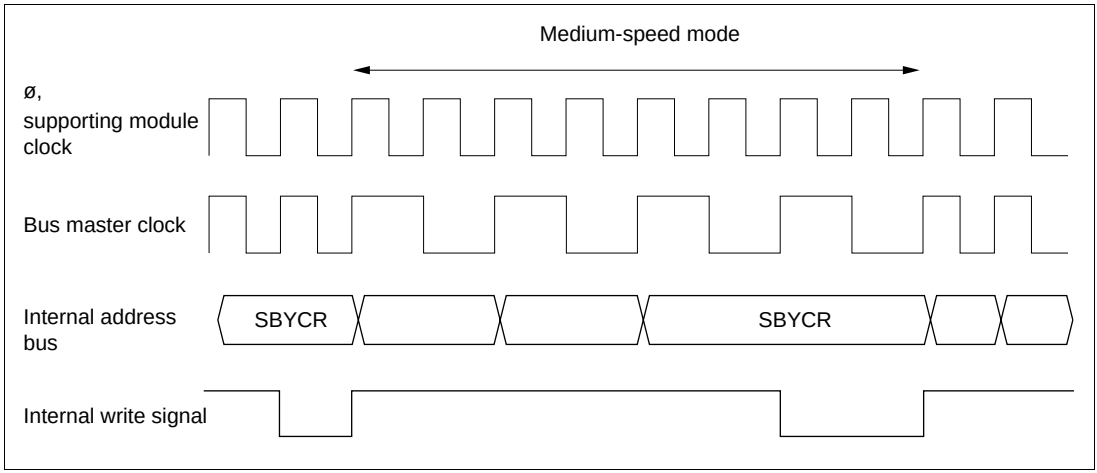


Figure 17-2 Medium-Speed Mode Transition and Clearance Timing

17.4 Sleep Mode

17.4.1 Sleep Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, the CPU enters sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other supporting modules do not stop.

17.4.2 Clearing Sleep Mode

Sleep mode is cleared by all interrupts, or with the $\overline{\text{RES}}$ pin, $\overline{\text{MRES}}$ pin or $\overline{\text{STBY}}$ pin.

Clearing with an Interrupt: When an interrupt request signal is input, sleep mode is cleared and interrupt exception handling is started. Sleep mode will not be cleared if interrupts are disabled, or if interrupts other than NMI have been masked by the CPU.

Clearing with the $\overline{\text{RES}}$ Pin and $\overline{\text{MRES}}$ Pin: When the $\overline{\text{RES}}$ pin and $\overline{\text{MRES}}$ pin is driven low, the reset state is entered. When the $\overline{\text{RES}}$ pin and $\overline{\text{MRES}}$ pin is driven high after the prescribed reset input period, 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.

17.5 Module Stop Mode

17.5.1 Module Stop Mode

Module stop mode can be set for individual on-chip supporting modules.

When the corresponding MSTP bit in MSTPCR 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.

Table 17-3 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating again at the end of the bus cycle. In module stop mode, the internal states of modules are retained.

After reset release, all modules other than the DMAC and DTC are in module stop mode.

When an on-chip supporting module is in module stop mode, read/write access to its registers is disabled.

When a transition is made to sleep mode with all modules stopped (MSTPCR = H'FFFFFF), the bus controller and I/O ports also stop operating, enabling current dissipation to be further reduced.

Table 17-3 MSTP Bits and Corresponding On-Chip Supporting Modules

Register	Bit	Module
MSTPCRA	MSTPA7	DMA controller (DMAC)
	MSTPA6	Data transfer controller (DTC)
	MSTPA5	16-bit timer pulse unit (TPU)
	MSTPA4	—*
	MSTPA3	—*
	MSTPA2	—*
	MSTPA1	—*
	MSTPA0	—*
MSTPCRB	MSTPB7	Serial communication interface 0 (SCI0)
	MSTPB6	Serial communication interface 1 (SCI1)
	MSTPB5	Serial communication interface 2 (SCI2)
	MSTPB4	—*
	MSTPB3	—*
	MSTPB2	—*
	MSTPB1	—*
	MSTPB0	External module expansion function
MSTPCRC	MSTPC7	—*
	MSTPC6	—*
	MSTPC5	D/A converter
	MSTPC4	—*
	MSTPC3	—*
	MSTPC2	—*
	MSTPC1	—*
	MSTPC0	—*

Note: * Reserved.

17.5.2 Usage Notes

DMAC and DTC Module Stop Mode: Depending on the operating status of the DMAC and DTC, the MSTPA7 and MSTPA6 bits may not be set to 1. Setting of the DTC module stop mode should be carried out only when the DTC is not activated.

For details, section 7, DMA Controller (DMAC) and section 8, Data Transfer Controller (DTC).

On-Chip Supporting 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, DMAC, or DTC activation source. Interrupts should therefore be disabled before setting module stop mode.

Writing to MSTPCR: MSTPCR should be written to only by the CPU.

17.6 Software Standby Mode

17.6.1 Software Standby Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, the LSON bit in software standby mode is entered. In this mode, the CPU, on-chip supporting modules, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip supporting module, and of the I/O ports, are retained. The address bus and bus control signals are placed in the high-impedance state.

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

17.6.2 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, $\overline{\text{MRES}}$ pin or $\overline{\text{STBY}}$ pin.

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 STS2 to STS0 in SYSCR, stable clocks are supplied to the entire H8S/2214 chip, software standby mode is cleared, and interrupt exception handling is started.

When software standby mode is cleared with an IRQ0 to IRQ7 interrupt, set the corresponding enable bit to 1 and ensure that an interrupt of higher priority than interrupts IRQ0 to IRQ7 is not generated. Software standby mode cannot be cleared if the interrupt has been masked by the CPU side or has been designated as a DTC activation source.

Clearing with the $\overline{\text{RES}}$ Pin and $\overline{\text{MRES}}$ Pin: When the $\overline{\text{RES}}$ pin and $\overline{\text{MRES}}$ pin are driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire H8S/2214 chip. Note that the $\overline{\text{RES}}$ pin and $\overline{\text{MRES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin and $\overline{\text{MRES}}$ pin go 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.

17.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode

Bits STS2 to STS0 in SBYCR should be set as described below.

Using a Crystal Oscillator: Set bits STS2 to STS0 so that the standby time is at least 8 ms (the oscillation stabilization time).

Table 17-4 shows the standby times for different operating frequencies and settings of bits STS2 to STS0.

Table 17-4 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	16 MHz	13 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 MHz	Unit
0	0	0	8192 states	0.51	0.6	0.8	1.0	1.3	2.0	4.1	ms
		1	16384 states	1.0	1.3	1.6	2.0	2.7	4.1	8.2	
	1	0	32768 states	2.0	2.5	3.3	4.1	5.5	8.2	16.4	
		1	65536 states	4.1	5.0	6.6	8.2	10.9	16.4	32.8	
1	0	0	131072 states	8.2	10.1	13.1	16.4	21.8	32.8	65.5	
		1	262144 states	16.4	20.2	26.2	32.8	43.6	65.6	131.2	
	1	0	2048 states	0.13	0.16	0.2	0.3	0.3	0.5	1.0	
		1	16 states	1.0	1.2	1.6	2.0	1.7	4.0	8.0	µs

: Recommended time setting

Using an External Clock: Any value can be set. Normally, use of the minimum time is recommended.

Note: In the F-ZTAT version, a 16-state standby time cannot be used with an external clock. Use 2048 states or more.

17.6.4 Software Standby Mode Application Example

Figure 17-3 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software 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 SYSCR 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.

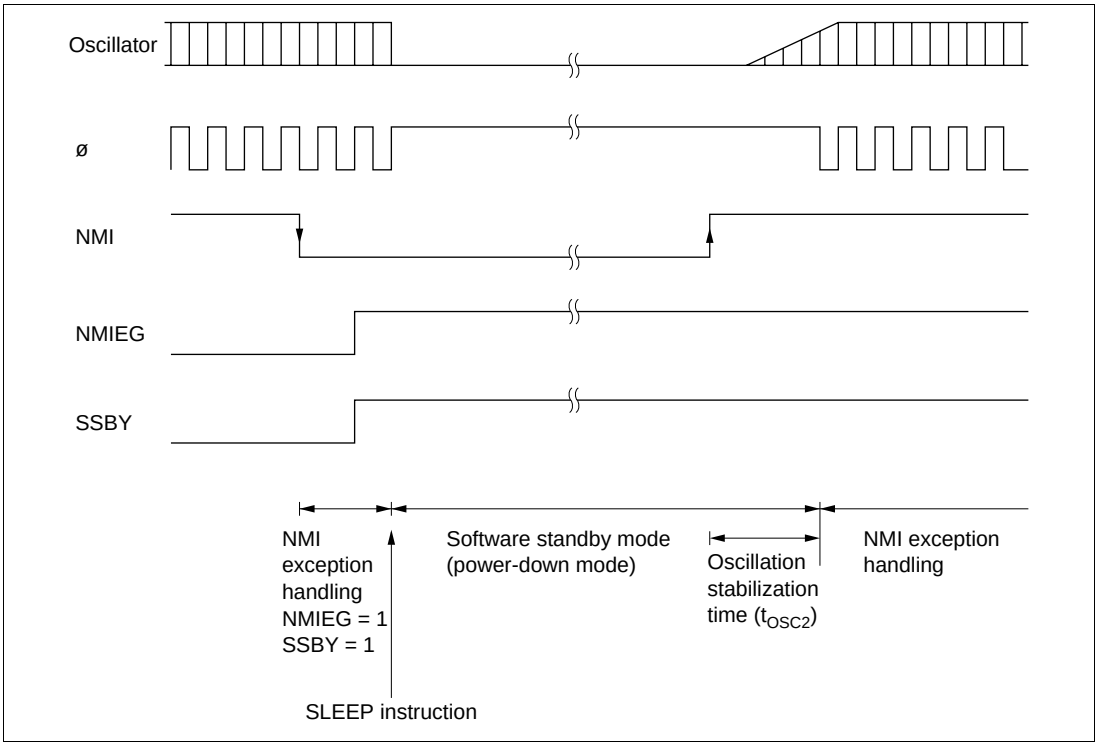


Figure 17-3 Software Standby Mode Application Example

17.6.5 Usage Notes

I/O Port States: In software standby mode, I/O port states are retained. If the OPE bit is set to 1, the address bus and bus control signal output is also retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

Current Dissipation During the Oscillation Stabilization Wait Period: Current dissipation increases during the oscillation stabilization wait period.

17.7 Hardware Standby Mode

17.7.1 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 the H8S/2214 is in 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 oscillation stabilizes (at least t_{osc1} —the oscillation stabilization time—when using a crystal oscillator). 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.

17.7.2 Hardware Standby Mode Timing

Figure 17-4 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.

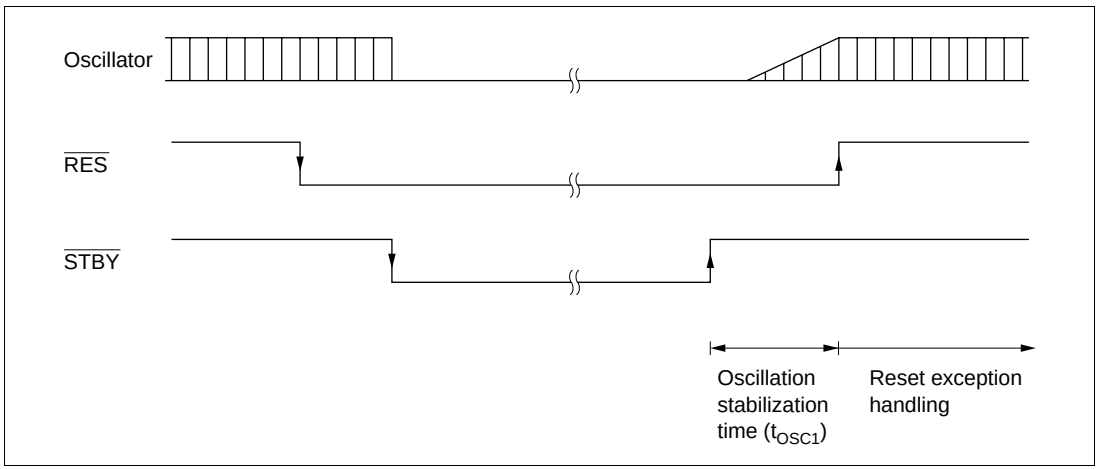


Figure 17-4 Hardware Standby Mode Timing (Example)

17.8 ø Clock Output Disabling Function

Output of the ø clock can be controlled by means of the PSTOP bit in SCKCR and the corresponding DDR bit. When the PSTOP bit is set to 1, the ø clock is stopped at the end of the bus cycle, and ø output goes high. ø clock output is enabled when PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0, ø clock output is disabled and input port mode is set. Table 17-5 shows the state of the ø pin in each processing mode.

Table 17-5 ø Pin State in Each Processing Mode

DDR	**0**	**1**	**1**
PSTOP	**—**	**0**	**1**
Hardware standby mode	High Impedance	High Impedance	High Impedance
Software standby mode	High Impedance	Fixed high	Fixed high
Sleep mode	High Impedance	ø output	Fixed high
High-speed mode, medium-speed mode	High Impedance	ø output	Fixed high

Section 18 Electrical Characteristics

18.1 Absolute Maximum Ratings

Table 18-1 lists the absolute maximum ratings.

Table 18-1 Absolute Maximum Ratings

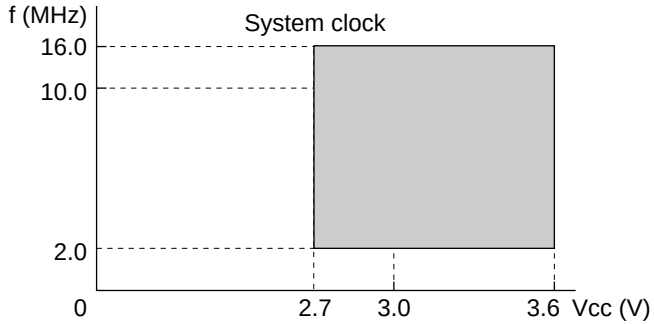
Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +4.6	V
Programming voltage	V_{PP}	–0.3 to +13.5	V
Input voltage (except port 9)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +4.6	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 chip may result if absolute maximum rating are exceeded.

18.2 Power Supply Voltage and Operating Frequency Range

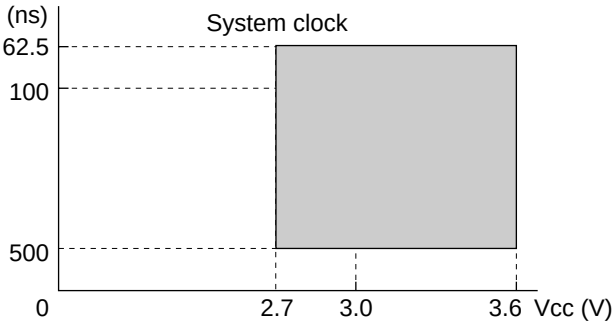
Power supply voltage and operating frequency ranges (shaded areas) are shown in figure 18.1.

(1) Power Supply Voltage and Oscillation Frequency Range



- Active (high-speed/medium-speed) mode
- Sleep mode

(2) Power Supply Voltage and Instruction Execution Time Range



- Active (high-speed/medium-speed) mode

Figure 18-1 Power Supply Voltage and Operating Ranges

18.3 DC Characteristics

Tables 18-2, 18-3, and 18-4 list the DC characteristics. Table 18-5 lists the permissible output currents.

Table 18-2 DC Characteristics (1)

Condition: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, $V_{ref} = AV_{CC}$,
$V_{SS} = AV_{SS}$ = 0 V, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)*

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	$\overline{IRQ7}$ to $\overline{IRQ0}$	V_T^-	$V_{CC} \times 0.2$	—	—	V	
	$\overline{EXIRQ0}$ to $\overline{EXIRQ7}$	V_T^+	—	—	$V_{CC} \times 0.8$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD2 to MD0, FWE	V_{IH}	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V	
	EXTAL, port 1, 3, 4, 7, A to G		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	V	
	Port 9		$V_{CC} \times 0.8$	—	$AV_{CC} + 0.3$	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, FWE, MD2 to MD0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, port 1, 3, 4, 7, 9, A to G		–0.3	—	$V_{CC} \times 0.2$	V	
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 0.4 mA
			—	—	0.4	V	I_{OL} = 0.8 mA
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{STBY}$, NMI, MD2 to MD0, port 4		—	—	1.0	µA	
	Port 9		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5 V

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Three-state leakage current (off state)	Port 1, 3, 7, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
MOS input pull-up current	Port A to E	$-I_P$	10	—	300	µA	V_{in} = 0 V

Note: * If the D/A converter is not used, do not leave the AV_{CC}, V_{ref} , and AV_{SS} pins open. Apply a voltage between 2.0 V and 3.6 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set $V_{ref} = AV_{CC}$.

Table 18-3 DC Characteristics (2)

Conditions: F-ZTAT version: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*[1]

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	NMI		—	—	30	pF	
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	
Current dissipation*[2]	Normal operation	I_{CC}*[4]	—	20 (V_{CC} = 3.0 V)	36.0 (V_{CC} = 3.6 V)	mA	f = 16 MHz
	Sleep mode		—	13 (V_{CC} = 3.0 V)	26.0 (V_{CC} = 3.6 V)	mA	f = 16 MHz
	All modules stopped		—	14	—	mA	f = 16 MHz, V_{CC} = 3.0 V (reference values)
	Medium-speed mode (ø/32)		—	9	—	mA	f = 16 MHz, V_{CC} = 3.0 V (reference values)
	Standby mode*[3]		—	1.0	10	µA	$T_a \leq 50°C$
			—	—	50		50°C < T_a
Analog power supply current	During D/A conversion	AI_{CC}	—	0.01	5	mA	AV_{CC} = 3.0 V
	Idle		—	0.01	5	µA	
Reference current	During D/A conversion	AI_{CC}	—	1.0	1.8	mA	V_{ref} = 3.0 V
	Idle		—	0.01	5	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: *1 If the D/A converter is not used, do not leave the AV_{CC}, V_{ref} , and AV_{SS} pins open. Apply a voltage between 2.0 V and 3.6 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set V_{ref} = AV_{CC}.

*2 Current dissipation values are for V_{IH} min = V_{CC} – 0.3 V, V_{IL} max = 0.3 V with all output pins unloaded and the on-chip pull-up resistors in the off state.

*3 The values are for $V_{RAM} \leq V_{CC} < 2.7$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.

*4 I_{CC} depends on V_{CC} and f as follows:

I_{CC} max = 1.0 (mA) + 0.61 (mA/(MHz × V)) × V_{CC} × f (normal operation)

I_{CC} max = 1.0 (mA) + 0.44 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

Table 18-4 DC Characteristics (3)

Conditions: Mask ROM version: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = AV_{CC}, V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*[1]

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	NMI		—	—	30	pF	
	All input pins except $\overline{RES}$ and NMI		—	—	15	pF	
Current dissipation*[2]	Normal operation	I_{CC}*[4]	—	20 V_{CC} = 3.0 V	36 V_{CC} = 3.6 V	mA	f = 16 MHz
	Sleep mode		—	13 V_{CC} = 3.0 V	26 V_{CC} = 3.6 V	mA	f = 16 MHz
	All modules stopped		—	14	—	mA	f = 16 MHz, V_{CC} = 3.0 V (reference values)
	Medium-speed mode (ø/32)		—	9	—	mA	f = 16 MHz, V_{CC} = 3.0 V (reference values)
	Standby mode*[3]		—	1.0	10	µA	$T_a \leq$ 50°C
			—	—	50		50°C < T_a
Analog power supply current	During D/A conversion	AI_{CC}	—	0.01	5	mA	AV_{CC} = 3.0 V
	Idle		—	0.01	5	µA	
Reference current	During D/A conversion	AI_{CC}	—	1.0	1.8	mA	V_{ref} = 3.0 V
	Idle		—	0.01	5	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: *1 If the D/A converter is not used, do not leave the AV_{CC}, V_{ref} , and AV_{SS} pins open. Apply a voltage between 2.0 V and 3.6 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set V_{ref} = AV_{CC}.

*2 Current dissipation values are for V_{IH} min = V_{CC} – 0.3 V, V_{IL} max = 0.3 V with all output pins unloaded and the on-chip pull-up resistors in the off state.

*3 The values are for $V_{RAM} \leq V_{CC}$ < 2.7 V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.

*4 I_{CC} depends on V_{CC} and f as follows:

I_{CC} max = 1.0 (mA) + 0.61 (mA/(MHz × V)) × V_{CC} × f (normal operation)

I_{CC} max = 1.0 (mA) + 0.44 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

Table 18-5 Permissible Output Currents

Conditions: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = AV_{CC},
V_{SS} = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)*

Item			Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	All output pins	V_{CC} = 2.7 to 3.6 V	I_{OL}	—	—	1.0	mA
Permissible output low current (total)	Total of all output pins	V_{CC} = 2.7 to 3.6 V	$\sum I_{OL}$	—	—	60	mA
Permissible output high current (per pin)	All output pins	V_{CC} = 2.7 to 3.6 V	$-I_{OH}$	—	—	1.0	mA
Permissible output high current (total)	Total of all output pins	V_{CC} = 2.7 to 3.6 V	$\sum -I_{OH}$	—	—	30	mA

Note: * To protect chip reliability, do not exceed the output current values in table 18-5.

18.4 AC Characteristics

Figure 18-2 shows, the test conditions for the AC characteristics.

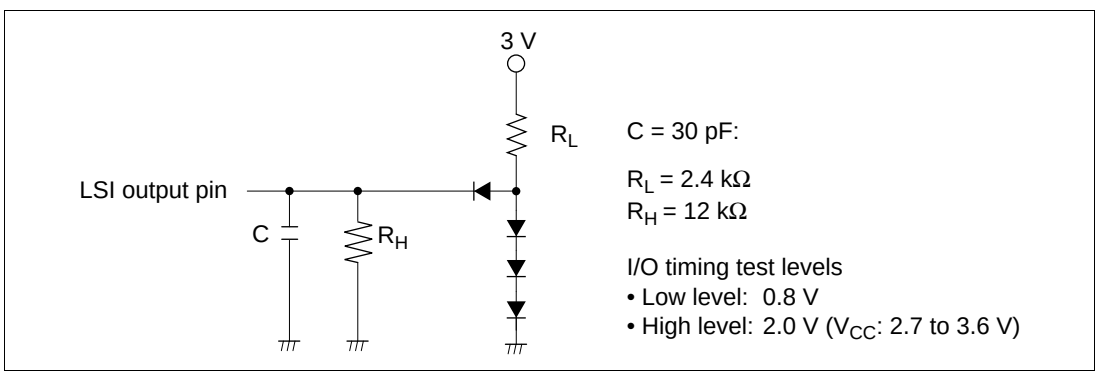


Figure 18-2 Output Load Circuit

18.4.1 Clock Timing

Table 18-6 lists the clock timing

Table 18-6 Clock Timing

Condition : V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, $V_{ref} = AV_{CC}$,
$V_{SS} = AV_{SS}$ = 0 V, ø = 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
Clock cycle time	t_{cyc}	62.5	500	ns	Figure 18-3
Clock high pulse width	t_{CH}	20	—	ns	
Clock low pulse width	t_{CL}	20	—	ns	
Clock rise time	t_{Cr}	—	10	ns	
Clock fall time	t_{Cf}	—	10	ns	
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 18-4
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	ms	Figure 17-3
External clock output stabilization delay time	t_{DEXT}	500	—	µs	Figure 18-4

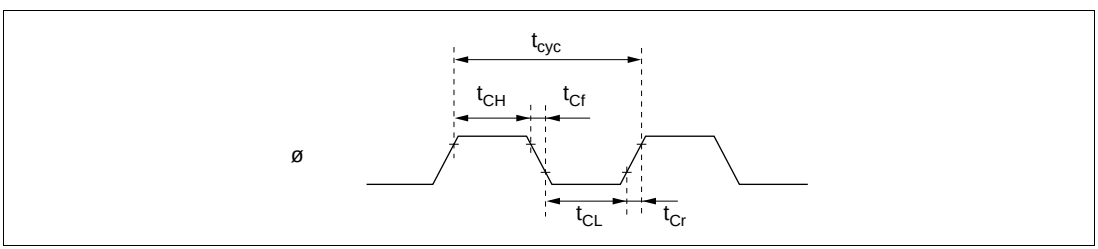


Figure 18-3 System Clock Timing

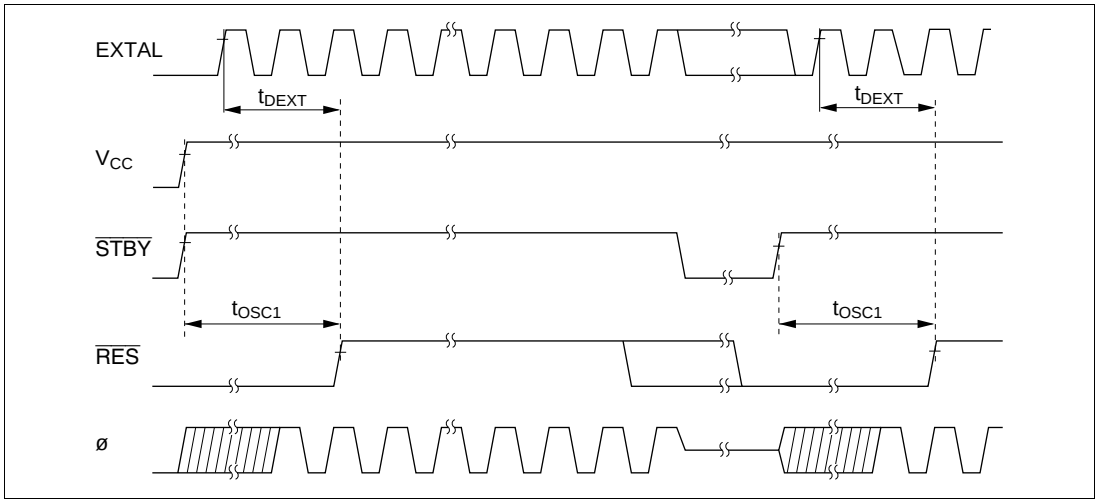


Figure 18-4 Oscillator Settling Timing

18.4.2 Control Signal Timing

Table 18-7 lists the control signal timing.

Table 18-7 Control Signal Timing

Condition : V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = AV_{CC},
V_{SS} = AV_{SS} = 0 V, ø = 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	250	—	ns	Figure 18-5
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
$\overline{MRES}$ setup time	t_{MRESS}	250	—	ns	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	250	—	ns	Figure 18-6
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	250	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

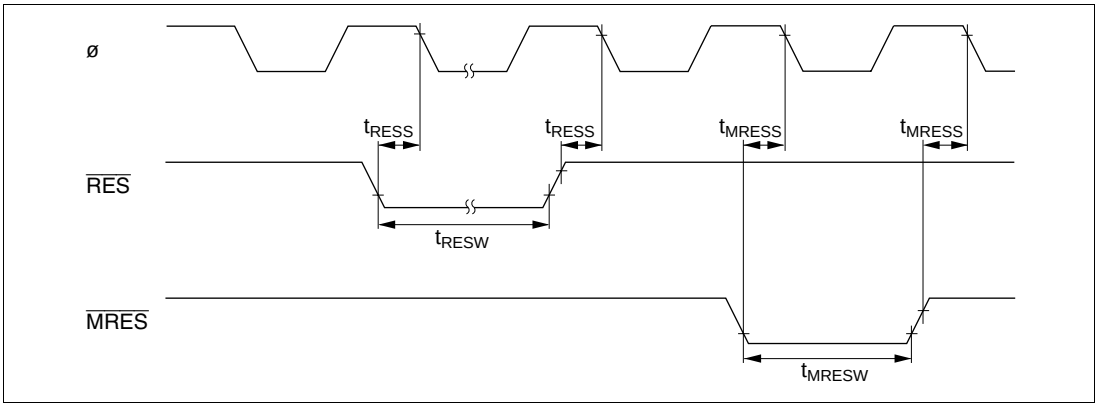


Figure 18-5 Reset Input Timing

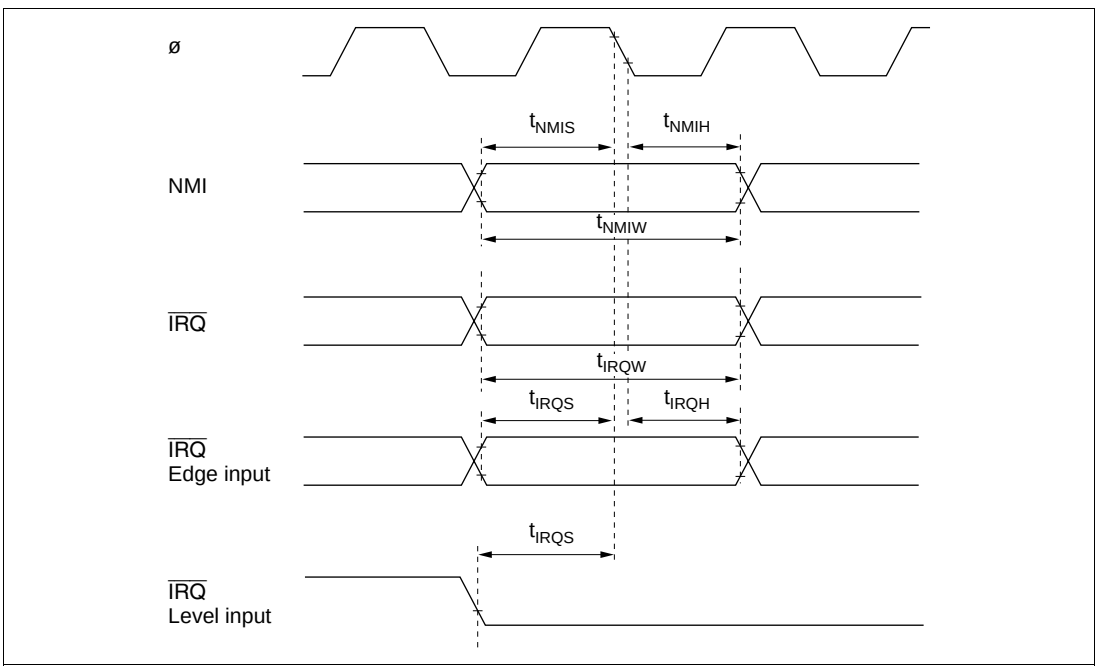


Figure 18-6 Interrupt Input Timing

18.4.3 Bus Timing

Table 18-8 lists the bus timing.

Table 18-8 Bus Timing

Condition: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, $V_{ref} = AV_{CC}$,
$V_{SS} = AV_{SS}$ = 0 V, ø = 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
Address delay time	t_{AD}	—	50	ns	Figure 18-7, Figure 18-8, Figure 18-10
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 30$	—	ns	
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 15$	—	ns	
$\overline{CS}$ delay time	t_{CSD}	—	50	ns	Figure 18-7, Figure 18-8
$\overline{AS}$ delay time	t_{ASD}	—	50	ns	Figure 18-7, Figure 18-8, Figure 18-10
$\overline{RD}$ delay time 1	t_{RSD1}	—	50	ns	Figure 18-7, Figure 18-8
$\overline{RD}$ delay time 2	t_{RSD2}	—	50	ns	Figure 18-7, Figure 18-8, Figure 18-10
Read data setup time	t_{RDS}	30	—	ns	Figure 18-7, Figure 18-8,
Read data hold time	t_{RDH}	0	—	ns	Figure 18-10
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 65$	ns	Figure 18-7
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc} - 65$	ns	Figure 18-7, Figure 18-10
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 65$	ns	Figure 18-8
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 65$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	50	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	50	ns	Figure 18-7, Figure 18-8
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 30$	—	ns	Figure 18-7
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 30$	—	ns	Figure 18-8

Item	Symbol	Min	Max	Unit	Test Conditions
Write data delay time	t_{WDD}	—	70	ns	Figure 18-7, Figure 18-8
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 30$	—	ns	Figure 18-8
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 15$	—	ns	Figure 18-7, Figure 18-8
$\overline{\text{WAIT}}$ setup time	t_{WTS}	50	—	ns	Figure 18-9
$\overline{\text{WAIT}}$ hold time	t_{WTH}	10	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BRQS}	50	—	ns	Figure 18-11
$\overline{\text{BACK}}$ delay time	t_{BACD}	—	50	ns	
Bus-floating time	t_{BZD}	—	80	ns	

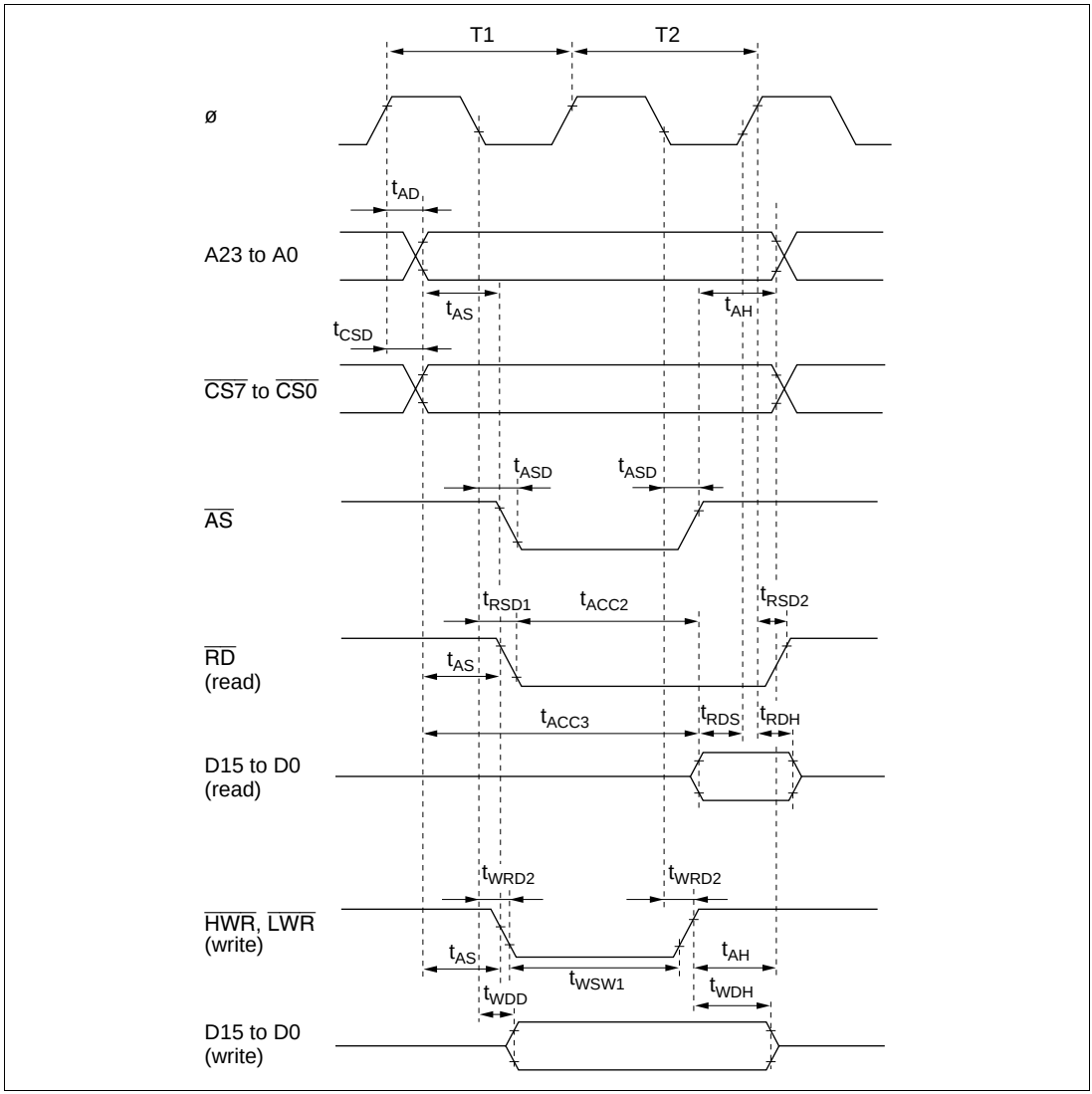


Figure 18-7 Basic Bus Timing (Two-State Access)

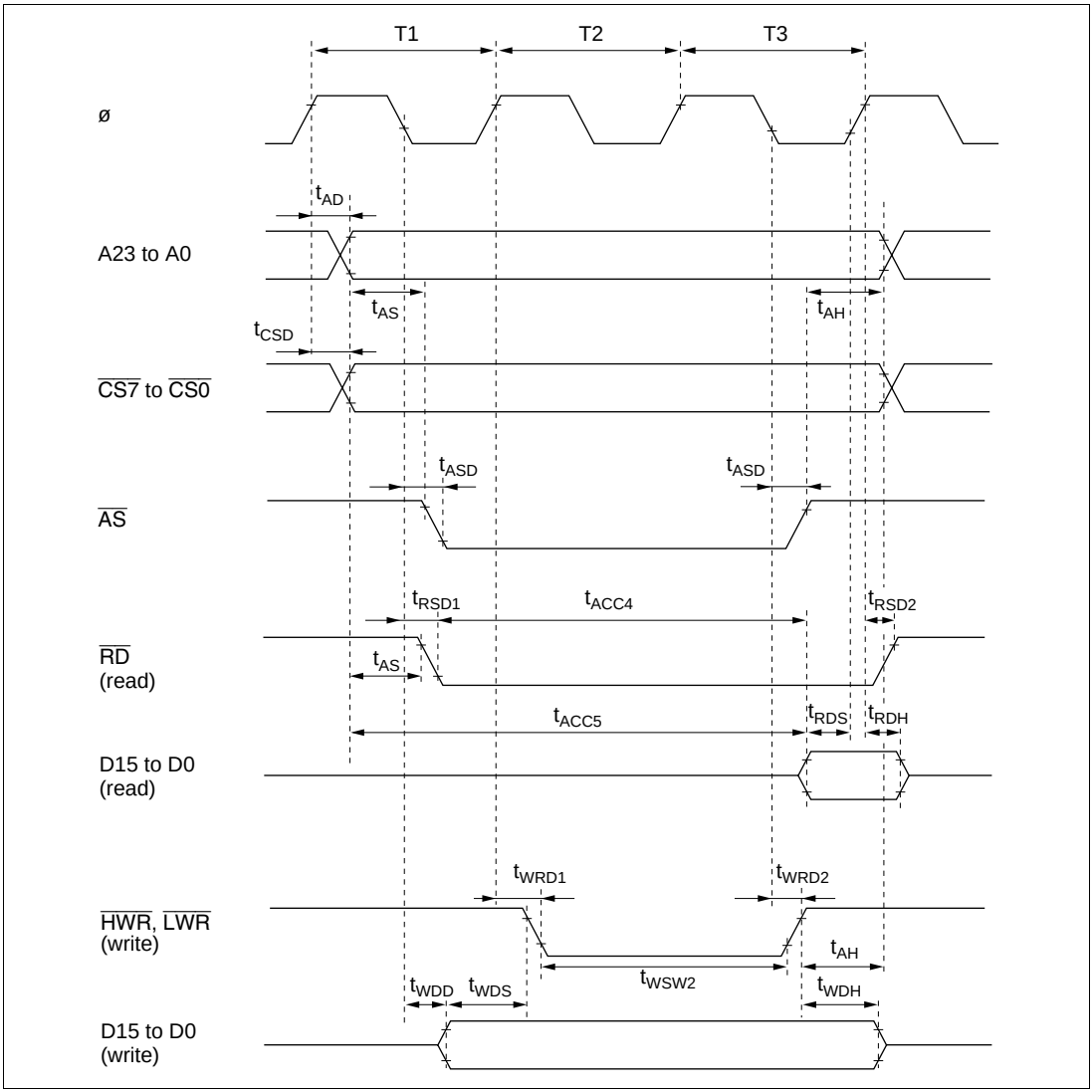


Figure 18-8 Basic Bus Timing (Three-State Access)

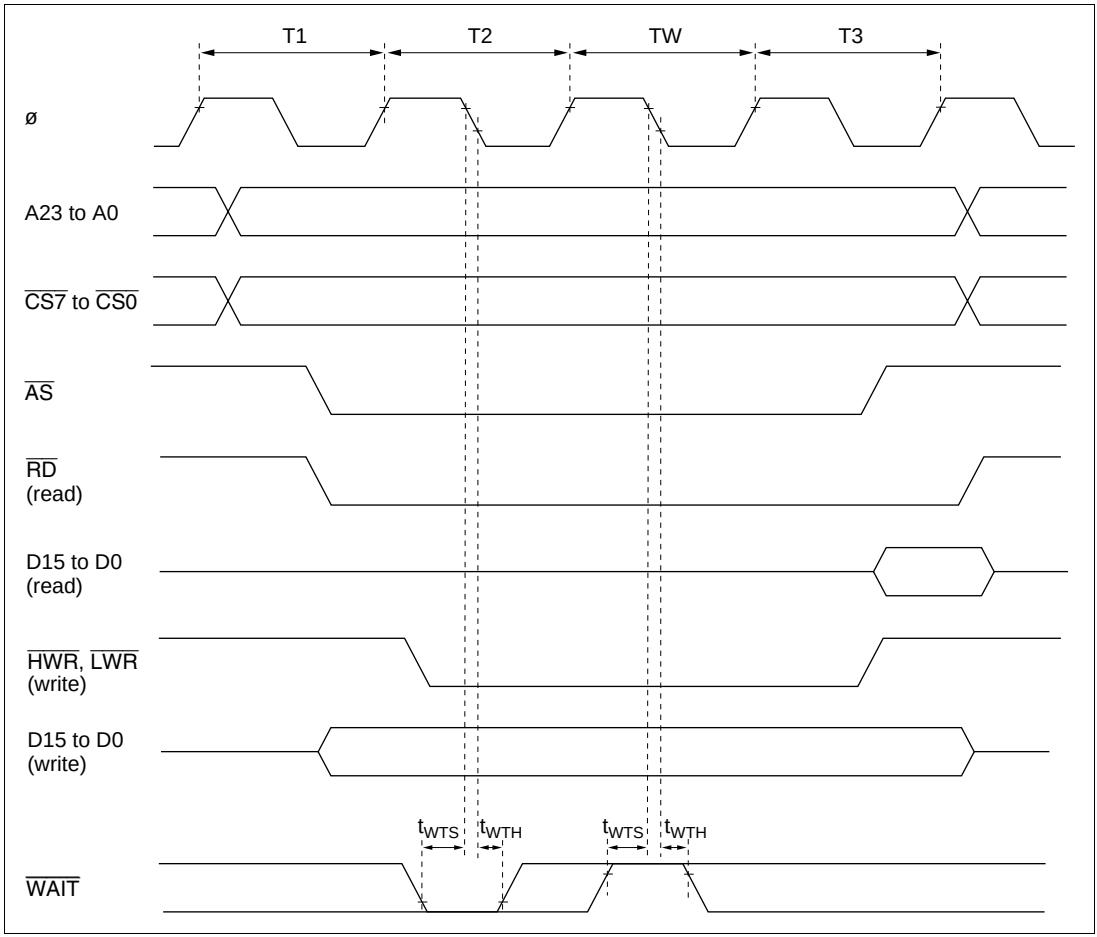


Figure 18-9 Basic Bus Timing (Three-State Access with One Wait State)

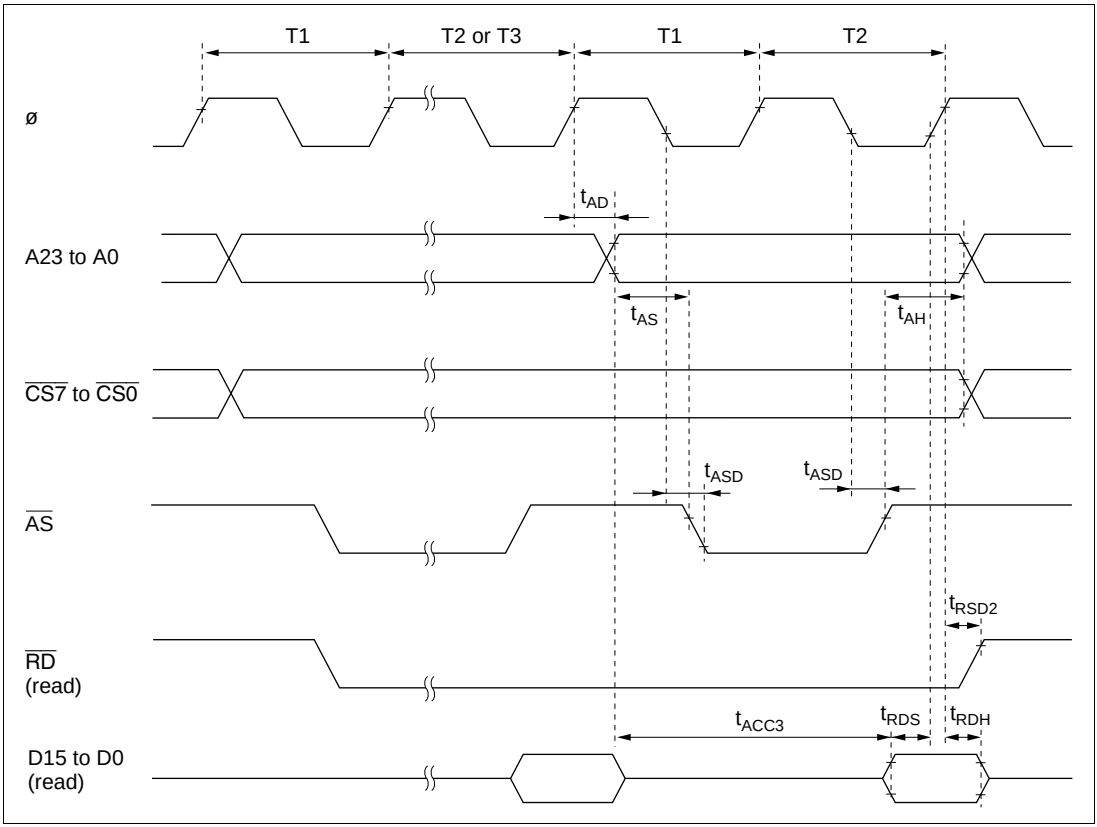


Figure 18-10 Burst ROM Access Timing (Two-State Access)

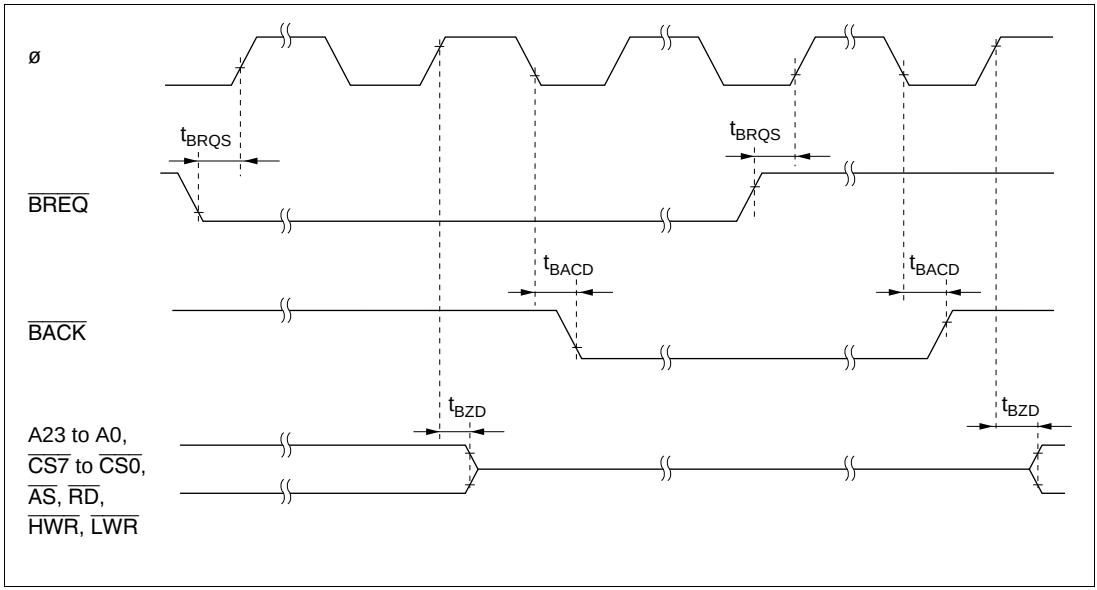


Figure 18-11 External Bus Release Timing

18.4.4 Timing of On-Chip Supporting Modules

Table 18-9 lists the timing of on-chip supporting modules.

Table 18-9 Timing of On-Chip Supporting Modules

Condition: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, $V_{ref} = AV_{CC}$,
$V_{SS} = AV_{SS}$ = 0 V, ø = 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item			Symbol	Min	Max	Unit	Test Conditions
I/O port	Output data delay time		t_{PWD}	—	100	ns	Figure 18-12
	Input data setup time		t_{PRS}	50	—		
	Input data hold time		t_{PRH}	50	—		
TPU	Timer output delay time		t_{TOCD}	—	100	ns	Figure 18-13
	Timer input setup time		t_{TICS}	40	—		
	Timer clock input setup time		t_{TCKS}	40	—	ns	Figure 18-14
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 18-15
		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}	—	100	ns	Figure 18-16
	Receive data setup time (synchronous)		t_{RXS}	75	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	75	—	ns	

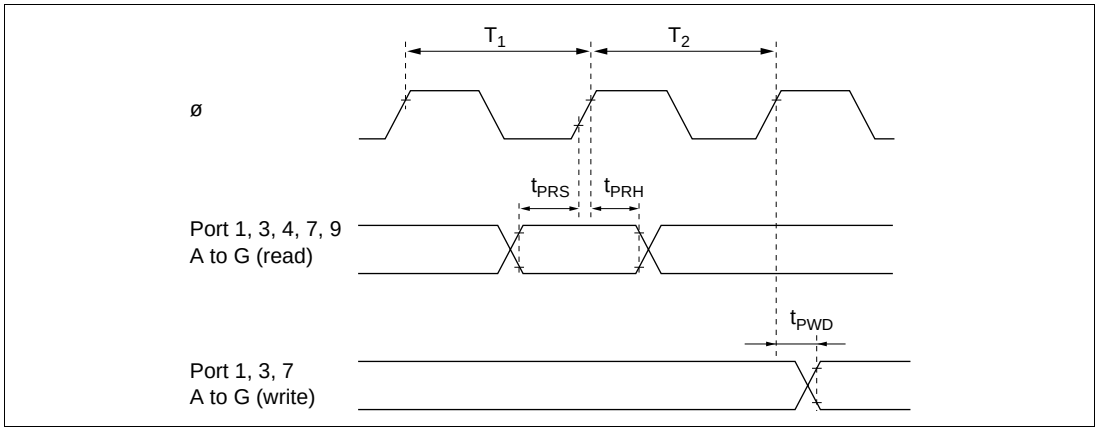


Figure 18-12 I/O Port Input/Output Timing

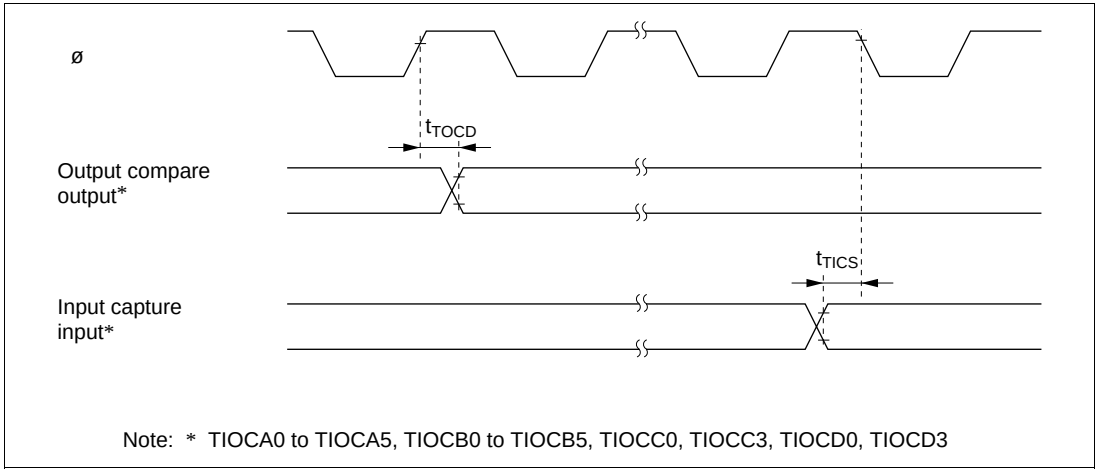


Figure 18-13 TPU Input/Output Timing

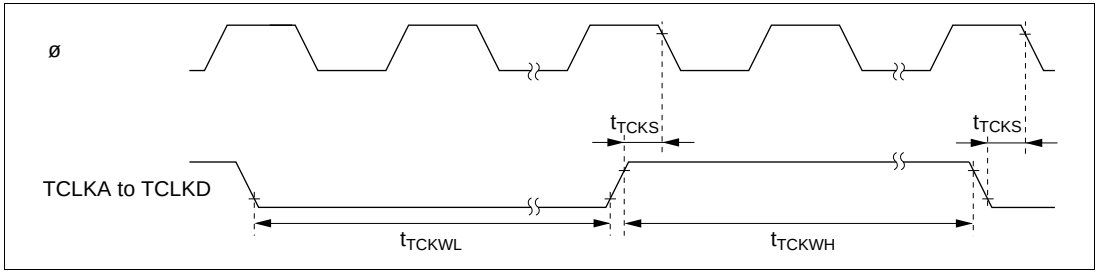


Figure 18-14 TPU Clock Input Timing

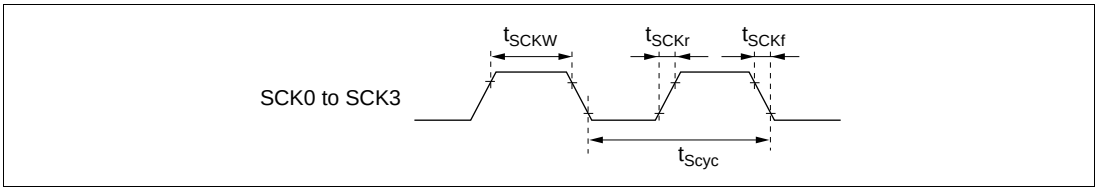


Figure 18-15 SCK Clock Input Timing

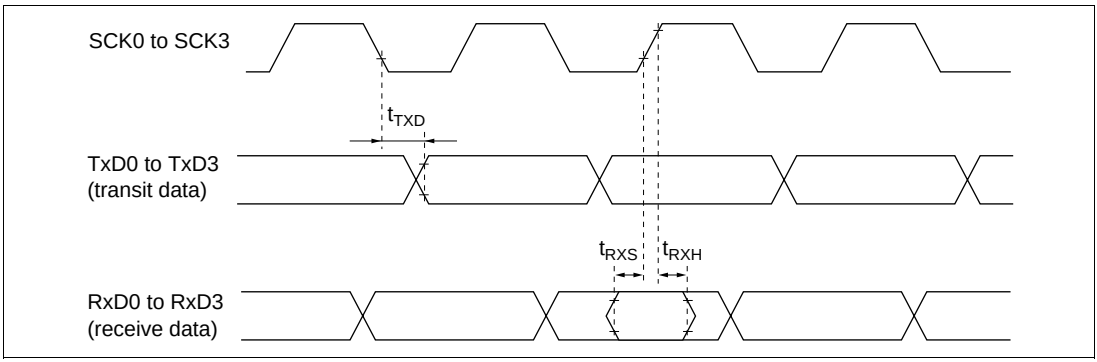


Figure 18-16 SCI Input/Output Timing (Clock Synchronous Mode)

18.4.5 DMAC Timing

Table 18-10 lists the DMAC timing.

Table 18-10 DMAC Timing

Condition: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = AV_{CC},
V_{SS} = AV_{SS} = 0 V, ø = 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{\text{DREQ}}$ setup time	t_{DRQS}	40	—	ns	Figure 18-18
$\overline{\text{DREQ}}$ hold time	t_{DRQH}	10	—		
$\overline{\text{DREQ}}$ delay time	t_{TED}	—	50		Figure 18-17

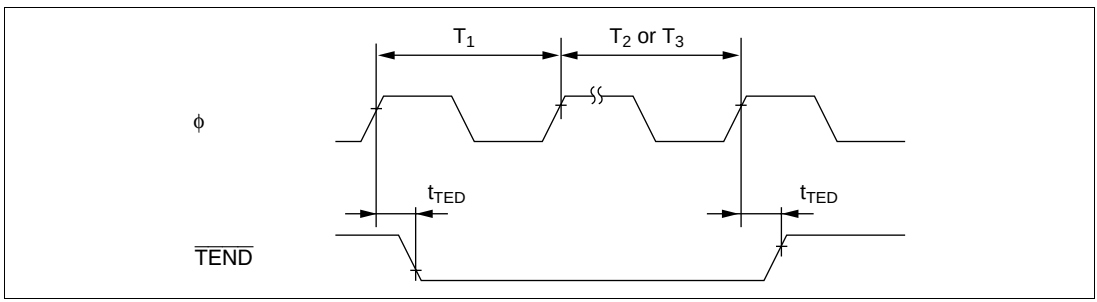


Figure 18-17 DMAC $\overline{\text{TEND}}$ Output Timing

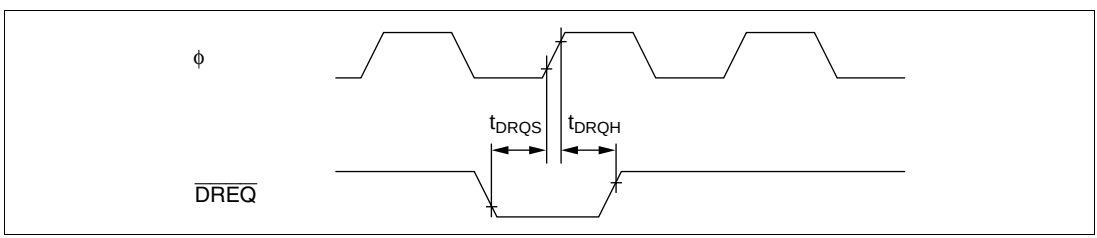


Figure 18-18 DMAC $\overline{\text{DREQ}}$ Output Timing

18.5 D/A Convervion Characteristics

Table 18-11 lists the D/A conversion characteristics.

Table 18-11 D/A Conversion Characteristics

Condition: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, V_{ref} = AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ø = 2 to 16 MHz, T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Item	Min	Typ	Max	Unit	Test Conditions
Resolution	8	8	8	bit	
Conversion time	—	—	10	µs	20-pF capacitive load
Absolute accuracy	—	±2.0	±3.0	LSB	2-MΩ resistive load
	—	—	±2.0	LSB	4-MΩ resistive load

18.6 Flash Memory Characteristics

Table 18-12 lists the flash memory characteristics.

Table 18-12 Flash Memory Characteristics

Conditions: V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V, $V_{ref} = AV_{CC}$, $V_{SS} = AV_{SS}$ = 0 V,
V_{CC} = 3.0 V to 3.6 V(program/erase operating voltage range),
T_a = -20°C to +75°C (program/erase operating temperature range)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Programming time*1, *2, *4		t_P	—	40	200	ms/128 bytes	
Erase time*1, *3, *5		t_E	—	20	1000	ms/block	
Rewrite time		N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE1 bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after PSU1 bit setting*1	t_{spsu}	50	50	—	µs	
	Wait time after P1 bit setting*1, *4	t_{sp10}	8	10	12	µs	
		t_{sp30}	28	30	32	µs	1 ≤ n ≤ 6
		t_{sp200}	198	200	202	µs	7 ≤ n ≤ 1000
	Wait time after P1 bit clearing*1	t_{cp}	5	5	—	µs	
	Wait time after PSU1 bit clearing*1	t_{cpsu}	5	5	—	µs	
	Wait time after PV1 bit setting*1	t_{spv}	4	4	—	µs	
	Wait time after H'FF dummy write*1	t_{spvr}	2	2	—	µs	
	Wait time after PV1 bit clearing*1	t_{cpv}	2	2	—	µs	
	Wait time after SWE1 bit clearing*1	t_{cswe}	100	100	—	µs	
	Maximum number of writes*1, *4	N1	—	—	6*4	Times	
		N2	—	—	994*4	Times	
Erasing	Wait time after SWE1 bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after ESU1 bit setting*1	t_{sesu}	100	100	—	µs	
	Wait time after E1 bit setting*1, *5	t_{se}	10	10	100	ms	
	Wait time after E1 bit clearing*1	t_{ce}	10	10	—	µs	
	Wait time after ESU1 bit clearing*1	t_{cesu}	10	10	—	µs	
	Wait time after EV1 bit setting*1	t_{sev}	20	20	—	µs	
	Wait time after H'FF dummy write*1	t_{sevr}	2	2	—	µs	
	Wait time after EV1 bit clearing*1	t_{sev}	4	4	—	µs	
	Wait time after SWE1 bit clearing*1	t_{cswe}	100	100	—	µs	
	Maximum number of erases*1, *5	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 P1 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 E1 bit is set in FLMCR1. Does not include the erase-verify time.)

*4 Maximum programming time.

t_P(max) = Wait time after P1 bit setting (t_{sp}) × maximum number of writes (N)
= (t_{sp30} + t_{sp10}) × 6 + (t_{sp200}) × 994

*5 For the maximum erase time (t_E) max), the following relationship applies between the wait time after E1 bit setting (t_{se}) and the maximum number of erase (N):

t_E(max) = Wait time after E1 bit setting (t_{se}) × maximum number of erases (N)

18.7 Usage Note

Although both the F-ZTAT and mask ROM versions fully meet the electrical specifications listed in this manual, due to differences in the fabrication process, the on-chip ROM, and the layout patterns, there will be differences in the actual values of the electrical characteristics, the operating margins, the noise margins, and other aspects.

Therefore, if a system is evaluated using the F-ZTAT version, a similar evaluation should also be performed using the mask ROM version.

Appendix A Instruction Set

A.1 Instruction List

Operand Notation

Rd	General register (destination)*1
Rs	General register (source)*1
Rn	General register*1
ERn	General register (32-bit register)
MAC	Multiply-and-accumulate register (32-bit register)*2
(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
+	Add
–	Subtract
×	Multiply
÷	Divide
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right
¬	Logical NOT (logical complement)
( ) < >	Contents of operand
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Notes: *1 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).

*2 The MAC register cannot be used in the H8S/2214.

Condition Code Notation

Symbol	
↕	Changes according to the result of instruction
*	Undetermined (no guaranteed value)
0	Always cleared to 0
1	Always set to 1
—	Not affected by execution of the instruction

Table A-1 Data Transfer Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
MOV	MOV.B #xx:8,Rd	B	2									#xx:8→Rd8	—	—	↕	↕	0	—	1
	MOV.B Rs,Rd	B		2								Rs8→Rd8	—	—	↕	↕	0	—	1
	MOV.B @ERs,Rd	B			2							@ERs→Rd8	—	—	↕	↕	0	—	2
	MOV.B @(d:16,ERs),Rd	B				4						@(d:16,ERs)→Rd8	—	—	↕	↕	0	—	3
	MOV.B @(d:32,ERs),Rd	B				8						@(d:32,ERs)→Rd8	—	—	↕	↕	0	—	5
	MOV.B @ERs+,Rd	B					2					@ERs→Rd8,ERs32+1→ERs32	—	—	↕	↕	0	—	3
	MOV.B @aa:8,Rd	B						2				@aa:8→Rd8	—	—	↕	↕	0	—	2
	MOV.B @aa:16,Rd	B						4				@aa:16→Rd8	—	—	↕	↕	0	—	3
	MOV.B @aa:32,Rd	B						6				@aa:32→Rd8	—	—	↕	↕	0	—	4
	MOV.B Rs,@ERd	B			2							Rs8→@ERd	—	—	↕	↕	0	—	2
	MOV.B Rs,@(d:16,ERd)	B				4						Rs8→@(d:16,ERd)	—	—	↕	↕	0	—	3
	MOV.B Rs,@(d:32,ERd)	B				8						Rs8→@(d:32,ERd)	—	—	↕	↕	0	—	5
	MOV.B Rs,@-ERd	B					2					ERd32-1→ERd32,Rs8→@ERd	—	—	↕	↕	0	—	3
	MOV.B Rs,@aa:8	B						2				Rs8→@aa:8	—	—	↕	↕	0	—	2
	MOV.B Rs,@aa:16	B						4				Rs8→@aa:16	—	—	↕	↕	0	—	3
	MOV.B Rs,@aa:32	B						6				Rs8→@aa:32	—	—	↕	↕	0	—	4
	MOV.W #xx:16,Rd	W	4									#xx:16→Rd16	—	—	↕	↕	0	—	2
	MOV.W Rs,Rd	W		2								Rs16→Rd16	—	—	↕	↕	0	—	1
	MOV.W @ERs,Rd	W			2							@ERs→Rd16	—	—	↕	↕	0	—	2

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
MOV	MOV.W @(d:16,ERs),Rd	W				4						@(d:16,ERs)→Rd16	—	—	↕	↕	0	—	3
	MOV.W @(d:32,ERs),Rd	W				8						@(d:32,ERs)→Rd16	—	—	↕	↕	0	—	5
	MOV.W @ERs+,Rd	W					2					@ERs→Rd16,ERs32+2→ERs32	—	—	↕	↕	0	—	3
	MOV.W @aa:16,Rd	W						4				@aa:16→Rd16	—	—	↕	↕	0	—	3
	MOV.W @aa:32,Rd	W						6				@aa:32→Rd16	—	—	↕	↕	0	—	4
	MOV.W Rs,@ERd	W			2							Rs16→@ERd	—	—	↕	↕	0	—	2
	MOV.W Rs,@(d:16,ERd)	W				4						Rs16→@(d:16,ERd)	—	—	↕	↕	0	—	3
	MOV.W Rs,@(d:32,ERd)	W				8						Rs16→@(d:32,ERd)	—	—	↕	↕	0	—	5
	MOV.W Rs,@-ERd	W					2					ERd32-2→ERd32,Rs16→@ERd	—	—	↕	↕	0	—	3
	MOV.W Rs,@aa:16	W						4				Rs16→@aa:16	—	—	↕	↕	0	—	3
	MOV.W Rs,@aa:32	W						6				Rs16→@aa:32	—	—	↕	↕	0	—	4
	MOV.L #xx:32,ERd	L	6									#xx:32→ERd32	—	—	↕	↕	0	—	3
	MOV.L ERs,ERd	L		2								ERs32→ERd32	—	—	↕	↕	0	—	1
	MOV.L @ERs,ERd	L			4							@ERs→ERd32	—	—	↕	↕	0	—	4
	MOV.L @(d:16,ERs),ERd	L				6						@(d:16,ERs)→ERd32	—	—	↕	↕	0	—	5
	MOV.L @(d:32,ERs),ERd	L				10						@(d:32,ERs)→ERd32	—	—	↕	↕	0	—	7
	MOV.L @ERs+,ERd	L					4					@ERs→ERd32,ERs32+4→@ERs32	—	—	↕	↕	0	—	5
	MOV.L @aa:16,ERd	L						6				@aa:16→ERd32	—	—	↕	↕	0	—	5
	MOV.L @aa:32,ERd	L						8				@aa:32→ERd32	—	—	↕	↕	0	—	6

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
MOV	MOV.L ERs,@ERd	L			4							ERs32→@ERd	—	—	↕	↕	0	—	4
	MOV.L ERs,@(d:16,ERd)	L				6						ERs32→@(d:16,ERd)	—	—	↕	↕	0	—	5
	MOV.L ERs,@(d:32,ERd)	L				10						ERs32→@(d:32,ERd)	—	—	↕	↕	0	—	7
	MOV.L ERs,@-ERd	L					4					ERd32-4→ERd32,ERs32→@ERd	—	—	↕	↕	0	—	5
	MOV.L ERs,@aa:16	L						6				ERs32→@aa:16	—	—	↕	↕	0	—	5
	MOV.L ERs,@aa:32	L						8				ERs32→@aa:32	—	—	↕	↕	0	—	6
POP	POP.W Rn	W									2	@SP→Rn16,SP+2→SP	—	—	↕	↕	0	—	3
	POP.L ERn	L									4	@SP→ERn32,SP+4→SP	—	—	↕	↕	0	—	5
PUSH	PUSH.W Rn	W									2	SP-2→SP,Rn16→@SP	—	—	↕	↕	0	—	3
	PUSH.L ERn	L									4	SP-4→SP,ERn32→@SP	—	—	↕	↕	0	—	5
LDM	LDM @SP+,(ERm-ERn)	L									4	(@SP→ERn32,SP+4→SP) Repeated for each register restored	—	—	—	—	—	—	7/9/11 [1]
STM	STM (ERm-ERn),@-SP	L									4	(SP-4→SP,ERn32→@SP) Repeated for each register saved	—	—	—	—	—	—	7/9/11 [1]
MOVFPE	MOVFPE @aa:16,Rd	Cannot be used in the H8S/2214																	[2]
MOVTPE	MOVTPE Rs,@aa:16	Cannot be used in the H8S/2214																	[2]

Table A-2 Arithmetic Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
ADD	ADD.B #xx:8,Rd	B	2									Rd8+#xx:8→Rd8	—	↕	↕	↕	↕	↕	1
	ADD.B Rs,Rd	B		2								Rd8+Rs8→Rd8	—	↕	↕	↕	↕	↕	1
	ADD.W #xx:16,Rd	W	4									Rd16+#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2
	ADD.W Rs,Rd	W		2								Rd16+Rs16→Rd16	—	[3]	↕	↕	↕	↕	1
	ADD.L #xx:32,ERd	L	6									ERd32+#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3
	ADD.L ERs,ERd	L		2								ERd32+ERs32→ERd32	—	[4]	↕	↕	↕	↕	1
ADDX	ADDX #xx:8,Rd	B	2									Rd8+#xx:8+C→Rd8	—	↕	↕	[5]	↕	↕	1
	ADDX Rs,Rd	B		2								Rd8+Rs8+C→Rd8	—	↕	↕	[5]	↕	↕	1
ADDS	ADDS #1,ERd	L		2								ERd32+1→ERd32	—	—	—	—	—	—	1
	ADDS #2,ERd	L		2								ERd32+2→ERd32	—	—	—	—	—	—	1
	ADDS #4,ERd	L		2								ERd32+4→ERd32	—	—	—	—	—	—	1
INC	INC.B Rd	B		2								Rd8+1→Rd8	—	—	↕	↕	↕	—	1
	INC.W #1,Rd	W		2								Rd16+1→Rd16	—	—	↕	↕	↕	—	1
	INC.W #2,Rd	W		2								Rd16+2→Rd16	—	—	↕	↕	↕	—	1
	INC.L #1,ERd	L		2								ERd32+1→ERd32	—	—	↕	↕	↕	—	1
	INC.L #2,ERd	L		2								ERd32+2→ERd32	—	—	↕	↕	↕	—	1
DAA	DAA Rd	B		2								Rd8 decimal adjust→Rd8	—	*	↕	↕	*	↕	1
SUB	SUB.B Rs,Rd	B		2								Rd8-Rs8→Rd8	—	↕	↕	↕	↕	↕	1
	SUB.W #xx:16,Rd	W	4									Rd16-#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
SUB	SUB.W Rs,Rd	W		2								Rd16-Rs16→Rd16	—	[3]	↕	↕	↕	↕	1
	SUB.L #xx:32,ERd	L	6									ERd32-#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3
	SUB.L ERs,ERd	L		2								ERd32-ERs32→ERd32	—	[4]	↕	↕	↕	↕	1
SUBX	SUBX #xx:8,Rd	B	2									Rd8-#xx:8-C→Rd8	—	↕	↕	[5]	↕	↕	1
	SUBX Rs,Rd	B		2								Rd8-Rs8-C→Rd8	—	↕	↕	[5]	↕	↕	1
SUBS	SUBS #1,ERd	L		2								ERd32-1→ERd32	—	—	—	—	—	—	1
	SUBS #2,ERd	L		2								ERd32-2→ERd32	—	—	—	—	—	—	1
	SUBS #4,ERd	L		2								ERd32-4→ERd32	—	—	—	—	—	—	1
DEC	DEC.B Rd	B		2								Rd8-1→Rd8	—	—	↕	↕	↕	—	1
	DEC.W #1,Rd	W		2								Rd16-1→Rd16	—	—	↕	↕	↕	—	1
	DEC.W #2,Rd	W		2								Rd16-2→Rd16	—	—	↕	↕	↕	—	1
	DEC.L #1,ERd	L		2								ERd32-1→ERd32	—	—	↕	↕	↕	—	1
	DEC.L #2,ERd	L		2								ERd32-2→ERd32	—	—	↕	↕	↕	—	1
DAS	DAS Rd	B		2								Rd8 decimal adjust→Rd8	—	*	↕	↕	*	—	1
MULXU	MULXU.B Rs,Rd	B		2								Rd8×Rs8→Rd16 (unsigned multiplication)	—	—	—	—	—	—	12
	MULXU.W Rs,ERd	W		2								Rd16×Rs16→ERd32 (unsigned multiplication)	—	—	—	—	—	—	20
MULXS	MULXS.B Rs,Rd	B		4								Rd8×Rs8→Rd16 (signed multiplication)	—	—	↕	↕	—	—	13
	MULXS.W Rs,ERd	W		4								Rd16×Rs16→ERd32 (signed multiplication)	—	—	↕	↕	—	—	21

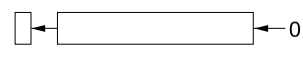
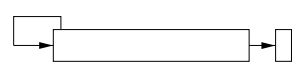
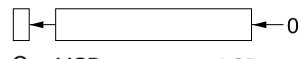
Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
DIVXU	DIVXU.B Rs,Rd	B		2								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	[6]	[7]	—	—	12
	DIVXU.W Rs,ERd	W		2								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	[6]	[7]	—	—	20
DIVXS	DIVXS.B Rs,Rd	B		4								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	[8]	[7]	—	—	13
	DIVXS.W Rs,ERd	W		4								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	[8]	[7]	—	—	21
CMP	CMP.B #xx:8,Rd	B	2									Rd8-#xx:8	—	↕	↕	↕	↕	↕	1
	CMP.B Rs,Rd	B		2								Rd8-Rs8	—	↕	↕	↕	↕	↕	1
	CMP.W #xx:16,Rd	W	4									Rd16-#xx:16	—	[3]	↕	↕	↕	↕	2
	CMP.W Rs,Rd	W		2								Rd16-Rs16	—	[3]	↕	↕	↕	↕	1
	CMP.L #xx:32,ERd	L	6									ERd32-#xx:32	—	[4]	↕	↕	↕	↕	3
	CMP.L ERs,ERd	L		2								ERd32-ERs32	—	[4]	↕	↕	↕	↕	1
NEG	NEG.B Rd	B		2								0-Rd8→Rd8	—	↕	↕	↕	↕	↕	1
	NEG.W Rd	W		2								0-Rd16→Rd16	—	↕	↕	↕	↕	↕	1
	NEG.L ERd	L		2								0-ERd32→ERd32	—	↕	↕	↕	↕	↕	1
EXTU	EXTU.W Rd	W		2								0→(<bit 15 to 8> of Rd16)	—	—	0	↕	0	—	1
	EXTU.L ERd	L		2								0→(<bit 31 to 16> of ERd32)	—	—	0	↕	0	—	1

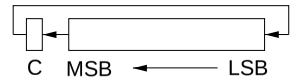
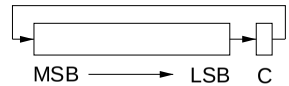
Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16)→ (<bit 15 to 8> of Rd16)	—	—	↕	↕	0	—	1
	EXTS.L ERd	L		2								(<bit 15> of ERd32)→ (<bit 31 to 16> of ERd32)	—	—	↕	↕	0	—	1
TAS	TAS @ERd*2	B			4							@ERd-0→CCR set, (1)→ (<bit 7> of @ERd)	—	—	↕	↕	0	—	4
MAC	MAC @ERn+, @ERm+	Cannot be used in the H8S/2214																	[2]
CLRMAC	CLRMAC																		
LDMAC	LDMAC ERs,MACH																		
	LDMAC ERs,MACL																		
STMAC	STMAC MACH,ERd																		
	STMAC MACL,ERd																		

Table A-3 Logical Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
AND	AND.B #xx:8,Rd	B	2									Rd8∧#xx:8→Rd8	—	—	↕	↕	0	—	1
	AND.B Rs,Rd	B		2								Rd8∧Rs8→Rd8	—	—	↕	↕	0	—	1
	AND.W #xx:16,Rd	W	4									Rd16∧#xx:16→Rd16	—	—	↕	↕	0	—	2
	AND.W Rs,Rd	W		2								Rd16∧Rs16→Rd16	—	—	↕	↕	0	—	1
	AND.L #xx:32,ERd	L	6									ERd32∧#xx:32→ERd32	—	—	↕	↕	0	—	3
	AND.L ERs,ERd	L		4								ERd32∧ERs32→ERd32	—	—	↕	↕	0	—	2
OR	OR.B #xx:8,Rd	B	2									Rd8∨#xx:8→Rd8	—	—	↕	↕	0	—	1
	OR.B Rs,Rd	B		2								Rd8∨Rs8→Rd8	—	—	↕	↕	0	—	1
	OR.W #xx:16,Rd	W	4									Rd16∨#xx:16→Rd16	—	—	↕	↕	0	—	2
	OR.W Rs,Rd	W		2								Rd16∨Rs16→Rd16	—	—	↕	↕	0	—	1
	OR.L #xx:32,ERd	L	6									ERd32∨#xx:32→ERd32	—	—	↕	↕	0	—	3
	OR.L ERs,ERd	L		4								ERd32∨ERs32→ERd32	—	—	↕	↕	0	—	2
XOR	XOR.B #xx:8,Rd	B	2									Rd8⊕#xx:8→Rd8	—	—	↕	↕	0	—	1
	XOR.B Rs,Rd	B		2								Rd8⊕Rs8→Rd8	—	—	↕	↕	0	—	1
	XOR.W #xx:16,Rd	W	4									Rd16⊕#xx:16→Rd16	—	—	↕	↕	0	—	2
	XOR.W Rs,Rd	W		2								Rd16⊕Rs16→Rd16	—	—	↕	↕	0	—	1
	XOR.L #xx:32,ERd	L	6									ERd32⊕#xx:32→ERd32	—	—	↕	↕	0	—	3
	XOR.L ERs,ERd	L		4								ERd32⊕ERs32→ERd32	—	—	↕	↕	0	—	2
NOT	NOT.B Rd	B		2								¬ Rd8→Rd8	—	—	↕	↕	0	—	1
	NOT.W Rd	W		2								¬ Rd16→Rd16	—	—	↕	↕	0	—	1
	NOT.L ERd	L		2								¬ ERd32→ERd32	—	—	↕	↕	0	—	1

Table A-4 Shift Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
SHAL	SHAL.B Rd	B		2									—	—	↕	↕	↕	↕	1
	SHAL.B #2,Rd	B		2									—	—	↕	↕	↕	↕	1
	SHAL.W Rd	W		2									—	—	↕	↕	↕	↕	1
	SHAL.W #2,Rd	W		2									—	—	↕	↕	↕	↕	1
	SHAL.L ERd	L		2									—	—	↕	↕	↕	↕	1
	SHAL.L #2,ERd	L		2									—	—	↕	↕	↕	↕	1
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHAR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	SHAR.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHAR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	SHAR.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHAR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHLL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	SHLL.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHLL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	SHLL.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHLL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
SHLR	SHLR.B Rd	B		2									—	—	0	↕	0	↕	1
	SHLR.B #2,Rd	B		2									—	—	0	↕	0	↕	1
	SHLR.W Rd	W		2									—	—	0	↕	0	↕	1
	SHLR.W #2,Rd	W		2									—	—	0	↕	0	↕	1
	SHLR.L ERd	L		2									—	—	0	↕	0	↕	1
	SHLR.L #2,ERd	L		2									—	—	0	↕	0	↕	1
ROTXL	ROTXL.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXL.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXL.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTXL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
ROTXR	ROTXR.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXR.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXR.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTXR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

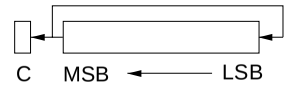
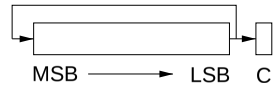
Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
ROTL	ROTL.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTL.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTL.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
ROTR	ROTR.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTR.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTR.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

Table A-5 Bit-Manipulation Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
BSET	BSET #xx:3,Rd	B		2								(#xx:3 of Rd8)←1	—	—	—	—	—	—	1
	BSET #xx:3,@ERd	B			4							(#xx:3 of @ERd)←1	—	—	—	—	—	—	4
	BSET #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←1	—	—	—	—	—	—	4
	BSET #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←1	—	—	—	—	—	—	5
	BSET #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←1	—	—	—	—	—	—	6
	BSET Rn,Rd	B		2								(Rn8 of Rd8)←1	—	—	—	—	—	—	1
	BSET Rn,@ERd	B			4							(Rn8 of @ERd)←1	—	—	—	—	—	—	4
	BSET Rn,@aa:8	B						4				(Rn8 of @aa:8)←1	—	—	—	—	—	—	4
	BSET Rn,@aa:16	B						6				(Rn8 of @aa:16)←1	—	—	—	—	—	—	5
	BSET Rn,@aa:32	B						8				(Rn8 of @aa:32)←1	—	—	—	—	—	—	6
BCLR	BCLR #xx:3,Rd	B		2								(#xx:3 of Rd8)←0	—	—	—	—	—	—	1
	BCLR #xx:3,@ERd	B			4							(#xx:3 of @ERd)←0	—	—	—	—	—	—	4
	BCLR #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←0	—	—	—	—	—	—	4
	BCLR #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←0	—	—	—	—	—	—	5
	BCLR #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←0	—	—	—	—	—	—	6
	BCLR Rn,Rd	B		2								(Rn8 of Rd8)←0	—	—	—	—	—	—	1
	BCLR Rn,@ERd	B			4							(Rn8 of @ERd)←0	—	—	—	—	—	—	4
	BCLR Rn,@aa:8	B						4				(Rn8 of @aa:8)←0	—	—	—	—	—	—	4
	BCLR Rn,@aa:16	B						6				(Rn8 of @aa:16)←0	—	—	—	—	—	—	5

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BCLR	BCLR Rn,@aa:32	B						8				(Rn8 of @aa:32)←0	—	—	—	—	—	—	6
BNOT	BNOT #xx:3,Rd	B		2								(#xx:3 of Rd8)←[¬ (#xx:3 of Rd8)]	—	—	—	—	—	—	1
	BNOT #xx:3,@ERd	B			4							(#xx:3 of @ERd)← [¬ (#xx:3 of @ERd)]	—	—	—	—	—	—	4
	BNOT #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)← [¬ (#xx:3 of @aa:8)]	—	—	—	—	—	—	4
	BNOT #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)← [¬ (#xx:3 of @aa:16)]	—	—	—	—	—	—	5
	BNOT #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)← [¬ (#xx:3 of @aa:32)]	—	—	—	—	—	—	6
	BNOT Rn,Rd	B		2								(Rn8 of Rd8)←[¬ (Rn8 of Rd8)]	—	—	—	—	—	—	1
	BNOT Rn,@ERd	B			4							(Rn8 of @ERd)←[¬ (Rn8 of @ERd)]	—	—	—	—	—	—	4
	BNOT Rn,@aa:8	B						4				(Rn8 of @aa:8)←[¬ (Rn8 of @aa:8)]	—	—	—	—	—	—	4
	BNOT Rn,@aa:16	B						6				(Rn8 of @aa:16)← [¬ (Rn8 of @aa:16)]	—	—	—	—	—	—	5
	BNOT Rn,@aa:32	B						8				(Rn8 of @aa:32)← [¬ (Rn8 of @aa:32)]	—	—	—	—	—	—	6
BTST	BTST #xx:3,Rd	B		2								¬ (#xx:3 of Rd8)→Z	—	—	—	↕	—	—	1
	BTST #xx:3,@ERd	B			4							¬ (#xx:3 of @ERd)→Z	—	—	—	↕	—	—	3
	BTST #xx:3,@aa:8	B						4				¬ (#xx:3 of @aa:8)→Z	—	—	—	↕	—	—	3
	BTST #xx:3,@aa:16	B						6				¬ (#xx:3 of @aa:16)→Z	—	—	—	↕	—	—	4

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
BTST	BTST #xx:3,@aa:32	B						8				¬ (#xx:3 of @aa:32)→Z	—	—	—	↕	—	—	5
	BTST Rn,Rd	B		2								¬ (Rn8 of Rd8)→Z	—	—	—	↕	—	—	1
	BTST Rn,@ERd	B			4							¬ (Rn8 of @ERd)→Z	—	—	—	↕	—	—	3
	BTST Rn,@aa:8	B						4				¬ (Rn8 of @aa:8)→Z	—	—	—	↕	—	—	3
	BTST Rn,@aa:16	B						6				¬ (Rn8 of @aa:16)→Z	—	—	—	↕	—	—	4
	BTST Rn,@aa:32	B						8				¬ (Rn8 of @aa:32)→Z	—	—	—	↕	—	—	5
BLD	BLD #xx:3,Rd	B		2								(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BLD #xx:3,@ERd	B			4							(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BLD #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BLD #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BLD #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BILD	BILD #xx:3,Rd	B		2								¬ (#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BILD #xx:3,@ERd	B			4							¬ (#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BILD #xx:3,@aa:8	B						4				¬ (#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BILD #xx:3,@aa:16	B						6				¬ (#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BILD #xx:3,@aa:32	B						8				¬ (#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BST	BST #xx:3,Rd	B		2								C→(#xx:3 of Rd8)	—	—	—	—	—	—	1
	BST #xx:3,@ERd	B			4							C→(#xx:3 of @ERd)	—	—	—	—	—	—	4
	BST #xx:3,@aa:8	B						4				C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
BST	BST #xx:3,@aa:16	B						6				C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5
	BST #xx:3,@aa:32	B						8				C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6
BIST	BIST #xx:3,Rd	B		2								¬ C→(#xx:3 of Rd8)	—	—	—	—	—	—	1
	BIST #xx:3,@ERd	B			4							¬ C→(#xx:3 of @ERd)	—	—	—	—	—	—	4
	BIST #xx:3,@aa:8	B						4				¬ C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4
	BIST #xx:3,@aa:16	B						6				¬ C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5
	BIST #xx:3,@aa:32	B						8				¬ C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6
BAND	BAND #xx:3,Rd	B		2								C∧(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BAND #xx:3,@ERd	B			4							C∧(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BAND #xx:3,@aa:8	B						4				C∧(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BAND #xx:3,@aa:16	B						6				C∧(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BAND #xx:3,@aa:32	B						8				C∧(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIAND	BIAND #xx:3,Rd	B		2								C∧[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIAND #xx:3,@ERd	B			4							C∧[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIAND #xx:3,@aa:8	B						4				C∧[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIAND #xx:3,@aa:16	B						6				C∧[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIAND #xx:3,@aa:32	B						8				C∧[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5
BOR	BOR #xx:3,Rd	B		2								C∨(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BOR #xx:3,@ERd	B			4							C∨(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
BOR	BOR #xx:3,@aa:8	B						4				C∨(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BOR #xx:3,@aa:16	B						6				C∨(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BOR #xx:3,@aa:32	B						8				C∨(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIOR	BIOR #xx:3,Rd	B		2								C∨[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIOR #xx:3,@ERd	B			4							C∨[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIOR #xx:3,@aa:8	B						4				C∨[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIOR #xx:3,@aa:16	B						6				C∨[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIOR #xx:3,@aa:32	B						8				C∨[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5
BXOR	BXOR #xx:3,Rd	B		2								C⊕(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BXOR #xx:3,@ERd	B			4							C⊕(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BXOR #xx:3,@aa:8	B						4				C⊕(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BXOR #xx:3,@aa:16	B						6				C⊕(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BXOR #xx:3,@aa:32	B						8				C⊕(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIXOR	BIXOR #xx:3,Rd	B		2								C⊕[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIXOR #xx:3,@ERd	B			4							C⊕[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIXOR #xx:3,@aa:8	B						4				C⊕[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIXOR #xx:3,@aa:16	B						6				C⊕[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIXOR #xx:3,@aa:32	B						8				C⊕[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5

Table A-6 Branch Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Branching Condition	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—			I	H	N	Z	V	C	Advanced
Bcc	BRA d:8(BT d:8)	—							2			if condition is true then	Always	—	—	—	—	—	—	2
	BRA d:16(BT d:16)	—							4			PC←PC+d		—	—	—	—	—	—	3
	BRN d:8(BF d:8)	—							2			else next;	Never	—	—	—	—	—	—	2
	BRN d:16(BF d:16)	—							4					—	—	—	—	—	—	3
	BHI d:8	—							2				C∨Z=0	—	—	—	—	—	—	2
	BHI d:16	—							4					—	—	—	—	—	—	3
	BLS d:8	—							2				C∨Z=1	—	—	—	—	—	—	2
	BLS d:16	—							4					—	—	—	—	—	—	3
	BCC d:B(BHS d:8)	—							2				C=0	—	—	—	—	—	—	2
	BCC d:16(BHS d:16)	—							4					—	—	—	—	—	—	3
	BCS d:8(BLO d:8)	—							2				C=1	—	—	—	—	—	—	2
	BCS d:16(BLO d:16)	—							4					—	—	—	—	—	—	3
	BNE d:8	—							2				Z=0	—	—	—	—	—	—	2
	BNE d:16	—							4					—	—	—	—	—	—	3
	BEQ d:8	—							2				Z=1	—	—	—	—	—	—	2
	BEQ d:16	—							4					—	—	—	—	—	—	3
	BVC d:8	—							2				V=0	—	—	—	—	—	—	2
	BVC d:16	—							4					—	—	—	—	—	—	3

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—	Branching Condition	I	H	N	Z	V	C	Advanced
Bcc	BVS d:8	—							2			V=1	—	—	—	—	—	—	2
	BVS d:16	—							4				—	—	—	—	—	—	3
	BPL d:8	—							2			N=0	—	—	—	—	—	—	2
	BPL d:16	—							4				—	—	—	—	—	—	3
	BMI d:8	—							2			N=1	—	—	—	—	—	—	2
	BMI d:16	—							4				—	—	—	—	—	—	3
	BGE d:8	—							2			N⊕V=0	—	—	—	—	—	—	2
	BGE d:16	—							4				—	—	—	—	—	—	3
	BLT d:8	—							2			N⊕V=1	—	—	—	—	—	—	2
	BLT d:16	—							4				—	—	—	—	—	—	3
	BGT d:8	—							2			Z∨(N⊕V)=0	—	—	—	—	—	—	2
	BGT d:16	—							4				—	—	—	—	—	—	3
	BLE d:8	—							2			Z∨(N⊕V)=1	—	—	—	—	—	—	2
	BLE d:16	—							4				—	—	—	—	—	—	3

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
JMP	JMP @ERn	—			2							PC←ERn	—	—	—	—	—	—	2
	JMP @aa:24	—						4				PC←aa:24	—	—	—	—	—	—	3
	JMP @@aa:8	—								2		PC←@aa:8	—	—	—	—	—	—	5
BSR	BSR d:8	—							2			PC→@-SP,PC←PC+d:8	—	—	—	—	—	—	4
	BSR d:16	—							4			PC→@-SP,PC←PC+d:16	—	—	—	—	—	—	5
JSR	JSR @ERn	—			2							PC→@-SP,PC←ERn	—	—	—	—	—	—	4
	JSR @aa:24	—						4				PC→@-SP,PC←aa:24	—	—	—	—	—	—	5
	JSR @@aa:8	—								2		PC→@-SP,PC←@aa:8	—	—	—	—	—	—	6
RTS	RTS	—									2	PC←@SP+	—	—	—	—	—	—	5

Table A-7 System Control Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
TRAPA	TRAPA #xx:2	—										PC→@-SP,CCR→@-SP, EXR→@-SP,<vector>→PC	1	—	—	—	—	—	8 [9]
RTE	RTE	—										EXR←@SP+,CCR←@SP+, PC←@SP+	↕	↕	↕	↕	↕	↕	5 [9]
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—	—	—	2
LDC	LDC #xx:8,CCR	B	2									#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	LDC #xx:8,EXR	B	4									#xx:8→EXR	—	—	—	—	—	—	2
	LDC Rs,CCR	B		2								Rs8→CCR	↕	↕	↕	↕	↕	↕	1
	LDC Rs,EXR	B		2								Rs8→EXR	—	—	—	—	—	—	1
	LDC @ERs,CCR	W			4							@ERs→CCR	↕	↕	↕	↕	↕	↕	3
	LDC @ERs,EXR	W			4							@ERs→EXR	—	—	—	—	—	—	3
	LDC @(d:16,ERs),CCR	W				6						@(d:16,ERs)→CCR	↕	↕	↕	↕	↕	↕	4
	LDC @(d:16,ERs),EXR	W				6						@(d:16,ERs)→EXR	—	—	—	—	—	—	4
	LDC @(d:32,ERs),CCR	W				10						@(d:32,ERs)→CCR	↕	↕	↕	↕	↕	↕	6
	LDC @(d:32,ERs),EXR	W				10						@(d:32,ERs)→EXR	—	—	—	—	—	—	6
	LDC @ERs+,CCR	W					4					@ERs→CCR,ERs32+2→ERs32	↕	↕	↕	↕	↕	↕	4
	LDC @ERs+,EXR	W					4					@ERs→EXR,ERs32+2→ERs32	—	—	—	—	—	—	4
	LDC @aa:16,CCR	W						6				@aa:16→CCR	↕	↕	↕	↕	↕	↕	4
	LDC @aa:16,EXR	W						6				@aa:16→EXR	—	—	—	—	—	—	4
	LDC @aa:32,CCR	W						8				@aa:32→CCR	↕	↕	↕	↕	↕	↕	5
	LDC @aa:32,EXR	W						8				@aa:32→EXR	—	—	—	—	—	—	5

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
STC	STC CCR,Rd	B		2								CCR→Rd8	—	—	—	—	—	—	1
	STC EXR,Rd	B		2								EXR→Rd8	—	—	—	—	—	—	1
	STC CCR,@ERd	W			4							CCR→@ERd	—	—	—	—	—	—	3
	STC EXR,@ERd	W			4							EXR→@ERd	—	—	—	—	—	—	3
	STC CCR,@(d:16,ERd)	W				6						CCR→@(d:16,ERd)	—	—	—	—	—	—	4
	STC EXR,@(d:16,ERd)	W				6						EXR→@(d:16,ERd)	—	—	—	—	—	—	4
	STC CCR,@(d:32,ERd)	W				10						CCR→@(d:32,ERd)	—	—	—	—	—	—	6
	STC EXR,@(d:32,ERd)	W				10						EXR→@(d:32,ERd)	—	—	—	—	—	—	6
	STC CCR,@-ERd	W					4					ERd32-2→ERd32,CCR→@ERd	—	—	—	—	—	—	4
	STC EXR,@-ERd	W					4					ERd32-2→ERd32,EXR→@ERd	—	—	—	—	—	—	4
	STC CCR,@aa:16	W						6				CCR→@aa:16	—	—	—	—	—	—	4
	STC EXR,@aa:16	W						6				EXR→@aa:16	—	—	—	—	—	—	4
	STC CCR,@aa:32	W						8				CCR→@aa:32	—	—	—	—	—	—	5
	STC EXR,@aa:32	W						8				EXR→@aa:32	—	—	—	—	—	—	5
ANDC	ANDC #xx:8,CCR	B	2									CCR∧#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	ANDC #xx:8,EXR	B	4									EXR∧#xx:8→EXR	—	—	—	—	—	—	2
ORC	ORC #xx:8,CCR	B	2									CCR∨#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	ORC #xx:8,EXR	B	4									EXR∨#xx:8→EXR	—	—	—	—	—	—	2
XORC	XORC #xx:8,CCR	B	2									CCR⊕#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	XORC #xx:8,EXR	B	4									EXR⊕#xx:8→EXR	—	—	—	—	—	—	2
NOP	NOP	—									2	PC←PC+2	—	—	—	—	—	—	1

Table A-8 Block Transfer Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
EEPMOV	EEPMOV.B	—									4	if R4L≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4L-1→R4L Until R4L=0 else next;	—	—	—	—	—	—	4+2n *3
	EEPMOV.W	—									4	if R4≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4-1→R4 Until R4=0 else next;	—	—	—	—	—	—	4+2n *3

Notes: *1 The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory.
*2 This instruction should be used with the ER0, ER1, ER4, or ER5 general register only.
*3 n is the initial value of R4L or R4.
[1] Seven states for saving or restoring two registers, nine states for three registers, or eleven states for four registers.
[2] Cannot be used in the H8S/2214.
[3] Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.
[4] Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.
[5] Retains its previous value when the result is zero; otherwise cleared to 0.
[6] Set to 1 when the divisor is negative; otherwise cleared to 0.
[7] Set to 1 when the divisor is zero; otherwise cleared to 0.
[8] Set to 1 when the quotient is negative; otherwise cleared to 0.
[9] One additional state is required for execution when EXR is valid.

A.2 Instruction Codes

Table A-9 shows the instruction codes.

Table A-9 Instruction Codes

Instruction	Mnemonic	Size	Instruction Format									
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
ADD	ADD.B #xx:8,Rd	B	8 rd	IMM								
	ADD.B Rs,Rd	B	0 8	rs rd								
	ADD.W #xx:16,Rd	W	7 9	1 rd	IMM							
	ADD.W Rs,Rd	W	0 9	rs rd								
	ADD.L #xx:32,ERd	L	7 A	1 0 erd	IMM							
	ADD.L ERs,ERd	L	0 A	1 ers 0 erd								
ADDS	ADDS #1,ERd	L	0 B	0 0 erd								
	ADDS #2,ERd	L	0 B	8 0 erd								
	ADDS #4,ERd	L	0 B	9 0 erd								
ADDX	ADDX #xx:8,Rd	B	9 rd	IMM								
	ADDX Rs,Rd	B	0 E	rs rd								
AND	AND.B #xx:8,Rd	B	E rd	IMM								
	AND.B Rs,Rd	B	1 6	rs rd								
	AND.W #xx:16,Rd	W	7 9	6 rd	IMM							
	AND.W Rs,Rd	W	6 6	rs rd								
	AND.L #xx:32,ERd	L	7 A	6 0 erd	IMM							
	AND.L ERs,ERd	L	0 1	F 0	6 6	0 ers 0 erd						
ANDC	ANDC #xx:8,CCR	B	0 6	IMM								
	ANDC #xx:8,EXR	B	0 1	4 1	0 6	IMM						
BAND	BAND #xx:3,Rd	B	7 6	0 IMM rd								
	BAND #xx:3,@ERd	B	7 C	0 erd 0	7 6	0 IMM 0						
	BAND #xx:3,@aa:8	B	7 E	abs	7 6	0 IMM 0						
	BAND #xx:3,@aa:16	B	6 A	1 0	abs		7 6	0 IMM 0				
	BAND #xx:3,@aa:32	B	6 A	3 0	abs				7 6	0 IMM 0		
Bcc	BRA d:8 (BT d:8)	—	4 0	disp								
	BRA d:16 (BT d:16)	—	5 8	0 0	disp							
	BRN d:8 (BF d:8)	—	4 1	disp								
	BRN d:16 (BF d:16)	—	5 8	1 0	disp							

Instruction	Mnemonic	Size	Instruction Format									
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
Bcc	BHI d:8	—	4 2	disp								
	BHI d:16	—	5 8	2 0	disp							
	BLS d:8	—	4 3	disp								
	BLS d:16	—	5 8	3 0	disp							
	BCC d:8 (BHS d:8)	—	4 4	disp								
	BCC d:16 (BHS d:16)	—	5 8	4 0	disp							
	BCS d:8 (BLO d:8)	—	4 5	disp								
	BCS d:16 (BLO d:16)	—	5 8	5 0	disp							
	BNE d:8	—	4 6	disp								
	BNE d:16	—	5 8	6 0	disp							
	BEQ d:8	—	4 7	disp								
	BEQ d:16	—	5 8	7 0	disp							
	BVC d:8	—	4 8	disp								
	BVC d:16	—	5 8	8 0	disp							
	BVS d:8	—	4 9	disp								
	BVS d:16	—	5 8	9 0	disp							
	BPL d:8	—	4 A	disp								
	BPL d:16	—	5 8	A 0	disp							
	BMI d:8	—	4 B	disp								
	BMI d:16	—	5 8	B 0	disp							
	BGE d:8	—	4 C	disp								
	BGE d:16	—	5 8	C 0	disp							
	BLT d:8	—	4 D	disp								
	BLT d:16	—	5 8	D 0	disp							
	BGT d:8	—	4 E	disp								
	BGT d:16	—	5 8	E 0	disp							
	BLE d:8	—	4 F	disp								
	BLE d:16	—	5 8	F 0	disp							

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
BCLR	BCLR #xx:3,Rd	B	7	2	0 IMM	rd														
	BCLR #xx:3,@ERd	B	7	D	0 erd	0	7	2	0 IMM	0										
	BCLR #xx:3,@aa:8	B	7	F	abs		7	2	0 IMM	0										
	BCLR #xx:3,@aa:16	B	6	A	1	8	abs				7	2	0 IMM	0						
	BCLR #xx:3,@aa:32	B	6	A	3	8	abs								7	2	0 IMM	0		
	BCLR Rn,Rd	B	6	2	rn	rd														
	BCLR Rn,@ERd	B	7	D	0 erd	0	6	2	rn	0										
	BCLR Rn,@aa:8	B	7	F	abs		6	2	rn	0										
	BCLR Rn,@aa:16	B	6	A	1	8	abs				6	2	rn	0						
	BCLR Rn,@aa:32	B	6	A	3	8	abs								6	2	rn	0		
BIAND	BIAND #xx:3,Rd	B	7	6	1 IMM	rd														
	BIAND #xx:3,@ERd	B	7	C	0 erd	0	7	6	1 IMM	0										
	BIAND #xx:3,@aa:8	B	7	E	abs		7	6	1 IMM	0										
	BIAND #xx:3,@aa:16	B	6	A	1	0	abs				7	6	1 IMM	0						
	BIAND #xx:3,@aa:32	B	6	A	3	0	abs								7	6	1 IMM	0		
BILD	BILD #xx:3,Rd	B	7	7	1 IMM	rd														
	BILD #xx:3,@ERd	B	7	C	0 erd	0	7	7	1 IMM	0										
	BILD #xx:3,@aa:8	B	7	E	abs		7	7	1 IMM	0										
	BILD #xx:3,@aa:16	B	6	A	1	0	abs				7	7	1 IMM	0						
	BILD #xx:3,@aa:32	B	6	A	3	0	abs								7	7	1 IMM	0		
BIOR	BIOR #xx:3,Rd	B	7	4	1 IMM	rd														
	BIOR #xx:3,@ERd	B	7	C	0 erd	0	7	4	1 IMM	0										
	BIOR #xx:3,@aa:8	B	7	E	abs		7	4	1 IMM	0										
	BIOR #xx:3,@aa:16	B	6	A	1	0	abs				7	4	1 IMM	0						
	BIOR #xx:3,@aa:32	B	6	A	3	0	abs								7	4	1 IMM	0		

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
BIST	BIST #xx:3,Rd	B	6	7	1 IMM	rd														
	BIST #xx:3,@ERd	B	7	D	0 erd	0	6	7	1 IMM	0										
	BIST #xx:3,@aa:8	B	7	F	abs		6	7	1 IMM	0										
	BIST #xx:3,@aa:16	B	6	A	1	8	abs				6	7	1 IMM	0						
	BIST #xx:3,@aa:32	B	6	A	3	8	abs								6	7	1 IMM	0		
BIXOR	BIXOR #xx:3,Rd	B	7	5	1 IMM	rd														
	BIXOR #xx:3,@ERd	B	7	C	0 erd	0	7	5	1 IMM	0										
	BIXOR #xx:3,@aa:8	B	7	E	abs		7	5	1 IMM	0										
	BIXOR #xx:3,@aa:16	B	6	A	1	0	abs				7	5	1 IMM	0						
	BIXOR #xx:3,@aa:32	B	6	A	3	0	abs								7	5	1 IMM	0		
BLD	BLD #xx:3,Rd	B	7	7	0 IMM	rd														
	BLD #xx:3,@ERd	B	7	C	0 erd	0	7	7	0 IMM	0										
	BLD #xx:3,@aa:8	B	7	E	abs		7	7	0 IMM	0										
	BLD #xx:3,@aa:16	B	6	A	1	0	abs				7	7	0 IMM	0						
	BLD #xx:3,@aa:32	B	6	A	3	0	abs								7	7	0 IMM	0		
BNOT	BNOT #xx:3,Rd	B	7	1	0 IMM	rd														
	BNOT #xx:3,@ERd	B	7	D	0 erd	0	7	1	0 IMM	0										
	BNOT #xx:3,@aa:8	B	7	F	abs		7	1	0 IMM	0										
	BNOT #xx:3,@aa:16	B	6	A	1	8	abs				7	1	0 IMM	0						
	BNOT #xx:3,@aa:32	B	6	A	3	8	abs								7	1	0 IMM	0		
	BNOT Rn,Rd	B	6	1	rn	rd														
	BNOT Rn,@ERd	B	7	D	0 erd	0	6	1	rn	0										
	BNOT Rn,@aa:8	B	7	F	abs		6	1	rn	0										
	BNOT Rn,@aa:16	B	6	A	1	8	abs				6	1	rn	0						
	BNOT Rn,@aa:32	B	6	A	3	8	abs								6	1	rn	0		

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
BOR	BOR #xx:3,Rd	B	7	4	0 IMM	rd														
	BOR #xx:3,@ERd	B	7	C	0 erd	0	7	4	0 IMM	0										
	BOR #xx:3,@aa:8	B	7	E	abs		7	4	0 IMM	0										
	BOR #xx:3,@aa:16	B	6	A	1	0	abs				7	4	0 IMM	0						
	BOR #xx:3,@aa:32	B	6	A	3	0	abs								7	4	0 IMM	0		
BSET	BSET #xx:3,Rd	B	7	0	0 IMM	rd														
	BSET #xx:3,@ERd	B	7	D	0 erd	0	7	0	0 IMM	0										
	BSET #xx:3,@aa:8	B	7	F	abs		7	0	0 IMM	0										
	BSET #xx:3,@aa:16	B	6	A	1	8	abs				7	0	0 IMM	0						
	BSET #xx:3,@aa:32	B	6	A	3	8	abs								7	0	0 IMM	0		
	BSET Rn,Rd	B	6	0	rn	rd														
	BSET Rn,@ERd	B	7	D	0 erd	0	6	0	rn	0										
	BSET Rn,@aa:8	B	7	F	abs		6	0	rn	0										
	BSET Rn,@aa:16	B	6	A	1	8	abs				6	0	rn	0						
	BSET Rn,@aa:32	B	6	A	3	8	abs								6	0	rn	0		
BSR	BSR d:8	—	5	5	disp															
	BSR d:16	—	5	C	0	0	disp													
BST	BST #xx:3,Rd	B	6	7	0 IMM	rd														
	BST #xx:3,@ERd	B	7	D	0 erd	0	6	7	0 IMM	0										
	BST #xx:3,@aa:8	B	7	F	abs		6	7	0 IMM	0										
	BST #xx:3,@aa:16	B	6	A	1	8	abs				6	7	0 IMM	0						
	BST #xx:3,@aa:32	B	6	A	3	8	abs								6	7	0 IMM	0		
BTST	BTST #xx:3,Rd	B	7	3	0 IMM	rd														
	BTST #xx:3,@ERd	B	7	C	0 erd	0	7	3	0 IMM	0										
	BTST #xx:3,@aa:8	B	7	E	abs		7	3	0 IMM	0										
	BTST #xx:3,@aa:16	B	6	A	1	0	abs				7	3	0 IMM	0						
	BTST #xx:3,@aa:32	B	6	A	3	0	abs								7	3	0 IMM	0		
	BTST Rn,Rd	B	6	3	rn	rd														
	BTST Rn,@ERd	B	7	C	0 erd	0	6	3	rn	0										

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
BTST	BTST Rn,@aa:8	B	7	E	abs		6	3	rn	0										
	BTST Rn,@aa:16	B	6	A	1	0	abs				6	3	rn	0						
	BTST Rn,@aa:32	B	6	A	3	0	abs								6	3	rn	0		
BXOR	BXOR #xx:3,Rd	B	7	5	0	IMM	rd													
	BXOR #xx:3,@ERd	B	7	C	0	erd	0	7	5	0	IMM	0								
	BXOR #xx:3,@aa:8	B	7	E	abs		7	5	0	IMM	0									
	BXOR #xx:3,@aa:16	B	6	A	1	0	abs				7	5	0	IMM	0					
	BXOR #xx:3,@aa:32	B	6	A	3	0	abs								7	5	0	IMM	0	
CLRMAC	CLRMAC	—	Cannot be used in the H8S/2214																	
CMP	CMP.B #xx:8,Rd	B	A	rd	IMM															
	CMP.B Rs,Rd	B	1	C	rs	rd														
	CMP.W #xx:16,Rd	W	7	9	2	rd	IMM													
	CMP.W Rs,Rd	W	1	D	rs	rd														
	CMP.L #xx:32,ERd	L	7	A	2	0 erd	IMM													
	CMP.L ERs,ERd	L	1	F	1 ers	0 erd														
DAA	DAA Rd	B	0	F	0	rd														
DAS	DAS Rd	B	1	F	0	rd														
DEC	DEC.B Rd	B	1	A	0	rd														
	DEC.W #1,Rd	W	1	B	5	rd														
	DEC.W #2,Rd	W	1	B	D	rd														
	DEC.L #1,ERd	L	1	B	7	0 erd														
	DEC.L #2,ERd	L	1	B	F	0 erd														
DIVXS	DIVXS.B Rs,Rd	B	0	1	D	0	5	1	rs	rd										
	DIVXS.W Rs,ERd	W	0	1	D	0	5	3	rs	0 erd										
DIVXU	DIVXU.B Rs,Rd	B	5	1	rs	rd														
	DIVXU.W Rs,ERd	W	5	3	rs	0 erd														
EEPMOV	EEPMOV.B	—	7	B	5	C	5	9	8	F										
	EEPMOV.W	—	7	B	D	4	5	9	8	F										

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte		10th byte	
EXTS	EXTS.W Rd	W	1	7	D	rd														
	EXTS.L ERd	L	1	7	F	0 erd														
EXTU	EXTU.W Rd	W	1	7	5	rd														
	EXTU.L ERd	L	1	7	7	0 erd														
INC	INC.B Rd	B	0	A	0	rd														
	INC.W #1,Rd	W	0	B	5	rd														
	INC.W #2,Rd	W	0	B	D	rd														
	INC.L #1,ERd	L	0	B	7	0 erd														
	INC.L #2,ERd	L	0	B	F	0 erd														
JMP	JMP @ERn	—	5	9	0 ern	0														
	JMP @aa:24	—	5	A	abs															
	JMP @@aa:8	—	5	B	abs															
JSR	JSR @ERn	—	5	D	0 ern	0														
	JSR @aa:24	—	5	E	abs															
	JSR @@aa:8	—	5	F	abs															
LDC	LDC #xx:8,CCR	B	0	7	IMM															
	LDC #xx:8,EXR	B	0	1	4	1	0	7	IMM											
	LDC Rs,CCR	B	0	3	0	rs														
	LDC Rs,EXR	B	0	3	1	rs														
	LDC @ERs,CCR	W	0	1	4	0	6	9	0 ers	0										
	LDC @ERs,EXR	W	0	1	4	1	6	9	0 ers	0										
	LDC @(d:16,ERs),CCR	W	0	1	4	0	6	F	0 ers	0	disp									
	LDC @(d:16,ERs),EXR	W	0	1	4	1	6	F	0 ers	0	disp									
	LDC @(d:32,ERs),CCR	W	0	1	4	0	7	8	0 ers	0	6	B	2	0	disp					
	LDC @(d:32,ERs),EXR	W	0	1	4	1	7	8	0 ers	0	6	B	2	0	disp					
	LDC @ERs+,CCR	W	0	1	4	0	6	D	0 ers	0										
	LDC @ERs+,EXR	W	0	1	4	1	6	D	0 ers	0										
	LDC @aa:16,CCR	W	0	1	4	0	6	B	0	0	abs									
	LDC @aa:16,EXR	W	0	1	4	1	6	B	0	0	abs									

Instruction	Mnemonic	Size	Instruction Format															
			1st byte		2nd byte		3rd byte		4th byte		5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
LDC	LDC @aa:32,CCR	W	0	1	4	0	6	B	2	0	abs					
	LDC @aa:32,EXR	W	0	1	4	1	6	B	2	0	abs					
LDM	LDM.L @SP+, (ERn-ERn+1)	L	0	1	1	0	6	D	7	0 ern+1						
	LDM.L @SP+, (ERn-ERn+2)	L	0	1	2	0	6	D	7	0 ern+2						
	LDM.L @SP+, (ERn-ERn+3)	L	0	1	3	0	6	D	7	0 ern+3						
LDMAC	LDMAC ERs,MACH	L	Cannot be used in the H8S/2214													
	LDMAC ERs,MACL	L														
MAC	MAC @ERn+,@ERm+	—														
MOV	MOV.B #xx:8,Rd	B	F	rd	IMM											
	MOV.B Rs,Rd	B	0	C	rs	rd										
	MOV.B @ERs,Rd	B	6	8	0 ers	rd										
	MOV.B @(d:16,ERs),Rd	B	6	E	0 ers	rd	disp									
	MOV.B @(d:32,ERs),Rd	B	7	8	0 ers	0	6	A	2	rd	disp					
	MOV.B @ERs+,Rd	B	6	C	0 ers	rd										
	MOV.B @aa:8,Rd	B	2	rd	abs											
	MOV.B @aa:16,Rd	B	6	A	0	rd	abs									
	MOV.B @aa:32,Rd	B	6	A	2	rd	abs									
	MOV.B Rs,@ERd	B	6	8	1 erd	rs										
	MOV.B Rs,@(d:16,ERd)	B	6	E	1 erd	rs	disp									
	MOV.B Rs,@(d:32,ERd)	B	7	8	0 erd	0	6	A	A	rs	disp					
	MOV.B Rs,@-ERd	B	6	C	1 erd	rs										
	MOV.B Rs,@aa:8	B	3	rs	abs											
	MOV.B Rs,@aa :16	B	6	A	8	rs	abs									
	MOV.B Rs,@aa:32	B	6	A	A	rs	abs									
	MOV.W #xx:16,Rd	W	7	9	0	rd	IMM									
	MOV.W Rs,Rd	W	0	D	rs	rd										
	MOV.W @ERs,Rd	W	6	9	0 ers	rd										
	MOV.W @(d:16,ERs),Rd	W	6	F	0 ers	rd	disp									
	MOV.W @(d:32,ERs),Rd	W	7	8	0 ers	0	6	B	2	rd	disp					

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
MOV	MOV.W @ERs+,Rd	W	6	D	0	ers	rd													
	MOV.W @aa:16,Rd	W	6	B	0		rd	abs												
	MOV.W @aa:32,Rd	W	6	B	2		rd	abs												
	MOV.W Rs,@ERd	W	6	9	1	erd	rs													
	MOV.W Rs,@(d:16,ERd)	W	6	F	1	erd	rs	disp												
	MOV.W Rs,@(d:32,ERd)	W	7	8	0	erd	0	6	B	A	rs	disp								
	MOV.W Rs,@-ERd	W	6	D	1	erd	rs													
	MOV.W Rs,@aa:16	W	6	B	8		rs	abs												
	MOV.W Rs,@aa:32	W	6	B	A		rs	abs												
	MOV.L #xx:32,Rd	L	7	A	0		0 erd	IMM												
	MOV.L ERs,ERd	L	0	F	1	ers	0 erd													
	MOV.L @ERs,ERd	L	0	1	0		0	6	9	0	ers 0 erd									
	MOV.L @(d:16,ERs),ERd	L	0	1	0		0	6	F	0	ers 0 erd	disp								
	MOV.L @(d:32,ERs),ERd	L	0	1	0		0	7	8	0	ers 0	6	B	2	0 erd	disp				
	MOV.L @ERs+,ERd	L	0	1	0		0	6	D	0	ers 0 erd									
	MOV.L @aa:16 ,ERd	L	0	1	0		0	6	B	0	0 erd	abs								
	MOV.L @aa:32 ,ERd	L	0	1	0		0	6	B	2	0 erd	abs								
	MOV.L ERs,@ERd	L	0	1	0		0	6	9	1	erd 0 ers									
	MOV.L ERs,@(d:16,ERd)	L	0	1	0		0	6	F	1	erd 0 ers	disp								
	MOV.L ERs,@(d:32,ERd)*1	L	0	1	0		0	7	8	0	erd 0	6	B	A	0 ers	disp				
	MOV.L ERs,@-ERd	L	0	1	0		0	6	D	1	erd 0 ers									
	MOV.L ERs,@aa:16	L	0	1	0		0	6	B	8	0 ers	abs								
	MOV.L ERs,@aa:32	L	0	1	0		0	6	B	A	0 ers	abs								
MOVFPE	MOVFPE @aa:16,Rd	B	Cannot be used in the H8S/2214																	
MOVTPE	MOVTPE Rs,@aa:16	B																		
MULXS	MULXS.B Rs,Rd	B	0	1	C		0	5	0	rs	rd									
	MULXS.W Rs,ERd	W	0	1	C		0	5	2	rs	0 erd									
MULXU	MULXU.B Rs,Rd	B	5	0	rs		rd													
	MULXU.W Rs,ERd	W	5	2	rs		0 erd													

Instruction	Mnemonic	Size	Instruction Format											
			1st byte		2nd byte		3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
NEG	NEG.B Rd	B	1	7	8	rd								
	NEG.W Rd	W	1	7	9	rd								
	NEG.L ERd	L	1	7	B	0 erd								
NOP	NOP	—	0	0	0	0								
NOT	NOT.B Rd	B	1	7	0	rd								
	NOT.W Rd	W	1	7	1	rd								
	NOT.L ERd	L	1	7	3	0 erd								
OR	OR.B #xx:8,Rd	B	C	rd	IMM									
	OR.B Rs,Rd	B	1	4	rs	rd								
	OR.W #xx:16,Rd	W	7	9	4	rd	IMM							
	OR.W Rs,Rd	W	6	4	rs	rd								
	OR.L #xx:32,ERd	L	7	A	4	0 erd	IMM							
	OR.L ERs,ERd	L	0	1	F	0	6 4	0 ers 0 erd						
ORC	ORC #xx:8,CCR	B	0	4	IMM									
	ORC #xx:8,EXR	B	0	1	4	1	0 4	IMM						
POP	POP.W Rn	W	6	D	7	rn								
	POP.L ERn	L	0	1	0	0	6 D	7 0 ern						
PUSH	PUSH.W Rn	W	6	D	F	rn								
	PUSH.L ERn	L	0	1	0	0	6 D	F 0 ern						
ROTL	ROTL.B Rd	B	1	2	8	rd								
	ROTL.B #2, Rd	B	1	2	C	rd								
	ROTL.W Rd	W	1	2	9	rd								
	ROTL.W #2, Rd	W	1	2	D	rd								
	ROTL.L ERd	L	1	2	B	0 erd								
	ROTL.L #2, ERd	L	1	2	F	0 erd								

Instruction	Mnemonic	Size	Instruction Format											
			1st byte		2nd byte		3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
ROTR	ROTR.B Rd	B	1	3	8	rd								
	ROTR.B #2, Rd	B	1	3	C	rd								
	ROTR.W Rd	W	1	3	9	rd								
	ROTR.W #2, Rd	W	1	3	D	rd								
	ROTR.L ERd	L	1	3	B	0 erd								
	ROTR.L #2, ERd	L	1	3	F	0 erd								
ROTXL	ROTXL.B Rd	B	1	2	0	rd								
	ROTXL.B #2, Rd	B	1	2	4	rd								
	ROTXL.W Rd	W	1	2	1	rd								
	ROTXL.W #2, Rd	W	1	2	5	rd								
	ROTXL.L ERd	L	1	2	3	0 erd								
	ROTXL.L #2, ERd	L	1	2	7	0 erd								
ROTXR	ROTXR.B Rd	B	1	3	0	rd								
	ROTXR.B #2, Rd	B	1	3	4	rd								
	ROTXR.W Rd	W	1	3	1	rd								
	ROTXR.W #2, Rd	W	1	3	5	rd								
	ROTXR.L ERd	L	1	3	3	0 erd								
	ROTXR.L #2, ERd	L	1	3	7	0 erd								
RTE	RTE	—	5	6	7	0								
RTS	RTS	—	5	4	7	0								
SHAL	SHAL.B Rd	B	1	0	8	rd								
	SHAL.B #2, Rd	B	1	0	C	rd								
	SHAL.W Rd	W	1	0	9	rd								
	SHAL.W #2, Rd	W	1	0	D	rd								
	SHAL.L ERd	L	1	0	B	0 erd								
	SHAL.L #2, ERd	L	1	0	F	0 erd								

Instruction	Mnemonic	Size	Instruction Format															
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte	8th byte	9th byte	10th byte
SHAR	SHAR.B Rd	B	1	1	8	rd												
	SHAR.B #2, Rd	B	1	1	C	rd												
	SHAR.W Rd	W	1	1	9	rd												
	SHAR.W #2, Rd	W	1	1	D	rd												
	SHAR.L ERd	L	1	1	B	0 erd												
	SHAR.L #2, ERd	L	1	1	F	0 erd												
SHLL	SHLL.B Rd	B	1	0	0	rd												
	SHLL.B #2, Rd	B	1	0	4	rd												
	SHLL.W Rd	W	1	0	1	rd												
	SHLL.W #2, Rd	W	1	0	5	rd												
	SHLL.L ERd	L	1	0	3	0 erd												
	SHLL.L #2, ERd	L	1	0	7	0 erd												
SHLR	SHLR.B Rd	B	1	1	0	rd												
	SHLR.B #2, Rd	B	1	1	4	rd												
	SHLR.W Rd	W	1	1	1	rd												
	SHLR.W #2, Rd	W	1	1	5	rd												
	SHLR.L ERd	L	1	1	3	0 erd												
	SHLR.L #2, ERd	L	1	1	7	0 erd												
SLEEP	SLEEP	—	0	1	8	0												
STC	STC.B CCR,Rd	B	0	2	0	rd												
	STC.B EXR,Rd	B	0	2	1	rd												
	STC.W CCR,@ERd	W	0	1	4	0	6	9	1 erd	0								
	STC.W EXR,@ERd	W	0	1	4	1	6	9	1 erd	0								
	STC.W CCR,@(d:16,ERd)	W	0	1	4	0	6	F	1 erd	0	disp							
	STC.W EXR,@(d:16,ERd)	W	0	1	4	1	6	F	1 erd	0	disp							
	STC.W CCR,@(d:32,ERd)	W	0	1	4	0	7	8	0 erd	0	6	B	A	0	disp			
	STC.W EXR,@(d:32,ERd)	W	0	1	4	1	7	8	0 erd	0	6	B	A	0	disp			
	STC.W CCR,@-ERd	W	0	1	4	0	6	D	1 erd	0								
	STC.W EXR,@-ERd	W	0	1	4	1	6	D	1 erd	0								

Instruction	Mnemonic	Size	Instruction Format															
			1st byte		2nd byte		3rd byte		4th byte		5th byte	6th byte	7th byte	8th byte	9th byte	10th byte		
STC	STC.W CCR,@aa:16	W	0	1	4	0	6	B	8	0	abs							
	STC.W EXR,@aa:16	W	0	1	4	1	6	B	8	0	abs							
	STC.W CCR,@aa:32	W	0	1	4	0	6	B	A	0	abs							
	STC.W EXR,@aa:32	W	0	1	4	1	6	B	A	0	abs							
STM	STM.L(ERn-ERn+1), @-SP	L	0	1	1	0	6	D	F	0 ern								
	STM.L (ERn-ERn+2), @-SP	L	0	1	2	0	6	D	F	0 ern								
	STM.L (ERn-ERn+3), @-SP	L	0	1	3	0	6	D	F	0 ern								
STMAC	STMAC MACH,ERd	L	Cannot be used in the H8S/2214															
	STMAC MACL,ERd	L																
SUB	SUB.B Rs,Rd	B	1	8	rs	rd												
	SUB.W #xx:16,Rd	W	7	9	3	rd	IMM											
	SUB.W Rs,Rd	W	1	9	rs	rd												
	SUB.L #xx:32,ERd	L	7	A	3	0 erd	IMM											
	SUB.L ERs,ERd	L	1	A	1 ers	0 erd												
SUBS	SUBS #1,ERd	L	1	B	0	0 erd												
	SUBS #2,ERd	L	1	B	8	0 erd												
	SUBS #4,ERd	L	1	B	9	0 erd												
SUBX	SUBX #xx:8,Rd	B	B	rd	IMM													
	SUBX Rs,Rd	B	1	E	rs	rd												
TAS	TAS @ERd *2	B	0	1	E	0	7	B	0 erd	C								
TRAPA	TRAPA #x:2	—	5	7	00 IMM	0												
XOR	XOR.B #xx:8,Rd	B	D	rd	IMM													
	XOR.B Rs,Rd	B	1	5	rs	rd												
	XOR.W #xx:16,Rd	W	7	9	5	rd	IMM											
	XOR.W Rs,Rd	W	6	5	rs	rd												
	XOR.L #xx:32,ERd	L	7	A	5	0 erd	IMM											
	XOR.L ERs,ERd	L	0	1	F	0	6	5	0 ers	0 erd								

Instruction	Mnemonic	Size	Instruction Format									
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
XORC	XORC #xx:8,CCR	B	0 5	IMM								
	XORC #xx:8,EXR	B	0 1	4 1	0 5	IMM						

Notes: *1 Bit 7 of the 4th byte of the MOV.L ERs, @(d:32,ERd) instruction can be either 1 or 0.
*2 This instruction should be used with the ER0, ER1, ER4, or ER5 general register only.

Legend

IMM: Immediate data (2, 3, 8, 16, or 32 bits)
abs: Absolute address (8, 16, 24, or 32 bits)
disp: Displacement (8, 16, or 32 bits)
rs, rd, rn: Register field (4 bits specifying an 8-bit or 16-bit register. The symbols rs, rd, and rn correspond to operand symbols Rs, Rd,and Rn.)
ers, erd, ern, erm: Register field (3 bits specifying an address register or 32-bit register. The symbols ers, erd, ern, and erm correspond to operand symbols ERs, ERd, ERn, and ERm.)

The register fields specify general registers as follows.

Address Register 32-Bit Register		16-Bit Register		8-Bit Register	
Register Field	General Register	Register Field	General Register	Register Field	General Register
000	ER0	0000	R0	0000	R0H
001	ER1	0001	R1	0001	R1H
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
111	ER7	0111	R7	0111	R7H
		1000	E0	1000	R0L
		1001	E1	1001	R1L
		•	•	•	•
		•	•	•	•
		•	•	•	•
		1111	E7	1111	R7L

A.3 Operation Code Map

Tables A-10 to A-13 show the operation code map.

Table A-10 Operation Code Map (1)

Instruction code

1st byte		2nd byte	
AH	AL	BH	BL

Instruction when most significant bit of BH is 0.

Instruction when most significant bit of BH is 1.

AH \ AL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	Table A.3(2)	STC / STMAC *	LDC / LDMAC *	ORC	XORC	ANDC	LDC	ADD		Table A.11	Table A.11	MOV		ADDX	Table A.11
1	Table A.11	Table A.11	Table A.11	Table A.11	OR	XOR	AND	Table A.11	SUB		Table A.11	Table A.11	CMP		SUBX	Table A.11
2	MOV.B															
3																
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.11	JMP			BSR	JSR		
6	BSET	BNOT	BCLR	BTST	OR / BOR / BIOR	XOR / BXOR / BIXOR	AND / BAND / BIAND	BST / BIST / BLD / BILD	MOV		Table A.11	MOV				
7										Table A.11	Table A.11	EEPMOV	Table A.12			
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

Note: * Cannot be used in the H8S/2214.

Table A-11 Operation Code Map (2)

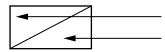
Instruction code	1st byte		2nd byte	
	AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01	MOV	LDM / STM			LDC / STC		MAC*		SLEEP		CLRMAC*		Table A.12	Table A.12	TAS	Table A.12
0A	INC								ADD							
0B	ADDS					INC		INC	ADDS					INC		INC
0F	DAA								MOV							
10	SHLL			SHLL				SHLL	SHAL			SHAL				SHAL
11	SHLR			SHLR				SHLR	SHAR			SHAR				SHAR
12	ROTXL			ROTXL				ROTXL	ROTL			ROTL				ROTL
13	ROTXR			ROTXR				ROTXR	ROTR			ROTR				ROTR
17	NOT			NOT		EXTU		EXTU	NEG			NEG		EXTS		EXTS
1A	DEC								SUB							
1B	SUBS					DEC		DEC	SUBS					DEC		DEC
1F	DAS								CMP							
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
6A	MOV	Table A.13	MOV	Table A.13	MOVFPE*				MOV		MOV		MOVTPE*			
79	MOV	ADD	CMP	SUB	OR	XOR	AND									
7A	MOV	ADD	CMP	SUB	OR	XOR	AND									

Note: * Cannot be used in the H8S/2214.

Table A-12 Operation Code Map (3)

Instruction code	1st byte		2nd byte		3rd byte		4th byte	
	AH	AL	BH	BL	CH	CL	DH	DL



Instruction when most significant bit of DH is 0.
Instruction when most significant bit of DH is 1.

CL / AH AL BH BL CH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01C05	MULXS		MULXS													
01D05		DIVXS		DIVXS												
01F06					OR	XOR	AND									
7Cr06 *1				BTST												
7Cr07 *1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
7Dr06 *1	BSET	BNOT	BCLR					BST / BIST								
7Dr07 *1	BSET	BNOT	BCLR													
7Eaa6 *2				BTST												
7Eaa7 *2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
7Faa6 *2	BSET	BNOT	BCLR					BST / BIST								
7Faa7 *2	BSET	BNOT	BCLR													

Notes: *1 r is the register specification field.
*2 aa is the absolute address specification.

Table A-13 Operation Code Map (4)

Instruction code	1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte	
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL

Instruction when most significant bit of FH is 0.
Instruction when most significant bit of FH is 1.

EL / AHALBHBLCHCLDHDLEH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6A10aaaa6*				BTST												
6A10aaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
6A18aaaa6*	BSET	BNOT	BCLR					BST / BIST								
6A18aaaa7*																

Instruction code	1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte	
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL	GH	GL	HH	HL

Instruction when most significant bit of HH is 0.
Instruction when most significant bit of HH is 1.

GL / AHALBHBL ... FHFLGH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6A30aaaaaaaa6*				BTST												
6A30aaaaaaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
6A38aaaaaaaa6*	BSET	BNOT	BCLR					BST / BIST								
6A38aaaaaaaa7*																

Note: * aa is the absolute address specification.

A.4 Number of States Required for Instruction Execution

The tables in this section can be used to calculate the number of states required for instruction execution by the H8S/2000 CPU. Table A-15 indicates the number of instruction fetch, data read/write, and other cycles occurring in each instruction. Table A-14 indicates the number of states required for each cycle, depending on its size. The number of states required for execution of an instruction can be calculated from these two tables as follows:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples: Advanced mode, program code and stack located in external memory, on-chip supporting modules accessed in two states with 8-bit bus width, external devices accessed in three states with one wait state and 16-bit bus width.

1. BSET #0, @FFFFB3:8
   From table A-15:

   $I = L = 2, \quad J = K = M = N = 0$

   From table A-14:

   $S_I = 4, \quad S_L = 2$

   Number of states required for execution = $2 \times 4 + 2 \times 2 = 12$

2. JSR @@30
   From table A-15:

   $I = J = K = 2, \quad L = M = N = 0$

   From table A-14:

   $S_I = S_J = S_K = 4$

   Number of states required for execution = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Table A-14 Number of States per Cycle

		Access Conditions					
		On-Chip Supporting Module		External Device			
				8-Bit Bus		16-Bit Bus	
Cycle		On-Chip Memory	8-Bit Bus	16-Bit Bus	2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch	S_I	1	4	2	4	6 + 2m	2	3 + m
Branch address read	S_J							
Stack operation	S_K							
Byte data access	S_L		2		2	3 + m		
Word data access	S_M		4		4	6 + 2m		
Internal operation	S_N	1	1	1	1	1	1	1

Legend

m: Number of wait states inserted into external device access

Table A-15 Number of Cycles in Instruction Execution

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8,Rd	1					
	ADD.B Rs,Rd	1					
	ADD.W #xx:16,Rd	2					
	ADD.W Rs,Rd	1					
	ADD.L #xx:32,ERd	3					
	ADD.L ERs,ERd	1					
ADDS	ADDS #1/2/4,ERd	1					
ADDX	ADDX #xx:8,Rd	1					
	ADDX Rs,Rd	1					
AND	AND.B #xx:8,Rd	1					
	AND.B Rs,Rd	1					
	AND.W #xx:16,Rd	2					
	AND.W Rs,Rd	1					
	AND.L #xx:32,ERd	3					
	AND.L ERs,ERd	2					
ANDC	ANDC #xx:8,CCR	1					
	ANDC #xx:8,EXR	2					
BAND	BAND #xx:3,Rd	1					
	BAND #xx:3,@ERd	2			1		
	BAND #xx:3,@aa:8	2			1		
	BAND #xx:3,@aa:16	3			1		
	BAND #xx:3,@aa:32	4			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
Bcc	BMI d:8	2					
	BGE d:8	2					
	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
	BRA d:16 (BT d:16)	2					1
	BRN d:16 (BF d:16)	2					1
	BHI d:16	2					1
	BLS d:16	2					1
	BCC d:16 (BHS d:16)	2					1
	BCS d:16 (BLO d:16)	2					1
	BNE d:16	2					1
	BEQ d:16	2					1
	BVC d:16	2					1
	BVS d:16	2					1
	BPL d:16	2					1
	BMI d:16	2					1
	BGE d:16	2					1
	BLT d:16	2					1
	BGT d:16	2					1
	BLE d:16	2					1
BCLR	BCLR #xx:3,Rd	1					
	BCLR #xx:3,@ERd	2			2		
	BCLR #xx:3,@aa:8	2			2		
	BCLR #xx:3,@aa:16	3			2		
	BCLR #xx:3,@aa:32	4			2		
	BCLR Rn,Rd	1					
	BCLR Rn,@ERd	2			2		
	BCLR Rn,@aa:8	2			2		
	BCLR Rn,@aa:16	3			2		
	BCLR Rn,@aa:32	4			2		

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BIAND	BIAND #xx:3,Rd	1					
	BIAND #xx:3,@ERd	2			1		
	BIAND #xx:3,@aa:8	2			1		
	BIAND #xx:3,@aa:16	3			1		
	BIAND #xx:3,@aa:32	4			1		
BILD	BILD #xx:3,Rd	1					
	BILD #xx:3,@ERd	2			1		
	BILD #xx:3,@aa:8	2			1		
	BILD #xx:3,@aa:16	3			1		
	BILD #xx:3,@aa:32	4			1		
BIOR	BIOR #xx:8,Rd	1					
	BIOR #xx:8,@ERd	2			1		
	BIOR #xx:8,@aa:8	2			1		
	BIOR #xx:8,@aa:16	3			1		
	BIOR #xx:8,@aa:32	4			1		
BIST	BIST #xx:3,Rd	1					
	BIST #xx:3,@ERd	2			2		
	BIST #xx:3,@aa:8	2			2		
	BIST #xx:3,@aa:16	3			2		
	BIST #xx:3,@aa:32	4			2		
BIXOR	BIXOR #xx:3,Rd	1					
	BIXOR #xx:3,@ERd	2			1		
	BIXOR #xx:3,@aa:8	2			1		
	BIXOR #xx:3,@aa:16	3			1		
	BIXOR #xx:3,@aa:32	4			1		
BLD	BLD #xx:3,Rd	1					
	BLD #xx:3,@ERd	2			1		
	BLD #xx:3,@aa:8	2			1		
	BLD #xx:3,@aa:16	3			1		
	BLD #xx:3,@aa:32	4			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BNOT	BNOT #xx:3,Rd	1					
	BNOT #xx:3,@ERd	2			2		
	BNOT #xx:3,@aa:8	2			2		
	BNOT #xx:3,@aa:16	3			2		
	BNOT #xx:3,@aa:32	4			2		
	BNOT Rn,Rd	1					
	BNOT Rn,@ERd	2			2		
	BNOT Rn,@aa:8	2			2		
	BNOT Rn,@aa:16	3			2		
	BNOT Rn,@aa:32	4			2		
BOR	BOR #xx:3,Rd	1					
	BOR #xx:3,@ERd	2			1		
	BOR #xx:3,@aa:8	2			1		
	BOR #xx:3,@aa:16	3			1		
	BOR #xx:3,@aa:32	4			1		
BSET	BSET #xx:3,Rd	1					
	BSET #xx:3,@ERd	2			2		
	BSET #xx:3,@aa:8	2			2		
	BSET #xx:3,@aa:16	3			2		
	BSET #xx:3,@aa:32	4			2		
	BSET Rn,Rd	1					
	BSET Rn,@ERd	2			2		
	BSET Rn,@aa:8	2			2		
	BSET Rn,@aa:16	3			2		
	BSET Rn,@aa:32	4			2		
BSR	BSR d:8	2		2			
	BSR d:16	2		2			1
BST	BST #xx:3,Rd	1					
	BST #xx:3,@ERd	2			2		
	BST #xx:3,@aa:8	2			2		
	BST #xx:3,@aa:16	3			2		
	BST #xx:3,@aa:32	4			2		

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BTST	BTST #xx:3,Rd	1					
	BTST #xx:3,@ERd	2			1		
	BTST #xx:3,@aa:8	2			1		
	BTST #xx:3,@aa:16	3			1		
	BTST #xx:3,@aa:32	4			1		
	BTST Rn,Rd	1					
	BTST Rn,@ERd	2			1		
	BTST Rn,@aa:8	2			1		
	BTST Rn,@aa:16	3			1		
	BTST Rn,@aa:32	4			1		
BXOR	BXOR #xx:3,Rd	1					
	BXOR #xx:3,@ERd	2			1		
	BXOR #xx:3,@aa:8	2			1		
	BXOR #xx:3,@aa:16	3			1		
	BXOR #xx:3,@aa:32	4			1		
CLRMAC	CLRMAC	Cannot be used in the H8S/2214					
CMP	CMP.B #xx:8,Rd	1					
	CMP.B Rs,Rd	1					
	CMP.W #xx:16,Rd	2					
	CMP.W Rs,Rd	1					
	CMP.L #xx:32,ERd	3					
	CMP.L ERs,ERd	1					
DAA	DAA Rd	1					
DAS	DAS Rd	1					
DEC	DEC.B Rd	1					
	DEC.W #1/2,Rd	1					
	DEC.L #1/2,ERd	1					
DIVXS	DIVXS.B Rs,Rd	2					11
	DIVXS.W Rs,ERd	2					19
DIVXU	DIVXU.B Rs,Rd	1					11
	DIVXU.W Rs,ERd	1					19

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
EEPMOV	EEPMOV.B	2			2n+2*2		
	EEPMOV.W	2			2n+2*2		
EXTS	EXTS.W Rd	1					
	EXTS.L ERd	1					
EXTU	EXTU.W Rd	1					
	EXTU.L ERd	1					
INC	INC.B Rd	1					
	INC.W #1/2,Rd	1					
	INC.L #1/2,ERd	1					
JMP	JMP @ERn	2					
	JMP @aa:24	2					1
	JMP @@aa:8	2	2				1
JSR	JSR @ERn	2		2			
	JSR @aa:24	2		2			1
	JSR @@aa:8	2	2	2			
LDC	LDC #xx:8,CCR	1					
	LDC #xx:8,EXR	2					
	LDC Rs,CCR	1					
	LDC Rs,EXR	1					
	LDC @ERs,CCR	2				1	
	LDC @ERs,EXR	2				1	
	LDC @(d:16,ERs),CCR	3				1	
	LDC @(d:16,ERs),EXR	3				1	
	LDC @(d:32,ERs),CCR	5				1	
	LDC @(d:32,ERs),EXR	5				1	
	LDC @ERs+,CCR	2				1	1
	LDC @ERs+,EXR	2				1	1
	LDC @aa:16,CCR	3				1	
	LDC @aa:16,EXR	3				1	
	LDC @aa:32,CCR	4				1	
	LDC @aa:32,EXR	4				1	

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
LDM	LDM.L @SP+, (ERn-ERn+1)	2		4			1
	LDM.L @SP+, (ERn-ERn+2)	2		6			1
	LDM.L @SP+, (ERn-ERn+3)	2		8			1
LDMAC	LDMAC ERs,MACH	Cannot be used in the H8S/2214					
	LDMAC ERs,MACL						
MAC	MAC @ERn+,@ERm+	Cannot be used in the H8S/2214					
MOV	MOV.B #xx:8,Rd	1					
	MOV.B Rs,Rd	1					
	MOV.B @ERs,Rd	1			1		
	MOV.B @(d:16,ERs),Rd	2			1		
	MOV.B @(d:32,ERs),Rd	4			1		
	MOV.B @ERs+,Rd	1			1		1
	MOV.B @aa:8,Rd	1			1		
	MOV.B @aa:16,Rd	2			1		
	MOV.B @aa:32,Rd	3			1		
	MOV.B Rs,@ERd	1			1		
	MOV.B Rs,@(d:16,ERd)	2			1		
	MOV.B Rs,@(d:32,ERd)	4			1		
	MOV.B Rs,@-ERd	1			1		1
	MOV.B Rs,@aa:8	1			1		
	MOV.B Rs,@aa:16	2			1		
	MOV.B Rs,@aa:32	3			1		
	MOV.W #xx:16,Rd	2					
	MOV.W Rs,Rd	1					
	MOV.W @ERs,Rd	1				1	
	MOV.W @(d:16,ERs),Rd	2				1	
	MOV.W @(d:32,ERs),Rd	4				1	
	MOV.W @ERs+,Rd	1				1	1
	MOV.W @aa:16,Rd	2				1	
	MOV.W @aa:32,Rd	3				1	
	MOV.W Rs,@ERd	1				1	

Instruction	Mnemonic	Instruction Fetch	Branch Address Read	Stack Operation	Byte Data Access	Word Data Access	Internal Operation
		I	J	K	L	M	N
MOV	MOV.W Rs,@(d:16,ERd)	2				1	
	MOV.W Rs,@(d:32,ERd)	4				1	
	MOV.W Rs,@-ERd	1				1	1
	MOV.W Rs,@aa:16	2				1	
	MOV.W Rs,@aa:32	3				1	
	MOV.L #xx:32,ERd	3					
	MOV.L ERs,ERd	1					
	MOV.L @ERs,ERd	2				2	
	MOV.L @(d:16,ERs),ERd	3				2	
	MOV.L @(d:32,ERs),ERd	5				2	
	MOV.L @ERs+,ERd	2				2	1
	MOV.L @aa:16,ERd	3				2	
	MOV.L @aa:32,ERd	4				2	
	MOV.L ERs,@ERd	2				2	
	MOV.L ERs,@(d:16,ERd)	3				2	
	MOV.L ERs,@(d:32,ERd)	5				2	
	MOV.L ERs,@-ERd	2				2	1
	MOV.L ERs,@aa:16	3				2	
	MOV.L ERs,@aa:32	4				2	
MOVFPE	MOVFPE @:aa:16,Rd	Can not be used in the H8S/2214					
MOVTPE	MOVTPE Rs,@:aa:16						
MULXS	MULXS.B Rs,Rd	2					11
	MULXS.W Rs,ERd	2					19
MULXU	MULXU.B Rs,Rd	1					11
	MULXU.W Rs,ERd	1					19
NEG	NEG.B Rd	1					
	NEG.W Rd	1					
	NEG.L ERd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
	NOT.W Rd	1					
	NOT.L ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
OR	OR.B #xx:8,Rd	1					
	OR.B Rs,Rd	1					
	OR.W #xx:16,Rd	2					
	OR.W Rs,Rd	1					
	OR.L #xx:32,ERd	3					
	OR.L ERs,ERd	2					
ORC	ORC #xx:8,CCR	1					
	ORC #xx:8,EXR	2					
POP	POP.W Rn	1				1	1
	POP.L ERn	2				2	1
PUSH	PUSH.W Rn	1				1	1
	PUSH.L ERn	2				2	1
ROTL	ROTL.B Rd	1					
	ROTL.B #2,Rd	1					
	ROTL.W Rd	1					
	ROTL.W #2,Rd	1					
	ROTL.L ERd	1					
	ROTL.L #2,ERd	1					
ROTR	ROTR.B Rd	1					
	ROTR.B #2,Rd	1					
	ROTR.W Rd	1					
	ROTR.W #2,Rd	1					
	ROTR.L ERd	1					
	ROTR.L #2,ERd	1					
ROTXL	ROTXL.B Rd	1					
	ROTXL.B #2,Rd	1					
	ROTXL.W Rd	1					
	ROTXL.W #2,Rd	1					
	ROTXL.L ERd	1					
	ROTXL.L #2,ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ROTXR	ROTXR.B Rd	1					
	ROTXR.B #2,Rd	1					
	ROTXR.W Rd	1					
	ROTXR.W #2,Rd	1					
	ROTXR.L ERd	1					
	ROTXR.L #2,ERd	1					
RTE	RTE	2		2/3*[1]			1
RTS	RTS	2		2			1
SHAL	SHAL.B Rd	1					
	SHAL.B #2,Rd	1					
	SHAL.W Rd	1					
	SHAL.W #2,Rd	1					
	SHAL.L ERd	1					
	SHAL.L #2,ERd	1					
SHAR	SHAR.B Rd	1					
	SHAR.B #2,Rd	1					
	SHAR.W Rd	1					
	SHAR.W #2,Rd	1					
	SHAR.L ERd	1					
	SHAR.L #2,ERd	1					
SHLL	SHLL.B Rd	1					
	SHLL.B #2,Rd	1					
	SHLL.W Rd	1					
	SHLL.W #2,Rd	1					
	SHLL.L ERd	1					
	SHLL.L #2,ERd	1					
SHLR	SHLR.B Rd	1					
	SHLR.B #2,Rd	1					
	SHLR.W Rd	1					
	SHLR.W #2,Rd	1					
	SHLR.L ERd	1					
	SHLR.L #2,ERd	1					
SLEEP	SLEEP	1					1

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
STC	STC.B CCR,Rd	1					
	STC.B EXR,Rd	1					
	STC.W CCR,@ERd	2				1	
	STC.W EXR,@ERd	2				1	
	STC.W CCR,@(d:16,ERd)	3				1	
	STC.W EXR,@(d:16,ERd)	3				1	
	STC.W CCR,@(d:32,ERd)	5				1	
	STC.W EXR,@(d:32,ERd)	5				1	
	STC.W CCR,@-ERd	2				1	1
	STC.W EXR,@-ERd	2				1	1
	STC.W CCR,@aa:16	3				1	
	STC.W EXR,@aa:16	3				1	
	STC.W CCR,@aa:32	4				1	
	STC.W EXR,@aa:32	4				1	
STM	STM.L (ERn-ERn+1), @-SP	2		4			1
	STM.L (ERn-ERn+2), @-SP	2		6			1
	STM.L (ERn-ERn+3), @-SP	2		8			1
STMAC	STMAC MACH,ERd	Cannot be used in the H8S/2214					
	STMAC MACL,ERd						
SUB	SUB.B Rs,Rd	1					
	SUB.W #xx:16,Rd	2					
	SUB.W Rs,Rd	1					
	SUB.L #xx:32,ERd	3					
	SUB.L ERs,ERd	1					
SUBS	SUBS #1/2/4,ERd	1					
SUBX	SUBX #xx:8,Rd	1					
	SUBX Rs,Rd	1					
TAS	TAS @ERd*3	2			2		
TRAPA	TRAPA #x:2	2	2	2/3*1			2

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
XOR	XOR.B #xx:8,Rd	1					
	XOR.B Rs,Rd	1					
	XOR.W #xx:16,Rd	2					
	XOR.W Rs,Rd	1					
	XOR.L #xx:32,ERd	3					
	XOR.L ERs,ERd	2					
XORC	XORC #xx:8,CCR	1					
	XORC #xx:8,EXR	2					

Notes: *1 2 when EXR is invalid, 3 when EXR is valid.

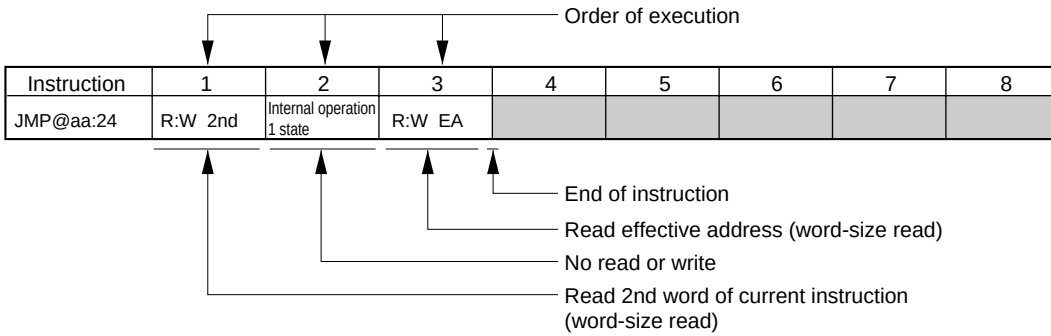
*2 When n bytes of data are transferred.

*3 This instruction should be used with the ER0, ER1, ER4, or ER5 general register only.

A.5 Bus States during Instruction Execution

Table A-16 indicates the types of cycles that occur during instruction execution by the CPU. See table A-14 for the number of states per cycle.

How to Read the Table:



Legend

R:B	Byte-size read
R:W	Word-size read
W:B	Byte-size write
W:W	Word-size write
:M	Transfer of the bus is not performed immediately after this cycle
2nd	Address of 2nd word (3rd and 4th bytes)
3rd	Address of 3rd word (5th and 6th bytes)
4th	Address of 4th word (7th and 8th bytes)
5th	Address of 5th word (9th and 10th bytes)
NEXT	Address of next instruction
EA	Effective address
VEC	Vector address

Figure A-1 shows timing waveforms for the address bus and the $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ signals during execution of the above instruction with an 8-bit bus, using three-state access with no wait states.

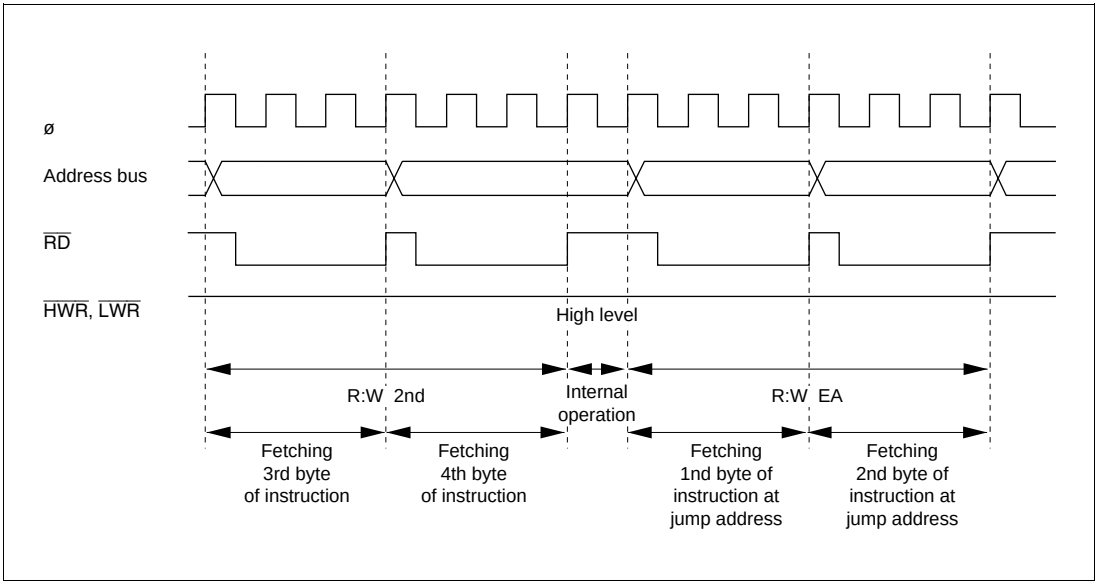


Figure A-1 Address Bus, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ Timing (8-Bit Bus, Three-State Access, No Wait States)

Table A-16 Instruction Execution Cycles

Instruction	1	2	3	4	5	6	7	8	9
ADD.B #xx:8,Rd	R:W NEXT								
ADD.B Rs,Rd	R:W NEXT								
ADD.W #xx:16,Rd	R:W 2nd	R:W NEXT							
ADD.W Rs,Rd	R:W NEXT								
ADD.L #xx:32,ERd	R:W 2nd	R:W 3rd	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	R:W NEXT							
AND.W Rs,Rd	R:W NEXT								
AND.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
AND.L ERs,ERd	R:W 2nd	R:W NEXT							
ANDC #xx:8,CCR	R:W NEXT								
ANDC #xx:8,EXR	R:W 2nd	R:W NEXT							
BAND #xx:3,Rd	R:W NEXT								
BAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M 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							

Instruction	1	2	3	4	5	6	7	8	9
BLE d:8	R:W NEXT	R:W EA							
BRA d:16 (BT d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BRN d:16 (BF d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BHI d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BLS d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BCC d:16 (BHS d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BCS d:16 (BLO d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BNE d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BEQ d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BVC d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BVS d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BPL d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BMI d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BGE d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BLT d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BGT d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BLE d:16	R:W 2nd	Internal operation, 1 state	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				

Instruction	1	2	3	4	5	6	7	8	9
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:M NEXT						
BIAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BILD #xx:3,Rd	R:W NEXT								
BILD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BILD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BILD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BILD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BIOR #xx:3,Rd	R:W NEXT								
BIOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BIOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M 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:M NEXT						
BIXOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIXOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIXOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BLD #xx:3,Rd	R:W NEXT								
BLD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BLD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BLD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BLD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BNOT #xx:3,Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
BNOT #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BNOT #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT Rn,Rd	R:W NEXT								
BNOT Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BNOT Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BOR #xx:3,Rd	R:W NEXT								
BOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M 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	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)					
BSR d:16	R:W 2nd	Internal operation, 1 state	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					
BST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
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:M NEXT						

Instruction	1	2	3	4	5	6	7	8	9
BTST #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BTST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BTST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BTST Rn,Rd	R:W NEXT								
BTST Rn,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BTST Rn,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BTST Rn,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BTST Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BXOR #xx:3,Rd	R:W NEXT								
BXOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BXOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BXOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BXOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
CLRMAC	Cannot be used in the H8S/2214								
CMP.B #xx:8,Rd	R:W NEXT								
CMP.B Rs,Rd	R:W NEXT								
CMP.W #xx:16,Rd	R:W 2nd	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	R:W NEXT	Internal operation, 11 states						
DIVXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 19 states						
DIVXU.B Rs,Rd	R:W NEXT	Internal operation, 11 states							
DIVXU.W Rs,ERd	R:W NEXT	Internal operation, 19 states							
EEPMOV.B	R:W 2nd	R:B EAs*1	R:B EAd*1	R:B EAs*2	W:B EAd*2	R:W NEXT			
EEPMOV.W	R:W 2nd	R:B EAs*1	R:B EAd*1	R:B EAs*2	W:B EAd*2	R:W NEXT			
EXTS.W Rd	R:W NEXT			← Repeated n times*2 →					
EXTS.L ERd	R:W NEXT								
EXTU.W Rd	R:W NEXT								
EXTU.L ERd	R:W NEXT								
INC.B Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
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	Internal operation, 1 state	R:W EA						
JMP @@aa:8	R:W NEXT	R:W:M aa:8	R:W aa:8	Internal operation, 1 state	R:W EA				
JSR @ERn	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)					
JSR @aa:24	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)				
JSR @@aa:8	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								
LDC @ERs,CCR	R:W 2nd	R:W NEXT	R:W EA						
LDC @ERs,EXR	R:W 2nd	R:W NEXT	R:W EA						
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	Internal operation, 1 state	R:W EA					
LDC @ERs+,EXR	R:W 2nd	R:W NEXT	Internal operation, 1 state	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+	R:W 2nd	R:W:M NEXT	Internal operation	R:W:M stack (H)*3	R:W stack (L)*3				

Instruction	1	2	3	4	5	6	7	8	9
LDMAC ERs,MACL MAC @ERn+,@ERm+	Cannot be used in the H8S/2214								
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:B EA						
MOV.B @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:B EA				
MOV.B @ERs+,Rd	R:W NEXT	Internal operation, 1 state	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:B EA						
MOV.B @aa:32,Rd	R:W 2nd	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	W:B EA						
MOV.B Rs,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:B EA				
MOV.B Rs,@–ERd	R:W NEXT	Internal operation, 1 state	W:B EA						
MOV.B Rs,@aa:8	R:W NEXT	W:B EA							
MOV.B Rs,@aa:16	R:W 2nd	R:W NEXT	W:B EA						
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	Internal operation, 1 state	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:E 4th	R:W NEXT	W:W EA				
MOV.W Rs,@–ERd	R:W NEXT	Internal operation, 1 state	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					

Instruction	1	2	3	4	5	6	7	8	9
MOV.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
MOV.L ERs,ERd	R:W NEXT								
MOV.L @ERs,ERd	R:W 2nd	R:W:M NEXT	R:W:M EA	R:W EA+2					
MOV.L @(d:16,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2				
MOV.L @(d:32,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	R:W:M EA	R:W EA+2		
MOV.L @ERs+,ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2				
MOV.L @aa:16,ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2				
MOV.L @aa:32,ERd	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	R:W:M EA	R:W EA+2			
MOV.L ERs,@ERd	R:W 2nd	R:W:M NEXT	W:W:M EA	W:W EA+2					
MOV.L ERs,@(d:16,ERd)	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2				
MOV.L ERs,@(d:32,ERd)	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	W:W:M EA	W:W EA+2		
MOV.L ERs,@–ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M EA	W:W EA+2				
MOV.L ERs,@aa:16	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2				
MOV.L ERs,@aa:32	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	W:W:M EA	W:W EA+2			
MOVFPE @aa:16,Rd	Cannot be used in the H8S/2214								
MOVTPE Rs,@aa:16									
MULXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 11 states						
MULXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 19 states						
MULXU.B Rs,Rd	R:W NEXT	Internal operation, 11 states							
MULXU.W Rs,ERd	R:W NEXT	Internal operation, 19 states							
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								

Instruction	1	2	3	4	5	6	7	8	9
POP.W Rn	R:W NEXT	Internal operation, 1 state	R:W EA						
POP.L ERn	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2				
PUSH.W Rn	R:W NEXT	Internal operation, 1 state	W:W EA						
PUSH.L ERn	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	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								
ROTXL.W #2,Rd	R:W 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	R:W*4			
RTS	R:W NEXT	R:W:M stack (H)	R:W stack (L)	Internal operation, 1 state	R:W*4				
SHAL.B Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
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								
SHAR.B #2,Rd	R:W 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 CCR,Rd	R:W NEXT								
STC EXR,Rd	R:W NEXT								
STC CCR,@ERd	R:W 2nd	R:W NEXT	W:W EA						
STC EXR,@ERd	R:W 2nd	R:W NEXT	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	Internal operation, 1 state	W:W EA					

Instruction	1	2	3	4	5	6	7	8	9
STC EXR,@–ERd	R:W 2nd	R:W NEXT	Internal operation, 1 state	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	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STM.L(ERn–ERn+2),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STM.L(ERn–ERn+3),@–SP	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STMAC MACH,ERd	Cannot be used in the H8S/2214								
STMAC MACL,ERd									
SUB.B Rs,Rd	R:W NEXT								
SUB.W #xx:16,Rd	R:W 2nd	R:W NEXT							
SUB.W Rs,Rd	R:W NEXT								
SUB.L #xx:32,ERd	R:W 2nd	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 NEXT								
TAS @ERd*5	R:W 2nd	R:W NEXT	R:B:M EA	W:B EA					
TRAPA #x:2	R:W NEXT	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W*8
XOR.B #xx8,Rd	R:W NEXT								
XOR.B Rs,Rd	R:W NEXT								
XOR.W #xx:16,Rd	R:W 2nd	R:W NEXT							
XOR.W Rs,Rd	R:W NEXT								
XOR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
XOR.L ERs,ERd	R:W 2nd	R:W NEXT							
XORC #xx:8,CCR	R:W NEXT								
XORC #xx:8,EXR	R:W 2nd	R:W NEXT							
Reset exception handling	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W*6					

Instruction	1	2	3	4	5	6	7	8	9
Interrupt exception handling	R:W*7	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W*8

Notes: *1 EAs is the contents of ER5. EAd is the contents of ER6.
*2 EAs is the contents of ER5. EAd is the contents of ER6. Both registers are incremented by 1 after execution of the instruction. n is the initial value of R4L or R4. If n = 0, these bus cycles are not executed.
*3 Repeated two times to save or restore two registers, three times for three registers, or four times for four registers.
*4 Start address after return.
*5 This instruction should be used with the ER0, ER1, ER4, or ER5 general register only.
*6 Start address of the program.
*7 Prefetch address, equal to two plus the PC value pushed onto the stack. In recovery from sleep mode or software standby mode the read operation is replaced by an internal operation.
*8 Start address of the interrupt-handling routine.

A.6 Condition Code Modification

This section indicates the effect of each CPU instruction on the condition code. The notation used in the table is defined below.

m = { 31 for longword operands; 15 for word operands; 7 for byte operands }

Si	The i-th bit of the source operand
Di	The i-th bit of the destination operand
Ri	The i-th bit of the result
Dn	The specified bit in the destination operand
—	Not affected
↕	Modified according to the result of the instruction (see definition)
0	Always cleared to 0
1	Always set to 1
*	Undetermined (no guaranteed value)
Z'	Z flag before instruction execution
C'	C flag before instruction execution

Table A-17 Condition Code Modification

Instruction	H	N	Z	V	C	Definition
ADD	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{Rm–4}$ + Sm–4 · $\overline{Rm–4}$
						N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = Sm · Dm · $\overline{Rm}$ + $\overline{Sm}$ · $\overline{Dm}$ · Rm
						C = Sm · Dm + Dm · $\overline{Rm}$ + Sm · $\overline{Rm}$
ADDS	—	—	—	—	—	
ADDX	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{Rm–4}$ + Sm–4 · $\overline{Rm–4}$
						N = Rm
						Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$
						V = Sm · Dm · $\overline{Rm}$ + $\overline{Sm}$ · $\overline{Dm}$ · Rm
						C = Sm · Dm + Dm · $\overline{Rm}$ + Sm · $\overline{Rm}$
AND	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
ANDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.
BAND	—	—	—	—	↕	C = C' · Dn
Bcc	—	—	—	—	—	
BCLR	—	—	—	—	—	
BIAND	—	—	—	—	↕	C = C' · $\overline{Dn}$
BILD	—	—	—	—	↕	C = $\overline{Dn}$
BIOR	—	—	—	—	↕	C = C' + $\overline{Dn}$
BIST	—	—	—	—	—	
BIXOR	—	—	—	—	↕	C = C' · Dn + $\overline{C'}$ · $\overline{Dn}$
BLD	—	—	—	—	↕	C = Dn
BNOT	—	—	—	—	—	
BOR	—	—	—	—	↕	C = C' + Dn
BSET	—	—	—	—	—	
BSR	—	—	—	—	—	
BST	—	—	—	—	—	
BTST	—	—	↕	—	—	Z = $\overline{Dn}$
BXOR	—	—	—	—	↕	C = C' · $\overline{Dn}$ + $\overline{C'}$ · Dn
CLRMAC						Cannot be used in the H8S/2214

Instruction	H	N	Z	V	C	Definition
CMP	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4
						N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm
						C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
DAA	*	↕	↕	*	↕	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						C: decimal arithmetic carry
DAS	*	↕	↕	*	↕	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						C: decimal arithmetic borrow
DEC	—	↕	↕	↕	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = Dm · $\overline{Rm}$
DIVXS	—	↕	↕	—	—	N = Sm · $\overline{Dm}$ + $\overline{Sm}$ · Dm
						Z = $\overline{Sm}$ · $\overline{Sm–1}$ · ····· · $\overline{S0}$
DIVXU	—	↕	↕	—	—	N = Sm
						Z = $\overline{Sm}$ · $\overline{Sm–1}$ · ····· · $\overline{S0}$
EEPMOV	—	—	—	—	—	
EXTS	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
EXTU	—	0	↕	0	—	Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
INC	—	↕	↕	↕	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = $\overline{Dm}$ · Rm
JMP	—	—	—	—	—	
JSR	—	—	—	—	—	
LDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.
LDM	—	—	—	—	—	
LDMAC						Cannnot be used in the H8S/2214
MAC						

Instruction	H	N	Z	V	C	Definition
MOV	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
MOVFPE						Can not be used in the H8S/2214
MOVTPE						
MULXS	—	↕	↕	—	—	N = R2m Z = $\overline{R2m} \cdot \overline{R2m-1} \cdot \cdots\cdots \cdot \overline{R0}$
MULXU	—	—	—	—	—	
NEG	↕	↕	↕	↕	↕	H = Dm–4 + Rm–4 N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = Dm · Rm C = Dm + Rm
NOP	—	—	—	—	—	
NOT	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
OR	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
ORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
POP	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
PUSH	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
ROTL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
ROTR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)

Instruction	H	N	Z	V	C	Definition
ROTXL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
ROTXR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
RTE	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
RTS	—	—	—	—	—	
SHAL	—	↕	↕	↕	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Dm \cdot Dm-1 + \overline{Dm} \cdot \overline{Dm-1}}$ (1-bit shift) V = $\overline{Dm \cdot Dm-1 \cdot Dm-2 \cdot \overline{Dm} \cdot \overline{Dm-1} \cdot \overline{Dm-2}}$ (2-bit shift) C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
SHAR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
SHLL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
SHLR	—	0	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
SLEEP	—	—	—	—	—	
STC	—	—	—	—	—	
STM	—	—	—	—	—	
STMAC						Cannot be used in the H8S/2214

Instruction	H	N	Z	V	C	Definition
SUB	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4
						N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm
						C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
SUBS	—	—	—	—	—	
SUBX	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4
						N = Rm
						Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$
						V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm
						C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
TAS*	—	↕	↕	0	—	N = Dm
						Z = $\overline{Dm}$ · $\overline{Dm–1}$ · ····· · $\overline{D0}$
TRAPA	—	—	—	—	—	
XOR	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
XORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.

Note: * This instruction should be used with the ER0, ER1, ER4, or ER5 general register only.

Appendix B Internal I/O Register

B.1 Addresses

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'EBC0 to H'EFBF	MRA	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz	DTC	16/32* bit
	MRB	CHNE	DISEL	—	—	—	—	—	—		
	SAR										
	DAR										
	CRA										
	CRB										
H'FDAC	DADR0									D/A converter	8 bit
H'FDAE	DACR	—	DAOE0	—	—	—	—	—	—		
H'FDE4	SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—	Power-down state	8 bit
H'FDE5	SYSCR	—	—	INTM1	INTM0	NMIEG	MRESE	—	RAME	MCU	8 bit
H'FDE6	SCKCR	PSTOP	—	—	—	—	SCK2	SCK1	SCK0	Clock pulse generator, power-down state	8 bit
H'FDE7	MDCR	—	—	—	—	—	MDS2	MDS1	MSD0	MCU, ROM	8 bit
H'FDE8	MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0	Power-down state	8 bit
H'FDE9	MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0		
H'FDEA	MSTPCRC	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0		

Note: * Located in on-chip RAM. The bus width is 32 bits when the DTC accesses this area as register information, and 16 bits otherwise.

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FDEB	PFCR	—	—	—	—	AE3	AE2	AE1	AE0	Bus controller	8 bit
H'FDEC	LPWRCR	—	—	—	—	RFCUT	—	STC1	STC0	Clock pulse generator	8 bit
H'FDF8	SEMR0	SSE	—	—	—	ABCS	ACS2	ACS1	ACS0	SCI0	8 bit
H'FE12	ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	Interrupt controller	8 bit
H'FE13	ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA		
H'FE14	IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E		
H'FE15	ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F		
H'FE16 to H'FE1E	DTCER	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	DTC	8 bit
H'FE1F	DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0		
H'FE30	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	Port	8 bit
H'FE32	P3DDR	—	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR		
H'FE36	P7DDR	P77DDR	P76DDR	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR		
H'FE39	PADDR	—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR		
H'FE3A	PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR		
H'FE3B	PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR		
H'FE3C	PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR		
H'FE3D	PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR		
H'FE3E	PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR		
H'FE3F	PGDDR	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR		
H'FE40	PAPCR	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR		
H'FE41	PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR		
H'FE42	PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR		
H'FE43	PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR		
H'FE44	PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR		
H'FE46	P3ODR	—	P36ODR	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR		
H'FE47	PAODR	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR		
H'FE4A	IPINSEL0	P36 IRQ7E	P47 IRQ6E	P46 IRQ5E	P44 IRQ4E	P43 IRQ3E	P42 IRQ2E	P41 IRQ1E	P40 IRQ0E		
H'FE4E	OPINSEL	—	P76 STPOE	P75 MSOE	P74 DTCOE	—	—	—	—		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FEB0	TSTR	—	—	—	—	—	CST2	CST1	CST0	TPU	8 bit
H'FEB1	TSYR	—	—	—	—	—	SYNC2	SYNC1	SYNC0		
H'FEC0	IPRA	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	Interrupt controller	8 bit
H'FEC1	IPRB	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC2	IPRC	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC3	IPRD	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC5	IPRF	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC6	IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC9	IPRJ	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECA	IPRK	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECC	IPRM	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FED0	ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	Bus controller	8 bit
H'FED1	ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0		
H'FED2	WCRH	W71	W70	W61	W60	W51	W50	W41	W40		
H'FED3	WCRL	W31	W30	W21	W20	W11	W10	W01	W00		
H'FED4	BCRH	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—		
H'FED5	BCRL	BRLE	—	—	—	—	—	—	WAITE		
H'FEDB	RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0	Flash	
H'FEE0	MAR0AH	—	—	—	—	—	—	—	—	DMAC	16 bit
H'FEE1											
H'FEE2	MAR0AL										
H'FEE3											
H'FEE4	IOAR0A										
H'FEE5											
H'FEE6	ETCR0A										
H'FEE7											
H'FEE8	MAR0BH	—	—	—	—	—	—	—	—		
H'FEE9											
H'FEEA	MAR0BL										
H'FEEB											
H'FEEC	IOAR0B										
H'FEED											
H'FEEE	ETCR0B										
H'FEEF											
H'FEF0	MAR1AH	—	—	—	—	—	—	—	—		
H'FEF1											
H'FEF2	MAR1AL										
H'FEF3											

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FEF4	IOAR1A									DMAC	16 bit
H'FEF5											
H'FEF6	ETCR1A										
H'FEF7											
H'FEF8	MAR1BH	—	—	—	—	—	—	—	—		
H'FEF9											
H'FEFA	MAR1BL										
H'FEFB											
H'FEFC	IOAR1B										
H'FEFD											
H'FEFE	ETCR1B										
H'FEFF											
H'FF00	P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	Port	8 bit
H'FF02	P3DR	—	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR		
H'FF06	P7DR	P77DR	P76DR	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR		
H'FF09	PADR	—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR		
H'FF0A	PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR		
H'FF0B	PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR		
H'FF0C	PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR		
H'FF0D	PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR		
H'FF0E	PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR		
H'FF0F	PGDR	—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR		
H'FF10	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU0	8 bit
H'FF11	TMDR0	—	—	BFB	BFA	MD3	MD2	MD1	MD0		
H'FF12	TIOR0H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FF13	TIOR0L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0		
H'FF14	TIER0	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
H'FF15	TSR0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA		
H'FF16	TCNT0										16 bit
H'FF17											
H'FF18	TGR0A										
H'FF19											
H'FF1A	TGR0B										
H'FF1B											
H'FF1C	TGR0C										
H'FF1D											
H'FF1E	TGR0D										
H'FF1F											

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FF20	TCR1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU1	8 bit
H'FF21	TMDR1	—	—	—	—	MD3	MD2	MD1	MD0		
H'FF22	TIOR1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FF24	TIER1	—	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FF25	TSR1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FF26	TCNT1										16 bit
H'FF27											
H'FF28	TGR1A										
H'FF29											
H'FF2A	TGR1B										
H'FF2B											
H'FF30	TCR2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU2	8 bit
H'FF31	TMDR2	—	—	—	—	MD3	MD2	MD1	MD0		
H'FF32	TIOR2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FF34	TIER2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FF35	TSR2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FF36	TCNT2										16 bit
H'FF37											
H'FF38	TGR2A										
H'FF39											
H'FF3A	TGR2B										
H'FF3B											
H'FF60	DMAWER	—	—	—	—	WE1B	WE1A	WE0B	WE0A	DMAC	8
H'FF61	DMATCR	—	—	TEE1	TEE0	—	—	—	—		
H'FF62	DMACR0A	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0		16
H'FF63	DMACR0B	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0		
H'FF64	DMACR1A	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0		
H'FF65	DMACR1B	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0		
H'FF66	DMABCRH	FAE1	FAE0	—	—	DTA1B	DTA1A	DTA0B	DTA0A		
H'FF67	DMABCRL	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A		
H'FF74	TCSR0	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	Watchdog timer 0	16 bit
H'FF74 (write)	TCNT0										
H'FF75 (read)	TCNT0										
H'FF76 (write)	RSTCSR0	WOVF	RSTE	RSTS	—	—	—	—	—		
H'FF77 (read)	RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FF78	SMR0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI0	8 bit
H'FF79	BRR0										
H'FF7A	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF7B	TDR0										
H'FF7C	SSR0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
H'FF7D	RDR0										
H'FF7E	SCMR0	—	—	—	—	SDIR	SINV	—	—		
H'FF80	SMR1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI1	8 bit
H'FF81	BRR1										
H'FF82	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF83	TDR1										
H'FF84	SSR1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
H'FF85	RDR1										
H'FF86	SCMR1	—	—	—	—	SDIR	SINV	—	—		
H'FF88	SMR2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI2	8 bit
H'FF89	BRR2										
H'FF8A	SCR2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF8B	TDR2										
H'FF8C	SSR2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
H'FF8D	RDR2										
H'FF8E	SCMR2	—	—	—	—	SDIR	SINV	—	—		
H'FFB0	PORT1	P17	P16	P15	P14	P13	P12	P11	P10	Port	8 bit
H'FFB2	PORT3	—	P36	P35	P34	P33	P32	P31	P30		
H'FFB3	PORT4	P47	P46	P45	P44	P43	P42	P41	P40		
H'FFB6	PORT7	P77	P76	P75	P74	P73	P72	P71	P70		
H'FFB8	PORT9	—	P96	—	—	—	—	—	—		
H'FFB9	PORTA	—	—	—	—	PA3	PA2	PA1	PA0		
H'FFBA	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0		
H'FFBB	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0		
H'FFBC	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0		
H'FFBD	PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0		
H'FFBE	PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0		
H'FFBF	PORTG	—	—	—	PG4	PG3	PG2	PG1	PG0		

B.2 Functions

MRA—DTC Mode Register A **H'EBC0 to H'EFBF** **DTC**

Bit	7	6	5	4	3	2	1	0
	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz
Initial value	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W	—	—	—	—	—	—	—	—

DTC Data Transfer Size

0	Byte-size transfer
1	Word-size transfer

DTC Transfer Mode Select

0	Destination side is repeat area or block area
1	Source side is repeat area or block area

DTC Mode

0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

Destination Address Mode 1 and 0

0	—	DAR is fixed
1	0	DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Source Address Mode 1 and 0

0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bit	7	6	5	4	3	2	1	0
	CHNE	DISEL	—	—	—	—	—	—
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	—	—	—	—	—	—	—	—

Reserved
Only 0 should be written to these bits

DTC Interrupt Select

0	After a data transfer ends, the CPU interrupt is disabled unless the transfer counter is 0 (the DTC clears the interrupt source flag of the activating interrupt to 0)
1	After a data transfer ends, the CPU interrupt is enabled (the DTC does not clear the interrupt source flag of the activating interrupt to 0)

DTC Chain Transfer Enable

0	End of DTC data transfer (activation waiting state)
1	DTC chain transfer (new register information is read, then data is transferred)

SAR—DTC Source Address Register H'EBC0 to H'EFBF DTC

Bit	23	22	21	20	19	- - -	4	3	2	1	0
						- - -					
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	- - -	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	—	—	—	—	—	- - -	—	—	—	—	—

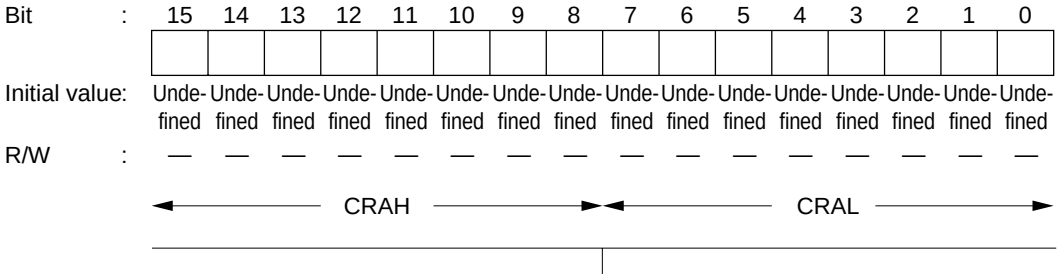
Specifies transfer data source address

DAR—DTC Destination Address Register H'EBC0 to H'EFBF DTC

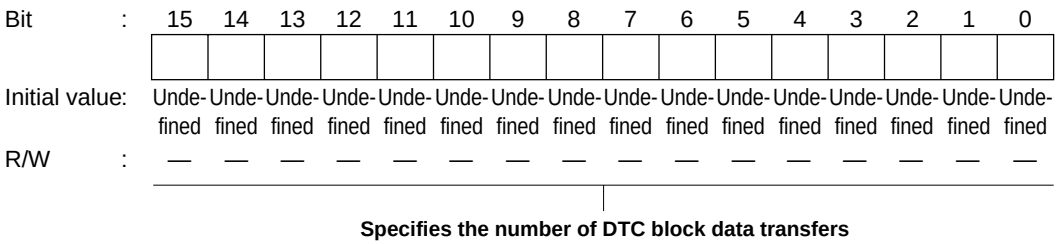
Bit	23	22	21	20	19	- - -	4	3	2	1	0
						- - -					
Initial value :	Undefined	Undefined	Undefined	Undefined	Undefined	- - -	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	—	—	—	—	—	- - -	—	—	—	—	—

Specifies transfer data destination address

CRA—DTC Transfer Count Register A **H'EBC0 to H'EFBF** **DTC**



CRB—DTC Transfer Count Register B **H'EBC0 to H'EFBF** **DTC**



DADR0—D/A Data Register 0 **H'FDAC** **D/A**

Bit :	7	6	5	4	3	2	1	0
	—	DAOE0	—	—	—	—	—	—
Initial value:	0	0	0	1	1	1	1	1
R/W :	R/W	R/W	R/W	—	—	—	—	—

Reserved
Only 0 should be written to this bit

D/A Output Enable 0

0	Analog output DA0 is disabled
1	Channel 0 D/A conversion is enabled; analog output DA0 is enabled

Reserved
Only 0 should be written to this bit

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	OPE	—	—	—
Initial value	0	0	0	0	1	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—

Output Port Enable

0	In software standby mode, address bus and bus control signals are high-impedance
1	In software standby mode, address bus and bus control signals retain their output state

Standby Timer Select

0	0	0	Standby time = 8192 states
		1	Standby time = 16384 states
	1	0	Standby time = 32768 states
		1	Standby time = 65536 states
1	0	0	Standby time = 131072 states
		1	Standby time = 262144 states
	1	0	Standby time = 2048 states
		1	Standby time = 16 states*

Note: * Cannot be used in the F-ZTAT version.

Software Standby

0	Transition to sleep mode after execution of SLEEP instruction
1	Transition to software standby mode after execution of SLEEP instruction

Bit	:	7	6	5	4	3	2	1	0
		—	—	INTM1	INTM0	NMIEG	MRESE	—	RAME
Initial value:		0	0	0	0	0	0	0	1
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—	R/W

RAM Enable

0	On-chip RAM is disabled
1	On-chip RAM is enabled

Manual Reset Select Bit

0	Manual reset is disabled
1	Manual reset is enabled

NMI Interrupt Input Edge Select

0	Interrupt request generated at falling edge of NMI input
1	Interrupt request generated at rising edge of NMI input

Interrupt Control Mode Select

0	0	Interrupt control mode 0 Interrupts controlled by I bit
	1	Setting prohibited
1	0	Interrupt control mode 2 Interrupts controlled by bits I2 to I0, and IPR
	1	Setting prohibited

Reserved

Only 0 should be written to this bit

Bit	7	6	5	4	3	2	1	0
	PSTOP	—	—	—	—	SCK2	SCK1	SCK0
Initial value:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	—	—	R/W	R/W	R/W	R/W

System Clock Select 2 to 0

0	0	0	Bus master is in high-speed mode
		1	Medium-speed clock is ø/2
	1	0	Medium-speed clock is ø/4
		1	Medium-speed clock is ø/8
1	0	0	Medium-speed clock is ø/16
		1	Medium-speed clock is ø/32
	1	—	—

Reserved

Only 0 should be written to this bit

Reserved

Only 0 should be written to this bit

ø Clock Output Disable

PSTOP	High-Speed Mode, Medium-Speed Mode	Sleep Mode	Software Standby Mode	Hardware Standby Mode
0	ø output	ø output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

MDCR—Mode Control Register H'FDE7 MCU

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	MDS2	MDS1	MDS0
Initial value:	1	0	0	0	0	—*	—*	—*
R/W :	—	—	—	—	—	R	R	R

Mode Select 2 to 0

These bits correspond to the mode pins (MD2 to MD0). When MDCR is read, the input levels at the mode pins (MD2 to MD0) are latched in bits MDS2 to MDS0.

Note: * Determined by pins MD_2 to MD_0.

MSTPCRA

Bit	7	6	5	4	3	2	1	0
	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	0	0	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRB

Bit	7	6	5	4	3	2	1	0
	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC

Bit	7	6	5	4	3	2	1	0
	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Module Stop

0	Module stop mode is cleared
1	Module stop mode is set

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	AE3	AE2	AE1	AE0
Modes 4 and 5								
Initial value	0	0	0	0	1	1	0	1
Modes 6 and 7								
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

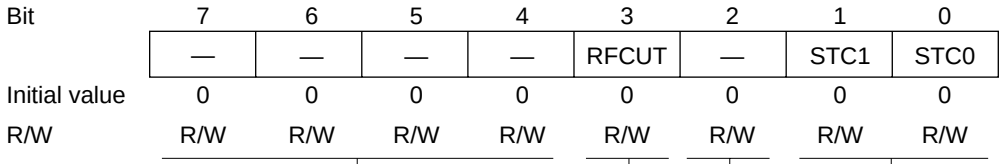
Address Output Enable 3 to 0

0	0	0	0	A8 to A23 output disabled
			1	A8 output enabled; A9 to A23 output disabled
		1	0	A8, A9 output enabled; A10 to A23 output disabled
			1	A8 to A10 output enabled; A11 to A23 output disabled
	1	0	0	A8 to A11 output enabled; A12 to A23 output disabled
			1	A8 to A12 output enabled; A13 to A23 output disabled
		1	0	A8 to A13 output enabled; A14 to A23 output disabled
			1	A8 to A14 output enabled; A15 to A23 output disabled
1	0	0	0	A8 to A15 output enabled; A16 to A23 output disabled
			1	A8 to A16 output enabled; A17 to A23 output disabled
		1	0	A8 to A17 output enabled; A18 to A23 output disabled
			1	A8 to A18 output enabled; A19 to A23 output disabled
	1	0	0	A8 to A19 output enabled; A20 to A23 output disabled
			1	A8 to A20 output enabled; A21 to A23 output disabled
		1	0	A8 to A21 output enabled; A22, A23 output disabled
			1	A8 to A23 output enabled

Note: In expanded mode with on-chip ROM enabled, address pins A0 to A7 are made address outputs by setting the corresponding DDR bits to 1; in expanded mode with on-chip ROM disabled, address pins A0 to A7 are always address outputs.

Reserved

Only 0 should be written to these bits



Bit	7	6	5	4	3	2	1	0
	—	—	—	—	RFCUT	—	STC1	STC0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Frequency Multiplication Factor

0	0	x1 (Initial value)
	1	x2 (Setting prohibited)
1	0	x4 (Setting prohibited)
	1	PLL is bypassed

Reserved

Only 0 should be written to this bit

Built-In Feedback Resistor Control

0	System clock oscillator's built-in feedback resistor and duty adjustment circuit are used
1	System clock oscillator's built-in feedback resistor and duty adjustment circuit are not used

Reserved

Only 0 should be written to these bits

Bit	7	6	5	4	3	2	1	0
	SSE	—	—	—	ABCS	ACS2	ACS1	ACS0
Initial value	0	Unde-fined	Unde-fined	Unde-fined	0	0	0	0
R/W	R/W	—	—	—	R/W	R/W	R/W	R/W

Asynchronous Clock Source Select 2 to 0

0	0	0	External clock input
		1	115.152 kbps average transfer rate (for ø = 10.667 MHz only) is selected (SCI0 operates on base clock with frequency of 16 times transfer rate)
	1	0	460.606 kbps average transfer rate (for ø = 10.667 MHz only) is selected (SCI0 operates on base clock with frequency of 8 times transfer rate)
		1	Reserved
1	0	0	TPU clock input (AND of TIOCA1 and TIOCA2)
		1	115.196 kbps average transfer rate (for ø = 16 MHz only) is selected (SCI0 operates on base clock with frequency of 16 times transfer rate)
	1	0	460.784 kbps average transfer rate (for ø = 16 MHz only) is selected (SCI0 operates on base clock with frequency of 16 times transfer rate)
		1	720 kbps average transfer rate (for ø = 16 MHz only) is selected (SCI0 operates on base clock with frequency of 8 times transfer rate)

Asynchronous Base Clock Select

0	SCI0 operates on base clock with frequency of 16 times transfer rate
1	SCI0 operates on base clock with frequency of 8 times transfer rate

Reserved Bits
Write 0 to these bits

SCI0 Select Enable

0	SCI0 select function disabled
1	SCI0 select function enabled When PG1/IRQ7 pin input = 1, TxD0 goes to high-impedance state and SCK0 clock input is fixed high

ISCRH

Bit :	15	14	13	12	11	10	9	8
	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
Initial value:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ7 to IRQ4 Sense Control A and B

ISCRL

Bit :	7	6	5	4	3	2	1	0
	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value:	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ3 to IRQ0 Sense Control A and B

IRQnSCB	IRQnSCA	Interrupt Request Generation
0	0	$\overline{\text{IRQn}}$ input low level
	1	Falling edge of $\overline{\text{IRQn}}$ input
1	0	Rising edge of $\overline{\text{IRQn}}$ input
	1	Both falling and rising edges of $\overline{\text{IRQn}}$ input

(n = 7 to 0)

IER—IRQ Enable Register — H'FE14 — Interrupt Controller

Bit	7	6	5	4	3	2	1	0
	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQn Enable

0	IRQn interrupts disabled
1	IRQn interrupts enabled

(n = 7 to 0)

ISR—IRQ Status Register — H'FE15 — Interrupt Controller

Bit	7	6	5	4	3	2	1	0
	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value:	0	0	0	0	0	0	0	0
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Indicates the status of IRQn interrupt requests

1	[Clearing conditions] (Initial value) • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed when low-level detection is set (IRQnSCB = IRQnSCA = 0) and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set (IRQnSCB = 1 or IRQnSCA = 1) • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0
2	[Setting conditions] • When $\overline{\text{IRQn}}$ input goes low when low-level detection is set (IRQnSCB = IRQnSCA = 0) • When a falling edge occurs in $\overline{\text{IRQn}}$ input when falling edge detection is set (IRQnSCB = 0, IRQnSCA = 1) • When a rising edge occurs in $\overline{\text{IRQn}}$ input when rising edge detection is set (IRQnSCB = 1, IRQnSCA = 0) • When a falling or rising edge occurs in $\overline{\text{IRQn}}$ input when both-edge detection is set (IRQnSCB = IRQnSCA = 1)

(n = 7 to 0)

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

Bit	7	6	5	4	3	2	1	0
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value:	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DTC Activation Enable

0	DTC activation by this interrupt is disabled [Clearing conditions] • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended
1	DTC activation by this interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified number of transfers have not ended

DTVECR—DTC Vector Register H'FE1F DTC

Bit	7	6	5	4	3	2	1	0
	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value :	0	0	0	0	0	0	0	0
R/W	R/(W)*1	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Sets vector number for DTC software activation

DTC Software Activation Enable

0	DTC software activation is disabled [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 is written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU
1	DTC software activation is enabled [Holding conditions] • When the DISEL bit is 1 and data transfer has ended • When the specified number of transfers have ended • During data transfer due to software activation

Notes: *1 Only 1 can be written to the SWDTE bit.
*2 Bits DTVEC6 to DTVEC0 can be written to when SWDTE = 0.

P1DDR—Port 1 Data Direction Register **H'FE30** **Port 1**

Bit	7	6	5	4	3	2	1	0
	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Specify input or output for the pins of port 1

P3DDR—Port 3 Data Direction Register **H'FE32** **Port 3**

Bit	7	6	5	4	3	2	1	0
	—	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	Undefined	0	0	0	0	0	0	0
R/W	—	W	W	W	W	W	W	W

Specify input or output for the pins of port 3

P7DDR—Port 7 Data Direction Register **H'FE36** **Port 7**

Bit	7	6	5	4	3	2	1	0
	P77DDR	P76DDR	P75DDR	P74DDR	P73DDR	P72DDR	P71DDR	P70DDR
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Specify input or output for the pins of port 7

PADDR—Port A Data Direction Register **H'FE39** **Port A**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR
Initial value	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	—	—	—	—	W	W	W	W

Specify input or output for the pins of port A

PBDDR—Port B Data Direction Register **H'FE3A** **Port B**

Bit	7	6	5	4	3	2	1	0
	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Specify input or output for the pins of port B

PCDDR—Port C Data Direction Register **H'FE3B** **Port C**

Bit	7	6	5	4	3	2	1	0
	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Specify input or output for the pins of port C

PDDDR—Port D Data Direction Register **H'FE3C** **Port D**

Bit	7	6	5	4	3	2	1	0
	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Specify input or output for the pins of port D

PEDDR—Port E Data Direction Register **H'FE3D** **Port E**

Bit	7	6	5	4	3	2	1	0
	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Specify input or output for the pins of port E

PFDDR—Port F Data Direction Register **H'FE3E** **Port F**

Bit	7	6	5	4	3	2	1	0
	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR
Modes 4 to 6								
Initial value	1	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W
Mode 7								
Initial value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Specify input or output for the pins of port F

PGDDR—Port G Data Direction Register **H'FE3F** **Port G**

Bit	7	6	5	4	3	2	1	0
	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR
Modes 4 and 5								
Initial value	Undefined	Undefined	Undefined	1	0	0	0	0
R/W	—	—	—	W	W	W	W	W
Modes 6 and 7								
Initial value	Undefined	Undefined	Undefined	0	0	0	0	0
R/W	—	—	—	W	W	W	W	W

Specify input or output for the pins of port G

PAPCR—Port A MOS Pull-Up Control Register H'FE40 Port A

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR
Initial value	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port A on a bit-by-bit basis

PBPCR—Port B MOS Pull-Up Control Register H'FE41 Port B

Bit	7	6	5	4	3	2	1	0
	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port B on a bit-by-bit basis

PCPCR—Port C MOS Pull-Up Control Register H'FE42 Port C

Bit	7	6	5	4	3	2	1	0
	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port C on a bit-by-bit basis

PDPCR—Port D MOS Pull-Up Control Register H'FE43 Port D

Bit	7	6	5	4	3	2	1	0
	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port D on a bit-by-bit basis

PEPCR—Port E MOS Pull-Up Control Register H'FE44 Port E

Bit	7	6	5	4	3	2	1	0
	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the MOS input pull-up function incorporated into port E on a bit-by-bit basis

P3ODR—Port 3 Open-Drain Control Register H'FE46 Port 3

Bit	7	6	5	4	3	2	1	0
	—	P36ODR	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR
Initial value	Undefined	0	0	0	0	0	0	0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Controls the PMOS on/off status for each port 3 pin (P36 to P30)

PAODR—Port A Open-Drain Control Register H'FE47 Port A

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR
Initial value	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

Controls the PMOS on/off status for each port A pin (PA3 to PA0)

Bit	:	7	6	5	4	3	2	1	0
		P36 IRQ7E	P47 IRQ6E	P46 IRQ5E	P44 IRQ4E	P43 IRQ3E	P42 IRQ2E	P41 IRQ1E	P40 IRQ0E
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Enable of $\overline{\text{EXIRQ0}}$ Input from P40 (P40IRQ0E)

0	P40 is not used as $\overline{\text{EXIRQ0}}$ input
1	P40 is used as $\overline{\text{EXIRQ0}}$ input

Enable of $\overline{\text{EXIRQ1}}$ Input from P41 (P41IRQ1E)

0	P41 is not used as $\overline{\text{EXIRQ1}}$ input
1	P41 is used as $\overline{\text{EXIRQ1}}$ input

Enable of $\overline{\text{EXIRQ2}}$ Input from P42 (P42IRQ2E)

0	P42 is not used as $\overline{\text{EXIRQ2}}$ input
1	P42 is used as $\overline{\text{EXIRQ2}}$ input

Enable of $\overline{\text{EXIRQ3}}$ Input from P43 (P43IRQ3E)

0	P43 is not used as $\overline{\text{EXIRQ3}}$ input
1	P43 is used as $\overline{\text{EXIRQ3}}$ input

Enable of $\overline{\text{EXIRQ4}}$ Input from P44 (P44IRQ4E)

0	P44 is not used as $\overline{\text{EXIRQ4}}$ input
1	P44 is used as $\overline{\text{EXIRQ4}}$ input

Enable of $\overline{\text{EXIRQ5}}$ Input from P46 (P46IRQ5E)

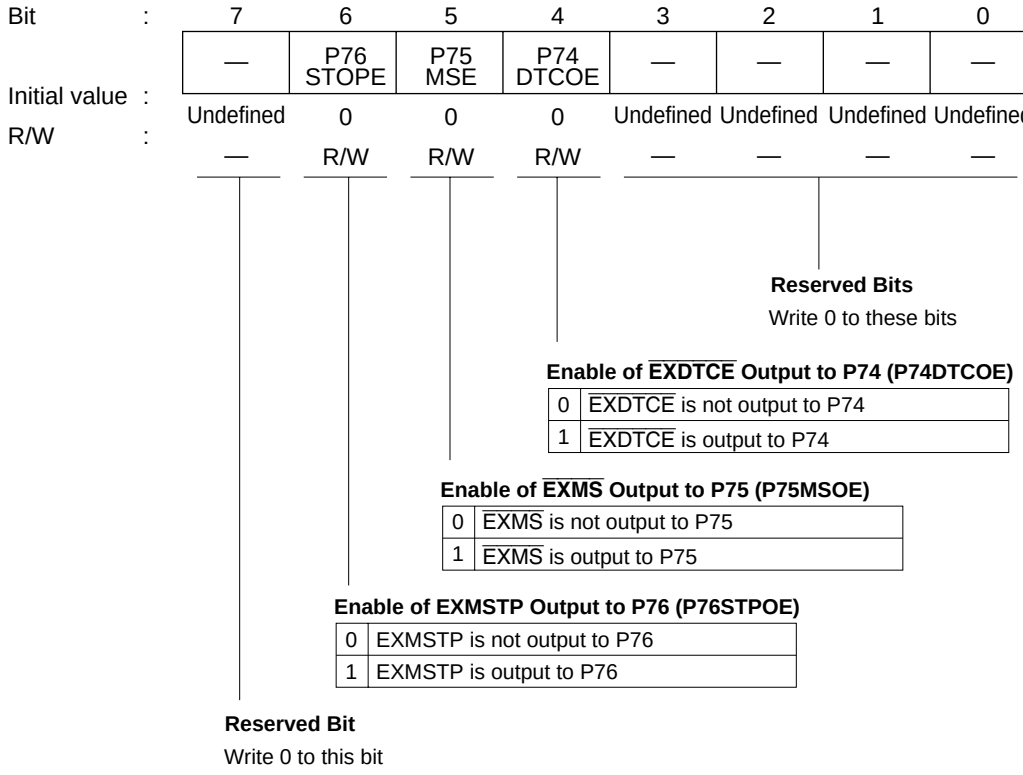
0	P46 is not used as $\overline{\text{EXIRQ5}}$ input
1	P46 is used as $\overline{\text{EXIRQ5}}$ input

Enable of $\overline{\text{EXIRQ6}}$ Input from P47 (P47IRQ6E)

0	P47 is not used as $\overline{\text{EXIRQ6}}$ input
1	P47 is used as $\overline{\text{EXIRQ6}}$ input

Enable of $\overline{\text{EXIRQ7}}$ Input from P36 (P36IRQ7E)

0	P36 is not used as $\overline{\text{EXIRQ7}}$ input
1	P36 is used as $\overline{\text{EXIRQ7}}$ input


Bit : 7 6 5 4 3 2 1 0
— P76 STOPE P75 MSE P74 DTCOE — — — —
Initial value : Undefined 0 0 0 Undefined Undefined Undefined Undefined
R/W : — R/W R/W R/W — — — —
Reserved Bits
Write 0 to these bits
Enable of EXDTCE Output to P74 (P74DTCOE)
0 EXDTCE is not output to P74
1 EXDTCE is output to P74
Enable of EXMS Output to P75 (P75MSOE)
0 EXMS is not output to P75
1 EXMS is output to P75
Enable of EXMSTP Output to P76 (P76STPOE)
0 EXMSTP is not output to P76
1 EXMSTP is output to P76
Reserved Bit
Write 0 to this bit

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	CST2	CST1	CST0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	—	R/W	R/W	R/W

Counter Start 2 to 0

0	TCNTn count operation is stopped
1	TCNTn performs count operation

(n = 2 to 0)

Note: 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.

TSYR—Timer Synchro Register **H'FEB1** **TPU**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	SYNC2	SYNC1	SYNC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	—	—	—	R/W	R/W	R/W

Timer Synchro

0	TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

(n = 2 to 0)

Notes: 1. To set synchronous operation, the SYNC bits for at least two channels must be set to 1.
2. 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.

IPRA	**—Interrupt Priority Register A**	**H'FEC0**
IPRB	**—Interrupt Priority Register B**	**H'FEC1**
IPRC	**—Interrupt Priority Register C**	**H'FEC2**
IPRD	**—Interrupt Priority Register D**	**H'FEC3**
IPRF	**—Interrupt Priority Register F**	**H'FEC5**
IPRG	**—Interrupt Priority Register G**	**H'FEC6**
IPRJ	**—Interrupt Priority Register J**	**H'FEC9**
IPRK	**—Interrupt Priority Register K**	**H'FECA**
IPRM	**—Interrupt Priority Register M**	**H'FECC**

Bit	:	7	6	5	4	3	2	1	0
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0
Initial value	:	0	1	1	1	0	1	1	1
R/W	:	—	R/W	R/W	R/W	—	R/W	R/W	R/W

Set priority (levels 7 to 0) for interrupt sources

Correspondence between Interrupt Sources and IPR Settings

Register	Bits	
	6 to 4	2 to 0
IPRA	IRQ0	IRQ1
IPRB	IRQ2, IRQ3	IRQ4, IRQ5
IPRC	IRQ6, IRQ7	DTC
IPRD	Watchdog timer 0	—*
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	—
IPRJ	DMAC	SCI channel 0
IPRK	SCI channel 1	SCI channel 2
IPRM	EXIRQ3 to EXIRQ0	EXIRQ7 to EXIRQ4

Note: * Reserved bits. These bits cannot be modified and are always read as 1.

Bit	7	6	5	4	3	2	1	0
	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
Modes 5 to 7								
Initial value	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Mode 4								
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 Bus Width Control

0	Area n is designated for 16-bit access
1	Area n is designated for 8-bit access

(n = 7 to 0)

ASTCR—Access State Control Register **H'FED1** **Bus Controller**

Bit	7	6	5	4	3	2	1	0
	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
Initial value	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 Access State Control

0	Area n is designated for 2-state access Wait state insertion in area n external space is disabled
1	Area n is designated for 3-state access Wait state insertion in area n external space is enabled

(n = 7 to 0)

Bit	:	7	6	5	4	3	2	1	0
		W71	W70	W61	W60	W51	W50	W41	W40
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 4 Wait Control 1 and 0

0	0	Program wait not inserted in area 4 external space access
	1	1 program wait state inserted in area 4 external space access
1	0	2 program wait states inserted in area 4 external space access
	1	3 program wait states inserted in area 4 external space access

Area 5 Wait Control 1 and 0

0	0	Program wait not inserted in area 5 external space access
	1	1 program wait state inserted in area 5 external space access
1	0	2 program wait states inserted in area 5 external space access
	1	3 program wait states inserted in area 5 external space access

Area 6 Wait Control 1 and 0

0	0	Program wait not inserted in area 6 external space access
	1	1 program wait state inserted in area 6 external space access
1	0	2 program wait states inserted in area 6 external space access
	1	3 program wait states inserted in area 6 external space access

Area 7 Wait Control 1 and 0

0	0	Program wait not inserted in area 7 external space access
	1	1 program wait state inserted in area 7 external space access
1	0	2 program wait states inserted in area 7 external space access
	1	3 program wait states inserted in area 7 external space access

Bit	:	7	6	5	4	3	2	1	0
		W31	W30	W21	W20	W11	W10	W01	W00
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 0 Wait Control 1 and 0

0	0	Program wait not inserted in area 0 external space access
	1	1 program wait state inserted in area 0 external space access
1	0	2 program wait states inserted in area 0 external space access
	1	3 program wait states inserted in area 0 external space access

Area 1 Wait Control 1 and 0

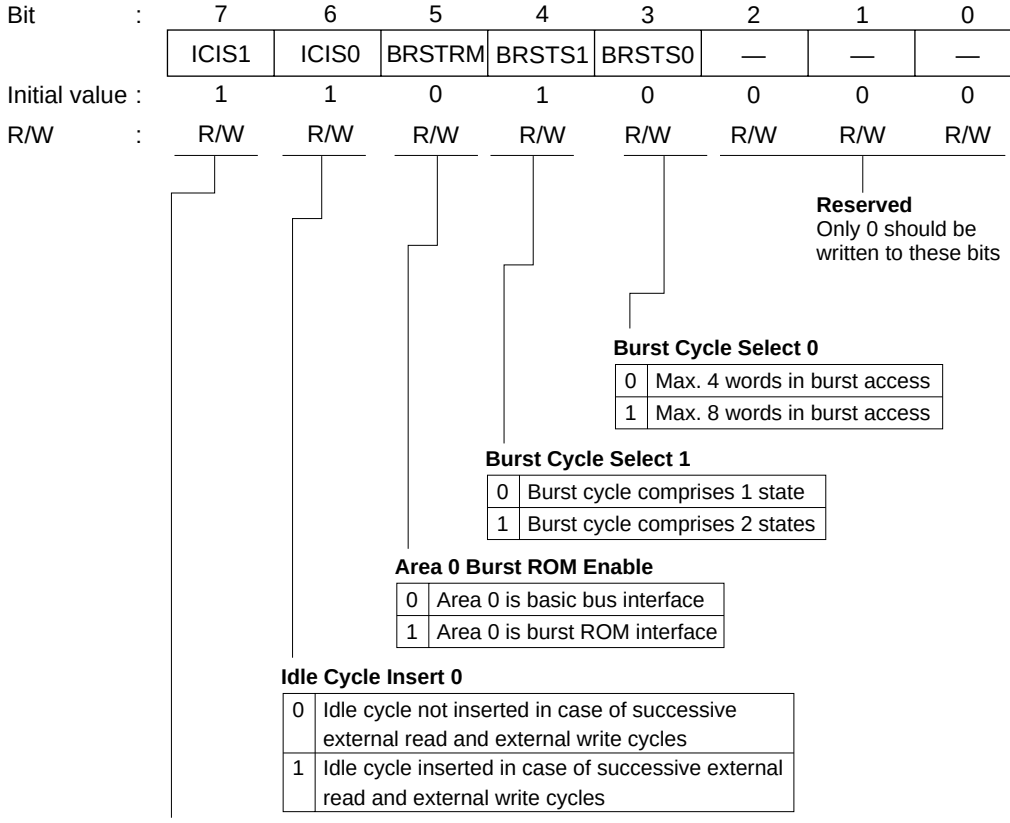
0	0	Program wait not inserted in area 1 external space access
	1	1 program wait state inserted in area 1 external space access
1	0	2 program wait states inserted in area 1 external space access
	1	3 program wait states inserted in area 1 external space access

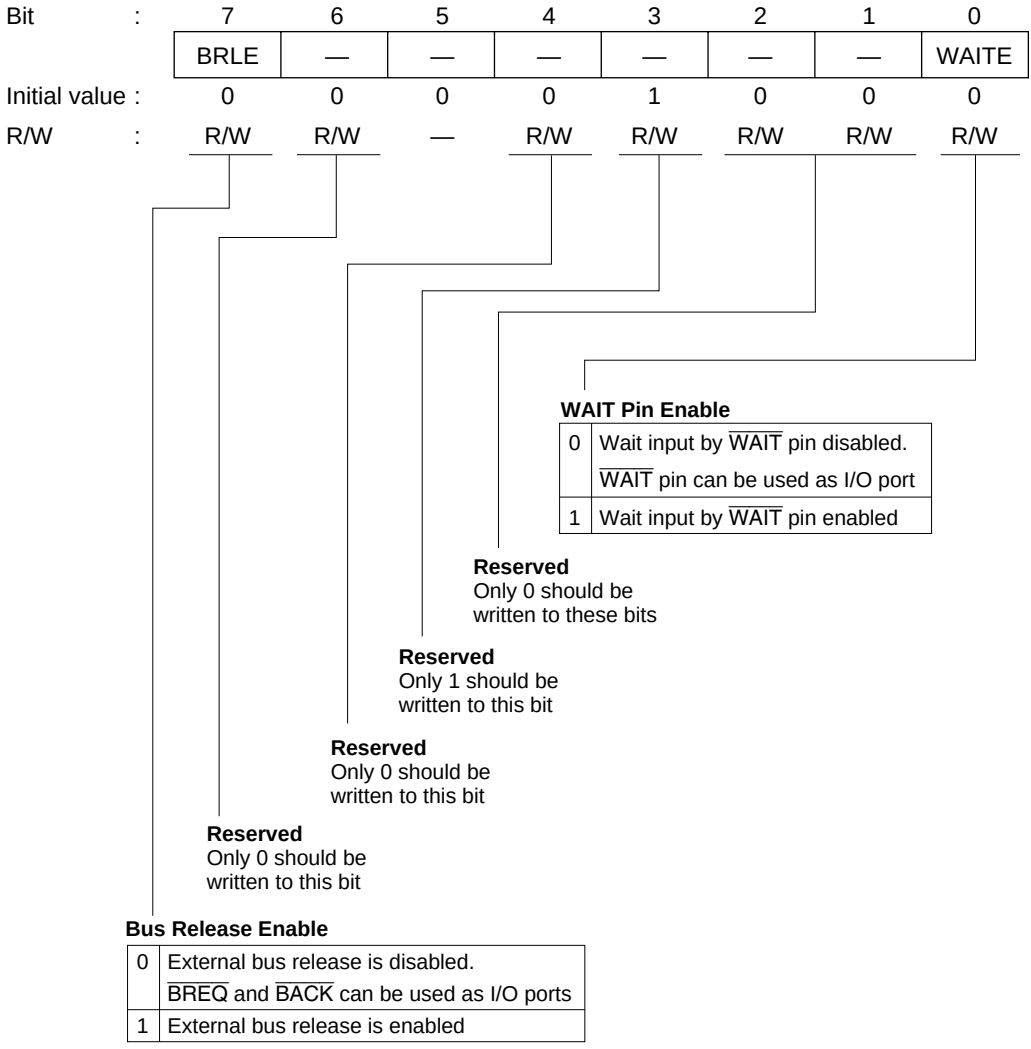
Area 2 Wait Control 1 and 0

0	0	Program wait not inserted in area 2 external space access
	1	1 program wait state inserted in area 2 external space access
1	0	2 program wait states inserted in area 2 external space access
	1	3 program wait states inserted in area 2 external space access

Area 3 Wait Control 1 and 0

0	0	Program wait not inserted in area 3 external space access
	1	1 program wait state inserted in area 3 external space access
1	0	2 program wait states inserted in area 3 external space access
	1	3 program wait states inserted in area 3 external space access


Bit : 7 6 5 4 3 2 1 0
ICIS1 ICIS0 BRSTRM BRSTS1 BRSTS0 — — —
Initial value : 1 1 0 1 0 0 0 0
R/W : R/W R/W R/W R/W R/W R/W R/W R/W
Reserved
Only 0 should be written to these bits
Burst Cycle Select 0
0 Max. 4 words in burst access
1 Max. 8 words in burst access
Burst Cycle Select 1
0 Burst cycle comprises 1 state
1 Burst cycle comprises 2 states
Area 0 Burst ROM Enable
0 Area 0 is basic bus interface
1 Area 0 is burst ROM interface
Idle Cycle Insert 0
0 Idle cycle not inserted in case of successive external read and external write cycles
1 Idle cycle inserted in case of successive external read and external write cycles
Idle Cycle Insert 1
0 Idle cycle not inserted in case of successive external read cycles in different areas
1 Idle cycle inserted in case of successive external read cycles in different areas


Bit : 7 6 5 4 3 2 1 0
BRLE — — — — — — WAITE
Initial value : 0 0 0 0 1 0 0 0
R/W : R/W R/W — R/W R/W R/W R/W R/W
WAIT Pin Enable
0 Wait input by WAIT pin disabled.
WAIT pin can be used as I/O port
1 Wait input by WAIT pin enabled
Reserved
Only 0 should be written to these bits
Reserved
Only 1 should be written to this bit
Reserved
Only 0 should be written to this bit
Reserved
Only 0 should be written to this bit
Bus Release Enable
0 External bus release is disabled.
BREQ and BACK can be used as I/O ports
1 External bus release is enabled

RAMER—RAM Emulation Register **H'FEDB** **Flash**

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

RAM Select

Flash Memory Area Selection

0	Emulation not selected Program/erase-protection of all flash memory blocks is disabled
1	Emulation selected Program/erase-protection of all flash memory blocks is enabled

MAR0A—Memory Address Register 0A **H'FEE0** **DMAC**

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

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

* : Undefined

IOAR0A—I/O Address Register 0A **H'FEE4** **DMAC**

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

* : Undefined

ETCR0A—Transfer Count Register 0A **H'FEE6** **DMAC**

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

* : Undefined

ETCR0B—Transfer Count Register 0B **H'FEEE** **DMAC**

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

* : Undefined

MAR1A—Memory Address Register 1A **H'FEF0** **DMAC**

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

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

* : Undefined

IOAR1A—I/O Address Register 1A **H'FEF4** **DMAC**

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

* : Undefined

Maintenance of transfer count

Bit	:	15	14	13	12	11	10	9	8
ETCRH	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transfer counter

Bit	:	7	6	5	4	3	2	1	0
ETCRL	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

* : Undefined

ETCR1B—Transfer Count Register 1B **H'FEFE** **DMAC**

Maintenance of transfer count

Bit	:	15	14	13	12	11	10	9	8
ETCRH	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transfer counter

Bit	:	7	6	5	4	3	2	1	0
ETCRL	:								
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

* : Undefined

P1DR—Port 1 Data Register **H'FF00** **Port 1**

Bit	:	7	6	5	4	3	2	1	0
		P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for the port 1 pins (P17 to P10)

P3DR—Port 3 Data Register **H'FF02** **Port 3**

Bit	:	7	6	5	4	3	2	1	0
		—	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
Initial value	:	Undefined	0	0	0	0	0	0	0
R/W	:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for the port 3 pins (P36 to P30)

P7DR—Port 7 Data Register **H'FF06** **Port 7**

Bit	:	7	6	5	4	3	2	1	0
		P77DR	P76DR	P75DR	P74DR	P73DR	P72DR	P71DR	P70DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for the port 7 pins (P77 to P70)

PADR—Port A Data Register **H'FF09** **Port A**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

Stores output data for the port A pins (PA3 to PA0)

PBDR—Port B Data Register **H'FF0A** **Port B**

Bit	7	6	5	4	3	2	1	0
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for the port B pins (PB7 to PB0)

PCDR—Port C Data Register **H'FF0B** **Port C**

Bit	7	6	5	4	3	2	1	0
	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for the port C pins (PC7 to PC0)

PDDR—Port D Data Register **H'FF0C** **Port D**

Bit	7	6	5	4	3	2	1	0
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for the port D pins (PD7 to PD0)

PEDR—Port E Data Register **H'FF0D** **Port E**

Bit	7	6	5	4	3	2	1	0
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for the port E pins (PE7 to PE0)

PFDR—Port F Data Register **H'FF0E** **Port F**

Bit	:	7	6	5	4	3	2	1	0
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for the port F pins (PF7 to PF0)

PGDR—Port G Data Register **H'FF0F** **Port G**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
Initial value	:	Undefined	Undefined	Undefined	0	0	0	0	0
R/W	:	—	—	—	R/W	R/W	R/W	R/W	R/W

Stores output data for the port G pins (PG4 to PG0)

Bit	:	7	6	5	4	3	2	1	0
		CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time Prescaler 2 to 0

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/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

Clock Edge 1 and 0

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Counter Clear 2 to 0

0	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input capture
	1	0	TCNT cleared by TGRB compare match/input capture
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation *1
1	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRC compare match/input capture *2
	1	0	TCNT cleared by TGRD compare match/input capture *2
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation *1

Notes: *1 Synchronous operation 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.

Bit	:	7	6	5	4	3	2	1	0
		—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Modes 3 to 0

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	*	*	*	—

*: Don't care

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.
2. Phase counting mode cannot be set for channel 0. In this case, 0 should always be written to MD2.

Buffer Operation A

0	TGRA operates normally
1	TGRA and TGRC used together for buffer operation

Buffer Operation B

0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

I/O Control A3 to A0

0	0	0	0	TGR0A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0A is input capture register	Capture input source isTIOCA0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Setting prohibited	

*: Don't care

I/O Control B3 to B0

0	0	0	0	TGR0B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0B is input capture register	Capture input source isTIOCB0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Setting prohibited	

*: Don't care

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Bit	:	7	6	5	4	3	2	1	0
		IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

I/O Control C3 to C0

0	0	0	0	TGR0C is output compare register*1	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0C is input capture register*1	Capture input source isTIOCC0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Setting prohibited	

*: Don't care

Note: *1 When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

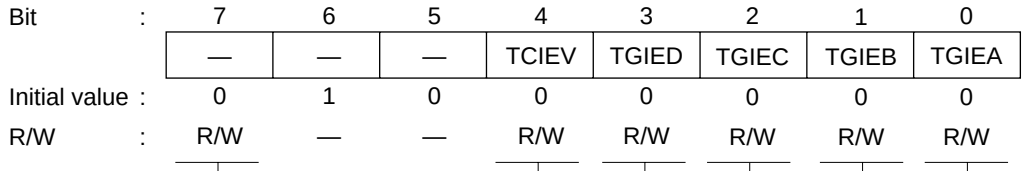
I/O Control D3 to D0

0	0	0	0	TGR0D is output compare register*1	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0D is input capture register*1	Capture input source is TIOCD0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Setting prohibited	

*: Don't care

Note: *1 When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.



Bit	7	6	5	4	3	2	1	0
	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
R/W	R/W	—	—	R/W	R/W	R/W	R/W	R/W

TGR Interrupt Enable A

0	Interrupt requests (TGIA) by TGFA bit disabled
1	Interrupt requests (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt requests (TGIB) by TGFB bit disabled
1	Interrupt requests (TGIB) by TGFB bit enabled

TGR Interrupt Enable C

0	Interrupt requests (TGIC) by TGFC bit disabled
1	Interrupt requests (TGIC) by TGFC bit enabled

TGR Interrupt Enable D

0	Interrupt requests (TGID) by TGFD bit disabled
1	Interrupt requests (TGID) by TGFD bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Reserved

Only 0 should be written to this bit

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Input Capture/Output Compare Flag C

0	[Clearing conditions] • When DTC is activated by TGIC interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register

Input Capture/Output Compare Flag D

0	[Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Note: * Can only be written with 0 for flag clearing.

TCNT0—Timer Counter 0 **H'FF16** **TPU0**

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

Up-counter

TGR0A—Timer General Register 0A **H'FF18** **TPU0**
TGR0B—Timer General Register 0B **H'FF1A**
TGR0C—Timer General Register 0C **H'FF1C**
TGR0D—Timer General Register 0D **H'FF1E**

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

Bit	:	7	6	5	4	3	2	1	0
		—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time Prescaler 2 to 0

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/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 ø/256
		1	Setting prohibited

Note: This setting is ignored when channel 1 is in phase counting mode.

Clock Edge 1 and 0

0	0	Count at rising edge
	1	Count at falling edge
1	—*	Count at both edges

Note: * The internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Counter Clear 2 to 0

0*1	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input capture
	1	0	TCNT cleared by TGRB compare match/input capture
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*2

Notes: *1 Bit 7 is reserved.
It cannot be modified and is always read as 0.
*2 Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

Mode

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	*	*	*	—

*: Don’t care

Note: MD3 is a reserved bit. In a write, it should always be written with 0.

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

I/O Control A3 to A0

0	0	0	0	TGR1A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR1A is input capture register	Capture input source isTIOCA1 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Setting prohibited	

*: Don’t care

I/O Control B3 to B0

0	0	0	0	TGR1B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR1B is input capture register	Capture input source isTIOCB1 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Setting prohibited	

*: Don’t care

Bit	7	6	5	4	3	2	1	0
	—	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
R/W	R/W	—	R/W	R/W	—	—	R/W	R/W

TGR Interrupt Enable A

0	Interrupt requests (TGIA) by TGFA bit disabled
1	Interrupt requests (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt requests (TGIB) by TGFB bit disabled
1	Interrupt requests (TGIB) by TGFB bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

Reserved
Only 0 should be written to this bit

Bit	:	7	6	5	4	3	2	1	0
		TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Underflow Flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

TCNT1—Timer Counter 1 **H'FF26** **TPU1**

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

Up/down-counter*

Note : * These counters can be used as up/down-counters only in phase counting mode.
In other cases they function as up-counters.

TGR1A—Timer General Register 1A **H'FF28** **TPU1**
TGR1B—Timer General Register 1B **H'FF2A**

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

Bit	:	7	6	5	4	3	2	1	0
		—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time Prescaler 2 to 0

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/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 ø/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Clock Edge 1 and 0

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: The internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Counter Clear 2 to 0

0*1	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input capture
	1	0	TCNT cleared by TGRB compare match/input capture
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*2

Notes: *1 Bit 7 is reserved.
It cannot be modified and is alway read as 0.
*2 Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

Mode

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	*	*	*	—

*: Don’t care

Note: MD3 is a reserved bit. In a write, it should always be written with 0.

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

I/O Control A3 to A0

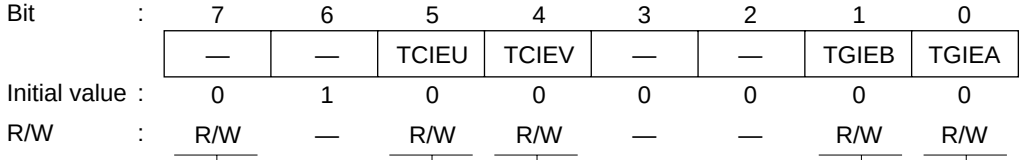
0	0	0	0	TGR2A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR2A is input capture register	Capture input source isTIOCA2 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don’t care

I/O Control B3 to B0

0	0	0	0	TGR2B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR2B is input capture register	Capture input source isTIOCB2 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don’t care



Bit	7	6	5	4	3	2	1	0
	—	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
R/W	R/W	—	R/W	R/W	—	—	R/W	R/W

TGR Interrupt Enable A

0	Interrupt requests (TGIA) by TGFA bit disabled
1	Interrupt requests (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt requests (TGIB) by TGFB bit disabled
1	Interrupt requests (TGIB) by TGFB bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

Reserved

Only 0 should be written to this bit

Bit	7	6	5	4	3	2	1	0
	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	1	1	0	0	0	0	0	0
R/W	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When DMAC is activated by TGIA interrupt while DTA bit of DMABCR in DMAC is 1 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Underflow Flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

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

Up/down-counter*

Note : * These counters can be used as up/down-counters only in phase counting mode. In other cases they function as up-counters.

TGR2A—Timer General Register 2A **H'FF38** **TPU2**
TGR2B—Timer General Register 2B **H'FF3A**

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

DMAWER—DMA Write Enable Register H'FF60 DMAC

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	WE1B	WE1A	WE0B	WE0A
Initial value	0	0	0	0	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

Write enable 1B

0	Disables writing to all DMACR1B bits, DMABCR bits 11, 7, and 3, and DMATCR bit 5.
1	Enables writing to all DMACR1B bits, DMABCR bits 11, 7, and 3, and DMATCR bit 5.

Write enable 1A

0	Disables writing to all DMACR1A bits, and DMABCR bits 10, 6, and 2.
1	Enables writing to all DMACR1A bits, and DMABCR bits 10, 6, and 2.

Write enable 0B

0	Disables writing to all DMACR0B bits, DMABCR bits 9, 5, and 1, and DMATCR bit 4.
1	Enables writing to all DMACR0B bits, DMABCR bits 9, 5, and 1, and DMATCR bit 4.

Write enable 0A

0	Disables writing to all DMACR0A bits, and DMABCR bits 8, 4, and 0.
1	Enables writing to all DMACR0A bits, and DMABCR bits 8, 4, and 0.

DMATCR—DMA Terminal Control Register H'FF61 DMAC

Bit	7	6	5	4	3	2	1	0
	—	—	TEE1	TEE0	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	—	—	R/W	R/W	—	—	—	—

Transfer end pin enable 0

0	Disables $\overline{\text{TEND0}}$ pin output.
1	Enables $\overline{\text{TEND0}}$ pin output.

Transfer end pin enable 1

0	Disables $\overline{\text{TEND1}}$ pin output.
1	Enables $\overline{\text{TEND1}}$ pin output.

Bit	:	7	6	5	4	3	2	1	0
DMACR	:	DTSZ	DTID	RPE	DTDIR	DTF3	DTF2	DTF1	DTF0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data transfer direction

DMABCR		
SAE	DTDIR	
0	0	Transfer from MAR as source address to IOAR as destination address.
	1	Transfer from IOAR as source address to MAR as destination address.
1	0	Transfer from MAR as source address with $\overline{\text{DACK}}$ pin as write strobe.
	1	Transfer with $\overline{\text{DACK}}$ pin as read strobe to MAR as destination address.

Repeat enable

	DMABCR	
RPE	DTIE	
0	0	Sequential mode transfer (no transfer end interrrupt).
	1	Sequential mode transfer (with transfer end interrupt).
1	0	Transfer in repeat mode (no transfer end interrupt).
	1	Transfer in idle mode (with transfer end interrupt).

Data transfer increment/decrement

0	MAR incremented after data transfer (1) When DTSZ=0, MAR + 1 after transfer. (2) When DTSZ=1, MAR + 2 after transfer.
1	MAR decremented after data transfer (1) When DTSZ=0, MAR – 1 after transfer. (2) When DTSZ=1, MAR – 2 after transfer.

Data transfer size

0	Byte size transfer
1	Word size transfer

Data transfer factor

Channel A

0	0	0	0	—
			1	—
		1	0	—
			1	—
	1	0	0	Starts on SCI channel 0 transmit end interrupt
			1	Starts on SCI channel 0 receive end interrupt
		1	0	Starts on SCI channel 1 transmit end interrupt
			1	Starts on SCI channel 1 receive end interrupt
1	0	0	0	Starts on TPU channel 0 compare match/input capture A interrupt
			1	Starts on TPU channel 1 compare match/input capture A interrupt
		1	0	Starts on TPU channel 2 compare match/input capture A interrupt
			1	—
	1	0	0	—
			1	—
		1	0	—
			1	—

Channel B

0	0	0	0	—
			1	—
		1	0	Starts on falling edge of input at $\overline{\text{DREQ}}$ pin
			1	Starts on LOW level input at $\overline{\text{DREQ}}$ pin
	1	0	0	Starts on SCI channel 0 transmit end interrupt
			1	Starts on SCI channel 0 receive end interrupt
		1	0	Starts on SCI channel 1 transmit end interrupt
			1	Starts on SCI channel 1 receive end interrupt
1	0	0	0	Starts on TPU channel 0 compare match/input capture A interrupt
			1	Starts on TPU channel 1 compare match/input capture A interrupt
		1	0	Starts on TPU channel 2 compare match/input capture A interrupt
			1	—
	1	0	0	—
			1	—
		1	0	—
			1	—

Note: * Detect the first transfer after transfers have been enabled as a LOW level signal.

Bit :	15	14	13	12	11	10	9	8
DMABCRH :	FAE1	FAE0	—	—	DTA1B	DTA1A	DTA0B	DTA0A
Initial value :	0	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W*	R/W*	R/W	R/W	R/W	R/W

Data transfer acknowledge 0A

0	Clearing of selected internal interrupt factor at DMA transfer disabled.
1	Clearing of selected internal interrupt factor at DMA transfer enabled.

Data transfer acknowledge 0B

0	Clearing of selected internal interrupt factor at DMA transfer disabled.
1	Clearing of selected internal interrupt factor at DMA transfer enabled.

Data transfer acknowledge 1A

0	Clearing of selected internal interrupt factor at DMA transfer disabled.
1	Clearing of selected internal interrupt factor at DMA transfer enabled.

Data transfer acknowledge 1B

0	Clearing of selected internal interrupt factor at DMA transfer disabled.
1	Clearing of selected internal interrupt factor at DMA transfer enabled.

Full address enable 0

0	Short address mode.
1	Full address mode.

Full address enable 1

0	Short address mode.
1	Full address mode.

Note: * Only 0 can be written to, writing 1 causes a malfunction error.

Bit	:	7	6	5	4	3	2	1	0
DMABCRL	:	DTE1B	DTE1A	DTE0B	DTE0A	DTIE1B	DTIE1A	DTIE0B	DTIE0A
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data transfer interrupt enable 0A

0	Transfer end interrupt disabled.
1	Transfer end interrupt enabled.

Data transfer interrupt enable 0B

0	Transfer end interrupt disabled.
1	Transfer end interrupt enabled.

Data transfer interrupt enable 1A

0	Transfer end interrupt disabled.
1	Transfer end interrupt enabled.

Data transfer interrupt enable 1B

0	Transfer end interrupt disabled.
1	Transfer end interrupt enabled.

Data transfer enable 0A

0	Data transfer disabled.
1	Data transfer enabled.

Data transfer enable 0B

0	Data transfer disabled.
1	Data transfer enabled.

Data transfer enable 1A

0	Data transfer disabled.
1	Data transfer enabled.

Data transfer enable 1B

0	Data transfer disabled.
1	Data transfer enabled.

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	0	0	0	1	1	0	0	0
Read/Write	R/(W)*	R/W	R/W	—	—	R/W	R/W	R/W

Clock Select 2 to 0

CKS2	CKS1	CKS0	Clock	Overflow Period* (when ø = 10 MHz)
0	0	0	ø/2 (Initial value)	51.2 µs
		1	ø/64	1.6 ms
	1	0	ø/128	3.2 ms
		1	ø/512	13.2 ms
1	0	0	ø/2048	52.4 ms
		1	ø/8192	209.8 ms
	1	0	ø/32768	838.8 ms
		1	ø/131072	3.36 s

Note: * The overflow period is the time from when TCNT starts counting up from H'00 until overflow occurs.

Timer Enable

0	TCNT is initialized to H'00 and count operation is halted
1	TCNT counts

Timer Mode Select

0	Interval timer mode: Interval timer interrupt (WOVI) request is sent to CPU when TCNT overflows
1	Watchdog timer mode: Internal reset can be selected when TCNT overflows*

Note: * For details of the case where TCNT overflows in watchdog timer mode, see section 11.2.3, Reset Control/Status Register (RSTCSR).

Overflow Flag

0	[Clearing conditions] Overflow Flag Read TCSR when OVF = 1, then write 0 in OVF
1	[Setting condition] When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.

Note: * Only 0 can be written, to clear the flag.
TCSR is write-protected by a password to prevent accidental overwriting. For details see section 11.2.4, Notes on Register Access.

TCNT0—Timer Counter **H'FF74(W)** **WDT0**
H'FF75(R)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RSTCSR—Reset Control/Status Register **H'FF76(W)** **WDT0**
H'FF77(R)

Bit	:	7	6	5	4	3	2	1	0
		WOVF	RSTE	RSTS	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
R/W	:	R/(W)*	R/W	R/W	—	—	—	—	—

Reset Select

0	Power-on reset
1	Manual reset

Reset Enable

0	No internal reset when TCNT overflows*
1	Internal reset is generated when TCNT overflows

Note: * The chip is not reset internally, but TCNT and TCSR in WDT0 are reset.

Watchdog Overflow Flag

0	[Clearing condition] Cleared by reading RSTCSR when WOVF = 1, then writing 0 to WOVF
1	[Setting condition] When TCNT overflows (from H'FF to H'00) in watchdog timer mode

Note: * Only 0 can be written, to clear the flag.
RSTCSR is write-protected by a password to prevent accidental overwriting.
For details see section 11.2.4, Notes on Register Access.

Bit	:	7	6	5	4	3	2	1	0
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select 1 and 0

0	0	ø clock
	1	ø/4 clock
1	0	ø/16 clock
	1	ø/64 clock

Multiprocessor Mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character Length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it is not possible to choose between LSB-first or MSB-first mode.

Communication Mode

0	Asynchronous mode
1	Clocked synchronous mode

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sets the serial transfer bit rate

Note: For details, see section 12.2.8, Bit Rate Register (BRR)

Bit	:	7	6	5	4	3	2	1	0
		TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable 1 and 0

0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*1
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input

Notes: *1 Outputs a clock of the same frequency as the bit rate.
*2 Inputs a clock with a frequency 16 times the bit rate.

Transmit End Interrupt Enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (ERI) requests, and settingof the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

0	Transmission disabled
1	Transmission enabled

Receive Interrupt Enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit data empty interrupt (TXI) requests disabled
1	Transmit data empty interrupt (TXI) requests enabled


Bit : 7 6 5 4 3 2 1 0
Initial value : 1 1 1 1 1 1 1 1
R/W : R/W R/W R/W R/W R/W R/W R/W R/W
Stores data for serial transmission

Bit	:	7	6	5	4	3	2	1	0
		TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	:	1	0	0	0	0	1	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor Bit Transfer

0	Data with a 0 multiprocessor bit is transmitted
1	Data with a 1 multiprocessor bit is transmitted

Multiprocessor Bit

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Parity Error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR

Framing Error

0	[Clearing condition] When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0

Overrun Error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive Data Register Full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt request and reads data to RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit Data Register Empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

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

RDR0—Receive Data Register 0 **H'FF7D** **SCI0**

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R

Stores received serial data

SCMR0—Smart Card Mode Register 0 **H'FF7E** **SCI0**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

Reserved
Only 0 should be written to this bit

Smart Card Data Invert

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

Smart Card Data Transfer Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

Bit	:	7	6	5	4	3	2	1	0
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select 1 and 0

0	0	ø clock
	1	ø/4 clock
1	0	ø/16 clock
	1	ø/64 clock

Multiprocessor Mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop Bit Length

0	1 stop bit
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character Length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it is not possible to choose between LSB-first or MSB-first mode.

Communication Mode

0	Asynchronous mode
1	Clocked synchronous mode

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sets the serial transfer bit rate

Note: For details, see section 12.2.8, Bit Rate Register (BRR)

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable 1 and 0

0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*1
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input

Notes: *1 Outputs a clock of the same frequency as the bit rate.
*2 Inputs a clock with a frequency 16 times the bit rate.

Transmit End Interrupt Enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (ERI) requests, and settingof the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

0	Transmission disabled
1	Transmission enabled

Receive Interrupt Enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit data empty interrupt (TXI) requests disabled
1	Transmit data empty interrupt (TXI) requests enabled


Bit : 7 6 5 4 3 2 1 0
Initial value : 1 1 1 1 1 1 1 1
R/W : R/W R/W R/W R/W R/W R/W R/W R/W
Stores data for serial transmission

Bit	:	7	6	5	4	3	2	1	0
		TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	:	1	0	0	0	0	1	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor Bit Transfer

0	Data with a 0 multiprocessor bit is transmitted
1	Data with a 1 multiprocessor bit is transmitted

Multiprocessor Bit

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Parity Error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR

Framing Error

0	[Clearing condition] When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0

Overrun Error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive Data Register Full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt request and reads data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit Data Register Empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

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

RDR1—Receive Data Register 1 H'FF85 SCI1

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R

Stores received serial data

SCMR1—Smart Card Mode Register 1 H'FF86 SCI1

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

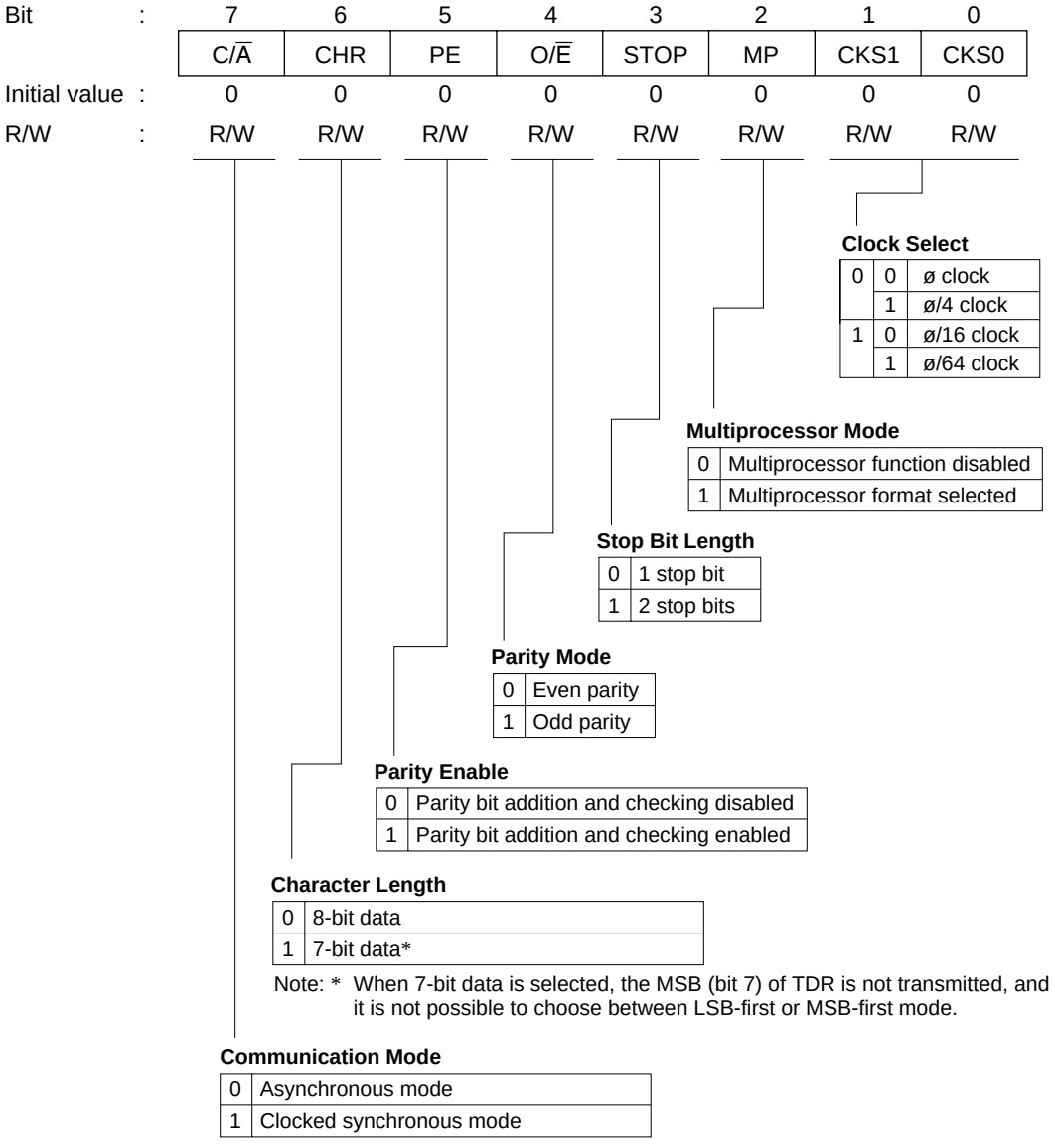
Reserved
Only 0 should be written to this bit

Smart Card Data Invert

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

Smart Card Data Transfer Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first


Bit : 7 6 5 4 3 2 1 0
C/A CHR PE O/E STOP MP CKS1 CKS0
Initial value : 0 0 0 0 0 0 0 0
R/W : R/W R/W R/W R/W R/W R/W R/W R/W
Clock Select
0 0 ø clock
1 ø/4 clock
1 0 ø/16 clock
1 ø/64 clock
Multiprocessor Mode
0 Multiprocessor function disabled
1 Multiprocessor format selected
Stop Bit Length
0 1 stop bit
1 2 stop bits
Parity Mode
0 Even parity
1 Odd parity
Parity Enable
0 Parity bit addition and checking disabled
1 Parity bit addition and checking enabled
Character Length
0 8-bit data
1 7-bit data*
Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it is not possible to choose between LSB-first or MSB-first mode.
Communication Mode
0 Asynchronous mode
1 Clocked synchronous mode

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sets the serial transfer bit rate

Note: For details, see section 12.2.8, Bit Rate Register (BRR)

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable 1 and 0

0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*1
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*2
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input

Notes: *1 Outputs a clock of the same frequency as the bit rate.
*2 Inputs a clock with a frequency 16 times the bit rate.

Transmit End Interrupt Enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (ERI) requests, and settingof the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

0	Transmission disabled
1	Transmission enabled

Receive Interrupt Enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit data empty interrupt (TXI) requests disabled
1	Transmit data empty interrupt (TXI) requests enabled

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores data for serial transmission

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
R/W	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor Bit Transfer

0	Data with a 0 multiprocessor bit is transmitted
1	Data with a 1 multiprocessor bit is transmitted

Multiprocessor Bit

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Parity Error

0	[Clearing condition] When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR

Framing Error

0	[Clearing condition] When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data when reception ends, and the stop bit is 0

Overrun Error

0	[Clearing condition] When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive Data Register Full

0	[Clearing conditions] • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt request and reads data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit Data Register Empty

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt request and writes data to TDR
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

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

RDR2—Receive Data Register 2 **H'FF8D** **SCI2**

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R

Stores received serial data

SCMR2—Smart Card Mode Register 2 **H'FF8E** **SCI2**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	—
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

Reserved
Only 0 should be written to this bit

Smart Card Data Invert

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

Smart Card Data Transfer Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

PORT1—Port 1 Register **H'FFB0** **Port 1**

Bit	:	7	6	5	4	3	2	1	0
		P17	P16	P15	P14	P13	P12	P11	P10
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

State of port 1 pins

Note: * Determined by the state of pins P17 to P10.

PORT3—Port 3 Register **H'FFB2** **Port 3**

Bit	:	7	6	5	4	3	2	1	0
		—	P36	P35	P34	P33	P32	P31	P30
Initial value	:	Undefined	—*	—*	—*	—*	—*	—*	—*
R/W	:	—	R	R	R	R	R	R	R

State of port 3 pins

Note: * Determined by the state of pins P36 to P30.

PORT4—Port 4 Register **H'FFB3** **Port 4**

Bit	:	7	6	5	4	3	2	1	0
		P47	P46	P45	P44	P43	P42	P41	P40
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

State of port 4 pins

Note: * Determined by the state of pins P47 to P40.

PORT7—Port 7 Register **H'FFB6** **Port 7**

Bit	:	7	6	5	4	3	2	1	0
		P77	P76	P75	P74	P73	P72	P71	P70
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

State of port 7 pins

Note: * Determined by the state of pins P77 to P70.

PORT9—Port 9 Register **H'FFB8** **Port 9**

Bit	:	7	6	5	4	3	2	1	0
		—	P96	—	—	—	—	—	—
Initial value	:	—	—*	—	—	—	—	—	—
R/W	:	R	R	R	R	R	R	R	R

State of pin P96

Note: * Determined by the state of pin P96.

PORTA—Port A Register **H'FFB9** **Port A**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3	PA2	PA1	PA0
Initial value	:	Undefined	Undefined	Undefined	Undefined	—*	—*	—*	—*
R/W	:	—	—	—	—	R	R	R	R

State of port A pins

Note: * Determined by the state of pins PA3 to PA0.

PORTB—Port B Register **H'FFBA** **Port B**

Bit	:	7	6	5	4	3	2	1	0
		PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

State of port B pins

Note: * Determined by the state of pins PB7 to PB0.

PORTC—Port C Register **H'FFBB** **Port C**

Bit	:	7	6	5	4	3	2	1	0
		PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

State of port C pins

Note: * Determined by the state of pins PC7 to PC0.

PORTD—Port D Register **H'FFBC** **Port D**

Bit	:	7	6	5	4	3	2	1	0
		PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

State of port D pins

Note: * Determined by the state of pins PD7 to PD0.

PORTE—Port E Register **H'FFBD** **Port E**

Bit	:	7	6	5	4	3	2	1	0
		PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

State of port E pins

Note: * Determined by the state of pins PE7 to PE0.

PORTF—Port F Register **H'FFBE** **Port F**

Bit	:	7	6	5	4	3	2	1	0
		PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

State of port F pins

Note: * Determined by the state of pins PF7 to PF0.

PORTG—Port G Register **H'FFBF** **Port G**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	PG4	PG3	PG2	PG1	PG0
Initial value	:	Undefined	Undefined	Undefined	—*	—*	—*	—*	—*
R/W	:	—	—	—	R	R	R	R	R

State of port G pins

Note: * Determined by the state of pins PG4 to PG0.

C.1 Port 1 Block Diagrams

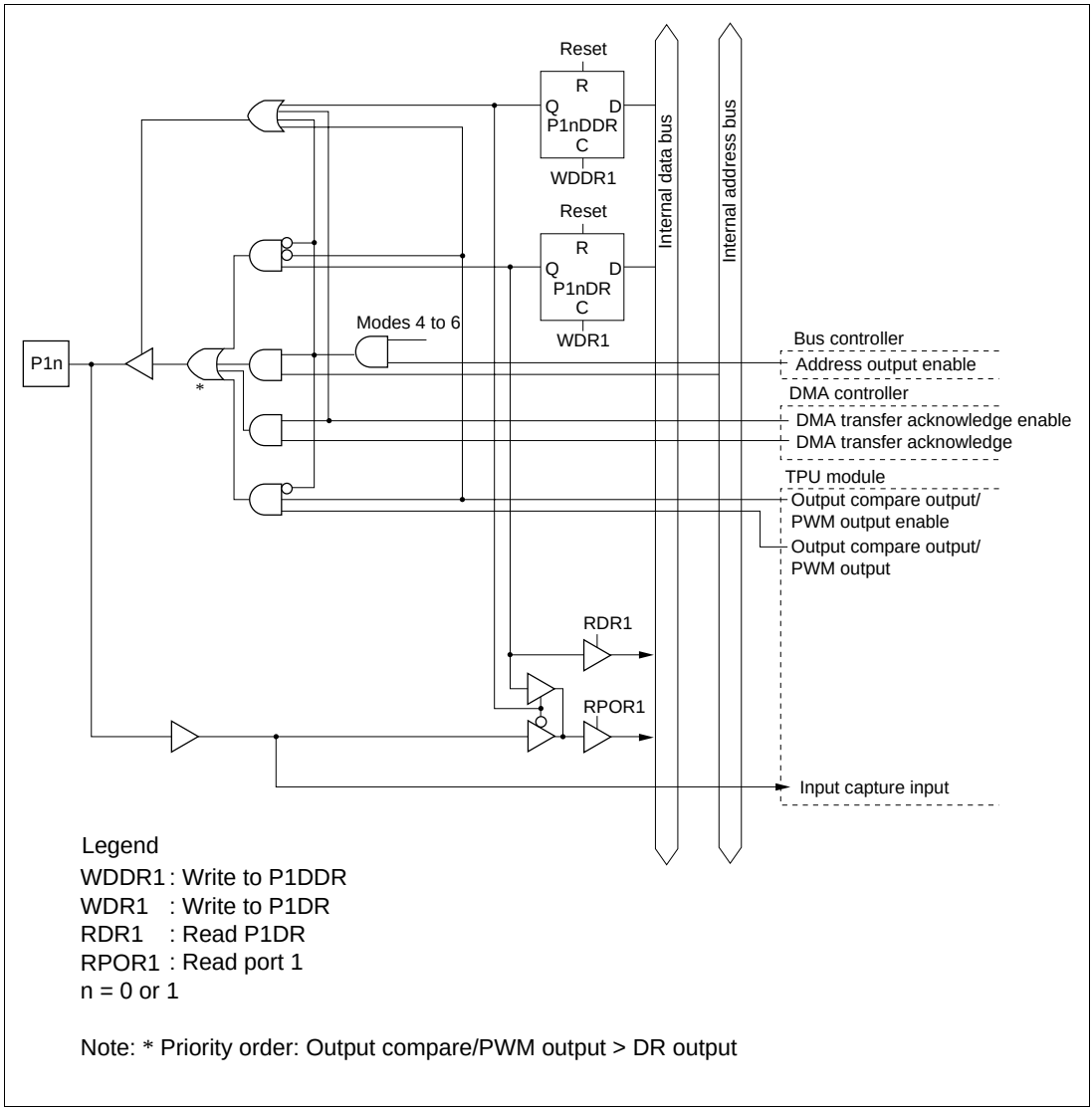


Figure C-1 Port 1 Block Diagram (Pins P10 and P11)

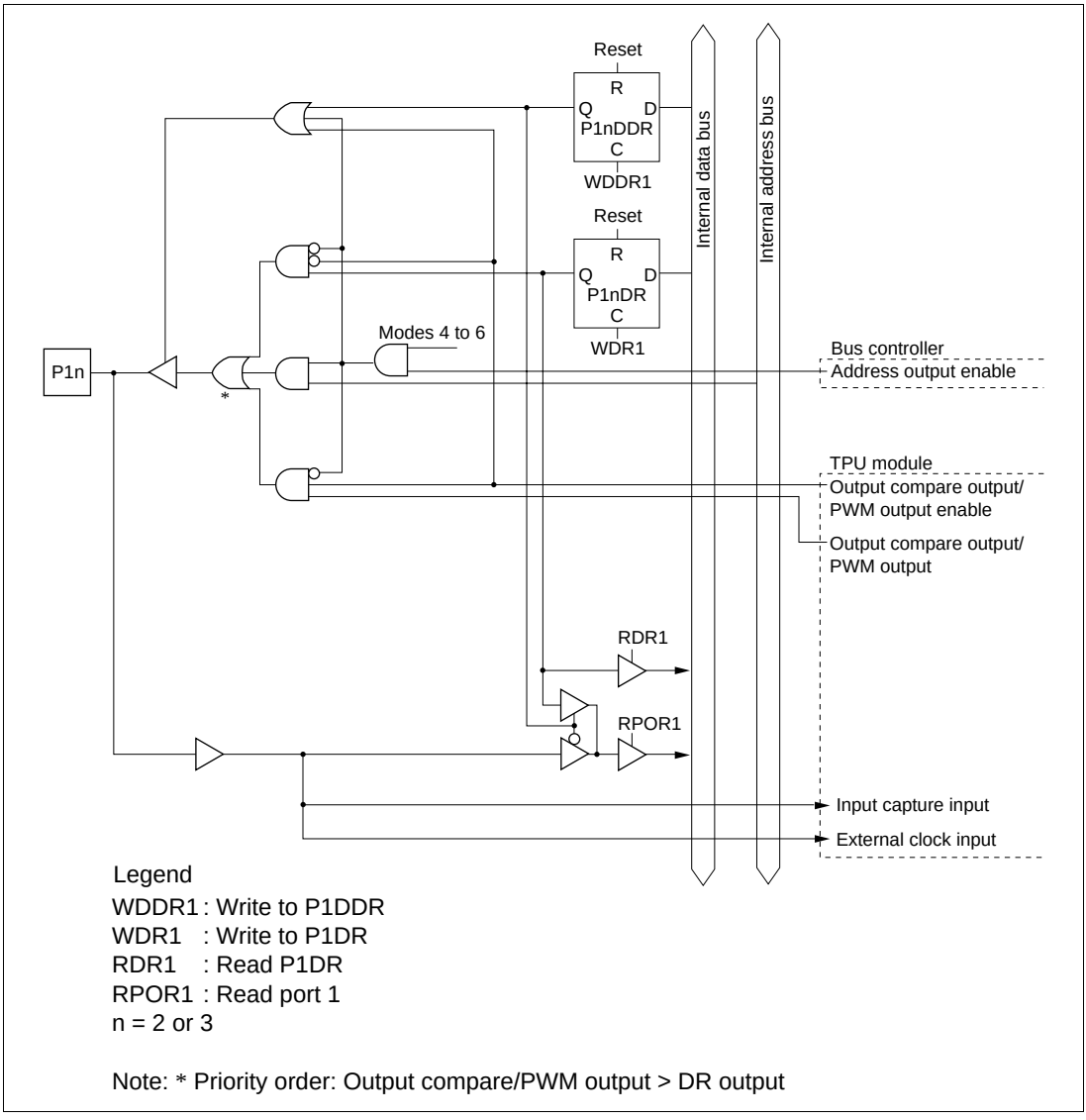


Figure C-2 Port 1 Block Diagram (Pins P12 and P13)

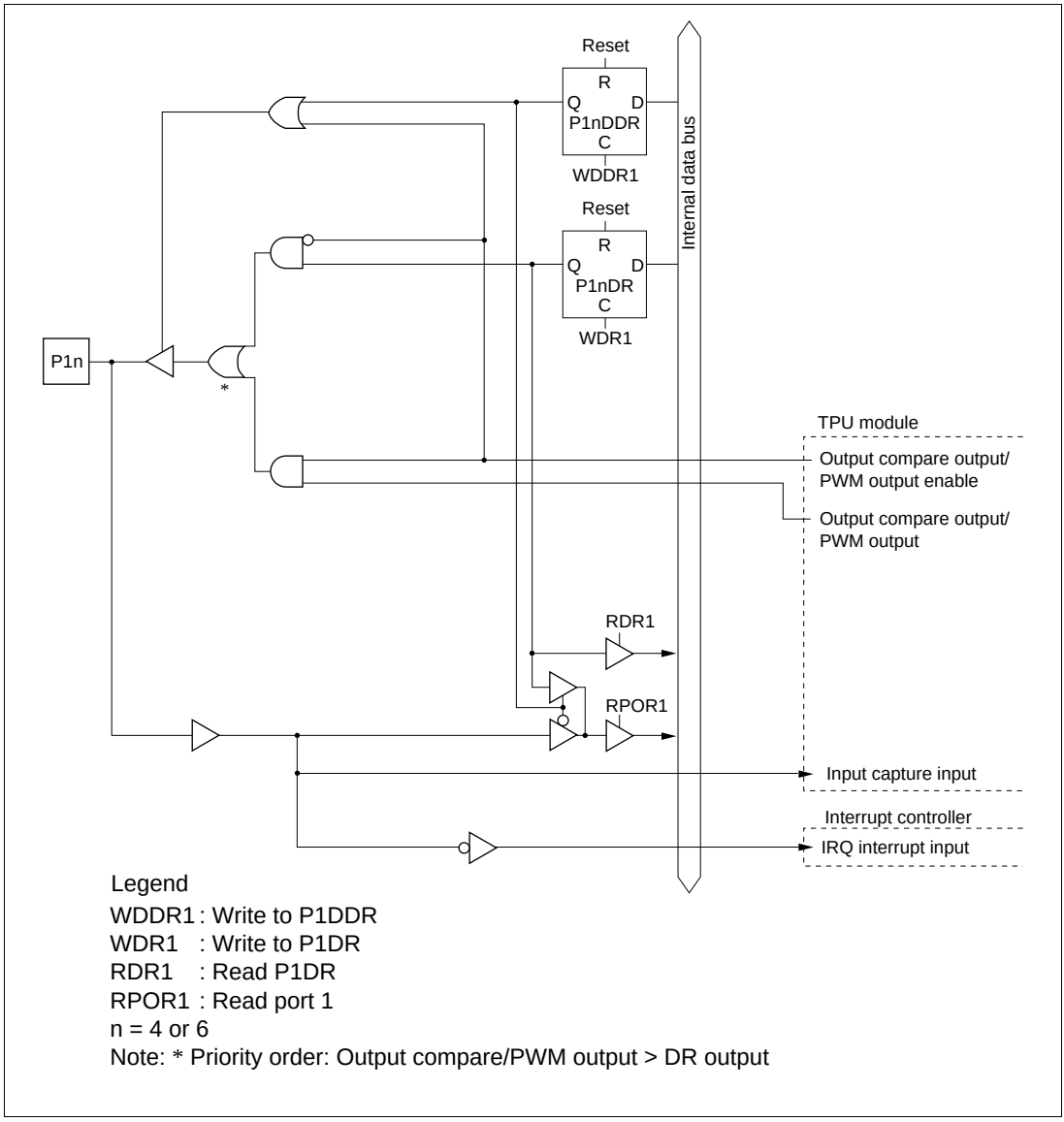


Figure C-3 Port 1 Block Diagram (Pins P14 and P16)

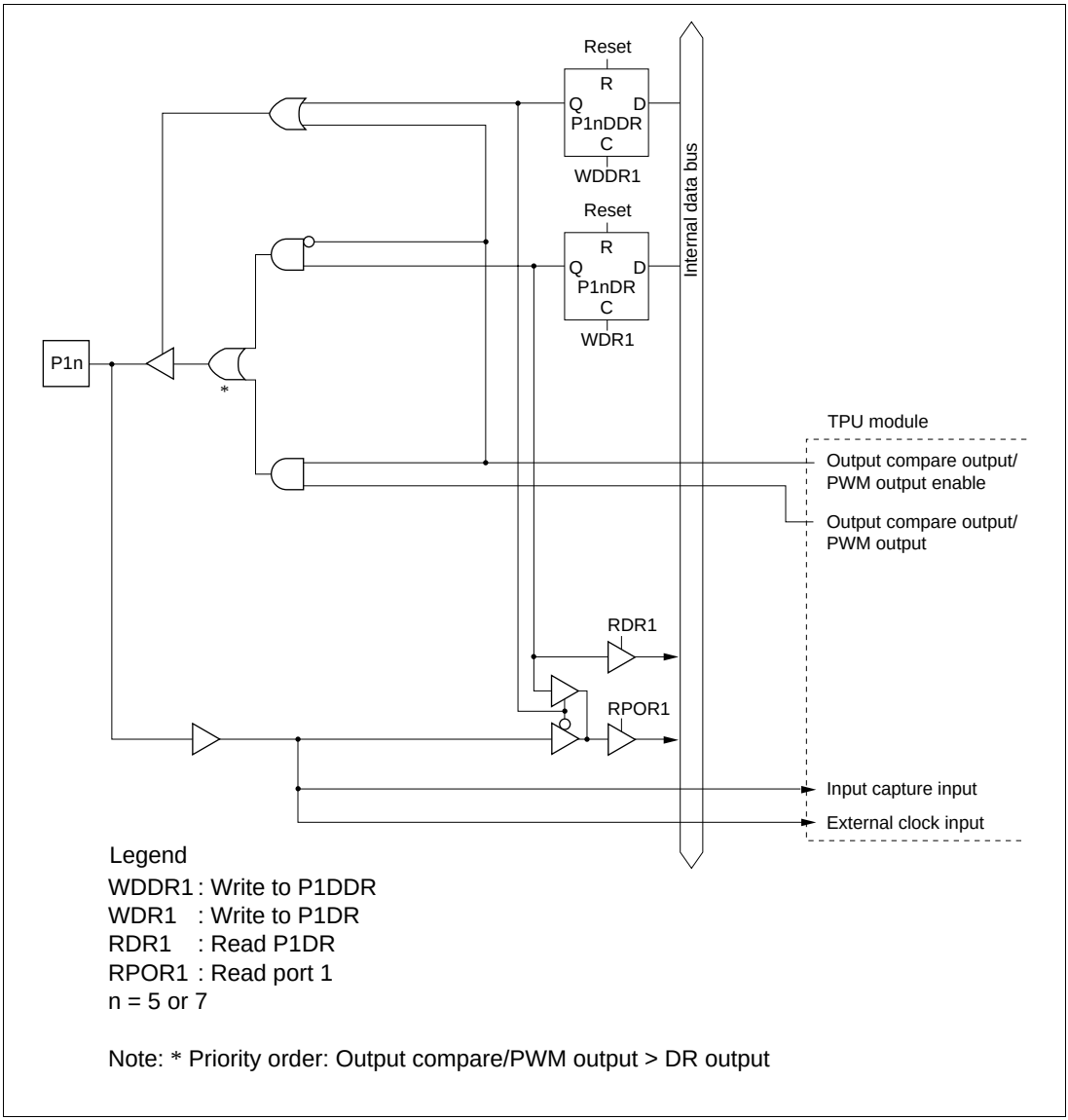


Figure C-4 Port 1 Block Diagram (Pins P15 and P17)

C.2 Port 3 Block Diagrams

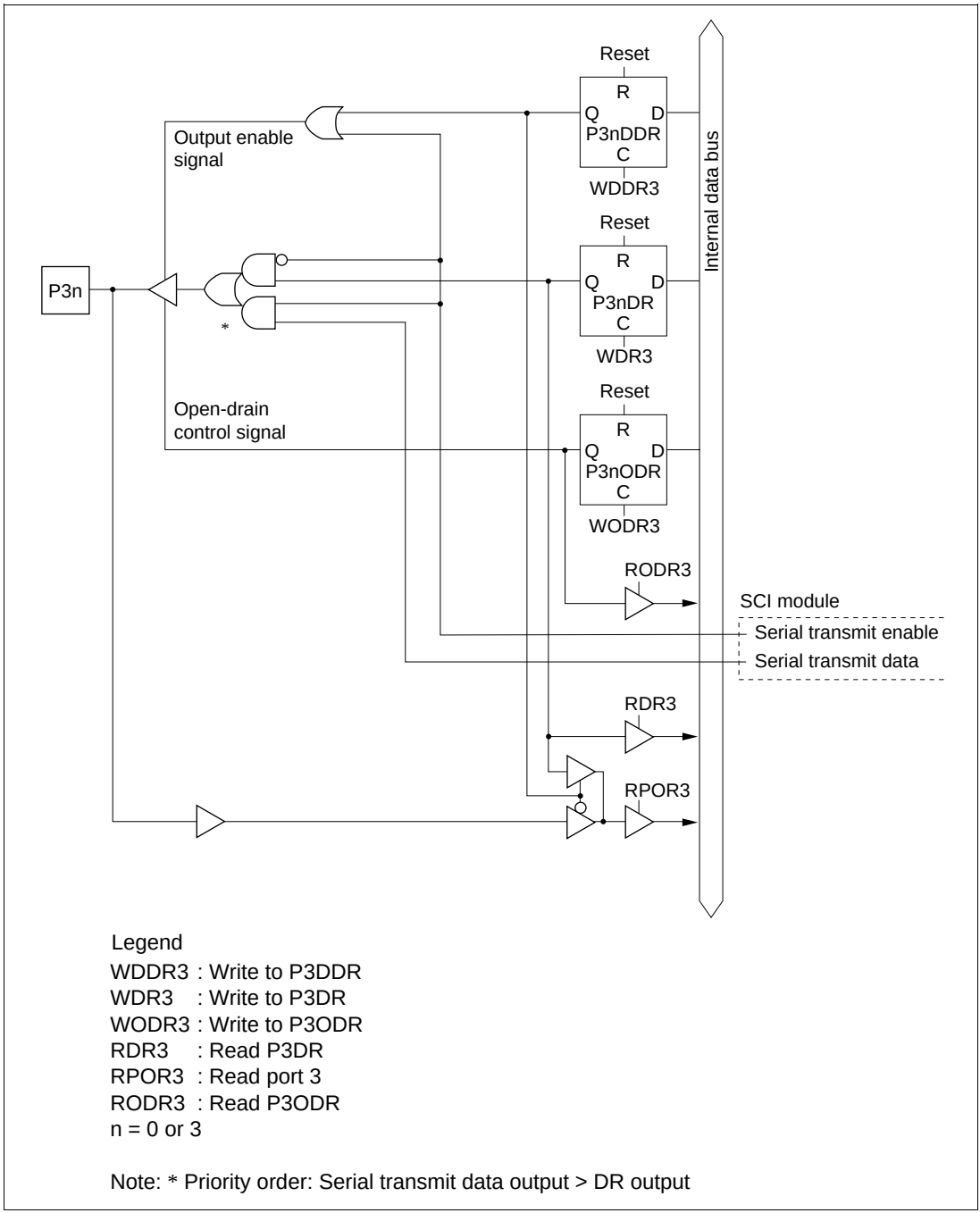
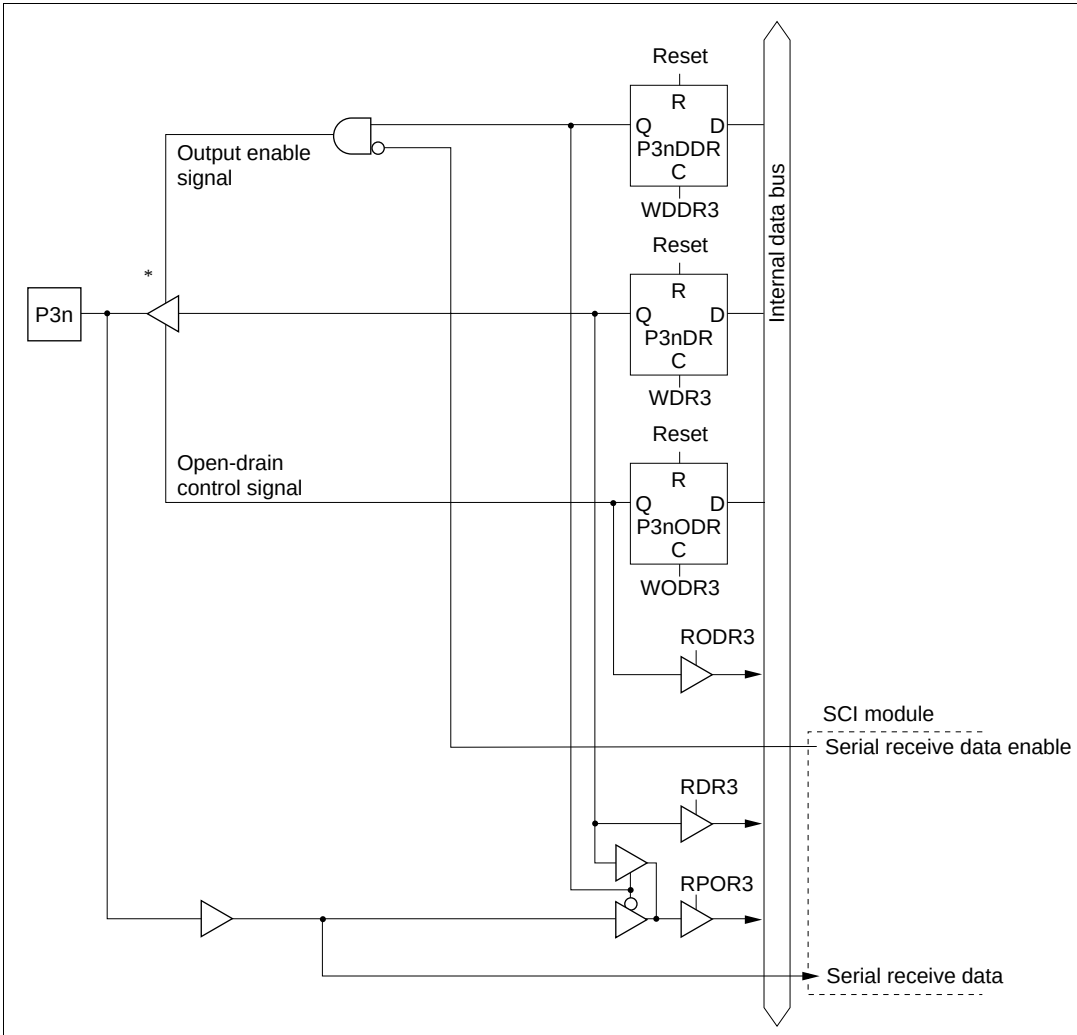


Figure C-5 Port 3 Block Diagram (Pins P30 and P33)

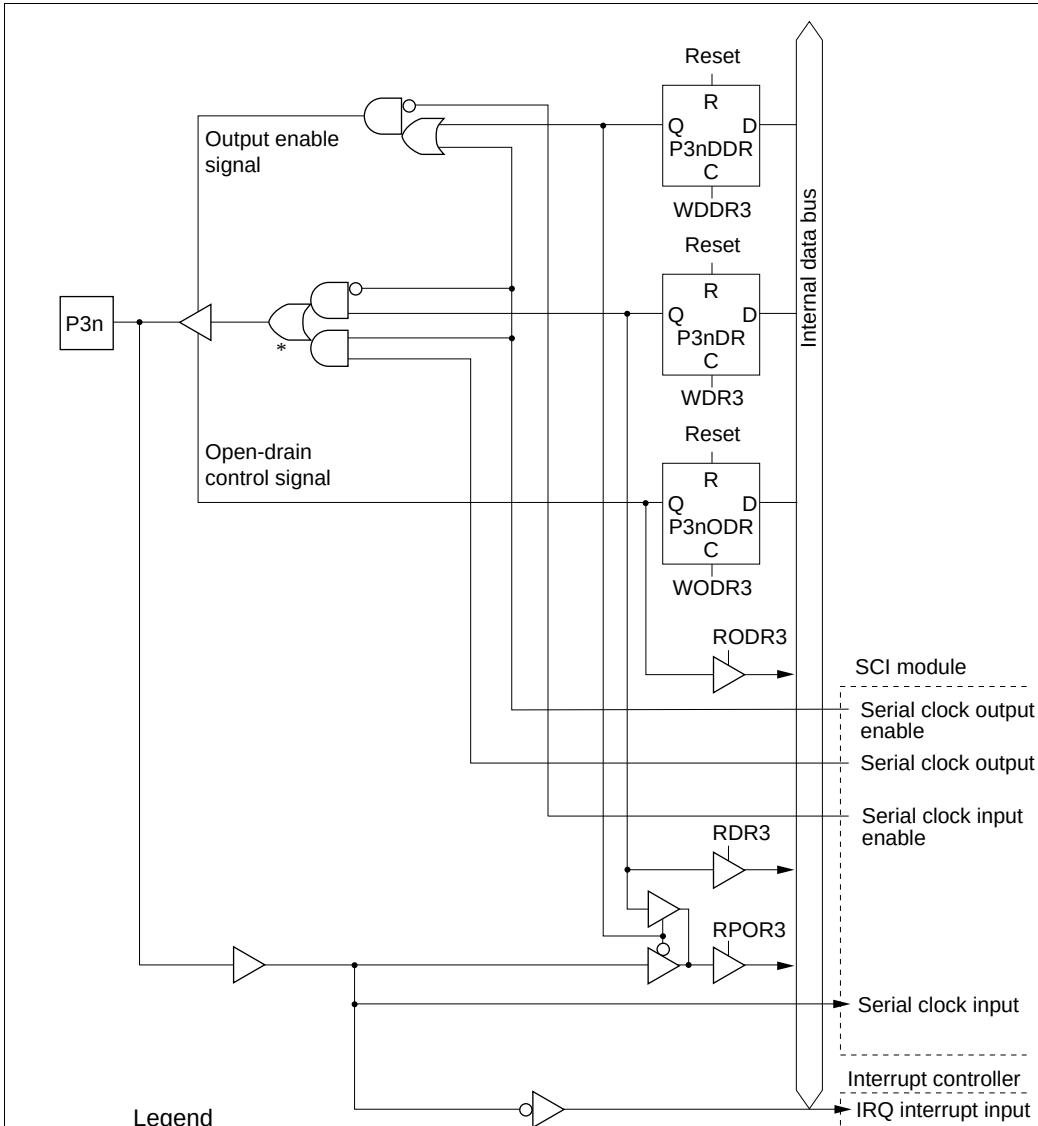


Legend

WDDR3 : Write to P3DDR
WDR3 : Write to P3DR
WODR3 : Write to P3ODR
RDR3 : Read P3DR
RPOR3 : Read port 3
RODR3 : Read P3ODR
n = 1 or 4

Note: * Priority order: Serial receive data input > DR output

Figure C-6 Port 3 Block Diagram (Pins P31 and P34)



Legend

WDDR3 : Write to P3DDR
WDR3 : Write to P3DR
WODR3 : Write to P3ODR
RDR3 : Read P3DR
RPOR3 : Read port 3
RODR3 : Read P3ODR
n = 2 or 5

Note: * Priority order: Serial clock input > Serial clock output > DR output

Figure C-7 Port 3 Block Diagram (Pins P32 and P35)

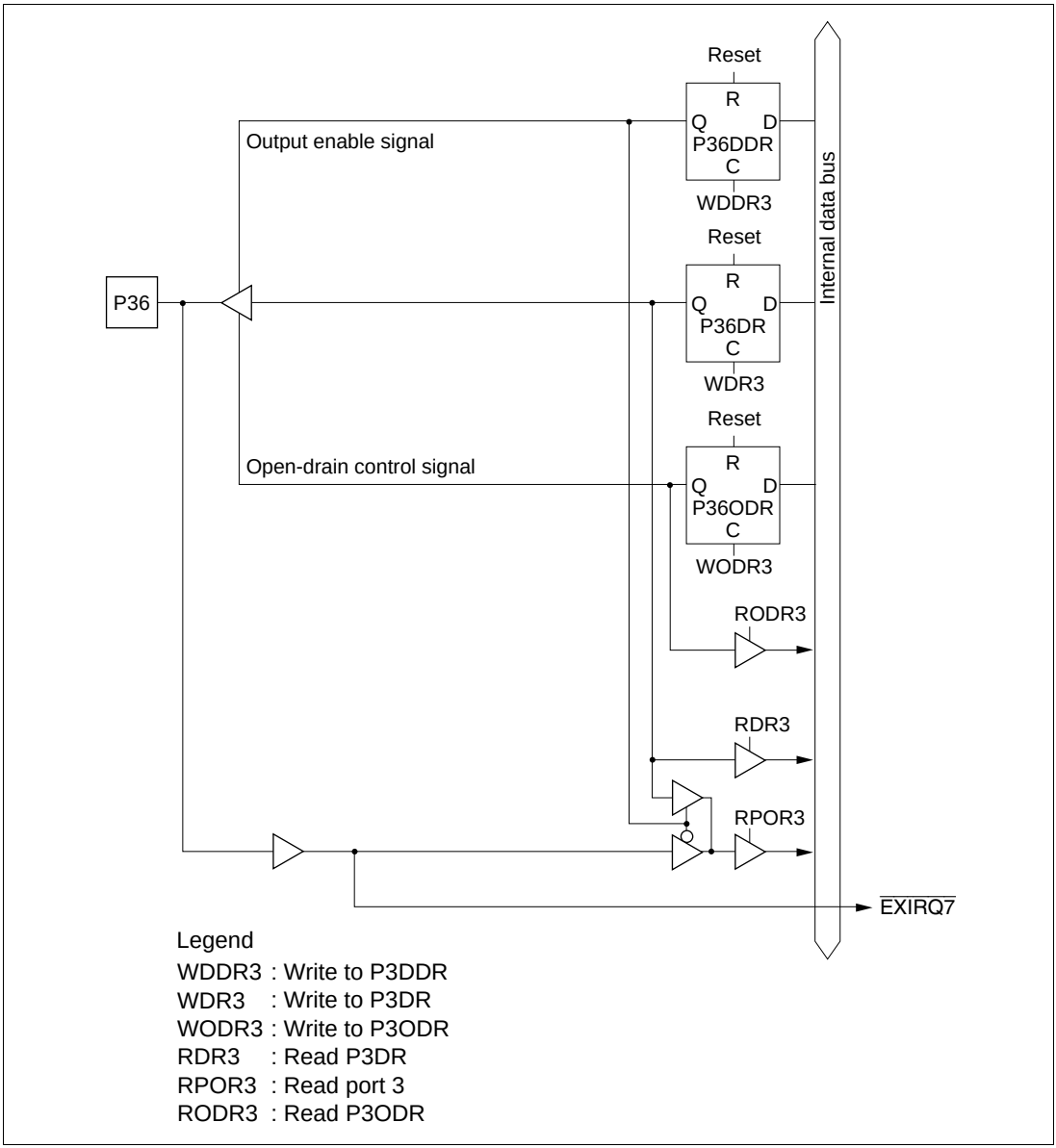


Figure C-8 Port 3 Block Diagram (Pin P36)

C.3 Port 4 Block Diagram

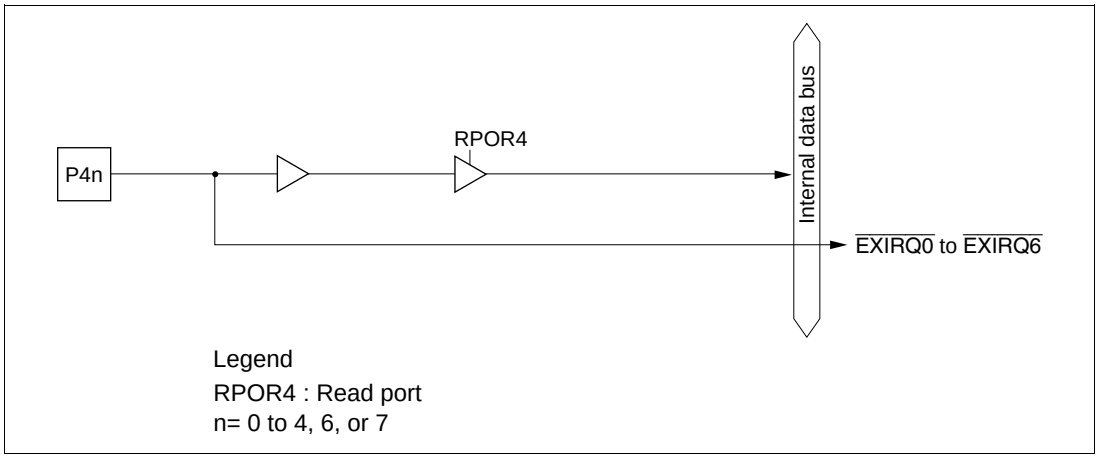


Figure C-9 Port 4 Block Diagram (Pins P40 to P44, P46, and P47)

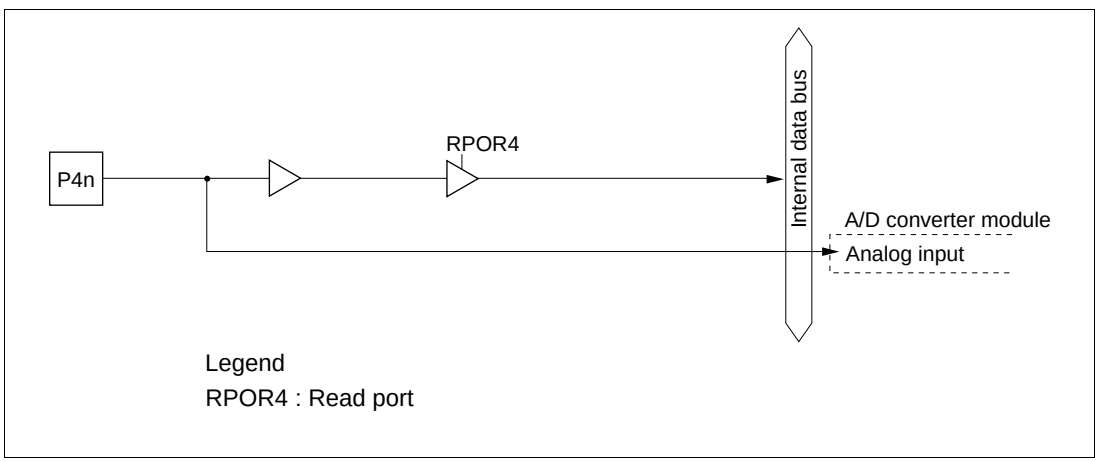


Figure C-10 Port 4 Block Diagram (Pin P45)

C.4 Port 7 Block Diagrams

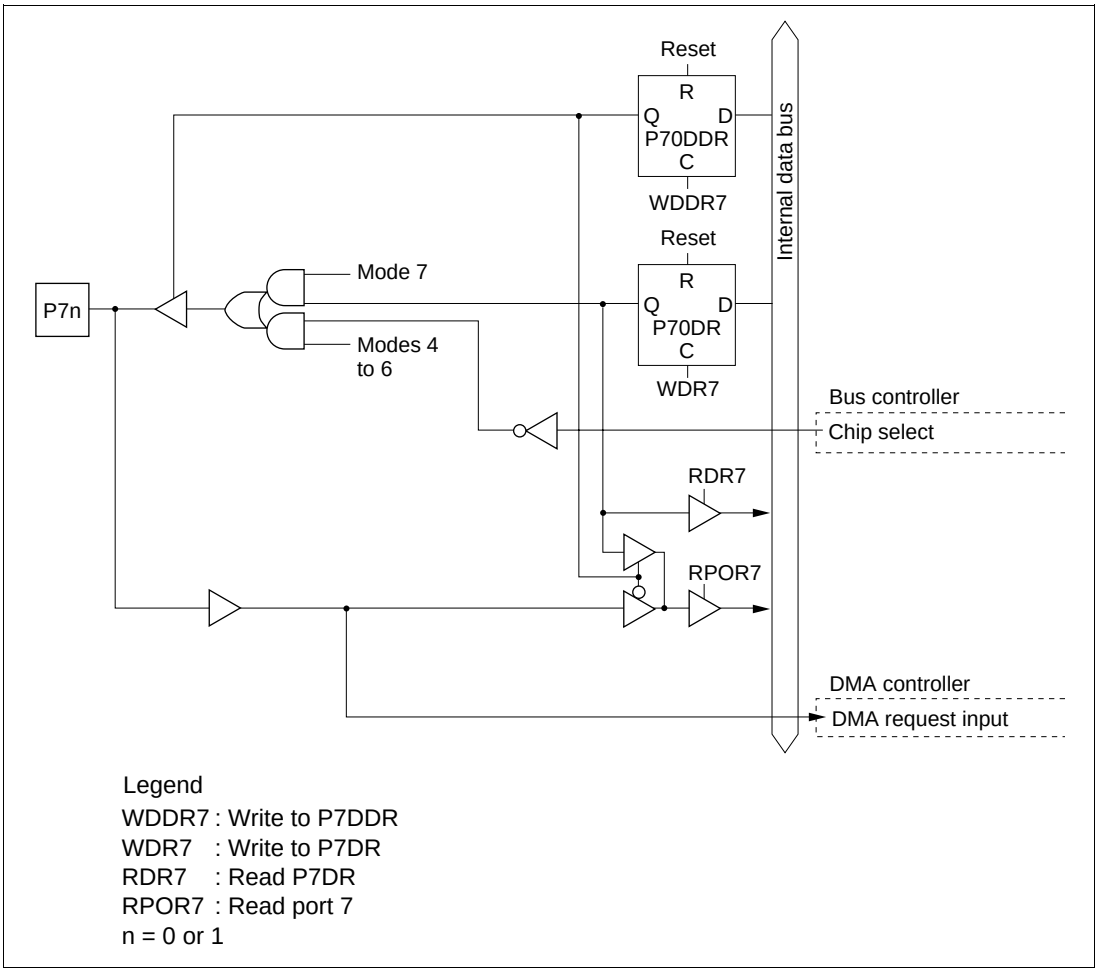


Figure C-11 Port 7 Block Diagram (Pins P70 and P71)

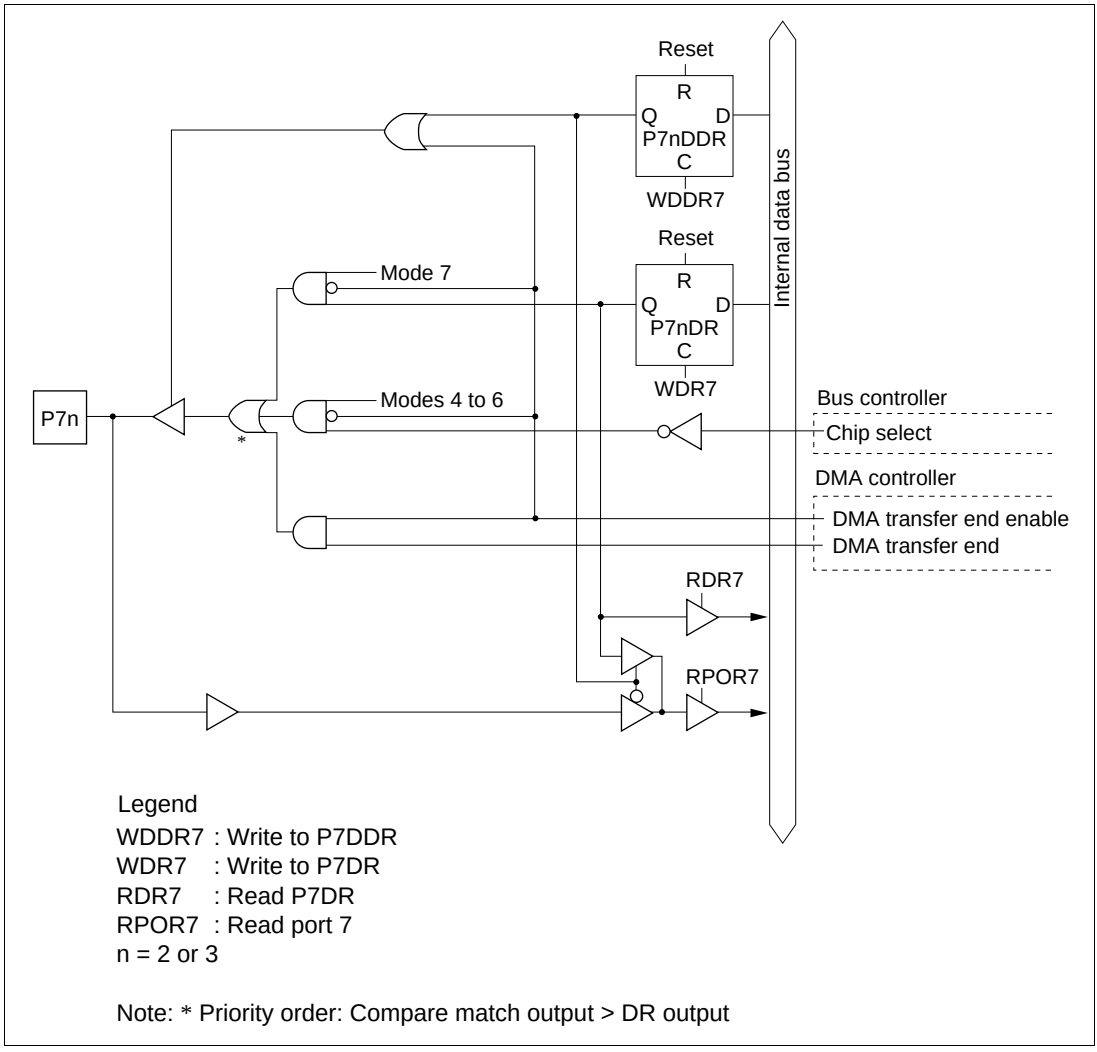


Figure C-12 Port 7 Block Diagram (Pins P72 and P73)

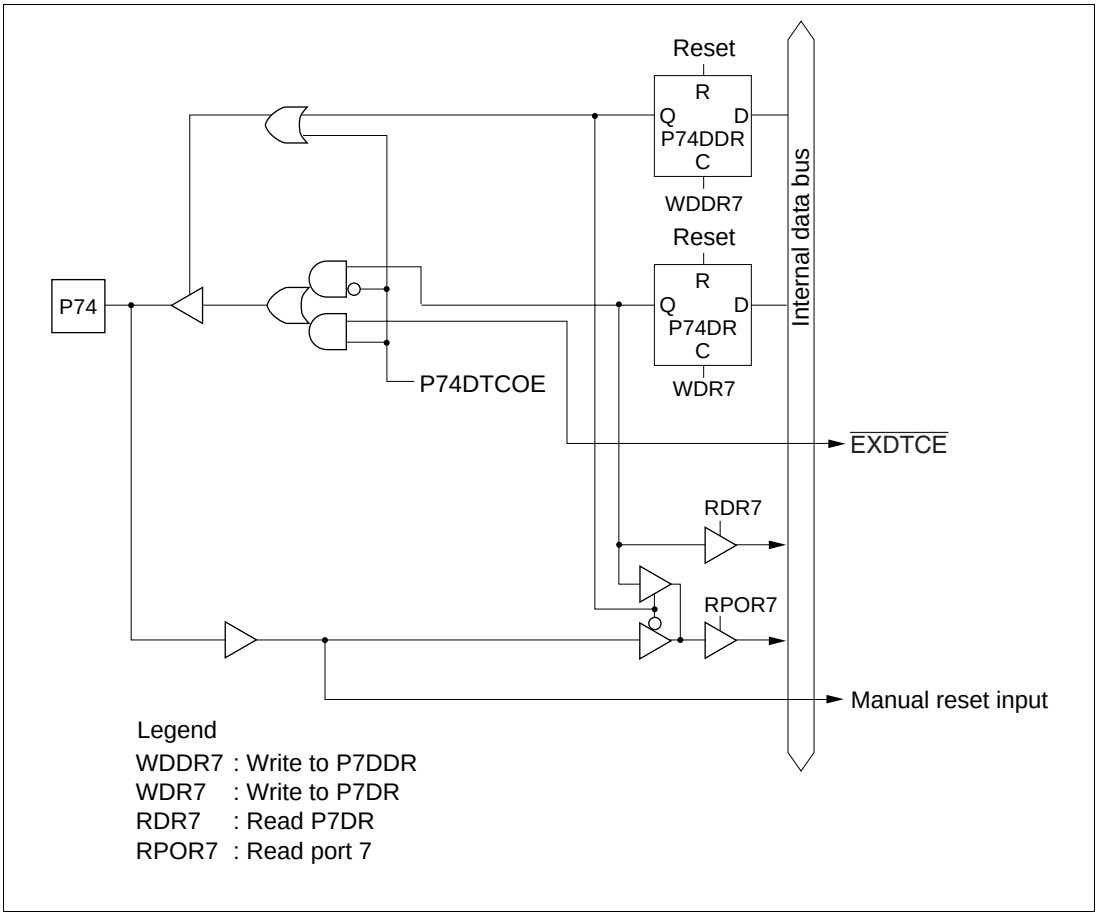


Figure C-13 Port 7 Block Diagram (Pin P74)

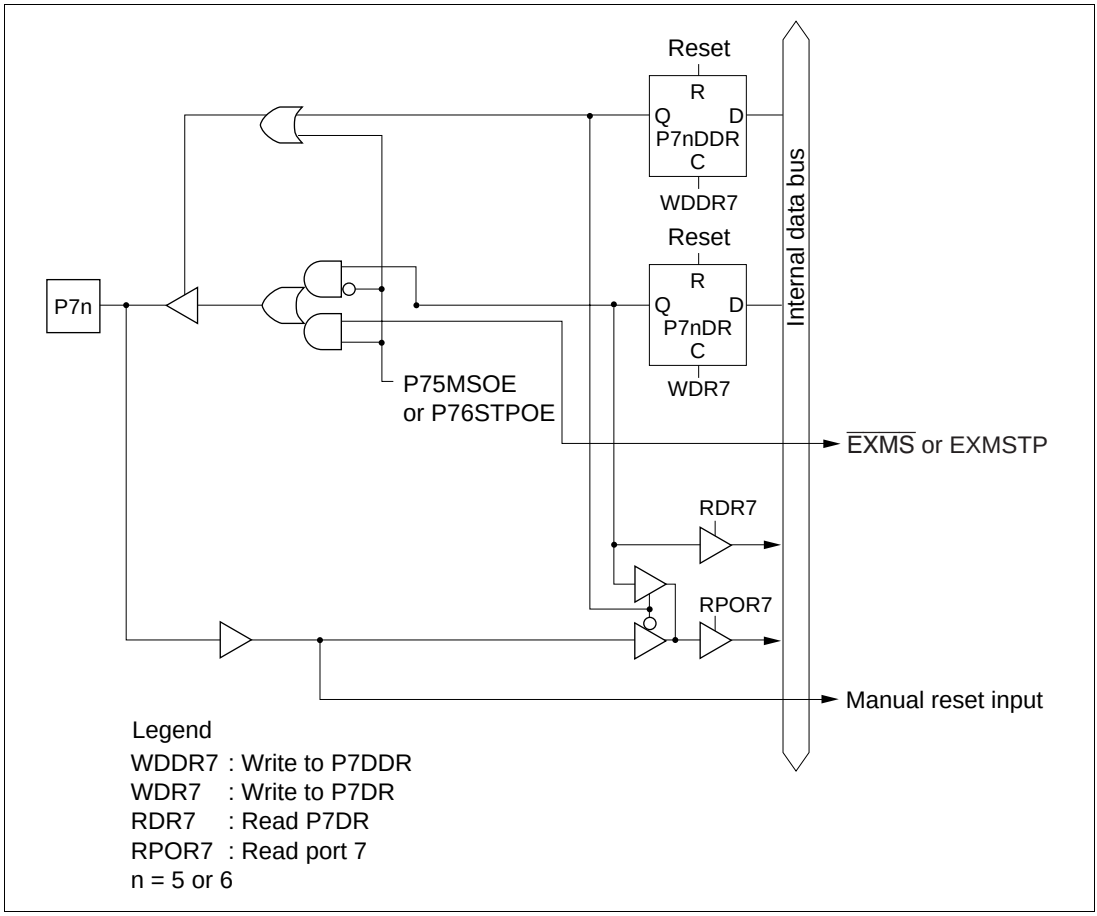


Figure C-14 Port 7 Block Diagram (Pins P75 and P76)

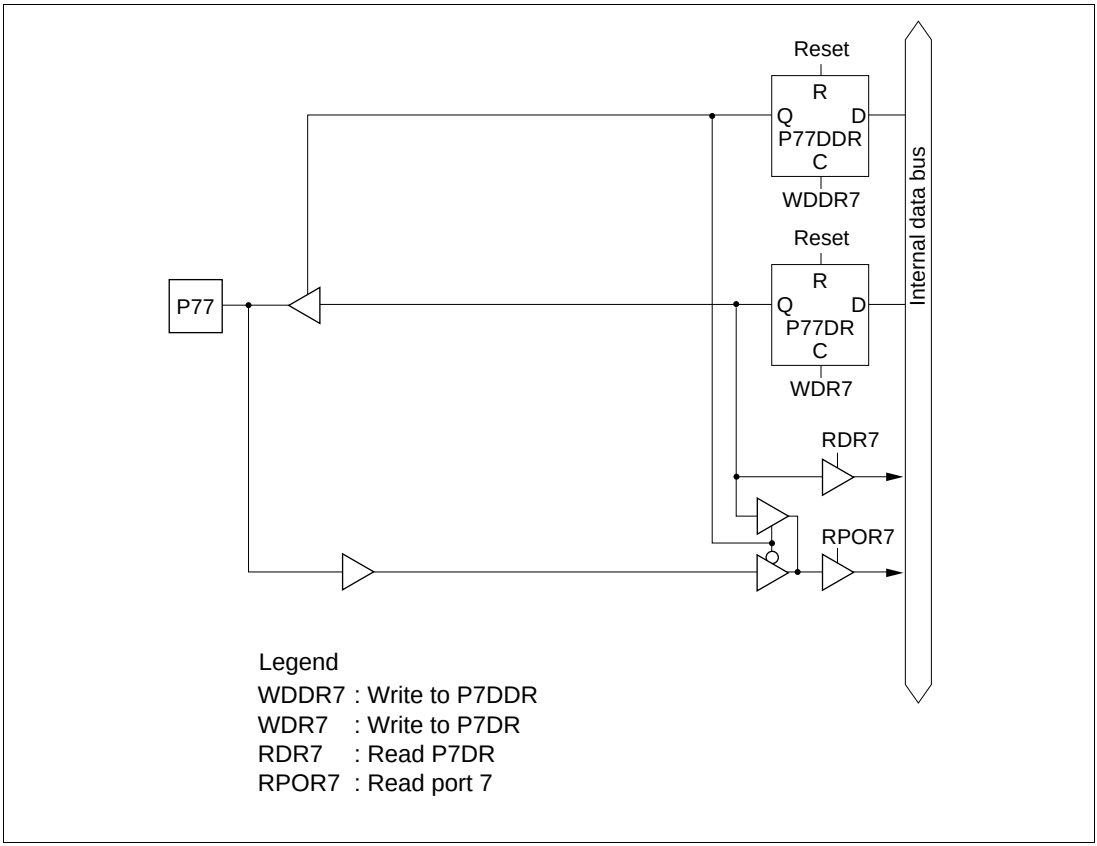


Figure C-15 Port 7 Block Diagram (Pin P77)

C.5 Port 9 Block Diagram

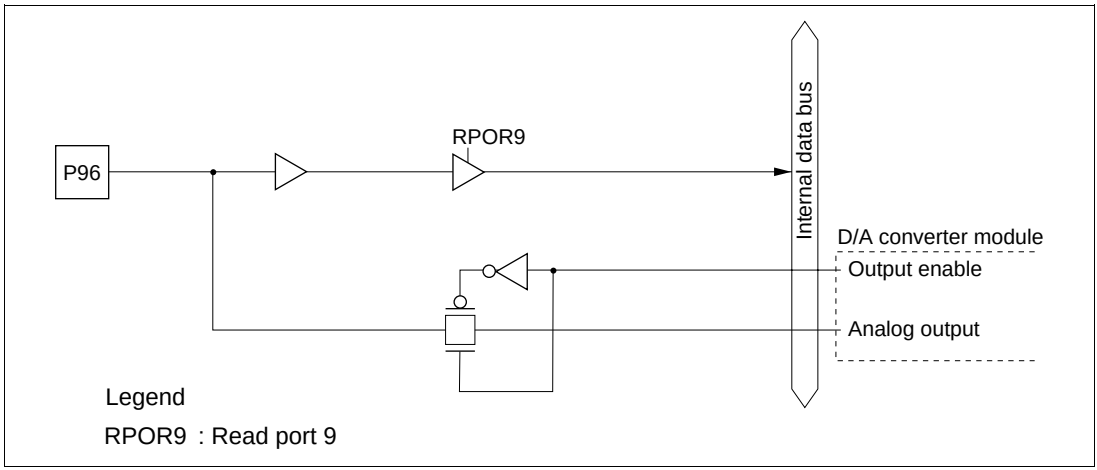
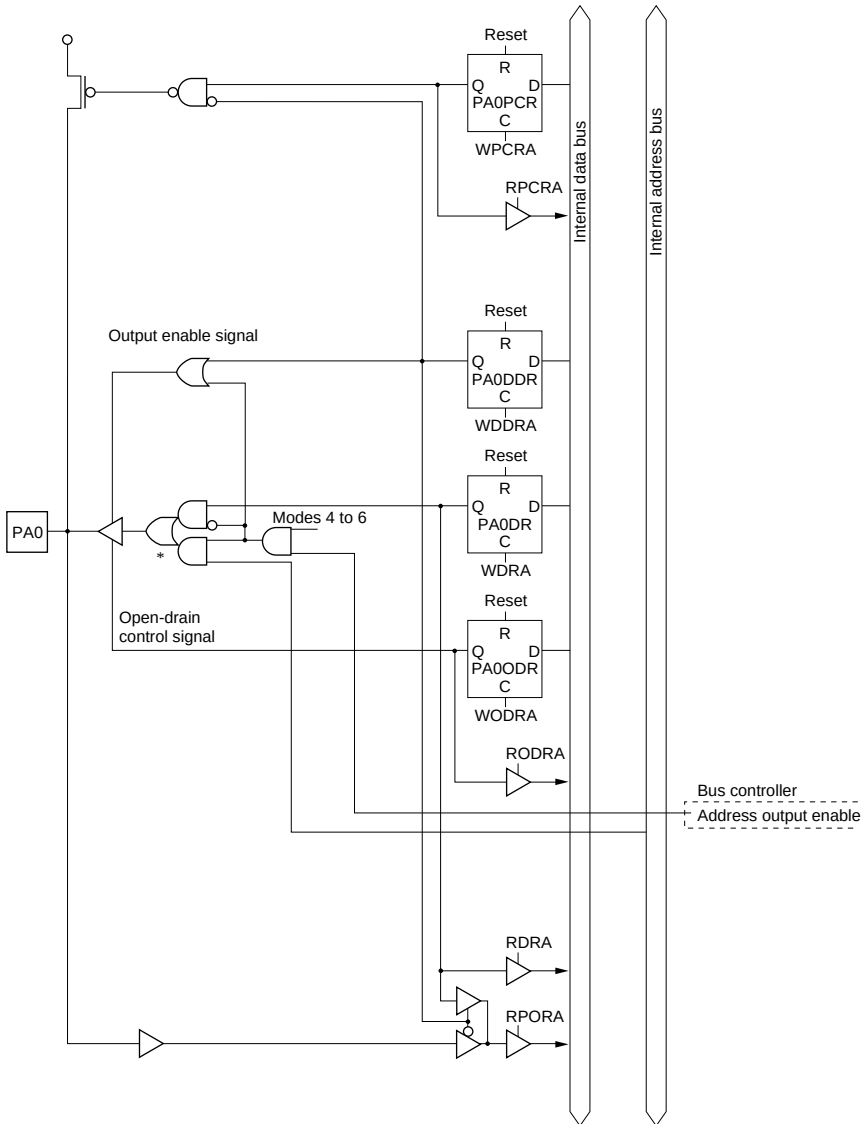


Figure C-16 Port 9 Block Diagram (Pin P96)

C.6 Port A Block Diagrams

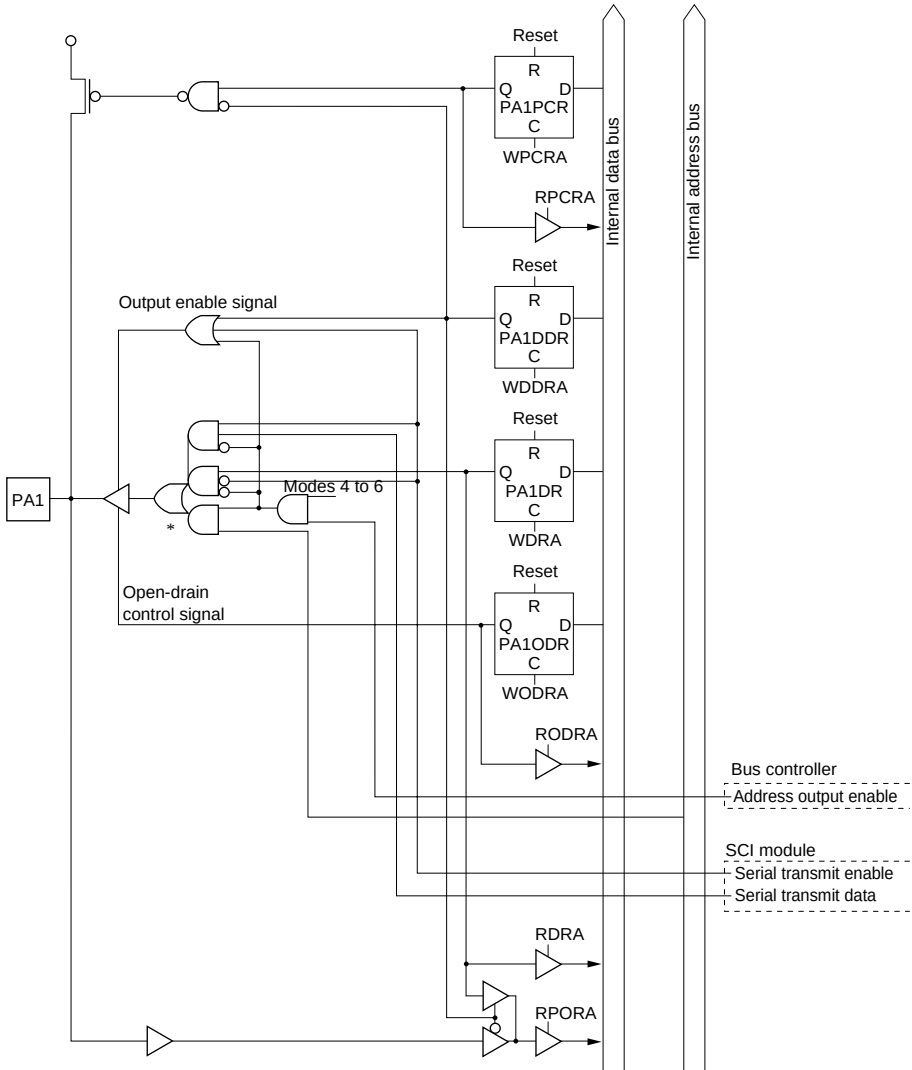


Legend

WDDRA: Write to PADDR	RDRA : Read PADR
WDRA : Write to PADR	RPORA: Read port A
WODRA: Write to PAODR	RODRA: Read PAODR
WPCRA: Write to PAPCR	RPCRA : Read PAPCR

Note: * Priority order: Address output > DR output

Figure C-17 Port A Block Diagram (Pin PA0)



Legend

WDDRA : Write to PADDR
WDRA : Write to PADR
WODRA : Write to PAODR
WPCRA : Write to PAPCR
RDRA : Read PADR
RPORA : Read port A
RODRA : Read PAODR
RPCRA : Read PAPCR

Note: * Priority order: Address output > Serial transmit data > DR output

Figure C-18 Port A Block Diagram (Pin PA1)

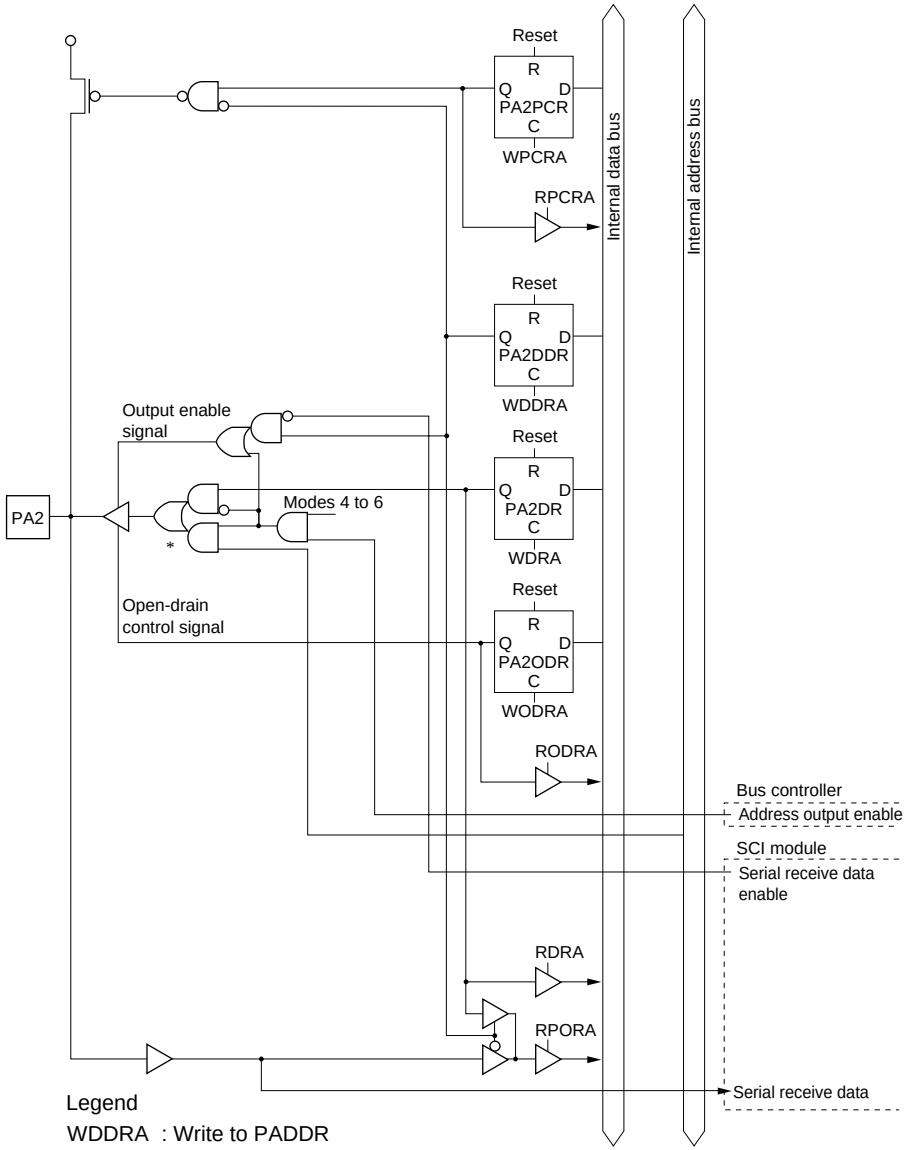


Figure C-19 Port A Block Diagram (Pin PA2)

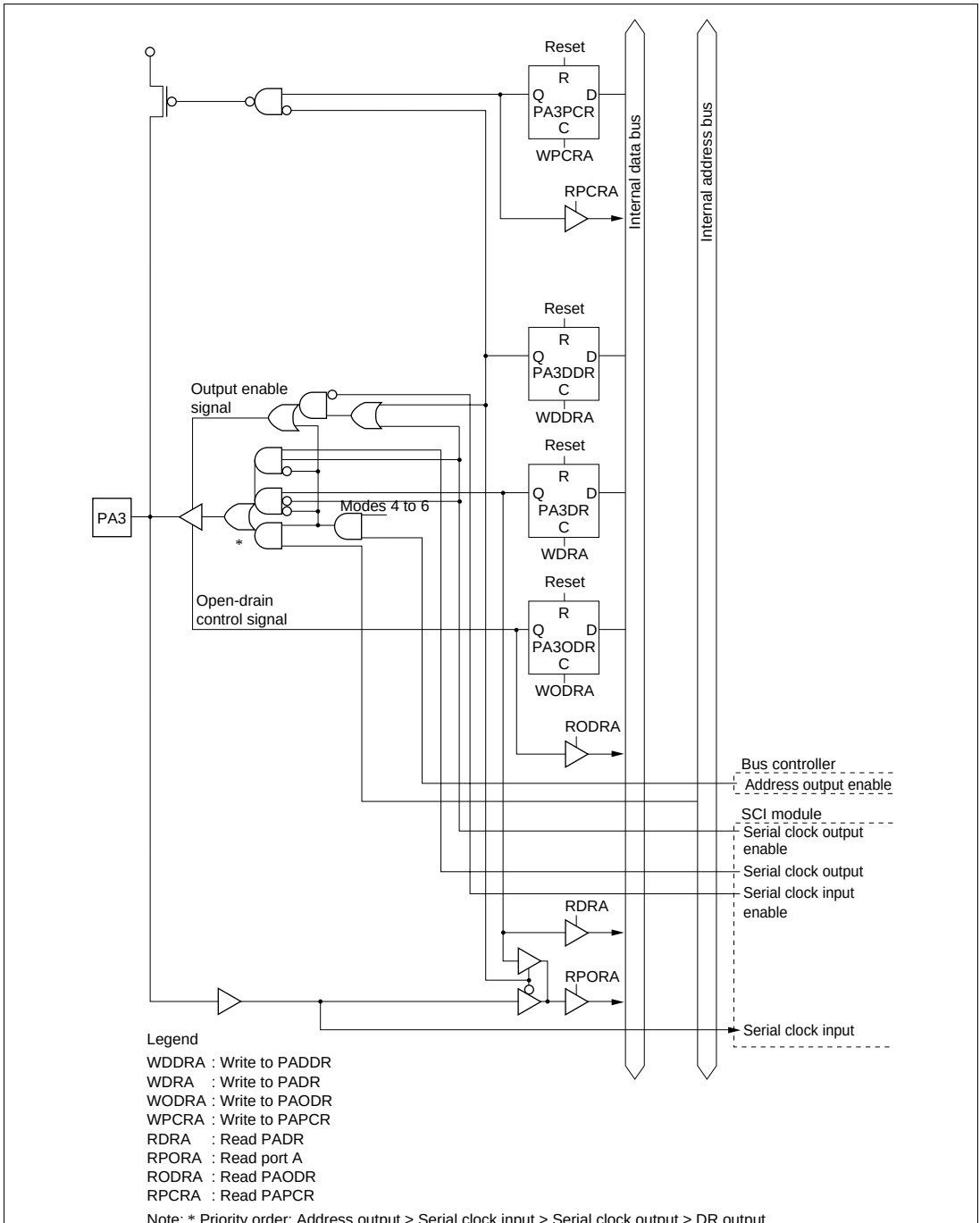
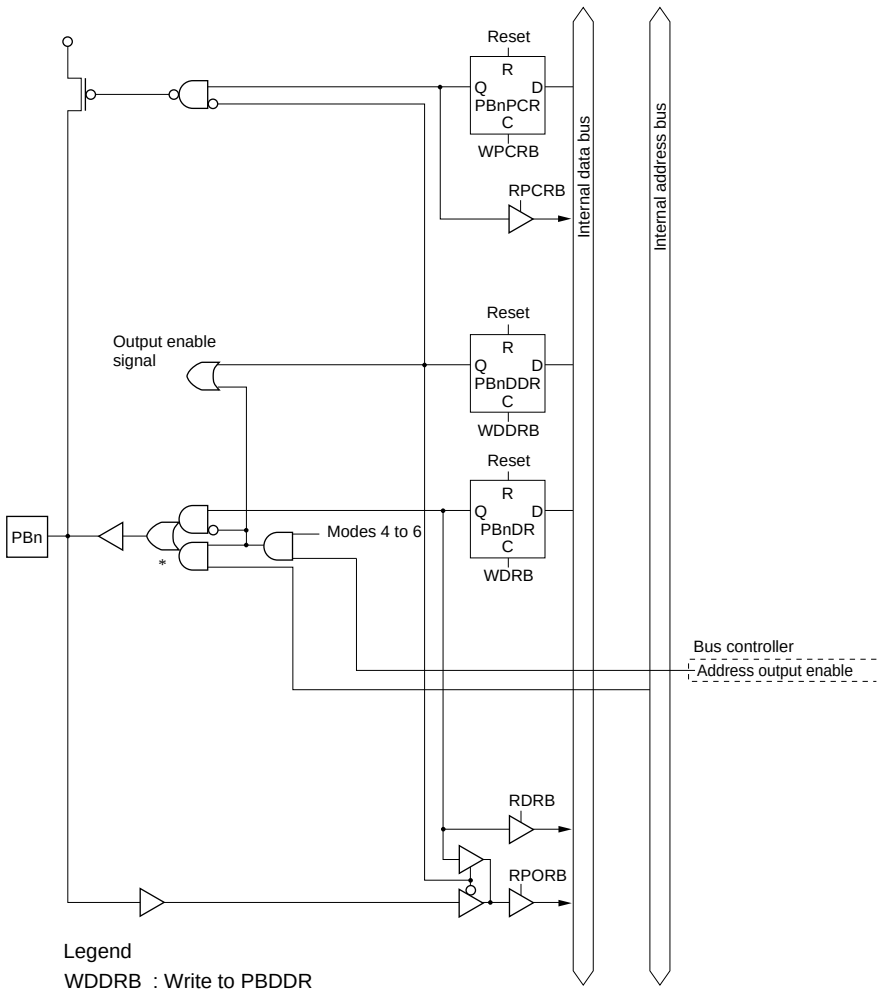


Figure C-20 Port A Block Diagram (Pin PA3)

C.7 Port B Block Diagram



Legend

WDDRB : Write to PBDDR
WDRB : Write to PBDR
WPCRB : Write to PBPCR
RDRB : Read PBDR
RPORB : Read port B
RPCRB : Read PBPCR
n = 0 to 7

Note: * Priority order: Address output > DR output

Figure C-21 Port B Block Diagram (Pins PB0 to PB7)

C.8 Port C Block Diagram

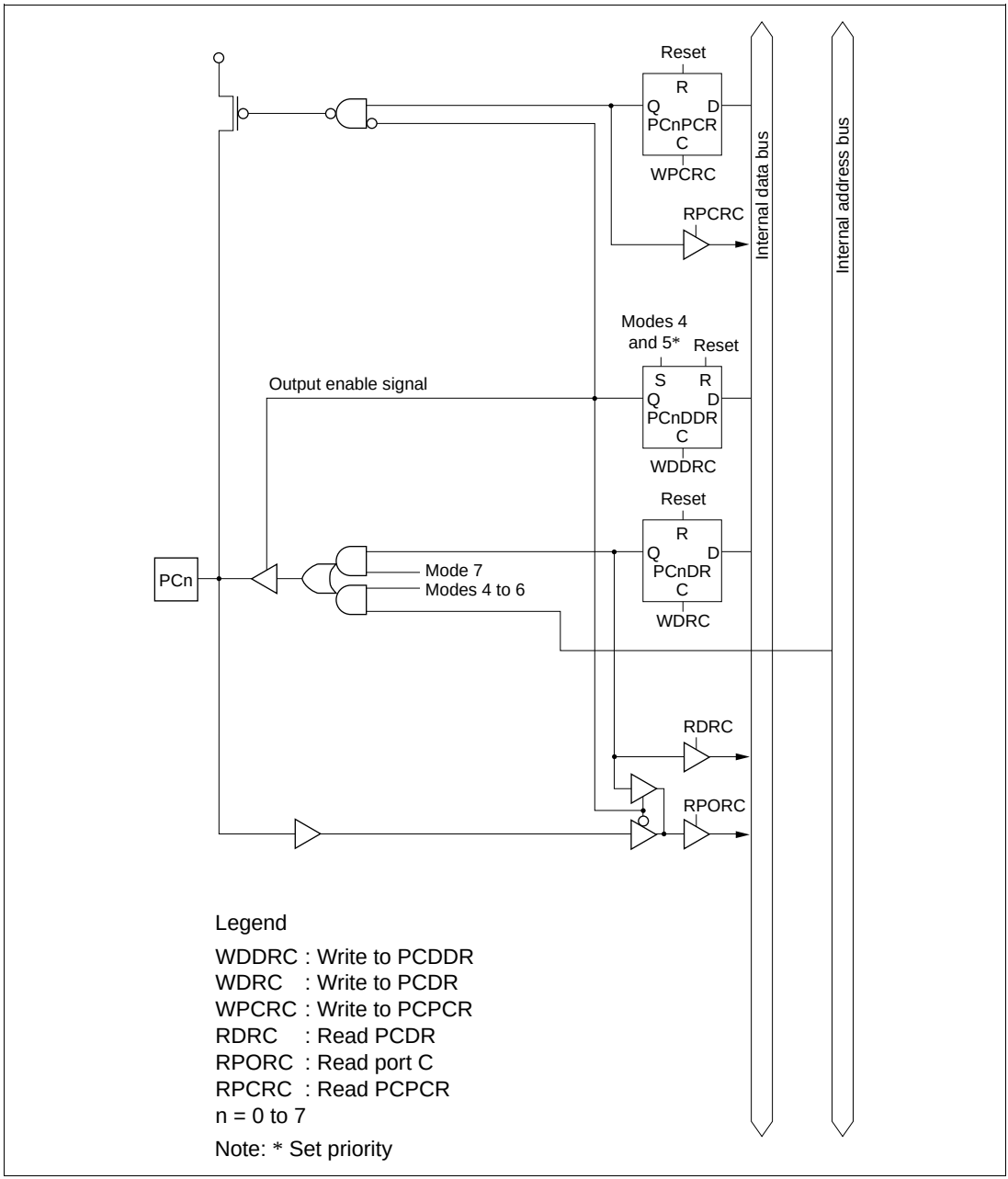


Figure C-22 Port C Block Diagram (Pins PC0 to PC7)

C.9 Port D Block Diagram

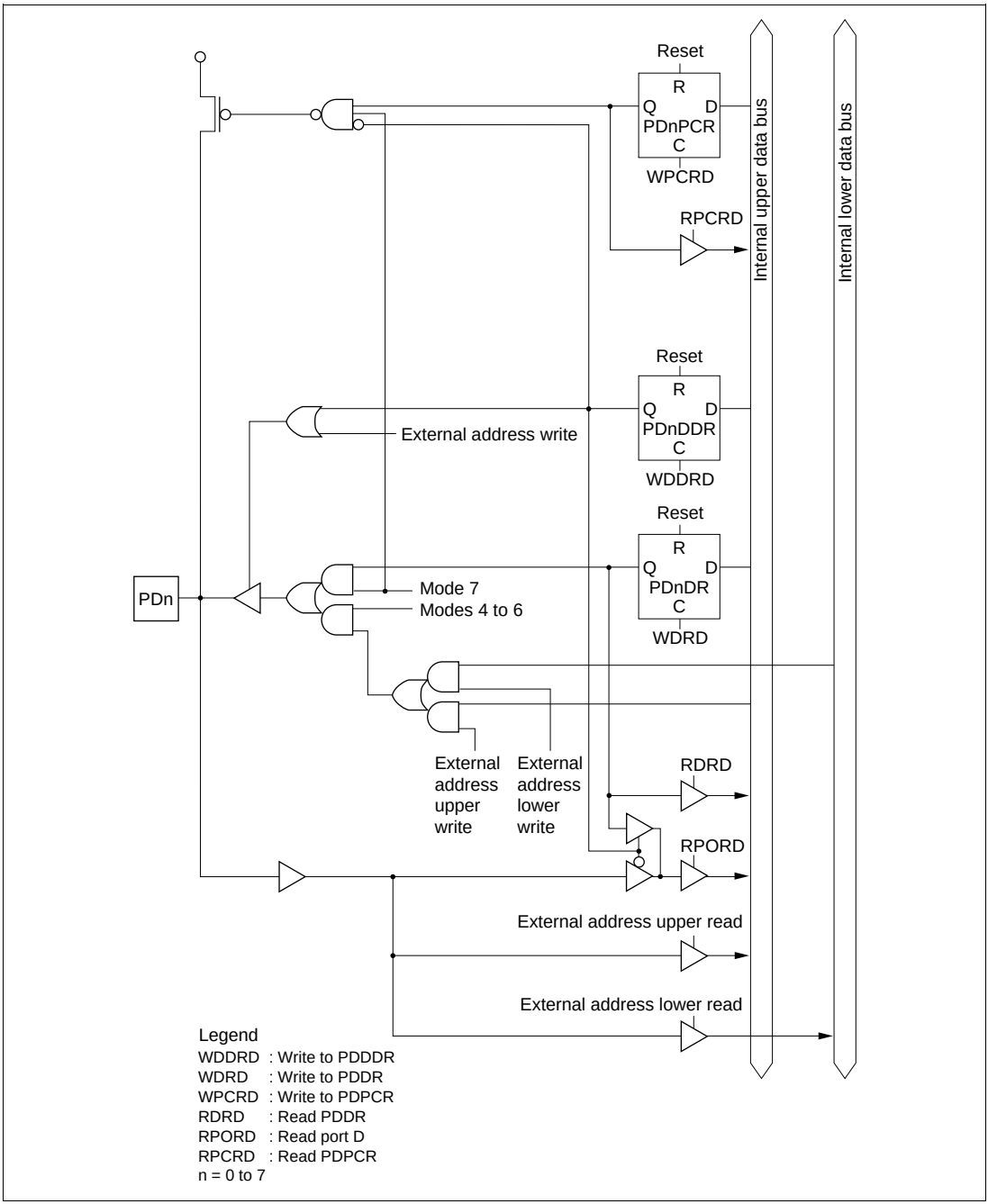


Figure C-23 Port D Block Diagram (Pins PD0 to PD7)

C.10 Port E Block Diagram

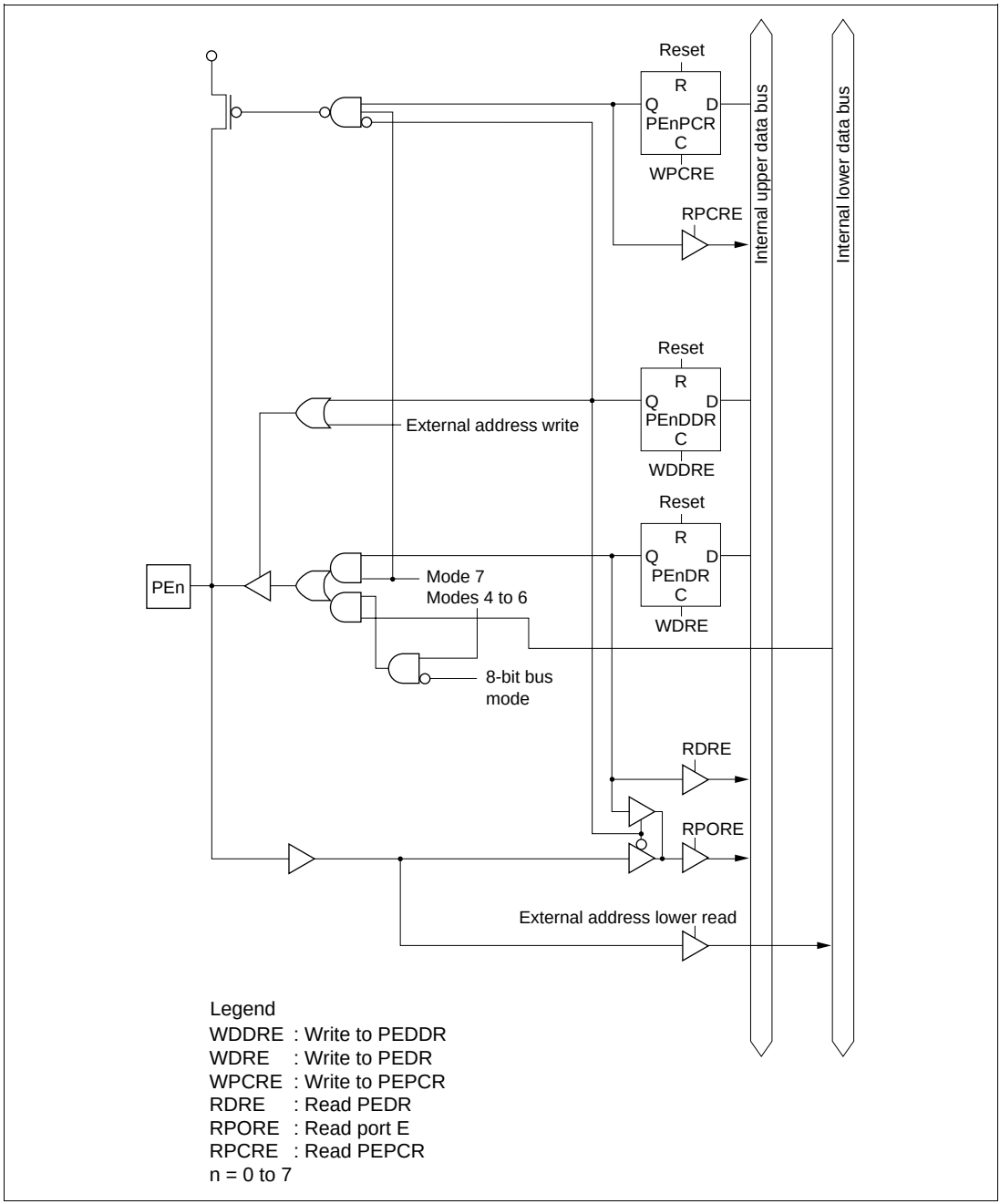


Figure C-24 Port E Block Diagram (Pins PE0 to PE7)

C.11 Port F Block Diagrams

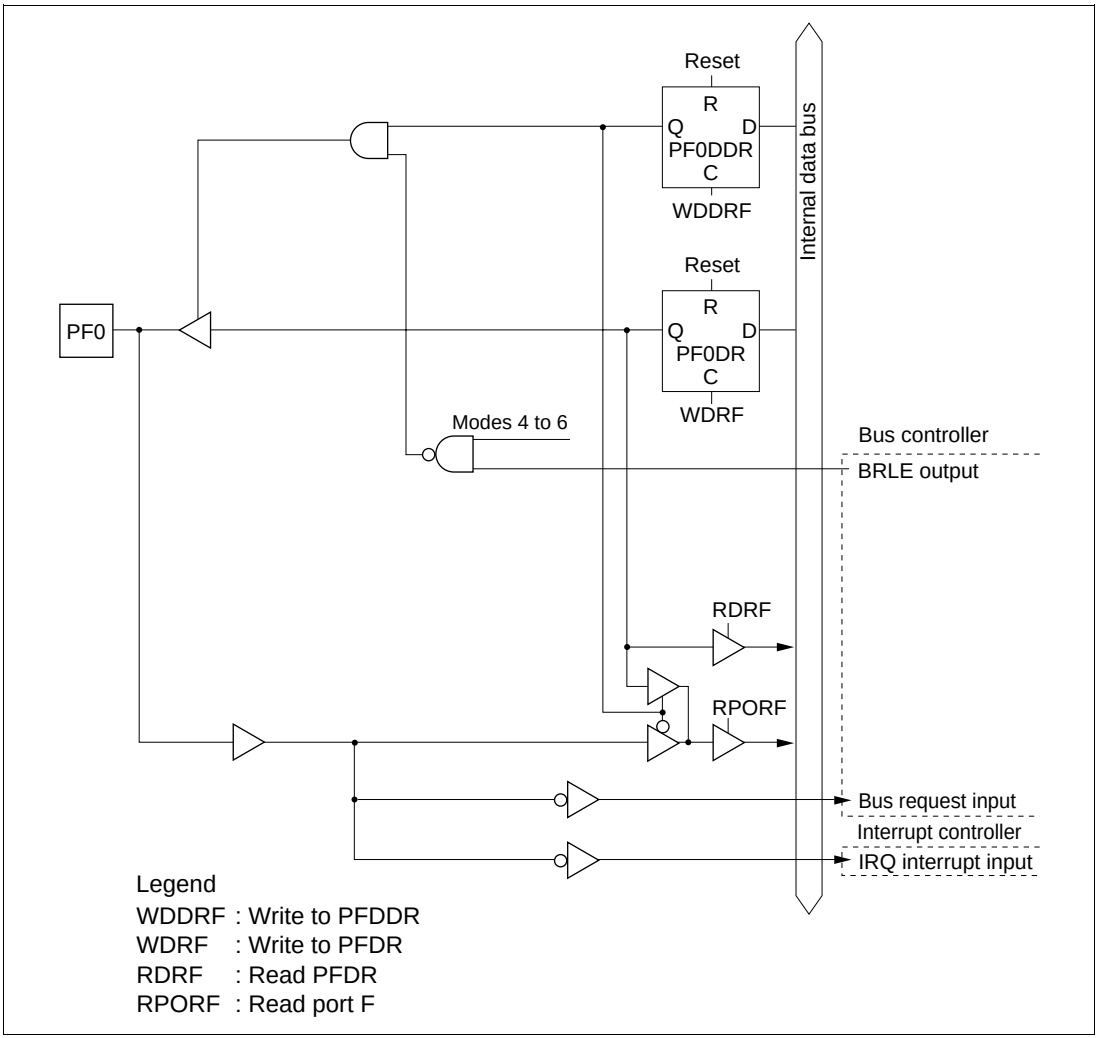


Figure C-25 Port F Block Diagram (Pin PF0)

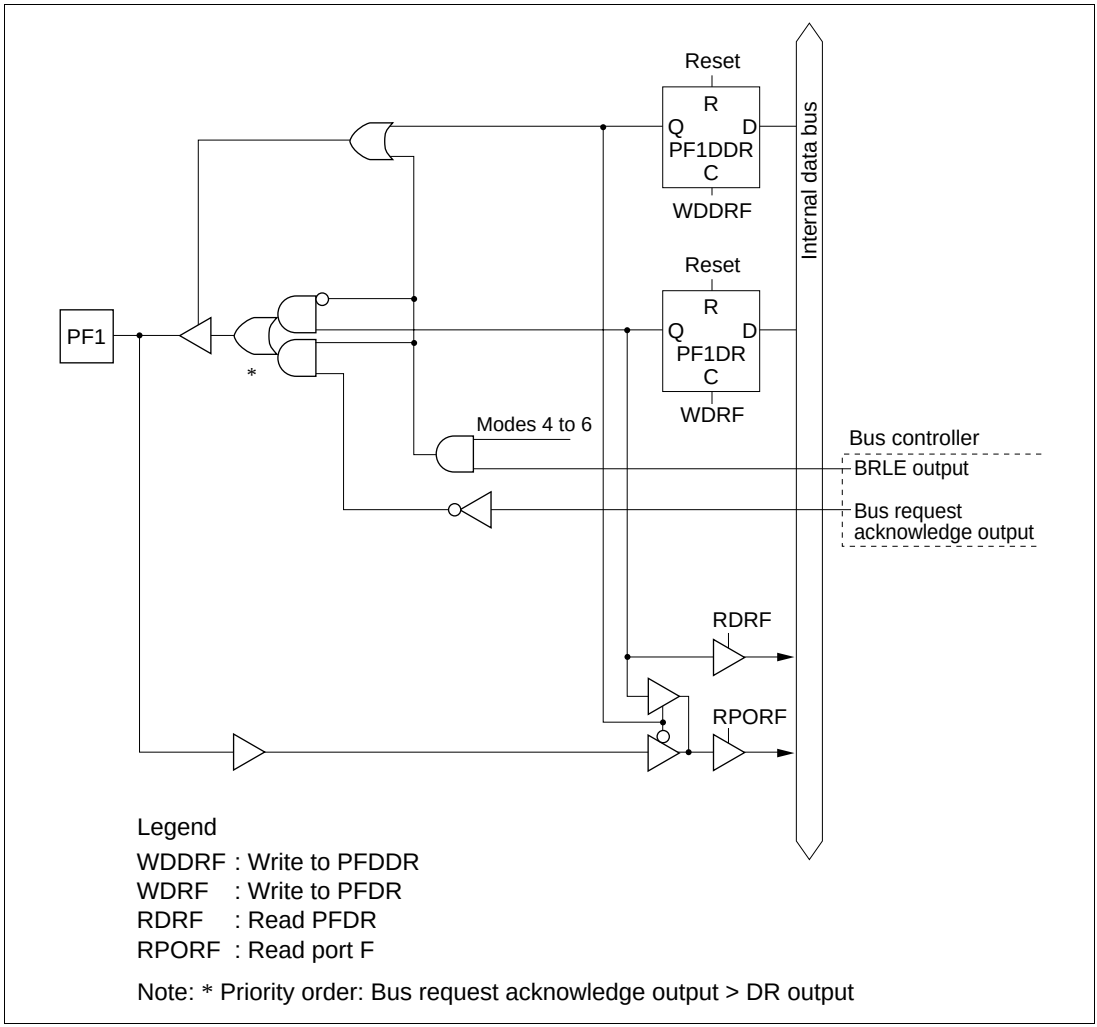


Figure C-26 Port F Block Diagram (Pin PF1)

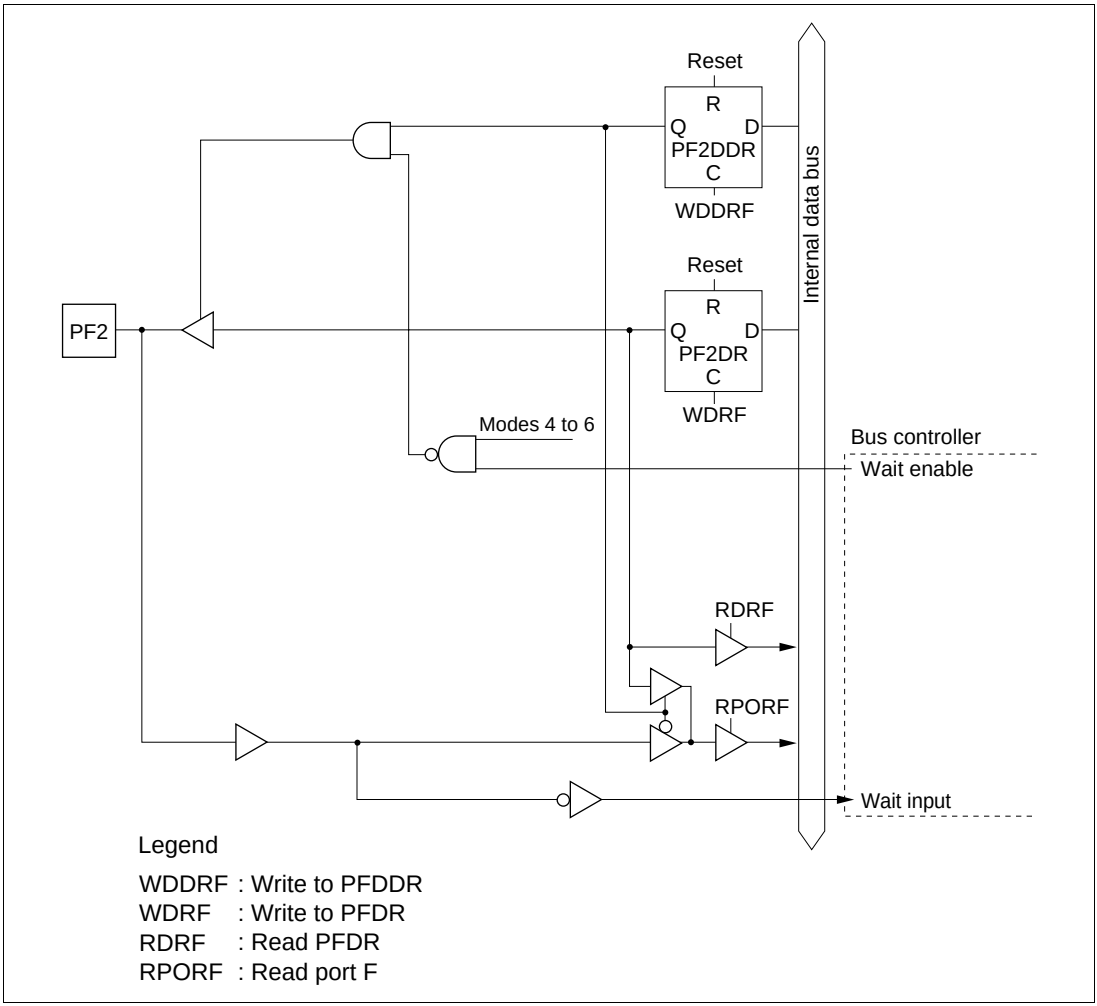


Figure C-27 Port F Block Diagram (Pin PF2)

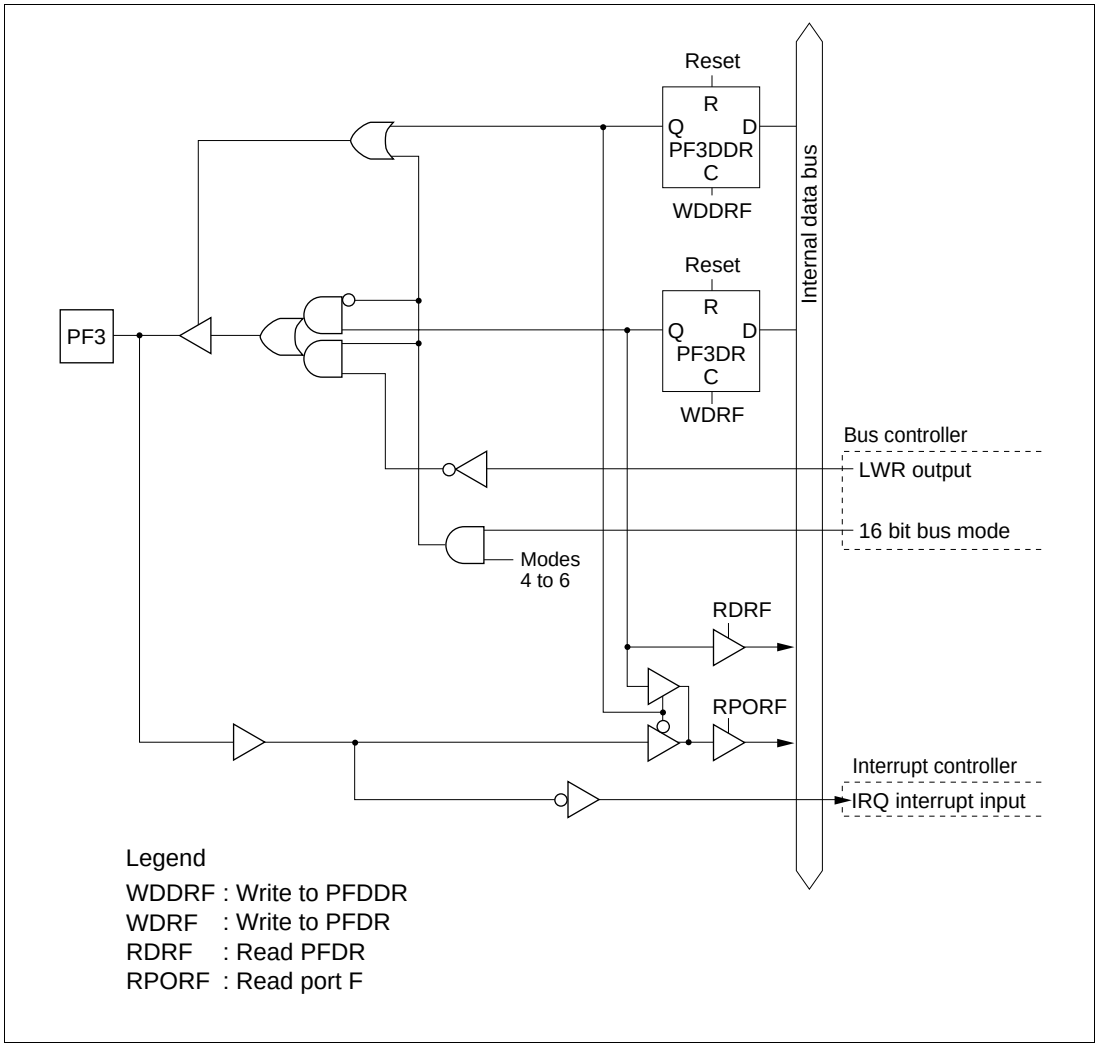


Figure C-28 Port F Block Diagram (Pin PF3)

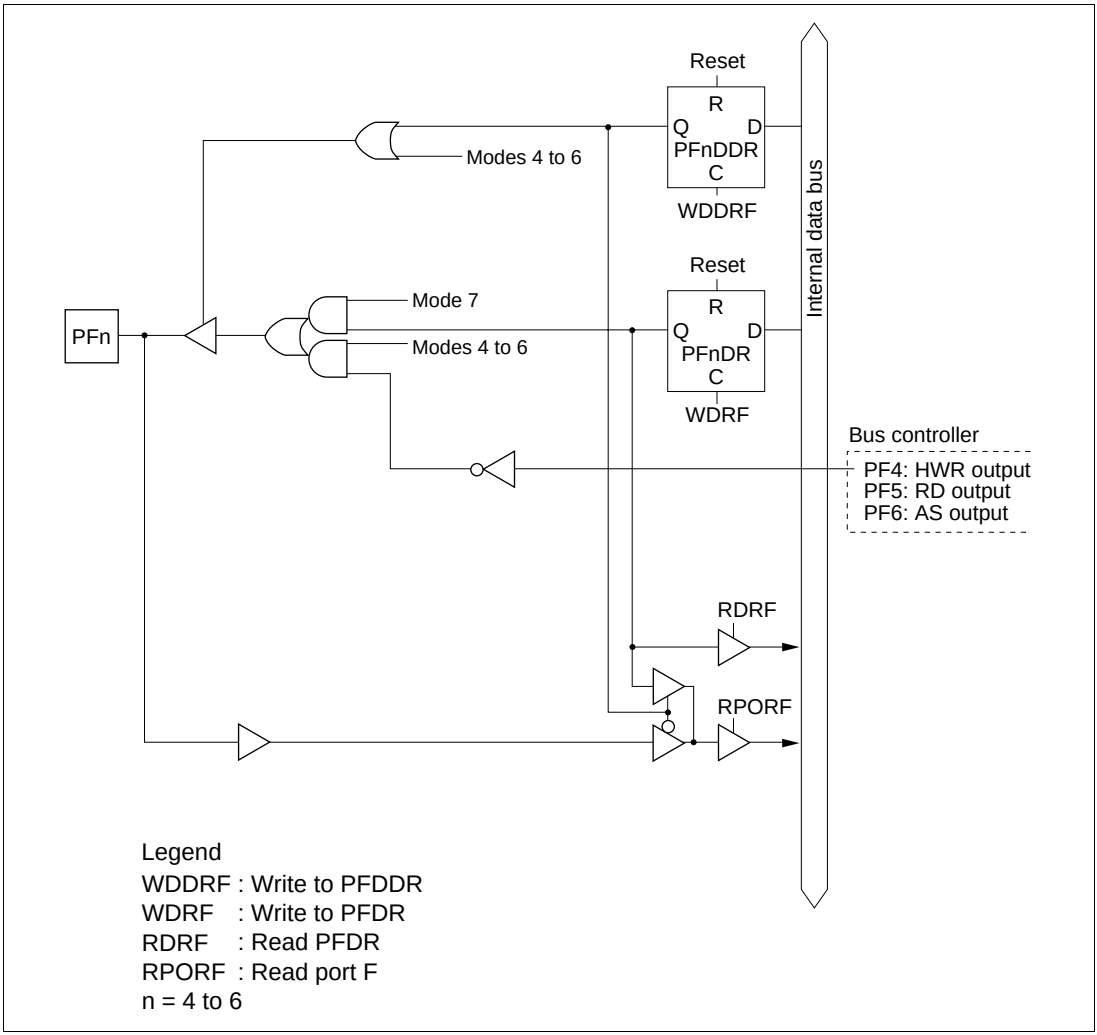


Figure C-29 Port F Block Diagram (Pins PF4 to PF6)

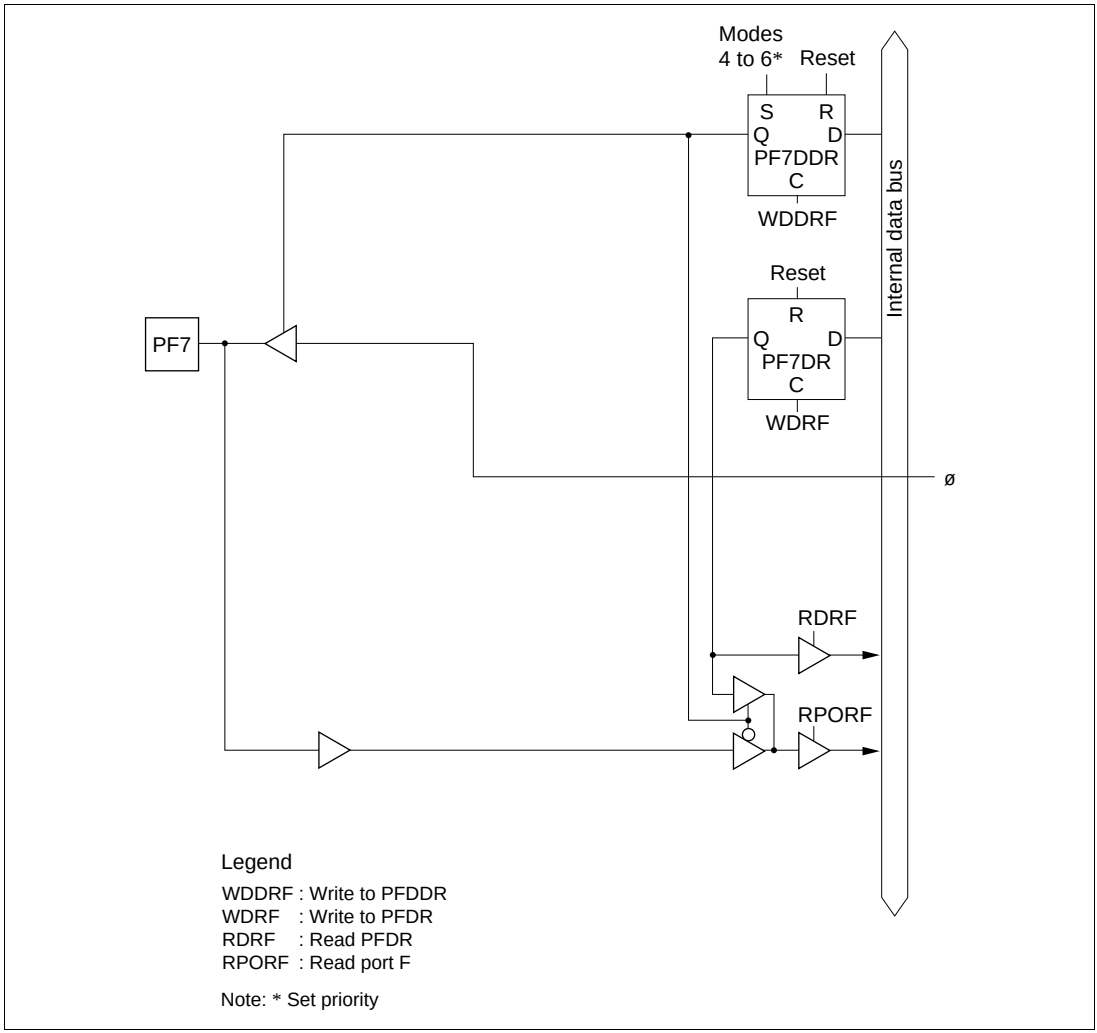


Figure C-30 Port F Block Diagram (Pin PF7)

C.12 Port G Block Diagrams

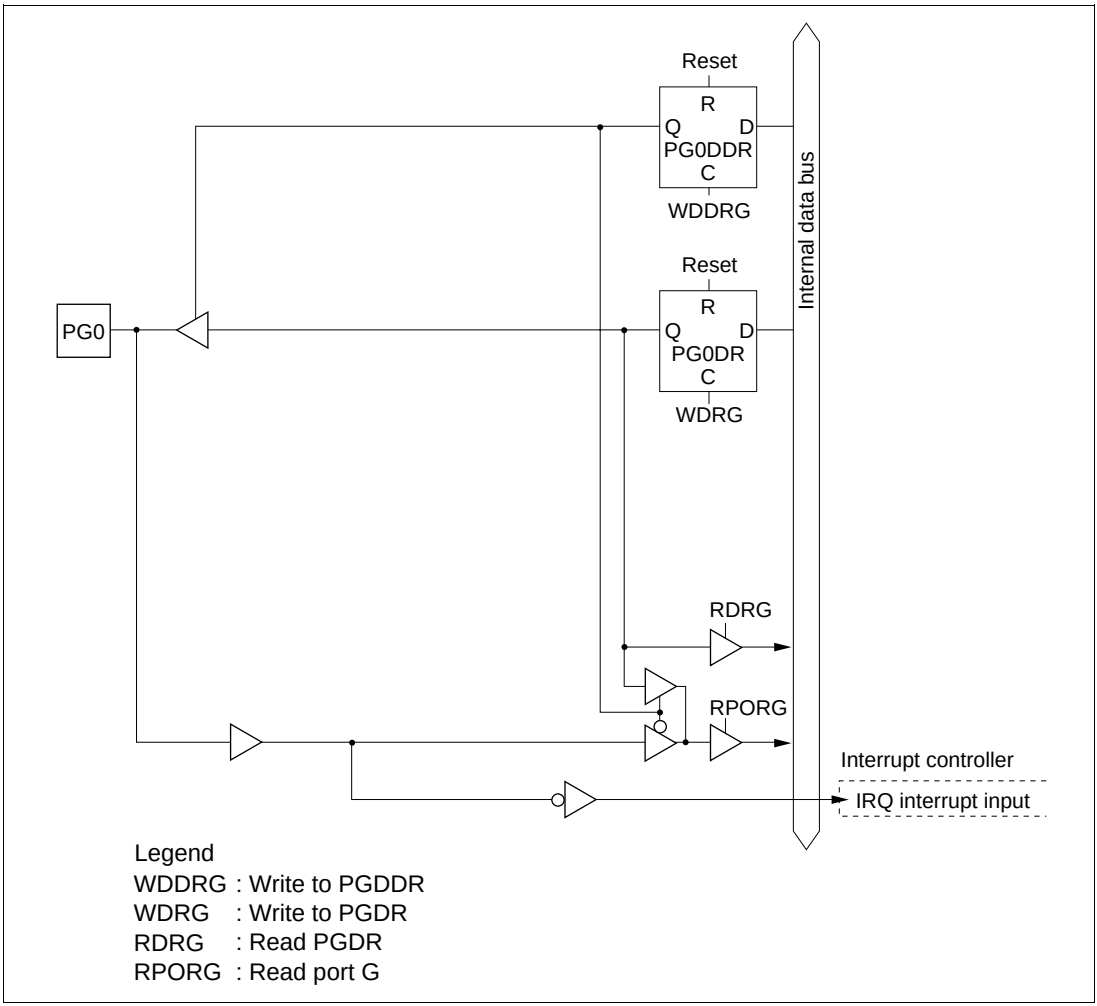


Figure C-31 Port G Block Diagram (Pin PG0)

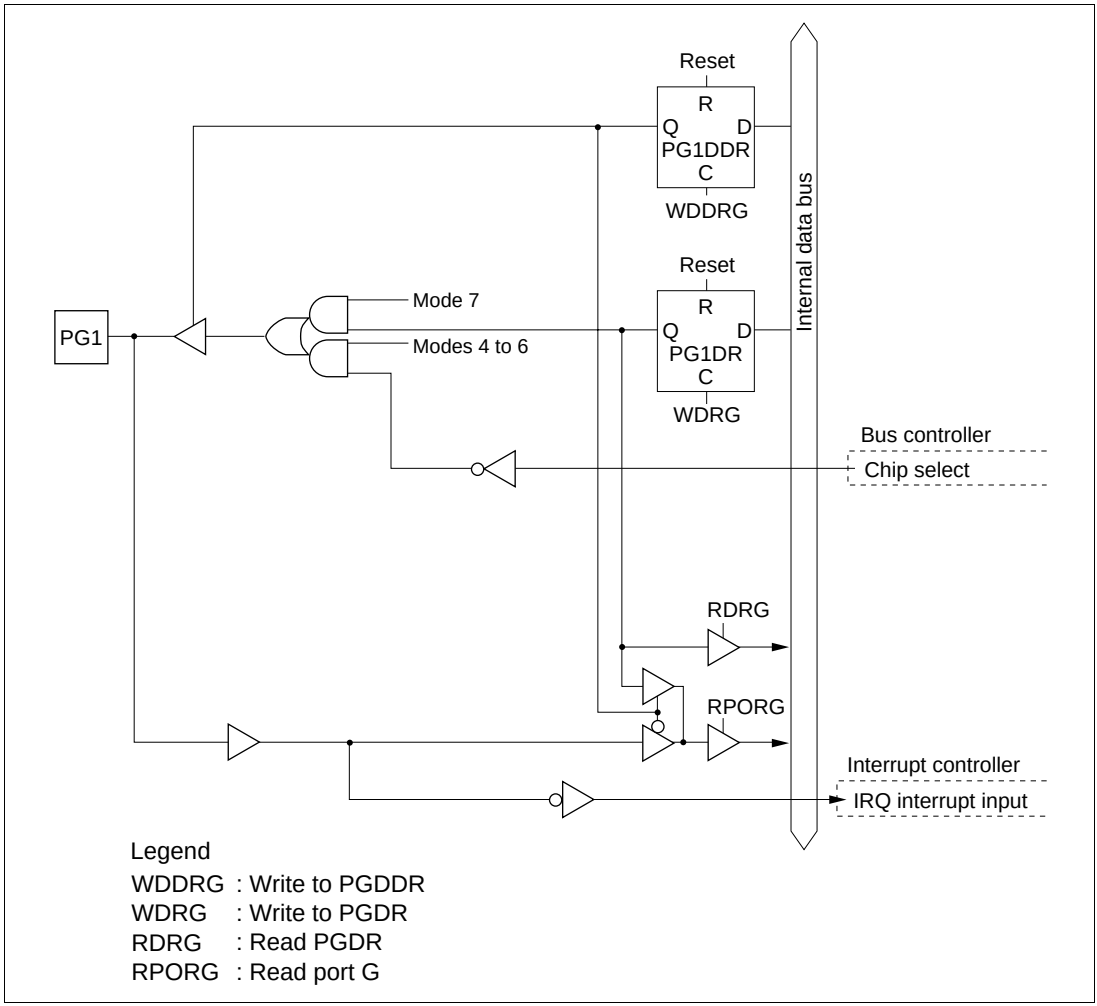


Figure C-32 Port G Block Diagram (Pin PG1)

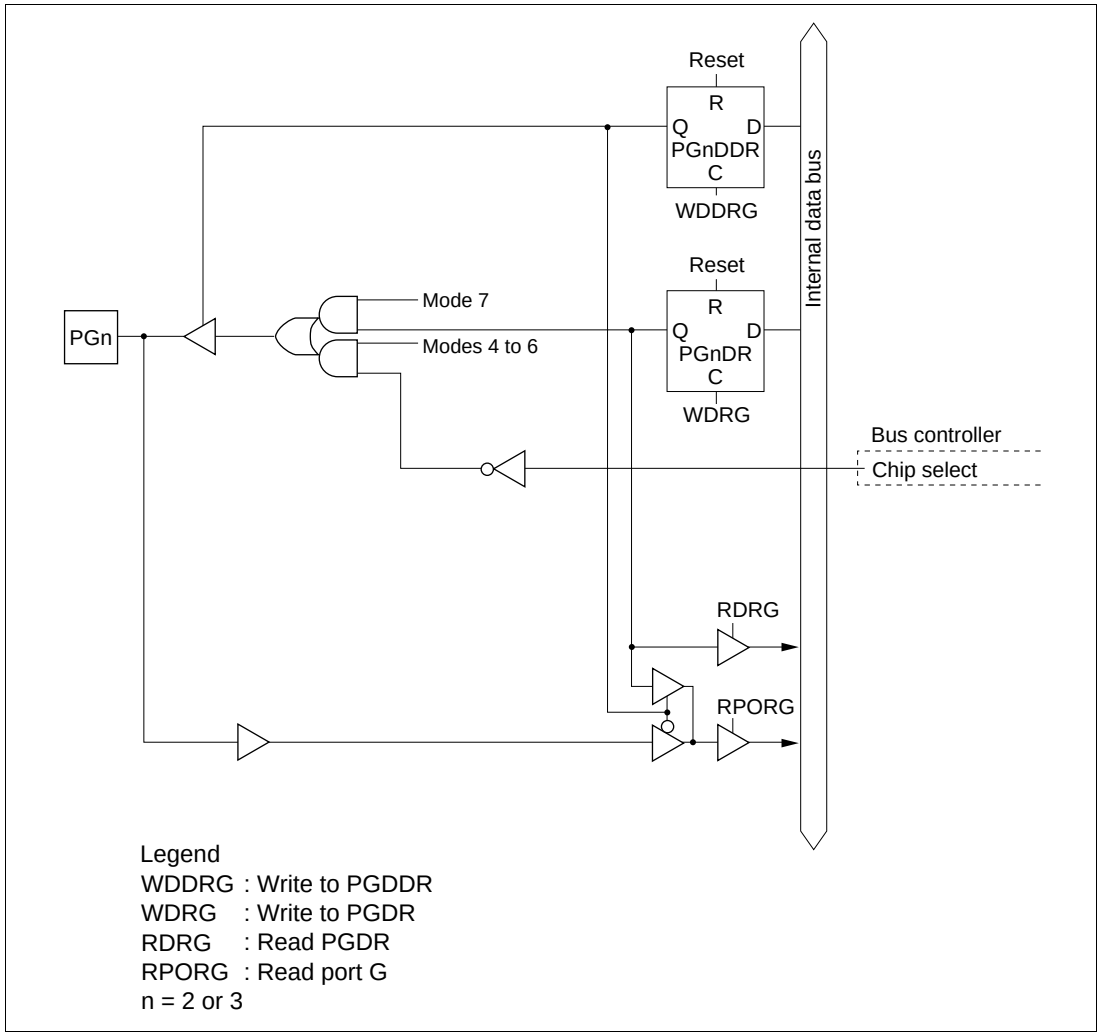


Figure C-33 Port G Block Diagram (Pins PG2 and PG3)

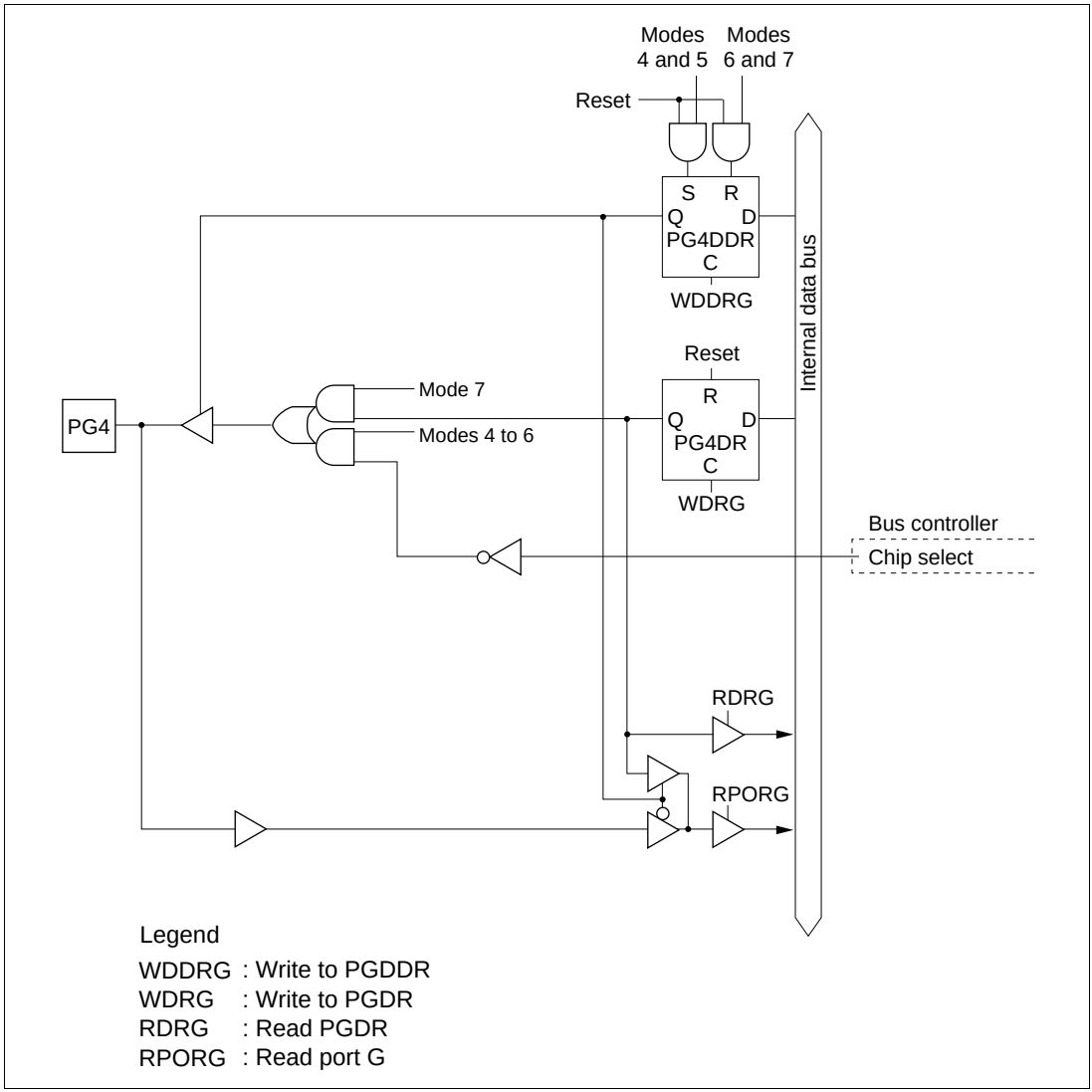


Figure C-34 Port G Block Diagram (Pin PG4)

Appendix D Pin States

D.1 Port States in Each Processing State

Table D-1 I/O Port States in Each Processing State

Port Name Pin Name		MCU Operating Mode	Power-On Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode, Watch Mode	Bus-Released State	Program Execution State, Sleep Mode, Subsleep Mode
P17 to P14		4 to 7	T	keep	T	keep	keep	I/O port
P13/TIOCD0/TCLKB/A23 P12/TIOCC0/TCLKA/A22 P11/TIOCB0/A21		7	T	keep	T	keep	keep	I/O port
	Address output selected by AEn bit	4 to 6	T	keep	T	[OPE= 0] T [OPE= 1] keep	T	Address output
	Port selected	4 to 6	T	keep	T	keep	keep	I/O port
P10/TIOCA0/A20		7	T	keep	T	keep	keep	I/O port
	Address output selected by AEn bit	4, 5	L	keep	T	[OPE= 0] T [OPE= 1] keep	T	Address output
		6	T					
	Port selected	4 to 6	T*	keep	T	keep	keep	I/O port
Port 3		4 to 7	T	keep	T	keep	keep	I/O port
Port 4		4 to 7	T	T	T	T	T	Input port
P77 to P74		4 to 7	T	keep	T	keep	keep	I/O port
P73/$\overline{CS7}$ P72/$\overline{CS6}$ P71/$\overline{CS5}$ P70/$\overline{CS4}$		7	T	keep	T	keep	keep	I/O port
		4 to 6	T	keep	T	[DDR·OPE= 0] T [DDR·OPE= 1] H	T	[DDR = 0] Input port [DDR = 1] $\overline{CS7}$ to $\overline{CS4}$
P96/DA0		4 to 7	T	T	T	[DAOEn= 1] keep [DAOEn= 0] T	keep	Input port
Port A		7	T	keep	T	keep	keep	I/O port
	Address output selected by AEn bit	4, 5	L	keep	T	[OPE= 0] T [OPE= 1] keep	T	Address output
		6	T					
	Port selected	4 to 6	T*	keep	T	keep	keep	I/O port

Port Name / Pin Name		MCU Operating Mode	Power-On Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode, Watch Mode	Bus-Released State	Program Execution State, Sleep Mode, Subsleep Mode
Port B		7	T	keep	T	keep	keep	I/O port
	Address output selected by AEn bit	4, 5	L	keep	T	[OPE= 0] T [OPE= 1] keep	T	Address output
		6	T					
	Port selected	4 to 6	T*	keep	T	keep	keep	I/O port
Port C		4, 5	L	keep	T	[OPE= 0] T [OPE= 1] keep	T	Address output
		6	T	keep	T	[DDR·OPE= 0] T [DDR·OPE= 1] keep	T	[DDR = 0] Input port [DDR = 1] Address output
		7	T	keep	T	keep	keep	I/O port
Port D		4 to 6	T	T	T	T	T	Data bus
		7	T	keep	T	keep	keep	I/O port
Port E	8-bit bus	4 to 6	T	keep	T	keep	keep	I/O port
	16-bit bus	4 to 6	T	T	T	T	T	Data bus
		7	T	keep	T	keep	keep	I/O port
PF7/ø		4 to 6	Clock output	[[DDR = 0] Input port [DDR = 1] Clock output	T	[DDR= 0] Input port [DDR= 1] H	[DDR= 0] Input port [DDR= 1] Clock output	[DDR= 0] Input port [DDR= 1] Clock output
		7	T	keep	T	[DDR= 0] Input port [DDR= 1] H	[DDR= 0] Input port [DDR= 1] Clock output	[DDR= 0] Input port [DDR= 1] Clock output
PF6/$\overline{\text{AS}}$, PF5/$\overline{\text{RD}}$, PF4/$\overline{\text{HWR}}$		4 to 6	H	H	T	[OPE= 0] T [OPE= 1] H	T	$\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$
		7	T	keep	T	keep	keep	I/O port
PF3/$\overline{\text{LWR}}$/$\overline{\text{IRQ3}}$		7	T	keep	T	keep	keep	I/O port
	8-bit bus	4 to 6	(Mode 4) H (Modes 5 and 6) T	keep	T	keep	keep	I/O port
	16-bit bus	4 to 6		H	T	[OPE= 0] T [OPE= 1] H	T	$\overline{\text{LWR}}$

Port Name Pin Name	MCU Operating Mode	Power-On Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode, Watch Mode	Bus-Released State	Program Execution State, Sleep Mode, Subsleep Mode
PF2/$\overline{\text{WAIT}}$	4 to 6	T	keep	T	[WAITE= 0] keep [WAITE= 1] T	[WAITE= 0] keep [WAITE= 1] T	[WAITE= 0] I/O port [WAITE= 1] $\overline{\text{WAIT}}$
	7	T	keep	T	keep	keep	I/O port
PF1/$\overline{\text{BACK}}$	4 to 6	T	keep	T	[BRLE= 0] keep [BRLE= 1] H	L	[BRLE= 0] I/O port [BRLE= 1] $\overline{\text{BACK}}$
	7	T	keep	T	keep	keep	I/O port
PF0/$\overline{\text{BREQ}}$/$\overline{\text{IREQ2}}$	4 to 6	T	keep	T	[BRLE= 0] keep [BRLE= 1] T	T	[BRLE= 0] I/O port [BRLE= 1] $\overline{\text{BREQ}}$
	7	T	keep	T	keep	keep	I/O port
PG4/$\overline{\text{CS0}}$	4, 5	H	keep	T	[DDR·OPE= 0] T [DDR·OPE= 1] H	T	[DDR = 0] Input port [DDR = 1] $\overline{\text{CS0}}$ (In sleep mode and subsleep mode: H)
	6	T					
	7	T	keep	T	keep	keep	I/O port
PG3/$\overline{\text{CS1}}$ PG2/$\overline{\text{CS2}}$ PG1/$\overline{\text{CS3}}$/$\overline{\text{IRQ7}}$	4 to 6	T	keep	T	[DDR·OPE= 0] T [DDR·OPE= 1] H	T	[DDR= 0] Input port [DDR= 1] $\overline{\text{CS1}}$ to $\overline{\text{CS3}}$
	7	T	keep	T	keep	keep	I/O port
PG0/$\overline{\text{IRQ6}}$	4 to 7	T	keep	T	keep	keep	I/O port

Legend:

H: High level

L: Low level

T: High impedance

keep: Input port becomes high-impedance, output port retains state

DDR Data direction register

OPE: Output port enable

WAITE: Wait input enable

BRLE: Bus release enable

Note: * L in modes 4 and 5 (address output)

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

Timing of Transition to Hardware Standby Mode

(1) To retain RAM contents with the RAME bit set to 1 in SYSCR, drive the $\overline{RES}$ signal low at least 10 states before the $\overline{STBY}$ signal goes low, as shown below. $\overline{RES}$ must remain low until $\overline{STBY}$ signal goes low (delay from $\overline{STBY}$ low to $\overline{RES}$ high: 0 ns or more).

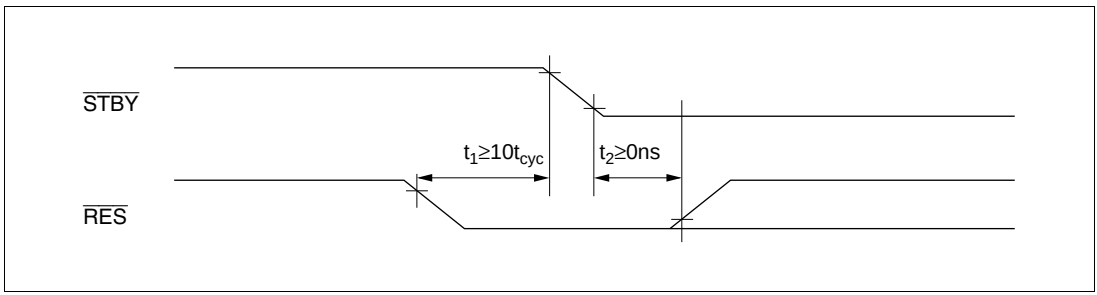


Figure E-1 Timing of Transition to Hardware Standby Mode

(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM contents do not need to be retained, $\overline{RES}$ does not have to be driven low as in (1).

Timing of Recovery from Hardware Standby Mode

Drive the $\overline{RES}$ signal low and the NMI signal high approximately 100 ns or more before $\overline{STBY}$ goes high to execute a power-on reset.

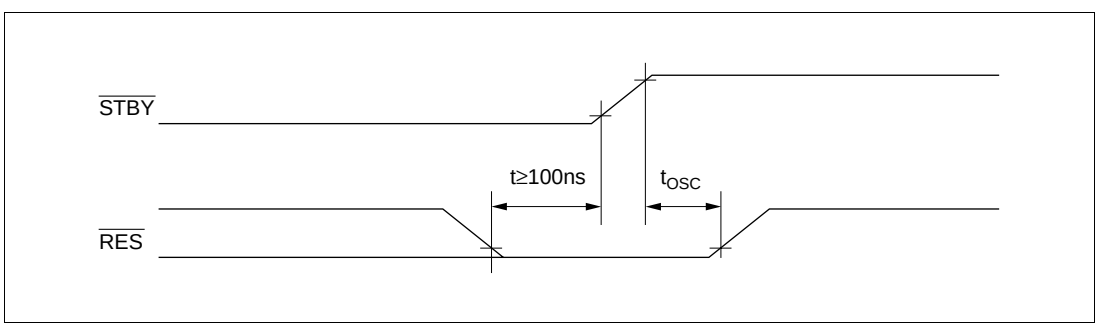


Figure E-2 Timing of Recovery from Hardware Standby Mode

Appendix F Product Code Lineup

Table F-1 H8S/2214 Product Code Lineup

Product Type		Product Code	Mark Code	Package
H8S/2214	Mask ROM version	HD6432214	6432214(***)TE	100-pin TQFP (TFP-100B)
			6432214(***)TF	100-pin TQFP (TFP-100G)
			6432214(***)BP	112-pin TFBGA (TBP-112)
	F-ZTAT version	HD64F2214	64F2214TE16	100-pin TQFP (TFP-100B)
			64F2214TF16	100-pin TQFP (TFP-100G)
			64F2214BP	112-pin TFBGA (TBP-112)

[Explanation of Symbol] (***) indicates ROM code.

Appendix G Package Dimensions

Figures G-1, G-2, and G-3 show the H8S/2214 package dimensions.

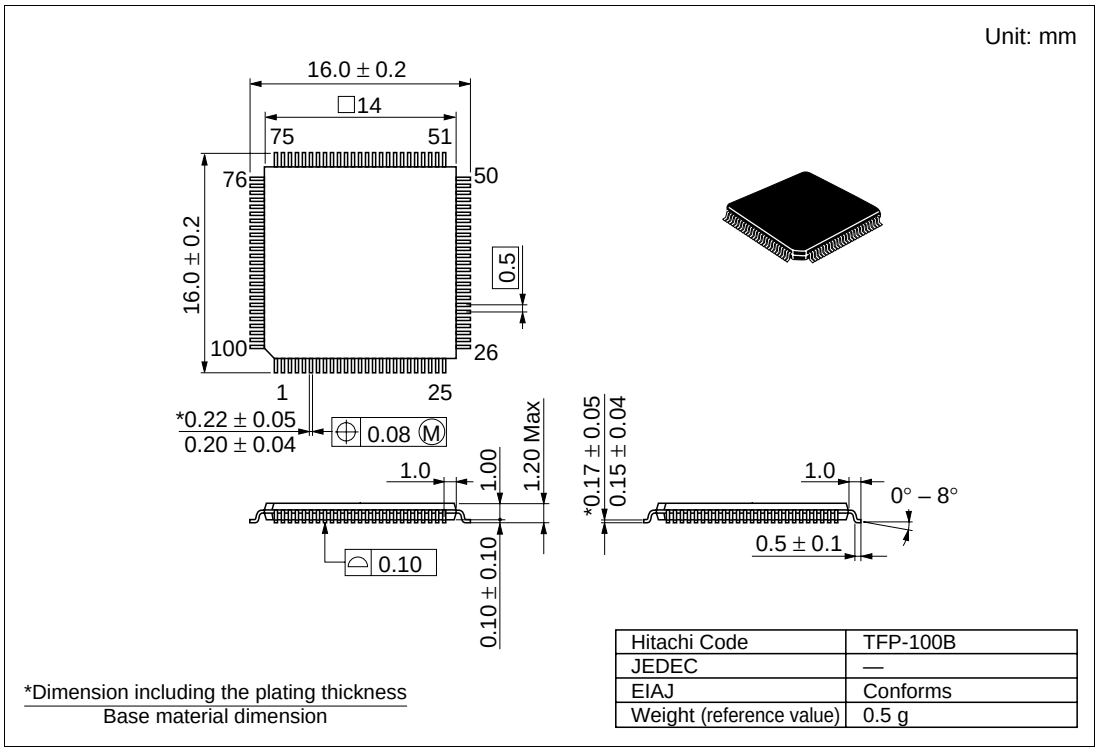
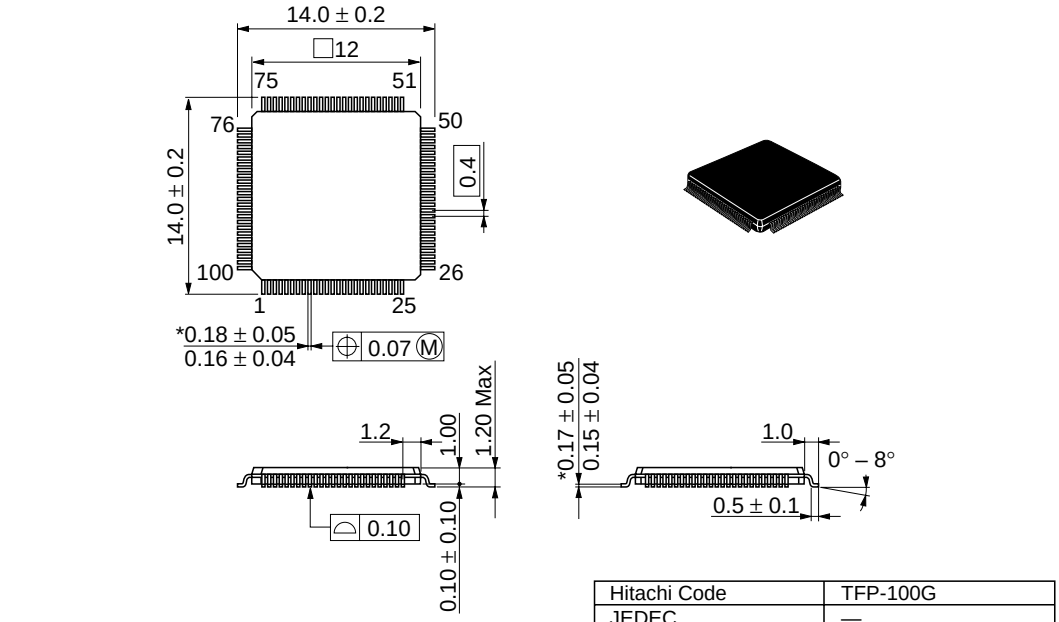


Figure G-1 TFP-100B Package Dimensions

Unit: mm

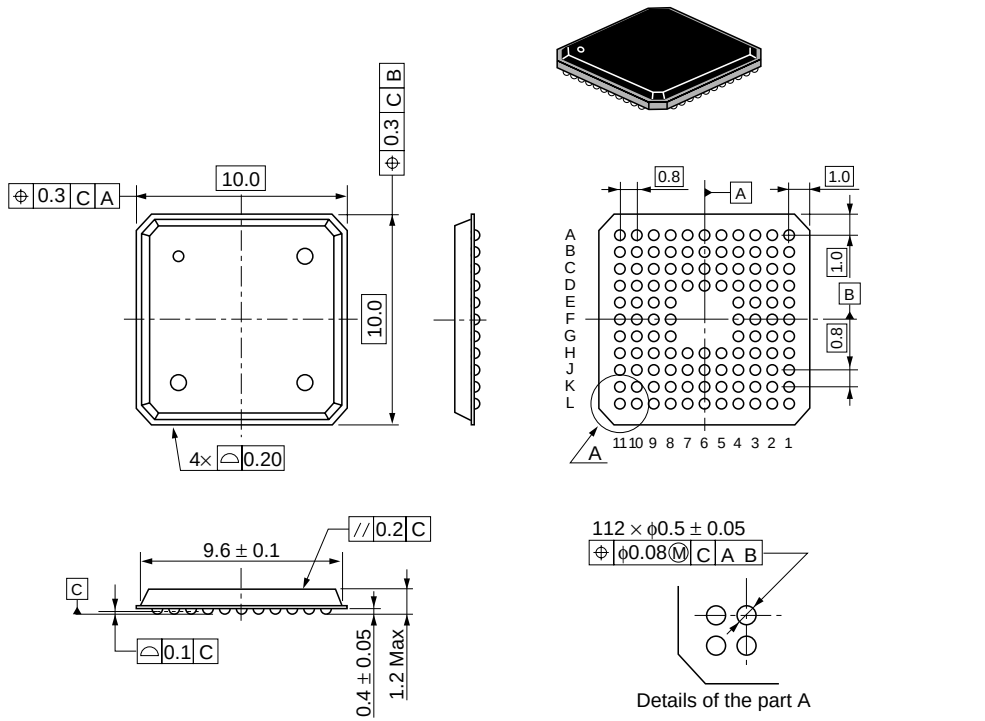


Hitachi Code	TFP-100G
JEDEC	—
EIAJ	Conforms
Weight (reference value)	0.4 g

*Dimension including the plating thickness
Base material dimension

Figure G-2 TFP-100G Package Dimensions

Unit: mm



Hitachi Code	TBP-112
JEDEC	—
EIAJ	—
Weight (reference value)	0.19 g

Figure G-3 TBP-112 Package Dimensions

H8S/2214, H8S/2214 F-ZTAT™ Hardware Manual

Publication Date: 1st Edition, April 2000
2nd Edition, July 2001
Published by: Customer Service Division
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.