To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

User's Manual

16

H8S/2345 Group, H8S/2345 F-ZTAT™

Hardware Manual

Renesas 16-Bit Single-Chip Microcomputer H8S Family/H8S/2300 Series

H8S/2345	HD6432345, HD6472345, HD64F2345
H8S/2344	HD6432344
H8S/2341	HD6432341
H8S/2340	HD64F2340

Renesas Electronics
www.renesas.com

Rev.4.00 2006.02

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

2. Treatment of Unused Input Pins

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

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined. The states of in
circuits are undefined until full power is supplied throughout the chip and a lo
input on the reset pin. During the period where the states are undefined, the re
settings and the output state of each pin are also undefined. Design your syste
does not malfunction because of processing while it is in this undefined state.
products which have a reset function, reset the LSI immediately after the pow
been turned on.

4. Prohibition of Access to Undefined or Reserved Addresses

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

on the high-level language C can also be run efficiently.

The address space is divided into eight areas. The data bus width and access states can for each of these areas, and various kinds of memory can be connected fast and easily.

On-chip memory consists of large-capacity ROM and RAM. With regard to on-chip R single power supply flash memory (F-ZTAT™*[2]), PROM (ZTAT®*[2]), and mask ROM are available, providing a quick and flexible response to conditions from ramp-up thro scale volume production, even for applications with frequently changing specification

On-chip supporting functions include a 16-bit timer pulse unit (TPU), 8-bit timers, wa (WDT), serial communication interface (SCI), A/D converter, D/A converter, and I/O

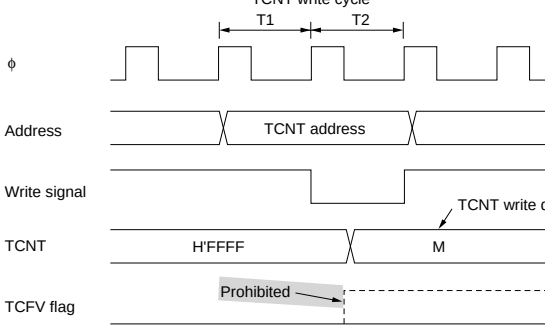
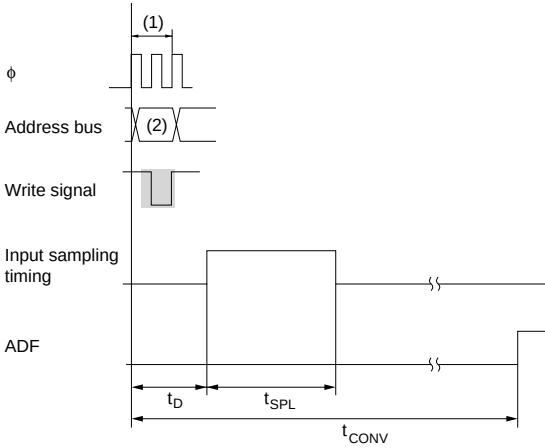
An on-chip data transfer controller (DTC) is also provided, enabling high-speed data t without CPU intervention.

Use of the H8S/2345 Group enables compact, high-performance systems to be implem easily.

This manual describes the hardware of the H8S/2345 Group. Refer to the H8S/2600 S H8S/2000 Series Programming Manual for a detailed description of the instruction set

Notes: 1. The H8S/2345, H8S/2344, H8S/2343, and H8S/2341 have on-chip ROM. The H8S/2340 does not have on-chip ROM.

2. F-ZTAT (Flexible-ZTAT) is a trademark of Renesas Technology Corp. ZTAT is a registered trademark of Renesas Technology Corp.

Figure 9.57 Contention between TCNT Write and Overflow		
11.2.2 Timer Control/Status Register (TCSR)	399	Table amended [Clearing condition] Cleared by reading TCSR when OVF = 1*, then wr OVF Note * added Note: * When OVF is polled and the interval timer i disabled. OVF = 1 must be read at least twice.
14.4.3 Input Sampling and A/D Conversion Time Figure 14.5 A/D Conversion Timing	521	Figure 14.5 amended 

		Notes: 7. Minimum number of times for which all cha are guaranteed after rewriting. (Guarantee range is minimum value.) 8. Reference value for 25°C (as a guideline, rewritin normally function up to this value). 9. Data retention characteristic when rewriting is per within the specification range, including the minimum
Appendix G Package Dimensions Figure G.1 TFP-100B Package Dimensions	897	Figure G.1 replaced
Figure G.2 TFP-100G Package Dimensions	898	Figure G.2 replaced
Figure G.3 FP-100A Package Dimensions	899	Figure G.3 replaced
Figure G.4 FP-100B Package Dimensions	900	Figure G.4 replaced

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

On-chip peripheral functions required for system configuration include data transfer c (DTC) bus masters, ROM and RAM, a 16-bit timer-pulse unit (TPU), an 8-bit timer, a timer (WDT), a serial communication interface (SCI), an A/D converter, a D/A conve ports.

The on-chip ROM*1 is either single power supply flash memory (F-ZTAT™*2), PRO (ZTAT®*2), or mask ROM, with a capacity of 128, 96, 64, or 32 kbytes. ROM is con CPU via a 16-bit data bus, enabling both byte and word data to be accessed in one sta Instruction fetching has been speeded up, and processing speed increased.

Seven operating modes, modes 1 to 7, are provided, and there is a choice of address sp single-chip mode or external expansion mode.

The features of the H8S/2345 Group are shown in table 1.1.

Notes: 1. The H8S/2345, H8S/2344, H8S/2343, and H8S/2341 have on-chip ROM. T H8S/2340 does not have on-chip ROM.

2. F-ZTAT is a trademark of Renesas Technology Corp.
   ZTAT is a registered trademark of Renesas Technology Corp.

	— High-speed arithmetic operations 8/16/32-bit register-register add/subtract : 50 ns 16 × 16-bit register-register multiply : 1000 ns 32 ÷ 16-bit register-register divide : 1000 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 (ZTAT, mask ROM, ROMless versions only) — Advanced mode: 16-Mbyte address space
Bus controller	• Address space divided into 8 areas, with bus specifications set independently for each area • Chip select output possible for areas 0 to 3 • Choice of 8-bit or 16-bit access space for each area • 2-state or 3-state access space can be designated for each are • Number of program wait states can be set for each area • Burst ROM directly connectable • External bus release function
Data transfer controller (DTC)	• Can be activated by internal interrupt or software • Multiple transfers or multiple types of transfer possible for one source • Transfer is possible in repeat mode, block transfer mode, etc. • Request can be sent to CPU for interrupt that activated DTC

Watchdog timer	• Watchdog timer or interval timer selectable
Serial communication interface (SCI) 2 channels	• Asynchronous mode or synchronous mode selectable • Multiprocessor communication function • Smart card interface function
A/D converter	• Resolution: 10 bits • Input: 8 channels • High-speed conversion: 6.7 µs minimum conversion time (at 2 operation) • Single or scan mode selectable • Sample and hold circuit • A/D conversion can be activated by external trigger or timer tr
D/A converter	• Resolution: 8 bits • Output: 2 channels
I/O ports	• 71 I/O pins, 8 input-only pins
Memory	• Flash memory, PROM, or mask ROM • High-speed static RAM

Product Name	ROM	RAM
H8S/2345	128 kbytes	4 kbytes
H8S/2344	96 kbytes	4 kbytes
H8S/2343	64 kbytes	2 kbytes
H8S/2341	32 kbytes	2 kbytes
H8S/2340	—	2 kbytes

Interrupt controller	• Nine external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$) • 43 internal interrupt sources • Eight priority levels settable

Mode	CPU Operating Mode	Description	On-Chip ROM	External I Initial Value
0	—	—	—	—
1				
2				
3				
4	Advanced	On-chip ROM disabled expansion mode	Disabled	16 bits
5				8 bits
6		On-chip ROM enabled expansion mode	Enabled	8 bits
7		Single-chip mode		—
8	—	—	—	—
9				
10	Advanced	Boot mode	Enabled	8 bits
11				—
12	—	—	—	—
13				
14	Advanced	User-programmable mode	Enabled	8 bits
15				—

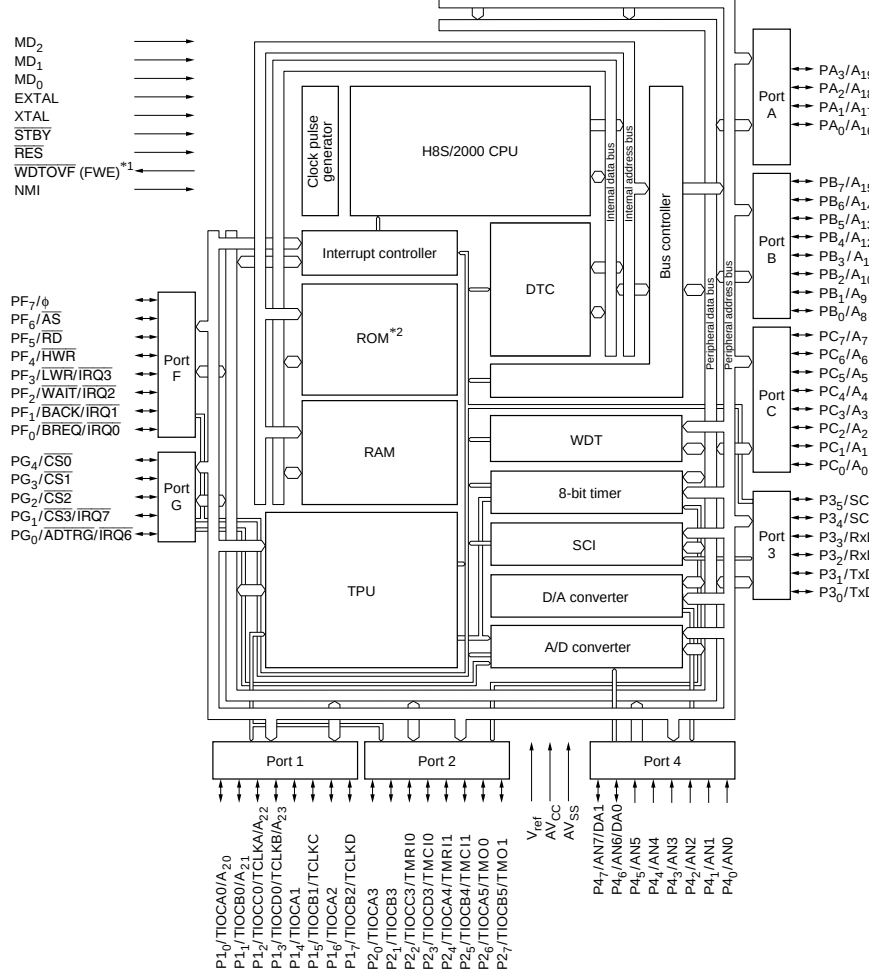
			expansion mode		
	3*		Single-chip mode	Enabled	—
	4	Advanced	On-chip ROM disabled expansion mode	Disabled	16 bits
	5		On-chip ROM disabled expansion mode	Disabled	8 bits
	6*		On-chip ROM enabled expansion mode	Enabled	8 bits
	7*		Single-chip mode	Enabled	—

Note: * Not used on ROMless versions.

Clock pulse generator	• Built-in duty correction circuit
Packages	• 100-pin plastic TQFP (TFP-100B, TFP-100G) • 100-pin plastic QFP (FP-100A, FP-100B)

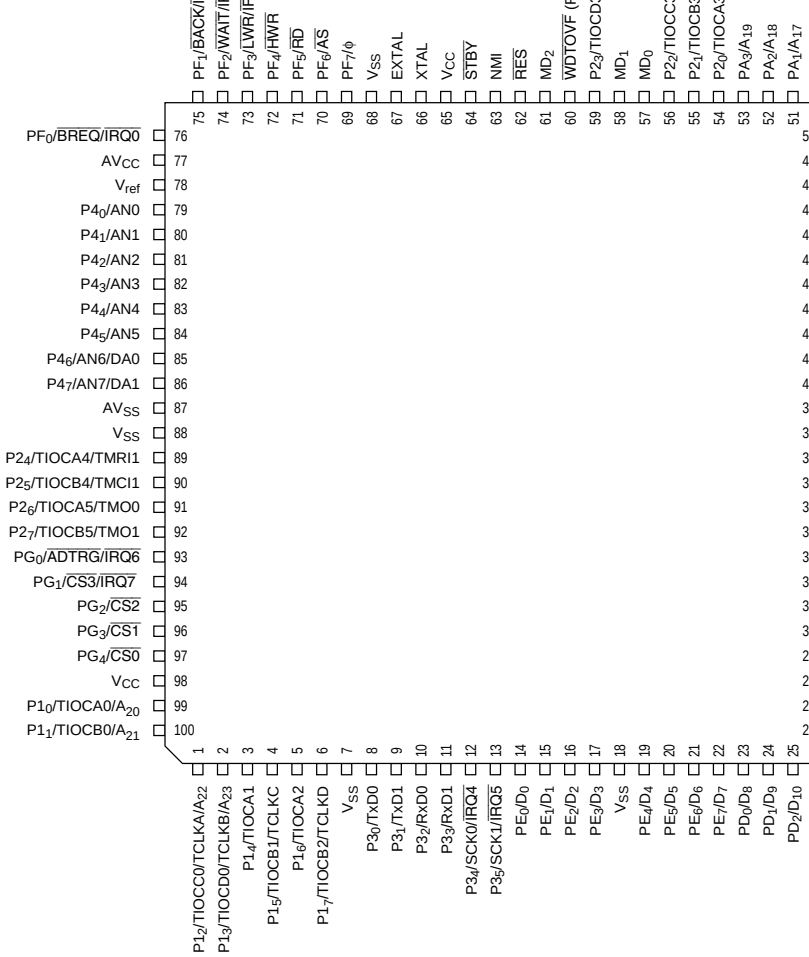
Product lineup

Model Name			
Mask ROM Version	**F-ZTAT**	**ZTAT**	**ROM/RAM (Bytes)**
HD6432345	HD64F2345	HD6472345	128 k/4 k
HD6432344	—	—	96 k/4 k
HD6432343	—	—	64 k/2 k
HD6432341	—	—	32 k/2 k
HD6412340 (ROMless versions)	—	—	—/2 k



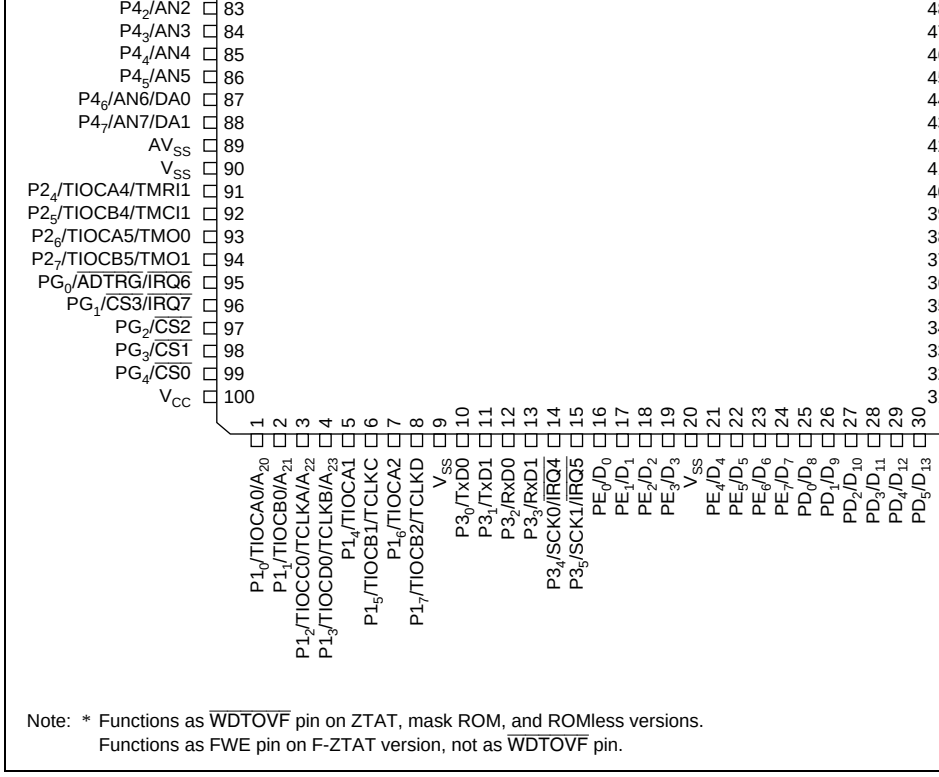
Notes: 1. Functions as $\overline{\text{WDTOVF}}$ pin on ZTAT, mask ROM, and ROMless versions.
Functions as FWE pin on F-ZTAT version, not as $\overline{\text{WDTOVF}}$ pin.
2. Not present on ROMless version.

Figure 1.1 Block Diagram



Note: * Functions as $\overline{\text{WDTOVF}}$ pin on ZTAT, mask ROM, and ROMless versions.
Functions as FWE pin on F-ZTAT version, not as $\overline{\text{WDTOVF}}$ pin.

Figure 1.2 Pin Arrangement (FP-100B, TFP-100B, TFP-100G: Top Vie



Note: * Functions as $\overline{\text{WDTOVF}}$ pin on ZTAT, mask ROM, and ROMless versions.
Functions as FWE pin on F-ZTAT version, not as $\overline{\text{WDTOVF}}$ pin.

Figure 1.3 Pin Arrangement (FP-100A: Top View)

TFP-100G	FP-100A	1*1	2*1 *2	3*1 *2	4	5	6*2	7*2	Mo
1	3	$P1_2$/ TIOCC0/ TCLKA	$P1_2$/ TIOCC0/ TCLKA	$P1_2$/ TIOCC0/ TCLKA	$P1_2$/ TIOCC0/ TCLKA/ A_{22}	$P1_2$/ TIOCC0/ TCLKA/ A_{22}	$P1_2$/ TIOCC0/ TCLKA/ A_{22}	$P1_2$/ TIOCC0/ TCLKA	NC
2	4	$P1_3$/ TIOCD0/ TCLKB	$P1_3$/ TIOCD0/ TCLKB	$P1_3$/ TIOCD0/ TCLKB	$P1_3$/ TIOCD0/ TCLKB/ A_{23}	$P1_3$/ TIOCD0/ TCLKB/ A_{23}	$P1_3$/ TIOCD0/ TCLKB/ A_{23}	$P1_3$/ TIOCD0/ TCLKB	NC
3	5	$P1_4$/ TIOCA1	$P1_4$/ TIOCA1	$P1_4$/ TIOCA1	$P1_4$/ TIOCA1	$P1_4$/ TIOCA1	$P1_4$/ TIOCA1	$P1_4$/ TIOCA1	NC
4	6	$P1_5$/ TIOCB1/ TCLKC	$P1_5$/ TIOCB1/ TCLKC	$P1_5$/ TIOCB1/ TCLKC	$P1_5$/ TIOCB1/ TCLKC	$P1_5$/ TIOCB1/ TCLKC	$P1_5$/ TIOCB1/ TCLKC	$P1_5$/ TIOCB1/ TCLKC	NC
5	7	$P1_6$/ TIOCA2	$P1_6$/ TIOCA2	$P1_6$/ TIOCA2	$P1_6$/ TIOCA2	$P1_6$/ TIOCA2	$P1_6$/ TIOCA2	$P1_6$/ TIOCA2	NC
6	8	$P1_7$/ TIOCB2/ TCLKD	$P1_7$/ TIOCB2/ TCLKD	$P1_7$/ TIOCB2/ TCLKD	$P1_7$/ TIOCB2/ TCLKD	$P1_7$/ TIOCB2/ TCLKD	$P1_7$/ TIOCB2/ TCLKD	$P1_7$/ TIOCB2/ TCLKD	NC
7	9	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{ss}
8	10	$P3_0$/TxD0	$P3_0$/TxD0	$P3_0$/TxD0	$P3_0$/TxD0	$P3_0$/TxD0	$P3_0$/TxD0	$P3_0$/TxD0	NC
9	11	$P3_1$/TxD1	$P3_1$/TxD1	$P3_1$/TxD1	$P3_1$/TxD1	$P3_1$/TxD1	$P3_1$/TxD1	$P3_1$/TxD1	NC
10	12	$P3_2$/RxD0	$P3_2$/RxD0	$P3_2$/RxD0	$P3_2$/RxD0	$P3_2$/RxD0	$P3_2$/RxD0	$P3_2$/RxD0	NC
11	13	$P3_3$/RxD1	$P3_3$/RxD1	$P3_3$/RxD1	$P3_3$/RxD1	$P3_3$/RxD1	$P3_3$/RxD1	$P3_3$/RxD1	NC
12	14	$P3_4$/ SCK0/ $\overline{\text{IRQ4}}$	$P3_4$/ SCK0/ $\overline{\text{IRQ4}}$	$P3_4$/ SCK0/ $\overline{\text{IRQ4}}$	$P3_4$/ SCK0/ $\overline{\text{IRQ4}}$	$P3_4$/ SCK0/ $\overline{\text{IRQ4}}$	$P3_4$/ SCK0/ $\overline{\text{IRQ4}}$	$P3_4$/ SCK0/ $\overline{\text{IRQ4}}$	NC
13	15	$P3_5$/ SCK1/ $\overline{\text{IRQ5}}$	$P3_5$/ SCK1/ $\overline{\text{IRQ5}}$	$P3_5$/ SCK1/ $\overline{\text{IRQ5}}$	$P3_5$/ SCK1/ $\overline{\text{IRQ5}}$	$P3_5$/ SCK1/ $\overline{\text{IRQ5}}$	$P3_5$/ SCK1/ $\overline{\text{IRQ5}}$	$P3_5$/ SCK1/ $\overline{\text{IRQ5}}$	NC
14	16	PE_0/D_0	PE_0/D_0	PE_0	PE_0/D_0	PE_0/D_0	PE_0/D_0	PE_0	NC
15	17	PE_1/D_1	PE_1/D_1	PE_1	PE_1/D_1	PE_1/D_1	PE_1/D_1	PE_1	NC
16	18	PE_2/D_2	PE_2/D_2	PE_2	PE_2/D_2	PE_2/D_2	PE_2/D_2	PE_2	NC
17	19	PE_3/D_3	PE_3/D_3	PE_3	PE_3/D_3	PE_3/D_3	PE_3/D_3	PE_3	NC

22	24	PE_7/D_7	PE_7/D_7	PE_7	PE_7/D_7	PE_7/D_7	PE_7/D_7	PE_7	NC
23	25	D_8	D_8	PD_0	D_8	D_8	D_8	PD_0	EO_0
24	26	D_9	D_9	PD_1	D_9	D_9	D_9	PD_1	EO_1
25	27	D_{10}	D_{10}	PD_2	D_{10}	D_{10}	D_{10}	PD_2	EO_2
26	28	D_{11}	D_{11}	PD_3	D_{11}	D_{11}	D_{11}	PD_3	EO_3
27	29	D_{12}	D_{12}	PD_4	D_{12}	D_{12}	D_{12}	PD_4	EO_4
28	30	D_{13}	D_{13}	PD_5	D_{13}	D_{13}	D_{13}	PD_5	EO_5
29	31	D_{14}	D_{14}	PD_6	D_{14}	D_{14}	D_{14}	PD_6	EO_6
30	32	D_{15}	D_{15}	PD_7	D_{15}	D_{15}	D_{15}	PD_7	EO_7
31	33	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
32	34	A_0	PC_0/A_0	PC_0	A_0	A_0	PC_0/A_0	PC_0	EA_0
33	35	A_1	PC_1/A_1	PC_1	A_1	A_1	PC_1/A_1	PC_1	EA_1
34	36	A_2	PC_2/A_2	PC_2	A_2	A_2	PC_2/A_2	PC_2	EA_2
35	37	A_3	PC_3/A_3	PC_3	A_3	A_3	PC_3/A_3	PC_3	EA_3
36	38	A_4	PC_4/A_4	PC_4	A_4	A_4	PC_4/A_4	PC_4	EA_4
37	39	A_5	PC_5/A_5	PC_5	A_5	A_5	PC_5/A_5	PC_5	EA_5
38	40	A_6	PC_6/A_6	PC_6	A_6	A_6	PC_6/A_6	PC_6	EA_6
39	41	A_7	PC_7/A_7	PC_7	A_7	A_7	PC_7/A_7	PC_7	EA_7
40	42	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
41	43	A_8	PB_0/A_8	PB_0	A_8	A_8	PB_0/A_8	PB_0	EA_8
42	44	A_9	PB_1/A_9	PB_1	A_9	A_9	PB_1/A_9	PB_1	$\overline{OE}$
43	45	A_{10}	PB_2/A_{10}	PB_2	A_{10}	A_{10}	PB_2/A_{10}	PB_2	EA_{10}
44	46	A_{11}	PB_3/A_{11}	PB_3	A_{11}	A_{11}	PB_3/A_{11}	PB_3	EA_{11}
45	47	A_{12}	PB_4/A_{12}	PB_4	A_{12}	A_{12}	PB_4/A_{12}	PB_4	EA_{12}
46	48	A_{13}	PB_5/A_{13}	PB_5	A_{13}	A_{13}	PB_5/A_{13}	PB_5	EA_{13}
47	49	A_{14}	PB_6/A_{14}	PB_6	A_{14}	A_{14}	PB_6/A_{14}	PB_6	EA_{14}
48	50	A_{15}	PB_7/A_{15}	PB_7	A_{15}	A_{15}	PB_7/A_{15}	PB_7	EA_{15}
49	51	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
50	52	PA_0	PA_0	PA_0	A_{16}	A_{16}	PA_0/A_{16}	PA_0	EA_{16}

55	57	$P2_1$/ TIOCB3	$P2_1$/ TIOCB3	$P2_1$/ TIOCB3	$P2_1$/ TIOCB3	$P2_1$/ TIOCB3	$P2_1$/ TIOCB3	$P2_1$/ TIOCB3	NC
56	58	$P2_2$/ TIOCC3/ TMRI0	$P2_2$/ TIOCC3/ TMRI0	$P2_2$/ TIOCC3/ TMRI0	$P2_2$/ TIOCC3/ TMRI0	$P2_2$/ TIOCC3/ TMRI0	$P2_2$/ TIOCC3/ TMRI0	$P2_2$/ TIOCC3/ TMRI0	NC
57	59	MD_0	MD_0	MD_0	MD_0	MD_0	MD_0	MD_0	V_{SS}
58	60	MD_1	MD_1	MD_1	MD_1	MD_1	MD_1	MD_1	V_{SS}
59	61	$P2_3$/ TIOCD3/ TMCI0	$P2_3$/ TIOCD3/ TMCI0	$P2_3$/ TIOCD3/ TMCI0	$P2_3$/ TIOCD3/ TMCI0	$P2_3$/ TIOCD3/ TMCI0	$P2_3$/ TIOCD3/ TMCI0	$P2_3$/ TIOCD3/ TMCI0	NC
60	62	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$	$\overline{\text{WDTOVF}}$ (FWE*5)	$\overline{\text{WDTOVF}}$ (FWE*5)	$\overline{\text{WDTOVF}}$ (FWE*5)	$\overline{\text{WDTOVF}}$ (FWE*5)	NC
61	63	MD_2	MD_2	MD_2	MD_2	MD_2	MD_2	MD_2	V_{SS}
62	64	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	V_{PP}
63	65	NMI	NMI	NMI	NMI	NMI	NMI	NMI	EA
64	66	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	V_{SS}
65	67	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
66	68	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	XTAL	NC
67	69	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	NC
68	70	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
69	71	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	PF_7/ϕ	NC
70	72	$\overline{\text{AS}}$	$\overline{\text{AS}}$	PF_6	$\overline{\text{AS}}$	$\overline{\text{AS}}$	$\overline{\text{AS}}$	PF_6	NC
71	73	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PF_5	$\overline{\text{RD}}$	$\overline{\text{RD}}$	$\overline{\text{RD}}$	PF_5	NC
72	74	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	PF_4	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	$\overline{\text{HWR}}$	PF_4	NC
73	75	$\overline{\text{LWR}}$	$\overline{\text{LWR}}$	PF_3/$\overline{\text{IRQ3}}$	$\overline{\text{LWR}}$	$\overline{\text{LWR}}$	$\overline{\text{LWR}}$	PF_3/$\overline{\text{IRQ3}}$	NC
74	76	PF_2/ $\overline{\text{WAIT}}$/ $\overline{\text{IRQ2}}$	PF_2/ $\overline{\text{WAIT}}$/ $\overline{\text{IRQ2}}$	PF_2/$\overline{\text{IRQ2}}$	PF_2/ $\overline{\text{WAIT}}$/ $\overline{\text{IRQ2}}$	PF_2/ $\overline{\text{WAIT}}$/ $\overline{\text{IRQ2}}$	PF_2/ $\overline{\text{WAIT}}$/ $\overline{\text{IRQ2}}$	PF_2/$\overline{\text{IRQ2}}$	CE
75	77	PF_1/ $\overline{\text{BACK}}$/ $\overline{\text{IRQ1}}$	PF_1/ $\overline{\text{BACK}}$/ $\overline{\text{IRQ1}}$	PF_1/$\overline{\text{IRQ1}}$	PF_1/ $\overline{\text{BACK}}$/ $\overline{\text{IRQ1}}$	PF_1/ $\overline{\text{BACK}}$/ $\overline{\text{IRQ1}}$	PF_1/ $\overline{\text{BACK}}$/ $\overline{\text{IRQ1}}$	PF_1/$\overline{\text{IRQ1}}$	PG

79	81	$P4_0$/AN0	$P4_0$/AN0	$P4_0$/AN0	$P4_0$/AN0	$P4_0$/AN0	$P4_0$/AN0	$P4_0$/AN0	NC
80	82	$P4_1$/AN1	$P4_1$/AN1	$P4_1$/AN1	$P4_1$/AN1	$P4_1$/AN1	$P4_1$/AN1	$P4_1$/AN1	NC
81	83	$P4_2$/AN2	$P4_2$/AN2	$P4_2$/AN2	$P4_2$/AN2	$P4_2$/AN2	$P4_2$/AN2	$P4_2$/AN2	NC
82	84	$P4_3$/AN3	$P4_3$/AN3	$P4_3$/AN3	$P4_3$/AN3	$P4_3$/AN3	$P4_3$/AN3	$P4_3$/AN3	NC
83	85	$P4_4$/AN4	$P4_4$/AN4	$P4_4$/AN4	$P4_4$/AN4	$P4_4$/AN4	$P4_4$/AN4	$P4_4$/AN4	NC
84	86	$P4_5$/AN5	$P4_5$/AN5	$P4_5$/AN5	$P4_5$/AN5	$P4_5$/AN5	$P4_5$/AN5	$P4_5$/AN5	NC
85	87	$P4_6$/AN6/ DA0	$P4_6$/AN6/ DA0	$P4_6$/AN6/ DA0	$P4_6$/AN6/ DA0	$P4_6$/AN6/ DA0	$P4_6$/AN6/ DA0	$P4_6$/AN6/ DA0	NC
86	88	$P4_7$/AN7/ DA1	$P4_7$/AN7/ DA1	$P4_7$/AN7/ DA1	$P4_7$/AN7/ DA1	$P4_7$/AN7/ DA1	$P4_7$/AN7/ DA1	$P4_7$/AN7/ DA1	NC
87	89	AV_{ss}	AV_{ss}	AV_{ss}	AV_{ss}	AV_{ss}	AV_{ss}	AV_{ss}	V_{ss}
88	90	V_{ss}	V_{ss}	V_{ss}	V_{ss}	V_{ss}	V_{ss}	V_{ss}	V_{ss}
89	91	$P2_4$/ TIOCA4/ TMRI1	$P2_4$/ TIOCA4/ TMRI1	$P2_4$/ TIOCA4/ TMRI1	$P2_4$/ TIOCA4/ TMRI1	$P2_4$/ TIOCA4/ TMRI1	$P2_4$/ TIOCA4/ TMRI1	$P2_4$/ TIOCA4/ TMRI1	NC
90	92	$P2_5$/ TIOCB4/ TMCI1	$P2_5$/ TIOCB4/ TMCI1	$P2_5$/ TIOCB4/ TMCI1	$P2_5$/ TIOCB4/ TMCI1	$P2_5$/ TIOCB4/ TMCI1	$P2_5$/ TIOCB4/ TMCI1	$P2_5$/ TIOCB4/ TMCI1	NC
91	93	$P2_6$/ TIOCA5/ TMO0	$P2_6$/ TIOCA5/ TMO0	$P2_6$/ TIOCA5/ TMO0	$P2_6$/ TIOCA5/ TMO0	$P2_6$/ TIOCA5/ TMO0	$P2_6$/ TIOCA5/ TMO0	$P2_6$/ TIOCA5/ TMO0	NC
92	94	$P2_7$/ TIOCB5/ TMO1	$P2_7$/ TIOCB5/ TMO1	$P2_7$/ TIOCB5/ TMO1	$P2_7$/ TIOCB5/ TMO1	$P2_7$/ TIOCB5/ TMO1	$P2_7$/ TIOCB5/ TMO1	$P2_7$/ TIOCB5/ TMO1	NC
93	95	PG_0/ $\overline{IRQ6}$/ $\overline{ADTRG}$	PG_0/ $\overline{IRQ6}$/ $\overline{ADTRG}$	PG_0/ $\overline{IRQ6}$/ $\overline{ADTRG}$	PG_0/ $\overline{IRQ6}$/ $\overline{ADTRG}$	PG_0/ $\overline{IRQ6}$/ $\overline{ADTRG}$	PG_0/ $\overline{IRQ6}$/ $\overline{ADTRG}$	PG_0/ $\overline{IRQ6}$/ $\overline{ADTRG}$	NC
94	96	PG_1/$\overline{IRQ7}$	PG_1/$\overline{IRQ7}$	PG_1/$\overline{IRQ7}$	PG_1/$\overline{CS3}$/ $\overline{IRQ7}$	PG_1/$\overline{CS3}$/ $\overline{IRQ7}$	PG_1/$\overline{CS3}$/ $\overline{IRQ7}$	PG_1/$\overline{IRQ7}$	NC
95	97	PG_2	PG_2	PG_2	PG_2/$\overline{CS2}$	PG_2/$\overline{CS2}$	PG_2/$\overline{CS2}$	PG_2	NC
96	98	PG_3	PG_3	PG_3	PG_3/$\overline{CS1}$	PG_3/$\overline{CS1}$	PG_3/$\overline{CS1}$	PG_3	NC
97	99	PG_4/$\overline{CS0}$	PG_4/$\overline{CS0}$	PG_4	PG_4/$\overline{CS0}$	PG_4/$\overline{CS0}$	PG_4/$\overline{CS0}$	PG_4	NC
98	100	V_{cc}	V_{cc}	V_{cc}	V_{cc}	V_{cc}	V_{cc}	V_{cc}	V_{cc}

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version.
2. Modes 2, 3, 6, and 7 are not available on the ROMless version.
3. ZTAT version only.
4. F-ZTAT version only.
5. The FWE pin is only used on the F-ZTAT version. It cannot be used as a $\overline{W}$
on the F-ZTAT version.

Type	Symbol	TFP-100G	FP-100A	I/O	Name and Function
Power supply	V_{CC}	40, 65, 98	42, 67, 100	Input	Power supply: For connec power supply. All V_{CC} pins connected to the system p supply.
	V_{SS}	7, 18, 31, 49, 68, 88	9, 20, 33, 51, 70, 90	Input	Ground: For connection to (0 V). All V_{SS} pins should b connected to the system p supply (0 V).
Clock	XTAL	66	68	Input	Connects to a crystal osci See section 18, Clock Pul Generator, for typical conn diagrams for a crystal osc external clock input.
	EXTAL	67	69	Input	Connects to a crystal osci The EXTAL pin can also i external clock. See section 18, Clock Pul Generator, for typical conn diagrams for a crystal osc external clock input.
	ϕ	69	71	Output	System clock: Supplies th clock to an external device

should not be changed w
H8S/2345 Group is opera

- F-ZTAT Version

FWE	MD_2	MD_1	MD_0
0	0	0	0
			1
		1	0
			1
	1	0	0
			1
		1	0
			1
1	0	0	0
			1
		1	0
			1
	1	0	0
			1
		1	0
			1

				0	0 0 —
					1 Mo
					1 0 Mo
					1 Mo
				1	0 0 Mo
					1 Mo
					1 0 Mo
					1 Mo
				Note: *	Not used on R version.
System control	$\overline{\text{RES}}$	62	64	Input	Reset input: When this pin low, the chip is reset. The reset can be selected acc the NMI input level. At pow NMI pin input level should high.
	$\overline{\text{STBY}}$	64	66	Input	Standby: When this pin is a transition is made to har standby mode.
	$\overline{\text{BREQ}}$	76	78	Input	Bus request: Used by an e bus master to issue a bus the H8S/2345 Group.
	$\overline{\text{BACK}}$	75	77	Output	Bus request acknowledge that the bus has been rele external bus master.
	FWE*[1]	60	62	Input	Flash write enable: Enable disables writing to flash m

		73 to 76	75 to 78		
Address bus	A_{23} to A_0	2, 1, 100, 99, 53 to 50, 48 to 41, 39 to 32	4 to 1, 55 to 52, 50 to 43, 41 to 34	Output	Address bus: These pins address.
Data bus	D_{15} to D_0	30 to 19, 17 to 14	32 to 21, 19 to 16	I/O	Data bus: These pins co bidirectional data bus.
Bus control	$\overline{CS3}$ to $\overline{CS0}$	94 to 97	96 to 99	Output	Chip select: Signals for s areas 3 to 0.
	$\overline{AS}$	70	72	Output	Address strobe: When th it indicates that address the address bus is enabl
	$\overline{RD}$	71	73	Output	Read: When this pin is lo indicates that the externa space can be read.
	$\overline{HWR}$	72	74	Output	High write: A strobe signa to external space and ind the upper half (D_{15} to D_8) bus is enabled.
	$\overline{LWR}$	73	75	Output	Low write: A strobe signa to external space and ind the lower half (D_7 to D_0) c bus is enabled.
	$\overline{WAIT}$	74	76	Input	Wait: Requests insertion state in the bus cycle wh accessing external 3-stat space.

	TIOCD0				compare output, or PWM
	TIOCA1, TIOCB1	3, 4	5, 6	I/O	Input capture/ output comp A1 and B1: The TGR1A a input capture input or outp compare output, or PWM
	TIOCA2, TIOCB2	5, 6	7, 8	I/O	Input capture/ output comp A2 and B2: The TGR2A a input capture input or outp compare output, or PWM
	TIOCA3, TIOCB3, TIOCC3, TIOCD3	54 to 56, 59	56 to 58, 61	I/O	Input capture/ output comp A3 to D3: The TGR3A to T input capture input or outp compare output, or PWM
	TIOCA4, TIOCB4	89, 90	91, 92	I/O	Input capture/ output comp A4 and B4: The TGR4A a input capture input or outp compare output, or PWM
	TIOCA5, TIOCB5	91, 92	93, 94	I/O	Input capture/ output comp A5 and B5: The TGR5A a input capture input or outp compare output, or PWM
8-bit timer	TMO0, TMO1	91, 92	93, 94	Output	Compare match output: Th compare match output pin
	TMCI0, TMCI1	59, 90	61, 92	Input	Counter external clock inp pins for the external clock counter.
	TMRI0, TMRI1	56, 89	58, 91	Input	Counter external reset inp counter reset input pins.
Watchdog timer (WDT)	$\overline{\text{WDTOVF}}$*2	60	62	Output	Watchdog timer overflows counter overflows signal o watchdog timer mode.

	SCK1 SCK0	13, 12	15, 14	I/O	Serial clock (channel 0, 1 Clock I/O pins.
A/D converter	AN7 to AN0	86 to 79	88 to 81	Input	Analog 7 to 0: Analog inp
	$\overline{\text{ADTRG}}$	93	95	Input	A/D conversion external Pin for input of an extern start A/D conversion.
D/A converter	DA1, DA0	86, 85	88, 87	Output	Analog output: D/A conv output pins.
A/D converter and D/A converters	AV_{CC}	77	79	Input	This is the power supply A/D converter and D/A c When the A/D converter converter are not used, t should be connected to t power supply (+5 V).
	AV_{SS}	87	89	Input	This is the ground pin for converter and D/A conve This pin should be conne system power supply (0
	V_{ref}	78	80	Input	This is the reference volt pin for the A/D converter converter. When the A/D converter converter are not used, t should be connected to t power supply (+5 V).
I/O ports	$P1_7$ to $P1_0$	6 to 1, 100, 99	8 to 1	I/O	Port 1: An 8-bit I/O port. output can be designated by means of the port 1 da register (P1DDR).
	$P2_7$ to $P2_0$	92 to 89, 59, 56 to 54	94 to 91, 61, 58 to 56	I/O	Port 2: An 8-bit I/O port. output can be designated by means of the port 2 da register (P2DDR).

$P4_0$				
PA_3 to PA_0	53 to 50	55 to 52	I/O	Port A: An 4-bit I/O port. I output can be designated by means of the port A da register (PADDR).
PB_7 to PB_0	48 to 41	50 to 43	I/O	Port B: An 8-bit I/O port. I output can be designated by means of the port B da register (PBDDR).
PC_7 to PC_0	39 to 32	41 to 34	I/O	Port C: An 8-bit I/O port. I output can be designated by means of the port C da register (PCDDR).
PD_7 to PD_0	30 to 23	32 to 25	I/O	Port D: An 8-bit I/O port. I output can be designated by means of the port D da register (PDDDR).
PE_7 to PE_0	22 to 19, 17 to 14	24 to 21, 19 to 16	I/O	Port E: An 8-bit I/O port. I output can be designated by means of the port E da register (PEDDR).
PF_7 to PF_0	69 to 76	71 to 78	I/O	Port F: An 8-bit I/O port. I output can be designated by means of the port F da register (PFDDR).
PG_4 to PG_0	97 to 93	99 to 95	I/O	Port G: A 5-bit I/O port. In output can be designated by means of the port G da register (PGDDR).

Notes: 1. F-ZTAT version only.
2. Applies to ZTAT, mask ROM, and ROMless versions only.

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 3 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)

— 32 ÷ 16-bit register-register divide : 1000 ns

- Two CPU operating modes
  — Normal mode (Supported on ZTAT, mask ROM, and ROMless versions only)
  — Advanced mode
- 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 belo

- 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
  H8S/2600 CPU.
- Number of execution states
  The number of execution 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, powe
state, etc., depending on the product.

ROM, and ROMless versions only)
  - Advanced mode supports a maximum 16-Mbyte address space.
- Enhanced addressing
  - The addressing modes have been enhanced to make effective use of the 16-Mb 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 enhanceme

- Additional control register
  - One 8-bit control register has been added.
- Enhanced instructions
  - Addressing modes of bit-manipulation instructions have been enhanced.
  - Two-bit shift 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.

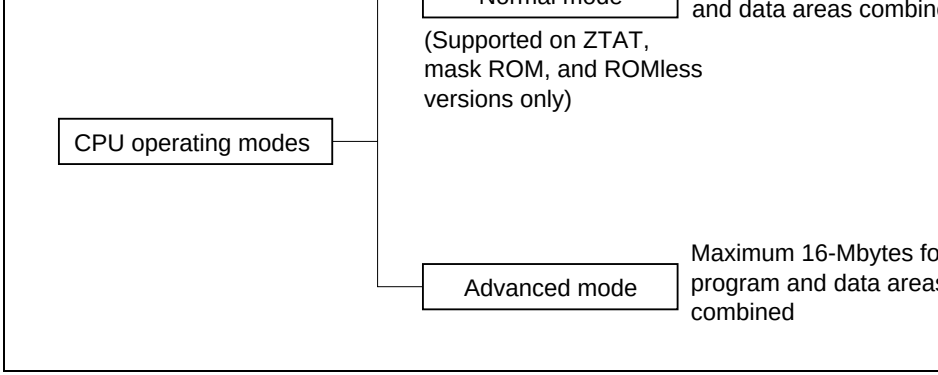


Figure 2.1 CPU Operating Modes

(1) Normal Mode (ZTAT, Mask ROM, and ROMless Versions Only)

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 regi
the upper 16-bit segments of 32-bit registers. When En is used as a 16-bit register it can
any value, even when the corresponding general register (Rn) is used as an address regi
general register is referenced in the register indirect addressing mode with pre-decreme
or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corr
extended register (En) will be affected.

Instruction Set: All instructions and addressing modes can be used. Only the lower 16
effective addresses (EA) are valid.

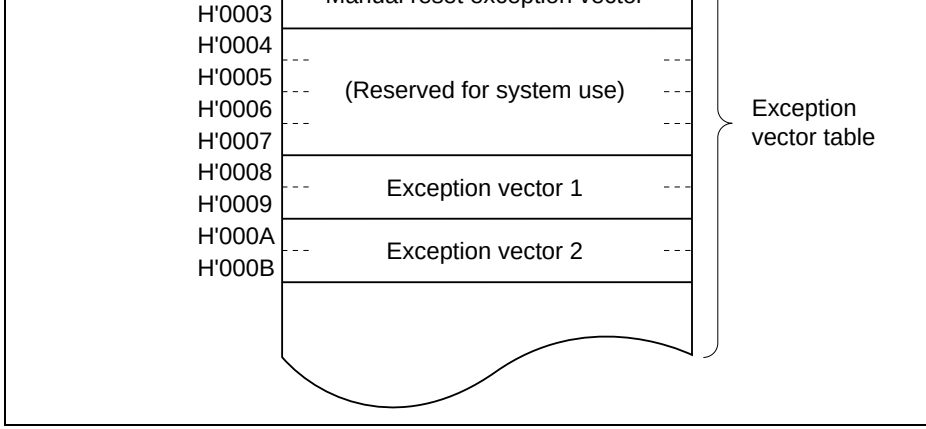
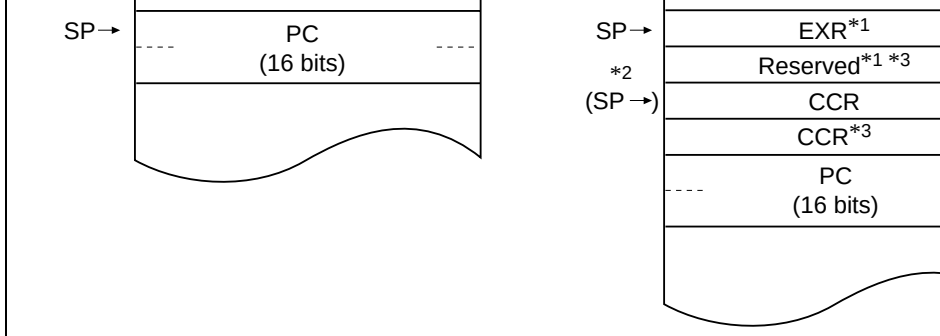


Figure 2.2 Exception Vector Table (Normal Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instru an 8-bit absolute address included in the instruction code to specify a memory operand contains a branch address. In normal mode the operand is a 16-bit word operand, prov bit branch address. Branch addresses can be stored in the top area from H'0000 to H'0 that this area is also used for the exception vector table.



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 (arch a maximum 16-Mbyte program area and a maximum 4-Gbyte data area, with a maximu Gbytes for program and data areas combined).

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit regi the upper 16-bit segments of 32-bit registers or address registers.

Instruction Set: All instructions and addressing modes can be used.

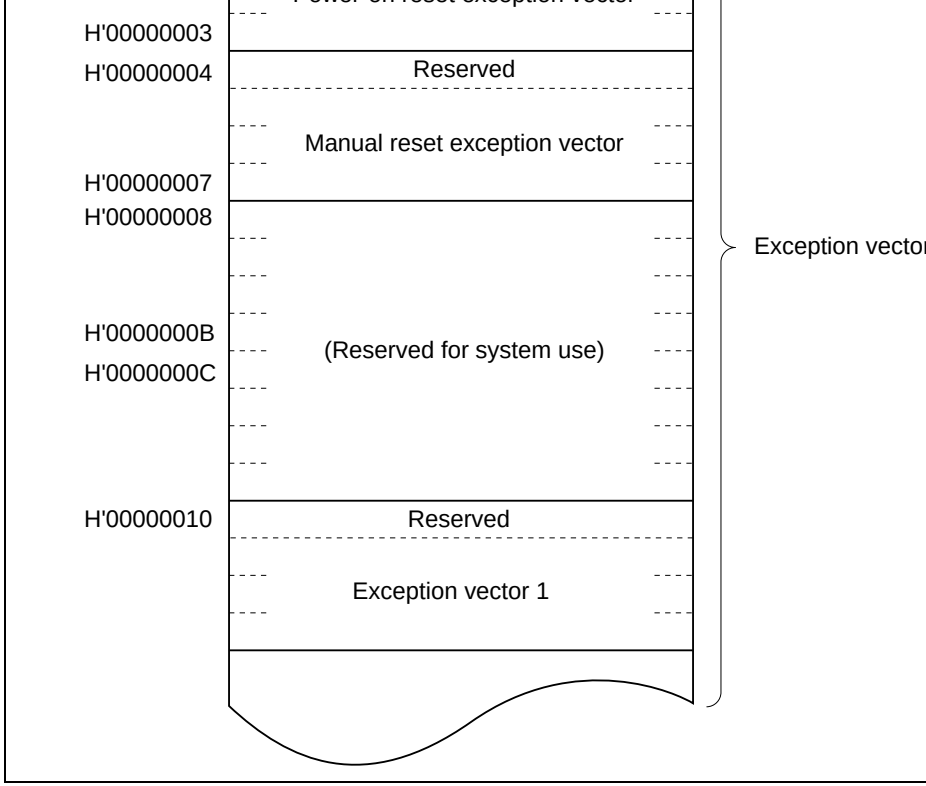
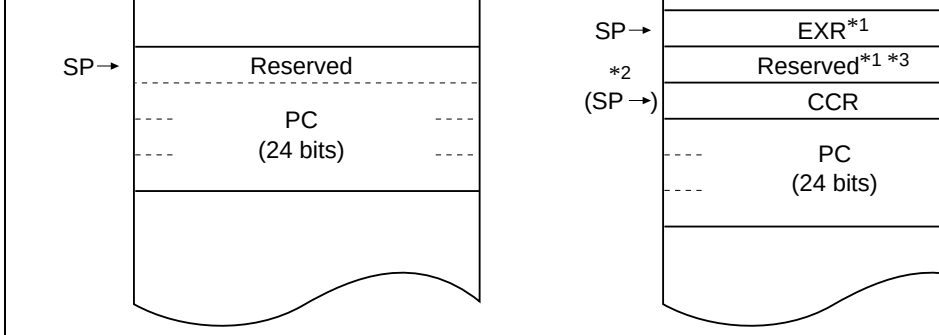


Figure 2.4 Exception Vector Table (Advanced Mode)

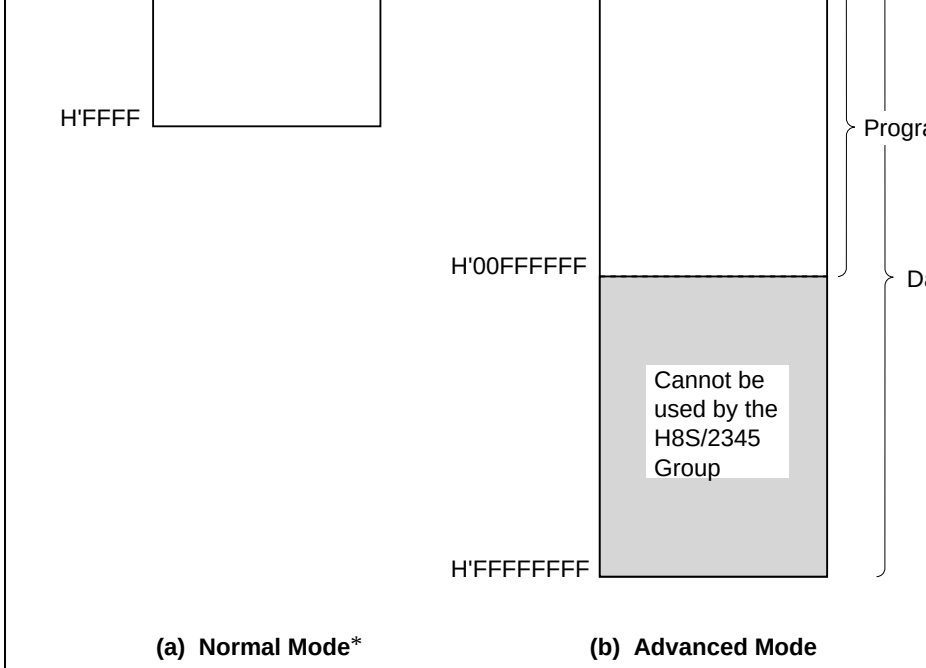
The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instru
an 8-bit absolute address included in the instruction code to specify a memory operand
contains a branch address. In advanced mode the operand is a 32-bit longword operan
a 32-bit branch address. The upper 8 bits of these 32 bits are a reserved area that is reg
H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. No
first part of this range is also the exception vector table.



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

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



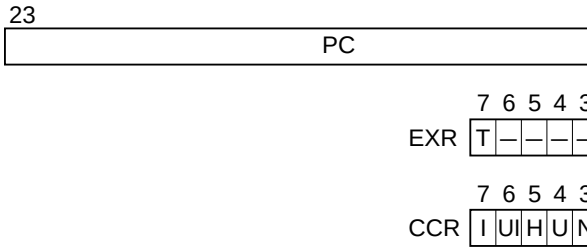
Note: * ZTAT, mask ROM, and ROMless versions only.

Figure 2.6 Memory Map

General Registers (Rn) and Extended Registers (En)

	15	07	07
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/2345 Group, this bit cannot be used as an interrupt mask.

Figure 2.7 CPU Registers

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
RL (R0L to R7L). These registers are functionally equivalent, providing a maximum s
registers.

Figure 2.8 illustrates the usage of the general registers. The usage of each register can
independently.

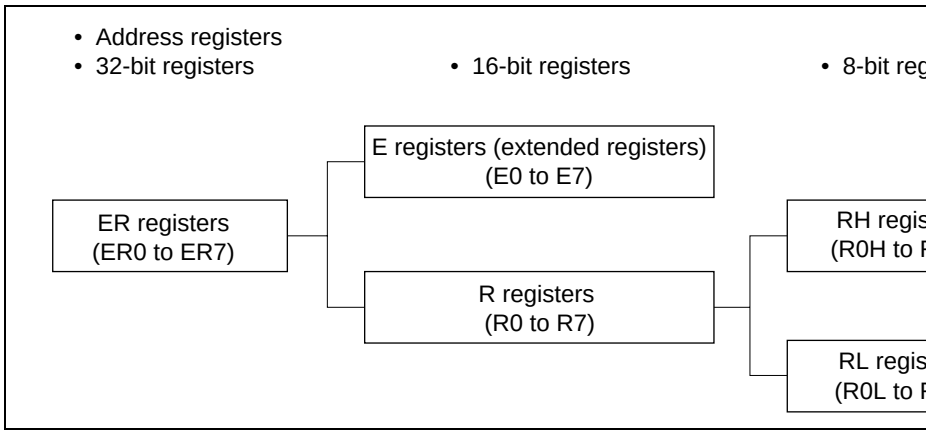


Figure 2.8 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-r
function, and is used implicitly in exception handling and subroutine calls. Figure 2.9
stack.

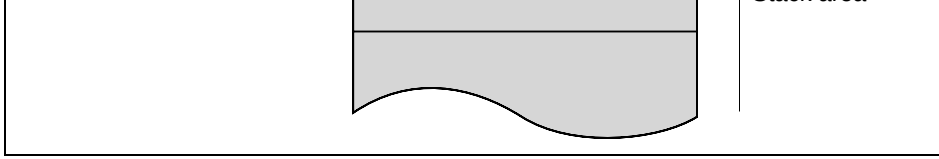


Figure 2.9 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC), 8-bit extended control registe 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. T of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. 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 three interrupt mask bits (I2 to I0).

Bit 7—Trace Bit (T): Selects trace mode. When this bit is cleared to 0, instructions are in sequence. When this bit is set to 1, a trace exception is generated each time an instru 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 le 7). For details, refer to section 5, Interrupt Controller.

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

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software us
LDC, STC, ANDC, ORC, and XORC instructions. With the H8S/2345 Group, this bit
used as an interrupt mask bit.

Bit 5—Half-Carry Flag (H): When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B
instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cle
otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, th
set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the AD
SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a ca
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, AND
XORC instructions.

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

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

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

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

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions, to store the value shifted out of the end bit

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
on the flag bits, refer to appendix A.1, Instruction List.

and the general registers are not initialized. In particular, the stack pointer (ER7) is not
The stack pointer should therefore be initialized by an MOV.L instruction executed imm
after a reset.

Figure 2.10 shows the data formats in general registers.

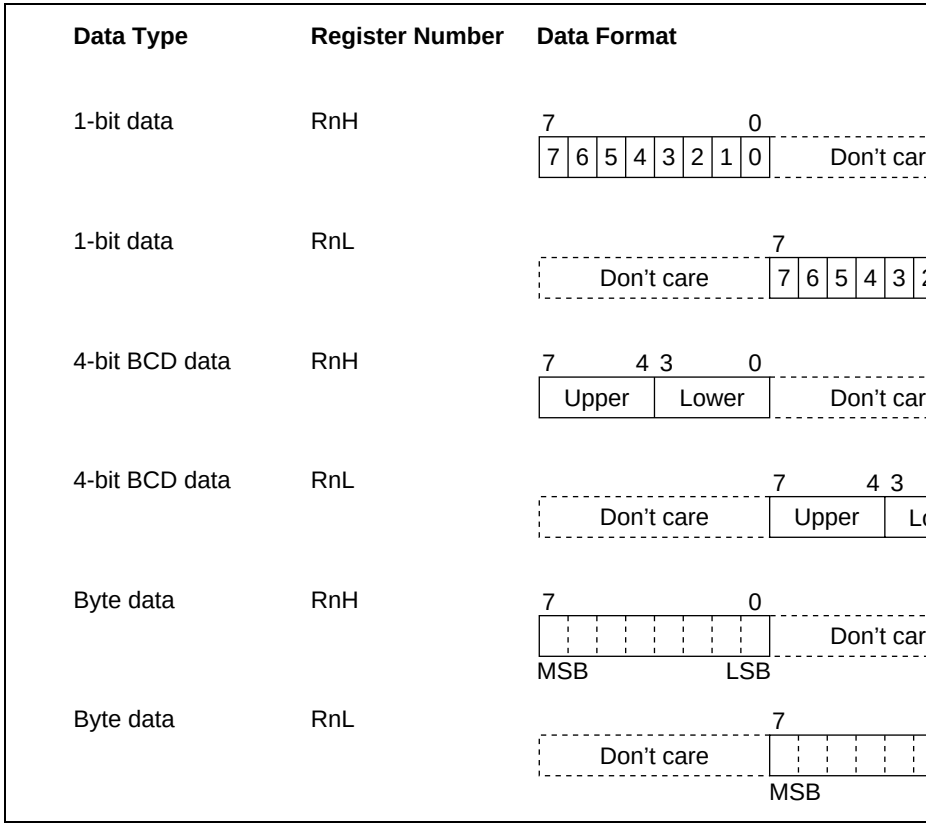
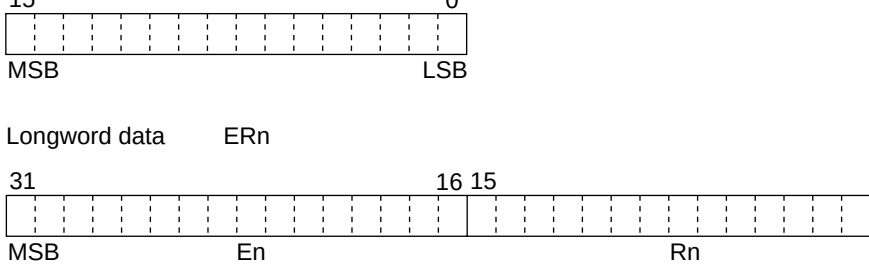


Figure 2.10 General Register Data Formats



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 2.10 General Register Data Formats (cont)

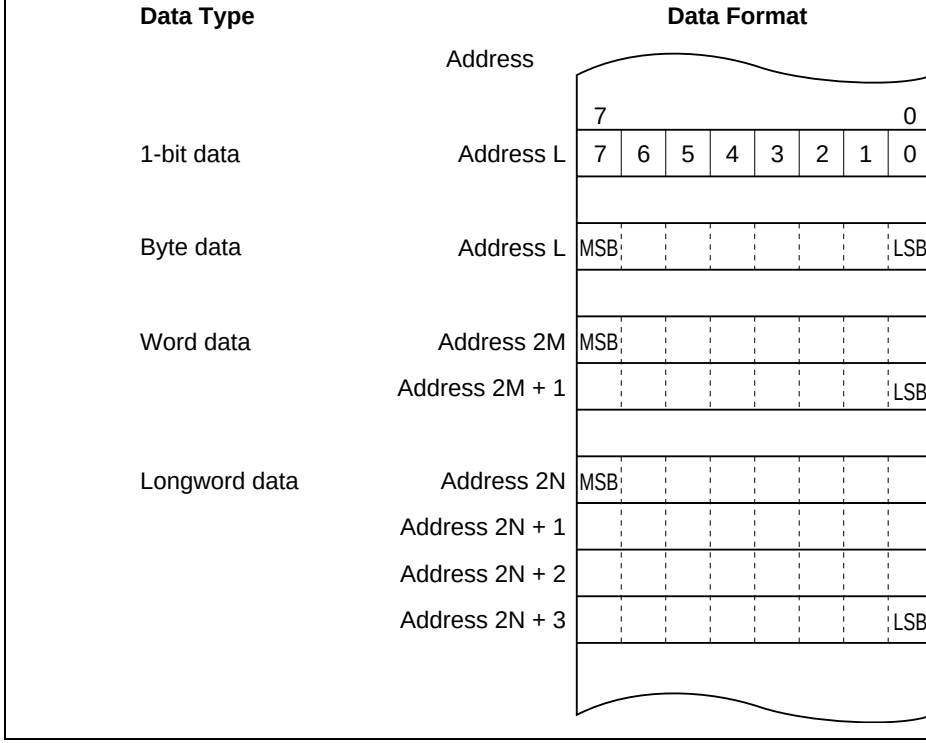


Figure 2.11 Memory Data Formats

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

Function	Instructions	Size
Data transfer	MOV	BWL
	POP*1, PUSH*1	WL
	LDM, STM	L
	MOVFPE, MOVTPE*3	B
Arithmetic operations	ADD, SUB, CMP, NEG	BWL
	ADDX, SUBX, DAA, DAS	B
	INC, DEC	BWL
	ADDS, SUBS	L
	MULXU, DIVXU, MULXS, DIVXS	BW
	EXTU, EXTS	WL
	TAS	B
Logic operations	AND, OR, XOR, NOT	BWL
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	BWL
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B
Branch	Bcc*2, JMP, BSR, JSR, RTS	—
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—
Block data transfer	EEPMOV	—

Legend: B: Byte
W: Word
L: Longword

Notes: 1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and ERn, @-SP.
2. Bcc is the general name for conditional branch instructions.
3. Cannot be used in the H8S/2345 Group.

Function	Instruction	#xx	Rn	@ERn	@(d:16,ERn	@(d:32,ERn	@–ERn/@E	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	—	BWL	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—
	LDM, STM	—	—	—	—	—	—	—	—	—	—	—	—
	MOVFPE*, MOVTPE*	—	—	—	—	—	—	—	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	—	—	B	—	—	—	—	—	—	—	—	—
Logic operations	AND, OR, XOR	BWL	BWL	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—
Shift		—	BWL	—	—	—	—	—	—	—	—	—	—
Bit manipulation		—	B	B	—	—	—	B	B	—	B	—	—
Branch	Bcc, BSR	—	—	—	—	—	—	—	—	—	—	○	○
	JMP, JSR	—	—	—	—	—	—	—	—	○	—	—	—
	RTS	—	—	—	—	—	—	—	—	—	—	—	—

	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		—	—	—	—	—	—	—	—	—	—	—	—

Legend:

B: Byte

W: Word

L: Longword

Note: * Cannot be used in the H8S/2345 Group.

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 r
to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

	POP	W/L	@SP+ → Rn Pops a register from the stack. POP.W Rn is ide MOV.W @SP+, Rn. POP.L ERn is identical to M @SP+, ERn.
	PUSH	W/L	Rn → @–SP Pushes a register onto the stack. PUSH.W Rn is MOV.W Rn, @–SP. PUSH.L ERn is identical to ERn, @–SP.
	LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the sta
	STM	L	Rn (register list) → @–SP Pushes two or more general registers onto the s
Arithmetic operations	ADD SUB	B/W/L	Rd ± Rs → Rd, Rd ± #IMM → Rd Performs addition or subtraction on data in two g registers, or on immediate data and data in a ge register. (Immediate byte data cannot be subtrac byte data in a general register. Use the SUBX or instruction.)
	ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry or bo byte data in two general registers, or on immedia and data in a general register.
	INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by (Byte operands can be incremented or decreme 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 32-bit register.

MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two g registers: either 8 bits × 8 bits → 16 bits or 16 b 16 bits → 32 bits.
DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two gene registers: either 16 bits ÷ 8 bits → 8-bit quotien remainder or 32 bits ÷ 16 bits → 16-bit quotient remainder.
DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two genera either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit
CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data i general register or with immediate data, and se according to the result.
NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic compl data in a general register.
EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to w the lower 16 bits of a 32-bit register to longword padding with zeros on the left.
EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to w the lower 16 bits of a 32-bit register to longword extending the sign bit.
TAS	B	@ERd – 0, 1 → (<bit 7> of @Erd) Tests memory contents, and sets the most sign (bit 7) to 1.

			Performs a logical exclusive OR operation on a register and another general register or immedia
	NOT	B/W/L	¬ (Rd) → (Rd) Takes the one's complement of general register
Shift operations	SHAL SHAR	B/W/L	Rd (shift) → Rd Performs an arithmetic shift on general register c 1-bit or 2-bit shift is possible.
	SHLL SHLR	B/W/L	Rd (shift) → Rd Performs a logical shift on general register conte 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 ca 1-bit or 2-bit rotation is possible.
Bit-manipulation instructions	BSET	B	1 → (<bit-No.> of <EAd>) Sets a specified bit in a general register or mem to 1. The bit number is specified by 3-bit immedi the lower three bits of a general register.
	BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in a general register or me operand to 0. The bit number is specified by 3-b 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 me operand. The bit number is specified by 3-bit imr 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 mem operand and sets or clears the Z flag accordingly number is specified by 3-bit immediate data or th three bits of a general register.

BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a gene or memory operand and stores the result in the
BIOR	B	C ∨ ¬ (<bit-No.> of <EAd>) → C ORs the carry flag with the inverse of a specifie general register or memory operand and stores the carry flag. The bit number is specified by 3-bit immediate
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with a specified bi general register or memory operand and stores the carry flag.
BIXOR	B	C ⊕ ¬ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with the inverse o bit in a general register or memory operand an result in the carry flag. The bit number is specified by 3-bit immediate
BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in a general register o operand to the carry flag.
BILD	B	¬ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a gen or memory operand to the carry flag. The bit number is specified by 3-bit immediate
BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit register or memory operand.
BIST	B	¬ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to bit in a general register or memory operand. The bit number is specified by 3-bit immediate

	BLS	Low or same	C ∨ Z
	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
	BLT	Less than	N ⊕ V
	BGT	Greater than	Z∨(N ⊕
	BLE	Less or equal	Z∨(N ⊕
	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
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 immediat CCR or EXR. Although CCR and EXR are 8-bit r word-size transfers are performed between them memory. The upper 8 bits are valid.

	ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with in data.
	XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR conte immediate data.
	NOP	—	PC + 2 → PC Only increments the program counter.
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 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 soo transfer is completed.

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

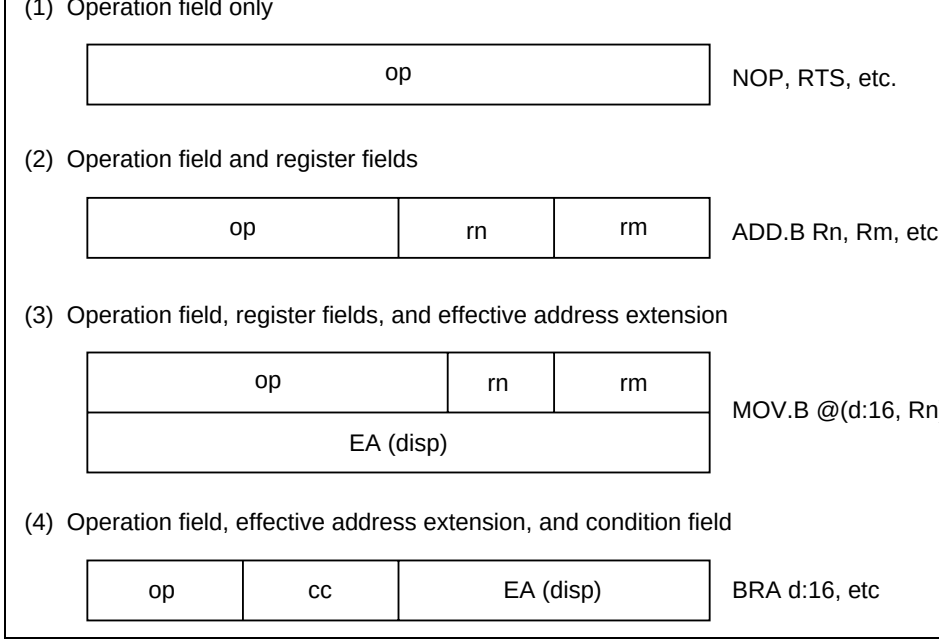


Figure 2.12 Instruction Formats (Examples)

(1) Operation Field: Indicates the function of the instruction, the addressing mode, an operation to be carried out on the operand. The operation field always includes the first the instruction. Some instructions have two operation fields.

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

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

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

absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BI
BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in th

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/@
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
general register containing the operand. R0H to R7H and R0L to R7L can be specifiec
registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can b
as 32-bit registers.

(2) Register Indirect—@ERn: The register field of the instruction code specifies an
register (ERn) which contains the address of the operand on memory. If the address is
instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to b

(3) Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn): A 16-b
displacement contained in the instruction is added to an address register (ERn) specifi
register field of the instruction, and the sum gives the address of a memory operand. A
displacement is sign-extended when added.

The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the reg in the instruction code, and the result becomes the address of a memory operand. Th also stored in the address register. The value subtracted is 1 for byte access, 2 for w instruction, or 4 for longword transfer instruction. For word or longword transfer in the register value should be even.

(5) Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32: The instruction code c absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8 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 b (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute a access the entire address space.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The 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'FF
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'00 H'FF8000 to H'FF
	32 bits (@aa:32)		H'000000 to H'FF
Program instruction address	24 bits (@aa:24)		

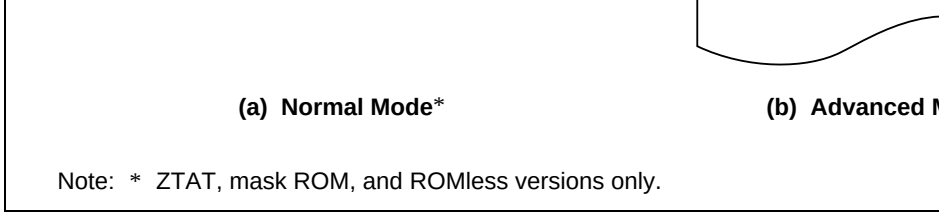
Note: * ZTAT, mask ROM, and ROMless versions only.

BSR instructions. An 8-bit or 16-bit displacement contained in the instruction is sign-
added to the 24-bit PC contents to generate a branch address. Only the lower 24 bits o
address are valid; the upper 8 bits are all assumed to be 0 (H'00). The PC value to whi
displacement is added is the address of the first byte of the next instruction, so the pos
branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes
+16384 words) from the branch instruction. The resulting value should be an even nu

(8) Memory Indirect—@@aa:8: This mode can be used by the JMP and JSR instru
instruction code contains an 8-bit absolute address specifying a memory operand. This
operand contains a branch address. The upper bits of the absolute address are all assu
so the address range is 0 to 255 (H'0000 to H'00FF in normal mode, H'000000 to H'00
advanced mode). In normal mode* the memory operand is a word operand and the bra
is 16 bits long. In advanced mode the memory operand is a longword operand, the firs
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 furth
refer to section 4, Exception Handling.

Note: * ZTAT, mask ROM, and ROMless versions only.



(a) Normal Mode* **(b) Advanced M**

Note: * ZTAT, mask ROM, and ROMless versions only.

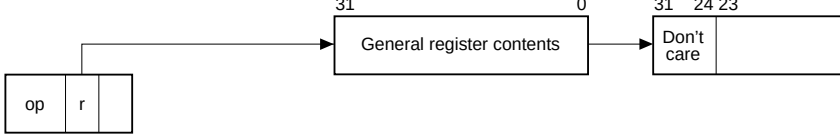
Figure 2.13 Branch Address Specification in Memory Indirect Mode

If an odd address is specified in word or longword memory access, or as a branch addre least significant bit is regarded as 0, causing data to be accessed or instruction code to b at the address preceding the specified address. (For further information, see section 2.5 Data Formats.)

2.7.2 Effective Address Calculation

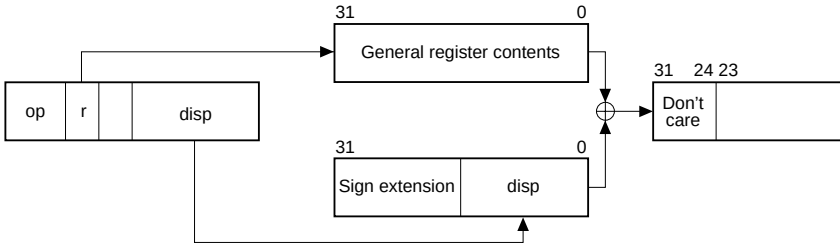
Table 2.6 indicates how effective addresses are calculated in each addressing mode. In mode* the upper 8 bits of the effective address are ignored in order to generate a 16-bit

Note: * ZTAT, mask ROM, and ROMless versions only.



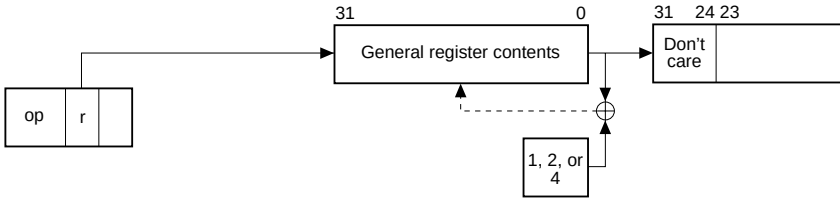
3 Register indirect with displacement

@(d:16, ERn) or @(d:32, ERn)

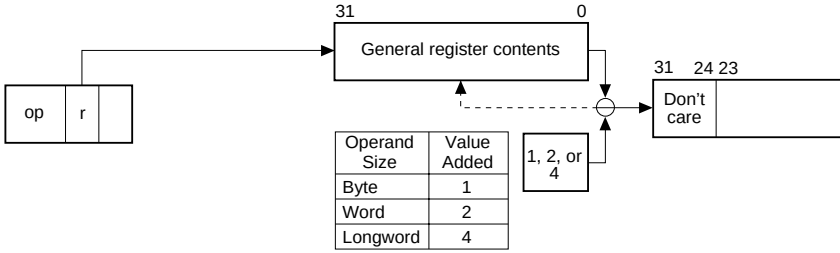


4 Register indirect with post-increment or pre-decrement

- Register indirect with post-increment @ERn+



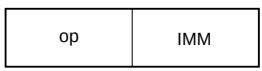
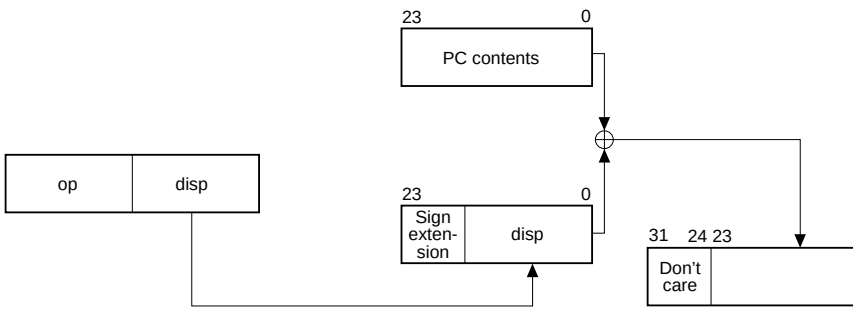
- Register indirect with pre-decrement @–ERn

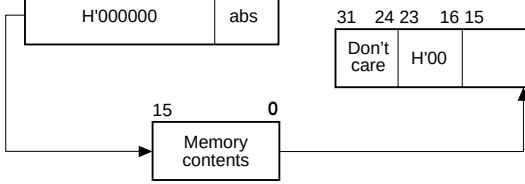


Operand Size	Value Added
Byte	1
Word	2
Longword	4

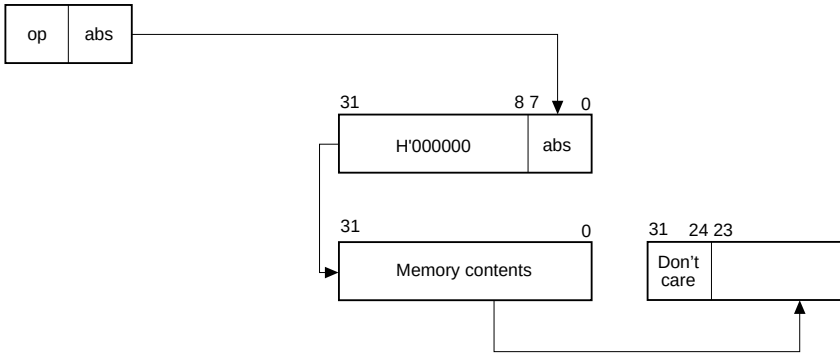
@aa:24

@aa:32

6	Immediate #xx:8/#xx:16/#xx:32 	Operand is imm data.
7	Program-counter relative @(d:8, PC)/@(d:16, PC) 	



- Advanced mode



Note: * ZTAT, mask ROM, and ROMless versions only.

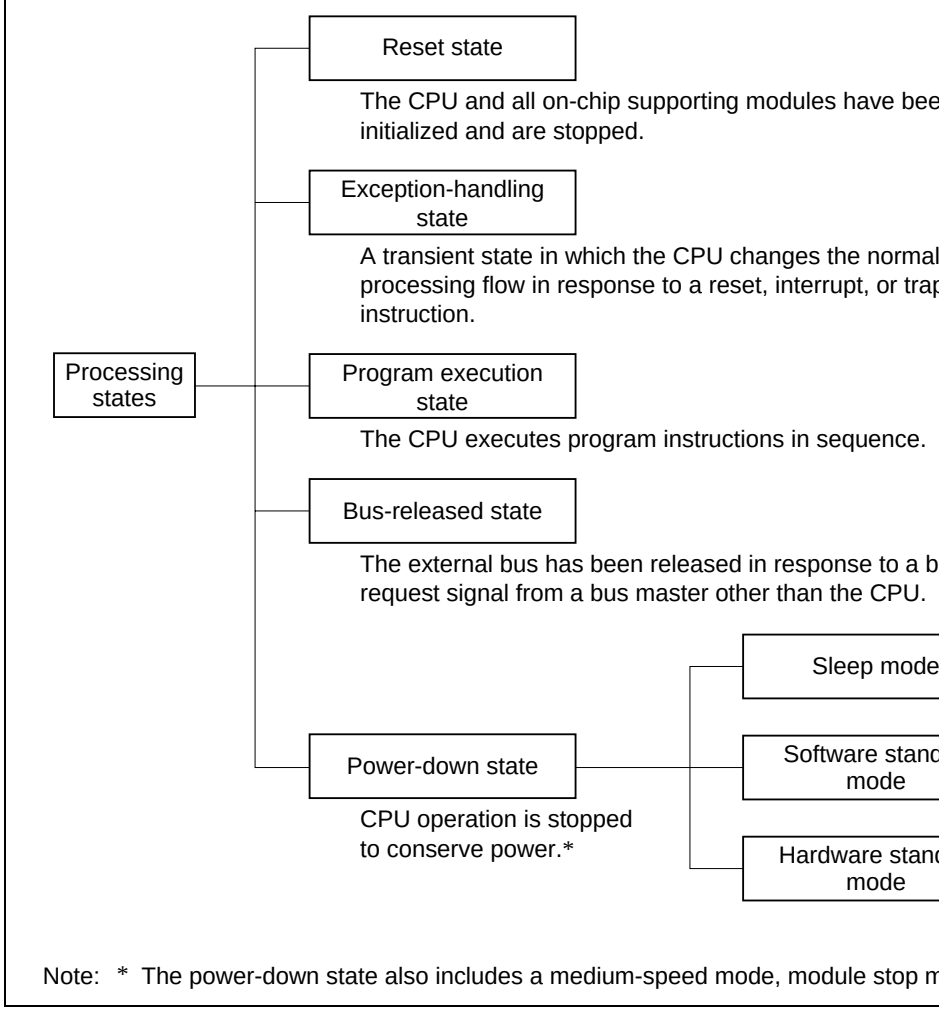
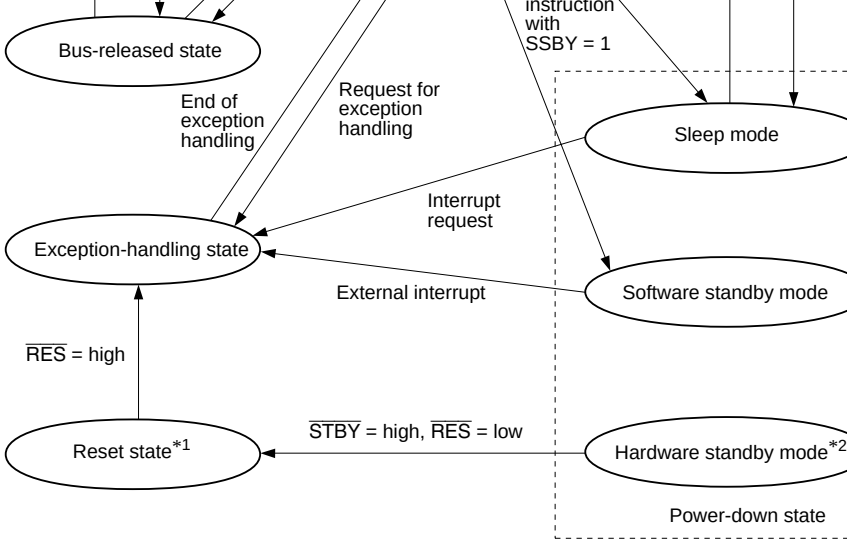


Figure 2.14 Processing States



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs when goes low. A transition can also be made to the reset state when the watchdog timer overfl
2. From any state, a transition to hardware standby mode occurs when STBY goes low.

Figure 2.15 State Transitions

2.8.2 Reset State

When the RES input goes low all current processing stops and the CPU enters the rese CPU enters the power-on reset state when the NMI pin is high, or the manual reset sta NMI pin is low. All interrupts are masked in the reset state. Reset exception handling the RES signal changes from low to high.

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

indicates the types of exception handling and their priority. Trap instruction exception h
always accepted, in the program execution state.

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

Table 2.7 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception H
High ▲	Reset	Synchronized with clock	Exception handling s immediately after a lo transition at the $\overline{\text{RES}}$ when the watchdog t overflows.
	Trace	End of instruction execution or end of exception-handling sequence*1	When the trace (T) b 1, the trace starts at the current instructio exception-handling s
	Interrupt	End of instruction execution or end of exception-handling sequence*2	When an interrupt is exception handling st end of the current ins current exception-ha sequence
	Trap instruction	When TRAPA instruction is executed	Exception handling s a trap (TRAPA) instru executed*3
Low			

Notes: 1. Traces are enabled only in interrupt control mode 2. Trace exception-handlin executed at the end of the RTE instruction.
2. Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC i or immediately after reset exception handling.
3. Trap instruction exception handling is always accepted, in the program exec

(3) Traces

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

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

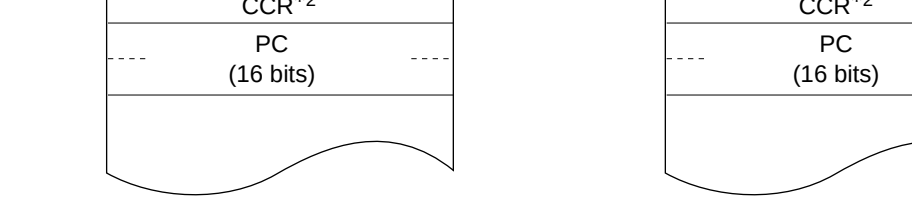
The T bit saved on the stack retains its value of 1, and when the RTE instruction is ex
return from the trace exception-handling routine, trace mode is entered again. Trace ex
handling is not executed at the end of the RTE instruction.

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

(4) Interrupt Exception Handling and Trap Instruction Exception Handling

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

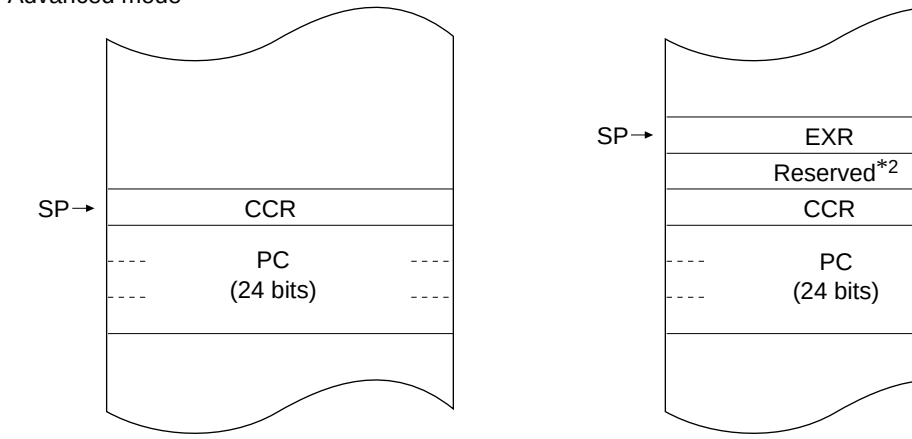
Figure 2.16 shows the stack after exception handling ends.



(a) Interrupt control mode 0

(b) Interrupt control m

Advanced mode



(c) Interrupt control mode 0

(d) Interrupt control m

Notes: 1. ZTAT, mask ROM, and ROMless versions only.
2. Ignored when returning.

Figure 2.16 Stack Structure after Exception Handling (Examples)

There is one other bus master in addition to the CPU: the data transfer controller (DT

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 mo
the CPU does not stop. There are three modes in which the CPU stops operating: sleep
software standby mode, and hardware standby mode. There are also two other power-
modes: medium-speed mode, and module stop mode. In medium-speed mode the CPU
bus masters operate on a medium-speed clock. Module stop mode permits halting of t
of individual modules, other than the CPU. For details, refer to section 19, Power-Dow

(1) Sleep Mode: A transition to sleep mode is made if the SLEEP instruction is execu
the software standby bit (SSBY) in the standby control register (SBYCR) is cleared to
mode, CPU operations stop immediately after execution of the SLEEP instruction. Th
CPU registers are retained.

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

(3) Hardware Standby Mode: A transition to hardware standby mode is made when
pin goes low. In hardware standby mode, the CPU and clock halt and all MCU operati
The on-chip supporting modules are reset, but as long as a specified voltage is supplie
RAM contents are retained.

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
word transfer instruction. Figure 2.17 shows the on-chip memory access cycle. Figure 2
the pin states.

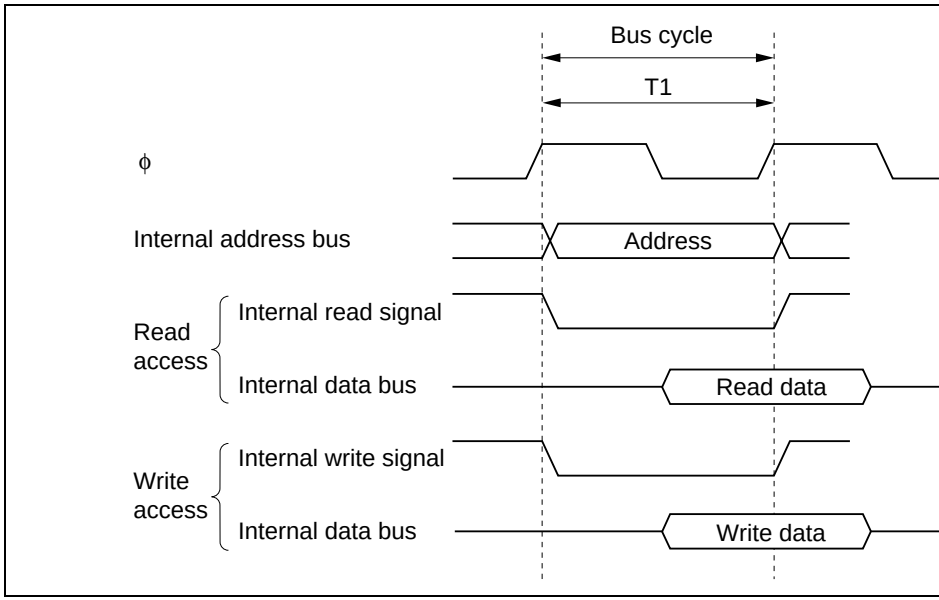


Figure 2.17 On-Chip Memory Access Cycle

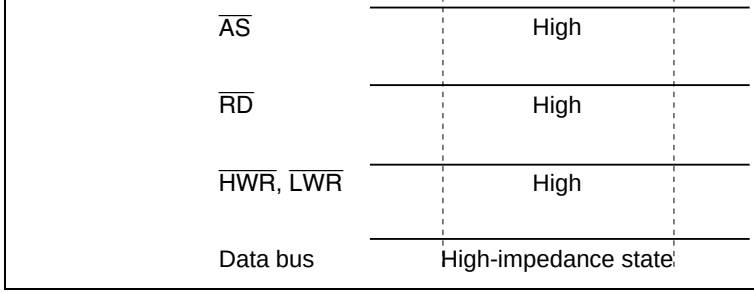


Figure 2.18 Pin States during On-Chip Memory Access

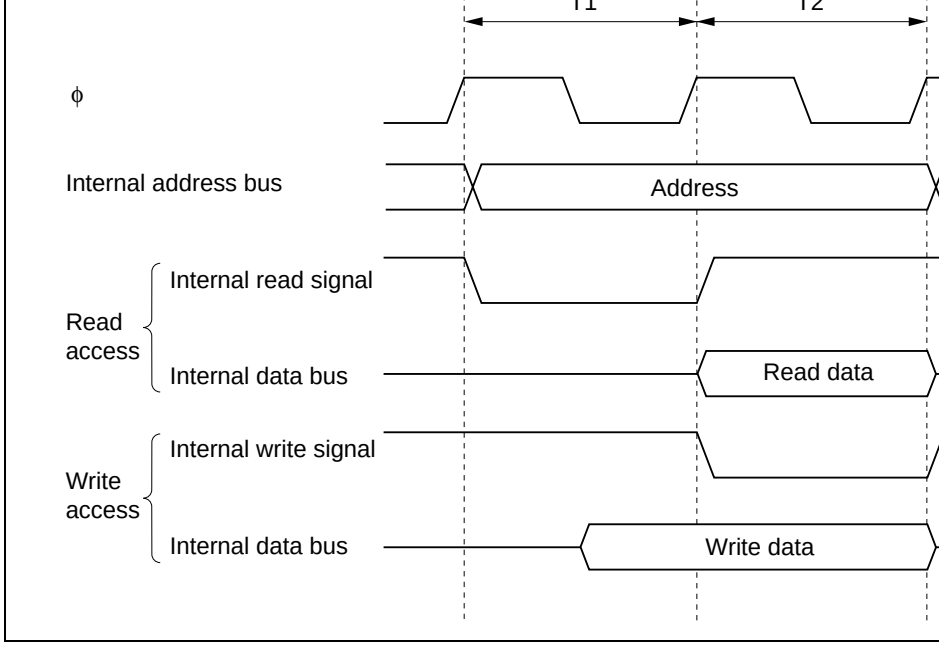


Figure 2.19 On-Chip Supporting Module Access Cycle

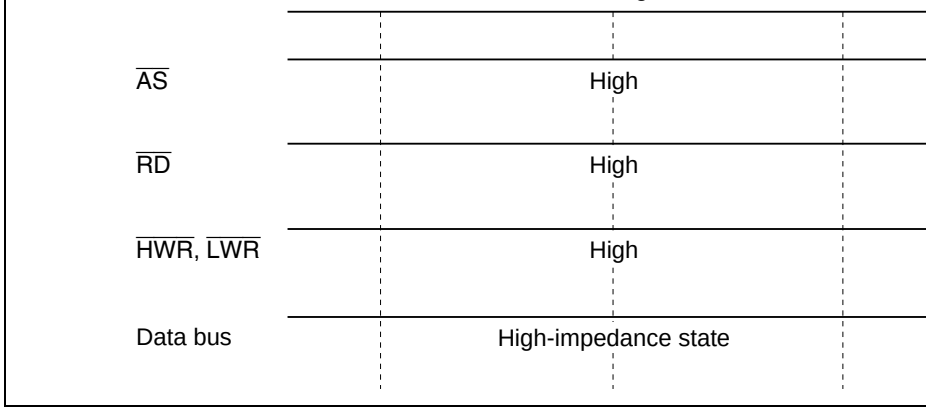


Figure 2.20 Pin States during On-Chip Supporting Module Access

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-
three-state bus cycle. In three-state access, wait states can be inserted. For further deta
section 6, Bus Controller.

are determined by the mode pin (MD_2 to MD_0) and flash write enable pin (FWE) settin
CPU operating mode and initial bus width can be selected as shown in table 3.1.

Table 3.1 lists the MCU operating modes.

Table 3.1 MCU Operating Mode Selection (F-ZTAT™ Version)

MCU Operating Mode	FWE	MD_2	MD_1	MD_0	CPU Operating Mode	Description	On-Chip ROM	Exter Initia Widtl
0	0	0	0	0	—	—	—	—
1				1				
2			1	0				
3				1				
4		1	0	0	Advanced	On-chip ROM disabled, expanded mode	Disabled	16 bit
5				1				8 bits
6			1	0		On-chip ROM enabled, expanded mode	Enabled	8 bits
7				1		Single-chip mode		—
8	1	0	0	0	—	—	—	—
9				1				
10			1	0	Advanced	Boot mode	Enabled	8 bits
11				1				—
12		1	0	0	—	—	—	—
13				1				
14			1	0	Advanced	User program mode	Enabled	8 bits
15				1				—

The CPU's architecture allows for 4 Gbytes of address space, but the H8S/2345 Group
accesses a maximum of 16 Mbytes.

Modes 10, 11, 14, and 15 are boot modes and user program modes in which the flash m
be programmed and erased. For details, see section 17, ROM.

The H8S/2345 Group can only be used in modes 4 to 7, 10, 11, 14, and 15. This means
flash write enable pin and 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 Operating Mode Selection (ZTAT, Mask ROM, and ROMless Versions)

The H8S/2345 Group has seven operating modes (modes 1 to 7). These modes enable s
the CPU operating mode, enabling/disabling of on-chip ROM, and the initial bus width
setting the mode pins (MD_2 to MD_0).

Table 3.2 lists the MCU operating modes.

					expanded mode		
3*			1		Single-chip mode		—
4	1	0	0	Advanced	On-chip ROM disabled, expanded mode	Disabled	16 bits
5			1				8 bits
6*		1	0		On-chip ROM enabled, expanded mode	Enabled	8 bits
7*			1		Single-chip mode		—

Note: * Not used on ROMless version.

The CPU's architecture allows for 4 Gbytes of address space, but the H8S/2345 Group accesses a maximum of 16 Mbytes.

Modes 1, 2, and 4 to 6 are externally expanded modes that allow access to external me peripheral devices.

The external expansion modes allow switching between 8-bit and 16-bit bus modes. A execution starts, an 8-bit or 16-bit address space can be set for each area, depending o controller setting. If 16-bit access is selected for any one area, 16-bit bus mode is set; 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/2345 Group can be used only in modes 1 to 7. This means that the mode pin set to select one of these modes. Do not change the inputs at the mode pins during op

3.1.3 Register Configuration

The H8S/2345 Group has a mode control register (MDCR) that indicates the inputs at pins (MD_2 to MD_0), and a system control register (SYSCR) and a system control regis (SYSCR2)*2 that control the operation of the H8S/2345 Group. Table 3.3 summarizes registers.

ROM, and ROMless versions, this register cannot be written to and will retur
undefined value of read.

3.2 Register Descriptions

3.2.1 Mode Control Register (MDCR)

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

Note: * Determined by pins MD_2 to MD_0.

MDCR is an 8-bit read-only register that indicates the current operating mode of the H8
Group.

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 level
MD_2 to MD_0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD_2 to
MDS2 to MDS0 are read-only bits-they cannot be written to. The mode pin (MD_2 to M
levels are latched into these bits when MDCR is read. These latches are canceled by a
reset, but are retained after a manual reset.

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

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

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

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

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

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

Bit 0	
RAME	Description
0	On-chip RAM is disabled
1	On-chip RAM is enabled

SYSCR2 is initialized to H'00 by a reset and in hardware standby mode.

SYSCR2 can only be accessed in the F-ZTAT version. In other versions, this register c
written to and will return an undefined value if read.

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

Bit 3—Flash Memory Control Register Enable (FLSHE): Controls CPU access to th
memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2). For details, see sec
ROM.

Bit 3

FLSHE	Description
0	Flash control registers are not selected for addresses H'FFFFC8 to H'FFFF (In
1	Flash control registers are selected for addresses H'FFFFC8 to H'FFFFCB

Bits 2 to 0—Reserved: Read-only bits, always read as 0.

bus control signals. However, note that if 16-bit access is designated by the bus contro
mode switches to 16 bits and port E becomes a data bus.

3.3.2 Mode 2[*1] (ZTAT and Mask ROM Versions Only)

The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is
8-bit bus mode is set. immediately after a reset.

Ports B and C function as input ports immediately after a reset. They can each be set
addresses by setting the corresponding bits in the data direction register (DDR) to 1.
functions as a data bus, and part of port F carries bus control signals. However, note th
access is designated by the bus controller, the bus mode switches to 16 bits and port E
data bus.

The amount of on-chip ROM that can be used is limited to 56 kbytes.

3.3.3 Mode 3[*1] (ZTAT and Mask ROM Versions Only)

The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is
external addresses cannot be accessed.

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

The amount of on-chip ROM that can be used is limited to 56 kbytes.

3.3.4 Mode 4[*2]

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

Pins $P1_3$ to $P1_0$, ports A, B, and C function as an address bus, ports D and E function a
and part of port F carries bus control signals. Pins $P1_3$ to $P1_0$ function as inputs immed
reset. Each of these pins can be set to output addresses by setting the corresponding bi
direction register (DDR) to 1.

part of port F carries bus control signals. Pins $P1_3$ to $P1_0$ function as inputs immediately
reset. They can each be set to output addresses by setting the corresponding bits in the d
direction register (DDR) to 1.

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

3.3.6 Mode 6[*1]

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

Pins $P1_3$ to $P1_0$, ports A, B, and C function as input ports immediately after a reset. The
be set to output addresses by setting the corresponding bits in the data direction register
1. Port D functions as a data bus, and part of port F carries bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, if ar
designated as 16-bit access space by the bus controller, 16-bit bus mode is set and port
a data bus.

3.3.7 Mode 7[*1]

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

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

Notes: 1. Not used on ROMless version.
2. The upper address pins (A_{23} to A_{20}) cannot be used as outputs in mode 4 or 5
immediately after a reset. To use the upper address pins (A_{23} to A_{20}) as outpu
necessary to first set the corresponding bits in the port 1 data direction regis
(P1DDR) to 1.

MCU operation is the same as in mode 6.

3.3.10 Mode 11 (F-ZTAT Version Only)

This is a flash memory boot mode. For details, see section 17, ROM.

MCU operation is the same as in mode 7.

3.3.11 Modes 12 and 13

Modes 12 and 13 are not supported in the H8S/2345 Group, and must not be set.

3.3.12 Mode 14 (F-ZTAT Version Only)

This is a flash memory user program mode. For details, see section 17, ROM.

MCU operation is the same as in mode 6.

3.3.13 Mode 15 (F-ZTAT Version Only)

This is a flash memory user program mode. For details, see section 17, ROM.

MCU operation is the same as in mode 7.

Port		Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 14
Port 1	$P1_3$ to $P1_0$	P*1/T	P*1/T	P*1/T	P*1/T/A	P*1/T/A	P*1/T/A
Port A	PA_3 to PA_0	P	P	P	A	A	P*1/A
Port B		A	P*1/A	P	A	A	P*1/A
Port C		A	P*1/A	P	A	A	P*1/A
Port D		D	D	P	D	D	D
Port E		P*1/D	P*1/D	P	P/D	P*1/D	P*1/D
Port F	PF_7	P/C*1	P/C*1	P*1/C	P/C*1	P/C*1	P/C*1
	PF_6 to PF_3	C	C	P	C	C	C
	PF_2 to PF_0	P*1/C	P*1/C		P*1/C	P*1/C	P*1/C

Legend:

P: I/O port

T: Timer I/O

A: Address bus output

D: Data bus I/O

C: Control signals, clock I/O

Notes: 1. After reset

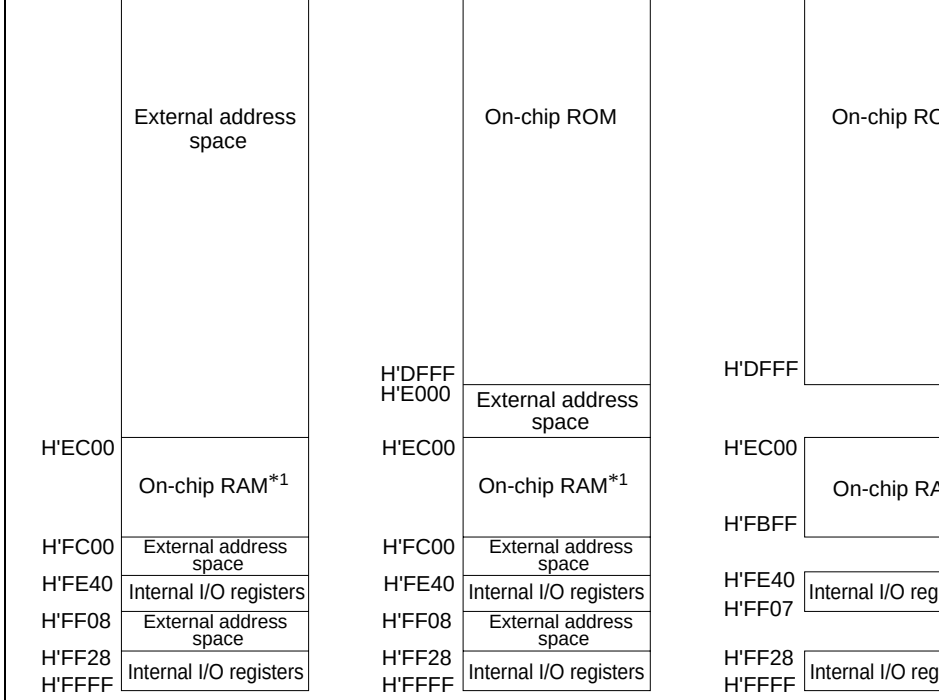
2. Not used on F-ZTAT.

3. Not used on ROMless version.

4. Applies to F-ZTAT version only.

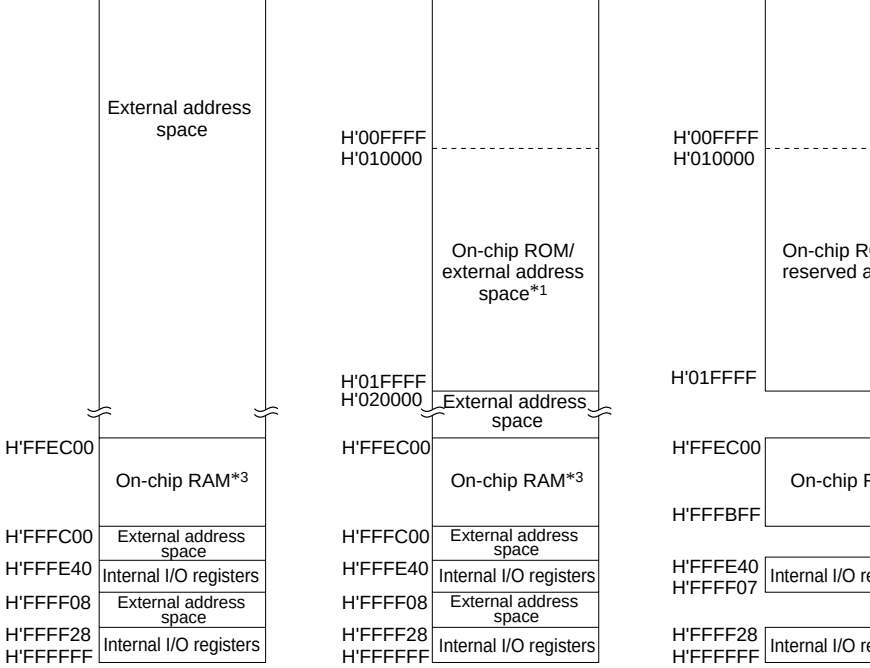
The address space is divided into eight areas for modes 4 to 6, 10, and 14. For details, 6, Bus Controller.

Note: * Not available on F-ZTAT version.



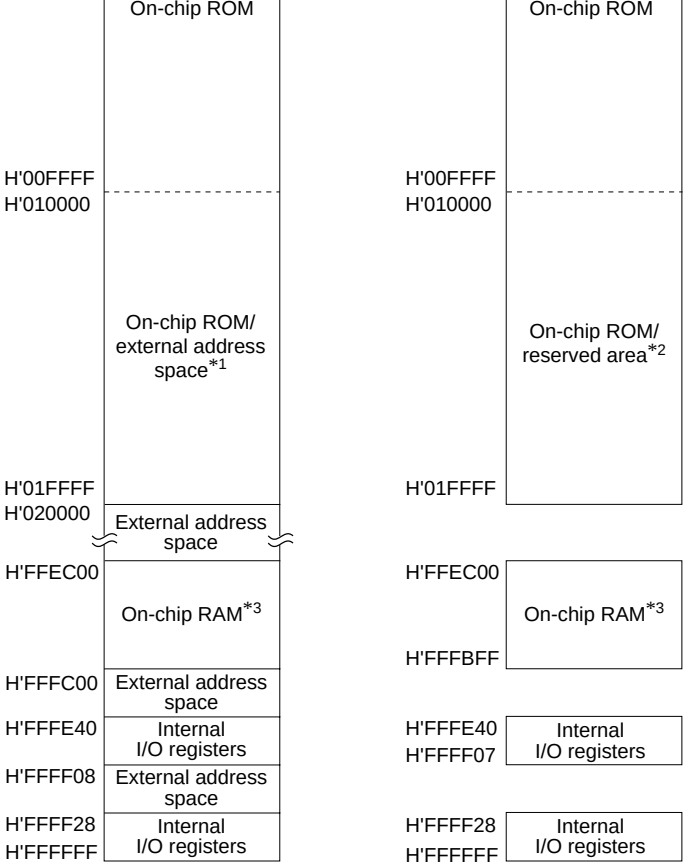
Notes: 1. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
2. Not available on F-ZTAT version.

Figure 3.1 Memory Map in Each Operating Mode in the H8S/2345



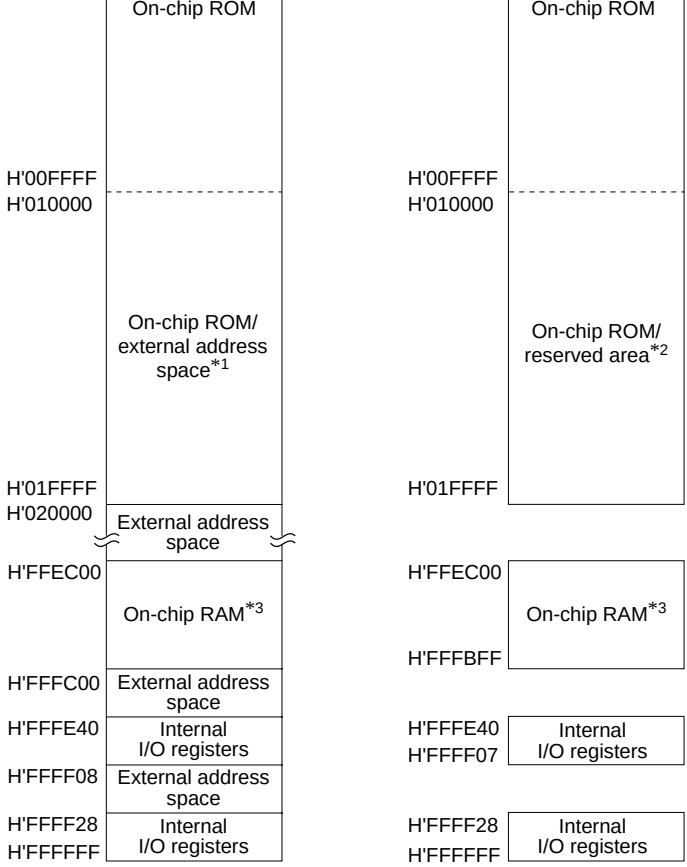
Notes: 1. When the EAE bit in BCRL is set to 1, this area is external address space.
When the EAE bit is cleared to 0, it is on-chip ROM.
2. When the EAE bit in BCRL is set to 1, this area is reserved.
When the EAE bit is cleared to 0, it is on-chip ROM.
3. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.1 Memory Map in Each Operating Mode in the H8S/2345 (co



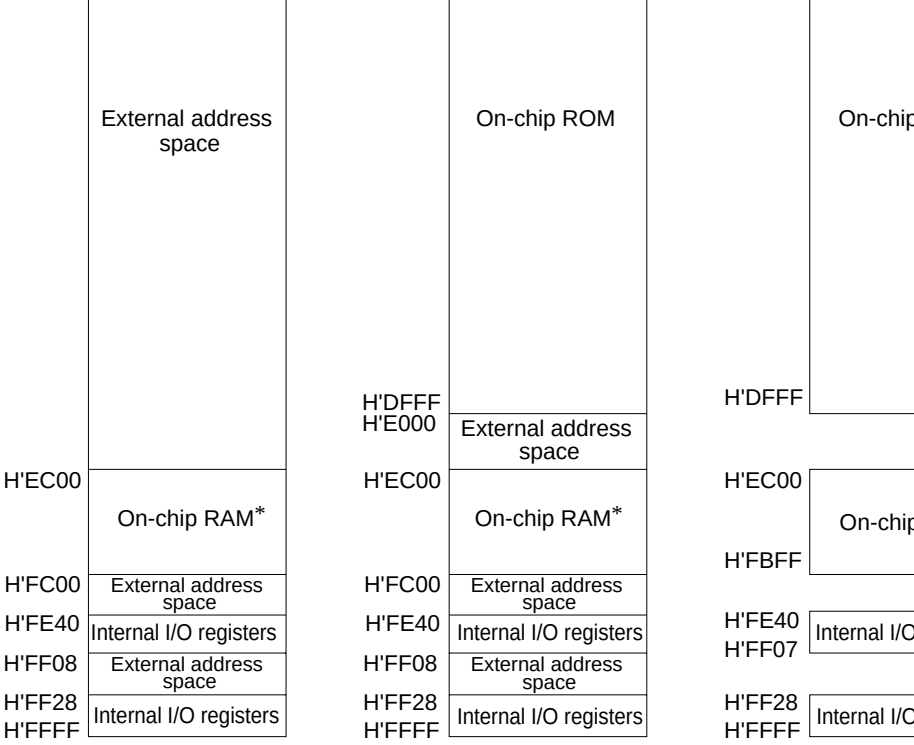
Notes: 1. When the EAE bit in BCRL is set to 1, this area is external address space.
When the EAE bit is cleared to 0, it is on-chip ROM.
2. When the EAE bit in BCRL is set to 1, this area is reserved.
When the EAE bit is cleared to 0, it is on-chip ROM.
3. On-chip RAM is used for flash memory programming. Do not clear the RAME bit to 0 in
4. Modes 10 and 11 are provided in the F-ZTAT version only.

Figure 3.1 Memory Map in Each Operating Mode in the H8S/2345 (con



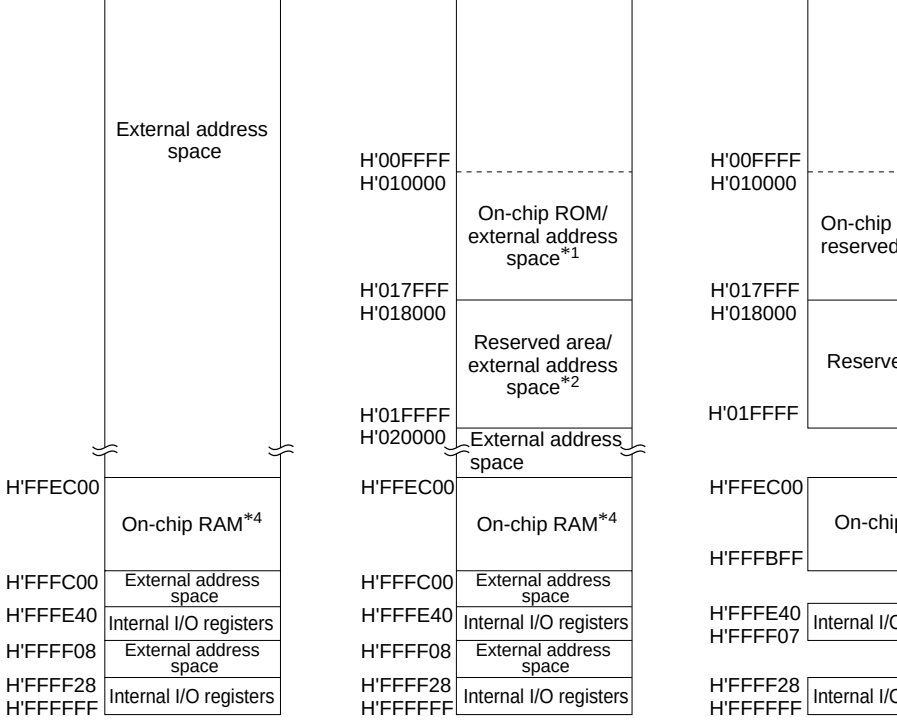
Notes: 1. When the EAE bit in BCRL is set to 1, this area is external address space.
When the EAE bit is cleared to 0, it is on-chip ROM.
2. When the EAE bit in BCRL is set to 1, this area is reserved.
When the EAE bit is cleared to 0, it is on-chip ROM.
3. On-chip RAM is used for flash memory programming. Do not clear the RAME bit to 0
4. Modes 14 and 15 are provided in the F-ZTAT version only.

Figure 3.1 Memory Map in Each Operating Mode in the H8S/2345 (co



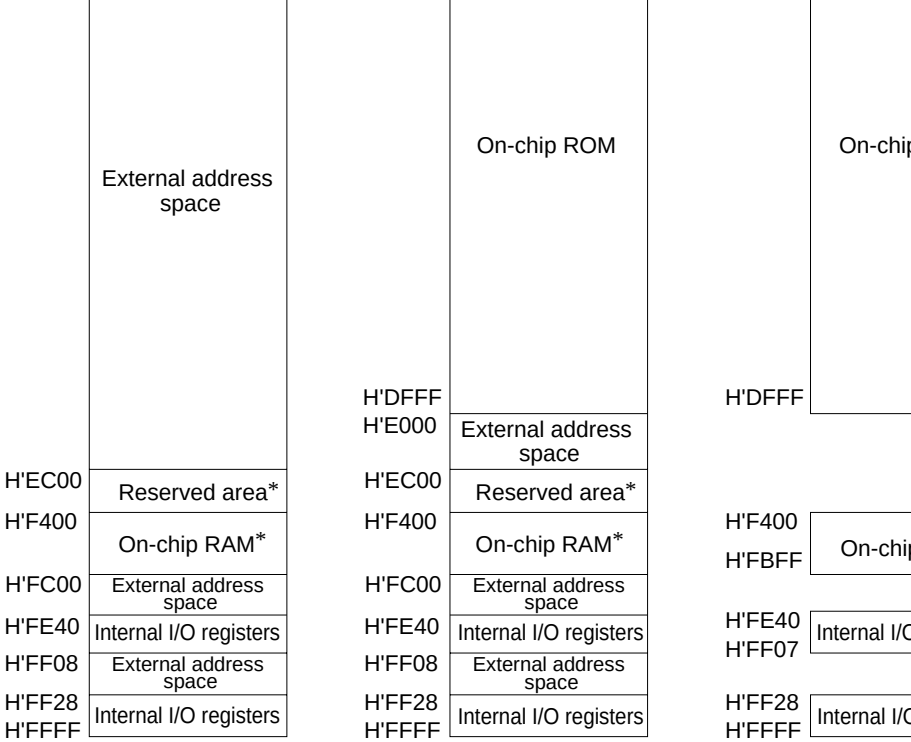
Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.2 Memory Map in Each Operating Mode in the H8S/2344



Notes: 1. When the EAE bit in BCRL is set to 1, this area is external address space.
When the EAE bit is cleared to 0, it is on-chip ROM.
2. When the EAE bit in BCRL is set to 1, this area is external address space.
When the EAE bit is cleared to 0, it is a reserved area.
3. This area is reserved when the EAE bit in BCRL is set to 1, and on-chip ROM when the E bit is cleared to 0.
4. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.2 Memory Map in Each Operating Mode in the H8S/2344 (co

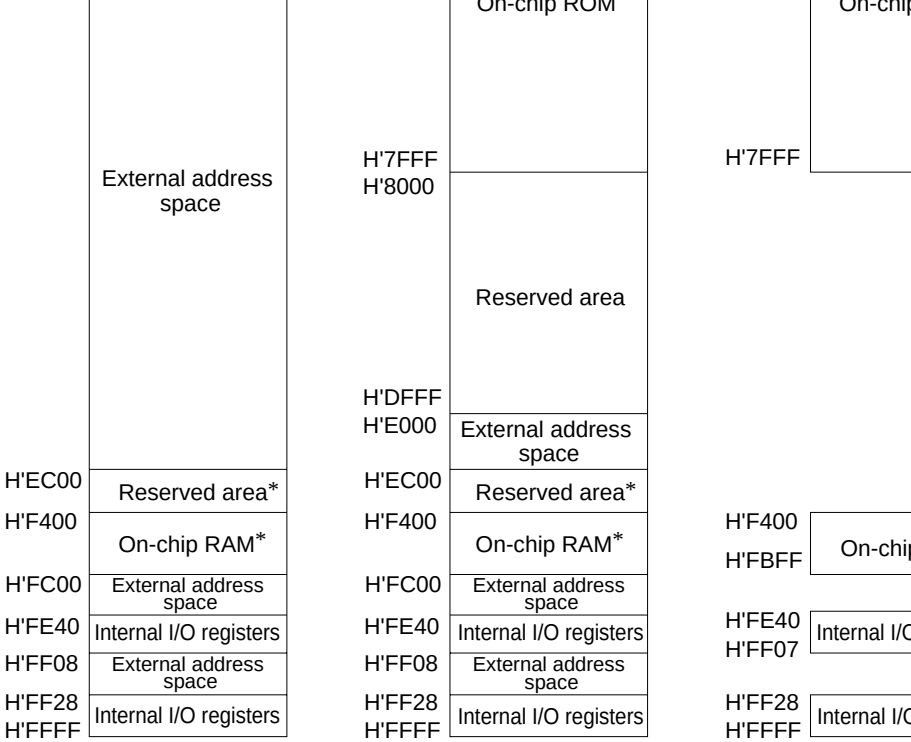


Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.3 Memory Map in Each Operating Mode in the H8S/2343

Notes: 1. When the EAE bit in BCRL is set to 1, this area is external address space. When the EA
cleared to 0, it is on-chip ROM.
2. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.3 Memory Map in Each Operating Mode in the H8S/2343 (co

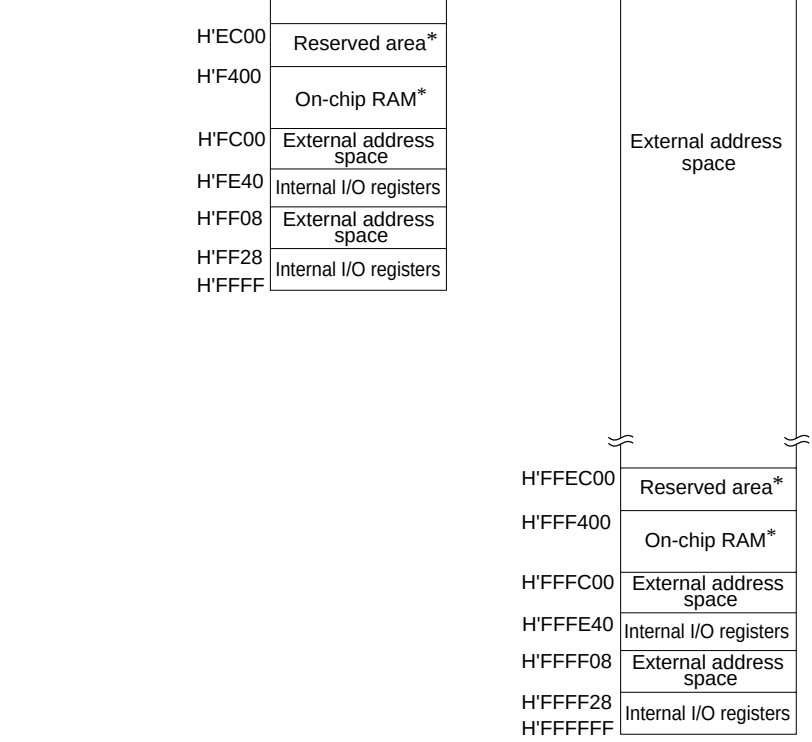


Note: * External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.4 Memory Map in Each Operating Mode in the H8S/2341

Notes: 1. When the EAE bit in BCRL is set to 1, this area is external address space. When the EA cleared to 0, it is on-chip ROM.
2. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.

Figure 3.4 Memory Map in Each Operating Mode in the H8S/2341 (co



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

Figure 3.5 Memory Map in Each Operating Mode in the H8S/2340 (Modes 1, 4, and 5 Only)

Exception handling is prioritized as shown in table 4.1. If two or more exceptions occ
simultaneously, they are accepted and processed in order of priority. Trap instruction
are accepted at all times, in the program execution state.

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

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High ▲	Reset	Starts immediately after a low-to-high transit $\overline{\text{RES}}$ pin, or when the watchdog timer overflo CPU enters the power-on reset state when t is high, or the manual reset state when the N low.
	Trace*1	Starts when execution of the current instruct exception handling ends, if the trace (T) bit is
	Interrupt	Starts when execution of the current instruct exception handling ends, if an interrupt requ been issued*2
Low	Trap instruction (TRAPA)*3	Started by execution of a trap instruction (TR

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

3. A vector address corresponding to the exception source is generated, and program e
starts from that address.

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

4.1.3 Exception Vector Table

The exception sources are classified as shown in figure 4.1. Different vector addresses
assigned to different exception sources.

Table 4.2 lists the exception sources and their vector addresses.

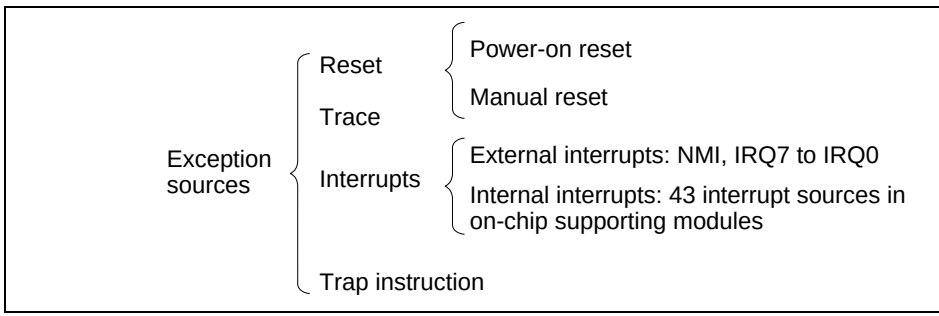


Figure 4.1 Exception Sources

In modes 6 and 7 in the H8S/2345, the on-chip ROM available for use after a power-on
64-kbyte area comprising addresses H'000000 to H'00FFFF. Care is required when sett
addresses. In this case, clearing the EAE bit in BCRL enables the 128-kbyte area comp
addresses H'000000 to H'01FFFF to be used.

		4	H'0008 to H'0009	H'0010 t
Trace		5	H'000A to H'000B	H'0014 t
Reserved for system use		6	H'000C to H'000D	H'0018 t
External interrupt	NMI	7	H'000E to H'000F	H'001C t
Trap instruction (4 sources)		8	H'0010 to H'0011	H'0020 t
		9	H'0012 to H'0013	H'0024 t
		10	H'0014 to H'0015	H'0028 t
		11	H'0016 to H'0017	H'002C t
Reserved for system use		12	H'0018 to H'0019	H'0030 t
		13	H'001A to H'001B	H'0034 t
		14	H'001C to H'001D	H'0038 t
		15	H'001E to H'001F	H'003C t
External interrupt	IRQ0	16	H'0020 to H'0021	H'0040 t
	IRQ1	17	H'0022 to H'0023	H'0044 t
	IRQ2	18	H'0024 to H'0025	H'0048 t
	IRQ3	19	H'0026 to H'0027	H'004C t
	IRQ4	20	H'0028 to H'0029	H'0050 t
	IRQ5	21	H'002A to H'002B	H'0054 t
	IRQ6	22	H'002C to H'002D	H'0058 t
	IRQ7	23	H'002E to H'002F	H'005C t
Internal interrupt*2		24 | 87	H'0030 to H'0031 | H'00AE to H'00AF	H'0060 t H'015C t

Notes: 1. Lower 16 bits of the address.

2. For details of internal interrupt vectors, see section 5.3.3, Interrupt Exceptio Vector Table.

3. ZTAT, mask ROM, and ROMless versions only.

Immediately after a reset, interrupt control mode 0 is set.

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

The level of the NMI pin at reset determines whether the type of reset is a power-on res
manual reset.

The H8S/2345 Group can also be reset by overflow of the watchdog timer. For details s
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 ar
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
all the registers in the on-chip supporting modules, while a manual reset initializes all t
in the on-chip supporting modules except for the bus controller and I/O ports, which re
previous states.
With a manual reset, since the on-chip supporting modules are initialized, ports used as
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	**NMI**	**$\overline{\text{RES}}$**	**CPU**	**On-Chip Supporting Mo**
Power-on reset	High	Low	Initialized	Initialized
Manual reset	Low	Low	Initialized	Initialized, except for bus and I/O ports

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

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

Figures 4.2 and 4.3 show examples of the reset sequence.

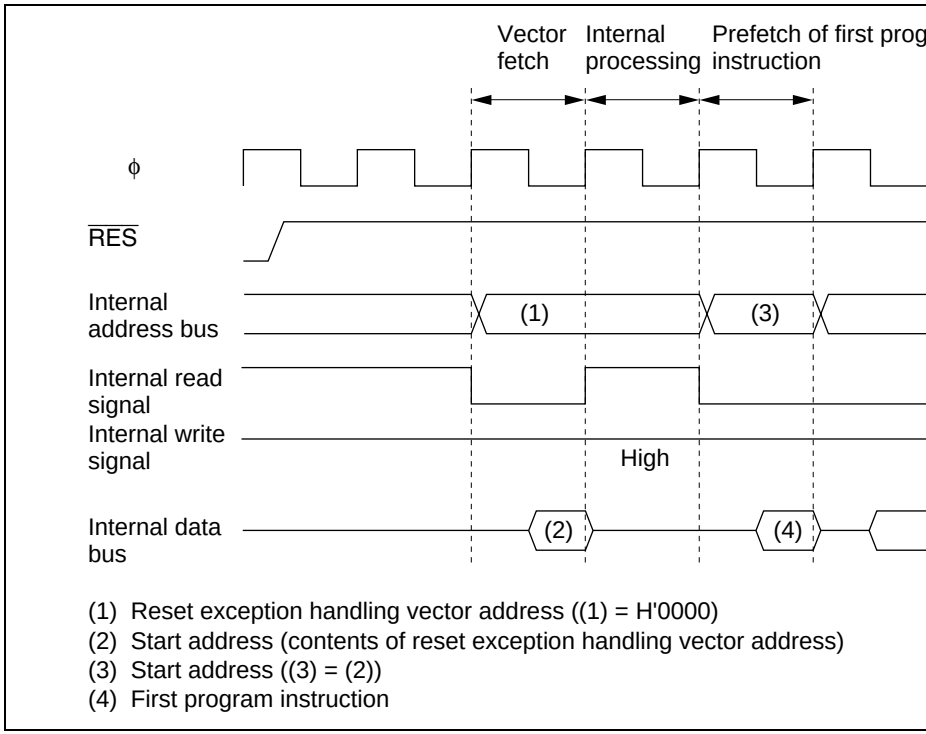
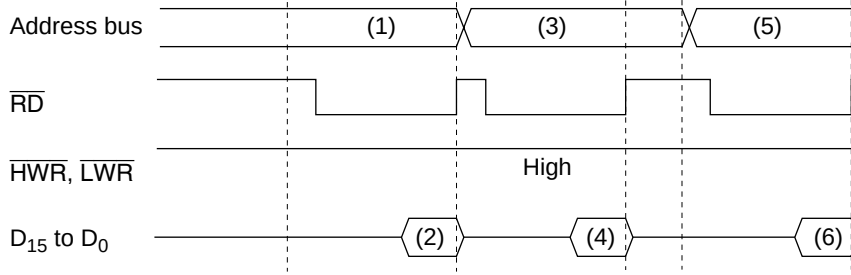


Figure 4.2 Reset Sequence (Modes 2 and 3)



(1) (3) Reset exception handling vector address ((1) = H'000000, (3) = H'00000
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Note: * 3 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, th
CCR will not be saved correctly, leading to a program crash. To prevent this, all interru
including NMI, are disabled immediately after a reset. Since the first instruction of a pr
always executed immediately after the reset state ends, make sure that this instruction i
the stack pointer (example: MOV.L #xx:32, SP).

4.2.5 State of On-Chip Supporting Modules after Reset Release

After reset release, MSTPCR is initialized to H'3FFF and all modules except the DTC e
module stop mode. Consequently, on-chip supporting module registers cannot be read o
to. Register reading and writing is enabled when module stop mode is exited.

Trace mode is canceled by clearing the T bit in EXR to 0. It is not affected by interru

Table 4.4 shows the state of CCR and EXR after execution of trace exception handlin

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 t
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

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

Legend:
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution.

NMI is the highest-priority interrupt. Interrupts are controlled by the interrupt controlle
interrupt controller has two interrupt control modes and can assign interrupts other than
eight priority/mask levels to enable multiplexed interrupt control.

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

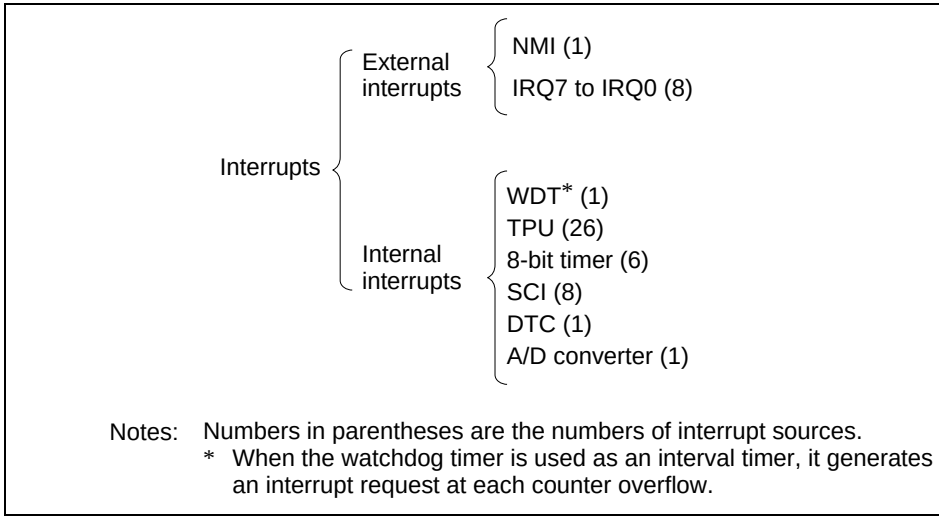


Figure 4.4 Interrupt Sources and Number of Interrupts

Table 4.5 Status of CCR and EXR after Trap Instruction Exception Handling

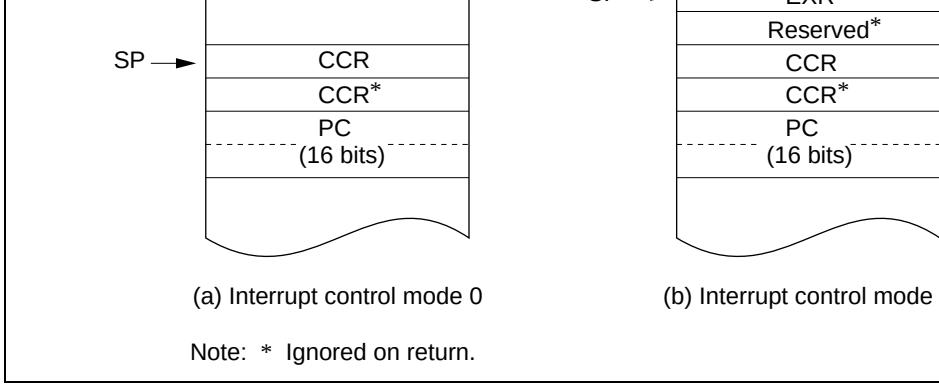
Interrupt Control Mode	CCR		EXR	
	I	UI	I2 to I0	
0	1	—	—	
2	1	—	—	

Legend:

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

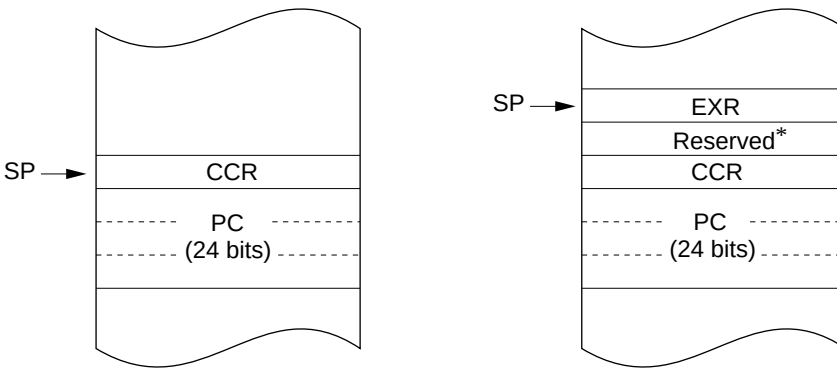


(a) Interrupt control mode 0

(b) Interrupt control mode

Note: * Ignored on return.

Figure 4.5 (1) Stack Status after Exception Handling (Normal Modes) (ZTAT, Mask ROM, and ROMless Versions Only)



(a) Interrupt control mode 0

(b) Interrupt control mode

Note: * Ignored on return.

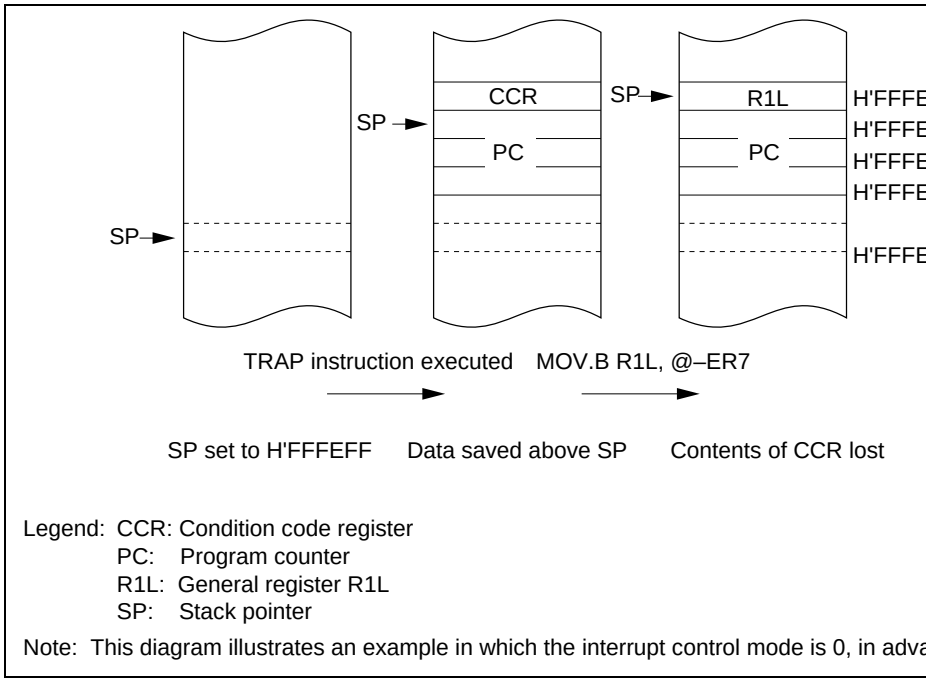
Figure 4.5 (2) Stack Status after Exception Handling (Advanced Modes

```
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 4.6 shows an example o
happens when the SP value is odd.



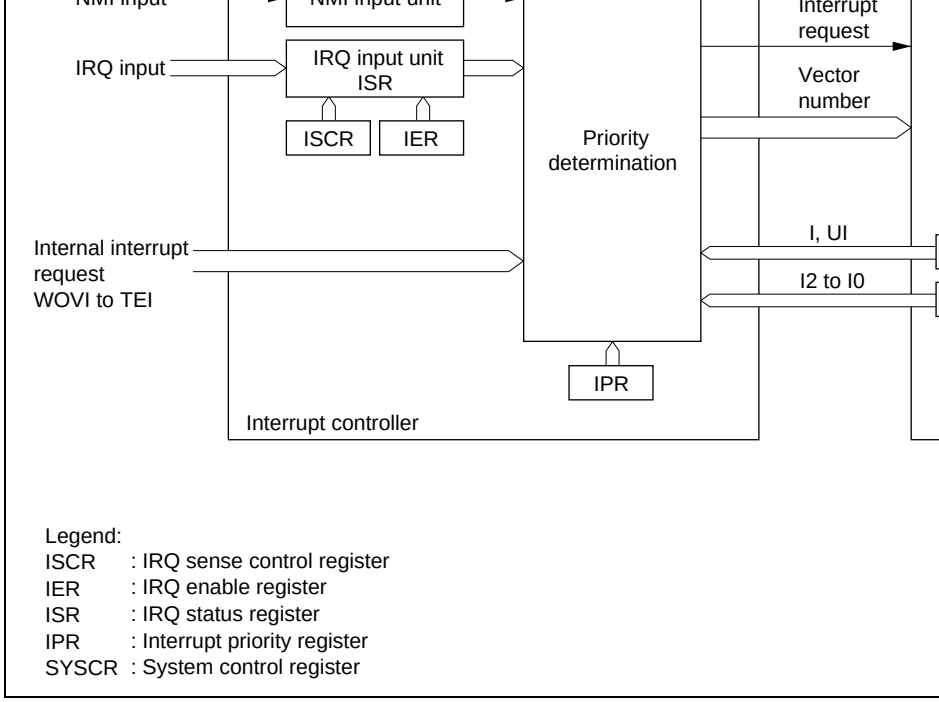
Legend: CCR: Condition code register
PC: Program counter
R1L: General register R1L
SP: Stack pointer

Note: This diagram illustrates an example in which the interrupt control mode is 0, in adva

Figure 4.6 Operation when SP Value is Odd

controller has the following features:

- Two interrupt control modes
  - Any of two interrupt control modes can be set by means of the INTM1 and INT the system control register (SYSCR).
- Priorities settable with IPR
  - An interrupt priority register (IPR) is provided for setting interrupt priorities. E 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 unn 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 o edge can be selected for NMI.
  - Falling edge, rising edge, or both edge detection, or level sensing, can be selec to IRQ0.
- DTC control
  - DTC activation is performed by means of interrupts.



Legend:
ISCR : IRQ sense control register
IER : IRQ enable register
ISR : IRQ status register
IPR : Interrupt priority register
SYSCR : System control register

Figure 5.1 Block Diagram of Interrupt Controller

External interrupt requests 7 to 0	$\overline{IRQ7}$ to $\overline{IRQ0}$	Input	Maskable external interrupts; risin both edges, or level sensing, can

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	Ad
System control register	SYSCR	R/W	H'01	H'F
IRQ sense control register H	ISCRH	R/W	H'00	H'F
IRQ sense control register L	ISCRL	R/W	H'00	H'F
IRQ enable register	IER	R/W	H'00	H'F
IRQ status register	ISR	R/(W)*2	H'00	H'F
Interrupt priority register A	IPRA	R/W	H'77	H'F
Interrupt priority register B	IPRB	R/W	H'77	H'F
Interrupt priority register C	IPRC	R/W	H'77	H'F
Interrupt priority register D	IPRD	R/W	H'77	H'F
Interrupt priority register E	IPRE	R/W	H'77	H'F
Interrupt priority register F	IPRF	R/W	H'77	H'F
Interrupt priority register G	IPRG	R/W	H'77	H'F
Interrupt priority register H	IPRH	R/W	H'77	H'F
Interrupt priority register I	IPRI	R/W	H'77	H'F
Interrupt priority register J	IPRJ	R/W	H'77	H'F
Interrupt priority register K	IPRK	R/W	H'77	H'F

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

R/W . 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 detected edge for NMI.

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

SYSCR is initialized to H'01 by a reset and in hardware standby mode. It is not initializ software standby mode.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select 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	(
	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	(
1	Interrupt request generated at rising edge of NMI input	

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.

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	—*
IPRE	—*	A/D converter
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRI	8-bit timer channel 0	8-bit timer channel 1
IPRJ	—*	SCI channel 0
IPRK	SCI channel 1	—*

Note: * Reserved bits. Only 1 should be written to these bits.

the priority level of the interrupt is higher than the set mask level, an interrupt request i
the CPU.

5.2.3 IRQ Enable Register (IER)

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

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

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

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

Bit n	
IRQnE	**Description**
0	IRQn interrupts disabled
1	IRQn interrupts enabled

ISCRL

Bit	:	7	6	5	4	3	2	1
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB
Initial value:		0	0	0	0	0	0	0
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 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.

Bits 15 to 0: IRQ7 Sense Control A and B (IRQ7SCA, IRQ7SCB) to IRQ0 Sense Co 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
	1	Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ to
1	0	Interrupt request generated at rising edge of $\overline{\text{IRQ7}}$ to
	1	Interrupt request generated at both falling and rising e $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input

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

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

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

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

be used to restore the H8S/2345 Group from software standby mode.

NMI Interrupt: NMI is the highest-priority interrupt, and is always accepted by the
regardless of the status of the CPU interrupt mask bits. The NMIEG bit in SYSCR ca
select whether an interrupt is requested at a rising edge or a falling edge on the NMI p

The vector number for NMI interrupt exception handling is 7.

IRQ7 to IRQ0 Interrupts: Interrupts IRQ7 to IRQ0 are requested by an input signal
$\overline{\text{IRQ7}}$ 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 leve
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 b
0 by software.

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

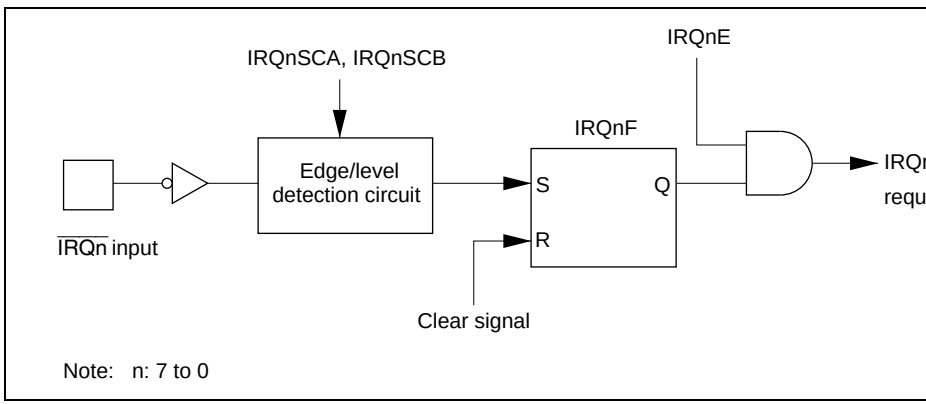


Figure 5.2 Block Diagram of Interrupts IRQ7 to IRQ0

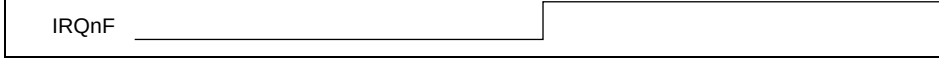


Figure 5.3 Timing of Setting IRQnF

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
input or output. However, when a pin is used as an external interrupt input pin, do not
corresponding DDR to 0 and use the pin as an I/O pin for another function.

5.3.2 Internal Interrupts

There are 43 sources for internal interrupts from on-chip supporting modules.

- For each on-chip supporting module there are flags that indicate the interrupt reque
and enable bits that select enabling or disabling of these interrupts. If both of these
for a particular interrupt source, an interrupt request is issued to the interrupt contro
- The interrupt priority level can be set by means of IPR.
- The DTC can be activated by a TPU, 8-bit timer, SCI, or other interrupt request. W
DTC is activated by an interrupt, the interrupt control mode and interrupt mask bits
affected.

5.3.3 Interrupt Exception Handling Vector Table

Table 5.4 shows interrupt exception handling sources, vector addresses, and interrupt p
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 m
modules are set to the same priority, and priorities within a module, are fixed as shown
table 5.4.

IRQ3		19	H'0026	H'004C	
IRQ4		20	H'0028	H'0050	IPRB2 to
IRQ5		21	H'002A	H'0054	
IRQ6		22	H'002C	H'0058	IPRC6 to
IRQ7		23	H'002E	H'005C	
SWDTEND (software activation interrupt end)	DTC	24	H'0030	H'0060	IPRC2 to
WOVI (interval timer)	Watchdog timer	25	H'0032	H'0064	IPRD6 to
Reserved	—	26	H'0034	H'0068	
		27	H'0036	H'006C	
ADI (A/D conversion end)	A/D	28	H'0038	H'0070	IPRE2 to
Reserved	—	29	H'003A	H'0074	
		30	H'003C	H'0078	
		31	H'003E	H'007C	
TGI0A (TGR0A input capture/compare match)	TPU channel 0	32	H'0040	H'0080	IPRF6 to
TGI0B (TGR0B input capture/compare match)		33	H'0042	H'0084	
TGI0C (TGR0C input capture/compare match)		34	H'0044	H'0088	
TGI0D (TGR0D input capture/compare match)		35	H'0046	H'008C	
TCI0V (overflow 0)		36	H'0048	H'0090	
Reserved	—	37	H'004A	H'0094	
		38	H'004C	H'0098	
		39	H'004E	H'009C	

TGI2A (TGR2A input capture/compare match)	TPU channel 2	44	H'0058	H'00B0	IPRG6 to 4
TGI2B (TGR2B input capture/compare match)		45	H'005A	H'00B4	
TCI2V (overflow 2)		46	H'005C	H'00B8	
TCI2U (underflow 2)		47	H'005E	H'00BC	
TGI3A (TGR3A input capture/compare match)	TPU channel 3	48	H'0060	H'00C0	IPRG2 to 0
TGI3B (TGR3B input capture/compare match)		49	H'0062	H'00C4	
TGI3C (TGR3C input capture/compare match)		50	H'0064	H'00C8	
TGI3D (TGR3D input capture/compare match)		51	H'0066	H'00CC	
TCI3V (overflow 1)		52	H'0068	H'00D0	
Reserved	—	53 54 55	H'006A H'006C H'006E	H'00D4 H'00D8 H'00DC	
TGI4A (TGR4A input capture/compare match)	TPU channel 4	56	H'0070	H'00E0	IPRH6 to 4
TGI4B (TGR4B input capture/compare match)		57	H'0072	H'00E4	
TCI4V (overflow 4)		58	H'0074	H'00E8	
TCI4U (underflow 4)		59	H'0076	H'00EC	
TGI5A (TGR5A input capture/compare match)	TPU channel 5	60	H'0078	H'00F0	IPRH2 to 0
TGI5B (TGR5B input capture/compare match)		61	H'007A	H'00F4	
TCI5V (overflow 5)		62	H'007C	H'00F8	
TCI5U (underflow 5)		63	H'007E	H'00FC	

CMIB1 (compare match B1)	channel 1	69	H'008A	H'0114	
OVI1 (overflow 1)		70	H'008C	H'0118	
Reserved	—	71	H'008E	H'011C	
		72	H'0090	H'0120	
		73	H'0092	H'0124	
		74	H'0094	H'0128	
		75	H'0096	H'012C	
		76	H'0098	H'0130	
		77	H'009A	H'0134	
		78	H'009C	H'0138	
		79	H'009E	H'013C	
ERI0 (receive error 0)	SCI channel 0	80	H'00A0	H'0140	IPRJ2 to
RXI0 (reception completed 0)		81	H'00A2	H'0144	
TXI0 (transmit data empty 0)		82	H'00A4	H'0148	
TEI0 (transmission end 0)		83	H'00A6	H'014C	
ERI1 (receive error 1)	SCI channel 1	84	H'00A8	H'0150	IPRK6 to
RXI1 (reception completed 1)		85	H'00AA	H'0154	
TXI1 (transmit data empty 1)		86	H'00AC	H'0158	
TEI1 (transmission end 1)		87	H'00AE	H'015C	

Notes: 1. Lower 16 bits of the start address.
2. ZTAT, mask ROM, and ROMless versions only.

each interrupt. Clearing an enable bit to 0 disables the corresponding interrupt request. 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 mo the INTM1 and INTM0 bits in SYSCR, the priorities set in IPR, and the masking state 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 con performed by the I
—		1	—	—	Setting prohibited
2	1	0	IPR	I2 to I0	8-level interrupt ma is performed by bit 8 priority levels ca with IPR.
—		1	—	—	Setting prohibited

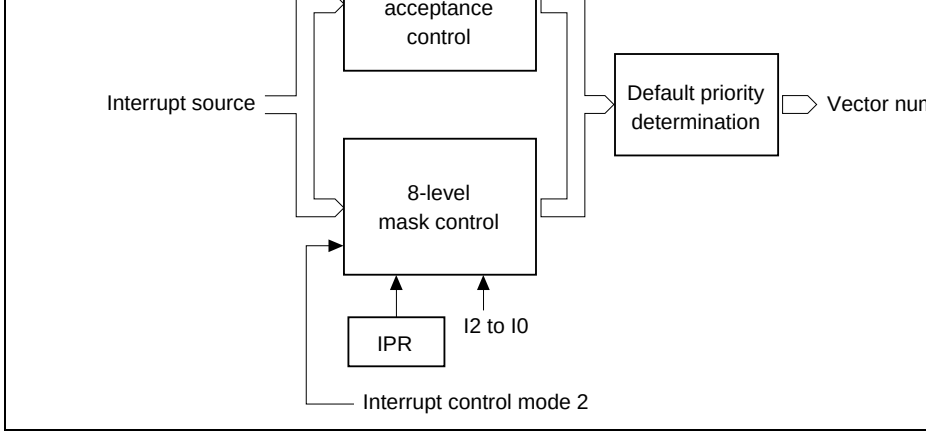


Figure 5.4 Block Diagram of Interrupt Control Operation

(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

Legend:
*: Don't care

Interrupt Control Mode	Selected Interrupts
0	All interrupts
2	Highest-priority-level (IPR) interrupt whose priority level is 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
generated.

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

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

Table 5.8 shows operations and control signal functions in each interrupt control mode.

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

Note: 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 ca
means of the I bit in the CPU's CCR. Interrupts are enabled when the I bit is cleared t
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
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 accep
bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are h

[3] Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt a
the priority system is accepted, and other interrupt requests are held pending.

[4] When an interrupt request is accepted, interrupt exception handling starts after exe
current instruction has been completed.

[5] The PC and CCR are saved to the stack area by interrupt exception handling. The
the stack shows the address of the first instruction to be executed after returning fr
interrupt handling routine.

[6] Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

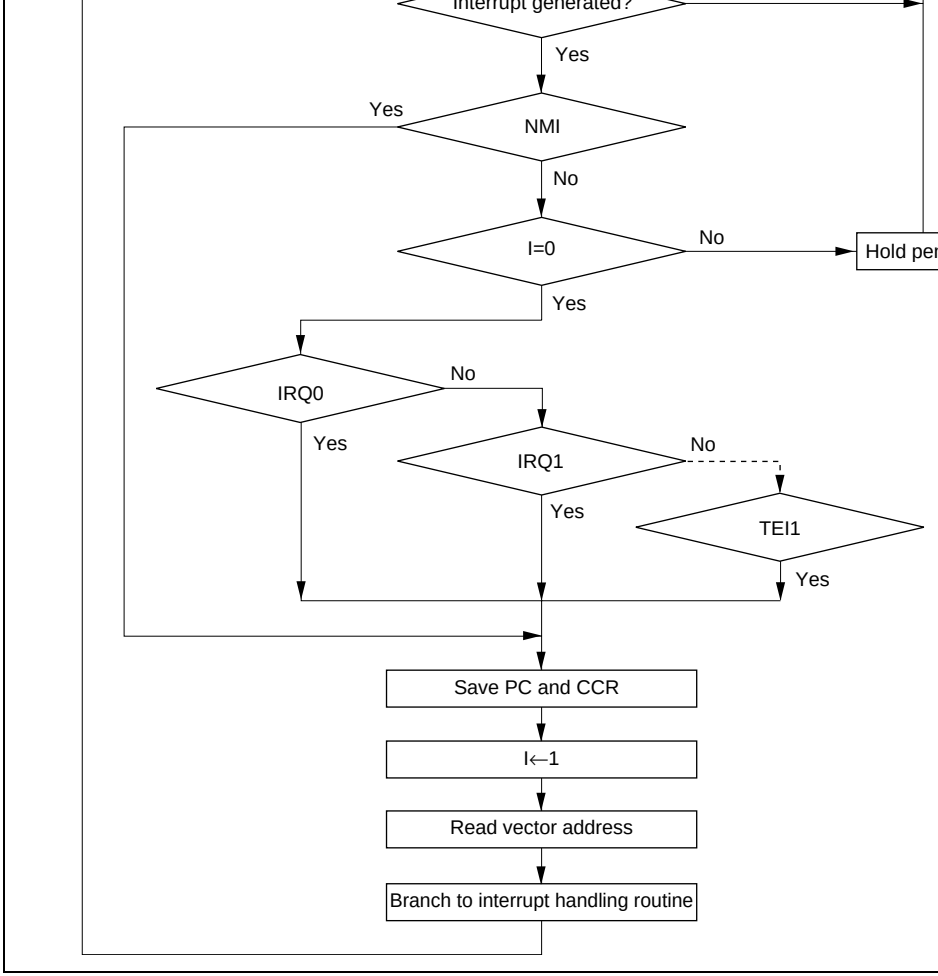


Figure 5.5 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

[2] When interrupt requests are sent to the interrupt controller, the interrupt with the h
priority according to the interrupt priority levels set in IPR is selected, and lower-p
interrupt requests are held pending. If a number of interrupt requests with the same
generated at the same time, the interrupt request with the highest priority according
priority system shown in table 5.4 is selected.

[3] Next, the priority of the selected interrupt request is compared with the interrupt m
in EXR. An interrupt request with a priority no higher than the mask level set at th
held pending, and only an interrupt request with a priority higher than the interrupt
is accepted.

[4] When an interrupt request is accepted, interrupt exception handling starts after exe
current instruction has been completed.

[5] The PC, CCR, and EXR are saved to the stack area by interrupt exception handling
saved on the stack shows the address of the first instruction to be executed after re
the interrupt handling routine.

[6] The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the pri
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 inter
handling routine starts at the address indicated by the contents of that vector addre

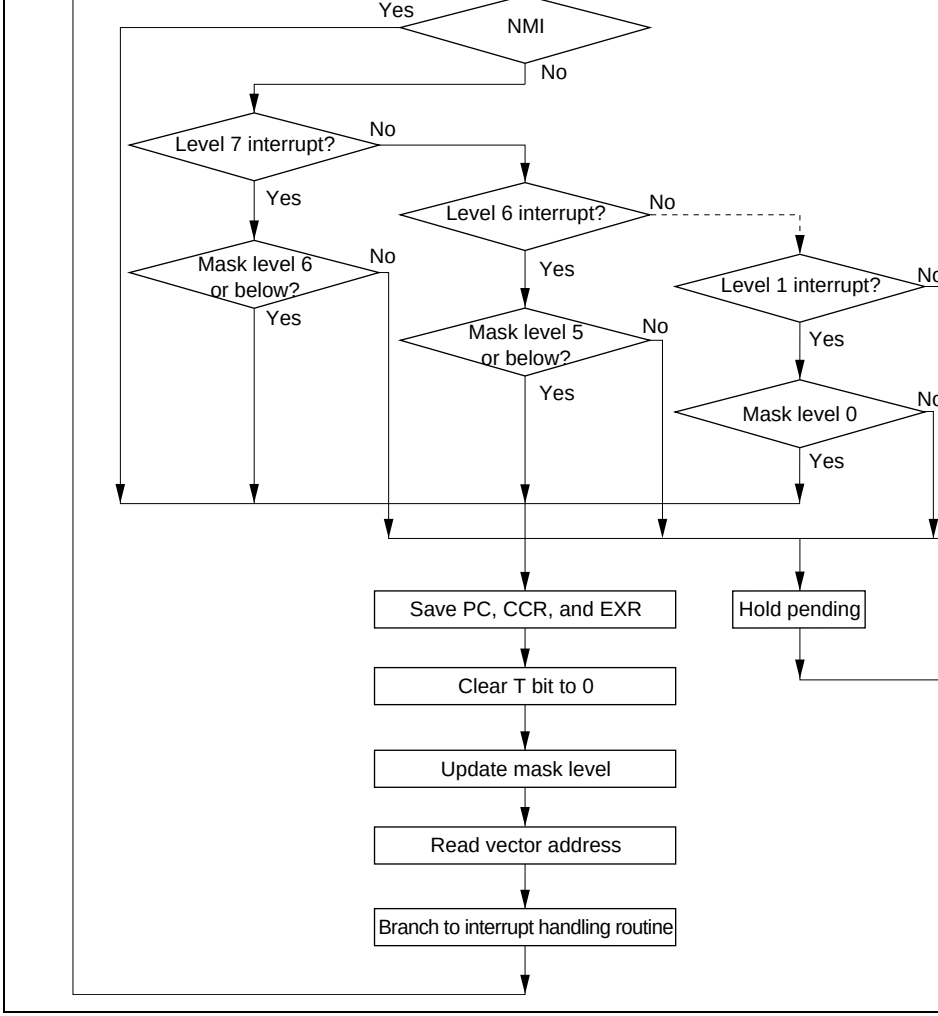


Figure 5.6 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

Figure 5.7 Interrupt Exception Handling

Table 5.9 Interrupt Response Times

No.	Execution Status	Normal Mode*5		Advanc
		INTM1 = 0	INTM1 = 1	INTM1 = 0
1	Interrupt priority determination*1	3	3	3
2	Number of wait states until executing instruction ends*2	1 to $19+2\cdot S_I$	1 to $19+2\cdot S_I$	1 to $19+2\cdot S_I$
3	PC, CCR, EXR stack save	$2\cdot S_K$	$3\cdot S_K$	$2\cdot S_K$
4	Vector fetch	S_I	S_I	$2\cdot S_I$
5	Instruction fetch*3	$2\cdot S_I$	$2\cdot S_I$	$2\cdot S_I$
6	Internal processing*4	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32

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
5. ZTAT, mask ROM, and ROMless versions only.

Branch address read	S_J
Stack manipulation	S_K

Legend:

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
after execution of the instruction.

In other words, when an interrupt enable bit is cleared to 0 by an instruction such as BC
MOV, if an interrupt is generated during execution of the instruction, the interrupt conc
still be enabled on completion of the instruction, and so interrupt exception handling fo
interrupt will be executed on completion of the instruction. However, if there is an inte
request of higher priority than that interrupt, interrupt exception handling will be execu
higher-priority interrupt, and the lower-priority interrupt will be ignored.

The same also applies when an interrupt source flag is cleared.

Figure 5.8 shows and example in which the CMIEA bit in 8-bit timer TCR is cleared to

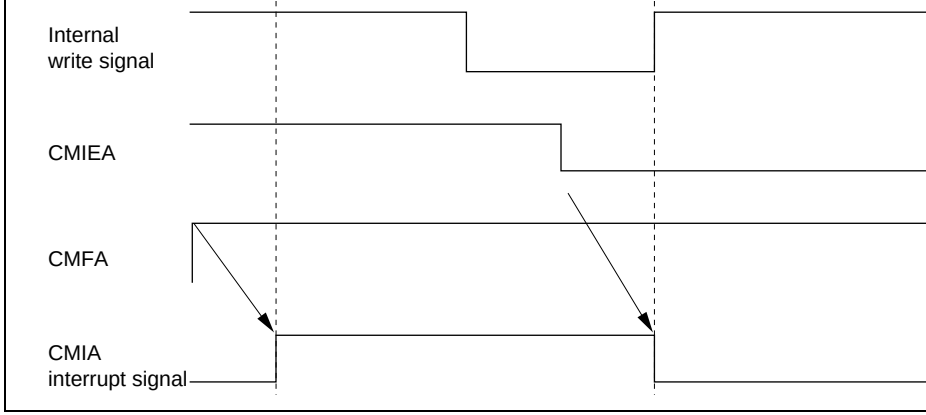


Figure 5.8 Contention between Interrupt Generation and Disabling

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

5.5.2 Instructions that Disable Interrupts

Instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of th
instructions is executed, all interrupts including NMI are disabled and the next instruc
always executed. When the I bit is set by one of these instructions, the new value becc
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
updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

case is the address of the next instruction.

Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, following coding should be used.

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

5.6 DTC Activation by Interrupt

5.6.1 Overview

The DTC can be activated by an interrupt. In this case, the following options are availa

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

For details of interrupt requests that can be used with to activate the DTC, see section 7 Transfer Controller.

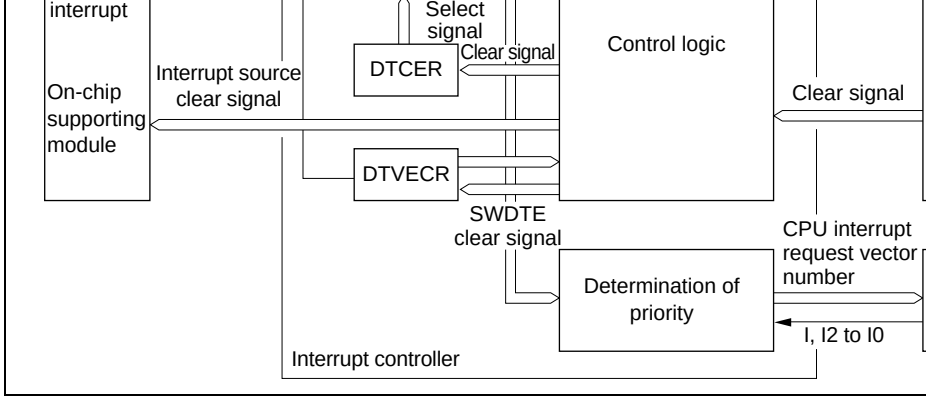


Figure 5.9 Interrupt Control for DTC and DMAC

5.6.3 Operation

The interrupt controller has three main functions in DTC control.

(1) Selection of Interrupt Source

Interrupt sources can be specified as DTC activation requests or CPU interrupt reques of the DTCE bit of DTCEA to DTCEE in the DTC.

After a DTC data transfer, the DTCE bit can be cleared to 0 and an interrupt request s CPU in accordance with the specification of the DISEL bit of MRB in the DTC.

When the DTC has performed the specified number of data transfers and the transfer c is zero, the DTCE bit is cleared to 0 and an interrupt request is sent to the CPU after t transfer.

data transfer is performed first, followed by CPU interrupt exception handling.

If the same interrupt is selected as a DTC activation source or CPU interrupt source, op
performed for them independently according to their respective operating statuses and l
mastership priorities.

Table 5.11 summarizes interrupt source selection and interrupt source clearance control
to the settings of the DTCE bit of DTCEA to DTCEE in the DTC and the DISEL bit of
the DTC.

Table 5.11 Interrupt Source Selection and Clearing Control

Settings			
DTC		**Interrupt Source Selection/Clearing C**	
DTCE	**DISEL**	**DTC**	**CPU**
0	*	X	Δ
1	0	Δ	X
	1	○	Δ

Legend:

Δ : 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

SCI and A/D converter interrupt sources are cleared when the DTC reads or writes to th
prescribed register, and are not dependent upon the DISEL bit.

The bus controller also has a bus arbitration function, and controls the operation of the masters: the CPU 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
  - In advanced mode, manages the external space as 8 areas of 2-Mbytes
  - In normal mode*, manages the external space as a single area
  - Bus specifications can be set independently for each area
- Basic bus interface
  - Chip select ($\overline{CS0}$ to $\overline{CS3}$) can be output for areas 0 to 3
  - 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 differen
  - An idle cycle can be inserted in case of an external write cycle immediately aft external read cycle
- Bus arbitration function
  - Includes a bus arbiter that arbitrates bus mastership among the CPU and DTC
- Other features
  - External bus release function

Note: * ZTAT, mask ROM, and ROMless versions only.

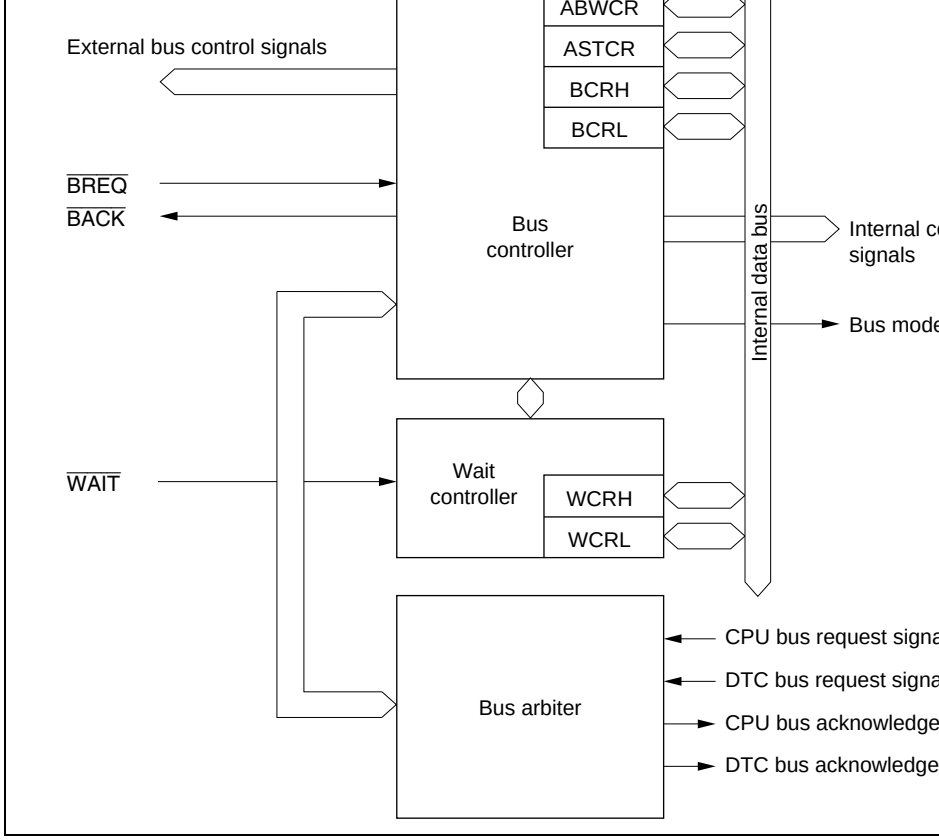


Figure 6.1 Block Diagram of Bus Controller

Read	$\overline{RD}$	Output	Strobe signal indicating that external space is
High write	$\overline{HWR}$	Output	Strobe signal indicating that external space is written, and upper half (D_{15} to D_8) of data bus i
Low write	$\overline{LWR}$	Output	Strobe signal indicating that external space is written, and lower half (D_7 to D_0) of data bus is
Chip select 0 to 3	$\overline{CS0}$ to $\overline{CS3}$	Output	Strobe signal indicating that areas 0 to 3 are s
Wait	$\overline{WAIT}$	Input	Wait request signal when accessing external 3 access space.
Bus request	$\overline{BREQ}$	Input	Request signal that releases bus to external d
Bus request acknowledge	$\overline{BACK}$	Output	Acknowledge signal indicating that bus has be released.

6.1.4 Register Configuration

Table 6.2 summarizes the registers of the bus controller.

Table 6.2 Bus Controller Registers

			Initial Value		
Name	**Abbreviation**	**R/W**	**Power-On Reset**	**Manual Reset**	**A**
Bus width control register	ABWCR	R/W	H'FF/H'00*2	Retained	R
Access state control register	ASTCR	R/W	H'FF	Retained	R
Wait control register H	WCRH	R/W	H'FF	Retained	R
Wait control register L	WCRL	R/W	H'FF	Retained	R
Bus control register H	BCRH	R/W	H'D0	Retained	R
Bus control register L	BCRL	R/W	H'3C	Retained	R

Notes: 1. Lower 16 bits of the address.
2. Determined by the MCU operating mode.

Initial value	:	1	1	1	1	1	1	1
RW	:	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
RW	:	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 16-bit access.

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

In normal mode*, the settings of bits ABW7 to ABW1 have no effect on operation.

After a power-on reset and in hardware standby mode, ABWCR is initialized to H'FF i 2, 3*, and 5, 6, 7, and to H'00 in mode 4. It is not initialized by a manual reset or in so standby mode.

Bits 7 to 0—Area 7 to 0 Bus Width Control (ABW7 to ABW0): These bits select wh corresponding area is to be designated for 8-bit access or 16-bit access. In normal mode part of area 0 is enabled, and the ABW0 bit selects whether external space is to be desi 8-bit access or 16-bit access.

Note: * ZTAT, mask ROM, and ROMless versions only.

Bit n

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

space or a 3-state access space.

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

In normal mode*, the settings of bits AST7 to AST1 have no effect on operation.

ASTCR is initialized to H'FF by a power-on reset and in hardware standby mode. It is
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 w
corresponding area is to be designated as a 2-state access space or a 3-state access spa
normal mode*, only part of area 0 is enabled, and the AST0 bit selects whether extern
to be designated for 2-state access or 3-state access.

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

Note: * ZTAT, mask ROM, and ROMless versions only.

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

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
They are not initialized by a manual reset or in software standby mode.

Note: * ZTAT, mask ROM, and ROMless versions only.

(1) WCRH

Bit	:	7	6	5	4	3	2	1
		W71	W70	W61	W60	W51	W50	W41
Initial value	:	1	1	1	1	1	1	1
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 numb
program wait states when area 7 in external space is accessed while the AST7 bit in AS
to 1.

Bit 7	Bit 6	
W71	**W70**	**Description**
0	0	Program wait not inserted when external space area 7 is acces
	1	1 program wait state inserted when external space area 7 is ac
1	0	2 program wait states inserted when external space area 7 is a
	1	3 program wait states inserted when external space area 7 is a (I

1	0	2 program wait states inserted when external space area 6 is
	1	3 program wait states inserted when external space area 6 is

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

Bit 3	Bit 2	
W51	**W50**	**Description**
0	0	Program wait not inserted when external space area 5 is acce
	1	1 program wait state inserted when external space area 5 is a
1	0	2 program wait states inserted when external space area 5 is
	1	3 program wait states inserted when external space area 5 is

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

Bit 1	Bit 0	
W41	**W40**	**Description**
0	0	Program wait not inserted when external space area 4 is acce
	1	1 program wait state inserted when external space area 4 is a
1	0	2 program wait states inserted when external space area 4 is
	1	3 program wait states inserted when external space area 4 is

program wait states when area 3 in external space is accessed while the AST3 bit in AS
to 1.

Bit 7	Bit 6	
W31	**W30**	**Description**
0	0	Program wait not inserted when external space area 3 is acces
	1	1 program wait state inserted when external space area 3 is ac
1	0	2 program wait states inserted when external space area 3 is a
	1	3 program wait states inserted when external space area 3 is a

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

Bit 5	Bit 4	
W21	**W20**	**Description**
0	0	Program wait not inserted when external space area 2 is acces
	1	1 program wait state inserted when external space area 2 is ac
1	0	2 program wait states inserted when external space area 2 is a
	1	3 program wait states inserted when external space area 2 is a

1	0	2 program wait states inserted when external space area 1 is
	1	3 program wait states inserted when external space area 1 is

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

Bit 1	Bit 0	
W01	**W00**	**Description**
0	0	Program wait not inserted when external space area 0 is acce
	1	1 program wait state inserted when external space area 0 is a
1	0	2 program wait states inserted when external space area 0 is
	1	3 program wait states inserted when external space area 0 is

6.2.4 Bus Control Register H (BCRH)

Bit	:	7	6	5	4	3	2	1
		ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—
Initial value	:	1	1	0	1	0	0	0
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 c insertion, and the memory interface for areas 2 to 5 and area 0.

BCRH is initialized to H'D0 by a power-on reset and in hardware standby mode. It is initialized by a manual reset or in software standby mode.

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

Bit 6

ICIS0	Description
0	Idle cycle not inserted in case of successive external read and external wri
1	Idle cycle inserted in case of successive external read and external write cy (I

Bit 5—Burst ROM Enable (BRSTRM): Selects whether area 0 is used as a burst RO
interface. In normal mode*, the selection can be made from the entire external space.

Burst ROM interface and PSRAM burst operation cannot be set at the same time.

Note: * ZTAT, mask ROM, and ROMless versions only.

Bit 5

BRSTRM	Description
0	Area 0 is basic bus interface (I
1	Area 0 is burst ROM interface

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

Bit 4

BRSTS1	Description
0	Burst cycle comprises 1 state
1	Burst cycle comprises 2 states (I

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
		BRLE	—	EAE	—	—	—	—
Initial value	:	0	0	1	1	1	1	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 state protocol, and enabling or disabling of $\overline{\text{WAIT}}$ pin input.

BCRL is initialized to H'3C by a power-on reset and in hardware standby mode. It is n 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 po
1	External bus release is enabled.

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

Bit 5—External Address Enable (EAE): Selects whether addresses H'010000 to H'0 to be internal addresses or external addresses.

This setting is invalid in normal mode*.

Note: * ZTAT, mask ROM, and ROMless versions only.

	or a reserved area* (single-chip mode)	(

Note: * Reserved areas should not be accessed.

Bits 4 to 2—Reserved: Only 1 should be written to these bits.

Bit 1—Reserved: Only 0 should be written to this bit.

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

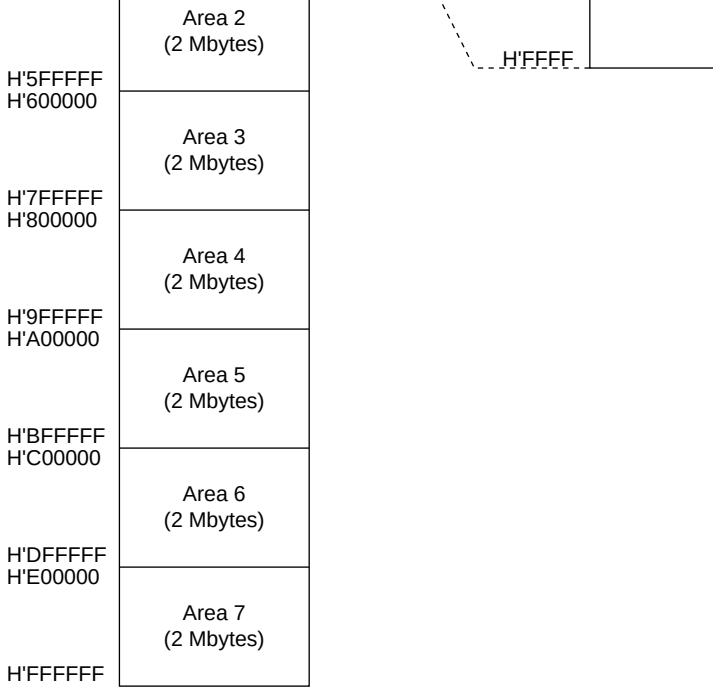
6.3 Bus Control

6.3.1 Area Divisions

In advanced mode, the bus controller partitions the 16-Mbyte address space into eight a
7, in 2-Mbyte units, and performs bus control for external space in area units. In norma
controls a 64-kbyte address space comprising part of area 0. Figure 6.2 shows an outlin
memory map.

Chip select signals ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$) can be output for areas 0 to 3.

Note: * ZTAT, mask ROM, and ROMless versions only.



(1) Advanced mode (2) Normal mod

Note: * ZTAT, mask ROM, and ROMless versions only.

Figure 6.2 Overview of Area Partitioning

8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit b
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 designate
access, 16-bit bus mode is set. When the burst ROM interface is designated, 16-bit bus
always set.

(2) Number of Access States: Two or three access states can be selected with ASTCR
for which 2-state access is selected functions as a 2-state access space, and an area for w
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
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 AS
number of program wait states to be inserted automatically is selected with WCRH and
From 0 to 3 program wait states can be selected.

Table 6.3 shows the bus specifications for each basic bus interface area.

		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/2345 Group memory interfaces comprise a basic bus interface that allows di
connection of ROM, SRAM, and so on, and a burst ROM interface (for area 0 only) th
direct connection of burst ROM.

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

6.3.4 Advanced Mode

The initial state of each area is basic bus interface, 3-state access space. The initial bus
selected according to the operating mode. The bus specifications described here cover
only, and the sections on each memory interface (6.4, Basic Bus Interface and 6.5, Bu
Interface) should be referred to for further details.

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

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

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

RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1
RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space be
external space.

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

6.3.5 Areas in Normal Mode (ZTAT, Mask ROM, and ROMless Versions Onl

In normal mode, a 64-kbyte address space comprising part of area 0 is controlled. Area
partitioning is not performed in normal mode. In ROM-disabled expansion mode, the s
excluding the on-chip RAM and internal I/O registers is external space. In ROM-enabl
expansion mode the space excluding the on-chip ROM, on-chip RAM, and internal I/O
external space. The on-chip RAM is enabled when the RAME bit in the system control
(SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled a
corresponding space becomes external space .

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

The basic bus interface or burst ROM interface can be selected.

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 po
Pins $\overline{CS1}$ to $\overline{CS3}$ are placed in the input state after a power-on reset, and so the corresp
DDR should be set to 1 when outputting signals $\overline{CS1}$ to $\overline{CS3}$.

In ROM-enabled expansion mode, pins $\overline{CS0}$ to $\overline{CS3}$ are all placed in the input state aft
on reset, and so the corresponding DDR should be set to 1 when outputting signals $\overline{CS}$

For details, see section 8, I/O Ports.

Note: * ZTAT, mask ROM, and ROMless versions only.

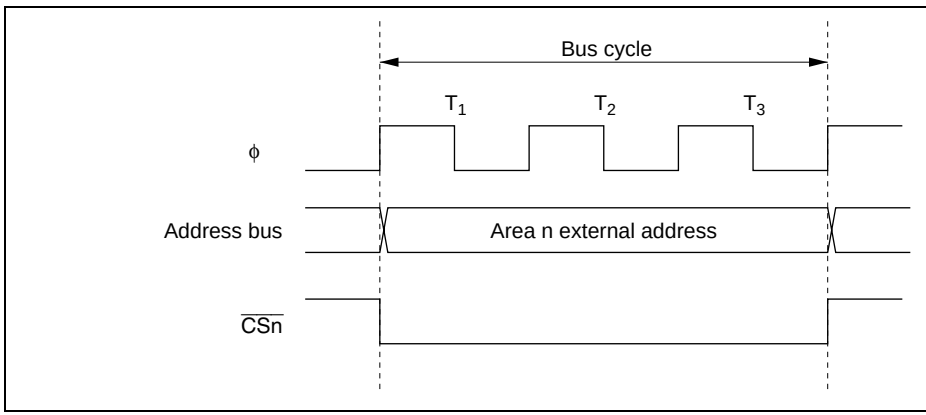


Figure 6.3 $\overline{CSn}$ Signal Output Timing (n = 0 to 3)

6.4.2 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. Th
controller has a data alignment function, and when accessing external space, controls w
upper data bus (D_{15} to D_8) or lower data bus (D_7 to D_0) is used according to the bus spec
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 sp
the 8-bit access space, the upper data bus (D_{15} to D_8) is always used for accesses. The ar
data that can be accessed at one time is one byte; a word transfer instruction is perform
byte accesses, and a longword transfer instruction, as four byte accesses.

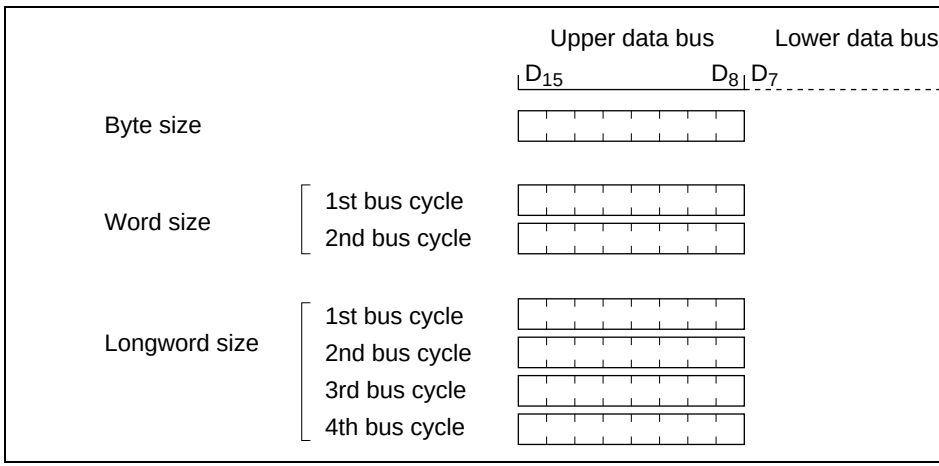


Figure 6.4 Access Sizes and Data Alignment Control (8-Bit Access Spac

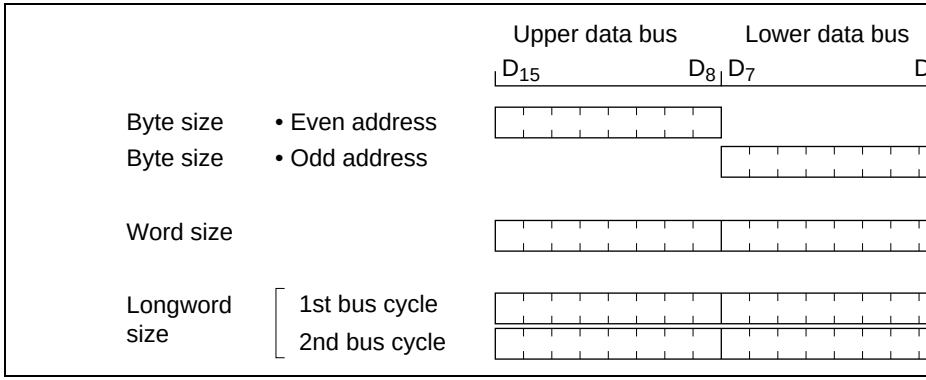


Figure 6.5 Access Sizes and Data Alignment Control (16-Bit Access Spa

Table 6.4 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D_{15} to D_8)	Lower (D_7 to
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$		Hi-Z
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Hi-Z
			Odd	$\overline{LWR}$	Hi-Z	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

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

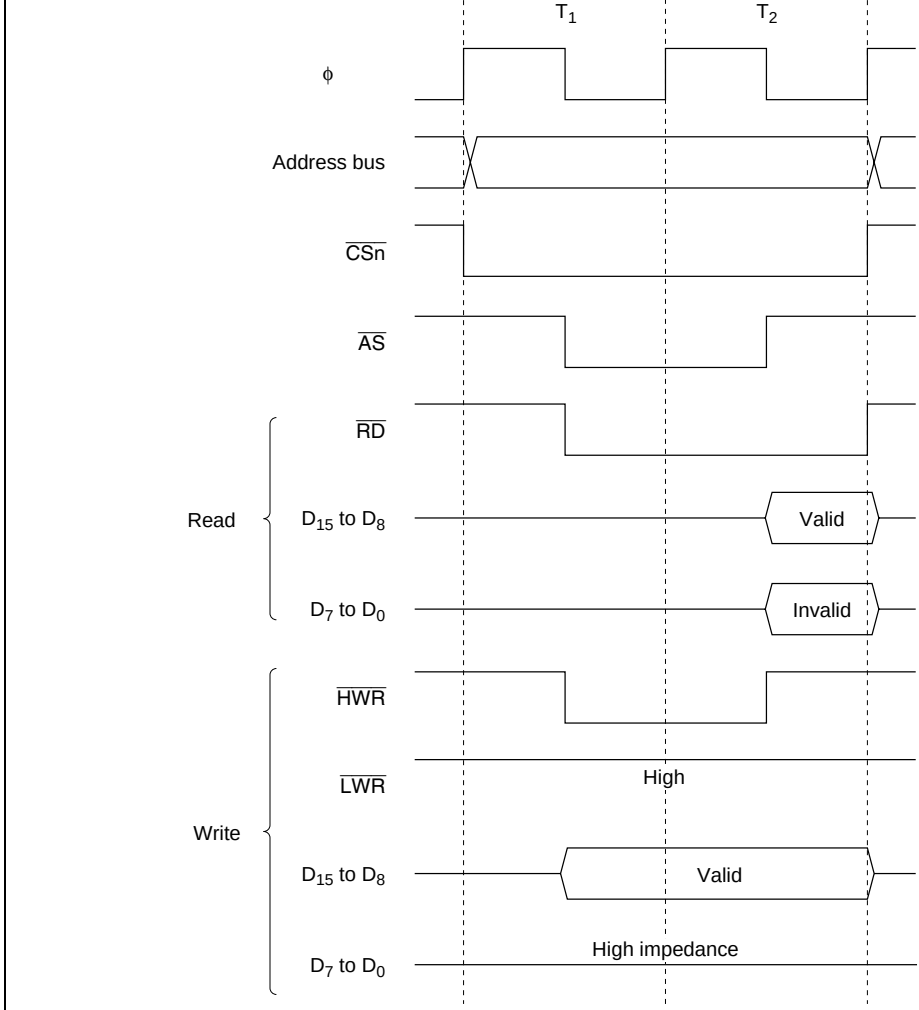


Figure 6.6 Bus Timing for 8-Bit 2-State Access Space

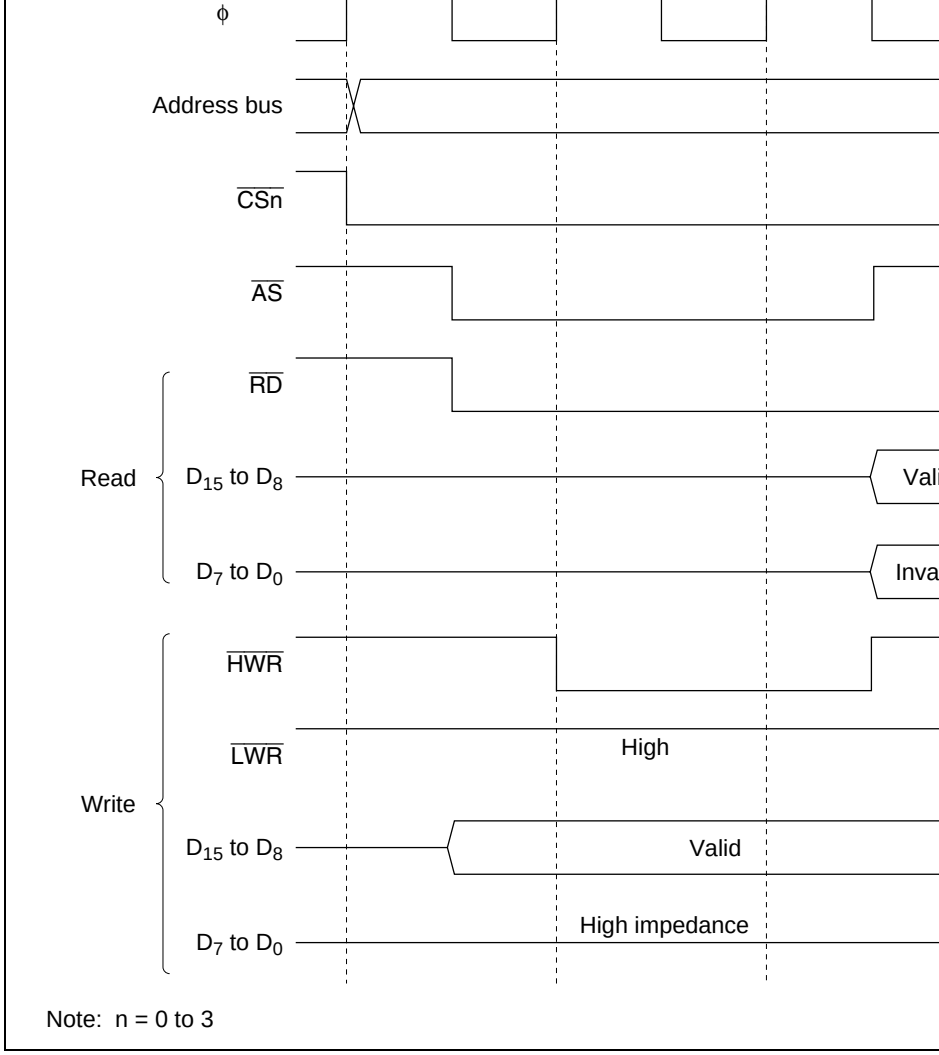


Figure 6.7 Bus Timing for 8-Bit 3-State Access Space

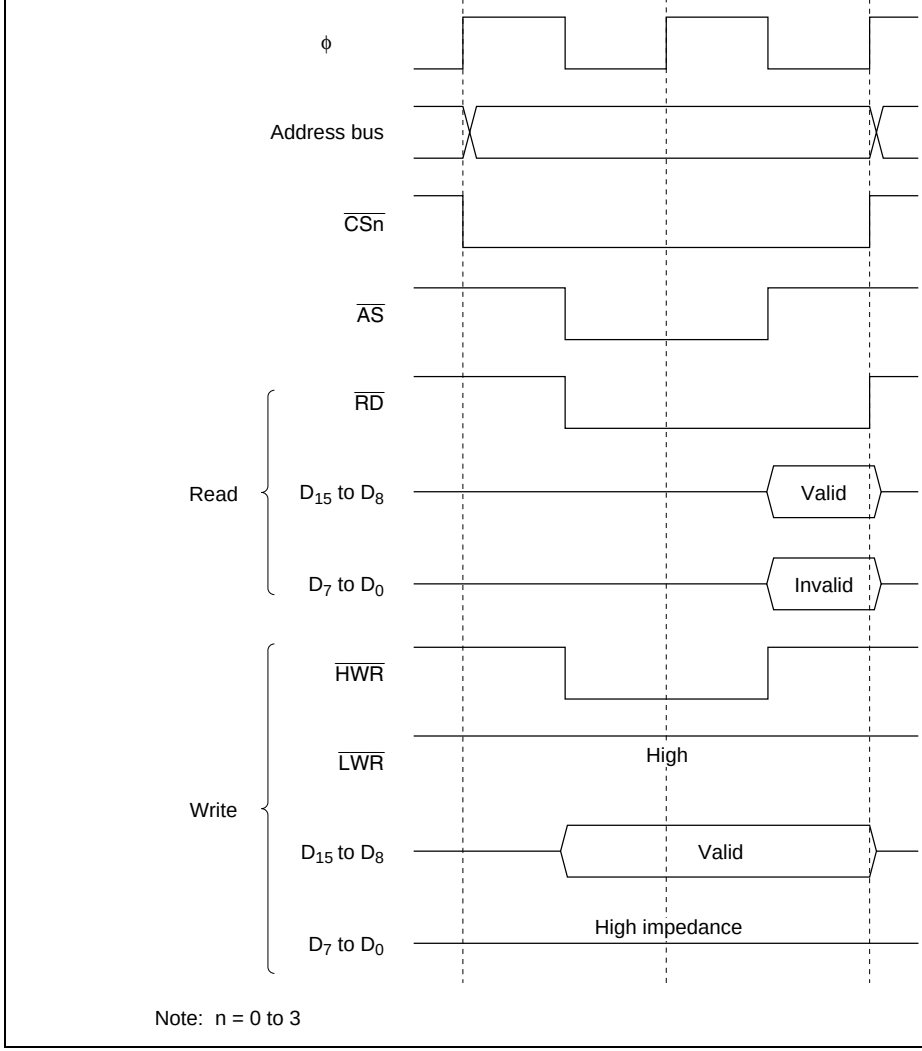


Figure 6.8 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byt

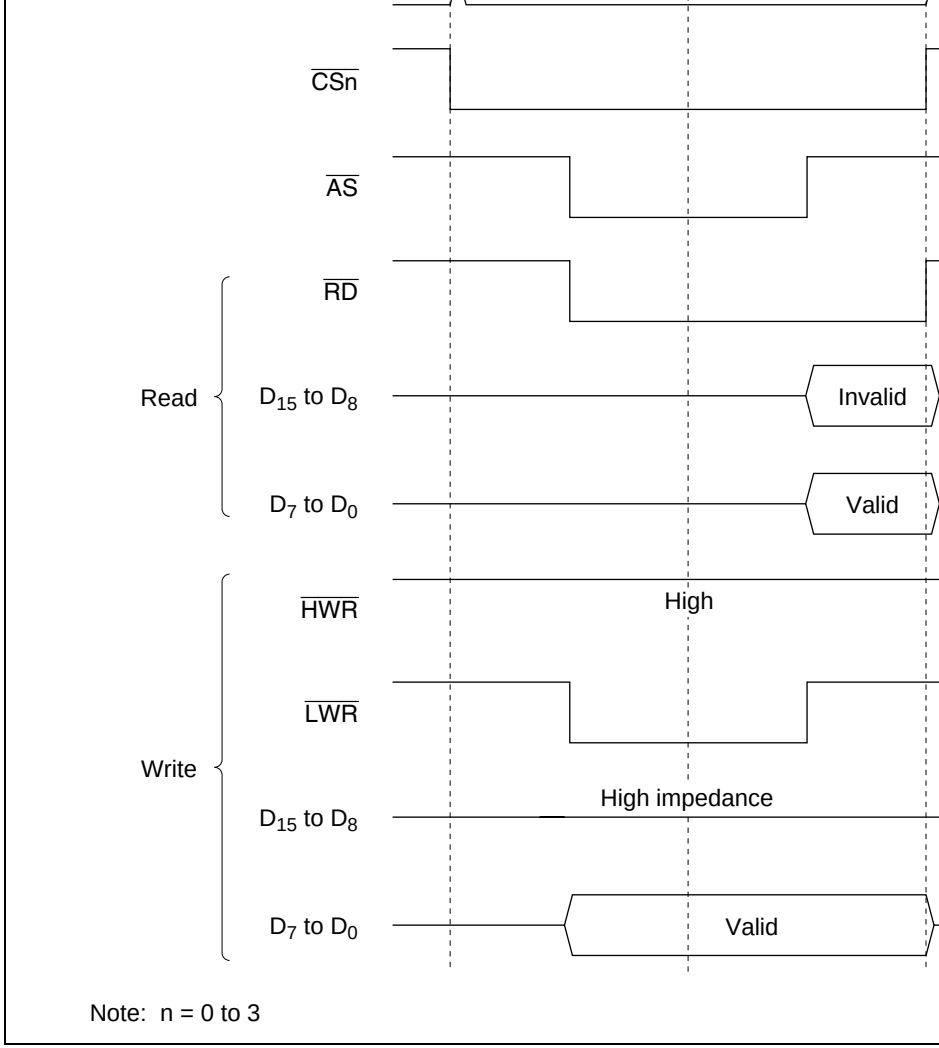


Figure 6.9 Bus Timing for 16-Bit 2-State Access Space (2) (Odd Address Byte

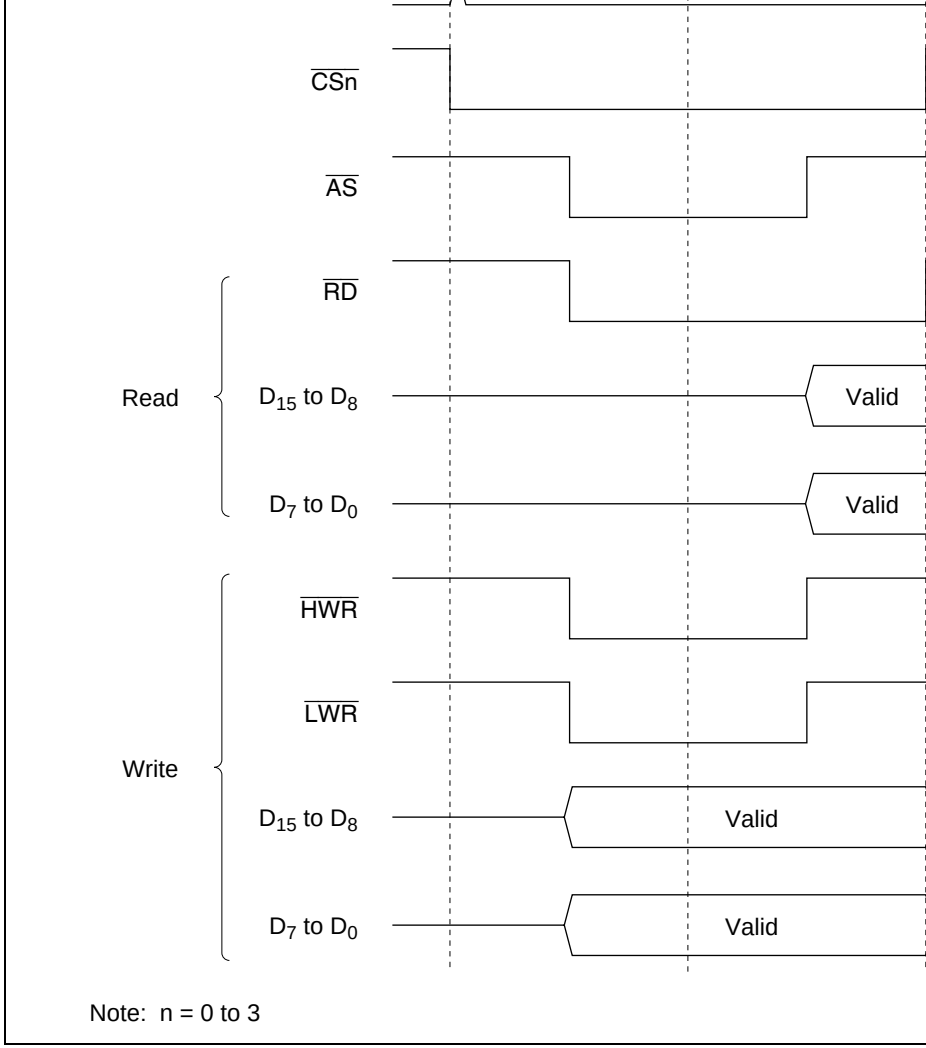


Figure 6.10 Bus Timing for 16-Bit 2-State Access Space (3) (Word Acc

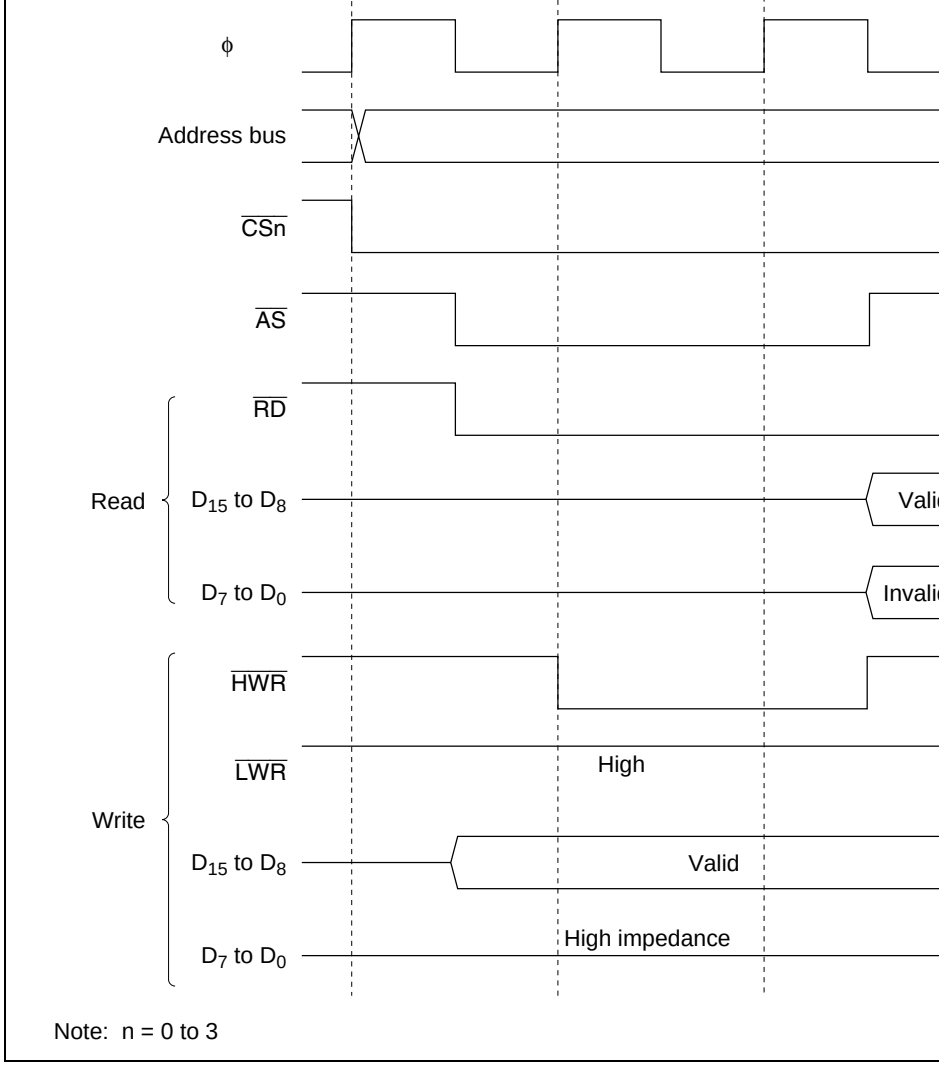


Figure 6.11 Bus Timing for 16-Bit 3-State Access Space (1) (Even Address Byte

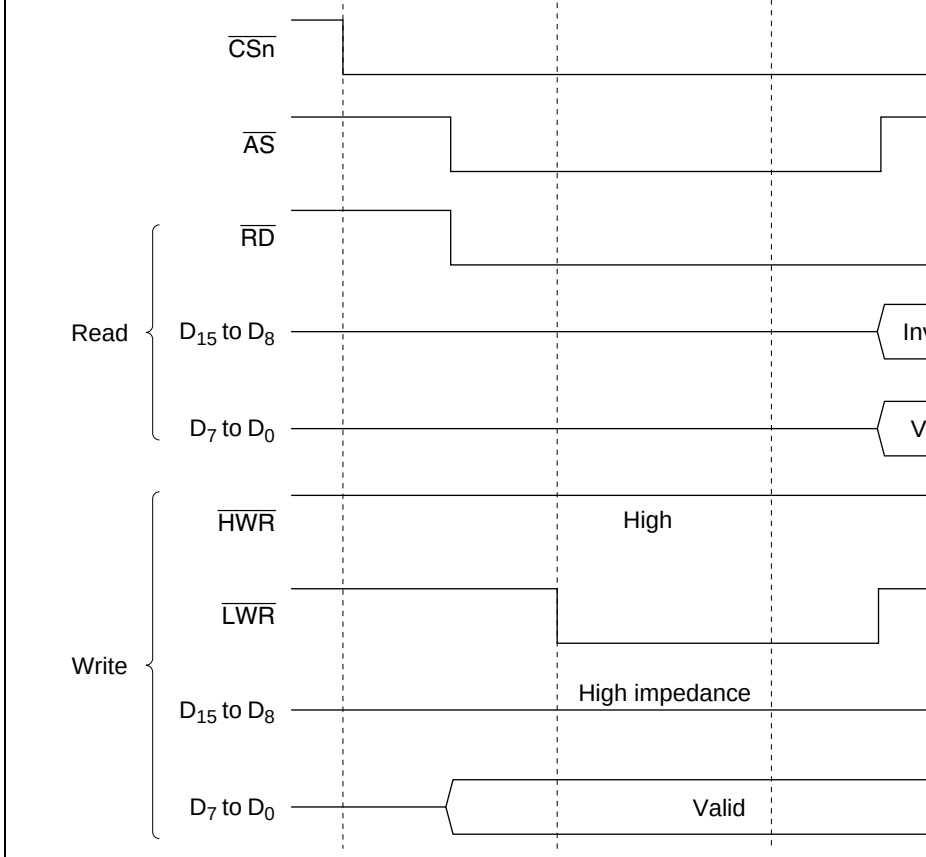


Figure 6.12 Bus Timing for 16-Bit 3-State Access Space (2) (Odd Address Byt

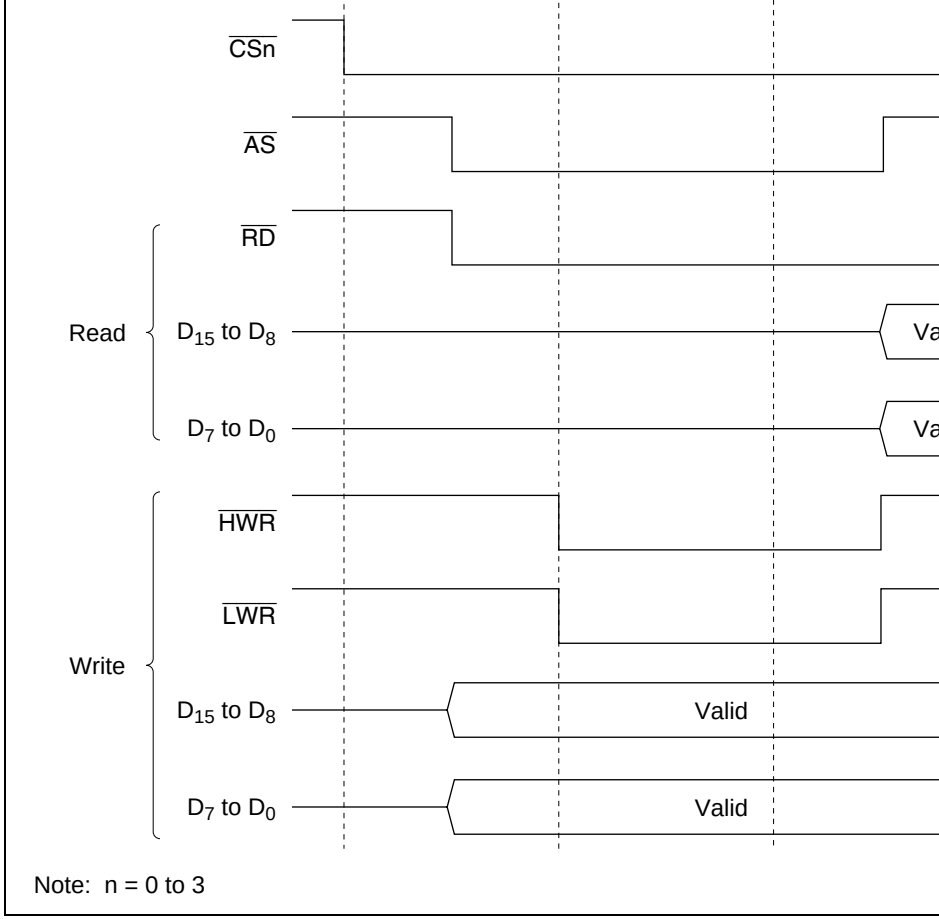


Figure 6.13 Bus Timing for 16-Bit 3-State Access Space (3) (Word Acces

individual area basis in 3-state access space, according to the settings of BWCRH and

Pin Wait Insertion

Setting the WAITE bit in BCRL to 1 enables wait insertion by means of the $\overline{\text{WAIT}}$ pi
wait insertion is first carried out according to the settings in WCRH and WCRL. Then
$\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_w state, a T_w state is inserted
$\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

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

The WAITE bit setting applies to all areas.

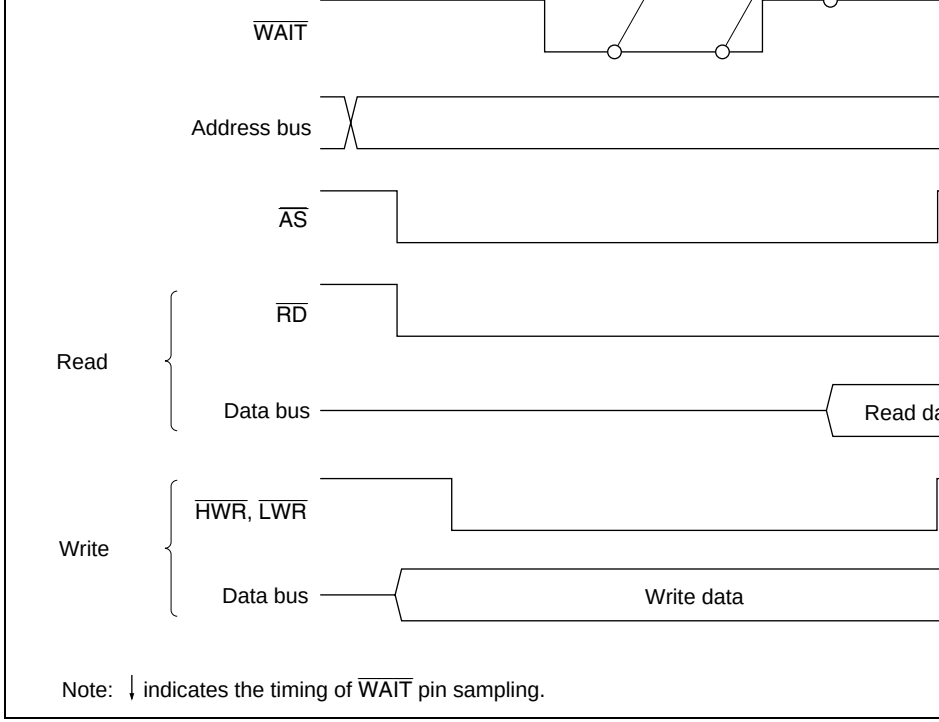


Figure 6.14 Example of Wait State Insertion Timing

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

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 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 (a) and (b). The timing shown in figure 6.15 (a) is for the case where the AST0 and BRSTS1 bits are both set to 1, and that in figure 6.15 (b) is for the case where both these bits are cleared to 0.

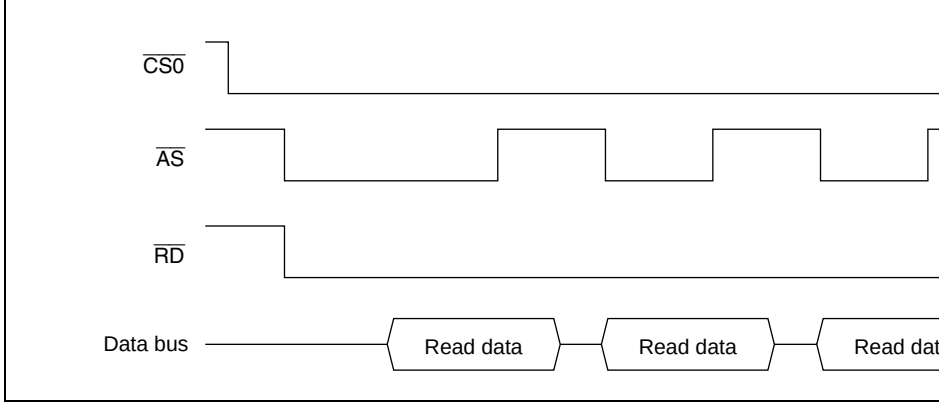


Figure 6.15 (a) Example of Burst ROM Access Timing (When AST0 = BRST

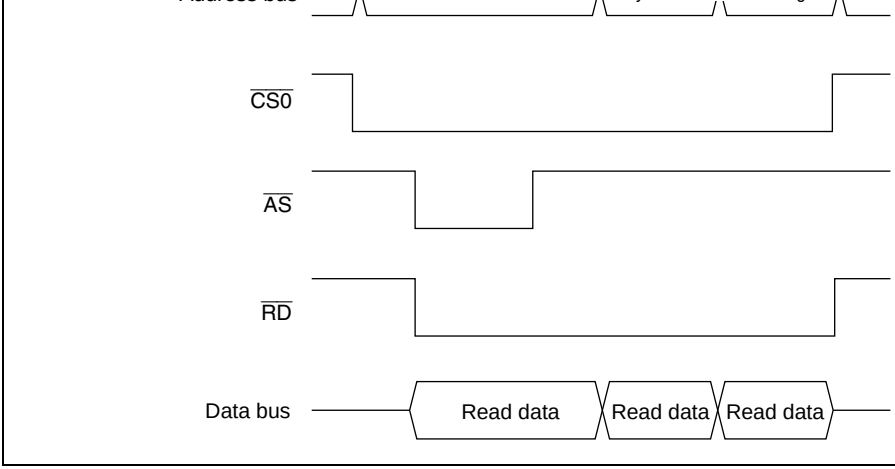


Figure 6.15 (b) Example of Burst ROM Access Timing (When AST0 = BRS

6.5.3 Wait Control

As with the basic bus interface, either program wait insertion or pin wait insertion usi
pin can be used in the initial cycle (full access) of the burst ROM interface. See sectio
Control.

Wait states cannot be inserted in a burst cycle.

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 t
cycle is inserted at the start of the second read cycle. This is enabled in advanced mode

Figure 6.16 shows an example of the operation in this case. In this example, bus cycle A
cycle from ROM with a long output floating time, and bus cycle B is a read cycle from
each being located in a different area. In (a), an idle cycle is not inserted, and a collision
cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is
and a data collision is prevented.

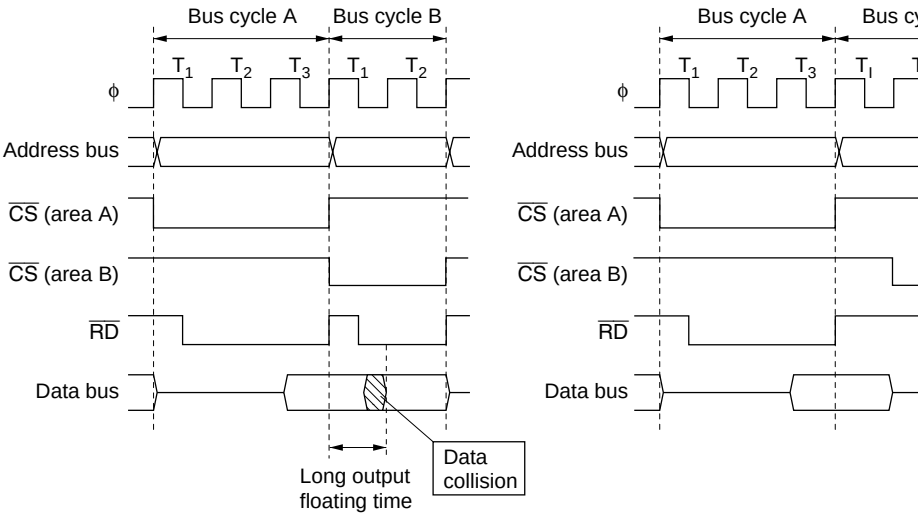


Figure 6.16 Example of Idle Cycle Operation (1)

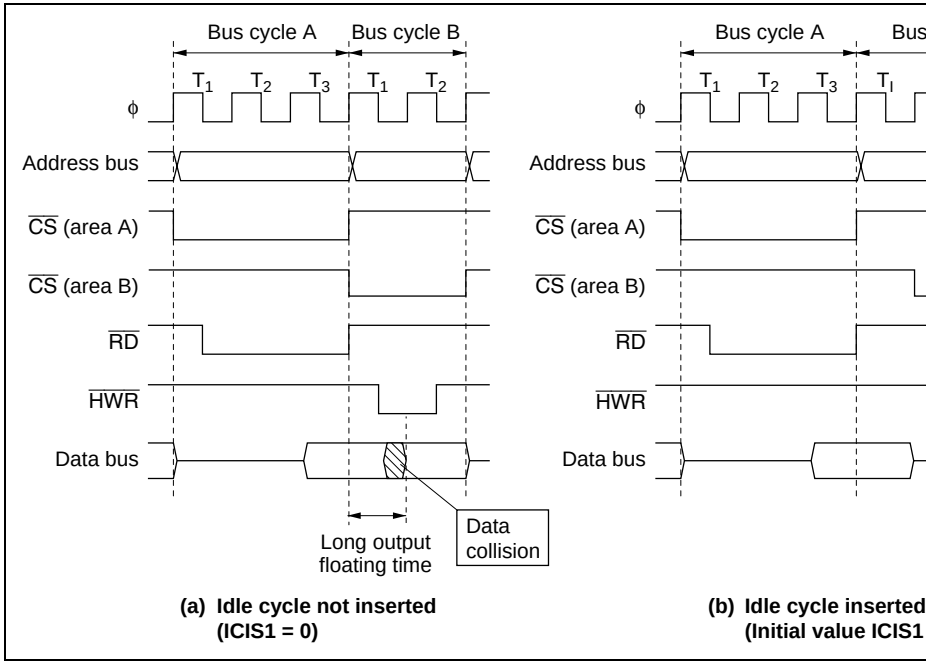


Figure 6.17 Example of Idle Cycle Operation (2)

signals.

In the initial state after reset release, idle cycle insertion (b) is set.

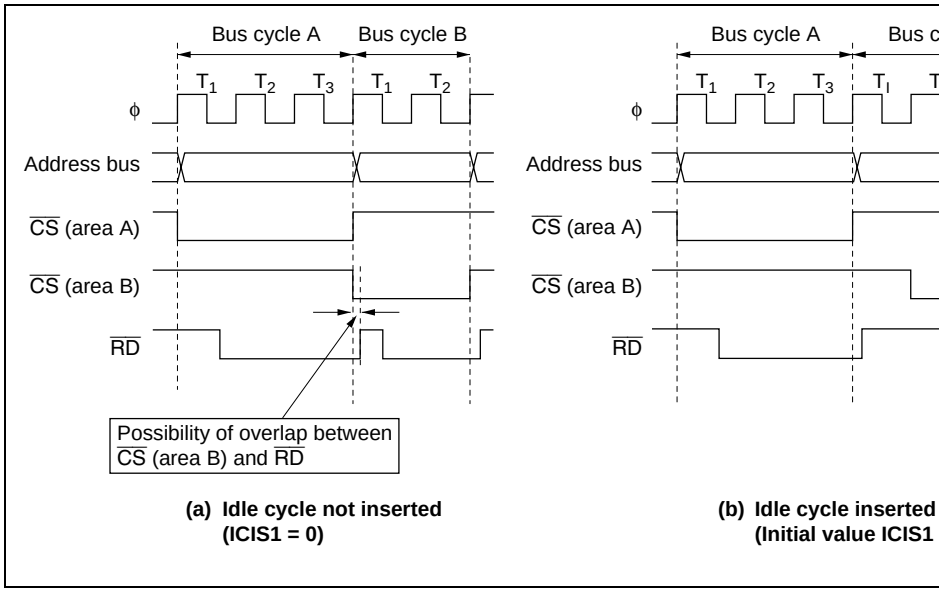


Figure 6.18 Relationship between Chip Select ($\overline{CS}$) and Read ($\overline{RD}$)

$\overline{CSn}$	High
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{HWR}$	High
$\overline{LWR}$	High

6.7 Bus Release

6.7.1 Overview

The H8S/2345 Group can release the external bus in response to a bus request from an
device. In the external bus released state, the internal bus master continues to operate
there is no external access.

6.7.2 Operation

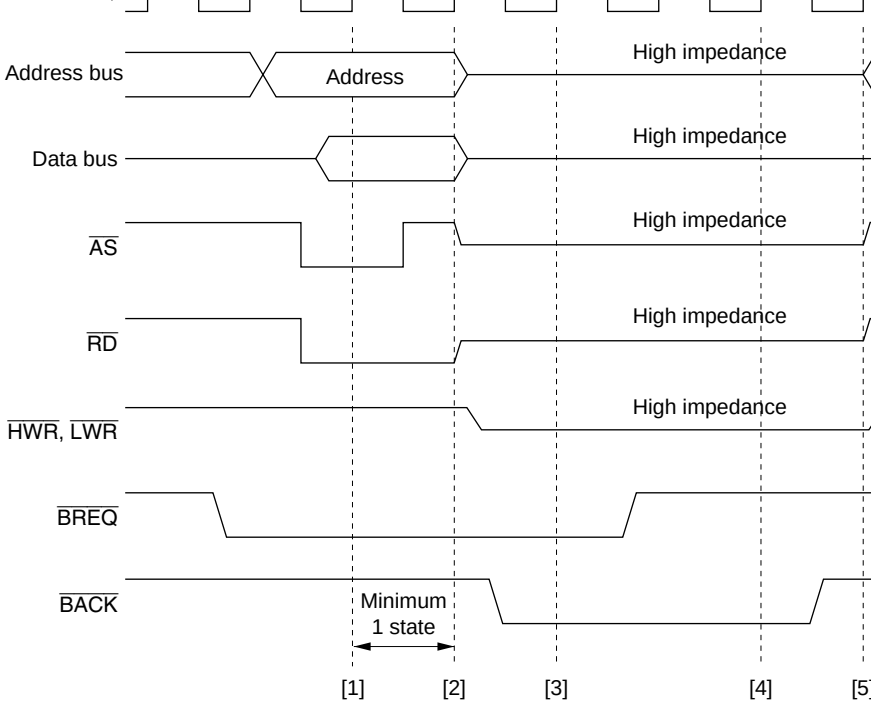
In external expansion mode, the bus can be released to an external device by setting th
in BCRL to 1. Driving the $\overline{BREQ}$ pin low issues an external bus request to the H8S/23
When the $\overline{BREQ}$ pin is sampled, at the prescribed timing the $\overline{BACK}$ pin is driven low,
address bus, data bus, and bus control signals are placed in the high-impedance state,
the external bus-released state.

In the external bus released state, an internal bus master can perform accesses using th
bus. When an internal bus master wants to make an external access, it temporarily def
activation of the bus cycle, and waits for the bus request from the external bus master
dropped.

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

Table 6.6 Pin States in Bus Released State

Pins	Pin State
A_{23} to A_0	High impedance
D_{15} to D_0	High impedance
$\overline{CSn}$	High impedance
$\overline{AS}$	High impedance
$\overline{RD}$	High impedance
$\overline{HWR}$	High impedance
$\overline{LWR}$	High impedance



[1] Low level of $\overline{\text{BREQ}}$ pin is sampled at rise of T_2 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.

Figure 6.19 Bus-Released State Transition Timing

6.8.1 Overview

The H8S/2345 Group has a bus arbiter that arbitrates bus master operations.

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

6.8.2 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested,
request acknowledge signal to the bus master making the request. If there are bus reque
more than one bus master, the bus request acknowledge signal is sent to the one with th
priority. When a bus master receives the bus request acknowledge signal, it takes posse
bus until that signal is canceled.

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

(High) DTC > CPU (Low)

An internal bus access by an internal bus master, and external bus release, can be execu
parallel.

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

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

- The bus is transferred at a break between bus cycles. However, if a bus cycle is ex
  discrete operations, as in the case of a longword-size access, the bus is not transfer
  the operations. See appendix A.5, Bus States during Instruction Execution, for tim
  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

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

6.8.4 External Bus Release Usage Note

External bus release can be performed on completion of an external bus cycle. The $\overline{R}$
$\overline{CS0}$ to $\overline{CS3}$ signals remain low until the end of the external bus cycle. Therefore, wh
bus release is performed, the $\overline{RD}$ and $\overline{CS0}$ to $\overline{CS3}$ signals may change from the low le
high-impedance state.

6.9 Resets and the Bus Controller

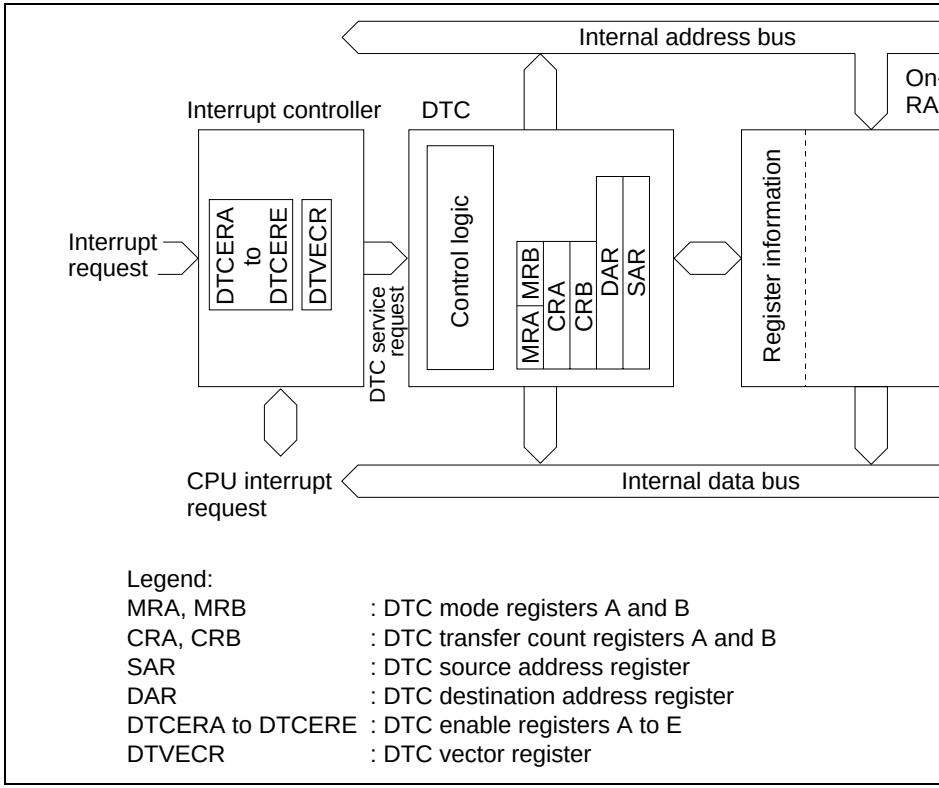
In a power-on reset, the H8S/2345, including the bus controller, enters the reset state a
and an executing bus cycle is discontinued.

In a manual reset, the bus controller's registers and internal state are maintained, and a
external bus cycle is completed. In this case, $\overline{WAIT}$ input is ignored and write data is
guaranteed.

7.1.1 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 ca
- 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 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 halt module stop mode.



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 DTCERE	: DTC enable registers A to E
DTVECR	: DTC vector register

Figure 7.1 Block Diagram of DTC

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'FF3
DTC vector register	DTVECR	R/W	H'00	H'FF3
Module stop control register	MSTPCR	R/W	H'3FFF	H'FF3

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'F800 to H'FBF be located in external space. When the DTC is used, do not clear the RAM SYSCR to 0.

Initial value :	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
R/W :	—	—	—	—	—	—	—

Bits 7 and 6—Source Address Mode 1 and 0 (SM1, SM0): These bits specify whethe 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 v 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)

Bit 1—DTC Transfer Mode Select (DTS): Specifies whether the source side or the d
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

MRB is an 8-bit register that controls the DTC operating mode.

Bit 7—DTC Chain Transfer Enable (CHNE): Specifies chain transfer. With chain t
number of data transfers can be performed consecutively in response to a single transfe

In data transfer with CHNE set to 1, determination of the end of the specified number o
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 transferre

Bit 6—DTC Interrupt Select (DISEL): Specifies whether interrupt requests to the CP
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 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 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/2345 G
should always be written with 0 in a write.

SAR is a 24-bit register that designates the source address of data to be transferred by
For word-size transfer, specify an even source address.

7.2.4 DTC Destination Address Register (DAR)

Bit	:	23	22	21	20	19	- - -	4	3	2
							- - -			
Initial value	:	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	- - -	Unde-fined	Unde-fined	Und fin
R/W	:	—	—	—	—	—	- - -	—	—	–

DAR is a 24-bit register that designates the destination address of data to be transferre
DTC. For word-size transfer, specify an even destination address.

CRA is a 16-bit register that designates the number of times data is to be transferred by

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65536). It i
decremented by 1 every time data is transferred, and transfer ends when the count reach

In repeat mode or block transfer mode, the CRA is divided into two parts: the upper 8 b
(CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CR
functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time
transferred, and the contents of CRAH are sent when the count reaches H'00. This oper
repeated.

7.2.6 DTC Transfer Count Register B (CRB)

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

CRB is a 16-bit register that designates the number of times data is to be transferred by
block transfer mode. It functions as a 16-bit transfer counter (1 to 65536) that is decrem
every time data is transferred, and transfer ends when the count reaches H'0000.

The DTC enable registers comprise five 8-bit readable/writable registers, DTCERA to with bits corresponding to the interrupt sources that can activate the DTC. These bits disable DTC service for the corresponding interrupt sources.

The DTC enable registers are initialized to H'00 by a reset and in hardware standby m

A DTCE bit can be set for each interrupt source that can activate the DTC. The corres between interrupt sources and DTCE bits is shown in table 7.4, together with the vecto generated for each interrupt controller.

For DTCE bit setting, read/write operations must be performed using bit-manipulation such as BSET and BCLR. For the initial setting only, however, when multiple activati are set at one time, it is possible to disable interrupts and write after executing a dumn the relevant register.

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
1	DTC activation by this interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified number of transfers have not en

A DTCE bit can be set for each interrupt source that can activate the DTC. The corres between interrupt sources and DTCE bits is shown in table 7.4, together with the vecto generated for each interrupt controller.

is read.

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation 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 activa software.

When clearing the SWDTE bit to 0 by software, write 0 to SWDTE after reading SWD

Bit 7

SWDTE	Description
0	DTC software activation is disabled ([Clearing condition] When the DISEL bit is 0 and the specified number of transfers have not en
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): Thes specify a vector number for DTC software activation.

The vector address is expressed as H'0400 + ((vector number) << 1). <<1 indicates a o shift. For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420.

MSTPCR is a 16-bit readable/writable register that performs module stop mode contro

When the MSTP14 bit in MSTPCR is set to 1, the DTC operation stops at the end of t
and a transition is made to module stop mode. However, 1 cannot be written in the MS
while the DTC is operating. For details, see section 19.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not in
software standby mode.

Bit 14—Module Stop (MSTP14): Specifies the DTC module stop mode.

Bit 14

MSTP14	Description
0	DTC module stop mode cleared
1	DTC module stop mode set

to perform a number of transfers with a single activation.

Figure 7.2 shows a flowchart of DTC operation.

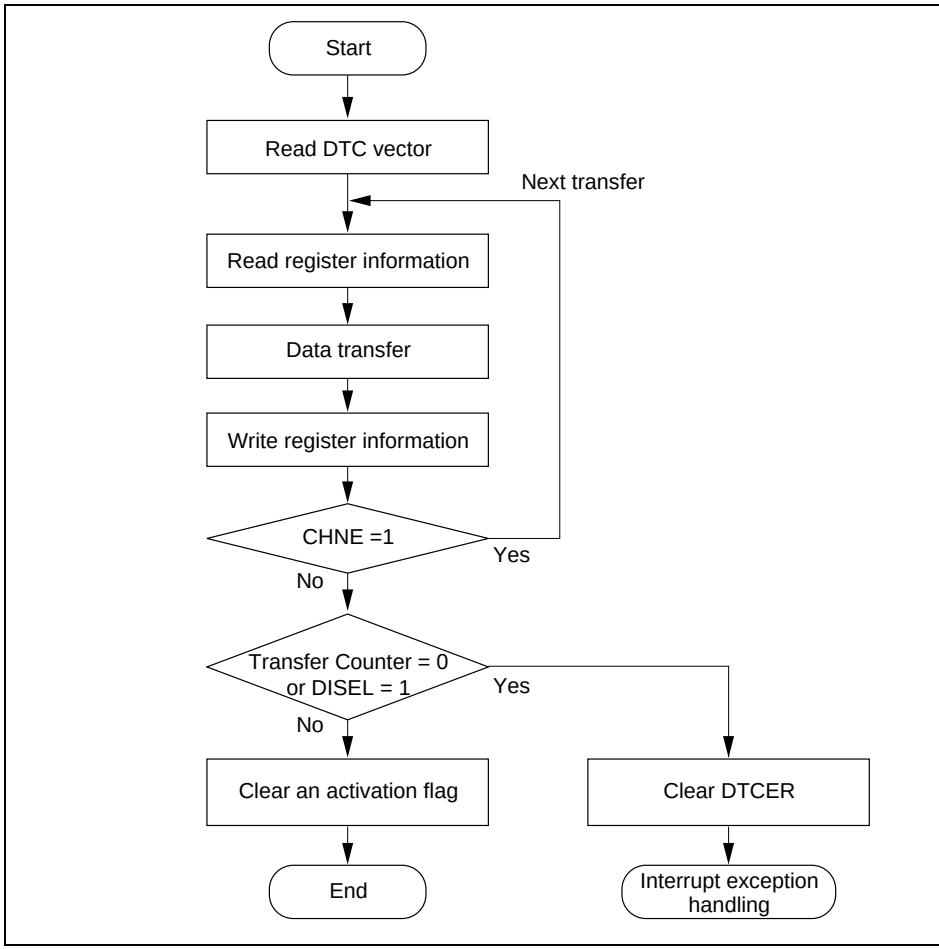


Figure 7.2 Flowchart of DTC Operation

Transfer Mode	Activation Source	Address R Transfer Source	T D
• 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	2

DTCER clearance. The activation source flag, in the case of RXI0, for example, is the I
of SCI0.

Table 7.3 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 whe 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 c The activation source flag remains A request is issued to the CPU for activation source interrupt

Figure 7.3 shows a block diagram of activation source control. For details see section 5
Controller.

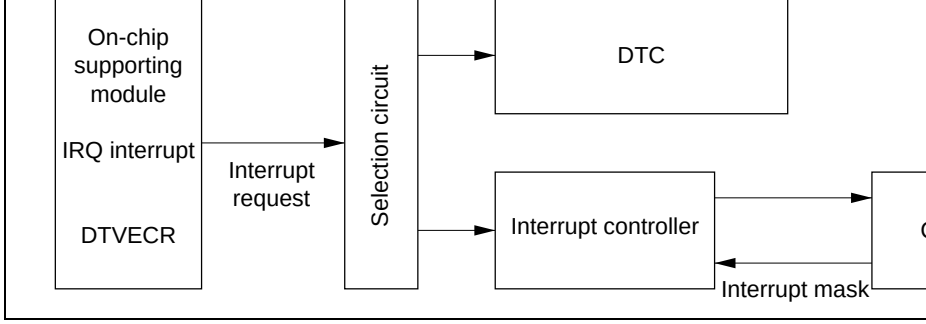


Figure 7.3 Block Diagram of DTC Activation Source Control

When an interrupt has been designated a DTC activation source, existing CPU mask l interrupt controller priorities have no effect. If there is more than one activation sourc time, the DTC operates in accordance with the default priorities.

7.3.3 DTC Vector Table

Figure 7.4 shows the correspondence between DTC vector addresses and register info

Table 7.4 shows the correspondence between activation, vector addresses, and DTCE the DTC is activated by software, the vector address is obtained from: H'0400 + (DTV << 1) (where << 1 indicates a 1-bit left shift). For example, if DTVECR is H'10, the v address is H'0420.

The DTC reads the start address of the register information from the vector address se activation source, and then reads the register information from that start address. The information can be placed at predetermined addresses in the on-chip RAM. The start the register information should be an integral multiple of four.

The configuration of the vector address is the same in both normal and advanced mod unit being used in both cases. These two bytes specify the lower bits of the address in RAM.

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
ADI (A/D conversion end)	A/D	28	H'0438	DTCEB6
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

TGI3D (GR3D compare match/ input capture)		51	H'0466	DTCEC2
TGI4A (GR4A compare match/ input capture)	TPU channel 4	56	H'0470	DTCEC1
TGI4B (GR4B compare match/ input capture)		57	H'0472	DTCEC0
TGI5A (GR5A compare match/ input capture)	TPU channel 5	60	H'0478	DTCED5
TGI5B (GR5B compare match/ input capture)		61	H'047A	DTCED4
CMIA0	8-bit timer channel 0	64	H'0480	DTCED3
CMIB0		65	H'0482	DTCED2
CMIA1	8-bit timer channel 1	68	H'0488	DTCED1
CMIB1		69	H'048A	DTCED0
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

Note: * DTCE bits with no corresponding interrupt are reserved, and should be writ

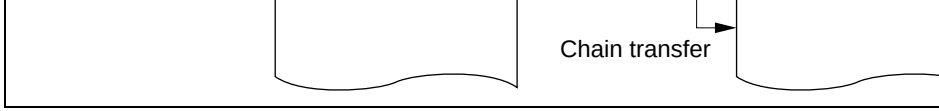


Figure 7.4 Correspondence between DTC Vector Address and Register Infor

7.3.4 Location of Register Information in Address Space

Figure 7.5 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 s of the register information (contents of the vector address). In the case of chain transfer information should be located in consecutive areas.

Locate the register information in the on-chip RAM (addresses: H'FFF800 to H'FFFBF

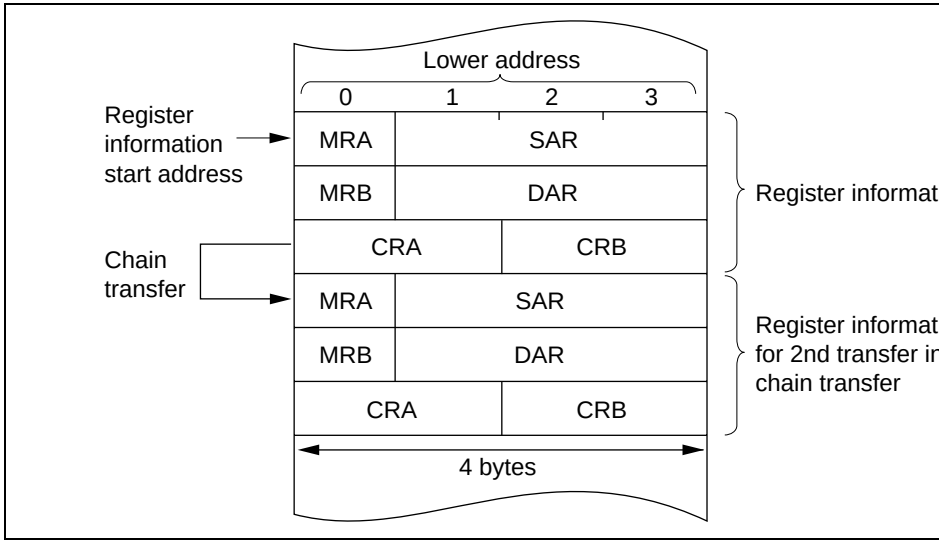


Figure 7.5 Location of Register Information in Address Space

Table 7.5 Register Information in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source addre
DTC destination address register	DAR	Designates destination a
DTC transfer count register A	CRA	Designates transfer coun
DTC transfer count register B	CRB	Not used

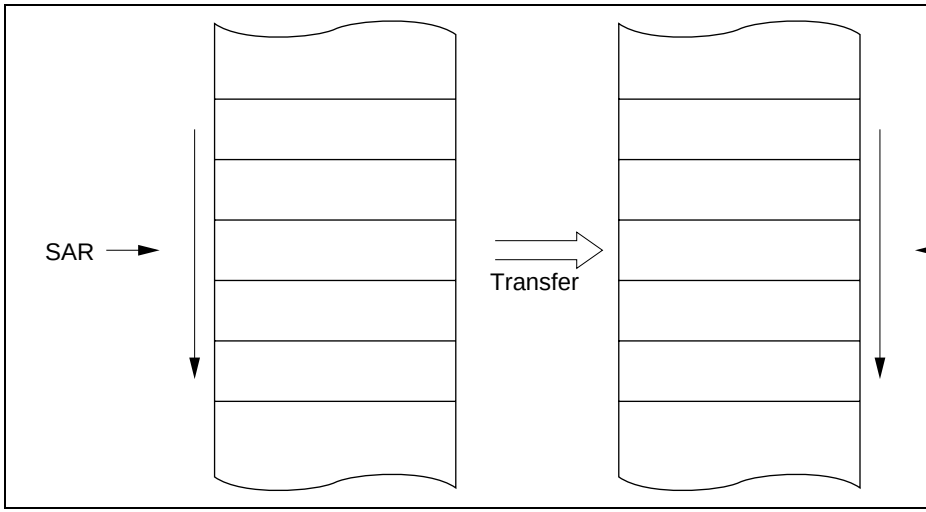


Figure 7.6 Memory Mapping in Normal Mode

Table 7.6 lists the register information in repeat mode and figure 7.7 shows memory ma
repeat mode.

Table 7.6 Register Information in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source addres
DTC destination address register	DAR	Designates destination ad
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

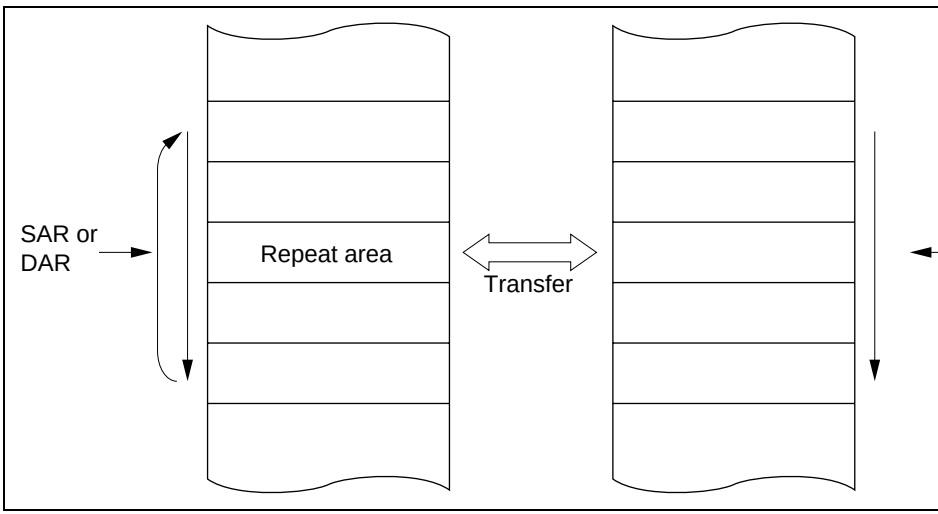


Figure 7.7 Memory Mapping in Repeat Mode

CPU interrupt is requested.

Table 7.7 lists the register information in block transfer mode and figure 7.8 shows me
mapping in block transfer mode.

Table 7.7 Register Information in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates transfer sour
DTC destination address register	DAR	Designates destination a
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Designates block size co
DTC transfer count register B	CRB	Transfer count

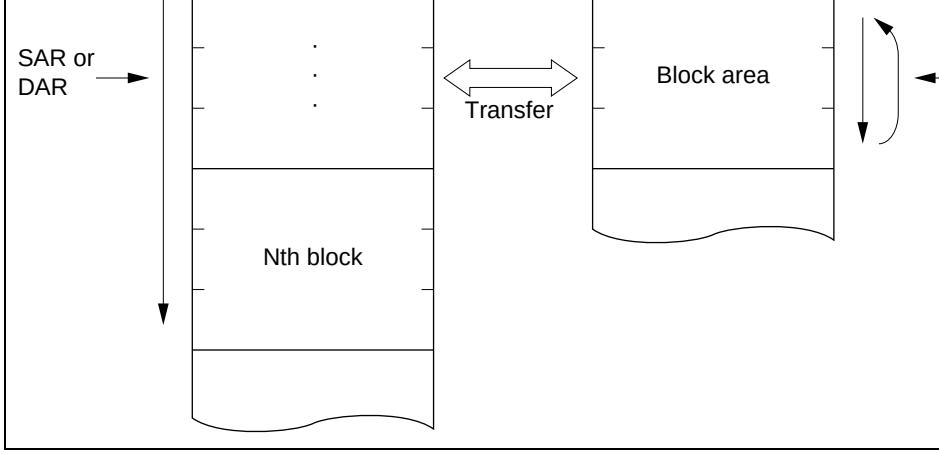


Figure 7.8 Memory Mapping in Block Transfer Mode

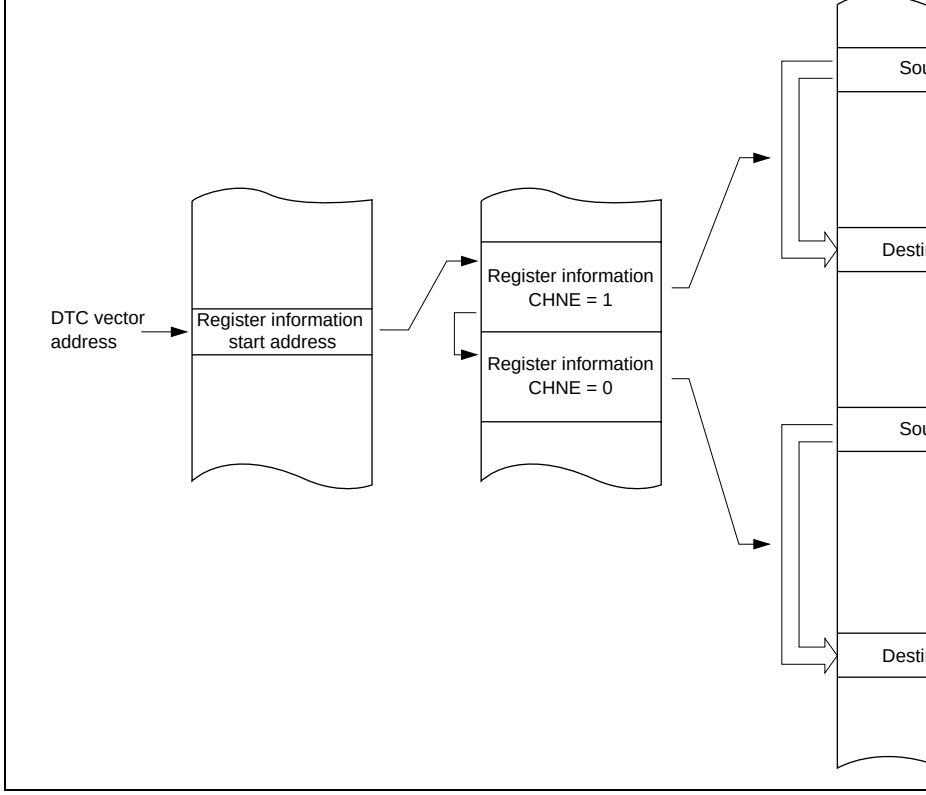


Figure 7.9 Chain Transfer Memory Map

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not gene end of the specified number of transfers or by setting of the DISEL bit to 1, and the in source flag for the activation source is not affected.

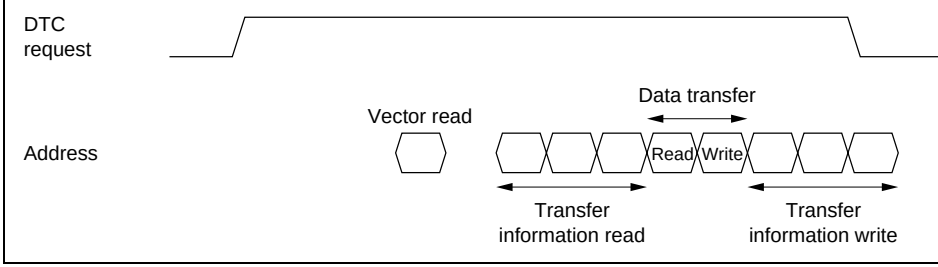


Figure 7.10 DTC Operation Timing (Example in Normal Mode or Repeat M

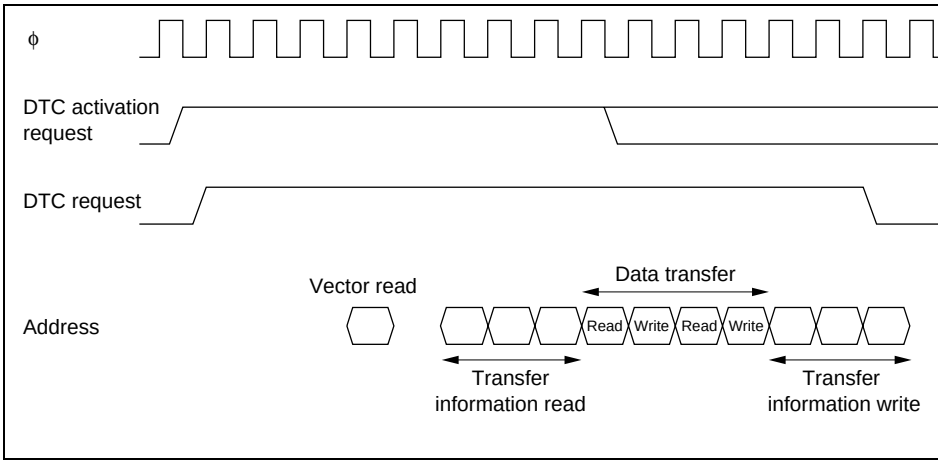


Figure 7.11 DTC Operation Timing (Example of Block Transfer Mode with Block Size of 2)

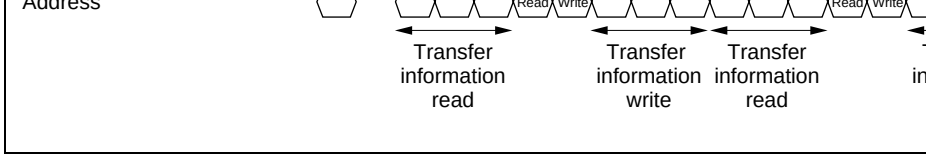


Figure 7.12 DTC Operation Timing (Example of Chain Transfer)

7.3.10 Number of DTC Execution States

Table 7.8 lists execution statuses for a single DTC data transfer, and table 7.9 shows t
states required for each execution status.

Table 7.8 DTC Execution Statuses

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	I C 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 CRA

	Register information read/write	S_J	1	—	—	—	—	—	—
	Byte data read	S_K	1	1	2	2	2	3+m	
	Word data read	S_K	1	1	4	2	4	6+2m	
	Byte data write	S_L	1	1	2	2	2	3+m	
	Word data write	S_L	1	1	4	2	4	6+2m	
	Internal operation	S_M	1						

The number of execution states is calculated from the formula below. Note that Σ mea
of all transfers activated by one activation event (the number in which the CHNE bit is
plus 1).

$$\text{Number of execution states} = I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$$

For example, when the DTC vector address table is located in on-chip ROM, normal m
and data is transferred from the on-chip ROM to an internal I/O register, the time requi
DTC operation is 13 states. The time from activation to the end of the data write is 10 s

[4] Set the enable bits for the interrupt sources to be used as the activation sources to 1
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 ha
the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to con
transferring data, set the DTCE bit to 1.

Activation by Software: The procedure for using the DTC with software activation is

[1] Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-ch

[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
the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the S
to 1. When the DISEL bit is 1, or after the specified number of data transfers have
SWDTE bit is held at 1 and a CPU interrupt is requested.

SCI RDR address in SAR, the start address of the RAM area where the data will be
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 t
reception complete (RXI) interrupt. Since the generation of a receive error during th
reception operation will disable subsequent reception, the CPU should be enabled to
receive error interrupts.

[5] Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is s
RXI interrupt is generated, and the DTC is activated. The receive data is transferred
to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF fl
automatically cleared to 0.

[6] When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is he
DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The inter
handling routine should perform wrap-up processing.

0). Set the transfer source address (H'1000) in SAR, the destination address (H'200
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 trans
by software.

[4] Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write d

[5] Read DTVECR again and check that it is set to the vector number (H'60). If it is n
indicates that the write failed. This is presumably because an interrupt occurred b
3 and 4 and led to a different software activation. To activate this transfer, go back

[6] If the write was successful, the DTC is activated and a block of 128 bytes of data i

[7] After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine
the SWDTE bit to 0 and perform other wrap-up processing.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of tra have ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND int 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 durin transfer wait or during data transfer even if the SWDTE bit is set to 1.

7.5 Usage Notes

Module Stop: When the MSTP14 bit in MSTPCR is set to 1, the DTC clock stops, and enters the module stop state. However, 1 cannot be written in the MSTP14 bit while th operating.

On-Chip RAM: The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

DTCE Bit Setting: For DTCE bit setting, read/write operations must be performed usi manipulation instructions such as BSET and BCLR. For the initial setting only, howeve multiple activation sources are set at one time, it is possible to disable interrupts and wr executing a dummy read on the relevant register.

Each port includes a data direction register (DDR) that controls input/output (not prov
input-only port), a data register (DR) that stores output data, and a port register (PORT
read the pin states.

Ports A to E have a built-in MOS input pull-up function, and in addition to DR and DD
MOS input pull-up control register (PCR) to control the on/off state of MOS input pul

Ports 3 and A include an open-drain control register (ODR) that controls the on/off sta
output buffer PMOS.

Ports 1, and A to F can drive a single TTL load and 90 pF capacitive load, and ports 2
can drive a single TTL load and 30 pF capacitive load. All the I/O ports can drive a D
transistor when in output mode. Ports 1, and A to C can drive an LED (10 mA sink cu

Port 2, and interrupt input pins ($\overline{\text{IRQ0}}$ to $\overline{\text{IRQ7}}$) are Schmitt-triggered inputs.

For block diagrams of the ports see appendix C, I/O Port Block Diagrams.

		$P1_4$/TIOCA1		
		$P1_3$/TIOCD0/ TCLKB/A_{23} $P1_2$/TIOCC0/ TCLKA/A_{22} $P1_1$/TIOCB0/ A_{21} $P1_0$/TIOCA0/ A_{20}		When DDR = 0: input port als functioning as TPU I/O pins (TCLKA, TCLKB, TIOCA0, TIOCB0, TIOCC0, TIOCD0) When DDR = 1: address outp
Port 2	• 8-bit I/O port • Schmitt-triggered input	$P2_7$/TIOCB5/ TMO1 $P2_6$/TIOCA5/ TMO0 $P2_5$/TIOCB4/ TMCI1 $P2_4$/TIOCA4/ TMRI1 $P2_3$/TIOCD3/ TMCI0 $P2_2$/TIOCC3/ TMRI0 $P2_1$/TIOCB3 $P2_0$/TIOCA3	8-bit I/O port also functioning as TPU I/O pins (TIOCA3, TIOCB TIOCD3, TIOCA4, TIOCB4, TIOCA5, TIOCB5), and 8-bit timer and 1) I/O pins (TMRI0, TMCI0, TMO0, TMRI1, TMCI1, TMO1)	
Port 3	• 6-bit I/O port • Open-drain output capability • Schmitt-triggered input ($\overline{IRQ5}$, $\overline{IRQ4}$)	$P3_5$/SCK1/ $\overline{IRQ5}$ $P3_4$/SCK0/ $\overline{IRQ4}$ $P3_3$/RxD1 $P3_2$/RxD0 $P3_1$/TxD1 $P3_0$/TxD0	6-bit I/O port also functioning as SCI (channels 0 and 1) I/O pin RxD0, SCK0, TxD1, RxD1, SCK1) and interrupt input pins ($\overline{IRQ}$	

		$P4_2$/AN2 $P4_1$/AN1 $P4_0$/AN0					
Port A	• 4-bit I/O port • Built-in MOS input pull-up • Open-drain output capability	PA_3/A_{19} to PA_0/A_{16}	I/O ports			Address output	When DDR = (after reset): input ports When DDR = addres output
Port B	• 8-bit I/O port • Built-in MOS input pull-up	PB_7/A_{15} to PB_0/A_8	Address output	When DDR = 0 (after reset): input port When DDR = 1: address output	I/O port	Address output	When DDR = (after reset): input p When DDR = addres output
Port C	• 8-bit I/O port • Built-in MOS input pull-up	PC_7/A_7 to PC_0/A_0	Address output	When DDR = 0 (after reset): input port When DDR = 1: address output	I/O port	Address output	When DDR = (after reset): input p When DDR = addres output
Port D	• 8-bit I/O port • Built-in MOS input pull-up	PD_7/D_{15} to PD_0/D_8	Data bus input/ output		I/O port	Data bus input/output	

	• Schmitt-triggered input ($\overline{IRQ3}$ to $\overline{IRQ0}$)		When DDR = 1 (after reset): ϕ output	(after reset): input port When DDR = 1: ϕ output	ϕ output
		$PF_6/\overline{AS}$ $PF_5/\overline{RD}$ $PF_4/\overline{HWR}$	$\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$ output	I/O port	$\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$ output
		$PF_3/\overline{LWR}/\overline{IRQ3}$		I/O port also functioning as interrupt input pins ($\overline{IRQ3}$ to $\overline{IRQ0}$)	
		$PF_2/\overline{WAIT}/\overline{IRQ2}$	When WAITE = 0 (after reset): I/O port also functioning as interrupt input pin ($\overline{IRQ2}$)		When WAITE = 0 (after reset): I/O port also functioning as interrupt input pin ($\overline{IRQ2}$)
			When WAITE = 1: $\overline{WAIT}$ input also functioning as interrupt input pin ($\overline{IRQ2}$)		When WAITE = 1: $\overline{WAIT}$ input also functioning as interrupt input pin ($\overline{IRQ2}$)
		$PF_1/\overline{BACK}/\overline{IRQ1}$ $PF_0/\overline{BREQ}/\overline{IRQ0}$	When BRLE = 0 (after reset): I/O port also functioning as interrupt input pins ($\overline{IRQ1}$, $\overline{IRQ0}$)		When BRLE = 0 (after reset): I/O port also functioning as interrupt input pins ($\overline{IRQ1}$, $\overline{IRQ0}$)
			When BRLE = 1: $\overline{BREQ}$ input, $\overline{BACK}$ output also functioning as interrupt input pins ($\overline{IRQ1}$, $\overline{IRQ0}$)		When BRLE = 1: $\overline{BREQ}$ input $\overline{BACK}$ output also functioning as interrupt input pins ($\overline{IRQ1}$, $\overline{IRQ0}$)

		$\overline{IRQ7}$	(IRQ7, IRQ6) and A/D converter input pin ($\overline{ADTRG}$)	converter input pin ($\overline{ADTRG}$)	When DDR = 1: CS1, CS2, $\overline{CS3}$ output also functioning interrupt input pin ($\overline{IRQ7}$)
		$PG_0/\overline{IRQ6}/\overline{ADTRG}$			I/O port also functioning as interrupt input pin ($\overline{IRQ6}$) and A/D converter input pin ($\overline{ADTRG}$)

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version.
2. Modes 2, 3, 6, and 7 are not available on the ROMless version.
3. After a reset in mode 2 or 6.
4. After a reset in mode 1, 4 or 5.

Figure 8.1 shows the port 1 pin configuration.

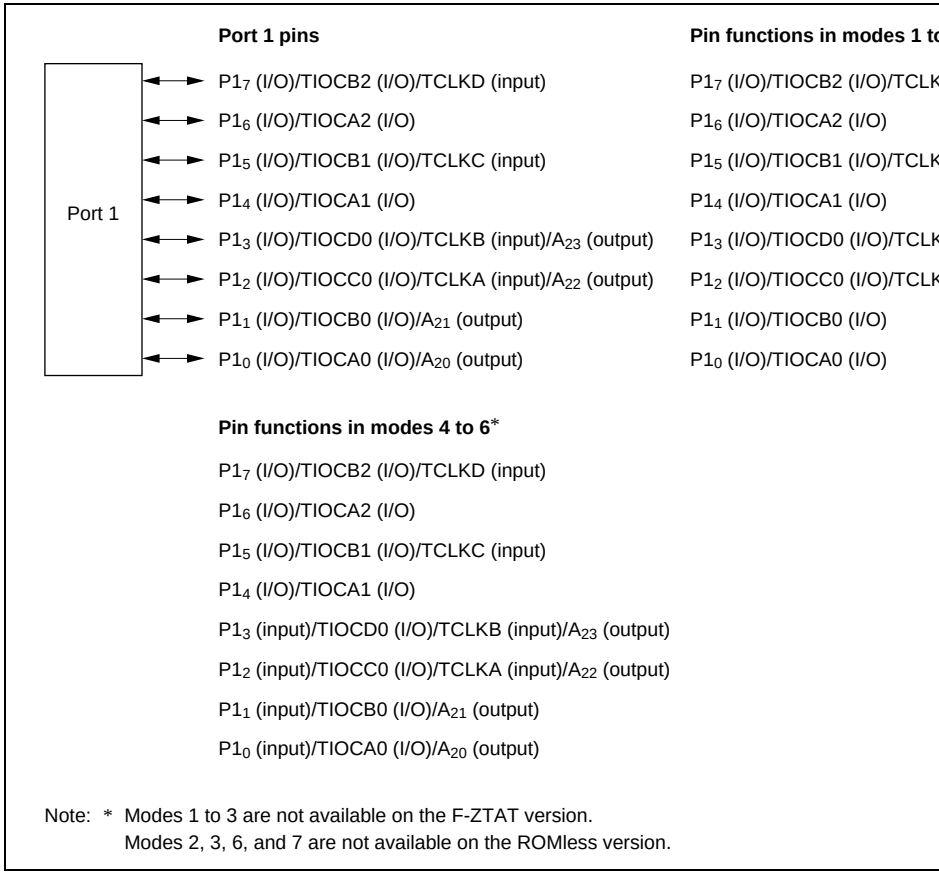


Figure 8.1 Port 1 Pin Functions

Port 1 register	PORT1	R	Undefined	H'

Note: * Lower 16 bits of the address.

Port 1 Data Direction Register (P1DDR)

Bit	:	7	6	5	4	3	2	1
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register, the individual bits of which specify input or ou pins of port 1. P1DDR cannot be read; if it is, an undefined value will be read.

Setting a P1DDR bit to 1 makes the corresponding port 1 pin an output pin, while clea to 0 makes the pin an input pin.

P1DDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It re prior state after a manual reset, and in software standby mode. As the TPU is initialize manual reset, the pin states are determined by the P1DDR and P1DR specifications.

Whether the address output pins maintain their output state or go to the high-impedanc transition to software standby mode is selected by the OPE bit in SBYCR.

- Modes 1 to 3 and 7*

  The corresponding port 1 pins are output ports when P1DDR is set to 1, and input cleared to 0.
- Modes 4 to 6*

  The corresponding port 1 pins are address outputs when P13DDR to P10DDR are input ports when cleared to 0.

  The corresponding port 1 pins are output ports when P17DDR to P14DDR are set input ports when cleared to 0.

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 (P

P1DR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retai
state after a manual reset, and in software standby mode.

Port 1 Register (PORT1)

Bit	:	7	6	5	4	3	2	1
		P17	P16	P15	P14	P13	P12	P11
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins $P1_7$ to $P1_0$.

PORT1 is an 8-bit read-only register that shows the pin states. It cannot be written to. V
output data for the port 1 pins ($P1_7$ to $P1_0$) 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. I
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
states, as P1DDR and P1DR are initialized. PORT1 retains its prior state after a manual
in software standby mode.

P1$_7$/TIOCB2/ TCLKD

The pin function is switched as shown below according to the com the TPU channel 2 setting by bits MD3 to MD0 in TMDR2, bits IOB TIOR2, bits CCLR1 and CCLR0 in TCR2, bits TPSC2 to TPSC0 in TCR5, and bit P17DDR.

TPU Channel 2 Setting	Table Below (1)	Table Belo	
P17DDR	—	0	
Pin function	TIOCB2 output	P1$_7$ input	
		TIOCB2 in	
	TCLKD input*^{2}		

TPU Channel 2 Setting	(2)	(1)	(2)	(2)	(1)
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 t
CCLR1, CCLR0	—	—	—	—	Other than B'1
Output function	—	Output compare output	—	—	PWM mode 2 output

Notes: 1. TIOCB2 input when TPU channel 2 is in normal operat (MD3 to MD0 = B'0000) and input capture is set (IOB3 B'1xxx).

2. TCLKD input when the setting for either TCR0 or TCR5 to TPSC0 = B'111.
TCLKD input when channels 2 and 4 are set to phase mode (MD3 to MD0 = B'01xx).

					TIOCA2 inp

TPU Channel 2 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0011	B'
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'	
CCLR1, CCLR0	—	—	—	—	Other than B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

x

Notes: 1. TIOCA2 input when TPU channel 2 is in normal operatio (MD3 to MD0 = B'0000) and input capture is set (IOA3 t B'10xx).

2. TIOCB2 output is disabled.

Pin function	TIOCB1 output			$P1_5$ input	
				TIOCB1 in	
	TCLKC input*2				

TPU Channel 1 Setting	(2)	(1)	(2)	(2)	(1)
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 t
CCLR1, CCLR0	—	—	—	—	Other than B'10
Output function	—	Output compare output	—	—	PWM mode 2 output

Notes: 1. TIOCB1 input when TPU channel 1 is in normal operat (MD3 to MD0 = B'0000) and input capture is set (IOB3 B'10xx).

2. TCLKC input when the setting for either TCR0 or TCR2 to TPSC0 = B'110; or when the setting for either TCR4 TPSC2 to TPSC0 = B'101.
TCLKC input when channels 2 and 4 are set to phase mode (MD3 to MD0 = B'01xx).

				TIOCA1 inp	

TPU Channel 1 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'(
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'	
CCLR1, CCLR0	—	—	—	—	Other than B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

x

Notes: 1. TIOCA1 input when TPU channel 1 is in normal operatio (MD3 to MD0 = B'0000) and input capture is set (IOA3 t B'10xx).

2. TIOCB1 output is disabled.

P13DDR	—	0	1	0	1	
Pin function	TIOCD0 output	$P1_3$ input	$P1_3$ output	TIOCD0 output	A_{23} output	P in
		TIOCD0 input*2				
	TCLKB input*3					

TPU Channel 0 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000		B'0010	B'001	
IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other
CCLR2 to CCLR0	—	—	—	—	Other than B'110
Output function	—	Output compare output	—	—	PWM mode outpu

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version. Modes 2, 3, 6, and 7 are not available on the ROMless

2. TIOCD0 input when TPU channel 0 is in normal operat (MD3 to MD0 = B'0000) and input capture is set (IOD3 B'10xx).

3. TCLKB input when the TCR0, TCR1, or TCR2 setting i TPSC0 = B'101.

   TCLKB input when channels 1 and 5 are set to phase c mode (MD3 to MD0 = B'01xx).

P12DDR	—	0	1	0	1	0
Pin function	TIOCC0 output	$P1_2$ input	$P1_2$ output	TIOCC0 output	A_{22} output	P1 inp
		TIOCC0 input*2				
	TCLKA input*3					

TPU Channel 0 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000		B'001x	B'0010	B'
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B	
CCLR2 to CCLR0	—	—	—	—	Other than B'101
Output function	—	Output compare output	—	PWM mode 1 output*4	PWM mode 2 output

x

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless v
2. TIOCC0 input when TPU channel 0 is in normal operatio
(MD3 to MD0 = B'0000) and input capture is set (IOC3 t
B'10xx).
3. TCLKA input when the TCR0 to TCR5 setting is: TPSC2
= B'100.
TCLKA input when channel 1 and 5 are set to phase cou
mode (MD3 to MD0 = B'01xx).
4. TIOCD0 output is disabled.
When BFA = 1 or BFB = 1 in TMDR0, output is disabled
setting (2) applies.

P11DDR	—	0	1	0	1	
Pin function	TIOCB0 output	$P1_1$ input	$P1_1$ output	TIOCB0 output	A_{21} output	I/O
		TIOCB0 input*2				

TPU Channel 0 Setting	(2)	(1)	(2)	(2)	(1)
MD3 to MD0	B'0000		B'0010		B'001
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other
CCLR2 to CCLR0	—	—	—	—	Othe than B'01
Output function	—	Output compare output	—	—	PWM mode outpu

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version. Modes 2, 3, 6, and 7 are not available on the ROMless
2. TIOCB0 input when TPU channel 0 is in normal operat (MD3 to MD0 = B'0000) and input capture is set (IOB3 B'10xx).

P10DDR	—	0	1	0	1	0
Pin function	TIOCA0 output	$P1_0$ input	$P1_0$ output	TIOCA0 output	A_{20} output	P inp
		TIOCA0 input*2				

TPU Channel 0 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000		B'001x	B'0010	B
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B	
CCLR2 to CCLR0	—	—	—	—	Other than B'001
Output function	—	Output compare output	—	PWM mode 1 output*3	PWM mode 2 output

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version. Modes 2, 3, 6, and 7 are not available on the ROMless v
2. TIOCA0 input when TPU channel 0 is in normal operatio (MD3 to MD0 = B'0000) and input capture is set (IOA3 t B'10xx).
3. TIOCB0 output is disabled.

Figure 8.2 shows the port 2 pin configuration.

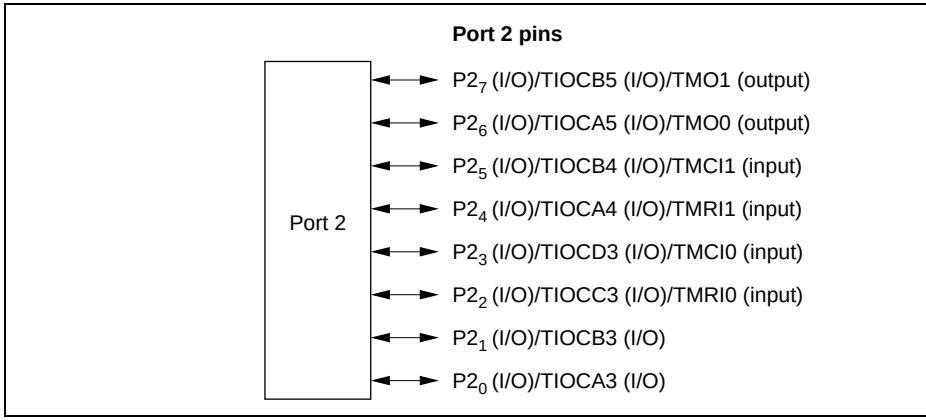


Figure 8.2 Port 2 Pin Functions

Port 2 data register	P2DR	R/W	H'00	H'
Port 2 register	PORT2	R	Undefined	H'

Note: * Lower 16 bits of the address.

Port 2 Data Direction Register (P2DDR)

Bit	:	7	6	5	4	3	2	1
		P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

P2DDR is an 8-bit write-only register, the individual bits of which specify input or outp pins of port 2. P2DDR cannot be read; if it is, an undefined value will be read.

Setting a P2DDR bit to 1 makes the corresponding port 2 pin an output pin, while clear to 0 makes the pin an input pin.

P2DDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It ret prior state after a manual reset, and in software standby mode. As the TPU and 8-bit tir initialized by a manual reset, the pin states are determined by the P2DDR and P2DR specifications.

P2DR is initialized to H'00 by a power-on reset, and in hardware standby mode. It reta
state after a manual reset, and in software standby mode.

Port 2 Register (PORT2)

Bit	:	7	6	5	4	3	2	1
		P27	P26	P25	P24	P23	P22	P21
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins $P2_7$ to $P2_0$.

PORT2 is an 8-bit read-only register that shows the pin states. It cannot be written to.
output data for the port 2 pins ($P2_7$ to $P2_0$) must always be performed on P2DR.

If a port 2 read is performed while P2DDR bits are set to 1, the P2DR values are read.
read is performed while P2DDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORT2 contents are determine
states, as P2DDR and P2DR are initialized. PORT2 retains its prior state after a manu
in software standby mode.

P2$_7$/TIOCB5/ TMO1

The pin function is switched as shown below according to the comb the TPU channel 5 setting by bits MD3 to MD0 in TMDR5, bits IOB3 TIOR5, bits CCLR1 and CCLR0 in TCR5, bits OS3 to OS0 in TCSR P27DDR.

OS3 to OS0	All 0			
TPU Channel 5 Setting	Table Below (1)	Table Below (2)		
P27DDR	—	0	1	
Pin function	TIOCB5 output	P2$_7$ input	P2$_7$ output	TM
		TIOCB5 input*		

TPU Channel 5 Setting	(2)	(1)	(2)	(2)	(1)
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 th
CCLR1, CCLR0	—	—	—	—	Other than B'10
Output function	—	Output compare output	—	—	PWM mode 2 output

x

Note: * TIOCB5 input when TPU channel 5 is in normal operatio (MD3 to MD0 = B'0000) and input capture is set (IOB3 t B'1xxx).

P26DDR	—	0	1	
NDER6	—	—	0	
Pin function	TIOCA5 output	$P2_6$ input	$P2_6$ output	T
		TIOCA5 input*1		

TPU Channel 5 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B	
CCLR1, CCLR0	—	—	—	—	Other than B'0
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

Notes: 1. TIOCA5 input when TPU channel 5 is in normal operat
(MD3 to MD0 = B'0000) and input capture is set (IOA3
B'1xxx).
2. TIOCB5 output is disabled.

Pin function	TIOCB4 output	$P2_5$ input	P
		TIOCB4 inp	
	TMCI1 input		

TPU Channel 4 Setting	(2)	(1)	(2)	(2)	(1)
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 th
CCLR1, CCLR0	—	—	—	—	Other than B'10
Output function	—	Output compare output	—	—	PWM mode 2 output

x

Note: * TIOCB4 input when TPU channel 4 is in normal operatio (MD3 to MD0 = B'0000) and input capture is set (IOB3 t B'10xx).

Pin function	TIOCA4 output	$P2_4$ input	
		TIOCA4 in	
	TMRI1 input		

TPU Channel 4 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B	
CCLR1, CCLR0	—	—	—	—	Other than B'0
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode output

Notes: 1. TIOCA4 input when TPU channel 4 is in normal operat (MD3 to MD0 = B'0000) and input capture is set (IOA3 B'10xx).

2. TIOCB4 output is disabled.

Pin function	TIOCD3 output	$P2_3$ input	P
		TIOCD3 inp	
	TMCI0 input		

TPU Channel 3 Setting	(2)	(1)	(2)	(2)	(1)
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 th
CCLR2 to CCLR0	—	—	—	—	Other than B'110
Output function	—	Output compare output	—	—	PWM mode 2 output

x

Note: * TIOCD3 input when TPU channel 3 is in normal operatio (MD3 to MD0 = B'0000) and input capture is set (IOD3 t B'10xx).

Pin function	TIOCC3 output		$P2_2$ input		
			TIOCC3 in		
	TMRI0 input				

TPU Channel 3 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000		B'001x	B'0010	B
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B	
CCLR2 to CCLR0	—	—	—	—	Other than B'101
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode output

Notes: 1. TIOCC3 input when TPU channel 3 is in normal operat (MD3 to MD0 = B'0000) and input capture is set (IOC3 B'10xx).

2. TIOCD3 output is disabled.
When BFA = 1 or BFB = 1 in TMDR3, output is disable setting (2) applies.

					TIOCB3 inp

TPU Channel 3 Setting	(2)	(1)	(2)	(2)	(1)
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 th
CCLR2 to CCLR0	—	—	—	—	Other than B'010
Output function	—	Output compare output	—	—	PWM mode 2 output

x

Note: * TIOCB3 input when TPU channel 3 is in normal operatio (MD3 to MD0 = B'0000) and input capture is set (IOB3 t B'10xx).

		TIOCA3 in

TPU Channel 3 Setting	(2)	(1)	(2)	(1)	(1)
MD3 to MD0	B'0000		B'001x	B'0010	B
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B	
CCLR2 to CCLR0	—	—	—	—	Other than B'001
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output

Notes: 1. TIOCA3 input when TPU channel 3 is in normal operat (MD3 to MD0 = B'0000) and input capture is set (IOA3 B'10xx).

2. TIOCB3 output is disabled.

Figure 8.3 shows the port 3 pin configuration.

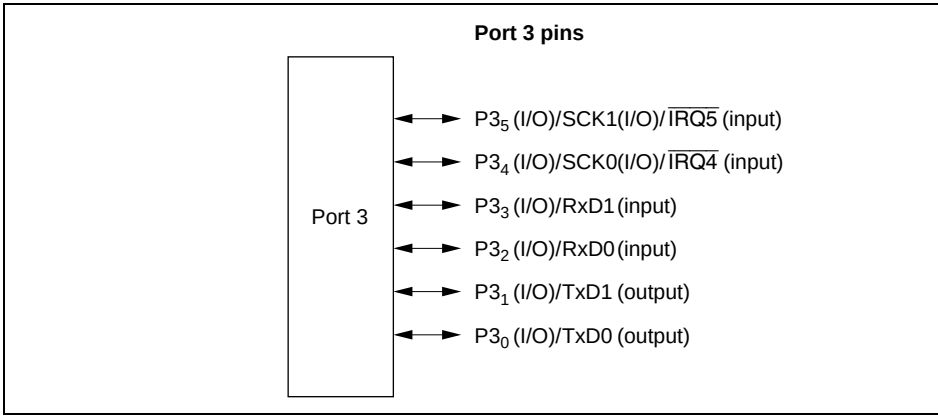


Figure 8.3 Port 3 Pin Functions

8.4.2 Register Configuration

Table 8.6 shows the port 3 register configuration.

Table 8.6 Port 3 Registers

Name	Abbreviation	R/W	Initial Value*1	Ad
Port 3 data direction register	P3DDR	W	H'00	H'
Port 3 data register	P3DR	R/W	H'00	H'
Port 3 register	PORT3	R	Undefined	H'
Port 3 open drain control register	P3ODR	R/W	H'00	H'

Notes: 1. Value of bits 5 to 0.
2. Lower 16 bits of the address.

pins of port 3. Bits 7 and 6 are reserved. P3DDR cannot be read; if it is, an undefined
read.

Setting a P3DDR bit to 1 makes the corresponding port 3 pin an output pin, while clea
to 0 makes the pin an input pin.

P3DDR is initialized to H'00 (bits 5 to 0) by a power-on reset, and in hardware standb
retains its prior state after a manual reset, and in software standby mode. As the SCI is
the pin states are determined by the P3DDR and P3DR specifications.

Port 3 Data Register (P3DR)

Bit	:	7	6	5	4	3	2	1
		—	—	P35DR	P34DR	P33DR	P32DR	P31DR
Initial value	:	Undefined	Undefined	0	0	0	0	0
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 (

Bits 7 and 6 are reserved; they return an undetermined value if read, and cannot be mo

P3DR is initialized to H'00 (bits 5 to 0) by a power-on reset, and in hardware standby
retains its prior state after a manual reset, and in software standby mode.

PORT3 is an 8-bit read-only register that shows the pin states. Writing of output data fo pins ($P3_5$ to $P3_0$) must always be performed on P3DR.

Bits 7 and 6 are reserved; they return an undetermined value if read, and cannot be mod

If a port 3 read is performed while P3DDR bits are set to 1, the P3DR values are read. I 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 states, as P3DDR and P3DR are initialized. PORT3 retains its prior state after a manual in software standby mode.

Port 3 Open Drain Control Register (P3ODR)

Bit	:	7	6	5	4	3	2	1
		—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR
Initial value	:	Undefined	Undefined	0	0	0	0	0
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 e pin ($P3_5$ to $P3_0$).

Bits 7 and 6 are reserved; they return an undetermined value if read, and cannot be mod

Setting a P3ODR bit to 1 makes the corresponding port 3 pin an NMOS open-drain out while clearing the bit to 0 makes the pin a CMOS output pin.

P3ODR is initialized to H'00 (bits 5 to 0) by a power-on reset, and in hardware standby retains its prior state after a manual reset, and in software standby mode.

bit C/$\overline{A}$ in the SCI1 SMR, bits CKE0 and CKE1 in SCR, and bit P35

CKE1	0			
C/$\overline{A}$	0			1
CKE0	0		1	—
P35DDR	0	1	—	—
Pin function	$P3_5$ input pin	$P3_5$ output pin*1	SCK1 output pin*1	SCK1 output pin*1
	$\overline{IRQ5}$ interrupt input pin*2			

Notes: 1. When P35ODR = 1, the pin becomes on NMOS open-
2. When this pin is used as an external interrupt input, it s
used as an input/output pin with other functions.

$P3_4$/SCK0/$\overline{IRQ4}$ The pin function is switched as shown below according to the com
bit C/$\overline{A}$ in the SCI0 SMR, bits CKE0 and CKE1 in SCR, and bit P34

CKE1	0			
C/$\overline{A}$	0			1
CKE0	0		1	—
P34DDR	0	1	—	—
Pin function	$P3_4$ input pin	$P3_4$ output pin*1	SCK0 output pin*1	SCK0 output pin*1
	$\overline{IRQ4}$ interrupt input pin*2			

Notes: 1. When P34ODR = 1, the pin becomes an NMOS open-
2. When this pin is used as an external interrupt input, it s
used as an input/output pin with other functions.

$P3_2$/RxD0	The pin function is switched as shown below according to the comb bit RE in the SCI0 SCR, and bit P32DDR.

RE	0		
P32DDR	0	1	
Pin function	$P3_2$ input pin	$P3_2$ output pin*	RxD0

Note: * When P32ODR = 1, the pin becomes an NMOS open-dr

$P3_1$/TxD1	The pin function is switched as shown below according to the comb bit TE in the SCI1 SCR, and bit P31DDR.

TE	0		
P31DDR	0	1	
Pin function	$P3_1$ input pin	$P3_1$ output pin*	TxD1 c

Note: * When P31ODR = 1, the pin becomes an NMOS open-dr

$P3_0$/TxD0	The pin function is switched as shown below according to the comb bit TE in the SCI0 SCR, and bit P30DDR.

TE	0		
P30DDR	0	1	
Pin function	$P3_0$ input pin	$P3_0$ output pin*	TxD0 c

Note: * When P30ODR = 1, the pin becomes an NMOS open-dr

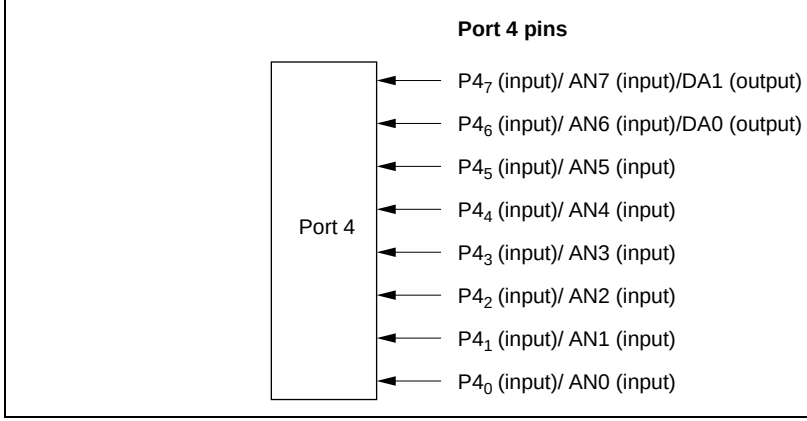


Figure 8.4 Port 4 Pin Functions

8.5.2 Register Configuration

Table 8.8 shows the port 4 register configuration. Port 4 is an input-only port, and doe
data direction register or data register.

Table 8.8 Port 4 Registers

Name	Abbreviation	R/W	Initial Value	A
Port 4 register	PORT4	R	Undefined	H

Note: * Lower 16 bits of the address.

PORT4 is an 8-bit read-only port. A read always returns the pin states. Writes are inval

8.5.3 Pin Functions

Port 4 pins also function as A/D converter analog input pins (AN0 to AN7) and D/A co
analog output pins (DA0 and DA1).

8.6 Port A

8.6.1 Overview

Port A is an 4-bit I/O port. Port A pins also function as address bus outputs. The pin fu
change according to the operating mode.

Port A has a built-in MOS input pull-up function that can be controlled by software.

Figure 8.5 shows the port A pin configuration.

Pin functions in modes 4 and 5	Pin functions in mode 6*
A_{19} (output)	PA_3 (input)/A_{19} (output)
A_{18} (output)	PA_2 (input)/A_{18} (output)
A_{17} (output)	PA_1 (input)/A_{17} (output)
A_{16} (output)	PA_0 (input)/A_{16} (output)

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.5 Port A Pin Functions

8.6.2 Register Configuration

Table 8.9 shows the port A register configuration.

Table 8.9 Port A Registers

Name	Abbreviation	R/W	Initial Value*1	A
Port A data direction register	PADDR	W	H'0	H
Port A data register	PADR	R/W	H'0	H
Port A register	PORTA	R	Undefined	H
Port A MOS pull-up control register	PAPCR	R/W	H'0	H
Port A open-drain control register	PAODR	R/W	H'0	H

Notes: 1. Value of bits 3 to 0.
2. Lower 16 bits of the address.

pins of port A. PADDR cannot be read; if it is, an undefined value will be read. Bits 7
reserved.

PADDR is initialized to H'0 (bits 3 to 0) by a power-on reset and in hardware standby r
retains its prior state after a manual reset, and in software standby mode. The OPE bit i
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.

- Modes 1, 2, 3, and 7*

  Setting a PADDR bit to 1 makes the corresponding port A pin an output port, while
  the bit to 0 makes the pin an input port.
- Modes 4 and 5

  The corresponding port A pins are address outputs irrespective of the value of bits P
  PA0DDR.
- Mode 6*

  Setting a PADDR bit to 1 makes the corresponding port A pin an address output wh
  the bit to 0 makes the pin an input port.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be mod

PADR is initialized to H'0 (bits 3 to 0) by a power-on reset, and in hardware standby r
retains its prior state after a manual reset, and in software standby mode.

Port A Register (PORTA)

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	PA3	PA2	PA1
Initial value	:	Undefined	Undefined	Undefined	Undefined	—*	—*	—*
R/W	:	—	—	—	—	R	R	R

Note: * Determined by state of pins PA_3 to PA_0.

PORTA is an 8-bit read-only register that shows the pin states. It cannot be written to.
output data for the port A pins (PA_3 to PA_0) must always be performed on PADR.

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be mod

If a port A read is performed while PADDR bits are set to 1, the PADR values are rea
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 determine
states, as PADDR and PADR are initialized. PORTA retains its prior state after a man
and in software standby mode.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

incorporated into port A on an individual bit basis.

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modif

Bits 3 to 0 are valid in modes 1, 2, 3, 6, and 7, and all the bits are invalid in modes 4 an a PADDR bit is cleared to 0 (input port setting), setting the corresponding PAPCR bit t 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 r retains its prior state after a manual reset, and in software standby mode.

Port A Open Drain Control Register (PAODR)

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	PA3ODR	PA2ODR	PA1ODR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W

PAODR is an 8-bit readable/writable register that controls whether PMOS is on or off f port A pin (PA_3 to PA_0).

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modif

All bits are valid in modes 1, 2, 3, and 7.*

Setting a PAODR bit to 1 makes the corresponding port A pin an NMOS open-drain ou clearing the bit to 0 makes the pin a CMOS output.

PAODR is initialized to H'0 (bits 3 to 0) by a power-on reset, and in hardware standby retains its prior state after a manual reset, and in software standby mode.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

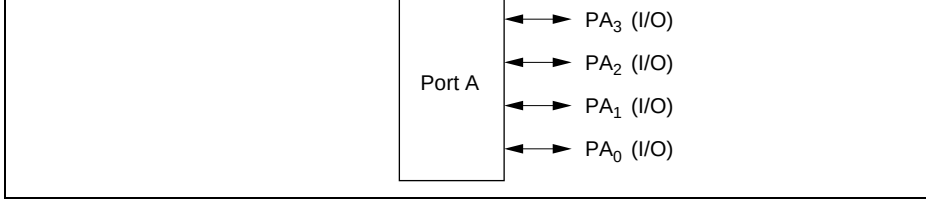


Figure 8.6 Port A Pin Functions (Modes 1, 2, 3, and 7)*

Modes 4 and 5: In modes 4 and 5, the lower 4 bits of port A are designated as address automatically.

Port A pin functions in modes 4 and 5 are shown in figure 8.7.

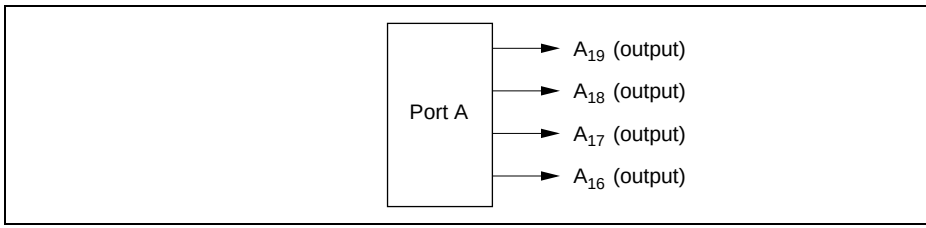


Figure 8.7 Port A Pin Functions (Modes 4 and 5)

Mode 6*: In mode 6*, port A pins function as address outputs or input ports. Input or be specified on an individual bit basis. Setting a PADDR bit to 1 makes the correspon pin an address output, while clearing the bit to 0 makes the pin an input port.

Port A pin functions in mode 6 are shown in figure 8.8.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.8 Port A Pin Functions (Mode 6)

8.6.4 MOS Input Pull-Up Function

Port A has a built-in MOS input pull-up function that can be controlled by software. Th
input pull-up function can be used in modes 1, 2, 3, 6, and 7*, and cannot be used in mo
5. MOS input pull-up can be specified as on or off on an individual bit basis.

When a PADDR bit is cleared to 0, setting the corresponding PAPCR bit to 1 turns on t
input pull-up for that pin.

The MOS input pull-up function is in the off state after a power-on reset, and in hardwa
mode. The prior state is retained after a manual reset, and in software standby mode.

Table 8.10 summarizes the MOS input pull-up states.

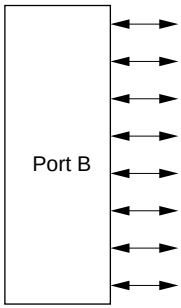
Table 8.10 MOS Input Pull-Up States (Port A)

Modes		Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In O
1 to 3, 6, 7*	PA_3 to PA_0	OFF	OFF	ON/OFF	ON/OFF	O
4, 5	PA_3 to PA_0			OFF	OFF	O

Legend:
OFF: MOS input pull-up is always off.
ON/OFF: On when PADDR = 0 and PAPCR = 1; otherwise off.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.9 shows the port B pin configuration.



Port B pins	Pin functions in modes 1, 4, and
PB_7/A_{15}	A_{15} (output)
PB_6/A_{14}	A_{14} (output)
PB_5/A_{13}	A_{13} (output)
PB_4/A_{12}	A_{12} (output)
PB_3/A_{11}	A_{11} (output)
PB_2/A_{10}	A_{10} (output)
PB_1/A_9	A_9 (output)
PB_0/A_8	A_8 (output)

Pin functions in modes 2 and 6*	Pin functions in modes 3 and 7*
PB_7 (input)/A_{15} (output)	PB_7 (I/O)
PB_6 (input)/A_{14} (output)	PB_6 (I/O)
PB_5 (input)/A_{13} (output)	PB_5 (I/O)
PB_4 (input)/A_{12} (output)	PB_4 (I/O)
PB_3 (input)/A_{11} (output)	PB_3 (I/O)
PB_2 (input)/A_{10} (output)	PB_2 (I/O)
PB_1 (input)/A_9 (output)	PB_1 (I/O)
PB_0 (input)/A_8 (output)	PB_0 (I/O)

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.9 Port B Pin Functions

Port B register	PORTB	R	Undefined	H'
Port B MOS pull-up control register	PBPCR	R/W	H'00	H'

Note: * Lower 16 bits of the address.

Port B Data Direction Register (PBDDR)

Bit	:	7	6	5	4	3	2	1
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PBDDR is an 8-bit write-only register, the individual bits of which specify input or out 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 ret prior state after a manual reset, and in software standby mode. The OPE bit in SBYCR select whether the address output pins retain their output state or become high-impedan transition is made to software standby mode.

- Modes 1, 4, and 5*

  The corresponding port B pins are address outputs irrespective of the value of the P
- Modes 2 and 6*

  Setting a PBDDR bit to 1 makes the corresponding port B pin an address output, w clearing the bit to 0 makes the pin an input port.
- Modes 3 and 7*

  Setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while the bit to 0 makes the pin an input port.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

PBDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It ret
state after a manual reset, and in software standby mode.

Port B Register (PORTB)

Bit	:	7	6	5	4	3	2	1
		PB7	PB6	PB5	PB4	PB3	PB2	PB1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PB_7 to PB_0.

PORTB is an 8-bit read-only register that shows the pin states. It cannot be written to.
output data for the port B pins (PB_7 to PB_0) 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
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 determine
states, as PBDDR and PBDR are initialized. PORTB retains its prior state after a manu
in software standby mode.

incorporated into port B on an individual bit basis.

When a PBDDR bit is cleared to 0 (input port setting) in mode 2, 3, 6, or 7*, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for the corresponding pi

PBPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It ret prior state after a manual reset, and in software standby mode.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

8.7.3 Pin Functions

Modes 1, 4, and 5*: In modes 1, 4, and 5*, port B pins are automatically designated as outputs.

Port B pin functions in modes 1, 4, and 5 are shown in figure 8.10.

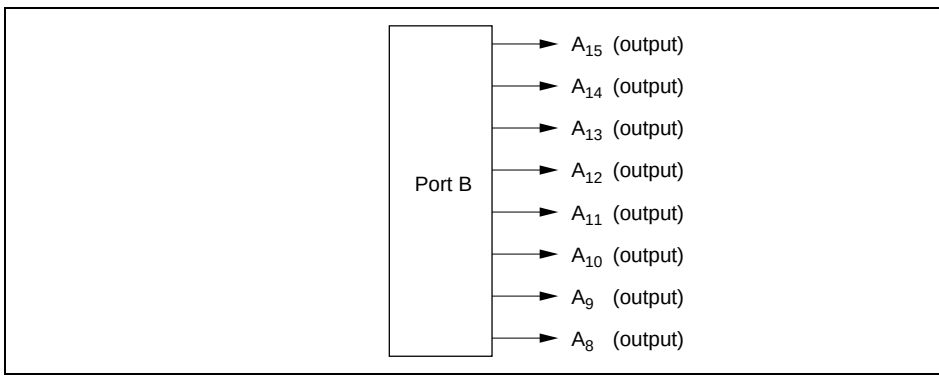


Figure 8.10 Port B Pin Functions (Modes 1, 4, and 5)*

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

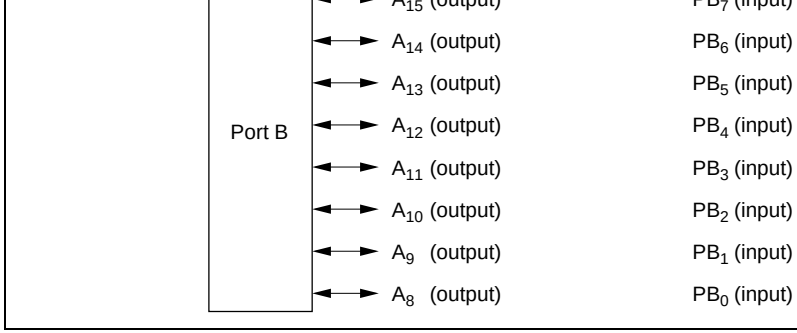


Figure 8.11 Port B Pin Functions (Modes 2 and 6)*

Modes 3 and 7*: In modes 3 and 7*, port B pins function as I/O ports. Input or outpu
specified for each pin on an individual bit basis. 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 i

Port B pin functions in modes 3 and 7 are shown in figure 8.12.

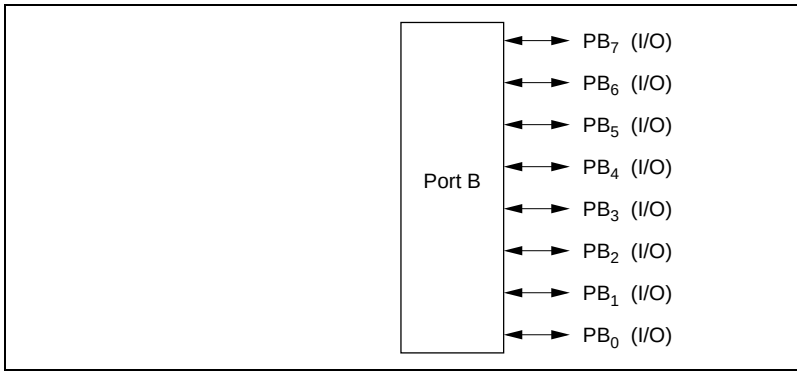


Figure 8.12 Port B Pin Functions (Modes 3 and 7)*

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

The MOS input pull-up function is in the off state after a power-on reset, and in hardwa
mode. The prior state is retained after a manual reset, and in software standby mode.

Table 8.12 summarizes the MOS input pull-up states.

Table 8.12 MOS Input Pull-Up States (Port B)

Modes	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In O
1, 4, 5*	OFF	OFF	OFF	OFF	O
2, 3, 6, 7*			ON/OFF	ON/OFF	O

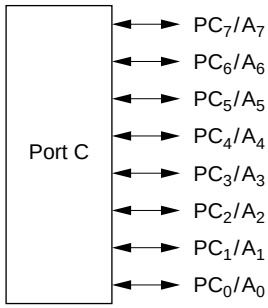
Legend:

OFF: MOS input pull-up is always off.

ON/OFF: On when PBDDR = 0 and PBPCR = 1; otherwise off.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.13 shows the port C pin configuration.



Pin functions in modes 1, 4, an

A_7 (output)	
A_6 (output)	
A_5 (output)	
A_4 (output)	
A_3 (output)	
A_2 (output)	
A_1 (output)	
A_0 (output)	

Pin functions in modes 2 and 6*	**Pin functions in modes 3 and 7**
PC_7 (input)/A_7 (output)	PC_7 (I/O)
PC_6 (input)/A_6 (output)	PC_6 (I/O)
PC_5 (input)/A_5 (output)	PC_5 (I/O)
PC_4 (input)/A_4 (output)	PC_4 (I/O)
PC_3 (input)/A_3 (output)	PC_3 (I/O)
PC_2 (input)/A_2 (output)	PC_2 (I/O)
PC_1 (input)/A_1 (output)	PC_1 (I/O)
PC_0 (input)/A_0 (output)	PC_0 (I/O)

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.13 Port C Pin Functions

Port C register	PORTC	R	Undefined	H'
Port C MOS pull-up control register	PCPCR	R/W	H'00	H'

Note: * Lower 16 bits of the address.

Port C Data Direction Register (PCDDR)

Bit	:	7	6	5	4	3	2	1
		PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PCDDR is an 8-bit write-only register, the individual bits of which specify input or out 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 ret prior state after a manual reset, and in software standby mode. The OPE bit in SBYCR select whether the address output pins retain their output state or become high-impedan transition is made to software standby mode.

- Modes 1, 4, and 5*
  The corresponding port C pins are address outputs irrespective of the value of the P
- Modes 2 and 6*
  Setting a PCDDR bit to 1 makes the corresponding port C pin an address output, w clearing the bit to 0 makes the pin an input port.
- Modes 3 and 7*
  Setting a PCDDR bit to 1 makes the corresponding port C pin an output port, while the bit to 0 makes the pin an input port.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

PCDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It ret
state after a manual reset, and in software standby mode.

Port C Register (PORTC)

Bit	:	7	6	5	4	3	2	1
		PC7	PC6	PC5	PC4	PC3	PC2	PC1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PC_7 to PC_0.

PORTC is an 8-bit read-only register that shows the pin states. It cannot be written to. output data for the port C pins (PC_7 to PC_0) 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 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 determine states, as PCDDR and PCDR are initialized. PORTC retains its prior state after a manu in software standby mode.

incorporated into port C on an individual bit basis.

When a PCDDR bit is cleared to 0 (input port setting) in mode 2, 3, 6, or 7*, setting the corresponding PCPCR bit to 1 turns on the MOS input pull-up for the corresponding pi

PCPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It ret prior state after a manual reset, and in software standby mode.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

8.8.3 Pin Functions

Modes 1, 4, and 5*: In modes 1, 4, and 5*, port C pins are automatically designated as outputs.

Port C pin functions in modes 1, 4, and 5 are shown in figure 8.14.

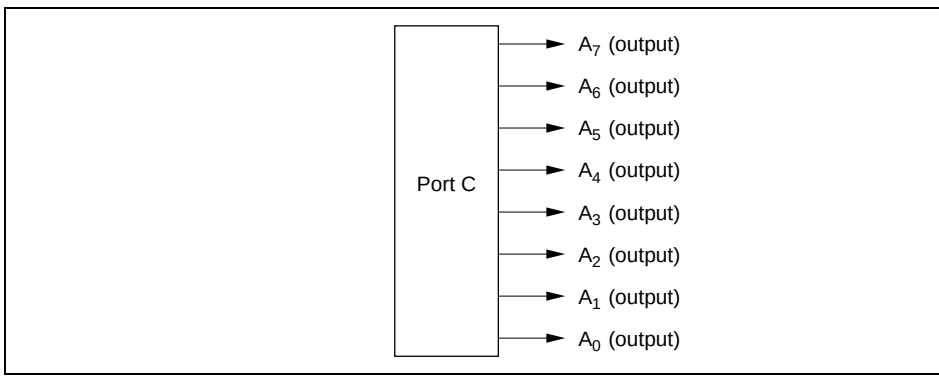


Figure 8.14 Port C Pin Functions (Modes 1, 4, and 5)*

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

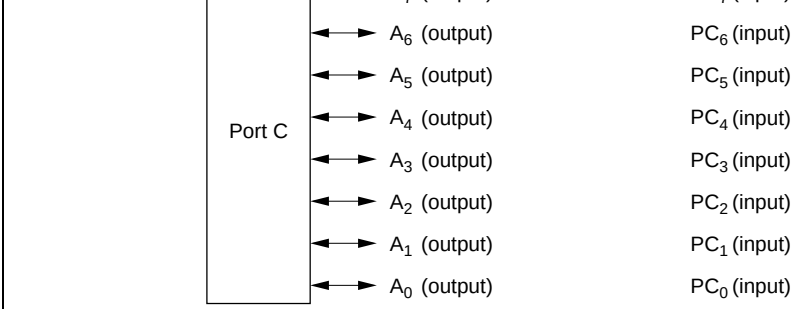


Figure 8.15 Port C Pin Functions (Modes 2 and 6)*

Modes 3 and 7*: In modes 3 and 7*, port C pins function as I/O ports. Input or outpu
specified for each pin on an individual bit basis. 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 i

Port C pin functions in modes 3 and 7 are shown in figure 8.16.

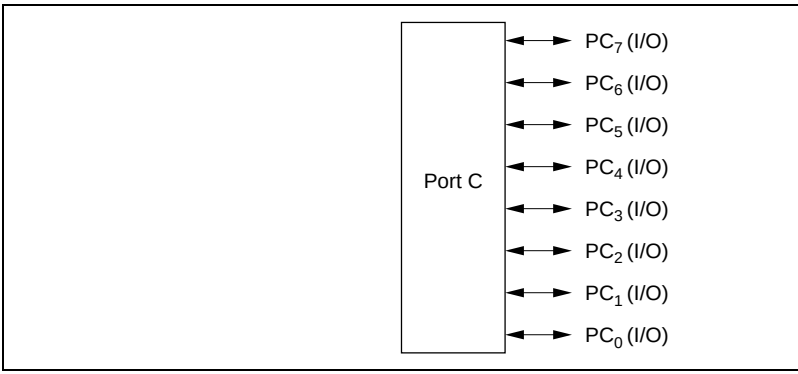


Figure 8.16 Port C Pin Functions (Modes 3 and 7)*

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

The MOS input pull-up function is in the off state after a power-on reset, and in hardwa
mode. The prior state is retained after a manual reset, and in software standby mode.

Table 8.14 summarizes the MOS input pull-up states.

Table 8.14 MOS Input Pull-Up States (Port C)

Modes	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In O
1, 4, 5*	OFF	OFF	OFF	OFF	O
2, 3, 6, 7*			ON/OFF	ON/OFF	O

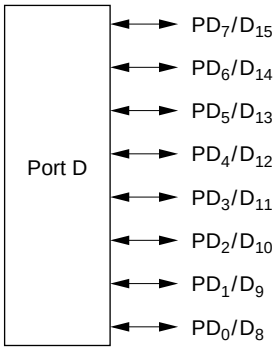
Legend:

OFF: MOS input pull-up is always off.

ON/OFF: On when PCDDR = 0 and PCPCR = 1; otherwise off.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.17 shows the port D pin configuration.



Port D pins	Pin functions in modes 1, 2, 4, 5, a
PD_7/D_{15}	D_{15} (I/O)
PD_6/D_{14}	D_{14} (I/O)
PD_5/D_{13}	D_{13} (I/O)
PD_4/D_{12}	D_{12} (I/O)
PD_3/D_{11}	D_{11} (I/O)
PD_2/D_{10}	D_{10} (I/O)
PD_1/D_9	D_9 (I/O)
PD_0/D_8	D_8 (I/O)

Pin functions in modes 3 and 7*
PD_7 (I/O)
PD_6 (I/O)
PD_5 (I/O)
PD_4 (I/O)
PD_3 (I/O)
PD_2 (I/O)
PD_1 (I/O)
PD_0 (I/O)

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.17 Port D Pin Functions

Port D register	PORTD	R	Undefined	H'
Port D MOS pull-up control register	PDPCR	R/W	H'00	H'

Note: * Lower 16 bits of the address.

Port D Data Direction Register (PDDDR)

Bit	:	7	6	5	4	3	2	1	
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	
Initial value	:	0	0	0	0	0	0	0	
R/W	:	W	W	W	W	W	W	W	

PDDDR is an 8-bit write-only register, the individual bits of which specify input or out 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 re prior state after a manual reset, and in software standby mode.

- Modes 1, 2, 4, 5, and 6*

  The input/output direction specification by PDDDR is ignored, and port D is autom designated for data I/O.

- Modes 3 and 7*

  Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, while the bit to 0 makes the pin an input port.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

PDDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It ret
state after a manual reset, and in software standby mode.

Port D Register (PORTD)

Bit	:	7	6	5	4	3	2	1
		PD7	PD6	PD5	PD4	PD3	PD2	PD1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PD_7 to PD_0.

PORTD is an 8-bit read-only register that shows the pin states. It cannot be written to.
output data for the port D pins (PD_7 to PD_0) must always be performed on PDDR.

If a port D read is performed while PDDDR bits are set to 1, the PDDR values are rea
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 determine
states, as PDDDR and PDDR are initialized. PORTD retains its prior state after a man
and in software standby mode.

incorporated into port D on an individual bit basis.

When a PDDDR bit is cleared to 0 (input port setting) in mode 3 or 7, setting the corres
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 ret
prior state after a manual reset, and in software standby mode.

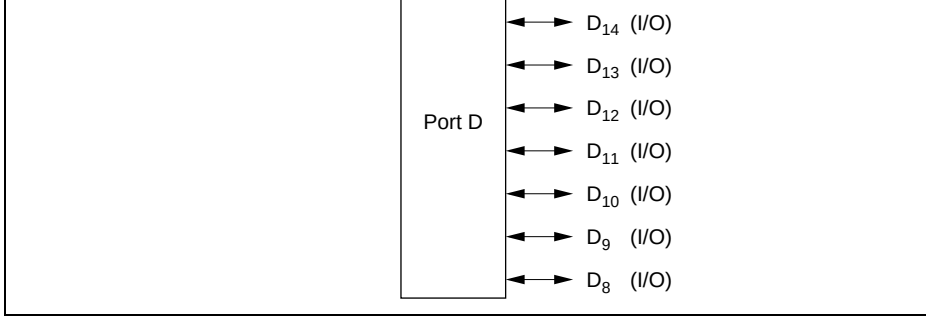


Figure 8.18 Port D Pin Functions (Modes 1, 2, 4, 5, and 6)*

Modes 3 and 7*: In modes 3 and 7*, port D pins function as I/O ports. Input or outpu
specified for each pin on an individual bit basis. 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 i

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Port D pin functions in modes 3 and 7 are shown in figure 8.19.

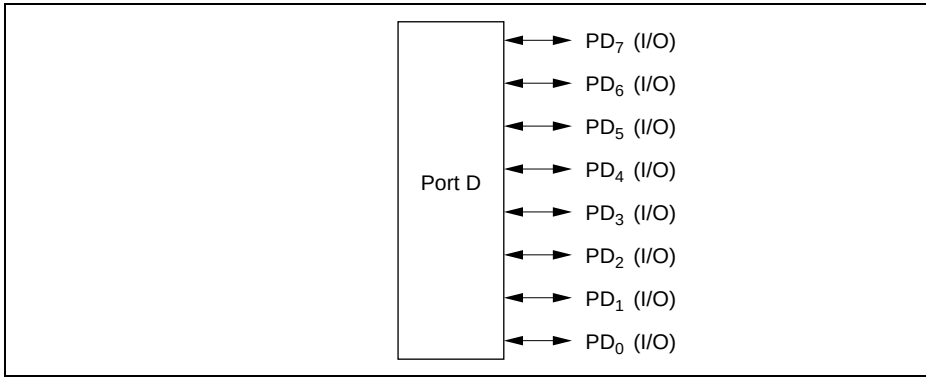


Figure 8.19 Port D Pin Functions (Modes 3 and 7)*

The MOS input pull-up function is in the off state after a power-on reset, and in hardwa
mode. The prior state is retained after a manual reset, and in software standby mode.

Table 8.16 summarizes the MOS input pull-up states.

Table 8.16 MOS Input Pull-Up States (Port D)

Modes	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In O
1, 2, 4 to 6*	OFF	OFF	OFF	OFF	O
3, 7*			ON/OFF	ON/OFF	O

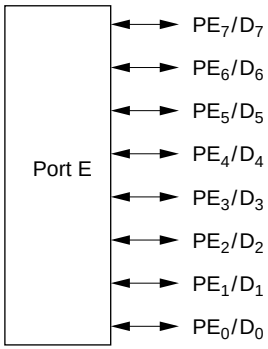
Legend:

OFF: MOS input pull-up is always off.

ON/OFF: On when PDDDR = 0 and PDPCR = 1; otherwise off.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.20 shows the port E pin configuration.



Port E pins	Pin functions in modes 1, 2, 4, 5, a
PE_7/D_7	PE_7 (I/O)/D_7 (I/O)
PE_6/D_6	PE_6 (I/O)/D_6 (I/O)
PE_5/D_5	PE_5 (I/O)/D_5 (I/O)
PE_4/D_4	PE_4 (I/O)/D_4 (I/O)
PE_3/D_3	PE_3 (I/O)/D_3 (I/O)
PE_2/D_2	PE_2 (I/O)/D_2 (I/O)
PE_1/D_1	PE_1 (I/O)/D_1 (I/O)
PE_0/D_0	PE_0 (I/O)/D_0 (I/O)

Pin functions in modes 3 and 7*
PE_7 (I/O)
PE_6 (I/O)
PE_5 (I/O)
PE_4 (I/O)
PE_3 (I/O)
PE_2 (I/O)
PE_1 (I/O)
PE_0 (I/O)

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.20 Port E Pin Functions

Port E register	PORTE	R	Undefined	H'
Port E MOS pull-up control register	PEPCR	R/W	H'00	H'

Note: * Lower 16 bits of the address.

Port E Data Direction Register (PEDDR)

Bit	:	7	6	5	4	3	2	1
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PEDDR is an 8-bit write-only register, the individual bits of which specify input or outp 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 ret prior state after a manual reset, and in software standby mode.

- Modes 1, 2, 4, 5, and 6*

  When 8-bit bus mode has been selected, port E pins function as I/O ports. Setting a to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 m pin an input port.

  When 16-bit bus mode has been selected, the input/output direction specification by ignored, and port E is designated for data I/O.

  For details of 8-bit and 16-bit bus modes, see section 6, Bus Controller.

- Modes 3 and 7*

  Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while bit to 0 makes the pin an input port.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

PEDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It reta
state after a manual reset, and in software standby mode.

Port E Register (PORTE)

Bit	:	7	6	5	4	3	2	1
		PE7	PE6	PE5	PE4	PE3	PE2	PE1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PE_7 to PE_0.

PORTE is an 8-bit read-only register that shows the pin states. It cannot be written to.
output data for the port E pins (PE_7 to PE_0) 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
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 determine
states, as PEDDR and PEDR are initialized. PORTE retains its prior state after a manu
in software standby mode.

Port E MOS Pull-Up Control Register (PEPCR)

Bit	:	7	6	5	4	3	2	1
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR
Initial value	:	0	0	0	0	0	0	0
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 func
incorporated into port E on an individual bit basis.

Modes 1, 2, 4, 5, and 6*: In modes 1, 2, 4, 5, and 6*, when 8-bit access is designated a bus mode is selected, port E pins are automatically designated as I/O ports. Setting a PE to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 make an input port.

When 16-bit bus mode is selected, the input/output direction specification by PEDDR i and port E is designated for data I/O.

Port E pin functions in modes 1, 2, 4, 5, and 6 are shown in figure 8.21.

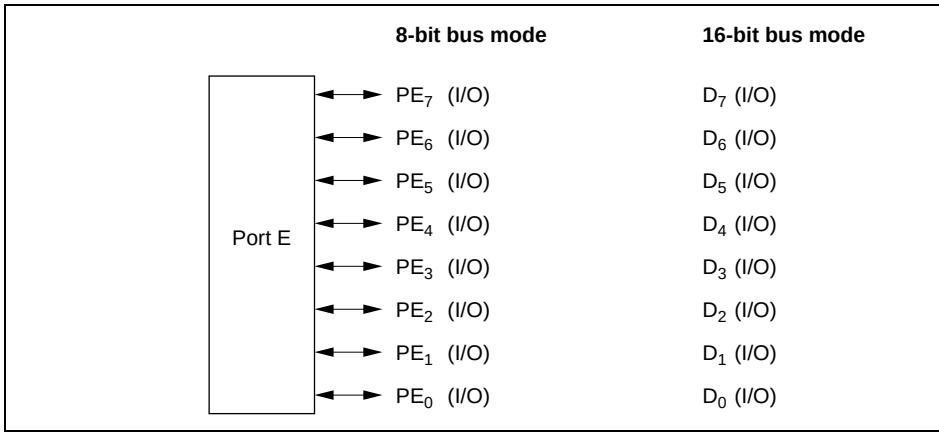


Figure 8.21 Port E Pin Functions (Modes 1, 2, 4, 5, and 6)*

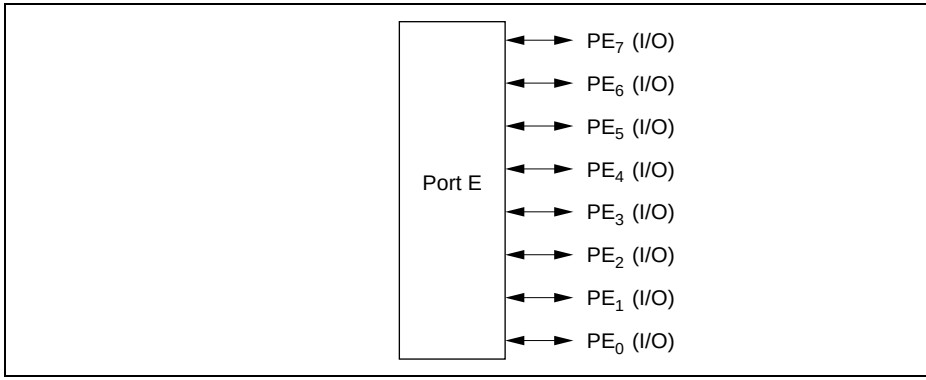


Figure 8.22 Port E Pin Functions (Modes 3 and 7)*

The MOS input pull-up function is in the off state after a power-on reset, and in hardwa
mode. The prior state is retained after a manual reset, and in software standby mode.

Table 8.18 summarizes the MOS input pull-up states.

Table 8.18 MOS Input Pull-Up States (Port E)

Modes		Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In O
3, 7*		OFF	OFF	ON/OFF	ON/OFF	O
1, 2, 4 to 6*	8-bit bus					
	16-bit bus			OFF	OFF	O

Legend:

OFF: MOS input pull-up is always off.

ON/OFF: On when PEDDR = 0 and PEPCR = 1; otherwise off.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

The interrupt input pins (IRQ0 to IRQ3) are Schmitt-triggered inputs.

Figure 8.23 shows the port F pin configuration.

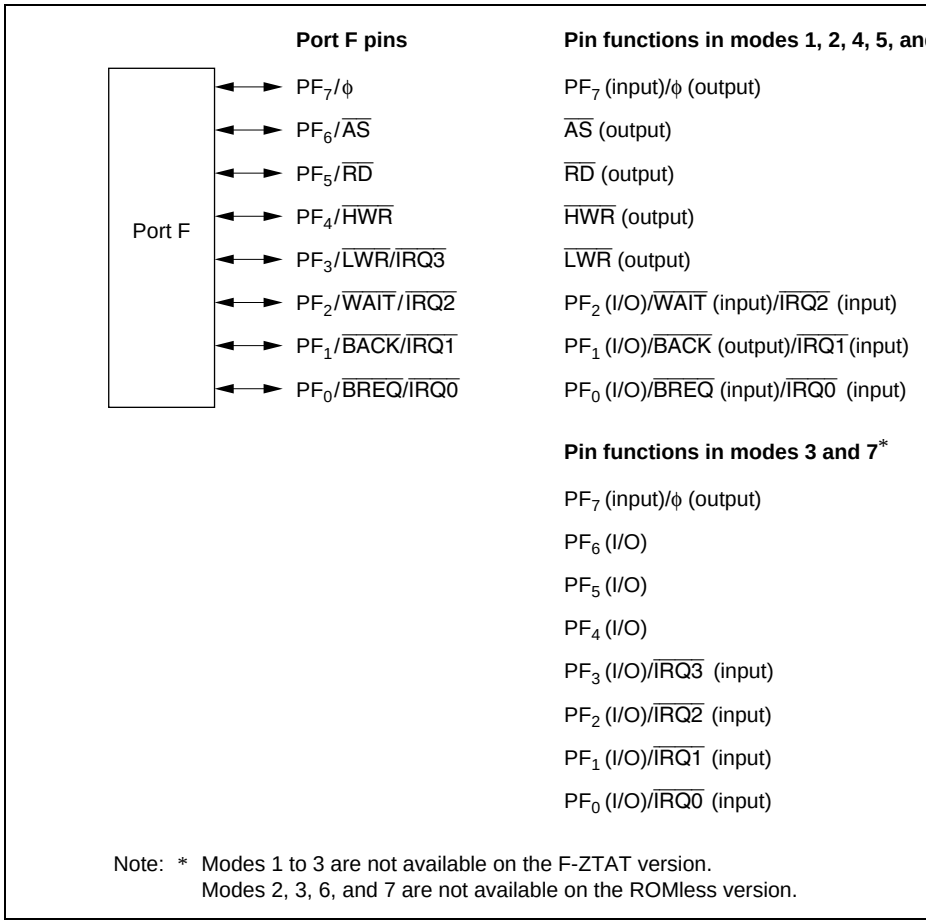


Figure 8.23 Port F Pin Functions

Port F register	PORTF	R	Undefined	H'

Notes: 1. Lower 16 bits of the address.
2. Initial value depends on the mode.

Port F Data Direction Register (PFDDR)

Bit	:	7	6	5	4	3	2	1
		PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR
Modes 1, 2, 4, 5, 6*								
Initial value	:	1	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W
Modes 3 and 7*								
Initial value	:	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W

PFDDR is an 8-bit write-only register, the individual bits of which specify input or outp
pins of port F. PFDDR cannot be read; if it is, an undefined value will be read.

PFDDR is initialized by a power-on reset, and in hardware standby mode, to H'80 in m
4, 5, and 6*, and to H'00 in modes 3 and 7*. It retains its prior state after a manual reset
software standby mode. The OPE bit in SBYCR is used to select whether the bus contr
pins retain their output state or become high-impedance when a transition is made to so
standby mode.

- Modes 1, 2, 4, 5, and 6*

  Pin PF_7 functions as the ϕ output pin when the corresponding PFDDR bit is set to 1,
  input port when the bit is cleared to 0.

  The input/output direction specified by PFDDR is ignored for pins PF_6 to PF_3, whic
  automatically designated as bus control outputs ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$).

Modes 2, 3, 6, and 7 are not available on the ROMless version.

Port F Data Register (PFDR)

Bit	:	7	6	5	4	3	2	1
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR
Initial value	:	0	0	0	0	0	0	0
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 (

PFDR is initialized to H'00 by a power-on reset, and in hardware standby mode. It reta
state after a manual reset, and in software standby mode.

Port F Register (PORTF)

Bit	:	7	6	5	4	3	2	1
		PF7	PF6	PF5	PF4	PF3	PF2	PF1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R

Note: * Determined by state of pins PF_7 to PF_0.

PORTF is an 8-bit read-only register that shows the pin states. Writing of output data
F pins (PF_7 to PF_0) 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
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 determine
states, as PFDDR and PFDR are initialized. PORTF retains its prior state after a manu
in software standby mode.

Pin	Selection Method and Pin Functions

PF_7/ϕ

The pin function is switched as shown below according to bit PF7DI

PF7DDR	0	1
Pin function	PF_7 input pin	ϕ output p

$PF_6/\overline{AS}$

The pin function is switched as shown below according to the opera and bit PF6DDR.

Operating Mode	Modes 1, 2, 4, 5, 6*	Modes 3 and 7*	
PF6DDR	—	0	
Pin function	$\overline{AS}$ output pin	PF_6 input pin	PF_6 ou

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless v

$PF_5/\overline{RD}$

The pin function is switched as shown below according to the opera and bit PF5DDR.

Operating Mode	Modes 1, 2, 4, 5, 6*	Modes 3 and 7*	
PF5DDR	—	0	
Pin function	$\overline{RD}$ output pin	PF_5 input pin	PF_5 ou

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless v

$PF_4/\overline{HWR}$

The pin function is switched as shown below according to the opera and bit PF4DDR.

Operating Mode	Modes 1, 2, 4, 5, 6*	Modes 3 and 7*	
PF4DDR	—	0	
Pin function	$\overline{HWR}$ output pin	PF_4 input pin	PF_4 ou

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless v

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version. Modes 2, 3, 6, and 7 are not available on the ROMless
2. When this pin is used as an external interrupt input, the should be set as a port (PF_3) input pin.

$PF_2/\overline{WAIT}/\overline{IRQ2}$ The pin function is switched as shown below according to the oper and WAITE bit in BCRL, and PF2DDR bit.

Operating Mode	Modes 1, 2, 4, 5, 6*1			Modes 3
WAITE	0		1	–
PF2DDR	0	1	—	0
Pin function	PF_2 input pin	PF_2 output pin	$\overline{WAIT}$ input pin	PF_2 input pin
	$\overline{IRQ2}$ interrupt input pin*2			

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version. Modes 2, 3, 6, and 7 are not available on the ROMless
2. When this pin is used as an external interrupt input, the should be set as a port (PF_2) input pin.

$PF_1/\overline{BACK}/\overline{IRQ1}$ The pin function is switched as shown below according to the oper and the BRLE bit in BCRL and PF1DDR bit.

Operating Mode	Modes 1, 2, 4, 5, 6*1			Modes 3
BRLE	0		1	–
PF1DDR	0	1	—	0
Pin function	PF_1 input pin	PF_1 output pin	$\overline{BACK}$ output pin	PF_1 input pin
	$\overline{IRQ1}$ interrupt input pin*2			

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version. Modes 2, 3, 6, and 7 are not available on the ROMless
2. When this pin is used as an external interrupt input, the should be set as a port (PF_1) input pin.

	Input pin	Output pin	Input pin	Input pin
	$\overline{IRQ0}$ interrupt input pin*2			

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless v
2. When this pin is used as an external interrupt input, the
should be set as a port (PF_0) input pin.

8.12 Port G

8.12.1 Overview

Port G is a 5-bit I/O port. Port G pins also function as bus control signal output pins ($\overline{C}$
The A/D converter input pin ($\overline{ADTRG}$), and interrupt input pins ($\overline{IRQ6}$, $\overline{IRQ7}$). The int
pins ($\overline{IRQ6}$, $\overline{IRQ7}$) are Schmitt-triggered inputs.

Figure 8.24 shows the port G pin configuration.

PG0 (I/O)/ADTRG (input)/IRQ6 (in

Pin functions in modes 3 and 7*	Pin functions in modes 4 to 6*
PG_4 (I/O)	PG_4 (input)/$\overline{CS0}$ (output)
PG_3 (I/O)	PG_3 (input)/$\overline{CS1}$ (output)
PG_2 (I/O)	PG_2 (input)/$\overline{CS2}$ (output)
PG_1 (I/O)/$\overline{IRQ7}$ (input)	PG_1 (input)/$\overline{CS3}$ (output)/$\overline{IRQ7}$ (in
PG_0 (I/O)/$\overline{ADTRG}$ (input)/$\overline{IRQ6}$ (input)	PG_0 (I/O)/$\overline{ADTRG}$ (input)/$\overline{IRQ6}$ (in

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Figure 8.24 Port G Pin Functions

8.12.2 Register Configuration

Table 8.21 shows the port G register configuration.

Table 8.21 Port G Registers

Name	Abbreviation	R/W	Initial Value*1	A
Port G data direction register	PGDDR	W	H'10/H'00*3	H
Port G data register	PGDR	R/W	H'00	H
Port G register	PORTG	R	Undefined	H

Notes: 1. Value of bits 4 to 0.
2. Lower 16 bits of the address.
3. Initial value depends on the mode.

Initial value	:	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	W	W	W	W

PGDDR is an 8-bit write-only register, the individual bits of which specify input or out pins of port G. PGDDR cannot be read, and bits 7 to 5 are reserved. If PGDDR is read, undefined value will be read.

The PGDDR is initialized by a power-on reset and in hardware standby mode, to H'10 in modes 1, 4, and 5*, and to H'00 (bits 4 to 0) in modes 2, 3, 6, and 7*. It retains its pri after a manual reset and in software standby mode. The OPE bit in SBYCR is used to s whether the bus control output pins retain their output state or become high-impedance transition is made to software standby mode.

- Modes 1 and 2*

  Pin PG_4 functions as a bus control output pin ($\overline{CS0}$) when the corresponding PGDD to 1, and as an input port when the bit is cleared to 0.

  For pins PG_3 to PG_0, setting the corresponding PGDDR bit to 1 makes the pin an ou while clearing the bit to 0 makes the pin an input port.
- Modes 3 and 7*

  Setting a PGDDR bit to 1 makes the corresponding port G pin an output port, while the bit to 0 makes the pin an input port.
- Modes 4, 5, and 6*

  Pins PG_4 to PG_1 function as bus control output pins ($\overline{CS0}$ to $\overline{CS3}$) when the corresp PGDDR bits are set to 1, and as input ports when the bits are cleared to 0.

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless version.

Bits 7 to 5 are reserved; they return an undetermined value if read, and cannot be mod

PGDR is initialized to H'00 (bits 4 to 0) by a power-on reset, and in hardware standby retains its prior state after a manual reset, and in software standby mode.

Port G Register (PORTG)

Bit	:	7	6	5	4	3	2	1
		—	—	—	PG4	PG3	PG2	PG1
Initial value	:	Undefined	Undefined	Undefined	—*	—*	—*	—*
R/W	:	—	—	—	R	R	R	R

Note: * Determined by state of pins PG_4 to PG_0.

PORTG is an 8-bit read-only register that shows the pin states. It cannot be written to. output data for the port G pins (PG_4 to PG_0) must always be performed on PGDR.

Bits 7 to 5 are reserved; they return an undetermined value if read, and cannot be mod

If a port G read is performed while PGDDR bits are set to 1, the PGDR values are rea 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 determine states, as PGDDR and PGDR are initialized. PORTG retains its prior state after a man and in software standby mode.

$PG_4/\overline{CS0}$ The pin function is switched as shown below according to the opera and bit PG4DDR.

Operating Mode	Modes 1, 2, 4, 5, 6*		Modes 3 an	
PG4DDR	0	1	0	
Pin function	PG_4 input pin	$\overline{CS0}$ output pin	PG_4 input pin	PG

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless v

$PG_3/\overline{CS1}$ The pin function is switched as shown below according to the opera and bit PG3DDR.

Operating Mode	Modes 1, 2, 3, 7*		Modes 4 to	
PG3DDR	0	1	0	
Pin function	PG_3 input pin	PG_3 output pin	PG_3 input pin	$\overline{CS}$

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless v

$PG_2/\overline{CS2}$ The pin function is switched as shown below according to the opera and bit PG2DDR.

Operating Mode	Modes 1, 2, 3, 7*		Modes 4 to	
PG2DDR	0	1	0	
Pin function	PG_2 input pin	PG_2 output pin	PG_2 input pin	$\overline{CS2}$

Note: * Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless v

Notes: 1. Modes 1 to 3 are not available on the F-ZTAT version.
Modes 2, 3, 6, and 7 are not available on the ROMless
2. When this pin is used as an external interrupt input, it s
used as an input/output pin with other functions.

PG_0/$\overline{ADTRG}$/$\overline{IRQ6}$ The pin function is switched as shown below according to the com
bits TRGS1 and TRGS0 (trigger select 1 and 0) in the A/D control
(ADCR).

PG0DDR	0	1
Pin function	PG_0 input	PG_0 outp
	$\overline{ADTRG}$ input pin*1	
	$\overline{IRQ6}$ interrupt input pin*2	

Notes: 1. $\overline{ADTRG}$ input when TRGS1 = TRGS0 = 1.
2. When this pin is used as an external interrupt input, it s
used as an input/output pin with other functions.

9.1.1 Features

- Maximum 16-pulse input/output
  - A total of 16 timer general registers (TGRs) are provided (four each for channe and two each for channels 1, 2, 4, and 5), each of which can be set independent output compare/input capture register
  - TGRC and TGRD for channels 0 and 3 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 dete
  - Counter clear operation: Counter clearing possible by compare match or input
  - Synchronous operation: Multiple timer counters (TCNT) can be written to simu 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 15-phase PWM output possible by combination with synchronou
- Buffer operation settable for channels 0 and 3
  - Input capture register double-buffering possible
  - Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
  - Two-phase encoder pulse up/down-count possible
- Cascaded operation
  - Channel 2 (channel 5) input clock operates as 32-bit counter by setting channel 4) overflow/underflow
- Fast access via internal 16-bit bus
  - Fast access is possible via a 16-bit bus interface

- A/D converter conversion start trigger can be generated
  - Channel 0 to 5 compare match A/input capture A signals can be used as A/D co conversion start trigger
- Module stop mode can be set
  - As the initial setting, TPU operation is halted. Register access is enabled by exit stop mode.

Table 9.1 lists the functions of the TPU.

		TCLKC TCLKD	TCLKB	TCLKB TCLKC	ϕ/4096 TCLKA	TCLKC
General registers		TGR0A TGR0B	TGR1A TGR1B	TGR2A TGR2B	TGR3A TGR3B	TGR4A TGR4B
General registers/ buffer registers		TGR0C TGR0D	—	—	TGR3C TGR3D	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
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	TGR compare match or input capture	TGR compare match or input capture

	input capture 0A	input capture 1A	input capture 2A	input capture 3A	input capture 4A
	• Compare match or input capture 0B	• Compare match or input capture 1B	• Compare match or input capture 2B	• Compare match or input capture 3B	• Compare match or input capture 4B
	• Compare match or input capture 0C	• Overflow • Underflow	• Overflow • Underflow	• Compare match or input capture 3C	• Overflow • Underflow
	• Compare match or input capture 0D			• Compare match or input capture 3D	
	• Overflow			• Overflow	

Legend:

— : Not possible

○ : Possible

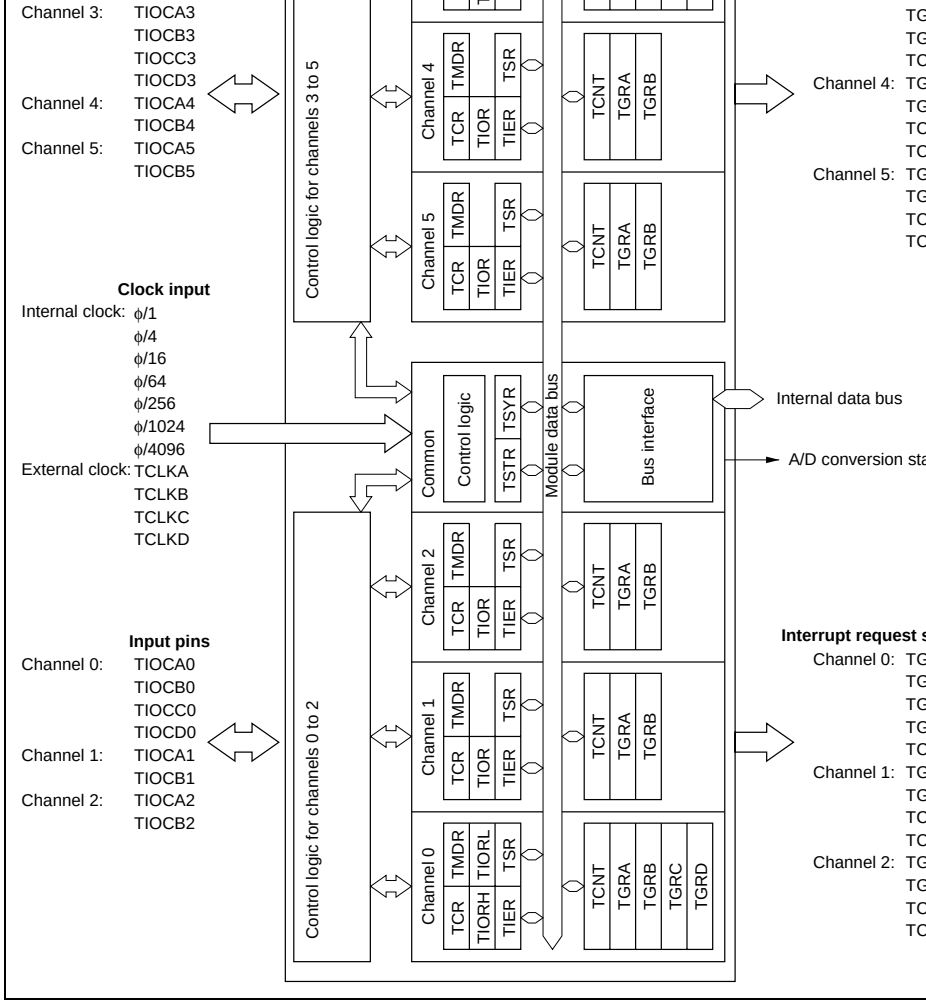


Figure 9.1 Block Diagram of TPU

				phase input)
	Clock input B	TCLKB	Input	External clock B input pin (Channel 1 and 5 phase counting phase input)
	Clock input C	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting phase input)
	Clock input D	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting phase input)
0	Input capture/out compare match A0	TIOCA0	I/O	TGR0A input capture input/outpu output/PWM output pin
	Input capture/out compare match B0	TIOCB0	I/O	TGR0B input capture input/outpu output/PWM output pin
	Input capture/out compare match C0	TIOCC0	I/O	TGR0C input capture input/outpu output/PWM output pin
	Input capture/out compare match D0	TIOCD0	I/O	TGR0D input capture input/outpu output/PWM output pin
1	Input capture/out compare match A1	TIOCA1	I/O	TGR1A input capture input/outpu output/PWM output pin
	Input capture/out compare match B1	TIOCB1	I/O	TGR1B input capture input/outpu output/PWM output pin
2	Input capture/out compare match A2	TIOCA2	I/O	TGR2A input capture input/outpu output/PWM output pin
	Input capture/out compare match B2	TIOCB2	I/O	TGR2B input capture input/outpu output/PWM output pin

	compare match D3			output/PWM output pin
4	Input capture/out compare match A4	TIOCA4	I/O	TGR4A input capture input/outp output/PWM output pin
	Input capture/out compare match B4	TIOCB4	I/O	TGR4B input capture input/outp output/PWM output pin
5	Input capture/out compare match A5	TIOCA5	I/O	TGR5A input capture input/outp output/PWM output pin
	Input capture/out compare match B5	TIOCB5	I/O	TGR5B input capture input/outp output/PWM output pin

	Timer I/O control register 0H	TIOR0H	R/W	H'00	H
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H
	Timer interrupt enable register 0	TIER0	R/W	H'40	H
	Timer status register 0	TSR0	R/(W)*2	H'C0	H
	Timer counter 0	TCNT0	R/W	H'0000	H
	Timer general register 0A	TGR0A	R/W	H'FFFF	H
	Timer general register 0B	TGR0B	R/W	H'FFFF	H
	Timer general register 0C	TGR0C	R/W	H'FFFF	H
	Timer general register 0D	TGR0D	R/W	H'FFFF	H
1	Timer control register 1	TCR1	R/W	H'00	H
	Timer mode register 1	TMDR1	R/W	H'C0	H
	Timer I/O control register 1	TIOR1	R/W	H'00	H
	Timer interrupt enable register 1	TIER1	R/W	H'40	H
	Timer status register 1	TSR1	R/(W)*2	H'C0	H
	Timer counter 1	TCNT1	R/W	H'0000	H
	Timer general register 1A	TGR1A	R/W	H'FFFF	H
	Timer general register 1B	TGR1B	R/W	H'FFFF	H
2	Timer control register 2	TCR2	R/W	H'00	H
	Timer mode register 2	TMDR2	R/W	H'C0	H
	Timer I/O control register 2	TIOR2	R/W	H'00	H
	Timer interrupt enable register 2	TIER2	R/W	H'40	H
	Timer status register 2	TSR2	R/(W)*2	H'C0	H
	Timer counter 2	TCNT2	R/W	H'0000	H
	Timer general register 2A	TGR2A	R/W	H'FFFF	H
	Timer general register 2B	TGR2B	R/W	H'FFFF	H

	Timer counter 3	TCNT3	R/W	H'0000	
	Timer general register 3A	TGR3A	R/W	H'FFFF	
	Timer general register 3B	TGR3B	R/W	H'FFFF	
	Timer general register 3C	TGR3C	R/W	H'FFFF	
	Timer general register 3D	TGR3D	R/W	H'FFFF	
4	Timer control register 4	TCR4	R/W	H'00	
	Timer mode register 4	TMDR4	R/W	H'C0	
	Timer I/O control register 4	TIOR4	R/W	H'00	
	Timer interrupt enable register 4	TIER4	R/W	H'40	
	Timer status register 4	TSR4	R/(W)*2	H'C0	
	Timer counter 4	TCNT4	R/W	H'0000	
	Timer general register 4A	TGR4A	R/W	H'FFFF	
	Timer general register 4B	TGR4B	R/W	H'FFFF	
5	Timer control register 5	TCR5	R/W	H'00	
	Timer mode register 5	TMDR5	R/W	H'C0	
	Timer I/O control register 5	TIOR5	R/W	H'00	
	Timer interrupt enable register 5	TIER5	R/W	H'40	
	Timer status register 5	TSR5	R/(W)*2	H'C0	
	Timer counter 5	TCNT5	R/W	H'0000	
	Timer general register 5A	TGR5A	R/W	H'FFFF	
	Timer general register 5B	TGR5B	R/W	H'FFFF	
All	Timer start register	TSTR	R/W	H'00	
	Timer synchro register	TSYR	R/W	H'00	
	Module stop control register	MSTPCR	R/W	H'3FFF	

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

	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1
Initial value :	0	0	0	0	0	0	0
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 1: TCR1
Channel 2: TCR2
Channel 4: TCR4
Channel 5: TCR5

Bit :	7	6	5	4	3	2	1
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1
Initial value :	0	0	0	0	0	0	0
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 six registers, one for each of channels 0 to 5. The TCR registers are initialized to H'00 by a in hardware standby mode.

Note: Make TCR settings only when TCNT operation is stopped.

		1	0	TCNT cleared by TGRB compare m capture
			1	TCNT cleared by counter clearing fo channel performing synchronous cle synchronous operation*1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare m capture*2
		1	0	TCNT cleared by TGRD compare m capture*2
			1	TCNT cleared by counter clearing fo channel performing synchronous cle synchronous operation*1

Channel	Bit 7 Reserved*3	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRA compare m capture
		1	0	TCNT cleared by TGRB compare m capture
			1	TCNT cleared by counter clearing fo channel performing synchronous cle synchronous operation*1

Notes: 1. Synchronous operation setting is performed by setting the SYNC bit in TSY
2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared bec buffer register setting has priority, and compare match/input capture does n
3. Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and canno modified.

	1	Count at falling edge
1	—	Count at both edges

Note: Internal clock edge selection is valid when the input clock is ϕ/4 or slower. This s
ignored if the input clock is ϕ/1, or when overflow/underflow of another channel is

Bits 2, 1, and 0—Time Prescaler 2, 1, and 0 (TPSC2 to TPSC0): These bits select th
counter clock. The clock source can be selected independently for each channel. Table
the clock sources that can be set for each channel.

Table 9.4 TPU Clock Sources

Channel	Internal Clock							External Clock			
	ϕ/1	ϕ/4	ϕ/16	ϕ/64	ϕ/256	ϕ/1024	ϕ/4096	TCLKA	TCLKB	TCLKC	TCLKD
0	○	○	○	○				○	○	○	○
1	○	○	○	○	○			○	○		
2	○	○	○	○		○		○	○	○	
3	○	○	○	○	○	○	○	○			
4	○	○	○	○		○		○		○	
5	○	○	○	○	○			○		○	○

Legend:
○ : Setting
Blank : No setting

			1	External clock: counts on TCLKB pi
		1	0	External clock: counts on TCLKC pi
			1	External clock: counts on TCLKD pi

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
1	0	0	0	Internal clock: counts on ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pi
			1	External clock: counts on TCLKB pi
		1	0	Internal clock: counts on ϕ/256
			1	Counts on TCNT2 overflow/underflo

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 ϕ/1
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pi
			1	External clock: counts on TCLKB pi
		1	0	External clock: counts on TCLKC pi
			1	Internal clock: counts on ϕ/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

			1	Internal clock: counts on ϕ/1024
		1	0	Internal clock: counts on ϕ/256
			1	Internal clock: counts on ϕ/4096

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
4	0	0	0	Internal clock: counts on ϕ/1 (I
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin
			1	External clock: counts on TCLKC pin
		1	0	Internal clock: counts on ϕ/1024
			1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description
5	0	0	0	Internal clock: counts on ϕ/1 (I
			1	Internal clock: counts on ϕ/4
		1	0	Internal clock: counts on ϕ/16
			1	Internal clock: counts on ϕ/64
	1	0	0	External clock: counts on TCLKA pin
			1	External clock: counts on TCLKC pin
		1	0	Internal clock: counts on ϕ/256
			1	External clock: counts on TCLKD pin

Note: This setting is ignored when channel 5 is in phase counting mode.

R/W	:	—	—	R/W	R/W	R/W	R/W	R/W

Channel 1: TMDR1
Channel 2: TMDR2
Channel 4: TMDR4
Channel 5: TMDR5

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	MD3	MD2	MD1
Initial value	:	1	1	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W

The TMDR registers are 8-bit readable/writable registers that are used to set the opera for each channel. The TPU has six TMDR registers, one for each channel. The TMDR are initialized to H'C0 by a reset, and in hardware standby mode.

Note: Make TMDR settings only when TCNT operation is stopped.

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 norr TGRB and TGRD are to be used together for buffer operation. When TGRD is used a register, TGRD input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as cannot be modified.

Bit 5

BFB	Description
0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

0	TGRA operates normally	(
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 operati

Bit 3	Bit 2	Bit 1	Bit 0		
MD3[*1]	**MD2**[*2]	**MD1**	**MD0**	**Description**	
0	0	0	0	Normal operation	(
			1	Reserved	
		1	0	PWM mode 1	
			1	PWM mode 2	
	1	0	0	Phase counting mode 1	
			1	Phase counting mode 2	
		1	0	Phase counting mode 3	
			1	Phase counting mode 4	
1	*	*	*	—	

*

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 channels 0 and 3. In this case, 0 sho be written to MD2.

Bit	:	7	6	5	4	3	2	1
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 0: TIOR0L
Channel 3: TIOR3L

Bit	:	7	6	5	4	3	2	1
		IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1
Initial value	:	0	0	0	0	0	0	0
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 invali
register operates as a buffer register.

The TIOR registers are 8-bit registers that control the TGR registers. The TPU has eig
registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. The
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 speci
TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note
PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

0	0	0	0	0	TGR0B is output compare register	Output disabled	(
				1		Initial output is 0 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at ri
				1			Input capture at fa
			1	*			Input capture at b
		1	*	*		Capture input source is channel 1/count clock	Input capture at T count- up/count-d

*

Note: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ϕ/1 is used as the count clock, this setting is invalid and input capture is not generated.

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR0D is input capture register*2	Capture input source is TIOCD0 pin	Input capture at
			1			Input capture at
		1	*			Input capture at
	1	*	*		Capture input source is channel 1/count clock	Input capture at count-up/count-d

Notes: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ϕ/1 is used as th count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer regi setting is invalid and input capture/output compare is not generated.

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at g of TGR0C compa match/input captu

*

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at
			1			Input capture at
		1	*			Input capture at

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR3B is input capture register	Capture input source is TIOCB3 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is channel 4/count clock	Input capture at T count-up/count-do

*

Note: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ϕ/1 is used as the count clock, this setting is invalid and input capture is not generated.

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR3D is input capture register*2	Capture input source is TIOCD3 pin	Input capture at
			1			Input capture at
		1	*			Input capture at
	1	*	*		Capture input source is channel 4/count clock	Input capture at count-up/count-d

Notes: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and φ/1 is used as th count clock, this setting is invalid and input capture is not generated.

2. When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer regi setting is invalid and input capture/output compare is not generated.

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR4B is input capture register	Capture input source is TIOCB4 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is TGR3C compare match/ input capture	Input capture at g of TGR3C compa input capture

*

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	*	0	0	TGR5B is input capture register	Capture input source is TIOCB5 pin	Input capture at
			1			Input capture at
		1	*			Input capture at

0	0	0	0	0	TGR0A is output compare register	Output disabled	(
				1		Initial output is 0 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compa
			1	0			1 output at compa
				1			Toggle output at c match
	1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at ri
				1			Input capture at fa
			1	*			Input capture at b
		1	*	*		Capture input source is channel 1/ count clock	Input capture at T count-up/count-do

*

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR0C is input capture register*1	Capture input source is TIOCC0 pin	Input capture at
			1			Input capture at
		1	*			Input capture at
	1	*	*		Capture input source is channel 1/count clock	Input capture at count-up/count-d

Note: 1. When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer regi setting is invalid and input capture/output compare is not generated.

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at g of channel 0/TGR compare match/in capture

*

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at
			1			Input capture at
		1	*			Input capture at

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR3A is input capture register	Capture input source is TIOCA3 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is channel 4/count clock	Input capture at T count-up/count-do

*

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	0	0	0	TGR3C is input capture register*1	Capture input source is TIOCC3 pin	Input capture at
			1			Input capture at
		1	*			Input capture at
	1	*	*		Capture input source is channel 4/count clock	Input capture at count-up/count-d

Note: 1. When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buffer regi setting is invalid and input capture/output compare is not generated.

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c match
1	0	0	0	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b
	1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at g of TGR3A compa match/input captu

*

						match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at comp
		1	0			1 output at comp
			1			Toggle output at match
1	*	0	0	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at
			1			Input capture at
		1	*			Input capture at

R/W	:	R/W	—	—	R/W	R/W	R/W	R/W

Channel 1: TIER1
Channel 2: TIER2
Channel 4: TIER4
Channel 5: TIER5

Bit	:	7	6	5	4	3	2	1
		TTGE	—	TCIEU	TCIEV	—	—	TGIEB
Initial value	:	0	1	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	—	—	R/W

The TIER registers are 8-bit registers that control enabling or disabling of interrupt requ
each channel. The TPU has six TIER registers, one for each channel. The TIER register
initialized to H'40 by a reset, and in hardware standby mode.

Bit 7—A/D Conversion Start Request Enable (TTGE): Enables or disables generati
conversion start requests by TGRA input capture/compare match.

Bit 7

TTGE	Description	
0	A/D conversion start request generation disabled	(
1	A/D conversion start request generation enabled	

Bit 6—Reserved: Read-only bit, always read as 1.

1	Interrupt requests (TCIU) by TCFU enabled

Bit 4—Overflow Interrupt Enable (TCIEV): Enables or disables interrupt requests the TCFV flag when the TCFV flag in TSR is set to 1.

Bit 4

TCIEV	Description
0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Bit 3—TGR Interrupt Enable D (TGIED): Enables or disables interrupt requests (T TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modifi

Bit 3

TGIED	Description
0	Interrupt requests (TGID) by TGFD bit disabled
1	Interrupt requests (TGID) by TGFD bit enabled

Bit 2—TGR Interrupt Enable C (TGIEC): Enables or disables interrupt requests (T TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modifi

Bit 2

TGIEC	Description
0	Interrupt requests (TGIC) by TGFC bit disabled
1	Interrupt requests (TGIC) by TGFC bit enabled

Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables interrupt requests (TG
TGFA bit when the TGFA bit in TSR is set to 1.

Bit 0

TGIEA	Description	
0	Interrupt requests (TGIA) by TGFA bit disabled	(
1	Interrupt requests (TGIA) by TGFA bit enabled	

R/W	:	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Can only be written with 0 for flag clearing.

Channel 1: TSR1
Channel 2: TSR2
Channel 4: TSR4
Channel 5: TSR5

Bit	:	7	6	5	4	3	2	1
		TCFD	—	TCFU	TCFV	—	—	TGFB
Initial value	:	1	1	0	0	0	0	0
R/W	:	R	—	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 registers, one for each channel. The TSR registers are initialized to H'C0 by a reset, ar hardware standby mode.

Bit 7—Count Direction Flag (TCFD): Status flag that shows the direction in which counts in channels 1, 2, 4, and 5.

In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified.

Bit 7

TCFD	Description
0	TCNT counts down
1	TCNT counts up

Bit 6—Reserved: Read-only bit, always read as 1.

	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)

Bit 4—Overflow Flag (TCFV): Status flag that indicates that TCNT overflow has occ

Bit 4

TCFV	Description
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)

Bit 3—Input Capture/Output Compare Flag D (TGFD): Status flag that indicates th occurrence of TGRD input capture or compare match in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modifie

Bit 3

TGFD	Description
0	[Clearing conditions] (• When DTC is activated by TGID interrupt while DISEL bit of MRB in DT • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare reg • When TCNT value is transferred to TGRD by input capture signal while functioning as input capture register

	• When DTC is activated by TGIC interrupt while DISEL bit of MRB in D
	• When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions]
	• When TCNT = TGRC while TGRC is functioning as output compare re
	• When TCNT value is transferred to TGRC by input capture signal while functioning as input capture register

Bit 1—Input Capture/Output Compare Flag B (TGFB): Status flag that indicates t occurrence of TGRB input capture or compare match.

Bit 1

TGFB	Description
0	[Clearing conditions]
	• When DTC is activated by TGIB interrupt while DISEL bit of MRB in D
	• When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions]
	• When TCNT = TGRB while TGRB is functioning as output compare re
	• When TCNT value is transferred to TGRB by input capture signal while functioning as input capture register

1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare reg • When TCNT value is transferred to TGRA by input capture signal while functioning as input capture register

9.2.6 Timer Counter (TCNT)

Channel 0: TCNT0 (up-counter)
Channel 1: TCNT1 (up/down-counter*)
Channel 2: TCNT2 (up/down-counter*)
Channel 3: TCNT3 (up-counter)
Channel 4: TCNT4 (up/down-counter*)
Channel 5: TCNT5 (up/down-counter*)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	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 m when counting overflow/underflow on another channel. In other cases they f as up-counters.

The TCNT registers are 16-bit counters. The TPU has six TCNT counters, one for each

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 unit.

The TGR registers are 16-bit registers with a dual function as output compare and inp
registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each
1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for opera
buffer registers*. The TGR registers are initialized to H'FFFF by a reset, and in hardw
mode.

The TGR registers cannot be accessed in 8-bit units; they must always be accessed as

Note: * TGR buffer register combinations are TGRA–TGRC and TGRB–TGRD.

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

Note: When setting the operating mode in TMDR or setting the count clock in TCR, f
the TCNT counter.

Bits 7 and 6—Reserved: Should always be written with 0.

Bits 5 to 0—Counter Start 5 to 0 (CST5 to CST0): These bits select operation or stop
TCNT.

Bit n

CSTn	Description
0	TCNTn count operation is stopped (I
1	TCNTn performs count operation

Note: If 0 is written to the CST bit during operation with the TIOC pin designated for ou
counter stops but the TIOC pin output compare output level is retained. If TIOR i
when the CST bit is cleared to 0, the pin output level will be changed to the set in
value.

operation for the channel 0 to 5 TCNT counters. A channel performs synchronous ope
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 and 6—Reserved: Should always be written with 0.

Bits 5 to 0—Timer Synchro 5 to 0 (SYNC5 to SYNC0): These bits select whether o
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.

Bit n

SYNCn	Description
0	TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

Notes: 1. To set synchronous operation, the SYNC bits for at least two channels mus
2. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing must also be set by means of bits CCLR2 to CCLR0 in TCR.

MSTPCR is a 16-bit readable/writable register that performs module stop mode control

When the MSTP13 bit in MSTPCR is set to 1, TPU operation stops at the end of the bu
a transition is made to module stop mode. Registers cannot be read or written to in mod
mode. For details, see section 19.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not ini
software standby mode.

Bit 13—Module Stop (MSTP13): Specifies the TPU module stop mode.

Bit 13	
MSTP13	**Description**
0	TPU module stop mode cleared
1	TPU module stop mode set (I

An example of 16-bit register access operation is shown in figure 9.2.

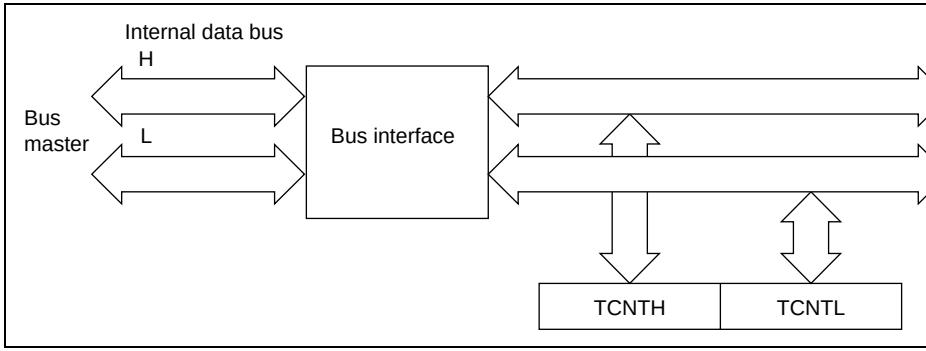


Figure 9.2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 B

9.3.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits w registers can be read and written to in 16-bit units. They can also be read and written t units.

Examples of 8-bit register access operation are shown in figures 9.3 to 9.5.

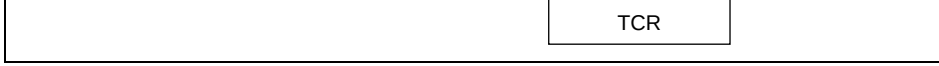


Figure 9.3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8

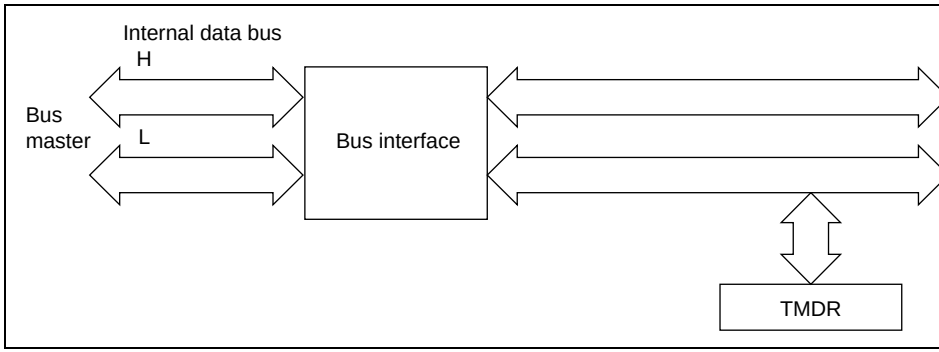


Figure 9.4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8

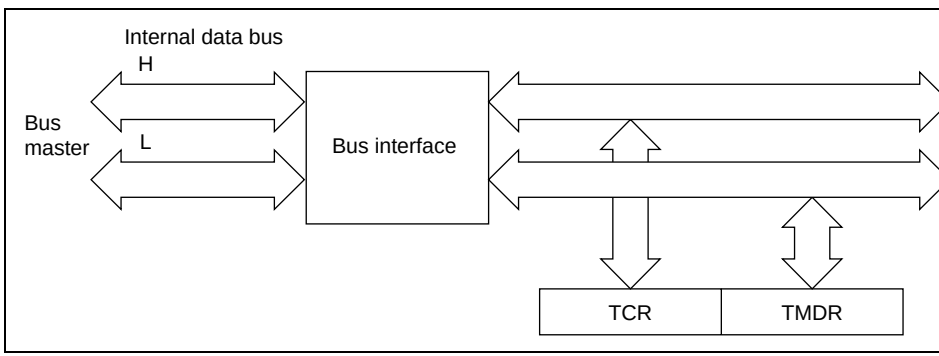


Figure 9.5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (

Each TGR can be used as an input capture register or output compare register.

Synchronous Operation: When synchronous operation is designated for a channel, T that channel performs synchronous presetting. That is, when TCNT for a channel desi synchronous operation is rewritten, the TCNT counters for the other channels are also the same time. Synchronous clearing of the TCNT counters is also possible by setting 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 cha 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 pr held in TGR is transferred to the buffer register.

Cascaded Operation: The channel 1 counter (TCNT1), channel 2 counter (TCNT2), counter (TCNT4), and channel 5 counter (TCNT5) can be connected together to opera bit counter.

PWM Mode: In this mode, a PWM waveform is output. The output level can be set b TIOR. A PWM waveform with a duty of between 0% and 100% can be output, accor setting of each TGR register.

Phase Counting Mode: In this mode, TCNT is incremented or decremented by detec phases of two clocks input from the external clock input pins in channels 1, 2, 4, and 5 phase counting mode is set, the corresponding TCLK pin functions as the clock pin, a performs up- or down-counting.

This can be used for two-phase encoder pulse input.

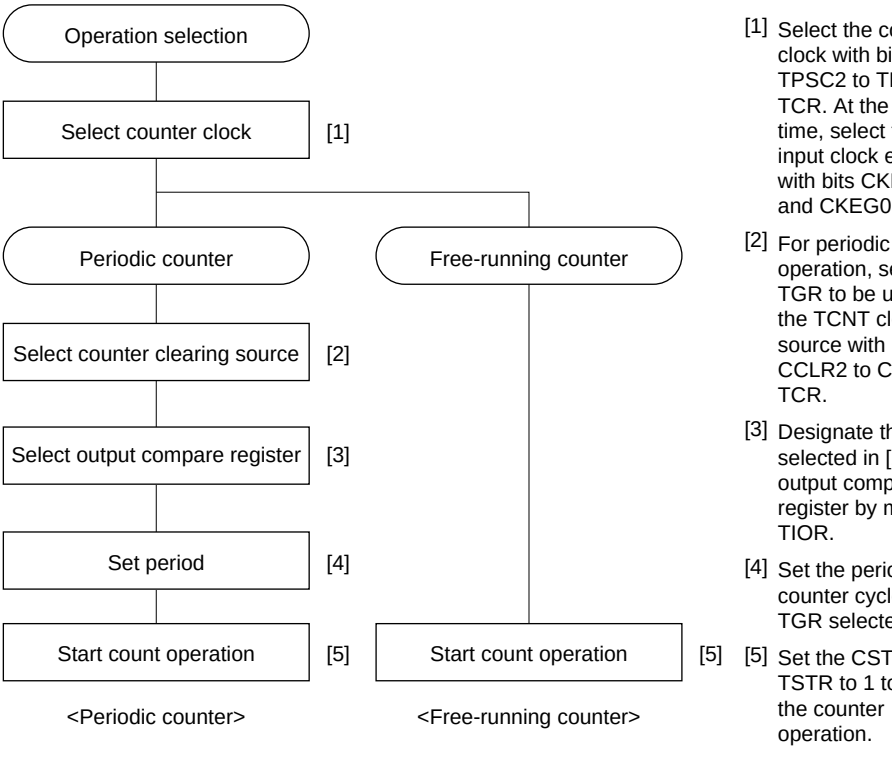


Figure 9.6 Example of Counter Operation Setting Procedure

Figure 9.7 illustrates free-running counter operation.

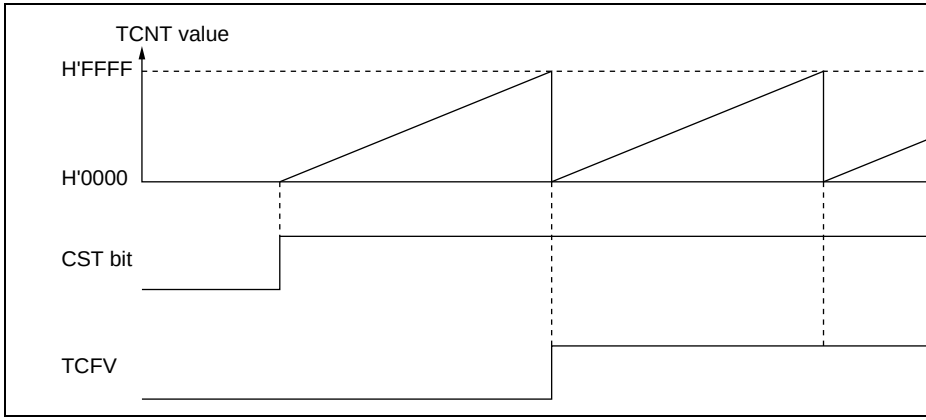


Figure 9.7 Free-Running Counter Operation

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

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

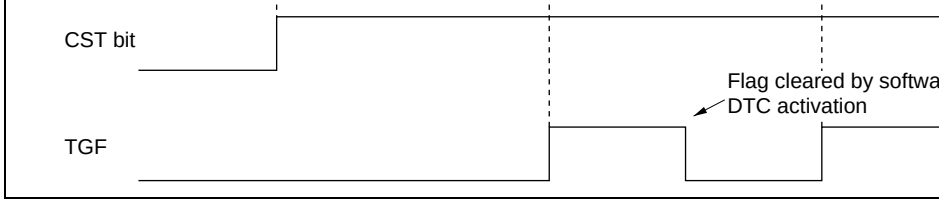


Figure 9.8 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output corresponding output pin using compare match.

- Example of setting procedure for waveform output by compare match
  Figure 9.9 shows an example of the setting procedure for waveform output by comp

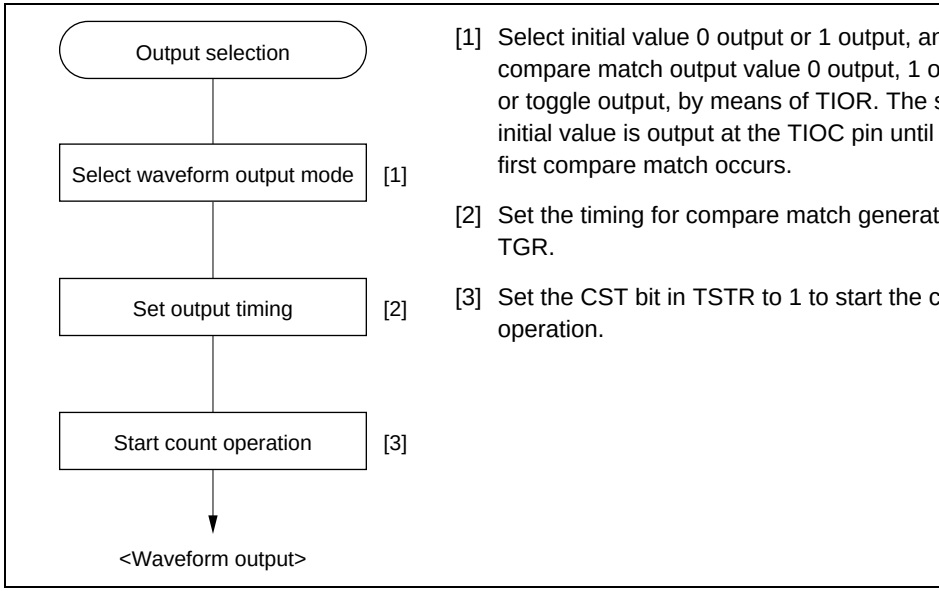


Figure 9.9 Example Of Setting Procedure for Waveform Output By Compare

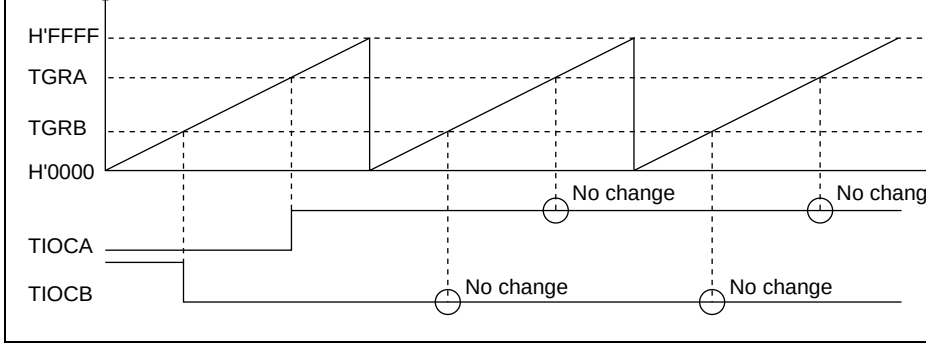


Figure 9.10 Example of 0 Output/1 Output Operation

Figure 9.11 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter cle
performed by compare match B), and settings have been made so that output is tog
compare match A and compare match B.

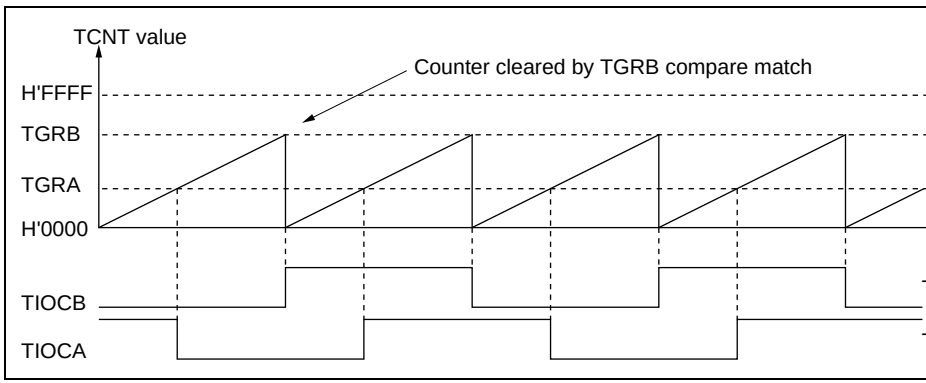


Figure 9.11 Example of Toggle Output Operation

Input capture will not be generated if φ/1 is selected.

- Example of input capture operation setting procedure
  Figure 9.12 shows an example of the input capture operation setting procedure.

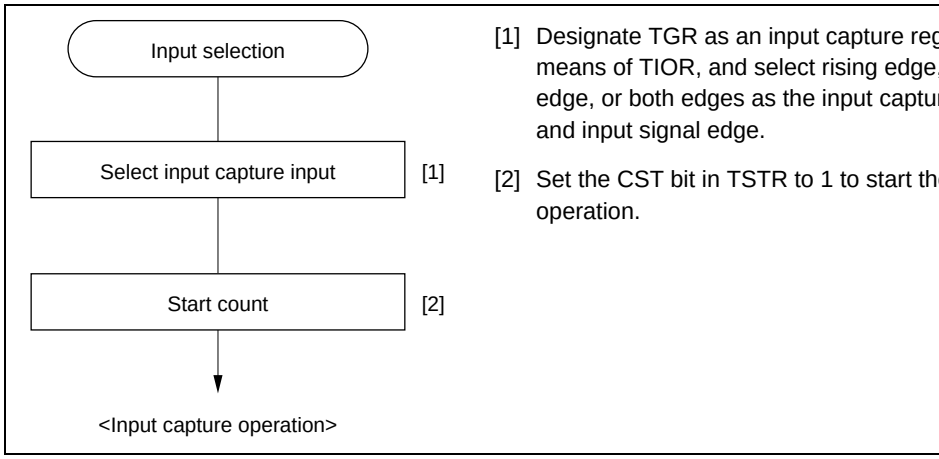


Figure 9.12 Example of Input Capture Operation Setting Procedure

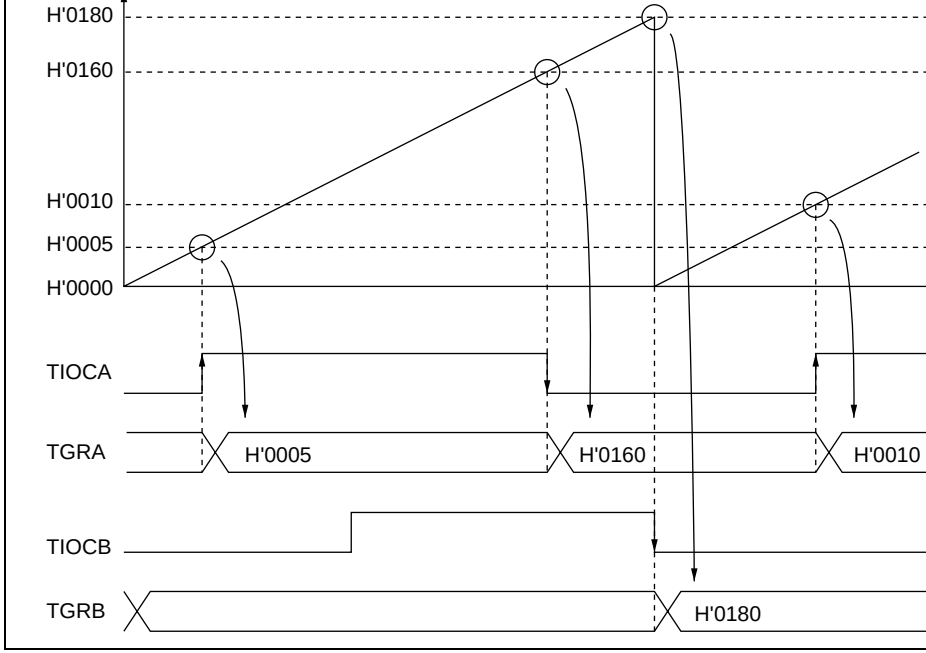
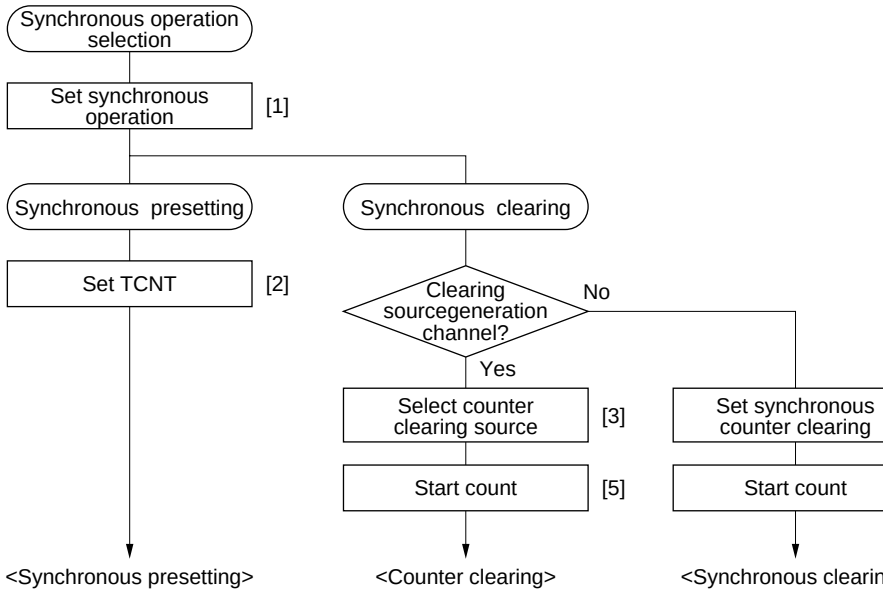


Figure 9.13 Example of Input Capture Operation

Example of Synchronous Operation Setting Procedure: Figure 9.14 shows an exam
synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchr 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 com etc.

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter cle source.

[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 9.14 Example of Synchronous Operation Setting Procedure

For details of PWM modes, see section 9.4.6, PWM Modes.

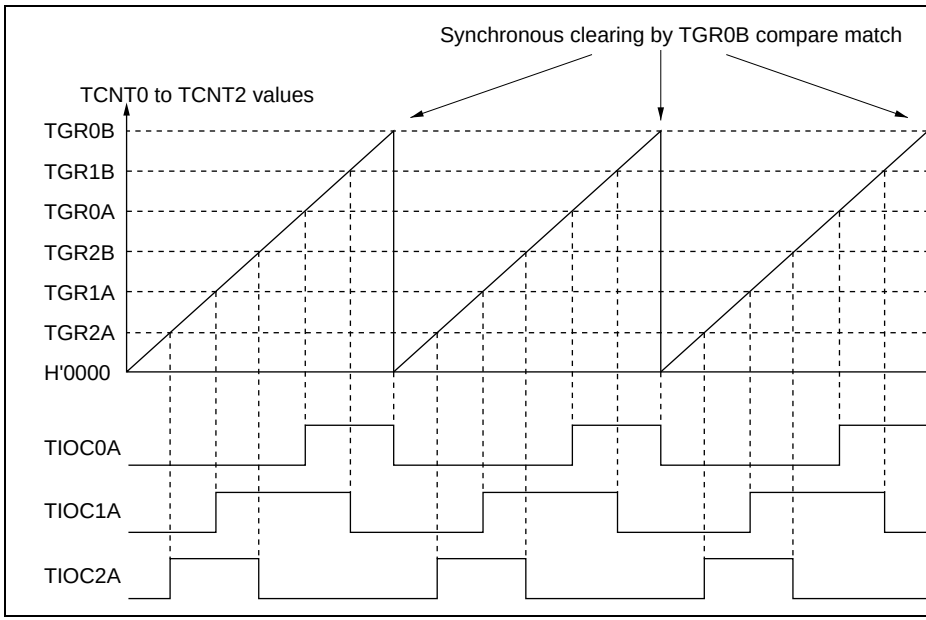


Figure 9.15 Example of Synchronous Operation

Table 9.5 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGR0A	TGR0C
	TGR0B	TGR0D
3	TGR3A	TGR3C
	TGR3B	TGR3D

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding
  transferred to the timer general register.

  This operation is illustrated in figure 9.16.

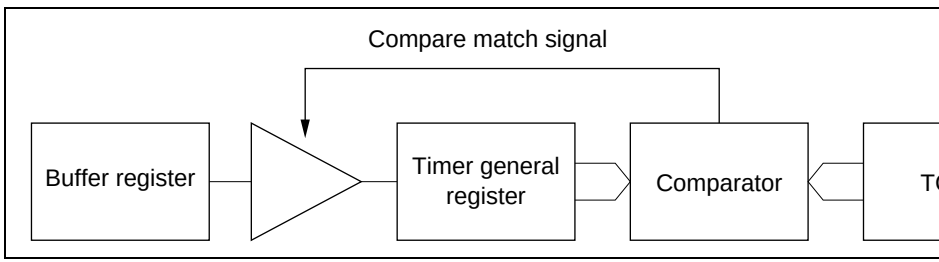


Figure 9.16 Compare Match Buffer Operation

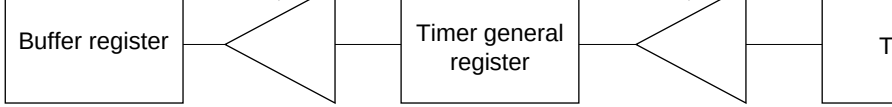
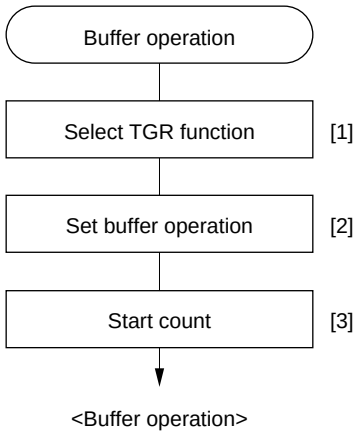


Figure 9.17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 9.18 shows an example of operation setting procedure.



[1] Designate TGR as an input capture regis output compare register by means of TIO

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

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

Figure 9.18 Example of Buffer Operation Setting Procedure

value in buffer register TGRC is simultaneously transferred to timer general registe
This operation is repeated each time compare match A occurs.

For details of PWM modes, see section 9.4.6, PWM Modes.

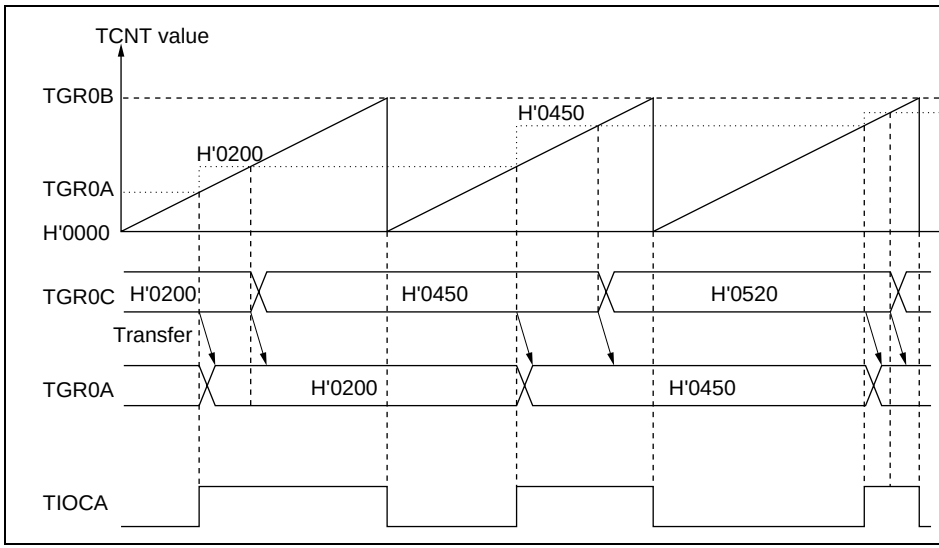


Figure 9.19 Example of Buffer Operation (1)

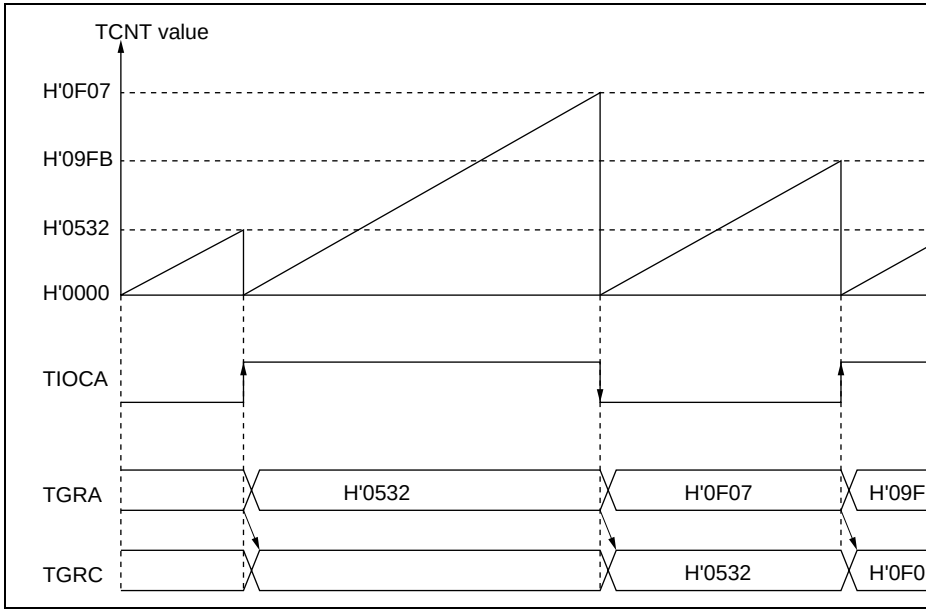


Figure 9.20 Example of Buffer Operation (2)

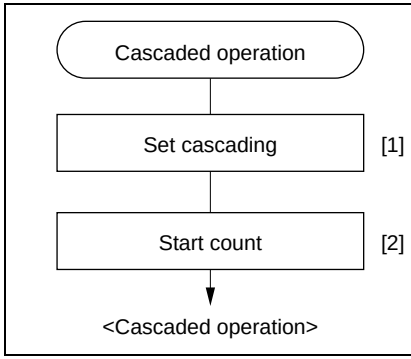
Table 9.6 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting i
and the counter operates independently in phase counting mode.

Table 9.6 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT1	TCNT2
Channels 4 and 5	TCNT4	TCNT5

Example of Cascaded Operation Setting Procedure: Figure 9.21 shows an example
setting procedure for cascaded operation.



[1] Set bits TPSC2 to TPSC0 in the channel
(channel 4) TCR to B'111 to select TCNT
(TCNT5) overflow/underflow counting.

[2] Set the CST bit in TSTR for the upper and
channel to 1 to start the count operation.

Figure 9.21 Cascaded Operation Setting Procedure

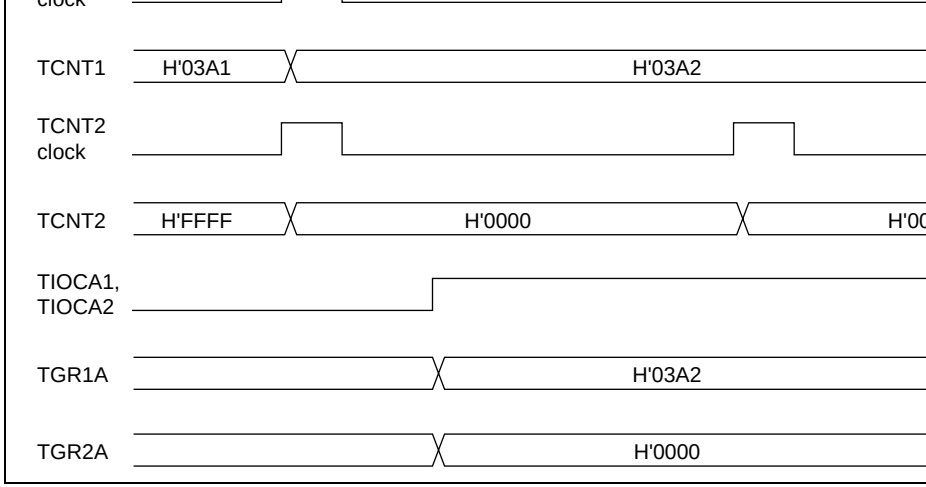


Figure 9.22 Example of Cascaded Operation (1)

Figure 9.23 illustrates the operation when counting upon TCNT2 overflow/underflow for TCNT1, and phase counting mode has been designated for channel 2.

TCNT1 is incremented by TCNT2 overflow and decremented by TCNT2 underflow.

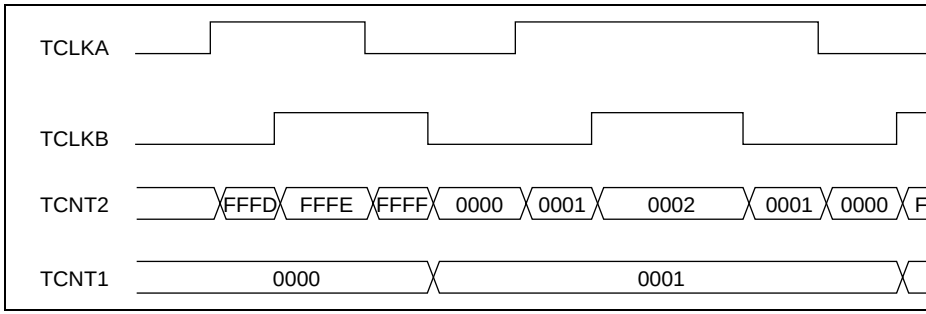


Figure 9.23 Example of Cascaded Operation (2)

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 TGRC with TGRD. The output specified by bits IOA3 to IOA0 and IOC3 to IOC0 output from the TIOCA and TIOCC pins at compare matches A and C, and the outp specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare m and D. The initial output value is the value set in TGRA or TGRC. If the set values TGRs are identical, the output value does not change when a compare match occurs

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

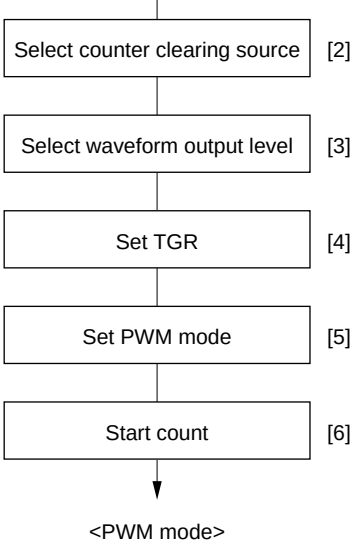
  PWM output is generated using one TGR as the cycle register and the others as duty The output specified in TIOR is performed by means of compare matches. Upon co clearing by a synchronization register compare match, the output value of each pin value set in TIOR. If the set values of the cycle and duty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

  In PWM mode 2, a maximum 15-phase PWM output is possible by combined use w synchronous operation.

The correspondence between PWM output pins and registers is shown in table 9.7.

1	TGR1A	TIOCA1	TIOCA1
	TGR1B		TIOCB1
2	TGR2A	TIOCA2	TIOCA2
	TGR2B		TIOCB2
3	TGR3A	TIOCA3	TIOCA3
	TGR3B		TIOCB3
	TGR3C	TIOCC3	TIOCC3
	TGR3D		TIOCD3
4	TGR4A	TIOCA4	TIOCA4
	TGR4B		TIOCB4
5	TGR5A	TIOCA5	TIOCA5
	TGR5B		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the



[2] [illegible] bits CCLR2 to CCLR0 in TCR to sele
TGR to be used as the TCNT clearing sou

[3] Use TIOR to designate the TGR as an ou
compare register, and select the initial val
output value.

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

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

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

Figure 9.24 Example of PWM Mode Setting Procedure

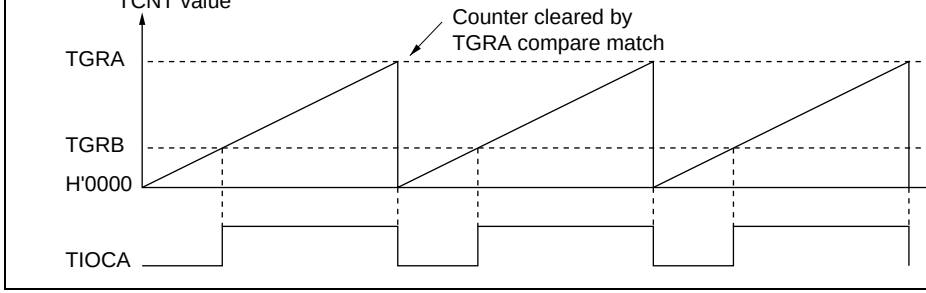


Figure 9.25 Example of PWM Mode Operation (1)

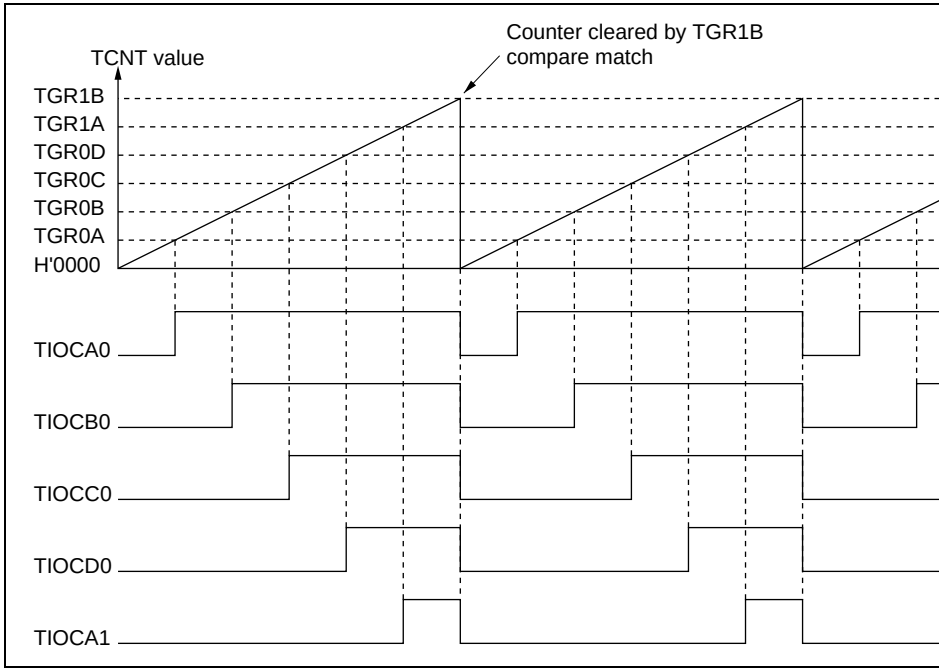


Figure 9.26 Example of PWM Mode Operation (2)

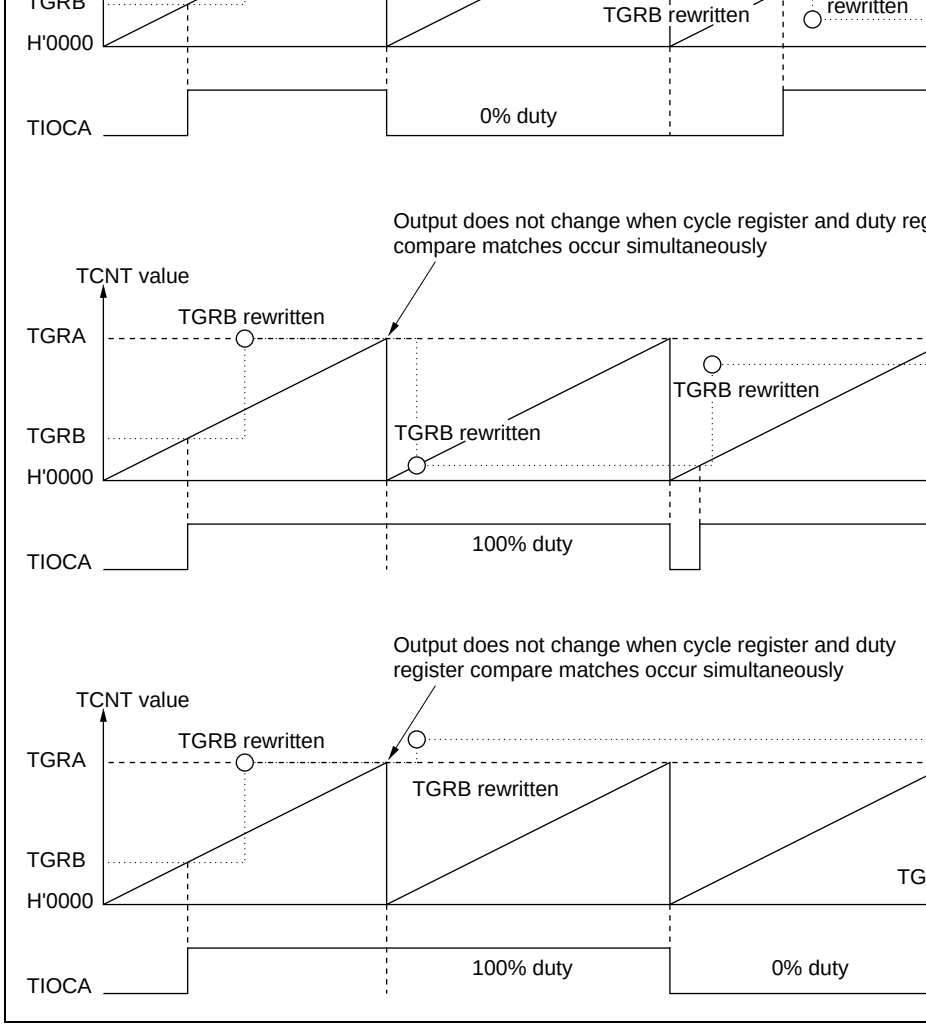


Figure 9.27 Example of PWM Mode Operation (3)

used.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when
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 in
whether TCNT is counting up or down.

Table 9.8 shows the correspondence between external clock pins and channels.

Table 9.8 Phase Counting Mode Clock Input Pins

	External Clock Pin	
Channels	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 9.28 shows an exampl
phase counting mode setting procedure.

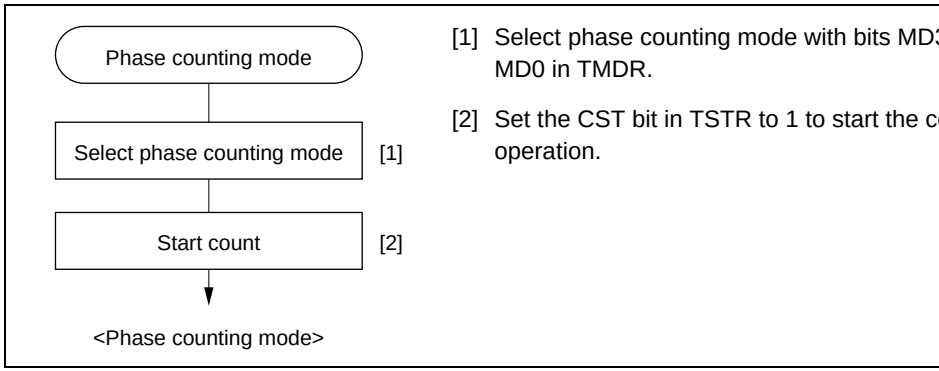


Figure 9.28 Example of Phase Counting Mode Setting Procedure

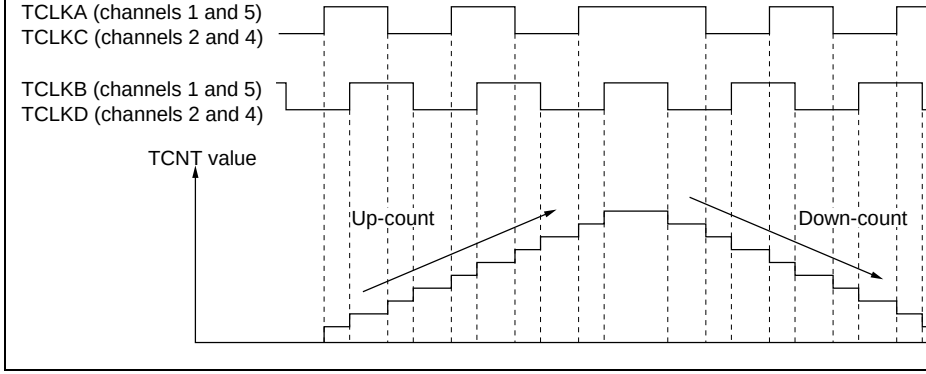


Figure 9.29 Example of Phase Counting Mode 1 Operation

Table 9.9 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-coun
Low level	↑	
↑	High level	
↓	Low level	

Legend:
↑ : Rising edge
↓ : Falling edge

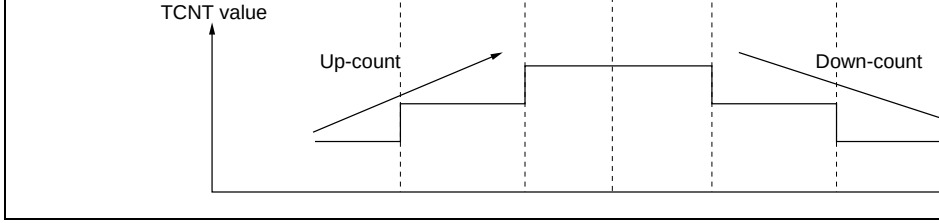


Figure 9.30 Example of Phase Counting Mode 2 Operation

Table 9.10 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	
↑	High level	
↓	Low level	Down-count

Legend:
↑ : Rising edge
↓ : Falling edge

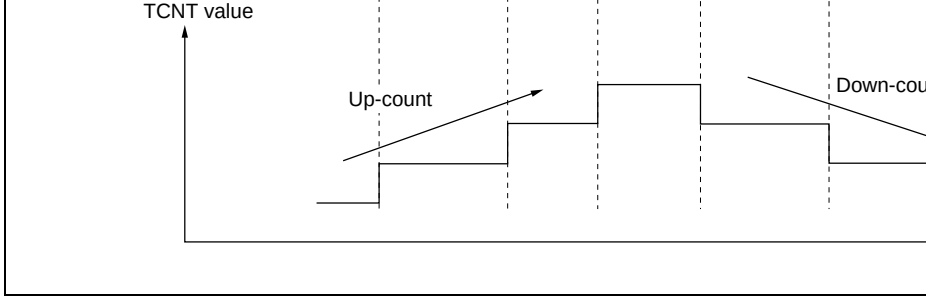


Figure 9.31 Example of Phase Counting Mode 3 Operation

Table 9.11 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	
↑	Low level	
↓	High level	Up-count
High level	↓	Down-cour
Low level	↑	Don't care
↑	High level	
↓	Low level	

Legend:
↑ : Rising edge
↓ : Falling edge

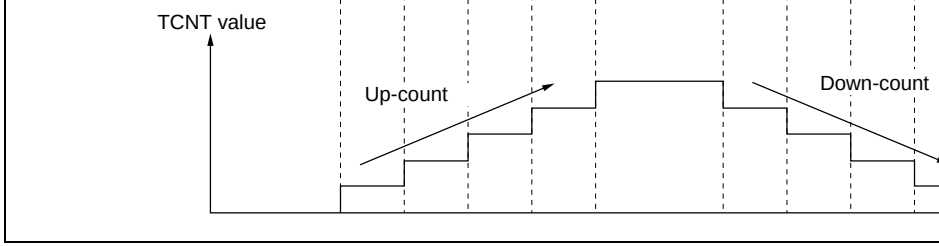


Figure 9.32 Example of Phase Counting Mode 4 Operation

Table 9.12 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	⌐↑	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

control period. TGR0B is used for input capture, with TGR0B and TGR0D operating mode. The channel 1 counter input clock is designated as the TGR0B input capture so detection of the pulse width of 2-phase encoder 4-multiplication pulses is performed.

TGR1A and TGR1B for channel 1 are designated for input capture, channel 0 TGR0A TGR0C compare matches are selected as the input capture source, and store the up/do values for the control periods.

This procedure enables accurate position/speed detection to be achieved.

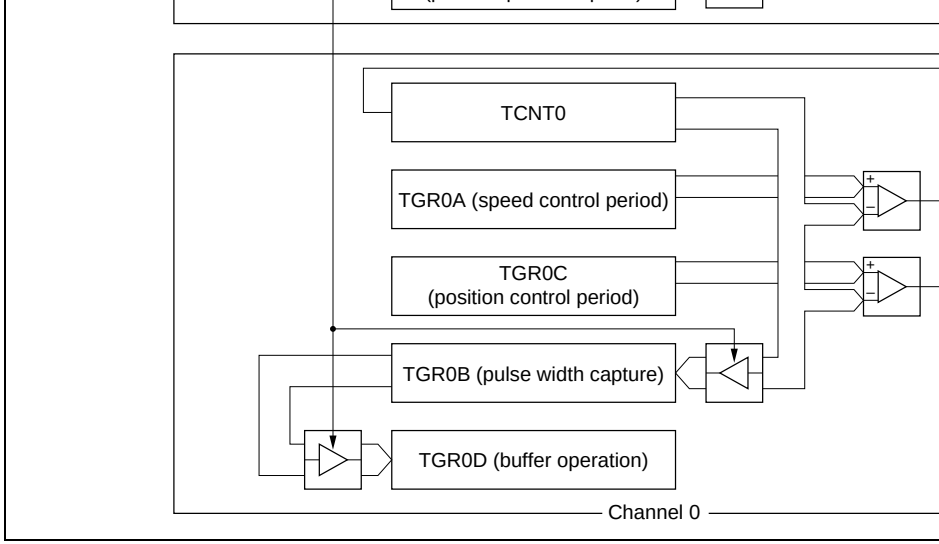


Figure 9.33 Phase Counting Mode Application Example

9.5 Interrupts

9.5.1 Interrupt Sources and Priorities

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCN overflow, and TCNT underflow. Each interrupt source has its own status flag and enabl bit, allowing generation of interrupt request signals to be enabled or disabled individual

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requeste interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority o a channel is fixed. For details, see section 5, Interrupt Controller.

	TGI0D	TGR0D input capture/compare match	Possible
	TCI0V	TCNT0 overflow	Not possible
1	TGI1A	TGR1A input capture/compare match	Possible
	TGI1B	TGR1B input capture/compare match	Possible
	TCI1V	TCNT1 overflow	Not possible
	TCI1U	TCNT1 underflow	Not possible
2	TGI2A	TGR2A input capture/compare match	Possible
	TGI2B	TGR2B input capture/compare match	Possible
	TCI2V	TCNT2 overflow	Not possible
	TCI2U	TCNT2 underflow	Not possible
3	TGI3A	TGR3A input capture/compare match	Possible
	TGI3B	TGR3B input capture/compare match	Possible
	TGI3C	TGR3C input capture/compare match	Possible
	TGI3D	TGR3D input capture/compare match	Possible
	TCI3V	TCNT3 overflow	Not possible
4	TGI4A	TGR4A input capture/compare match	Possible
	TGI4B	TGR4B input capture/compare match	Possible
	TCI4V	TCNT4 overflow	Not possible
	TCI4U	TCNT4 underflow	Not possible
5	TGI5A	TGR5A input capture/compare match	Possible
	TGI5B	TGR5B input capture/compare match	Possible
	TCI5V	TCNT5 overflow	Not possible
	TCI5U	TCNT5 underflow	Not possible

Note: This table shows the initial state immediately after a reset. The relative channel can be changed by the interrupt controller.

each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 wh
TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The
request is cleared by clearing the TCFU flag to 0. The TPU has four overflow interrupt
for channels 1, 2, 4, and 5.

9.5.2 DTC Activation

DTC Activation: The DTC can be activated by the TGR input capture/compare match
for a channel. For details, see section 7, Data Transfer Controller.

A total of 16 TPU input capture/compare match interrupts can be used as DTC activatio
four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

9.5.3 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a cl

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurr
TGRA input capture/compare match on a particular channel, a request to start A/D con
sent to the A/D converter. If the TPU conversion start trigger has been selected on the A
converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as
converter conversion start sources, one for each channel.

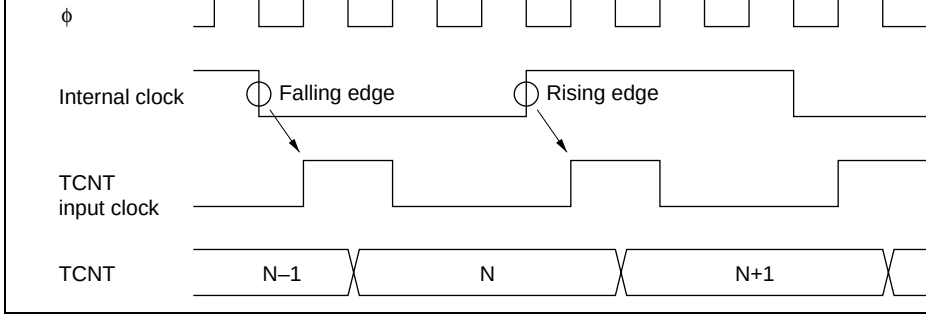


Figure 9.34 Count Timing in Internal Clock Operation

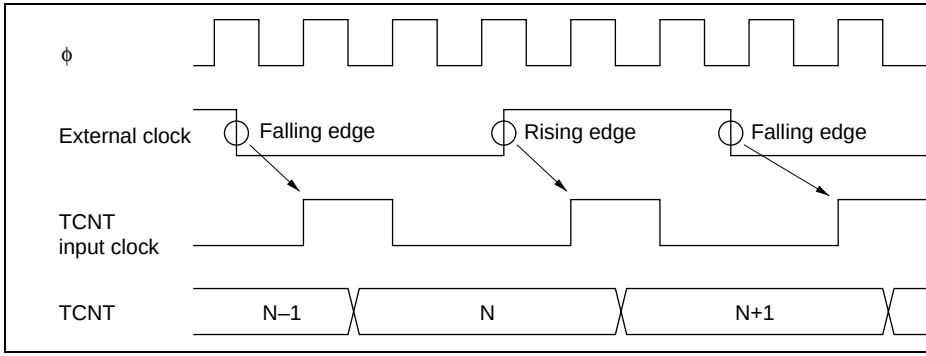


Figure 9.35 Count Timing in External Clock Operation

Output Compare Output Timing: A compare match signal is generated in the final s which TCNT and TGR match (the point at which the count value matched by TCNT i When a compare match signal is generated, the output value set in TIOR is output at t compare output pin. After a match between TCNT and TGR, the compare match signa generated until the TCNT input clock is generated.

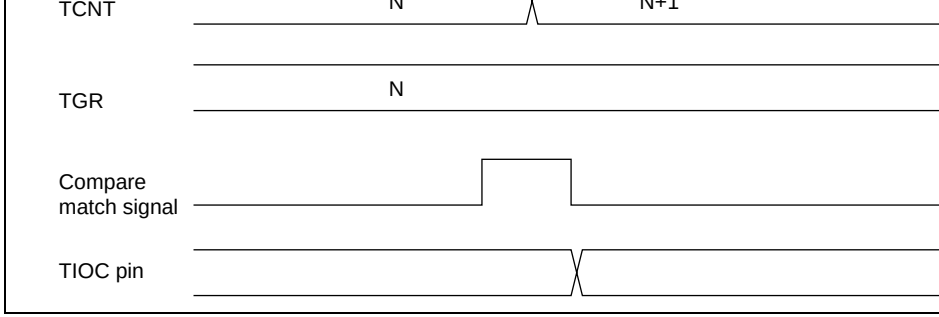


Figure 9.36 Output Compare Output Timing

Input Capture Signal Timing: Figure 9.37 shows input capture signal timing.

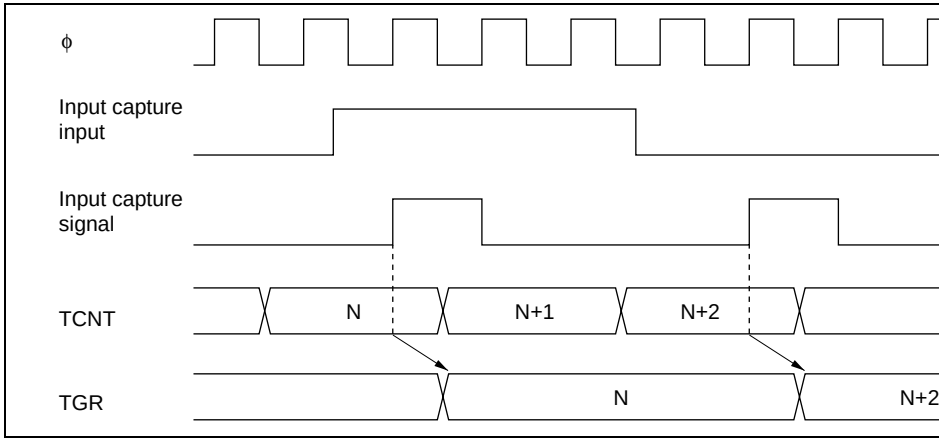


Figure 9.37 Input Capture Input Signal Timing

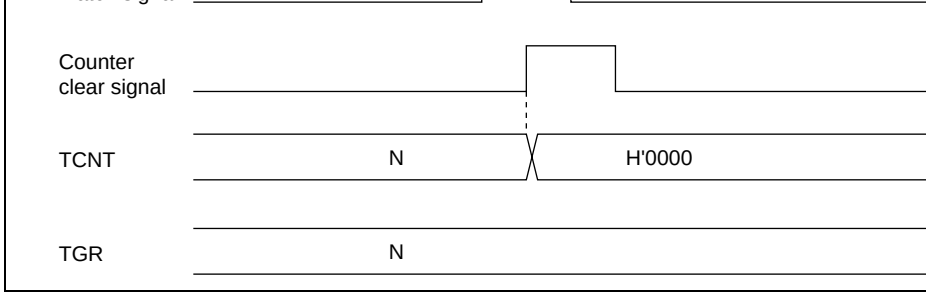


Figure 9.38 Counter Clear Timing (Compare Match)

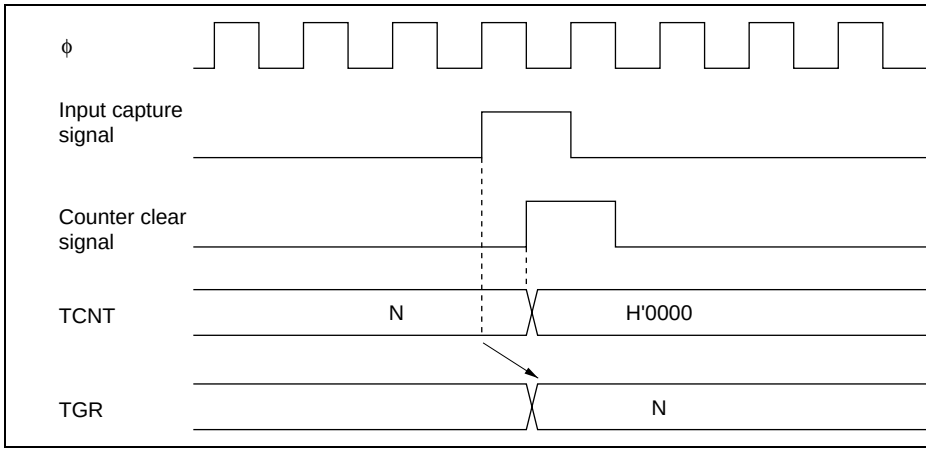


Figure 9.39 Counter Clear Timing (Input Capture)

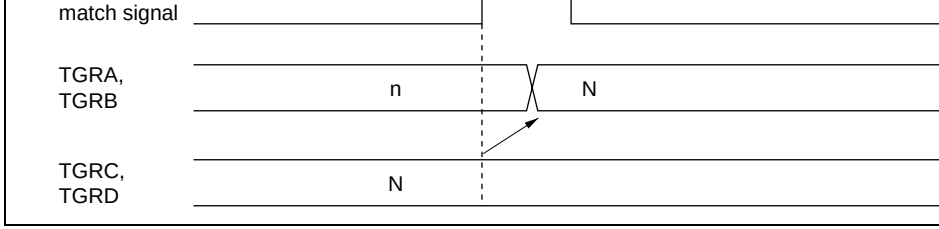


Figure 9.40 Buffer Operation Timing (Compare Match)

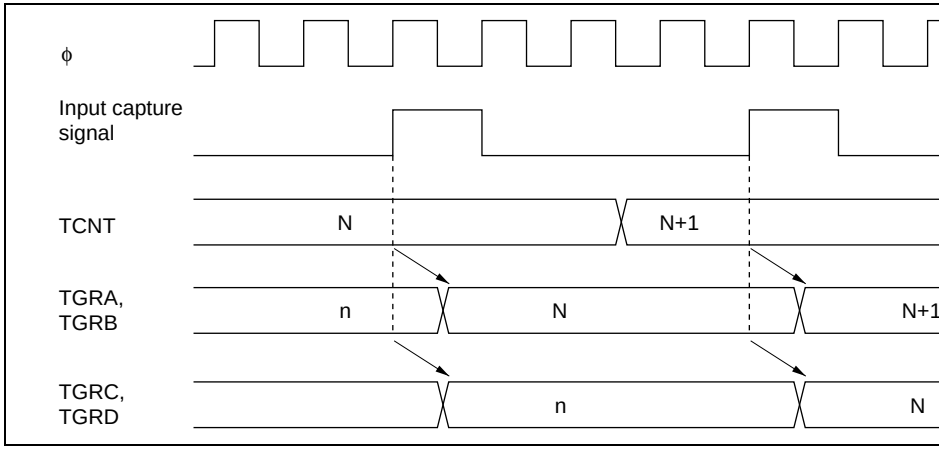


Figure 9.41 Buffer Operation Timing (Input Capture)

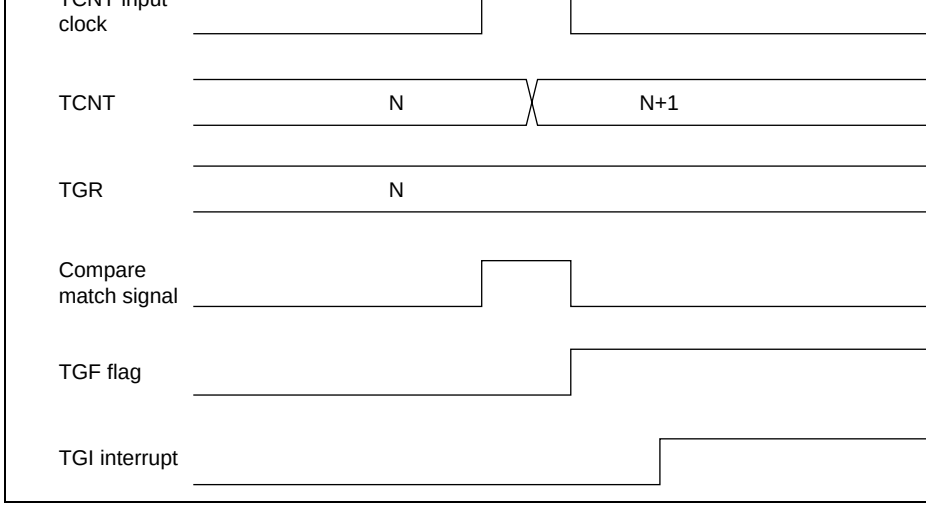


Figure 9.42 TGI Interrupt Timing (Compare Match)

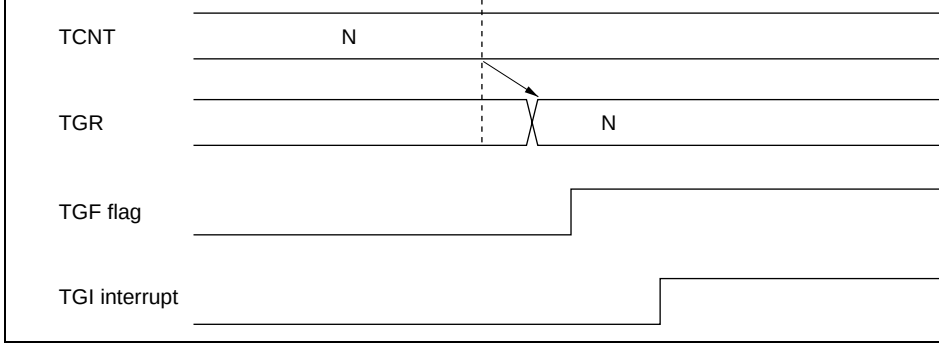


Figure 9.43 TGI Interrupt Timing (Input Capture)

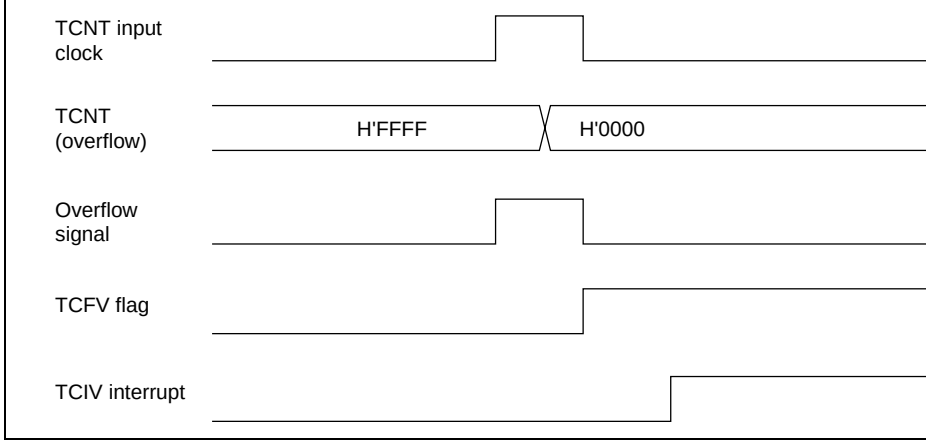


Figure 9.44 TCIV Interrupt Setting Timing

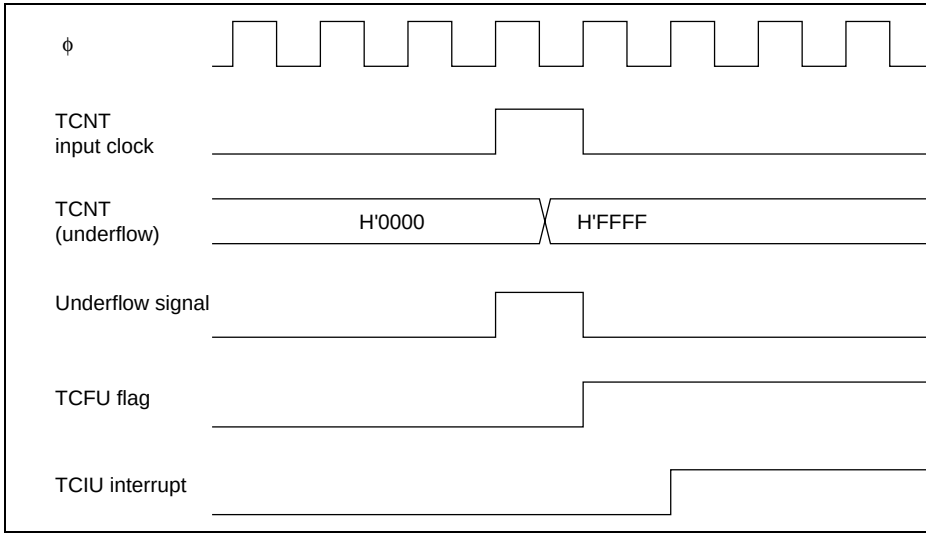


Figure 9.45 TCIU Interrupt Setting Timing

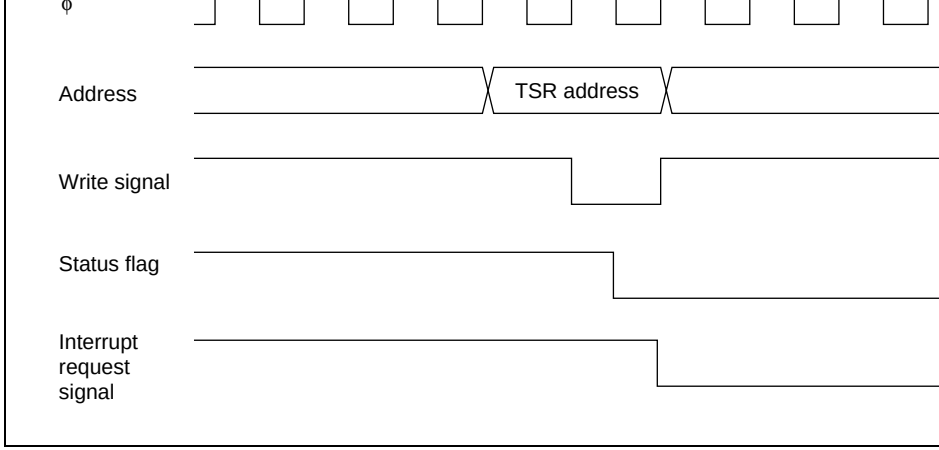


Figure 9.46 Timing for Status Flag Clearing by CPU

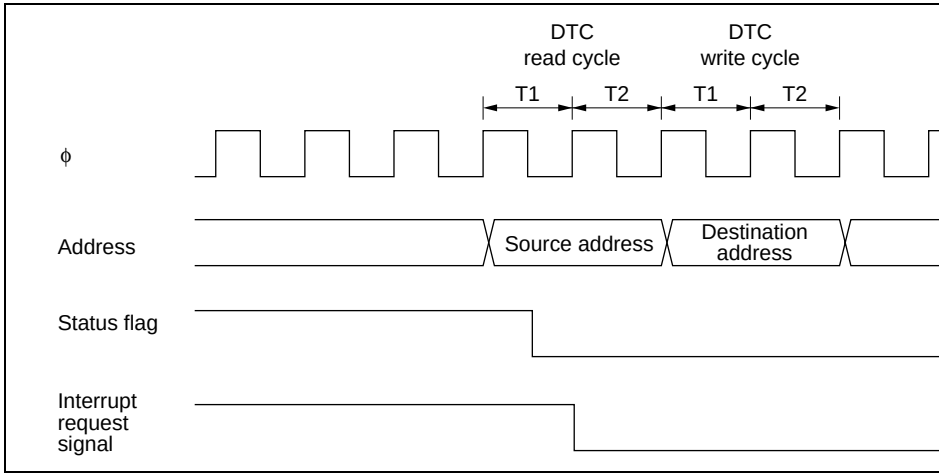


Figure 9.47 Timing for Status Flag Clearing by DTC Activation

least 1.5 states, and the pulse width must be at least 2.5 states. Figure 9.48 shows the i
conditions in phase counting mode.

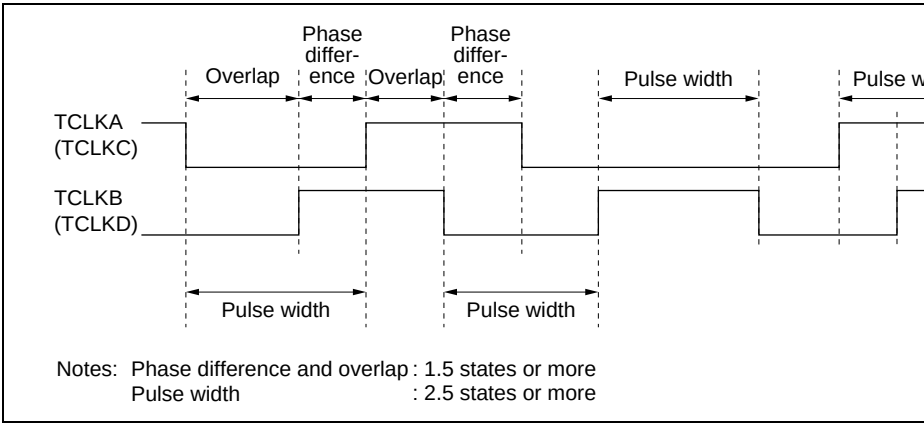


Figure 9.48 Phase Difference, Overlap, and Pulse Width in Phase Counting

Caution on Period Setting: When counter clearing by compare match is set, TCNT i
the final state in which it matches the TGR value (the point at which the count value n
TCNT is updated). Consequently, the actual counter frequency is given by the followi

$$f = \frac{\phi}{(N + 1)}$$

Where f : Counter frequency
ϕ : Operating frequency
N : TGR set value

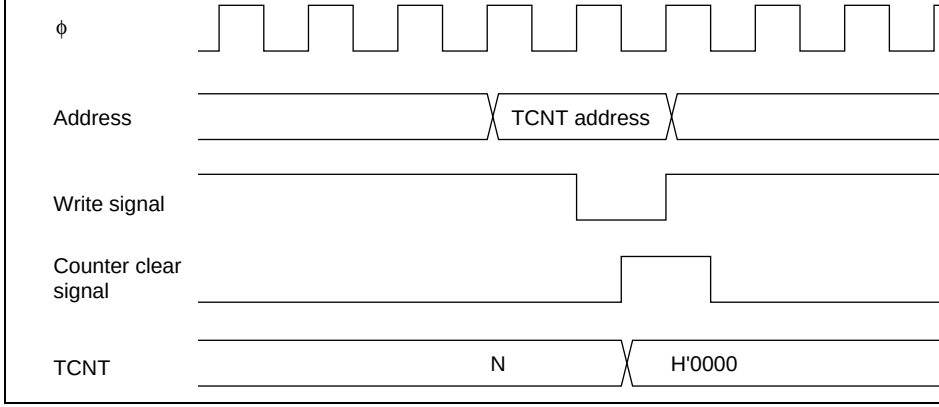


Figure 9.49 Contention between TCNT Write and Clear Operations

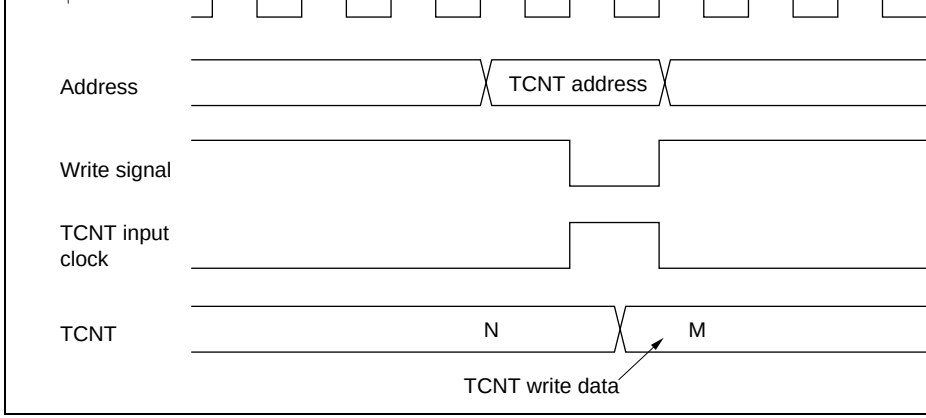


Figure 9.50 Contention between TCNT Write and Increment Operatio

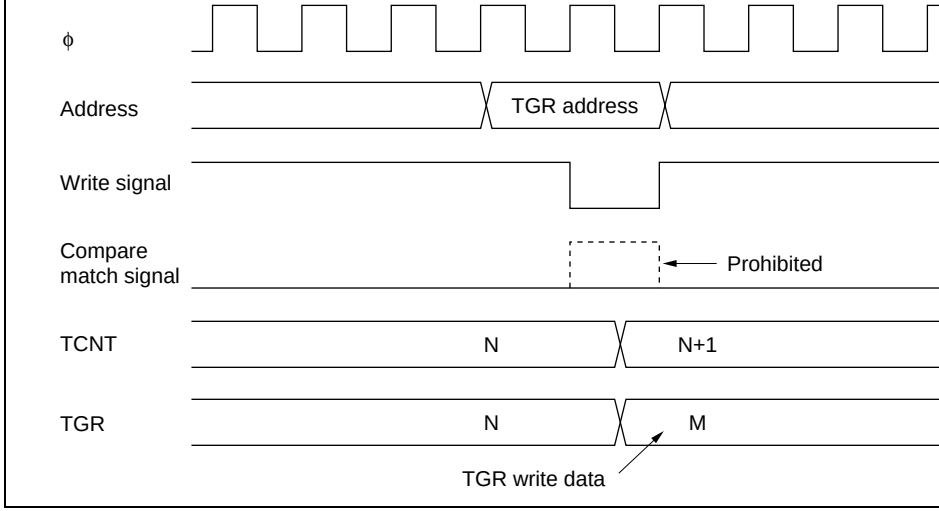


Figure 9.51 Contention between TGR Write and Compare Match

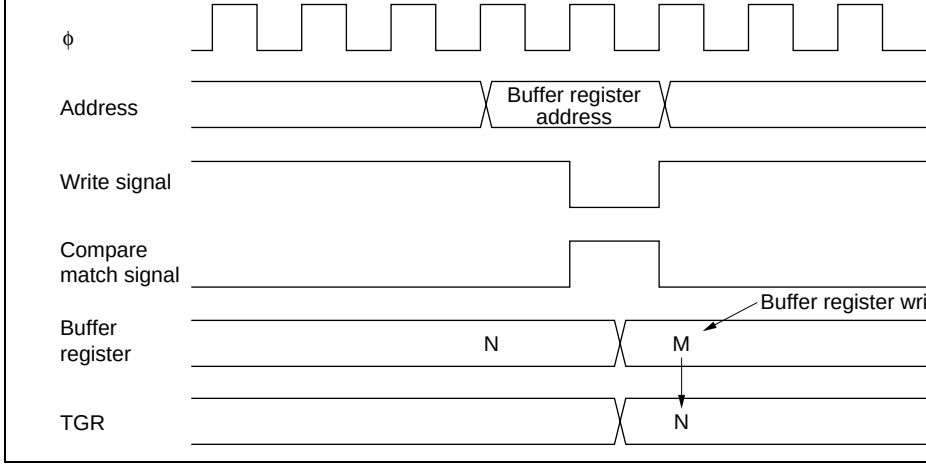


Figure 9.52 Contention between Buffer Register Write and Compare M

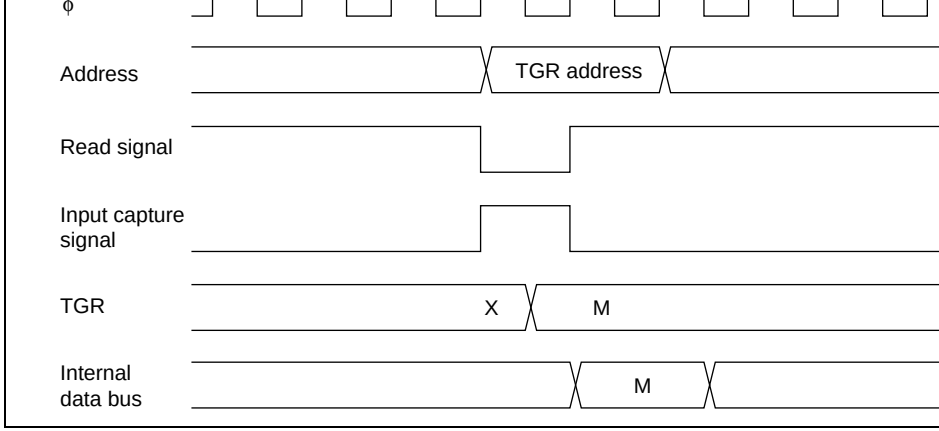


Figure 9.53 Contention between TGR Read and Input Capture

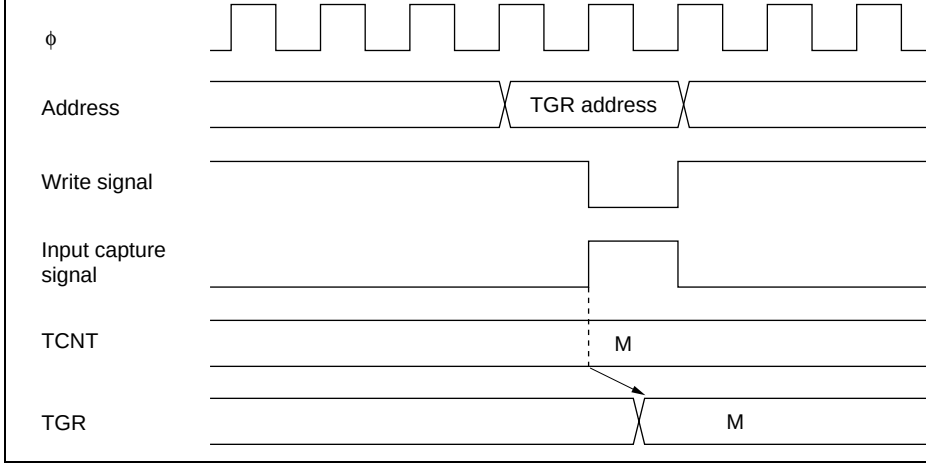


Figure 9.54 Contention between TGR Write and Input Capture

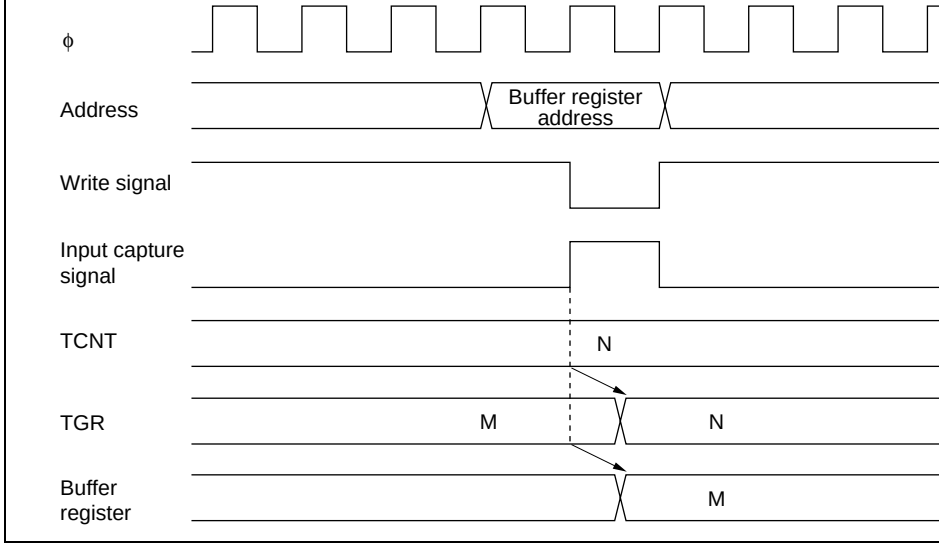


Figure 9.55 Contention between Buffer Register Write and Input Captu

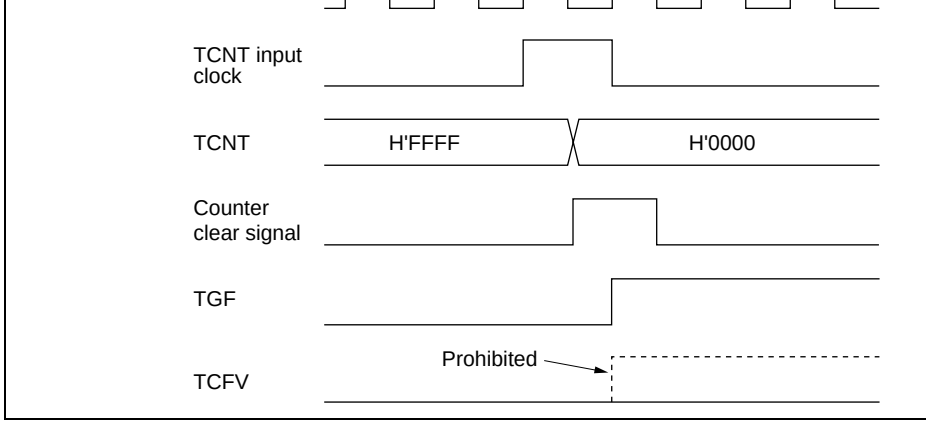


Figure 9.56 Contention between Overflow and Counter Clearing

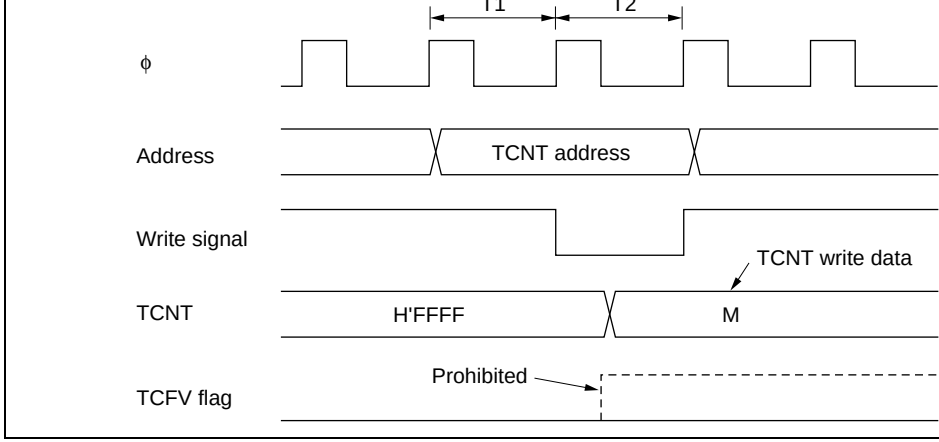


Figure 9.57 Contention between TCNT Write and Overflow

Multiplexing of I/O Pins: In the H8S/2345 Group, the TCLKA input pin is multiplexe TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pi TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an externa 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 interrup requested, it will not be possible to clear the CPU interrupt source or DTC activation so Interrupts should therefore be disabled before entering module stop mode.

timer module can thus be used for a variety of functions, including pulse output with a duty cycle.

10.1.1 Features

The features of the 8-bit timer module are listed below.

- Selection of four clock sources

  The counters can be driven by one of three internal clock signals (ϕ/8, ϕ/64, or ϕ/8 external clock input (enabling use as an external event counter).
- Selection of three ways to clear the counters

  The counters can be cleared on compare match A or B, or by an external reset sign
- Timer output control by a combination of two compare match signals

  The timer output signal in each channel is controlled by a combination of two inde compare match signals, enabling the timer to generate output waveforms with an a cycle or PWM output.
- Provision for cascading of two channels
  - Operation as a 16-bit timer is possible, using channel 0 for the upper 8 bits and for the lower 8 bits (16-bit count mode).
  - Channel 1 can be used to count channel 0 compare matches (compare match co
- Three independent interrupts

  Compare match A and B and overflow interrupts can be requested independently.
- A/D converter conversion start trigger can be generated

  Channel 0 compare match A signal can be used as an A/D converter conversion st
- Module stop mode can be set
  - As the initial setting, 8-bit timer operation is halted. Register access is enabled module stop mode.

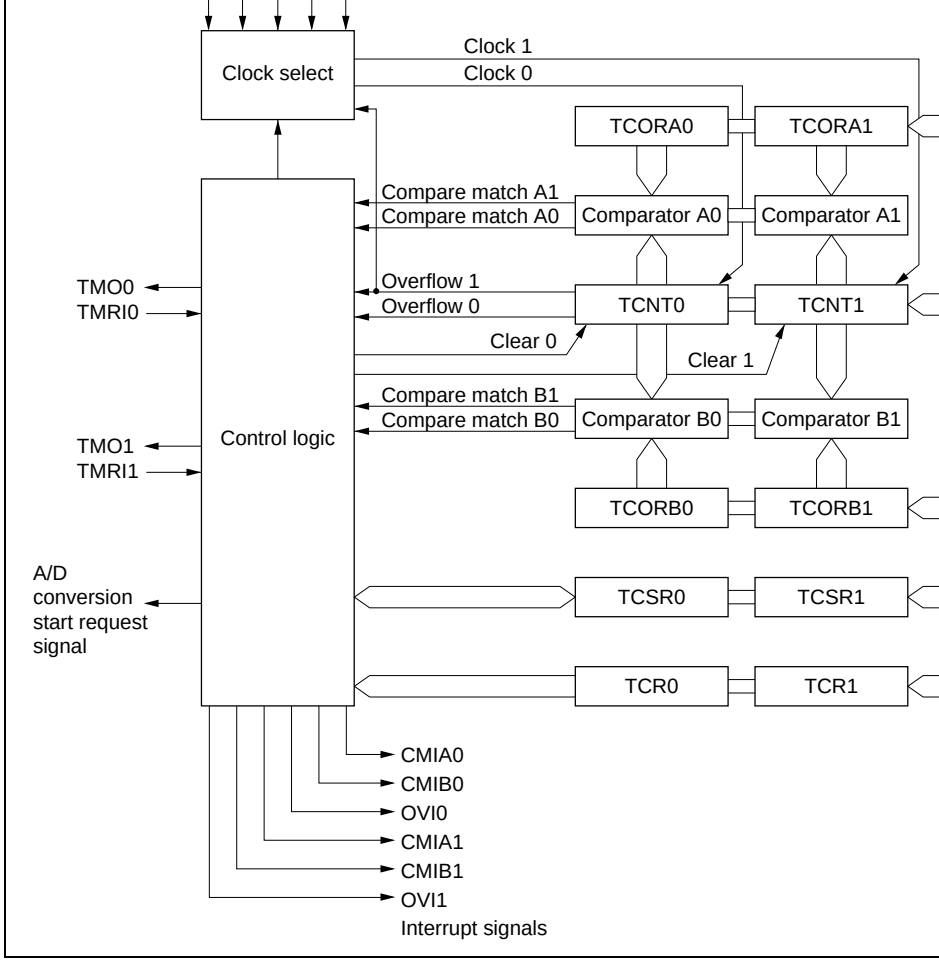


Figure 10.1 Block Diagram of 8-Bit Timer

	Timer reset input pin 0	TMRI0	Input	Inputs external reset to
1	Timer output pin 1	TMO1	Output	Outputs at compare ma
	Timer clock input pin 1	TMCI1	Input	Inputs external clock fo
	Timer reset input pin 1	TMRI1	Input	Inputs external reset to

10.1.4 Register Configuration

Table 10.2 summarizes the registers of the 8-bit timer module.

Table 10.2 8-Bit Timer Registers

Channel	Name	Abbreviation	R/W	Initial value	A
0	Timer control register 0	TCR0	R/W	H'00	H
	Timer control/status register 0	TCSR0	R/(W)*2	H'00	H
	Time constant register A0	TCORA0	R/W	H'FF	H
	Time constant register B0	TCORB0	R/W	H'FF	H
	Timer counter 0	TCNT0	R/W	H'00	H
1	Timer control register 1	TCR1	R/W	H'00	H
	Timer control/status register 1	TCSR1	R/(W)*2	H'10	H
	Time constant register A1	TCORA1	R/W	H'FF	H
	Time constant register B1	TCORB1	R/W	H'FF	H
	Timer counter 1	TCNT1	R/W	H'00	H
All	Module stop control register	MSTPCR	R/W	H'3FFF	H

Notes: 1. Lower 16 bits of the address
2. Only 0 can be written to bits 7 to 5, to clear these flags.

Each pair of registers for channel 0 and channel 1 is a 16-bit register with the upper 8 channel 0 and the lower 8 bits for channel 1, so they can be accessed together by word instruction.

Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT0 and TCNT1 are 8-bit readable/writable up-counters that increment on pulses ge
from an internal or external clock source. This clock source is selected by clock select b
to CKS0 of TCR. The CPU can read or write to TCNT0 and TCNT1 at all times.

TCNT0 and TCNT1 comprise a single 16-bit register, so they can be accessed together
transfer instruction.

TCNT0 and TCNT1 can be cleared by an external reset input or by a compare match si
Which signal is to be used for clearing is selected by clock clear bits CCLR1 and CCLE

When a timer counter overflows from H'FF to H'00, OVF in TCSR is set to 1.

TCNT0 and TCNT1 are each initialized to H'00 by a reset and in hardware standby mo

10.2.2 Time Constant Registers A0 and A1 (TCORA0, TCORA1)

	TCORA0								TCORA1					
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCORA0 and TCORA1 are 8-bit readable/writable registers. TCORA0 and TCORA1
single 16-bit register so they can be accessed together by word transfer instruction.

TCORA is continually compared with the value in TCNT. When a match is detected, th
corresponding CMFA flag of TCSR is set. Note, however, that comparison is disabled
T2 state of a TCOR write cycle.

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

TCORB0 and TCORB1 are 8-bit readable/writable registers. TCORB0 and TCORB1 single 16-bit register so they can be accessed together by word transfer instruction.

TCORB is continually compared with the value in TCNT. When a match is detected, corresponding CMFB flag of TCSR is set. Note, however, that comparison is disable T2 state of a TCOR write cycle.

The timer output can be freely controlled by these compare match signals and the setti output select bits OS3 and OS2 of TCSR.

TCORB0 and TCORB1 are each initialized to H'FF by a reset and in hardware standb

10.2.4 Time Control Registers 0 and 1 (TCR0, TCR1)

Bit	:	7	6	5	4	3	2	1
		CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1
Initial value:		0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCR0 and TCR1 are 8-bit readable/writable registers that select the clock source and t which TCNT is cleared, and enable interrupts.

TCR0 and TCR1 are each initialized to H'00 by a reset and in hardware standby mode

For details of this timing, see section 10.3, Operation.

Bit 6—Compare Match Interrupt Enable A (CMIEA): Selects whether CMFA inter
requests (CMIA) are enabled or disabled when the CMFA flag of TCSR is set to 1.

Bit 6	
CMIEA	**Description**
0	CMFA interrupt requests (CMIA) are disabled
1	CMFA interrupt requests (CMIA) are enabled

Bit 5—Timer Overflow Interrupt Enable (OVIE): Selects whether OVF interrupt re
(OVI) are enabled or disabled when the OVF flag of TCSR is set to 1.

Bit 5	
OVIE	**Description**
0	OVF interrupt requests (OVI) are disabled
1	OVF interrupt requests (OVI) are enabled

Bits 4 and 3—Counter Clear 1 and 0 (CCLR1 and CCLR0): These bits select the m
which TCNT is cleared: by compare match A or B, or by an external reset input.

Bit 4	Bit 3	
CCLR1	**CCLR0**	**Description**
0	0	Clear is disabled
	1	Clear by compare match A
1	0	Clear by compare match B
	1	Clear by rising edge of external reset input

Some functions differ between channel 0 and channel 1.

Bit 2	Bit 1	Bit 0	
CKS2	**CKS1**	**CKS0**	**Description**
0	0	0	Clock input disabled
		1	Internal clock, counted at falling edge of ϕ/8
	1	0	Internal clock, counted at falling edge of ϕ/64
		1	Internal clock, counted at falling edge of ϕ/8192
1	0	0	For channel 0: count at TCNT1 overflow signal*
			For channel 1: count at TCNT0 compare match A*
		1	External clock, counted at rising edge
	1	0	External clock, counted at falling edge
		1	External clock, counted at both rising and falling edges

Note: * If the count input of channel 0 is the TCNT1 overflow signal and that of cha
TCNT0 compare match signal, no incrementing clock is generated. Do not
setting.

TCSR1

Bit	:	7	6	5	4	3	2	1
		CMFB	CMFA	OVF	—	OS3	OS2	OS1
Initial value	:	0	0	0	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 bits 7 to 5, to clear these flags.

TCSR0 and TCSR1 are 8-bit registers that display compare match and overflow statuse control compare match output.

TCSR0 is initialized to H'00, and TCSR1 to H'10, by a reset and in hardware standby m

Bit 7—Compare Match Flag B (CMFB): Status flag indicating whether the values of TCORB match.

Bit 7

CMFB	Description
0	[Clearing conditions] (• Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When DTC is activated by CMIB interrupt while DISEL bit of MRB in DT
1	[Setting condition] Set when TCNT matches TCORB

1	[Setting condition] Set when TCNT matches TCORA

Bit 5—Timer Overflow Flag (OVF): Status flag indicating that TCNT has overflowe
from H'FF to H'00).

Bit 5

OVF	Description
0	[Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 to OVF
1	[Setting condition] Set when TCNT overflows from H'FF to H'00

Bit 4—A/D Trigger Enable (ADTE) (TCSR0 Only): Selects enabling or disabling o
converter start requests by compare-match A.

In TCSR1, this bit is reserved: it is always read as 1 and cannot be modified.

Bit 4

ADTE	Description
0	A/D converter start requests by compare match A are disabled
1	A/D converter start requests by compare match A are enabled

priority.

Timer output is disabled when bits OS3 to OS0 are all 0.

After a reset, the timer output is 0 until the first compare match event occurs.

Bit 3	Bit 2	
OS3	**OS2**	**Description**
0	0	No change when compare match B occurs
	1	0 is output when compare match B occurs
1	0	1 is output when compare match B occurs
	1	Output is inverted when compare match B occurs (toggle output)

Bit 1	Bit 0	
OS1	**OS0**	**Description**
0	0	No change when compare match A occurs
	1	0 is output when compare match A occurs
1	0	1 is output when compare match A occurs
	1	Output is inverted when compare match A occurs (toggle output)

MSTPCR is a 16-bit readable/writable register that performs module stop mode contro

When the MSTP12 bit in MSTPCR is set to 1, the 8-bit timer operation stops at the en cycle and a transition is made to module stop mode. Registers cannot be read or writte module stop mode. For details, see section 19.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not in software standby mode.

Bit 12—Module Stop (MSTP12): Specifies the 8-bit timer stop mode.

Bit 12	
MSTP12	**Description**
0	8-bit timer module stop mode cleared
1	8-bit timer module stop mode set

count timing.

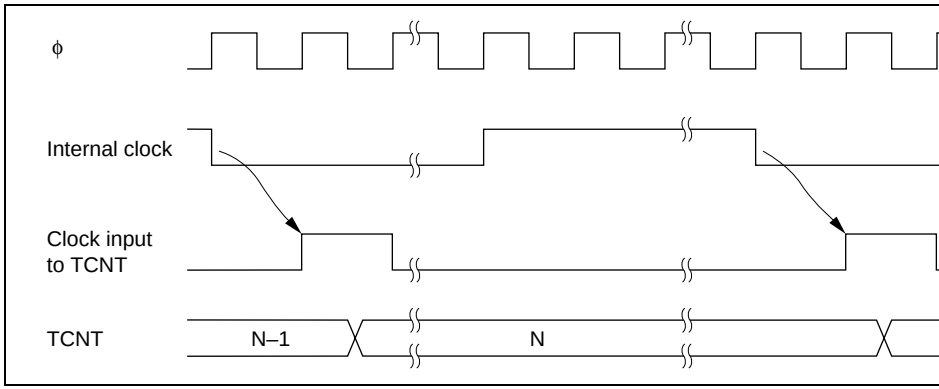


Figure 10.2 Count Timing for Internal Clock Input

External Clock: Three incrementation methods can be selected by setting bits CKS2 to
TCR: at the rising edge, the falling edge, and both rising and falling edges.

Note that the external clock pulse width must be at least 1.5 states for incrementation at
edge, and at least 2.5 states for incrementation at both edges. The counter will not incr
correctly if the pulse width is less than these values.

Figure 10.3 shows the timing of incrementation at both edges of an external clock signa

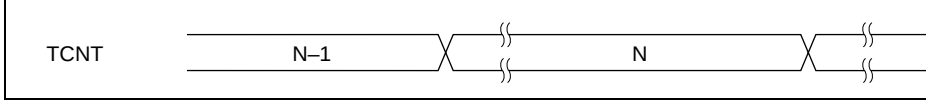


Figure 10.3 Count Timing for External Clock Input

10.3.2 Compare Match Timing

Setting of Compare Match Flags A and B (CMFA, CMFB): The CMFA and CMF
TCSR are set to 1 by a compare match signal generated when the TCOR and TCNT v
The compare match signal is generated at the last state in which the match is true, just
timer counter is updated.

Therefore, when TCOR and TCNT match, the compare match signal is not generated
next incrementation clock input. Figure 10.4 shows this timing.

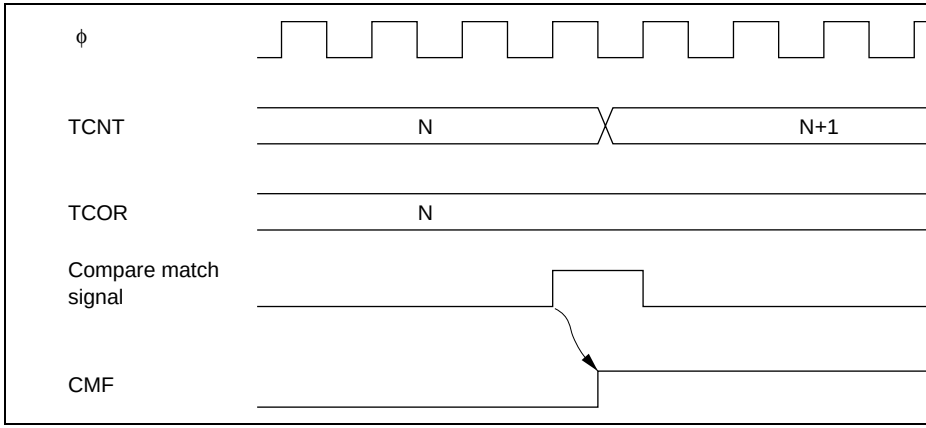


Figure 10.4 Timing of CMF Setting

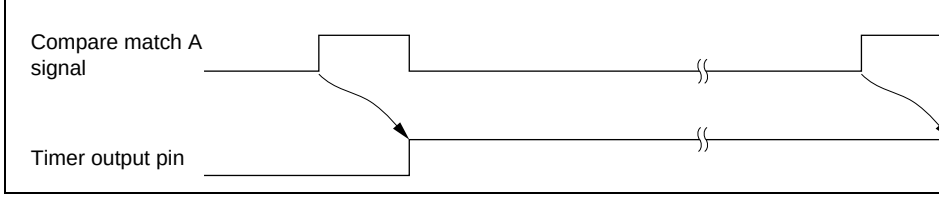


Figure 10.5 Timing of Timer Output

Timing of Compare Match Clear: The timer counter is cleared when compare match occurs, depending on the setting of the CCLR1 and CCLR0 bits in TCR. Figure 10.6 sh timing of this operation.

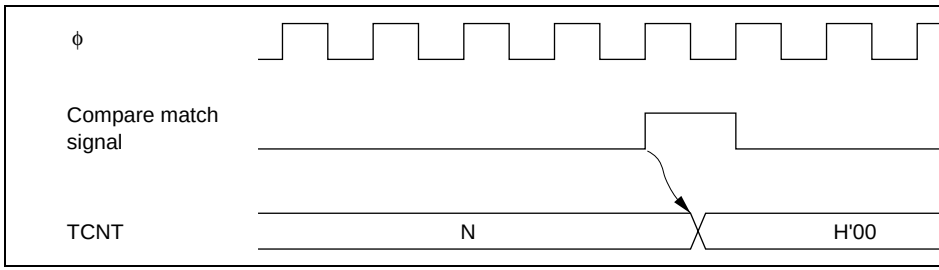


Figure 10.6 Timing of Compare Match Clear

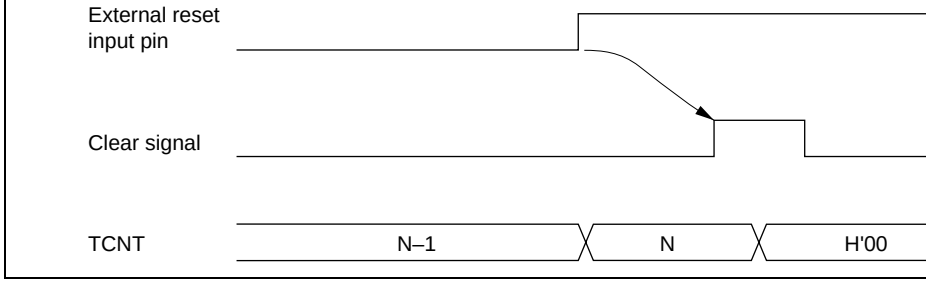


Figure 10.7 Timing of External Reset

10.3.4 Timing of Overflow Flag (OVF) Setting

The OVF in TCSR is set to 1 when the timer count overflows (changes from H'FF to H
10.8 shows the timing of this operation.

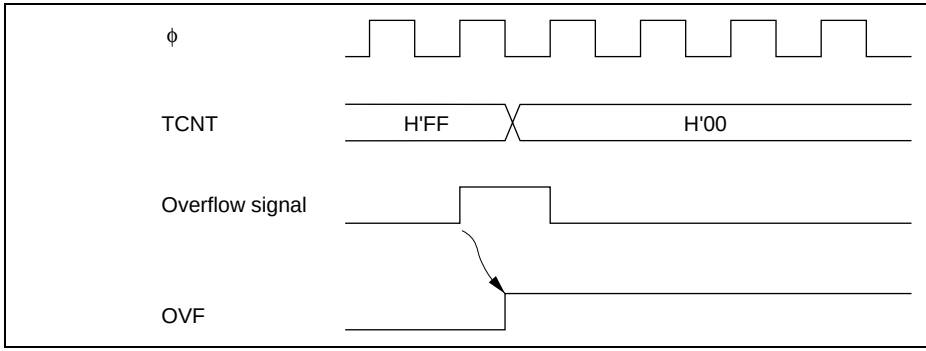


Figure 10.8 Timing of OVF Setting

lower 8 bits.

- Setting of compare match flags
  - The CMF flag in TCSR0 is set to 1 when a 16-bit compare match event occurs.
  - The CMF flag in TCSR1 is set to 1 when a lower 8-bit compare match event oc

- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR0 have been set for counter clear at comp the 16-bit counter (TCNT0 and TCNT1 together) is cleared when a 16-bit comp event occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even clear by the TMRI0 pin has also been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR1 are ignored. The lower 8 b be cleared independently.

- Pin output
  - Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR0 is in accord the 16-bit compare match conditions.
  - Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR1 is in accord the lower 8-bit compare match conditions.

Compare Match Counter Mode: When bits CKS2 to CKS0 in TCR1 are B'100, TCN compare match A's for channel 0.

Channels 0 and 1 are controlled independently. Conditions such as setting of the CMF generation of interrupts, output from the TMO pin, and counter clear are in accordance settings for each channel.

Note on Usage: If the 16-bit counter mode and compare match counter mode are set simultaneously, the input clock pulses for TCNT0 and TCNT1 are not generated and th counters will stop operating. Software should therefore avoid using both these modes.

Table 10.3 8-Bit Timer Interrupt Sources

Interrupt Source	Description	DTC Activation	P
CMIA0	Interrupt by CMFA	Possible	H
CMIB0	Interrupt by CMFB	Possible	
OVI0	Interrupt by OVF	Not possible	
CMIA1	Interrupt by CMFA	Possible	
CMIB1	Interrupt by CMFB	Possible	
OVI1	Interrupt by OVF	Not possible	L

Note: This table shows the initial state immediately after a reset. The relative channel can be changed by the interrupt controller.

10.4.2 A/D Converter Activation

The A/D converter can be activated only by channel 0 compare match A.

If the ADTE bit in TCSR0 is set to 1 when the CMFA flag is set to 1 by the occurrenc 0 compare match A, a request to start A/D conversion is sent to the A/D converter. If t timer conversion start trigger has been selected on the A/D converter side at this time, conversion is started.

compare match and to 0 at a TCORB compare match.

With these settings, the 8-bit timer provides output of pulses at a rate determined by TC
a pulse width determined by TCORB. No software intervention is required.

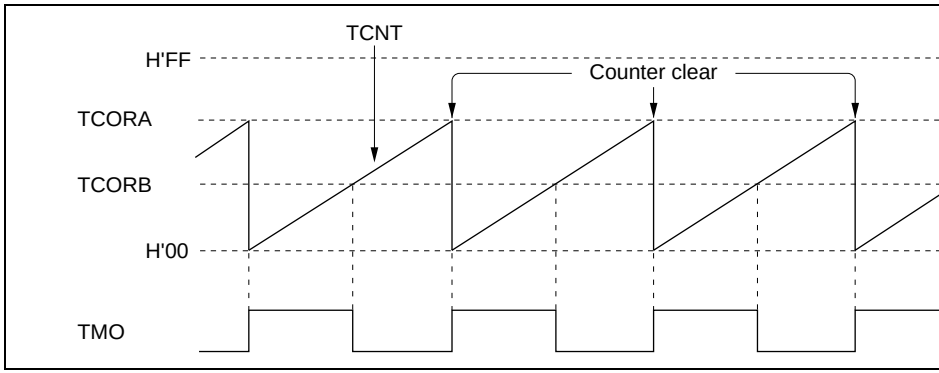


Figure 10.9 Example of Pulse Output

takes priority, so that the counter is cleared and the write is not performed.

Figure 10.10 shows this operation.

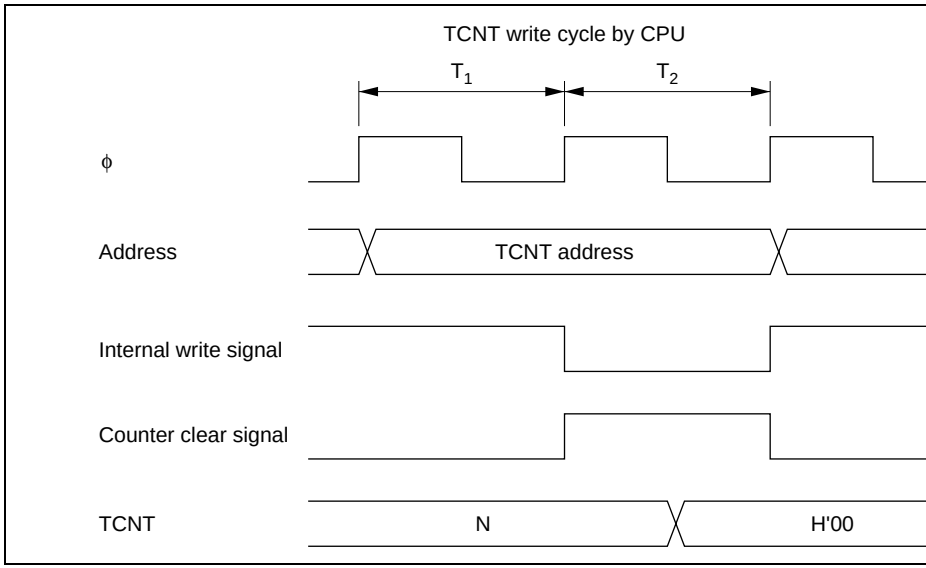


Figure 10.10 Contention between TCNT Write and Clear

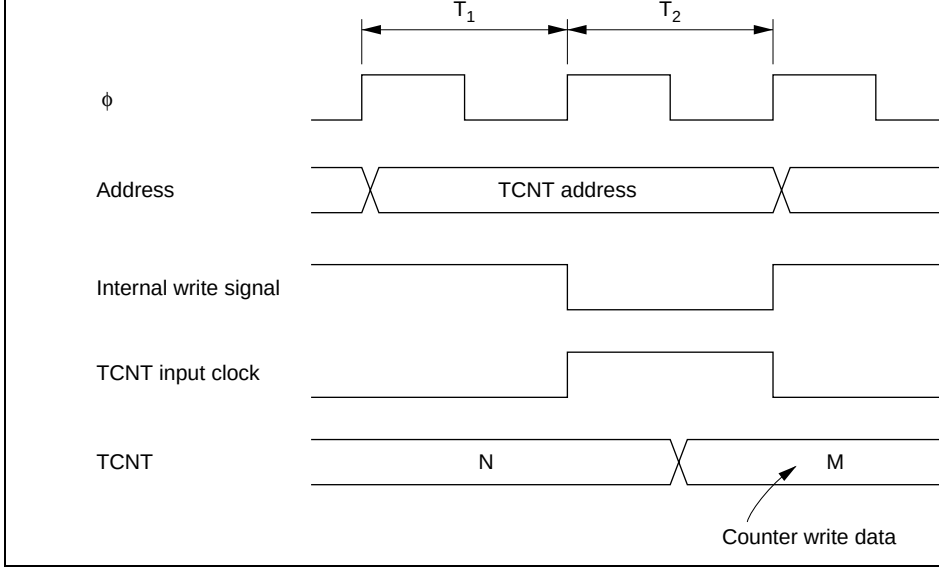


Figure 10.11 Contention between TCNT Write and Increment

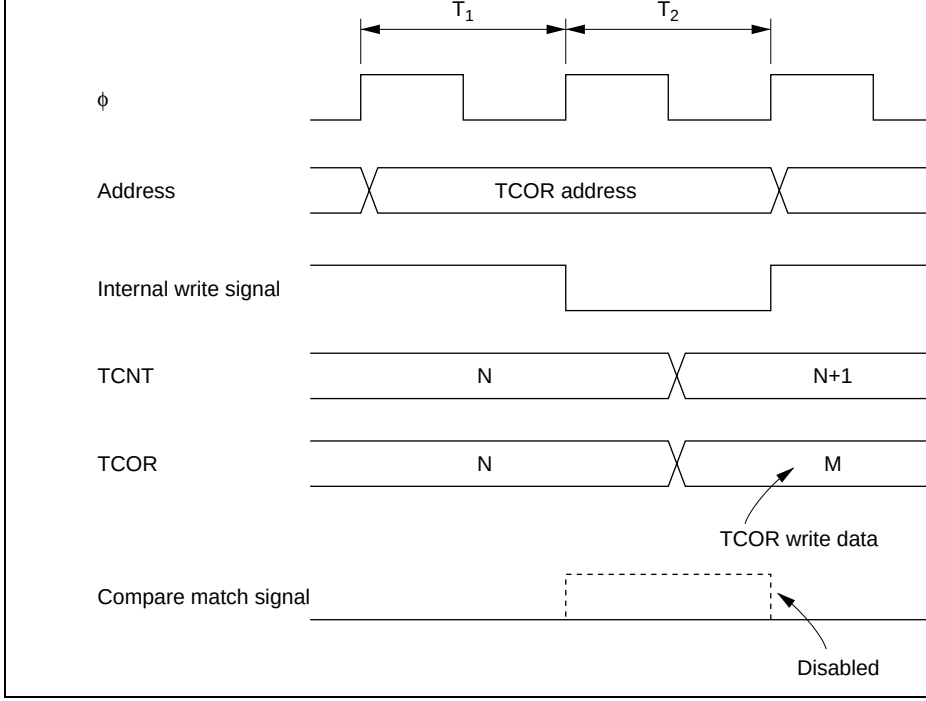


Figure 10.12 Contention between TCOR Write and Compare Match

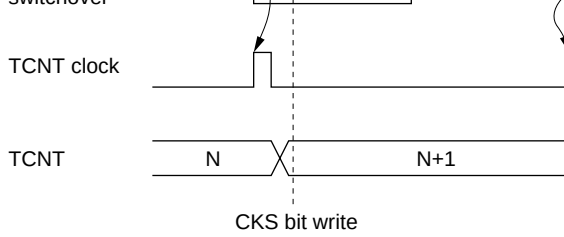
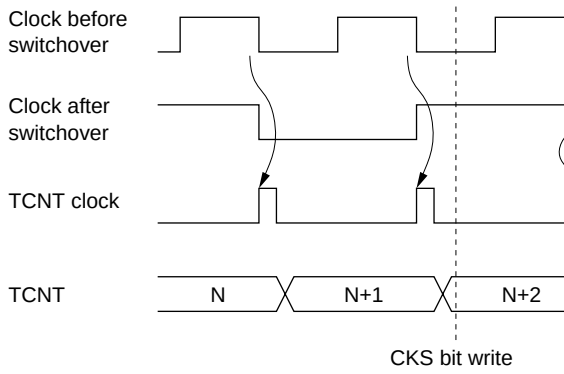
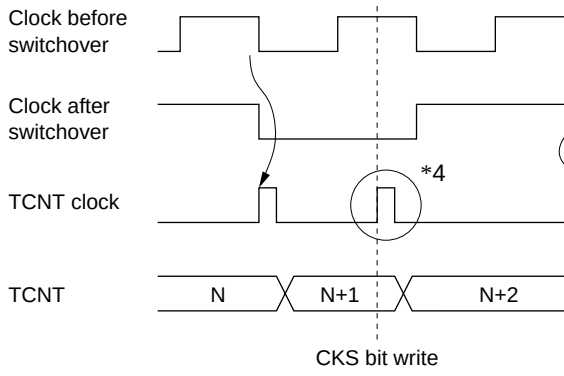
Toggle output	High
1 output	
0 output	
No change	Low

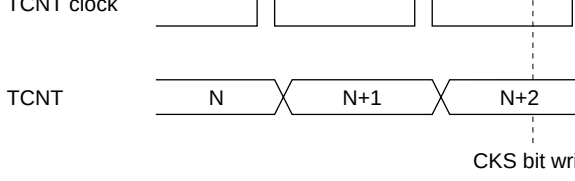
10.6.5 Switching of Internal Clocks and TCNT Operation

TCNT may increment erroneously when the internal clock is switched over. Table 10.5 relationship between the timing at which the internal clock is switched (by writing to th and CKS0 bits) and the TCNT operation.

When the TCNT clock is generated from an internal clock, the falling edge of the inter pulse is detected. If clock switching causes a change from high to low level, as shown table 10.5, a TCNT clock pulse is generated on the assumption that the switchover is a edge. This increments TCNT.

The erroneous incrementation can also happen when switching between internal and ex clocks.

		
2	Switching from low to high*2	
3	Switching from high to low*3	



Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

10.6.6 Usage Note

Interrupts and Module Stop Mode: If module stop mode is entered when an interrupt requested, it will not be possible to clear the CPU interrupt source or DTC activation so Interrupts should therefore be disabled before entering module stop mode.

generate an internal reset signal for the H8S/2345 Group.

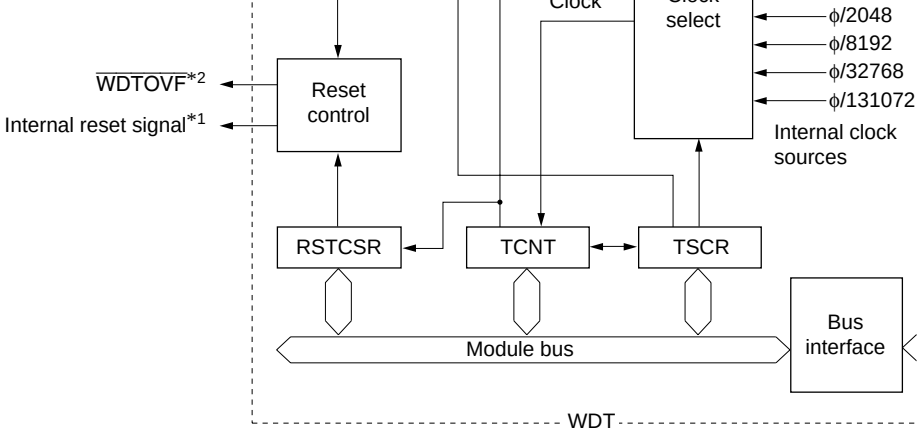
When this watchdog function is not needed, the WDT can be used as an interval timer
timer operation, an interval timer interrupt is generated each time the counter overflow

11.1.1 Features

WDT features are listed below.

- Switchable between watchdog timer mode and interval timer mode
- $\overline{\text{WDTOVF}}$ output* when in watchdog timer mode
  If the counter overflows, the WDT outputs $\overline{\text{WDTOVF}}$.* It is possible to select whe
  the entire H8S/2345 Group is reset at the same time. This internal reset can be a p
  reset or a manual reset.
- Interrupt generation when in interval timer mode
  If the counter overflows, the WDT generates an interval timer interrupt.
- Choice of eight counter clock sources.

Note: * The $\overline{\text{WDTOVF}}$ pin function is not supported by the F-ZTAT version.



Legend:
TCSR : Timer control/status register
TCNT : Timer counter
RSTCSR : Reset control/status register

Notes: 1. The type of internal reset signal depends on a register setting. Either power-on reset
reset can be selected.
2. The $\overline{\text{WDTOVF}}$ pin function is not supported by the F-ZTAT version.

Figure 11.1 Block Diagram of WDT

Note: * The $\overline{\text{WDTOVF}}$ pin function is not supported by the F-ZTAT version.

11.1.4 Register Configuration

The WDT has three registers, as summarized in table 11.2. These registers control clo
WDT mode switching, and the reset signal.

Table 11.2 WDT Registers

				Add
Name	**Abbreviation**	**R/W**	**Initial Value**	**Write**[*2]
Timer control/status register	TCSR	R/(W)[*3]	H'18	H'FFBC
Timer counter	TCNT	R/W	H'00	H'FFBC
Reset control/status register	RSTCSR	R/(W)[*3]	H'1F	H'FFBE

Notes: 1. Lower 16 bits of the address.
2. For details of write operations, see section 11.2.4, Notes on Register Acces
3. Only a write of 0 is permitted to bit 7, to clear the flag.

R/W . R/W R/W R/W R/W R/W R/W R/W

TCNT is an 8-bit readable/writable*[1] up-counter.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from the clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (chan H'FF to H'00), either the watchdog timer overflow signal ($\overline{\text{WDTOVF}}$)*[2] or an interval t interrupt (WOVI) is generated, depending on the mode selected by the WT/$\overline{\text{IT}}$ bit in TC

TCNT is initialized to H'00 by a reset, in hardware standby mode, or when the TME bit to 0. It is not initialized in software standby mode.

Note: 1. The method for writing to TCNT is different from that for general registers inadvertent overwriting. For details see section 11.2.4, Notes on Register A
2. The $\overline{\text{WDTOVF}}$ pin function is not supported by the F-ZTAT version.

Note: * Can only be written with 0 for flag clearing.

TCSR is an 8-bit readable/writable* register. Its functions include selecting the clock s
input to TCNT, and the timer mode.

TCR is initialized to H'18 by a reset and in hardware standby mode. It is not initialized
standby mode.

Note: * The method for writing to TCSR is different from that for general registers
inadvertent overwriting. For details see section 11.2.4, Notes on Register A

Bit 7—Overflow Flag (OVF): Indicates that TCNT has overflowed from H'FF to H'0
interval timer mode. This flag cannot be set during watchdog timer operation.

Bit 7

OVF	Description
0	[Clearing condition]
	Cleared by reading TCSR when OVF = 1*, then writing 0 to OVF
1	[Setting condition]
	Set when TCNT overflows (changes from H'FF to H'00) in interval timer m

Note: * When OVF is polled and the interval timer interrupt is disabled. OVF = 1 mu
at least twice.

Bit 6—Timer Mode Select (WT/$\overline{\text{IT}}$): Selects whether the WDT is used as a watchdo
interval timer. If used as an interval timer, the WDT generates an interval timer interru
(WOVI) when TCNT overflows. If used as a watchdog timer, the WDT generates the
signal* when TCNT overflows.

Note: * The $\overline{\text{WDTOVF}}$ pin function is not supported by the F-ZTAT version.

Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted.

Bit 5	
TME	**Description**
0	TCNT is initialized to H'00 and halted
1	TCNT counts

Bits 4 and 3—Reserved: Read-only bits, always read as 1.

Bits 2 to 0: Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight internal sources, obtained by dividing the system clock (ϕ), for input to TCNT.

Bit 2	Bit 1	Bit 0	Description	
CKS2	**CKS1**	**CKS0**	**Clock**	**Overflow Period (when ϕ = 20 MHz)**
0	0	0	ϕ/2 (initial value)	25.6 µs
		1	ϕ/64	819.2 µs
	1	0	ϕ/128	1.6 ms
		1	ϕ/512	6.6 ms
1	0	0	ϕ/2048	26.2 ms
		1	ϕ/8192	104.9 ms
	1	0	ϕ/32768	419.4 ms
		1	ϕ/131072	1.68 s

Note: * The overflow period is the time from when TCNT starts counting up from H'0 overflow occurs.

Note: * Can only be written with 0 for flag clearing.

RSTCSR is an 8-bit readable/writable* register that controls the generation of the inte
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 WDT
reset signal caused by overflows.

Note: * The method for writing to RSTCSR is different from that for general regist
prevent inadvertent overwriting. For details see section 11.2.4, Notes on R
Access.

Bit 7—Watchdog Overflow Flag (WOVF): Indicates that TCNT has overflowed (ch
H'FF to H'00) during watchdog timer operation. This bit is not set in interval timer mo

Bit 7

WOVF	Description
0	[Clearing condition] Cleared by reading RSTCSR when WOVF = 1, then writing 0 to WOVF
1	[Setting condition] Set when TCNT overflows (changed from H'FF to H'00) during watchdog t operation

WDT are reset.

Bit 5—Reset Select (RSTS): Selects the type of internal reset generated if TCNT over during watchdog timer operation.

For details of the types of resets, see section 4, Exception Handling.

Bit 5

RSTS	Description
0	Power-on reset (
1	Manual reset

Bits 4 to 0—Reserved: Read-only bits, always read as 1.

11.2.4 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers more difficult to write to. The procedures for writing to and reading these registers are below.

Writing to TCNT and TCSR: These registers must be written to by a word transfer in They cannot be written to with byte instructions.

Figure 11.2 shows the format of data written to TCNT and TCSR. TCNT and TCSR bo same write address. For a write to TCNT, the upper byte of the written word must cont and the lower byte must contain the write data. For a write to TCSR, the upper byte of t word must contain H'A5 and the lower byte must contain the write data. This transfers data from the lower byte to TCNT or TCSR.

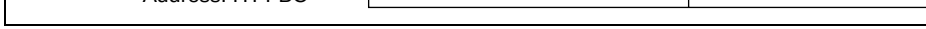


Figure 11.2 Format of Data Written to TCNT and TCSR

Writing to RSTCSR: RSTCSR must be written to by word transfer instruction to add H'FFBE. 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 bit differs from that for writing to the RSTE and RSTS bits.

To write 0 to the WOVF bit, the write data must have H'A5 in the upper byte and H'00 lower byte. This clears the WOVF bit to 0, but has no effect on the RSTE and RSTS b to the RSTE and RSTS bits, the upper byte must contain H'5A and the lower byte mus write data. This writes the values in bits 6 and 5 of the lower byte into the RSTE and but has no effect on the WOVF bit.

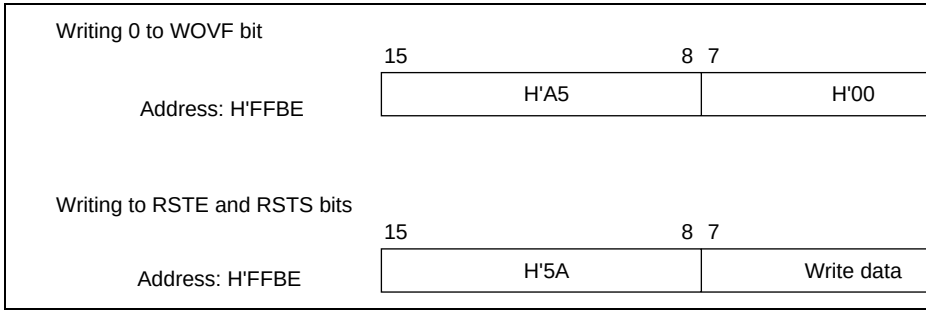


Figure 11.3 Format of Data Written to RSTCSR

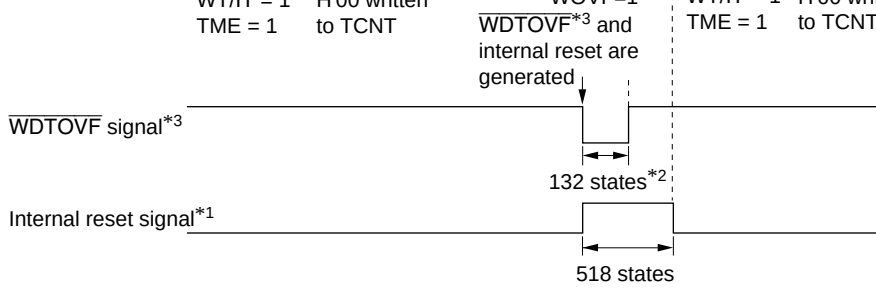
Reading TCNT, TCSR, and RSTCSR: These registers are read in the same way as o registers. The read addresses are H'FFBC for TCSR, H'FFBD for TCNT, and H'FFBF RSTCSR.

signal* is output. This is shown in figure 11.4. This WDTOVF signal* can be used to r
system. The $\overline{\text{WDTOVF}}$ signal* is output for 132 states when RSTE = 1, and for 130 sta
RSTE = 0.

If TCNT overflows when 1 is set in the RSTE bit in RSTCSR, a signal that resets the H
Group internally is generated at the same time as the $\overline{\text{WDTOVF}}$ signal*. This reset can
as a power-on reset or a manual reset, depending on the setting of the RSTS bit in RST
internal reset signal is output for 518 states.

If a reset caused by a signal input to the $\overline{\text{RES}}$ pin occurs at the same time as a reset caus
WDT overflow, the $\overline{\text{RES}}$ pin reset has priority and the WOVF bit in RSTCSR is cleared

Note: * The $\overline{\text{WDTOVF}}$ pin function is not supported by the F-ZTAT version.



Legend:
WT/$\overline{IT}$: Timer mode select bit
TME : Timer enable bit

Notes: 1. The internal reset signal is generated only if the RSTE bit is set to 1.
2. 130 states when the RSTE bit is cleared to 0.
3. The $\overline{WDTOVF}$ pin function is not supported by the F-ZTAT version.

Figure 11.4 Watchdog Timer Operation

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 TM
An interval timer interrupt (WOVI) is generated each time TCNT overflows, provided
WDT is operating as an interval timer, as shown in figure 11.5. This function can be u
generate interrupt requests at regular intervals.

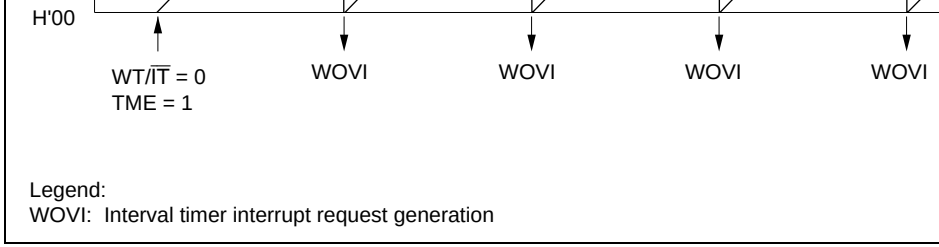


Figure 11.5 Interval Timer Operation

11.3.3 Timing of Setting Overflow Flag (OVF)

The OVF flag is set to 1 if TCNT overflows during interval timer operation. At the sam
interval timer interrupt (WOVI) is requested. This timing is shown in figure 11.6.

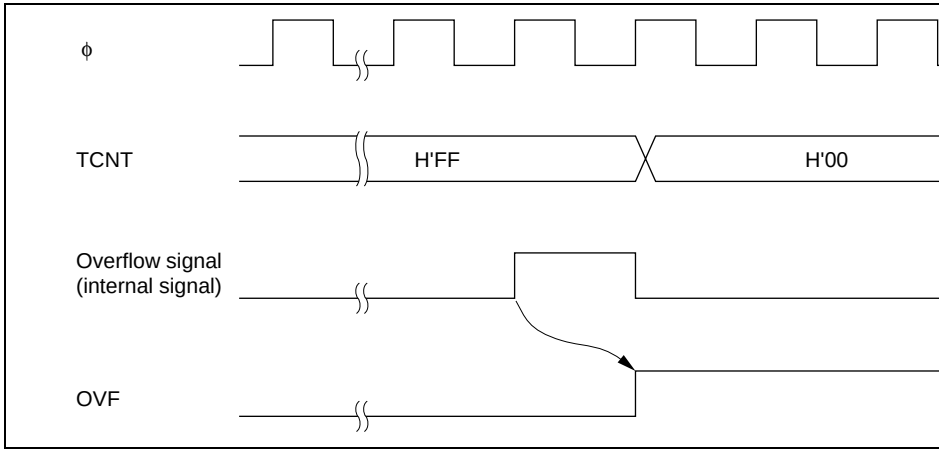
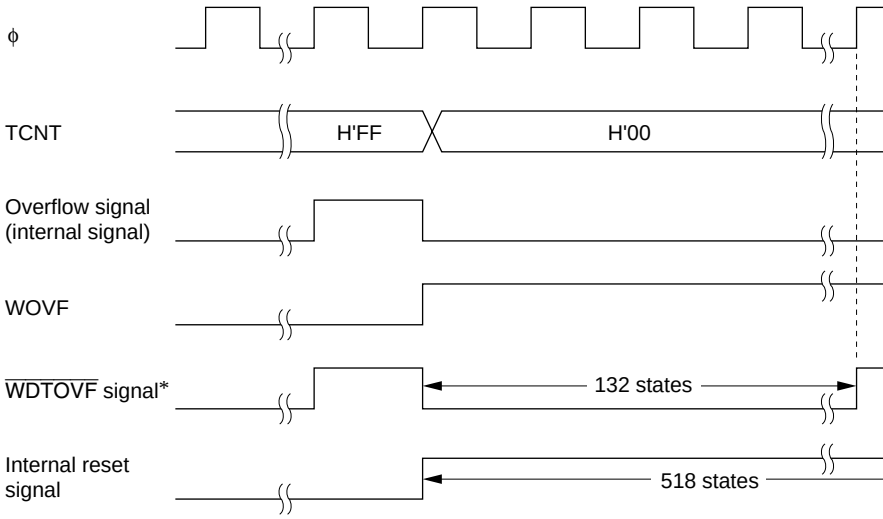


Figure 11.6 Timing of Setting of OVF



Note: * The WDTOVF pin function is not supported by the F-ZTAT version.

Figure 11.7 Timing of Setting of WOVF

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, th
takes priority and the timer counter is not incremented. Figure 11.8 shows this operatio

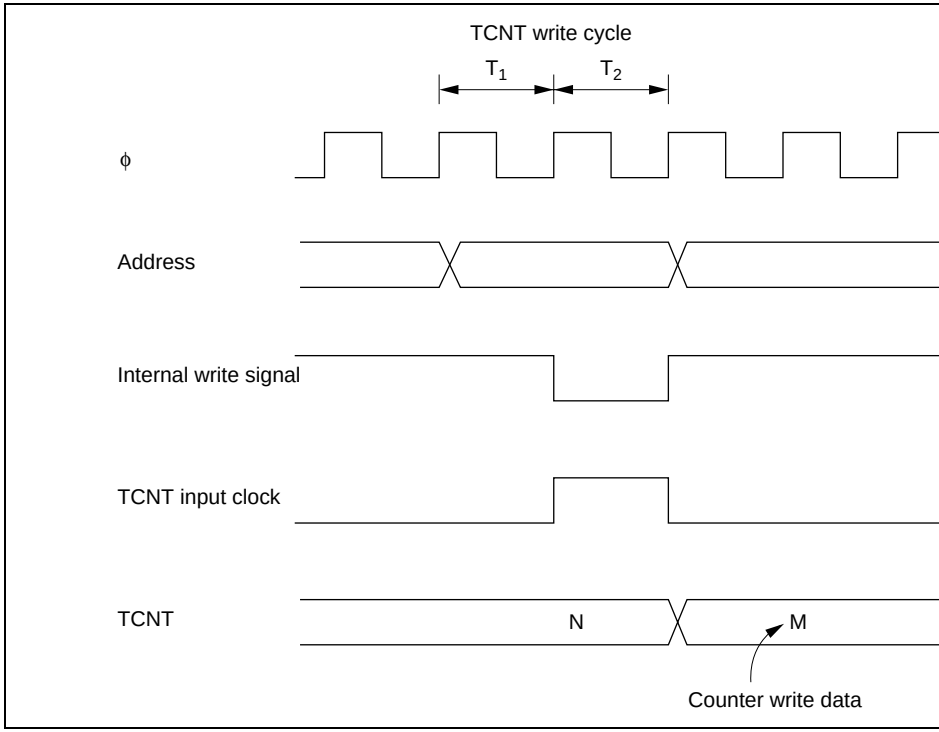


Figure 11.8 Contention between TCNT Write and Increment

If the mode is switched from watchdog timer to interval timer, or vice versa, while the operating, errors could occur in the incrementation. Software must stop the watchdog clearing the TME bit to 0) before switching the mode.

11.5.4 System Reset by $\overline{\text{WDTOVF}}$ Signal

If the $\overline{\text{WDTOVF}}$ output signal* is input to the $\overline{\text{RES}}$ pin of the H8S/2345 Group, the H Group will not be initialized correctly. Make sure that the $\overline{\text{WDTOVF}}$ signal* is not inp to the $\overline{\text{RES}}$ pin. To reset the entire system by means of the $\overline{\text{WDTOVF}}$ signal*, use the shown in figure 11.9.

Note: * The $\overline{\text{WDTOVF}}$ pin function is not supported by the F-ZTAT version.

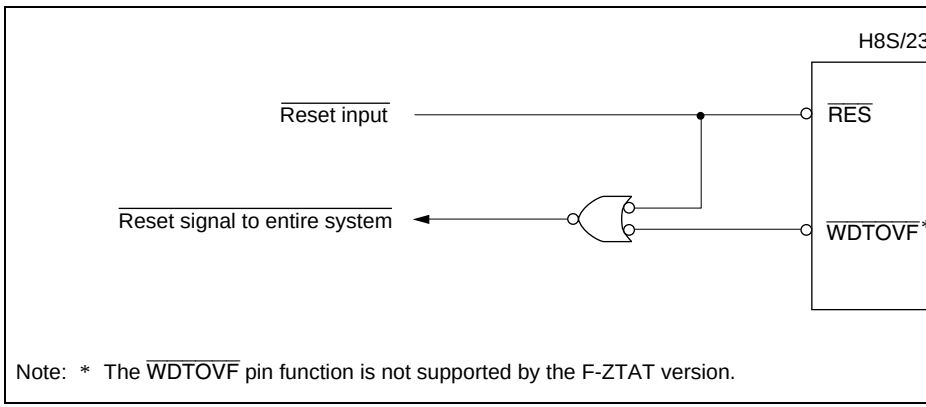


Figure 11.9 Circuit for System Reset by $\overline{\text{WDTOVF}}$ Signal (Example)

Note: * The $\overline{\text{WDTOVF}}$ pin function is not supported by the F-ZTAT version.

processors (multiprocessor communication function).

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 sync is achieved character by character

  — Serial data communication can be carried out with standard asynchronous com chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asy 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 case of a framing error

  Clocked Synchronous mode

  — Serial data communication synchronized with a clock

  Serial data communication can be carried out with other chips that have a sync communication function

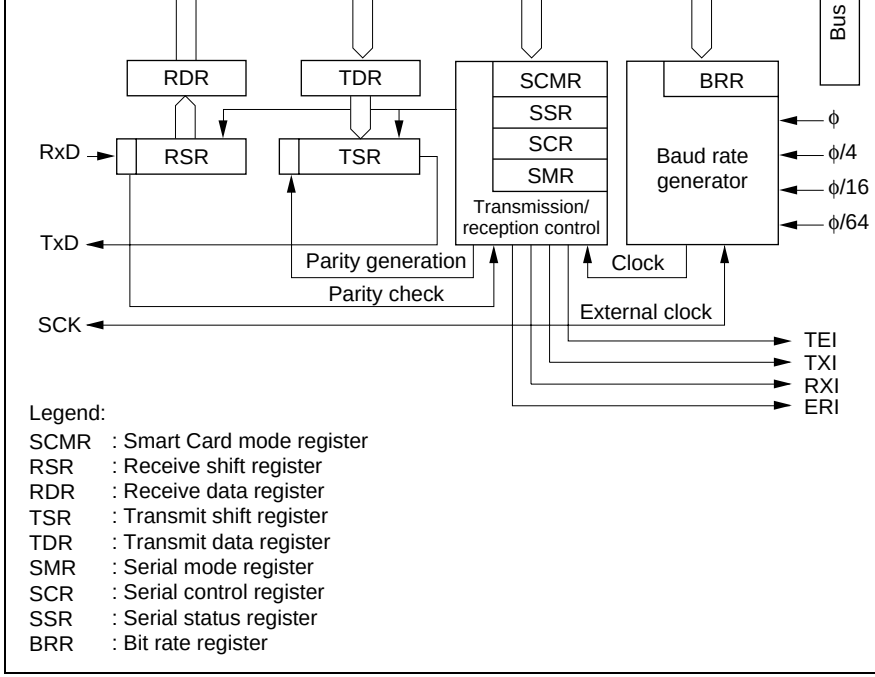
  — One serial data transfer format

Data length	:	8 bits

  — Receive error detection : Overrun errors detected

- Four interrupt sources
  - Four interrupt sources — transmit-data-empty, transmit-end, receive-data-full, a error — that can issue requests independently
  - The transmit-data-empty interrupt and receive data full interrupts can activate th transfer controller (DTC) to execute data transfer
- Choice of LSB-first or MSB-first transfer
  - Can be selected regardless of the communication mode* (except in the case of asynchronous mode bit data)
- Module stop mode can be set
  - As the initial setting, SCI operation is halted. Register access is enabled by exiti stop mode.

Note: * Descriptions in this section refer to LSB-first transfer.



Legend:

SCMR	: Smart Card mode register
RSR	: Receive shift register
RDR	: Receive data register
TSR	: Transmit shift register
TDR	: Transmit data register
SMR	: Serial mode register
SCR	: Serial control register
SSR	: Serial status register
BRR	: Bit rate register

Figure 12.1 Block Diagram of SCI

	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

0	Serial mode register 0	SMR0	R/W	H'00	H
	Bit rate register 0	BRR0	R/W	H'FF	H
	Serial control register 0	SCR0	R/W	H'00	H
	Transmit data register 0	TDR0	R/W	H'FF	H
	Serial status register 0	SSR0	R/(W)*2	H'84	H
	Receive data register 0	RDR0	R	H'00	H
	Smart card mode register 0	SCMR0	R/W	H'F2	H
1	Serial mode register 1	SMR1	R/W	H'00	H
	Bit rate register 1	BRR1	R/W	H'FF	H
	Serial control register 1	SCR1	R/W	H'00	H
	Transmit data register 1	TDR1	R/W	H'FF	H
	Serial status register 1	SSR1	R/(W)*2	H'84	H
	Receive data register 1	RDR1	R	H'00	H
	Smart card mode register 1	SCMR1	R/W	H'F2	H
All	Module stop control register	MSTPCR	R/W	H'3FFF	H

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

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 w
LSB (bit 0), and converts it to parallel data. When one byte of data has been received, i
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
Initial value	:	0	0	0	0	0	0	0
R/W	:	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 f
RDR where it is stored, and completes the receive operation. After this, RSR is receive

Since RSR and RDR function as a double buffer in this way, enables continuous receiv
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, and in standby mode or module stop mode.

To perform serial data transmission, the SCI first transfers transmit data from TDR to
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
TSR, and transmission started, automatically. However, data transfer from TDR to TS
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
Initial value	:	1	1	1	1	1	1	1
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
starts serial transmission. Continuous serial transmission can be carried out by writing
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, and in standby mode or module stop mode.

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 standby mode or module stop mode.

Bit 7—Communication Mode (C/$\overline{A}$): Selects asynchronous mode or clocked synchro
as the SCI operating mode.

Bit 7

C/$\overline{A}$	Description	
0	Asynchronous mode	(
1	Clocked synchronous mode	

Bit 6—Character Length (CHR): Selects 7 or 8 bits as the data length in asynchronou
clocked synchronous mode, a fixed data length of 8 bits is used regardless of the CHR s

Bit 6

CHR	Description	
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
possible to choose between LSB-first or MSB-first transfer.

Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit transmit data before transmission. In reception, the parity bit is checked for (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 checking.

The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit additio checking, in asynchronous mode. The O/$\overline{E}$ bit setting is invalid in clocked synchronou when parity addition and checking is disabled in asynchronous mode.

Bit 4

O/$\overline{E}$	Description
0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is performed in transmission so t 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 character plus the parity bit is even.
2. When odd parity is set, parity bit addition is performed in transmission so th 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 character plus the parity bit is odd.

Bit 3—Stop Bit Length (STOP): Selects 1 or 2 bits as the stop bit length in asynchro The STOP bits setting is only valid in asynchronous mode. If clocked synchronous mo STOP bit setting is invalid since stop bits are not added.

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 tra
character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiproces
is selected, the PE bit and O/$\overline{E}$ bit parity settings are invalid. The MP bit setting is only
asynchronous mode; it is invalid in clocked synchronous mode.

For details of the multiprocessor communication function, see section 12.3.3, Multipro
Communication Function.

Bit 2

MP	Description	
0	Multiprocessor function disabled	(
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source
baud rate generator. The clock source can be selected from ϕ, ϕ/4, ϕ/16, and ϕ/64, acco
setting of bits CKS1 and CKS0.

For the relation between the clock source, the bit rate register setting, and the baud rate
section 12.2.8, Bit Rate Register (BRR).

Bit 1	Bit 0		
CKS1	CKS0	Description	
0	0	ϕ clock	(
	1	ϕ/4 clock	
1	0	ϕ/16 clock	
	1	ϕ/64 clock	

SCR is a register that performs enabling or disabling of SCI transfer operations, serial 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 standby mode or module stop mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit data empty i (TXI) request generation when serial transmit data is transferred from TDR to TSR an flag in SSR is set to 1.

Bit 7

TIE	Description
0	Transmit data empty interrupt (TXI) requests disabled*
1	Transmit data empty interrupt (TXI) requests enabled

Note: * TXI interrupt request cancellation can be performed by reading 1 from the T then clearing it to 0, or clearing the TIE bit to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables receive data full interru request and receive error interrupt (ERI) request generation when serial receive data is 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) r disabled*
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) r enabled

Note: * RXI and ERI interrupt request cancellation can be performed by reading 1 f RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or c RIE bit to 0.

TDRE flag in SSR is cleared to 0.

SMR setting must be performed to decide the transfer format before setting t
to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of serial reception by the SC

Bit 4

RE	Description
0	Reception disabled*1 (I
1	Reception enabled*2

Notes: 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER fla
retain their states.

2. Serial reception is started in this state when a start bit is detected in asynchr
mode or serial clock input is detected in clocked synchronous mode.

SMR setting must be performed to decide the transfer format before setting t
to 1.

[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, ar of the RDRF, FER, and ORER flags in SSR are disabled until data with th multiprocessor bit set to 1 is received.

Note: * When receive data including MPB = 0 is received, receive data transfer fro RDR, receive error detection, and setting of the RDRF, FER, and ORER fla is not performed. When receive data including MPB = 1 is received, the MP is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and OR setting is enabled.

Bit 2—Transmit End Interrupt Enable (TEIE): Enables or disables transmit end in (TEI) request generation when there is no valid transmit data in TDR in MSB data tra

Bit 2

TEIE	Description
0	Transmit end interrupt (TEI) request disabled*
1	Transmit end interrupt (TEI) request enabled*

Note: * TEI cancellation can be performed by reading 1 from the TDRE flag in SSR clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0

SMR before setting the CKE1 and CKE0 bits.

For details of clock source selection, see table 12.9.

Bit 1	Bit 0		
CKE1	**CKE0**	**Description**	
0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O p
		Clocked synchronous mode	Internal clock/SCK pin functions as seria output
	1	Asynchronous mode	Internal clock/SCK pin functions as cloc
		Clocked synchronous mode	Internal clock/SCK pin functions as seria output
1	0	Asynchronous mode	External clock/SCK pin functions as cloc
		Clocked synchronous mode	External clock/SCK pin functions as seri input
	1	Asynchronous mode	External clock/SCK pin functions as cloc
		Clocked synchronous mode	External clock/SCK pin functions as seri 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.

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 multiprocessor bits.

SSR can be read or written to by the CPU at all times. However, 1 cannot be written t TDRE, RDRF, ORER, PER, and FER. Also note that in order to clear these flags they read as 1 beforehand. The TEND flag and MPB flag are read-only flags and cannot be

SSR is initialized to H'84 by a reset, and in standby mode or module stop mode.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that data has been transfe 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] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to

	When serial reception ends normally and receive data is transferred from R

Note: RDR and the RDRF flag are not affected and retain their previous values when a 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 error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that an overrun error occurred during recep causing abnormal termination.

Bit 5

ORER	Description
0	[Clearing condition] (In
	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 cleared to 0.

2. The receive data prior to the overrun error is retained in RDR, and the data r subsequently is lost. Also, subsequent serial reception cannot be continued ORER flag is set to 1. In clocked synchronous mode, serial transmission can continued, either.

	When the SCI checks whether the stop bit at the end of the receive data w 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 cleared to 0.

2. In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the sec is not checked. If a framing error occurs, the receive data is transferred to R RDRF flag is not set. Also, subsequent serial reception cannot be continue FER flag is set to 1. In clocked synchronous mode, serial transmission can continued, either.

Bit 3—Parity Error (PER): Indicates that a parity error occurred during reception us addition in asynchronous mode, causing abnormal termination.

Bit 3

PER	Description
0	[Clearing condition] (I
	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 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 cleared to 0.

2. If a parity error occurs, the receive data is transferred to RDR but the RDRF set. Also, subsequent serial reception cannot be continued while the PER f 1. In clocked synchronous mode, serial transmission cannot be continued,

	• 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] (
	• When the TE bit in SCR is 0
	• When TDRE = 1 at transmission of the last bit of a 1-byte serial transmi

Bit 1—Multiprocessor Bit (MPB): When reception is performed using multiprocesso 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] (In
	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 multipr format.

0	Data with a 0 multiprocessor bit is transmitted
1	Data with a 1 multiprocessor bit is transmitted

12.2.8 Bit Rate Register (BRR)

Bit	:	7	6	5	4	3	2	1
Initial value	:	1	1	1	1	1	1	1
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 bau 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 standby mode or module stop mode.

As baud rate generator control is performed independently for each channel, different be set for each channel.

Table 12.3 shows sample BRR settings in asynchronous mode, and table 12.4 shows s settings in clocked synchronous mode.

600	0	103	0.16	0	108	0.21	0	127	0.00	0	155
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19
9600	0	6	—	0	6	–2.48	0	7	0.00	0	9
19200	0	2	—	0	2	—	0	3	0.00	0	4
31250	0	1	0.00	0	1	—	0	1	—	0	2
38400	0	1	—	0	1	—	0	1	0.00	—	—

	ϕ = 3.6864 MHz			**ϕ = 4 MHz**			**ϕ = 4.9152 MHz**			**ϕ =**	
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15
19200	0	5	0.00	0	6	—	0	7	0.00	0	7
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4
38400	0	2	0.00	0	2	—	0	3	0.00	0	3

1200	0	155	0.16	0	159	0.00	0	191	0.00	0	20
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	10
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12
31250	0	5	0.00	0	5	2.40	0	6	—	0	7
38400	0	4	–2.34	0	4	0.00	0	5	0.00	0	6

	ϕ = 9.8304 MHz			**ϕ = 10 MHz**			**ϕ = 12 MHz**			**ϕ = 1**	
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	21
150	2	127	0.00	2	129	0.16	2	155	0.16	2	15
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79
600	1	127	0.00	1	129	0.16	1	155	0.16	1	15
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	15
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9

1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16
38400	0	10	—	0	11	0.00	0	12	0.16	0	13

	ϕ = 18 MHz			**ϕ = 19.6608 MHz**			**ϕ = 20 MHz**		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	3	79	–0.12	3	86	0.31	3	88	–0.25
150	2	233	0.16	2	255	0.00	3	64	0.16
300	2	116	0.16	2	127	0.00	2	129	0.16
600	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	17	0.00	0	19	–1.70	0	19	0.00
38400	0	14	–2.34	0	15	0.00	0	15	1.73

Note: Settings with an error of 1% or less are recommended.

Legend:

—: Setting possible, but error occurs

2.5 k	0	199	1	99	1	199	1	249	2	99	2
5 k	0	99	0	199	1	99	1	124	1	199	1
10 k	0	49	0	99	0	199	0	249	1	99	1
25 k	0	19	0	39	0	79	0	99	0	159	0
50 k	0	9	0	19	0	39	0	49	0	79	0
100 k	0	4	0	9	0	19	0	24	0	39	0
250 k	0	1	0	3	0	7	0	9	0	15	0
500 k	0	0*	0	1	0	3	0	4	0	7	0
1 M			0	0*	0	1	—	—	0	3	0
2.5 M					—	—	0	0*	—	—	0
5 M									—	—	0

Legend:

Blank : Cannot be set.

— : Can be set, but there will be a degree of error.

* : Continuous transfer is not possible.

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ϕ: 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	ϕ	0	0
1	$\phi/4$	0	1
2	$\phi/16$	1	0
3	$\phi/64$	1	1

The bit rate error in asynchronous mode is found from the following formula:

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

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
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0

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
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500

12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3

mode 7-bit data, LSB-first or MSB-first can be selected regardless of the serial commu
mode. The descriptions in this chapter refer to LSB-first transfer.

For details of the other bits in SCMR, see section 13.2.1, Smart Card Mode Register (S

SCMR is initialized to H'F2 by a reset, and in standby mode or module stop 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 conv
format.

The transmit/receive format is valid for 8-bit data.

Bit 3

SDIR	Description
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 2—Smart Card Data Invert (SINV): When the smart card interface operates as a
SCI, 0 should be written in this bit.

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

Bit 0—Smart Card Interface Mode Select (SMIF): When the smart card interface op
normal SCI, 0 should be written in this bit.

MSTPCR is a 16-bit readable/writable register that performs module stop mode contro

When the corresponding bit of bit MSTP6 or MSTP5 is set to 1, SCI operation stops a the bus cycle and a transition is made to module stop mode. Registers cannot be read o in module stop mode. For details, see section 19.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not in software standby mode.

Bit 6—Module Stop (MSTP6): Specifies the SCI channel 1 module stop mode.

Bit 6	
MSTP6	**Description**
0	SCI channel 1 module stop mode cleared
1	SCI channel 1 module stop mode set

Bit 5—Module Stop (MSTP5): Specifies the SCI channel 0 module stop mode.

Bit 5	
MSTP5	**Description**
0	SCI channel 0 module stop mode cleared
1	SCI channel 0 module stop mode set

Selection of asynchronous or clocked synchronous mode and the transmission format is using SMR as shown in table 12.8. The SCI clock is determined by a combination of th 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 bi combination of these parameters determines the transfer format and character lengt
- 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 fre 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 bau 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 of
  - When external clock is selected:

    The on-chip baud rate generator is not used, and the SCI operates on the input s

			1	0				Yes
				1				
	1		0	0		7-bit data		No
				1				
			1	0				Yes
				1				
	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Yes	No
			—	1				
	1		—	0		7-bit data		
			—	1				
1	—	—	—	—	Clocked synchronous mode	8-bit data	No	

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 Source**	**SCK Pin Function**
C/$\overline{A}$	**CKE1**	**CKE0**	**Mode**		
0	0	0	Asynchronous mode	Internal	SCI does not use SCK pin
		1			Outputs clock with same freq rate
	1	0		External	Inputs clock with frequency o the bit rate
		1			
1	0	0	Clocked synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

that data can be read or written during transmission or reception, enabling continuous d
transfer.

Figure 12.2 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark
level). The SCI monitors the transmission line, and when it goes to the space state (low
recognizes a start bit and starts serial communication.

One serial communication character consists of a start bit (low level), followed by data
first order), a parity bit (high or low level), and finally one or two stop bits (high level).

In asynchronous mode, the SCI performs synchronization at the falling edge of the star
reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16
length of one bit, so that the transfer data is latched at the center of each bit.

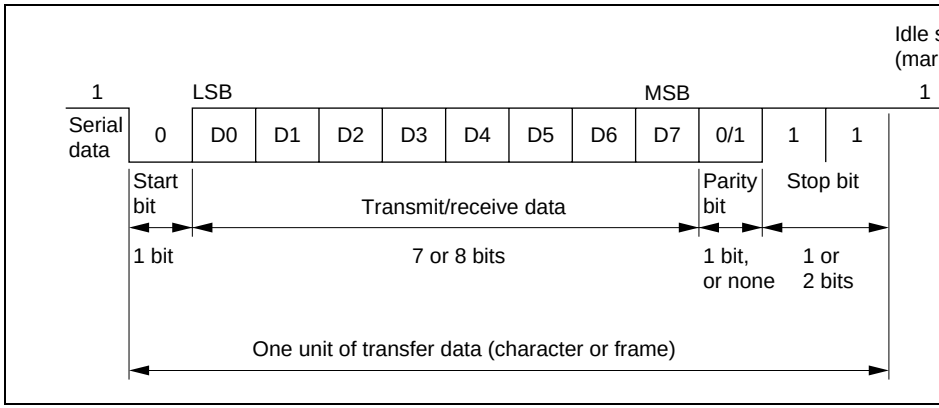


Figure 12.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11
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
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
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

When the SCI is operated on an internal clock, the clock can be output from the SCK p
frequency of the clock output in this case is equal to the bit rate, and the phase is such t
rising edge of the clock is in the middle of the transmit data, as shown in figure 12.3.

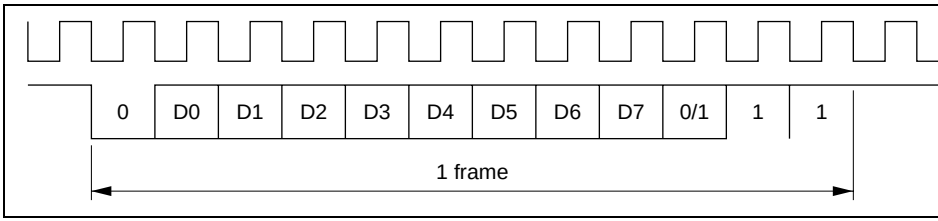


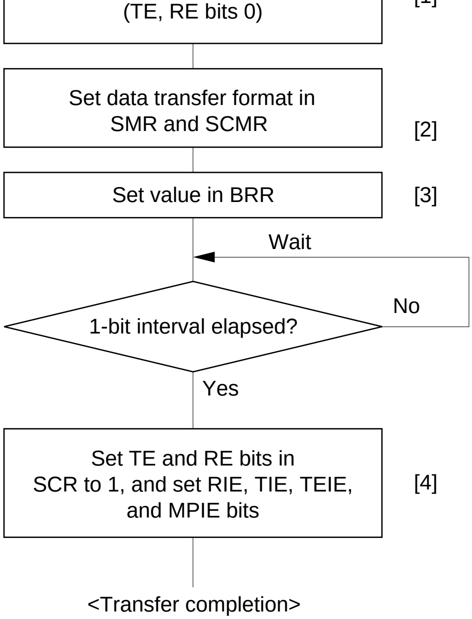
Figure 12.3 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

Data Transfer Operations

SCI initialization (asynchronous mode): Before transmitting and receiving data, you 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 0 before making the change using the following procedure. When the TE bit is cleared TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not 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, inclu initialization, since operation is uncertain.



immediately after SCR settin
made.

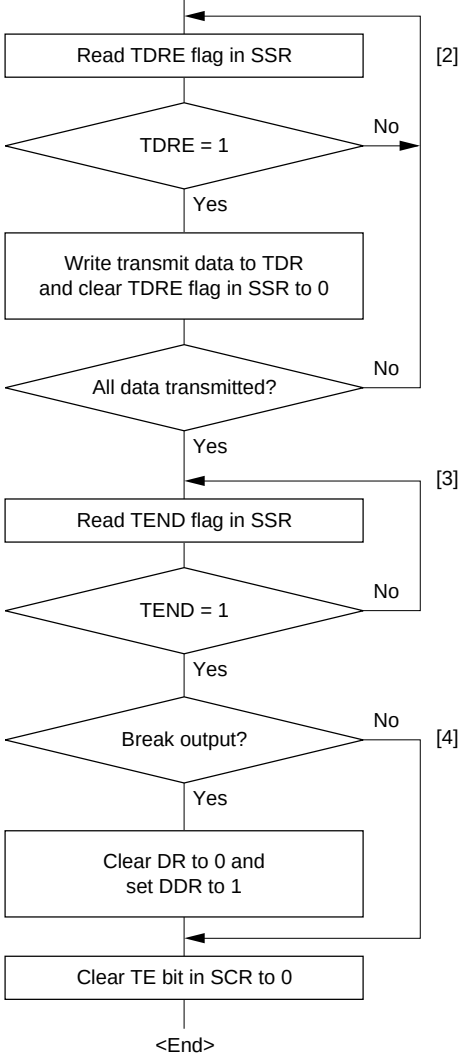
[2] Set the data transfer format i
and SCMR.

[3] Write a value corresponding
bit rate to BRR. Not necessa
external clock is used.

[4] Wait at least one bit interval,
set the TE bit or RE bit in SC
Also set the RIE, TIE, TEIE,
MPIE bits.

Setting the TE and RE bits e
the TxD and RxD pins to be

Figure 12.4 Sample SCI Initialization Flowchart



After the TE bit is set to 1, a fram
of 1s is output, and transmission
enabled.

[2] SCI status check and transmit d
write:
Read SSR and check that the
TDRE flag is set to 1, then write
transmit data to TDR and clear t
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. Check
and clearing of the TDRE flag is
automatic when the DTC is
activated by a transmit data emp
interrupt (TXI) request, and date
written to TDR.

[4] Break output at the end of serial
transmission:
To output a break in serial
transmission, set DDR for the po
corresponding to the TxD pin to
clear DR to 0, then clear the TE
in SCR to 0.

Figure 12.5 Sample Serial Transmission Flowchart

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 als 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 se

[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 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, a "mark state" is entered in which 1 is output continuously. If the TEIE bit in SCR is this time, a TEI interrupt request is generated.

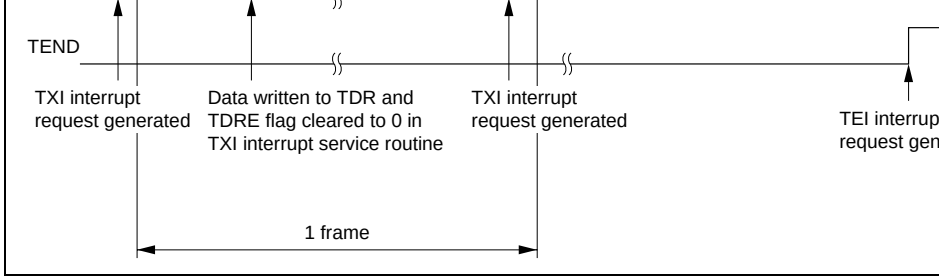
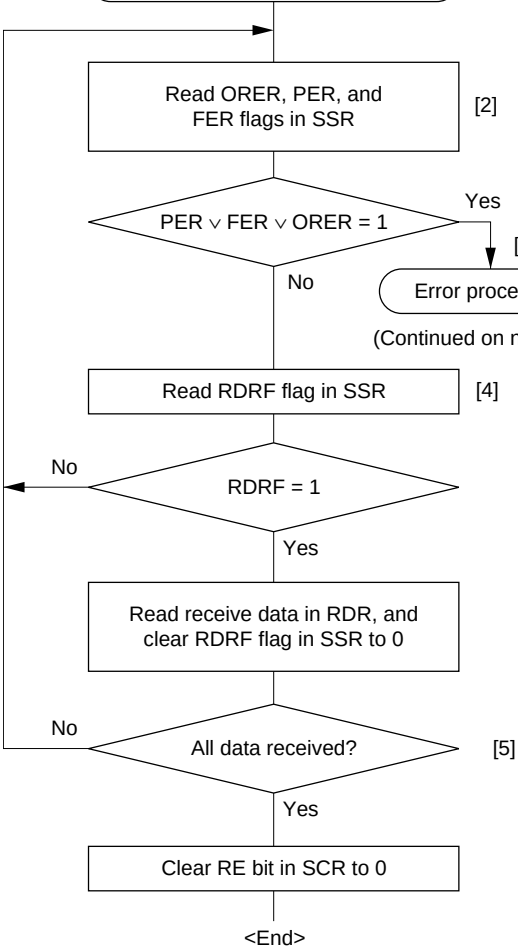


Figure 12.6 Example of Operation in Transmission in Asynchronous Mo (Example with 8-Bit Data, Parity, One Stop Bit)



[2] [3] Receive error process
break detection:
If a receive error occurs,
ORER, PER, and FER fl
SSR to identify the error
performing the appropria
processing, ensure that
ORER, PER, and FER fl
all cleared to 0. Recepti
be resumed if any of the
are set to 1. In the case
framing error, a break ca
detected by reading the
the input port correspon
the RxD pin.

[4] SCI status check and re
data read :
Read SSR and check th
= 1, then read the receiv
RDR and clear the RDR
0. Transition of the RDR
from 0 to 1 can also be i
by an RXI interrupt.

[5] Serial reception continua
procedure:
To continue serial recep
before the stop bit for th
frame is received, read t
RDRF flag, read RDR, a
the RDRF flag to 0. The
flag is cleared automatic
when the DTC is activat
RXI interrupt and the RD
is read.

Figure 12.7 Sample Serial Reception Data Flowchart

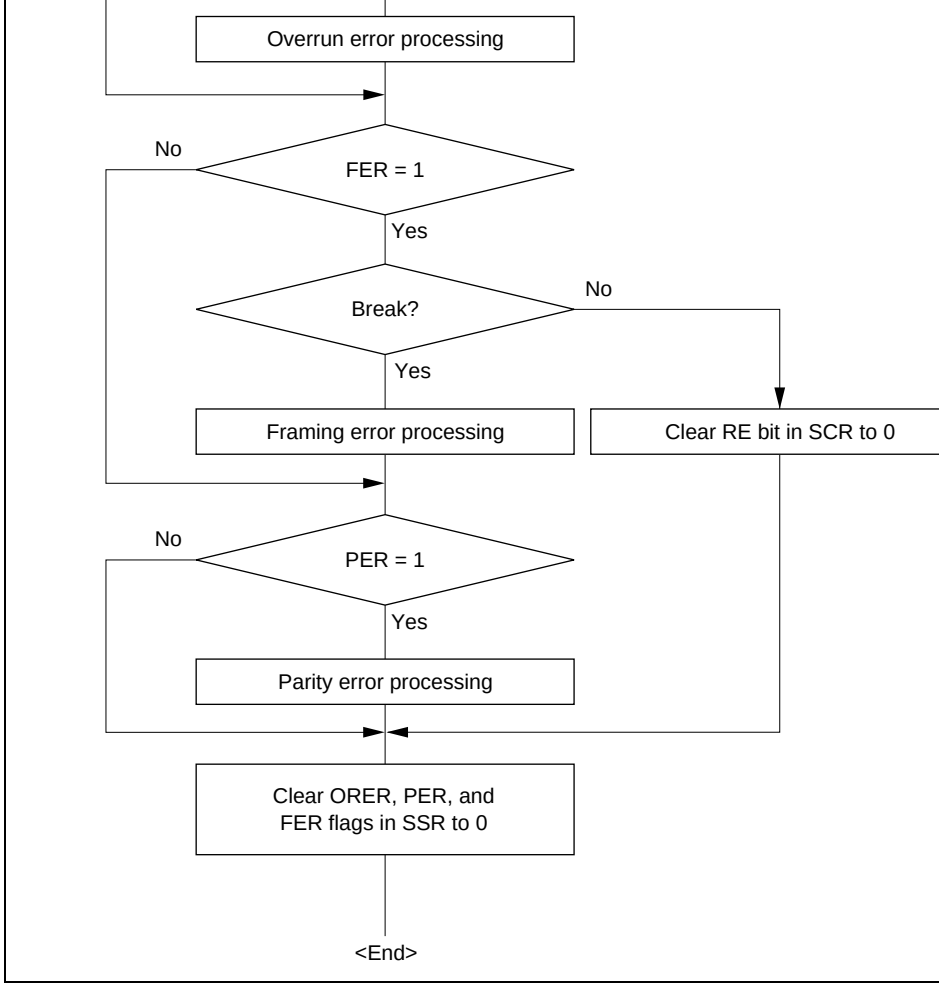


Figure 12.7 Sample Serial Reception Data Flowchart (cont)

[a] Parity check:

The SCI checks whether the number of 1 bits in the receive data agrees with th (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 ca transferred from RSR to RDR.

If all the above checks are passed, the RDRF flag is set to 1, and the receive data is RDR.

If a receive error* is detected in the error check, the operation is as shown in table

Note: * Subsequent receive operations cannot be performed when a receive error h Also note that the RDRF flag is not set to 1 in reception, and so the error fl cleared to 0.

[4] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data f (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER, PER, or FER flag changes receive error interrupt (ERI) request is generated.

from the parity (even or odd) set in SMR | from RSR to RD

Figure 12.8 shows an example of the operation for reception in asynchronous mode.

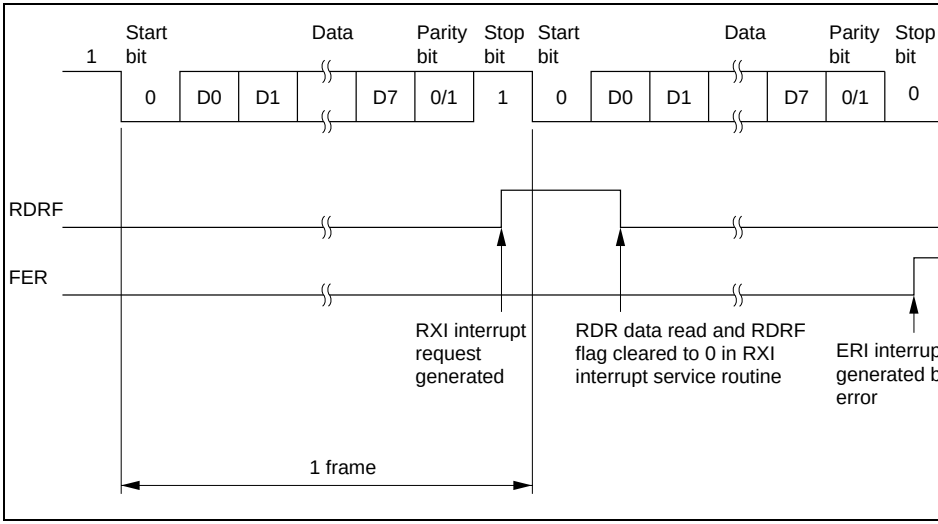


Figure 12.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

The serial communication cycle consists of two component cycles: an ID transmission which specifies the receiving station, and a data transmission cycle. The multiprocesso 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 serial communication as data with a 1 multiprocessor bit added. It then sends transmit 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 own ID. The station whose ID matches then receives the data sent next. Stations whos not match continue to skip the data until data with a 1 multiprocessor bit is again recei way, data communication is carried out among a number of processors.

Figure 12.9 shows an example of inter-processor communication using the multiproce

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 12.10.

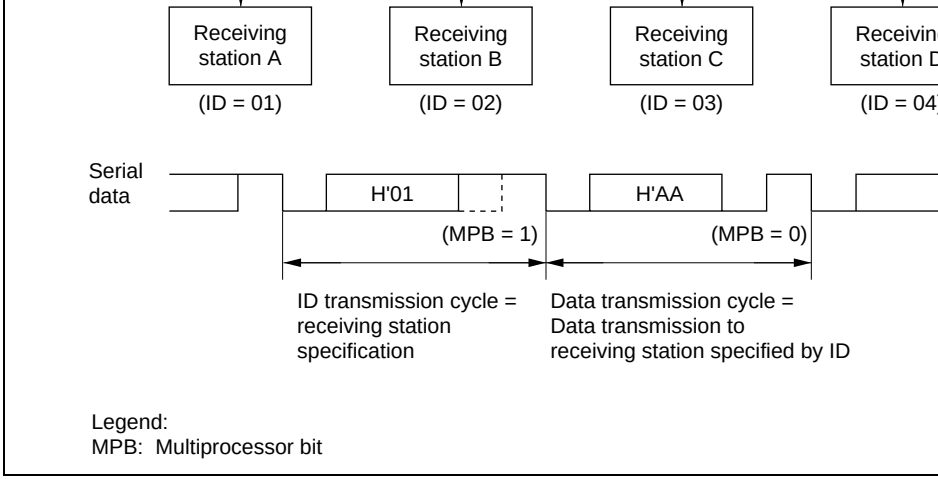
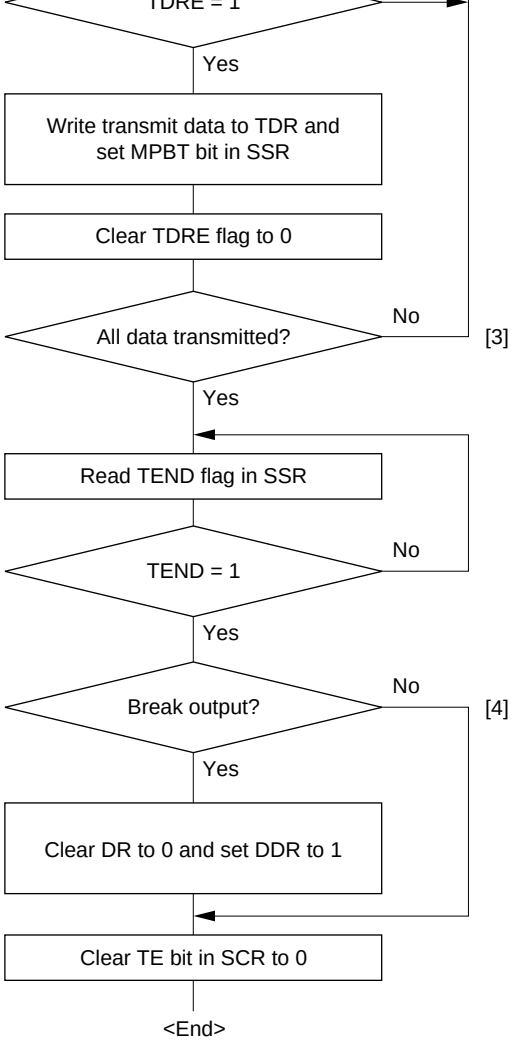


Figure 12.9 Example of Inter-Processor Communication Using Multiprocessor (Transmission of Data H'AA to Receiving Station A)

Data Transfer Operations

Multiprocessor serial data transmission: Figure 12.10 shows a sample flowchart for multiprocessor serial data transmission.

The following procedure should be used for multiprocessor serial data transmission.



data write.
Read SSR and check that t
TDRE flag is set to 1, then
transmit data to TDR. Set t
MPBT bit in SSR to 0 or 1.
Finally, clear the TDRE flag

[3] Serial transmission continu
procedure:
To continue serial transmis
be sure to read 1 from the
flag to confirm that writing i
possible, then write data to
and then clear the TDRE fl
0. Checking and clearing o
TDRE flag is automatic wh
DTC is activated by a trans
data empty interrupt (TXI)
request, and data is written
TDR.

[4] Break output at the end of
transmission:
To output a break in serial
transmission, set the port D
1, clear DR to 0, then clear
TE bit in SCR to 0.

Figure 12.10 Sample Multiprocessor Serial Transmission Flowchart

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 i 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, ar mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is se time, a transmission end interrupt (TEI) request is generated.

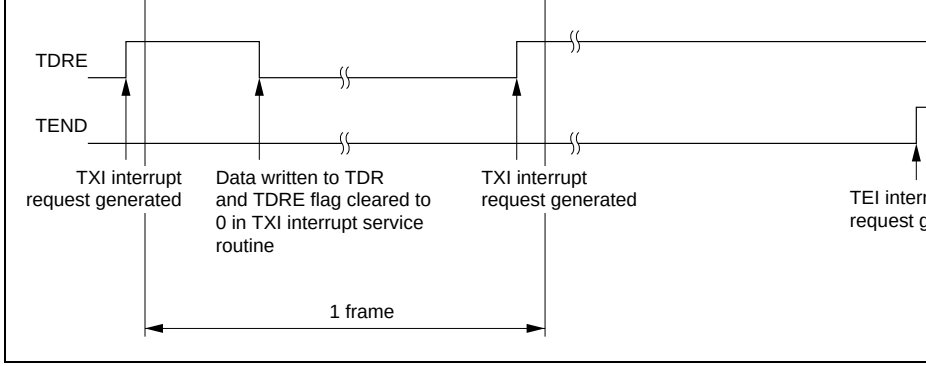
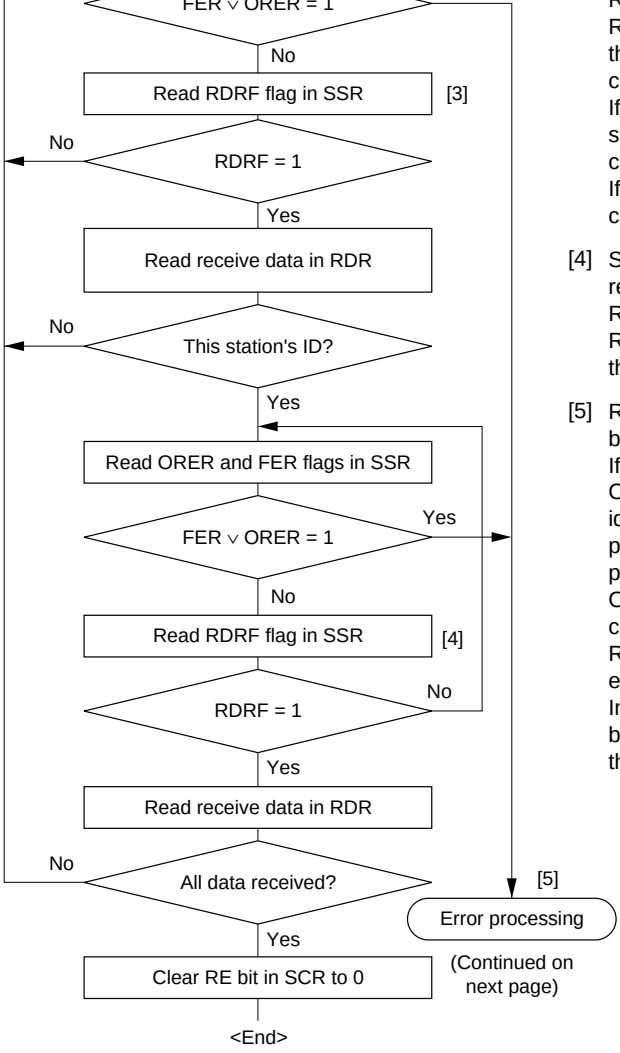


Figure 12.11 Example of SCI Operation in Transmission (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Multiprocessor serial data reception: Figure 12.12 shows a sample flowchart for mu serial reception.

The following procedure should be used for multiprocessor serial data reception.



[4] SCI status check and data reception:
Read SSR and check that t
RDRF flag is set to 1, then
the data in RDR.

[5] Receive error processing a
break detection:
If a receive error occurs, re
ORER and FER flags in SS
identify the error. After
performing the appropriate
processing, ensure that the
ORER and FER flags are a
cleared to 0.
Reception cannot be resum
either of these flags is set t
In the case of a framing err
break can be detected by re
the RxD pin value.

Figure 12.12 Sample Multiprocessor Serial Reception Flowchart

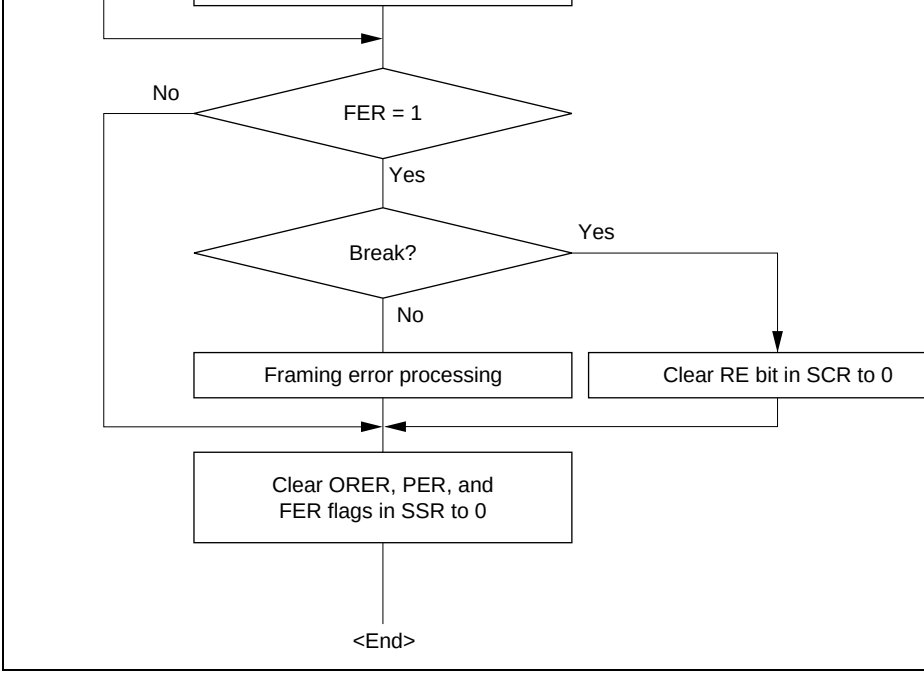


Figure 12.12 Sample Multiprocessor Serial Reception Flowchart (con

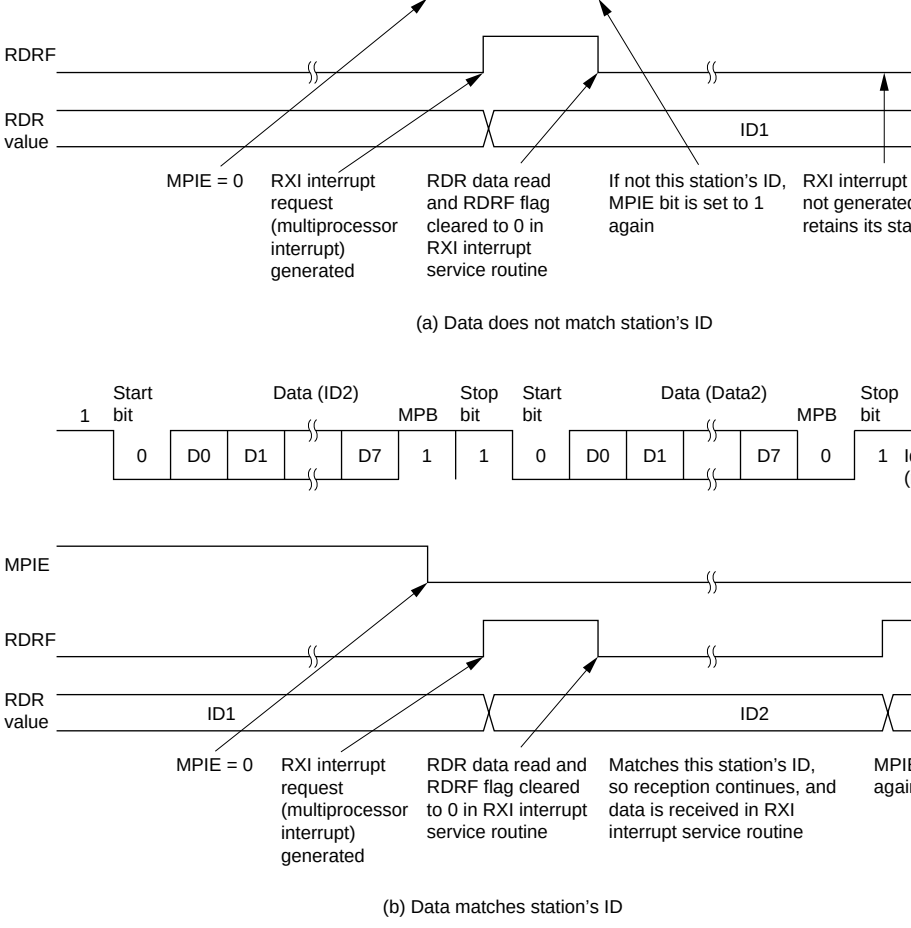


Figure 12.13 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Figure 12.14 shows the general format for clocked synchronous serial communication

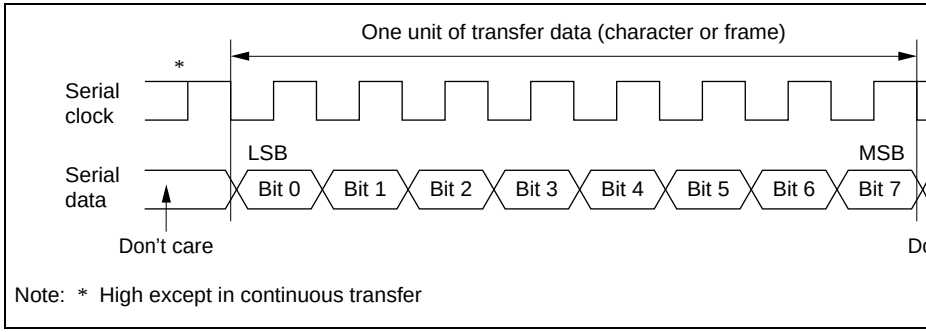


Figure 12.14 Data Format in Synchronous Communication

In clocked synchronous serial communication, data on the transmission line is output
falling edge of the serial clock to the next. Data confirmation is guaranteed at the risin
the serial clock.

In clocked serial communication, one character consists of data output starting with th
ending with the MSB. After the MSB is output, the transmission line holds the MSB s

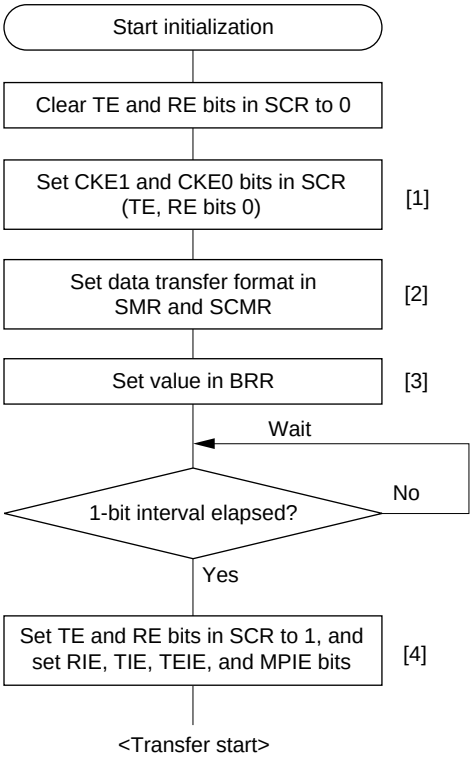
In clocked synchronous mode, the SCI receives data in synchronization with the rising
serial clock.

Either an internal clock generated by the on-chip baud rate generator or an external seri
input at the SCK pin can be selected, according to the setting of the C/$\overline{A}$ bit in SMR an
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

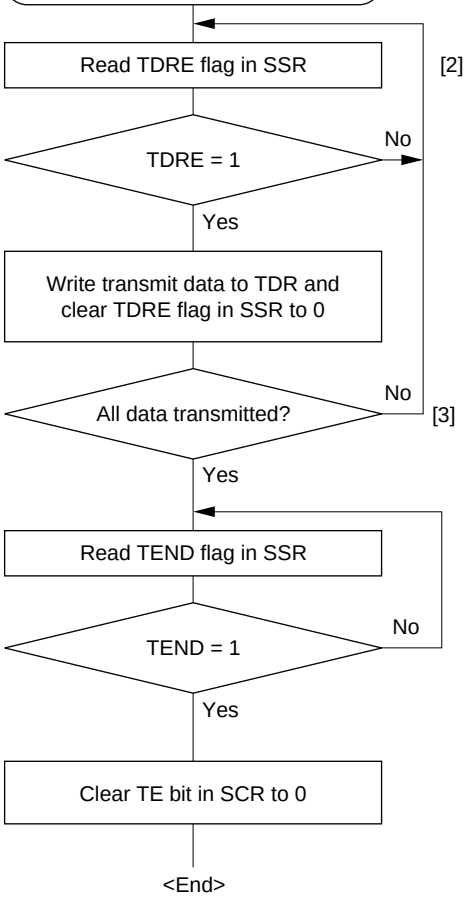
Eight serial clock pulses are output in the transfer of one character, and when no transfe
performed the clock is fixed high. When only receive operations are performed, howev
serial clock is output until an overrun error occurs or the RE bit is cleared to 0. If you w
perform receive operations in units of one character, you should select an external cloc
clock source.

Figure 12.15 shows a sample SCI initialization flowchart.



Note: When transmitting and receiving data simultaneously, the TE and RE bits should be cleared to 0 and then set to 1 at the same time.

Figure 12.15 Sample SCI Initialization Flowchart



pin.

[2] SCI status check and transmit d
write:
Read SSR and check that the T
flag is set to 1, then write transm
to TDR and clear the TDRE flag

[3] Serial transmission continuation
procedure:
To continue serial transmission,
sure to read 1 from the TDRE fl
confirm that writing is possible,
write data to TDR, and then clea
TDRE flag to 0.
Checking and clearing of the TD
flag is automatic when the DTC
activated by a transmit data em
interrupt (TXI) request and data
written to TDR.

Figure 12.16 Sample Serial Transmission Flowchart

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 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 tr 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 se 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.

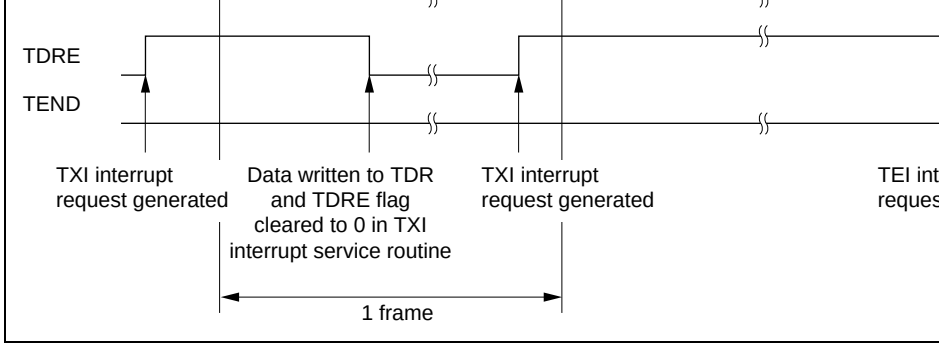


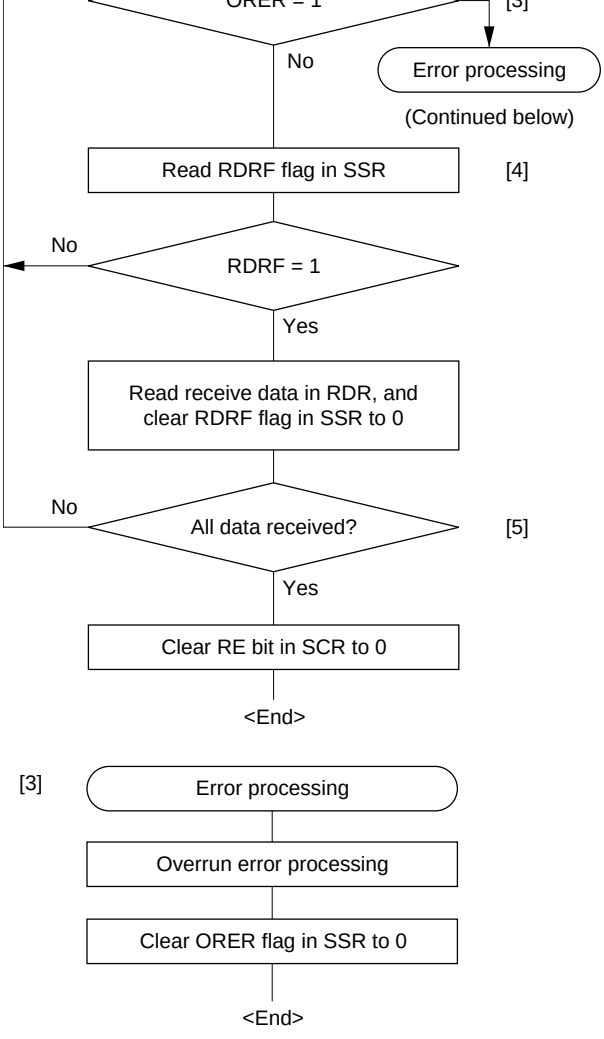
Figure 12.17 Example of SCI Operation in Transmission

Serial data reception (clocked synchronous mode): Figure 12.18 shows a sample flo serial reception.

The following procedure should be used for serial data reception.

When changing the operating mode from asynchronous to clocked synchronous, be sur 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 n operations will be possible.



[4] SCI status check and re
data read:
Read SSR and check th
RDRF flag is set to 1, th
the receive data in RDR
clear the RDRF flag to 0
Transition of the RDRF
0 to 1 can also be ident
an RXI interrupt.

[5] Serial reception continu
procedure:
To continue serial recep
before the MSB (bit 7) o
current frame is receive
reading the RDRF flag,
RDR, and clearing the R
to 0. The RDRF flag is c
automatically when the
activated by a receive d
interrupt (RXI) request a
RDR value is read.

Figure 12.18 Sample Serial Reception Flowchart

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 rece 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 fu (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER flag changes to 1, a receive e interrupt (ERI) request is generated.

Figure 12.19 shows an example of SCI operation in reception.

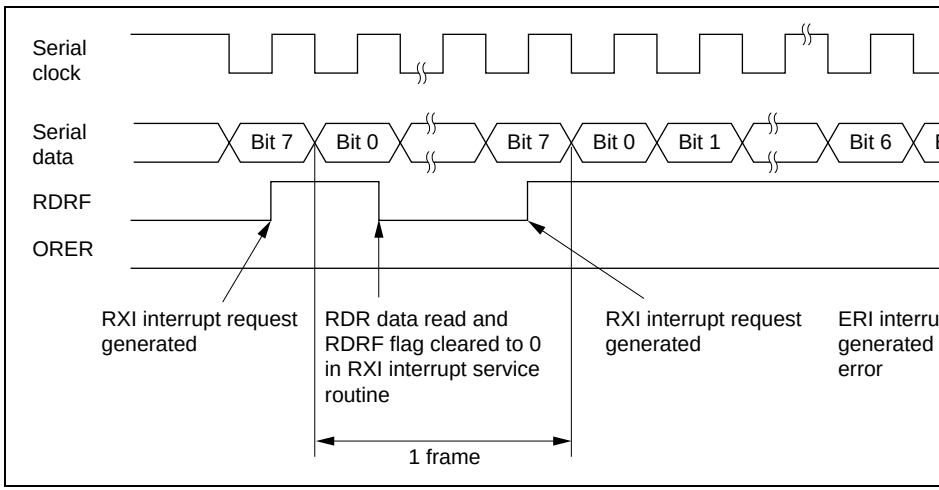
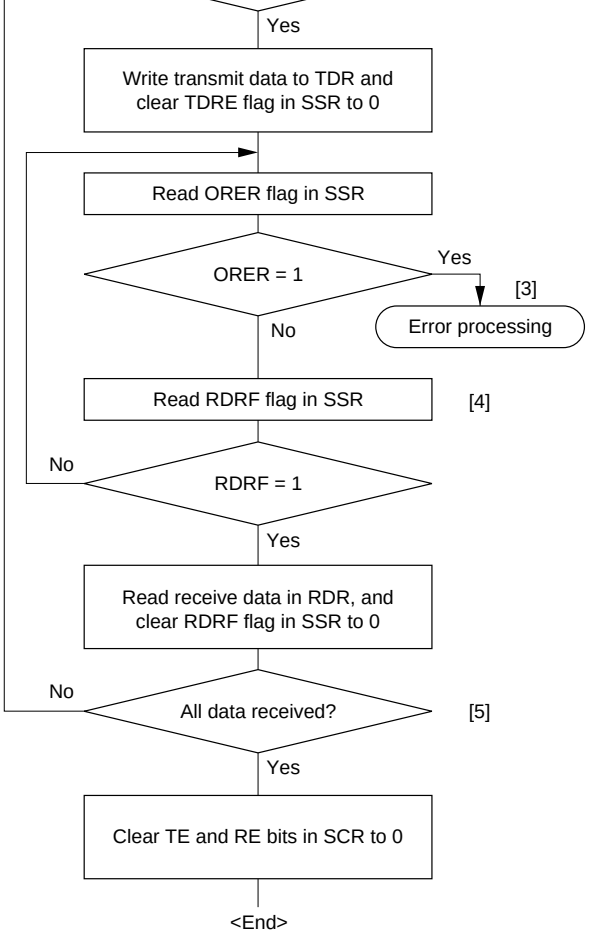


Figure 12.19 Example of SCI Operation in Reception

Simultaneous serial data transmission and reception (clocked synchronous mode): 12.20 shows a sample flowchart for simultaneous serial transmit and receive operations

The following procedure should be used for simultaneous serial data transmit and recei operations.



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE and RE bits to 0, then set both of these bits to 1.

transmit data to TDR and
TDRE flag to 0.
Transition of the TDRE fla
to 1 can also be identified
interrupt.

[3] Receive error processing
If a receive error occurs, r
ORER flag in SSR, and a
performing the appropriat
processing, clear the ORE
0. Transmission/reception
resumed if the ORER flag
1.

[4] SCI status check and rec
read:
Read SSR and check tha
RDRF flag is set to 1, the
receive data in RDR and
RDRF flag to 0. Transition
RDRF flag from 0 to 1 can
identified by an RXI interr

[5] Serial transmission/recep
continuation procedure:
To continue serial transm
reception, before the MSB
the current frame is receiv
reading the RDRF flag, re
RDR, and clearing the RD
0. Also, before the MSB (
the current frame is transm
read 1 from the TDRE flag
confirm that writing is pos
Then write data to TDR a
the TDRE flag to 0.
Checking and clearing of
flag is automatic when the
activated by a transmit da
interrupt (TXI) request an
written to TDR. Also, the
is cleared automatically w
DTC is activated by a rec
full interrupt (RXI) request
RDR value is read.

Figure 12.20 Sample Flowchart of Simultaneous Serial Transmit and Receive

in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the
perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer i
performed by the DTC. The 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 th
PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI inte
activate the DTC to perform data transfer. The RDRF flag is cleared to 0 automatically
transfer is performed by the DTC. The DTC cannot be activated by an ERI interrupt re

Table 12.12 SCI Interrupt Sources

Channel	Interrupt Source	Description	DTC Activation
0	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible
	RXI	Interrupt due to receive data full state (RDRF)	Possible
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible
	TEI	Interrupt due to transmission end (TEND)	Not possible
1	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible
	RXI	Interrupt due to receive data full state (RDRF)	Possible
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible
	TEI	Interrupt due to transmission end (TEND)	Not possible

Note: * This table shows the initial state immediately after a reset. Relative priorities
channels can be changed by means of ICR and IPR.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to
TEND flag is cleared at the same time as the TDRE flag. Consequently, if a TEI interr
TXI interrupt are requested simultaneously, the TXI interrupt may be accepted first, w

The TDRE flag in SSR is a status flag that indicates that transmit data has been transf
TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to

Data can be written to TDR regardless of the state of the TDRE flag. However, if new
written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be los
not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is
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 SSI
shown in table 12.13. If there is an overrun error, data is not transferred from RSR to R
the receive data is lost.

Table 12.13 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 Sta
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 + fra
1	1	0	1	X	Overrun error + pa
0	0	1	1	○	Framing error + pa
1	1	1	1	X	Overrun error + fra parity error

Notes: ○: Receive data is transferred from RSR to RDR.
X: Receive data is not transferred from RSR to RDR.

whose direction (input or output) is determined by DR and DDR. This can be used to se

Between serial transmission initialization and setting of the TE bit to 1, the mark state i
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 tra
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 t
the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before startin
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
rate.

In reception, the SCI samples the falling edge of the start bit using the basic clock, and
internal synchronization. Receive data is latched internally at the rising edge of the 8th
basic clock. This is illustrated in figure 12.21.

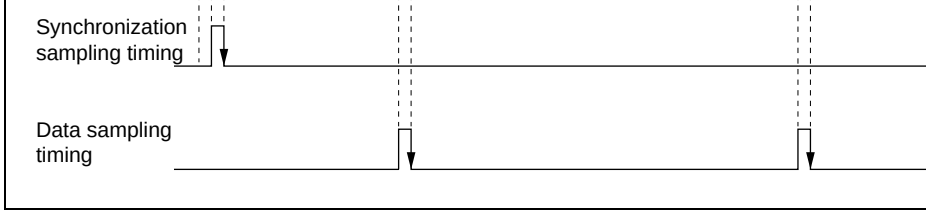


Figure 12.21 Receive Data Sampling Timing in Asynchronous Mode

Thus the reception margin in asynchronous mode is given by formula (1) below.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}\,(1 + F) \right| \times 100\% \quad \text{..... Fo}$$

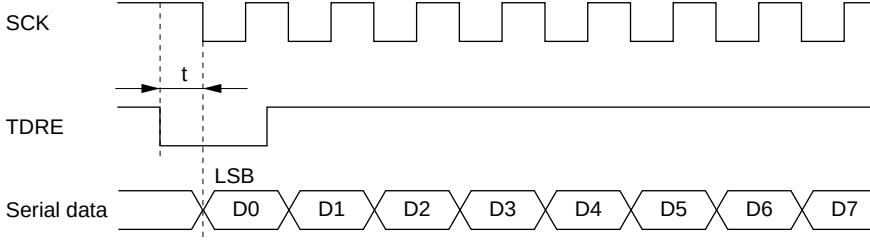
Where M : Reception margin (%)
N : Ratio of bit rate to clock (N = 16)
D : Clock duty (D = 0 to 1.0)
L : Frame length (L = 9 to 12)
F : Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), a reception margin of 46.875% i
formula (2) below.

When D = 0.5 and F = 0,

$$M = (0.5 - \frac{1}{2 \times 16}) \times 100\%$$
$$= 46.875\% \quad \text{..... Fo}$$

However, this is only the computed value, and a margin of 20% to 30% should be allo
system design.



Note: When operating on an external clock, set t >4 clocks.

Figure 12.22 Example of Clocked Synchronous Transmission by DTC

Interrupts and Module Stop Mode: If module stop mode is entered when an interrupt requested, it will not be possible to clear the CPU interrupt source or DTC activation so Interrupts should therefore be disabled before entering module stop mode.

4. Occurrence of low-level output (see figure 12.23)

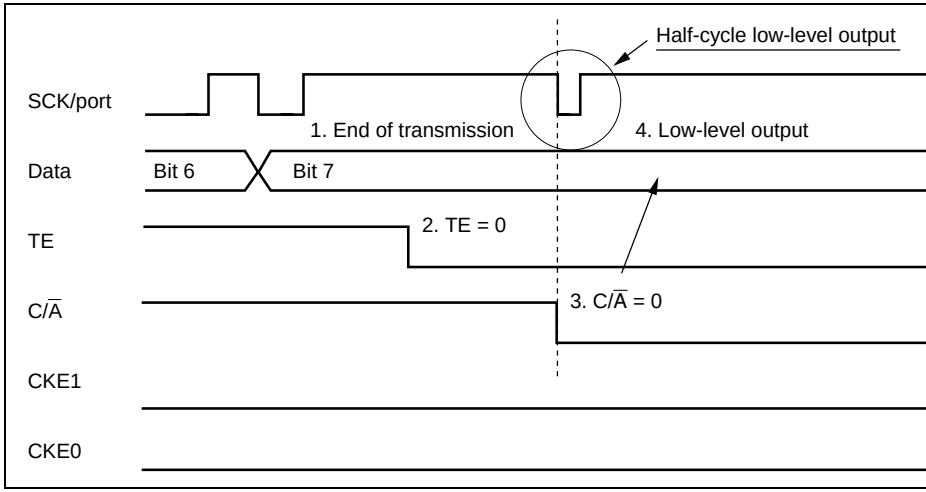


Figure 12.23 Operation when Switching from SCK Pin Function to Port Pin

4. C/$\overline{A}$ bit = 0 ... switchover to port output
5. CKE1 bit = 0

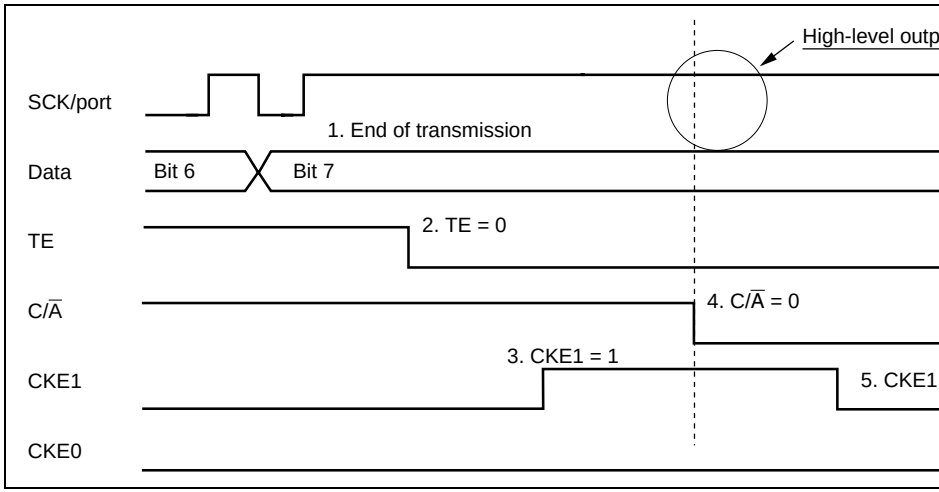


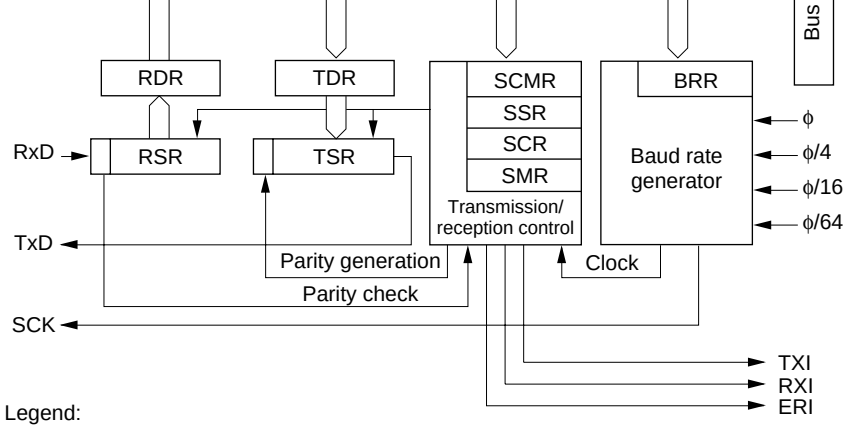
Figure 12.24 Operation when Switching from SCK Pin Function to Port Pin F (Example of Preventing Low-Level Output)

Switching between the normal serial communication interface and the Smart Card inte
carried out by means of a register setting.

13.1.1 Features

Features of the Smart Card interface supported by the H8S/2345 Group are as follows

- Asynchronous mode
  - Data length: 8 bits
  - Parity bit generation and checking
  - Transmission of error signal (parity error) in receive mode
  - Error signal detection and automatic data retransmission in transmit mode
  - Direct convention and inverse convention both supported
- On-chip baud rate generator allows any bit rate to be selected
- Three interrupt sources
  - Three interrupt sources (transmit data empty, receive data full, and transmit/rec that can issue requests independently
  - The transmit data empty interrupt and receive data full interrupt can activate th transfer controller (DTC) to execute data transfer



Legend:

SCMR : Smart Card mode register
RSR : Receive shift register
RDR : Receive data register
TSR : Transmit shift register
TDR : Transmit data register
SMR : Serial mode register
SCR : Serial control register
SSR : Serial status register
BRR : Bit rate register

Figure 13.1 Block Diagram of Smart Card Interface

	Transmit data pin 0	TxD0	Output	SCI0 transmit data outpu
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 outpu

0	Serial mode register 0	SMR0	R/W	H'00	H'
	Bit rate register 0	BRR0	R/W	H'FF	H'
	Serial control register 0	SCR0	R/W	H'00	H'
	Transmit data register 0	TDR0	R/W	H'FF	H'
	Serial status register 0	SSR0	R/(W)*2	H'84	H'
	Receive data register 0	RDR0	R	H'00	H'
	Smart card mode register 0	SCMR0	R/W	H'F2	H'
1	Serial mode register 1	SMR1	R/W	H'00	H'
	Bit rate register 1	BRR1	R/W	H'FF	H'
	Serial control register 1	SCR1	R/W	H'00	H'
	Transmit data register 1	TDR1	R/W	H'FF	H'
	Serial status register 1	SSR1	R/(W)*2	H'84	H'
	Receive data register 1	RDR1	R	H'00	H'
	Smart card mode register 1	SCMR1	R/W	H'F2	H'
All	Module stop control register	MSTPCR	R/W	H'3FFF	H'

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

	—	—	—	—	SDIR	SINV	—
Initial value :	1	1	1	1	0	0	1
R/W :	—	—	—	—	R/W	R/W	—

SCMR is an 8-bit readable/writable register that selects the Smart Card interface funct

SCMR is initialized to H'F2 by a reset, and in standby mode or module stop 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 con format.

Bit 3

SDIR	Description
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

	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

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

Bit 0—Smart Card Interface Mode Select (SMIF): Enables or disables the Smart Ca function.

Bit 0

SMIF	Description
0	Smart Card interface function is disabled
1	Smart Card interface function is enabled

13.2.2 Serial Status Register (SSR)

Bit	:	7	6	5	4	3	2	1
		TDRE	RDRF	ORER	ERS	PER	TEND	MPB
Initial value	:	1	0	0	0	0	1	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R

Note: * Only 0 can be written to bits 7 to 3, to clear these flags.

Bit 4 of SSR has a different function in Smart Card interface mode. Coupled with this, conditions for bit 2, TEND, are also different.

Bits 7 to 5—Operate in the same way as for the normal SCI. For details, see section 12 Status Register (SSR).

	• Upon reset, and in standby mode or module stop mode • When 0 is written to ERS after reading ERS = 1
1	Indicates that an error signal was sent from the receiving side showing tha error was detected [Setting condition] When the low level of the error signal is sampled

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its p state.

Bits 3 to 0—Operate in the same way as for the normal SCI. For details, see section 1 Status Register (SSR).

However, the setting conditions for the TEND bit, are as shown below.

Bit 2

TEND	Description
0	Indicates data transmission in progress [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	Indicates that data transmission is finished [Setting conditions] • Upon reset, and in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 and ERS = 0 (normal transmission) 2.5 etu after a 1-l character is transmitted when GM = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after a 1-l character is transmitted when GM = 1.

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

Note: * When the smart card interface is used, be sure to make the 0 or 1 setting sh
bits 6, 5, 3, and 2.

Bit 7 of SMR has a different function in smart card interface mode.

Bit 7—GSM Mode (GM): Sets the smart card interface function to GSM mode.

This bit is cleared to 0 when the normal smart card interface is used. In GSM mode, th
to 1, the timing of setting of the TEND flag that indicates transmission completion is ad
and clock output control mode addition is performed. The contents of the clock output
mode addition are specified by bits 1 and 0 of the serial control register (SCR).

Bit 7

GM	Description
0	Normal smart card interface mode operation (In • TEND flag generation 12.5 etu after beginning of start bit • Clock output ON/OFF control only
1	GSM mode smart card interface mode operation • TEND flag generation 11.0 etu after beginning of start bit • High/low fixing control possible in addition to clock output ON/OFF cont SCR)

Note: etu: Elementary time unit (time for transfer of 1 bit)

Bits 6 to 0—Operate in the same way as for the normal SCI.

For details, see section 12.2.5, Serial Mode Register (SMR).

Bits 7 to 2—Operate in the same way as for the normal SCI.

For details, see section 12.2.6, Serial Control Register (SCR).

Bits 1 and 0—Clock Enable (CKE1, CKE0): Selects the clock source, and enables c
clock output from the SCK pin.

In smart card interface mode, it is possible to switch between enabling and disabling c
clock output, and specify a fixed high level or fixed low level for the clock output.

SCMR	SMR	SCR Setting		SCK Pin Function Description
SMIF	C/$\overline{A}$, GM	CKE1	CKE0	
0	Refer to SCI designation			
1	0	0	0	The pin functions as an I/O port
			1	The pin outputs the clock as the SCK output p
	1		0	The pin outputs fixed low level as the SCK ou
			1	The pin outputs the clock as the SCK output p
		1	0	The pin outputs fixed high level as the SCK ou
			1	The pin outputs the clock as the SCK output p

one bit) is left between the end of the parity bit and the start of the next frame.

- If a parity error is detected during reception, a low error signal level is output for on period, 10.5 etu after the start bit.
- If the error signal is sampled during transmission, the same data is transmitted auto after the elapse of 2 etu or longer.
- Only start-stop asynchronous communication is supported; there is no clocked sync communication function.

13.3.2 Pin Connections

Figure 13.2 shows a schematic diagram of Smart Card interface related pin connections

In communication with an IC card, since both transmission and reception are carried ou single data transmission line, the TxD pin and RxD pin should be connected with the L data transmission line should be pulled up to the V_{CC} power supply with a resistor.

When the clock generated on the Smart Card interface is used by an IC card, the SCK p input to the CLK pin of the IC card. No connection is needed if the IC card uses an inte

LSI port output is used as the reset signal.

Other pins must normally be connected to the power supply or ground.

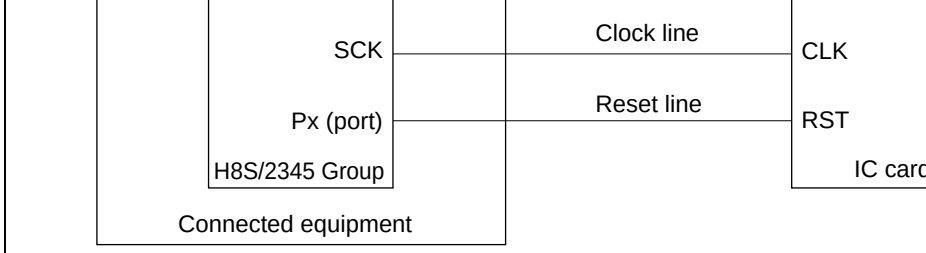
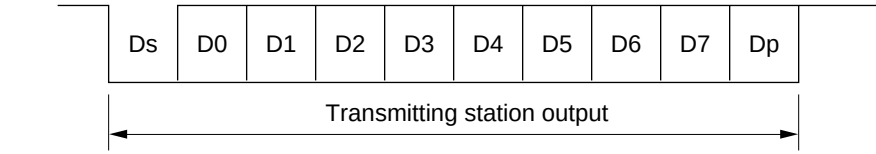
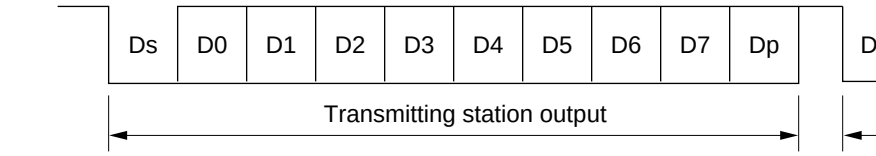


Figure 13.2 Schematic Diagram of Smart Card Interface Pin Connecti

Note: If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out.



When a parity error occurs



Legend:
Ds : Start bit
D0 to D7 : Data bits
Dp : Parity bit
DE : Error signal

Figure 13.3 Smart Card Interface Data Format

line is pulled high with a pull-up resistor.

[4] The receiving station carries out a parity check.

If there is no parity error and the data is received normally, the receiving station w reception of the next data.

If a parity error occurs, however, the receiving station outputs an error signal (DE, to request retransmission of the data. After outputting the error signal for the presc of time, the receiving station places the signal line in the high-impedance state aga signal line is pulled high again by a pull-up resistor.

[5] If the transmitting station does not receive an error signal, it proceeds to transmit t frame.

If it does receive an error signal, however, it returns to step [2] and retransmits the data.

	Bit						
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**
SMR	GM	0	1	O/$\overline{E}$	1	0	CKS1
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1
SCR	TIE	RIE	TE	RE	0	0	CKE1*
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1
SSR	TDRE	RDRF	ORER	ERS	PER	TEND	0
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1
SCMR	—	—	—	—	SDIR	SINV	—

Notes: — : Unused bit.

*: The CKE1 bit must be cleared to 0 when the GM bit in SMR is cleared to 0.

SMR Setting: The GM bit is cleared to 0 in normal smart card interface mode, and set GSM mode. The O/$\overline{E}$ bit is cleared to 0 if the IC card is of the direct convention type, a if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the on-chip baud rate generator. See se 13.3.5, Clock.

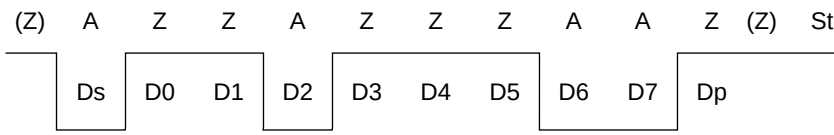
BRR Setting: BRR is used to set the bit rate. See section 13.3.5, Clock, for the method calculating the value to be set.

SCR Setting: The function of the TIE, RIE, TE, and RE bits is the same as for the norm For details, see section 12, Serial Communication Interface (SCI).

Bits CKE1 and CKE0 specify the clock output. When the GM bit in SMR is cleared to bits to B'00 if a clock is not to be output, or to B'01 if a clock is to be output. When the SMR is set to 1, clock output is performed. The clock output can also be fixed high or

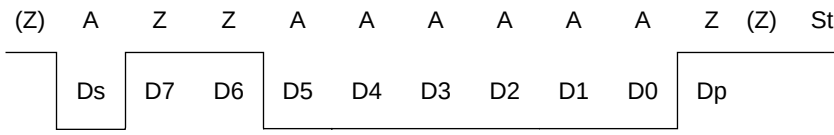
types of IC card (direct convention and inverse convention).

- Direct convention (SDIR = SINV = O/$\overline{E}$ = 0)



With the direct convention type, the logic 1 level corresponds to state Z and the lo
state A, and transfer is performed in LSB-first order. The start character data abov
The parity bit is 1 since even parity is stipulated for the Smart Card.

- Inverse convention (SDIR = SINV = O/$\overline{E}$ = 1)



With the inverse convention type, the logic 1 level corresponds to state A and the l
to state Z, and transfer is performed in MSB-first order. The start character data ab
The parity bit is 0, corresponding to state Z, since even parity is stipulated for the S
With the H8S/2345 Group, inversion specified by the SINV bit applies only to the
D7 to D0. For parity bit inversion, the O/$\overline{E}$ bit in SMR is set to odd parity mode (th
applies to both transmission and reception).

$$B = \frac{\phi}{1488 \times 2^{2n-1} \times (N + 1)} \times 10^6$$

Where: N = Value set in BRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
ϕ = Operating frequency (MHz)
n = See table 13.4

Table 13.4 Correspondence between n and CKS1, CKS0

n	CKS1	CKS0
0	0	0
1		1
2	1	0
3		1

Table 13.5 Examples of Bit Rate B (bit/s) for Various BRR Settings (When n = 0

	ϕ (MHz)					
N	**10.00**	**10.714**	**13.00**	**14.285**	**16.00**	**18.00**
0	13441	14400	17473	19200	21505	24194
1	6720	7200	8737	9600	10753	12097
2	4480	4800	5824	6400	7168	8065

Note: Bit rates are rounded to the nearest whole number.

	φ (MHz)													
	7.1424		10.00		10.7136		13.00		14.2848		16.00		18.00	
bit/s	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Erro
9600	0	0.00	1	30	1	25	1	8.99	1	0.00	1	12.01	2	15.9

Table 13.7 Maximum Bit Rate at Various Frequencies (Smart Card Interface M

φ (MHz)	Maximum Bit Rate (bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0
20.00	26882	0	0

The bit rate error is given by the following formula:

$$\text{Error (\%)} = \left(\frac{\phi}{1488 \times 2^{2n-1} \times B \times (N + 1)} \times 10^6 - 1 \right) \times 100$$

[3] Set the O/$\overline{E}$ bit and CKS1 and CKS0 bits in SMR. Clear the C/$\overline{A}$, CHR, and MP bit
set the STOP and PE bits to 1.

[4] Set the SMIF, SDIR, and SINV bits in SCMR.
When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports
and are placed in the high-impedance state.

[5] Set the value corresponding to the bit rate in BRR.

[6] Set the CKE0 bit in SCR. Clear the TIE, RIE, TE, RE, MPIE, TEIE, and CKE1 bit
If the CKE0 bit is set to 1, the clock is output from the SCK pin.

[7] Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do no
bit and RE bit at the same time, except for self-diagnosis.

[2] Check that the ERS error flag in SSR is cleared to 0.

[3] Repeat steps [2] and [3] until it can be confirmed that the TEND flag in SSR is set

[4] Write the transmit data to TDR, clear the TDRE flag to 0, and perform the transmi
The TEND flag is cleared to 0.

[5] When transmitting data continuously, go back to step [2].

[6] To end transmission, clear the TE bit to 0.
With the above processing, interrupt servicing or data transfer by the DTC is possi

If transmission ends and the TEND flag is set to 1 while the TIE bit is set to 1 and inte
requests are enabled, a transmit data empty interrupt (TXI) request will be generated.
occurs in transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and int
requests are enabled, a transfer error interrupt (ERI) request will be generated.

The timing for setting the TEND flag depends on the value of the GM bit in SMR. Th
timing is shown in figure 13.6.

If the DTC is activated by a TXI request, the number of bytes set in the DTC can be tr
automatically, including automatic retransmission.

For details, see Interrupt Operations and Data Transfer Operation by DTC below.

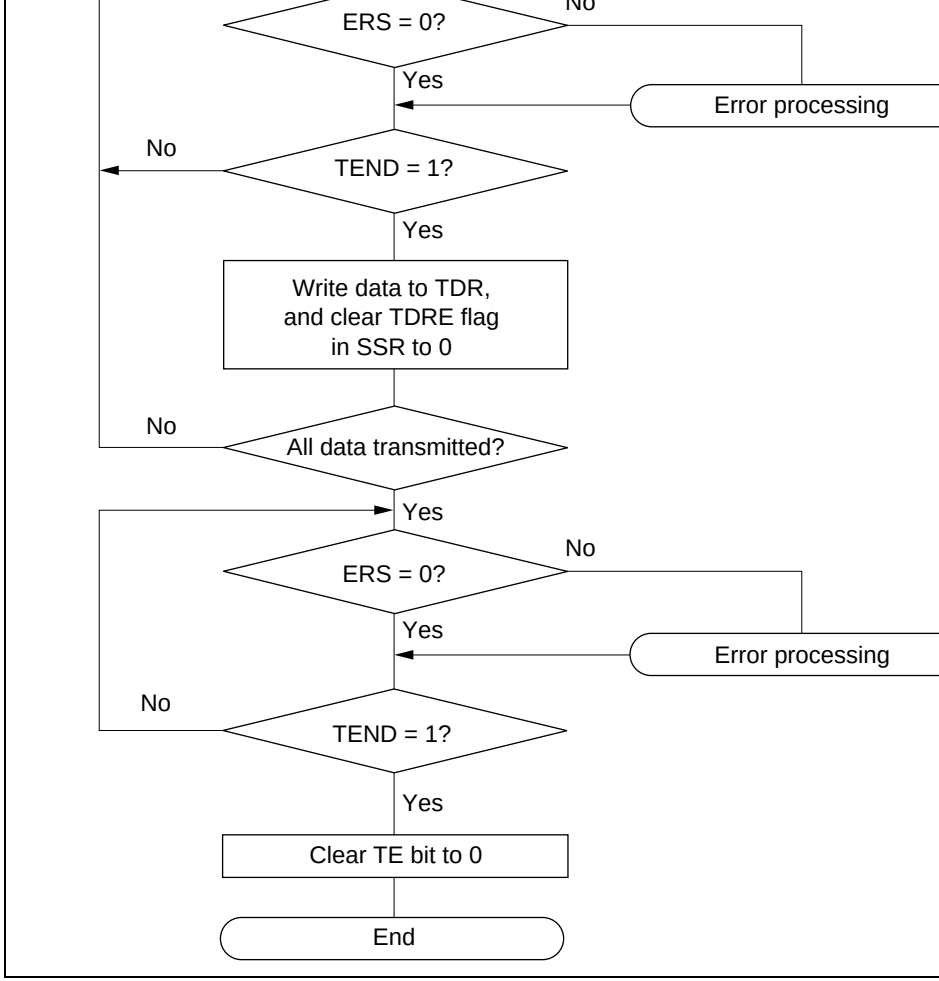
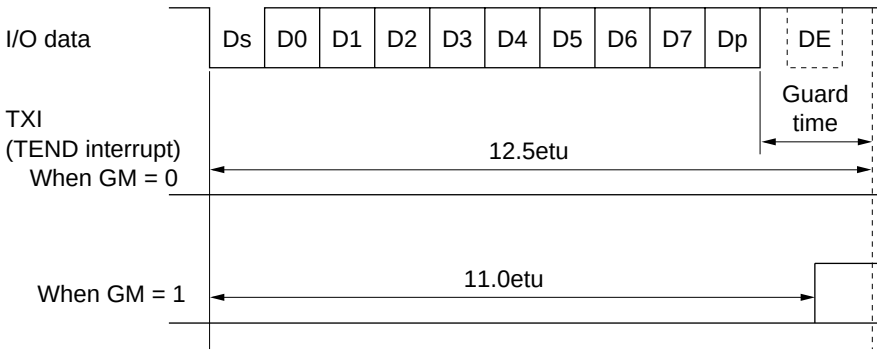


Figure 13.4 Example of Transmission Processing Flow

In case of normal transmission: TEND flag is set
In case of transmit error: ERS flag is set
Steps (2) and (3) above are repeated until the TEN

Note: When the ERS flag is set, it should be cleared until transfer of the last bit (D7 in transmission, D0 in MSB-first transmission) of the next transfer data to be transm been completed.

Figure 13.5 Relation between Transmit Operation and Internal Regist



Legend:
Ds : Start bit
D0 to D7 : Data bits
Dp : Parity bit
DE : Error signal

Figure 13.6 TEND Flag Generation Timing in Transmission Operatio

[4] Read the receive data from RDR.

[5] When receiving data continuously, clear the RDRF flag to 0 and go back to step [2]

[6] To end reception, clear the RE bit to 0.

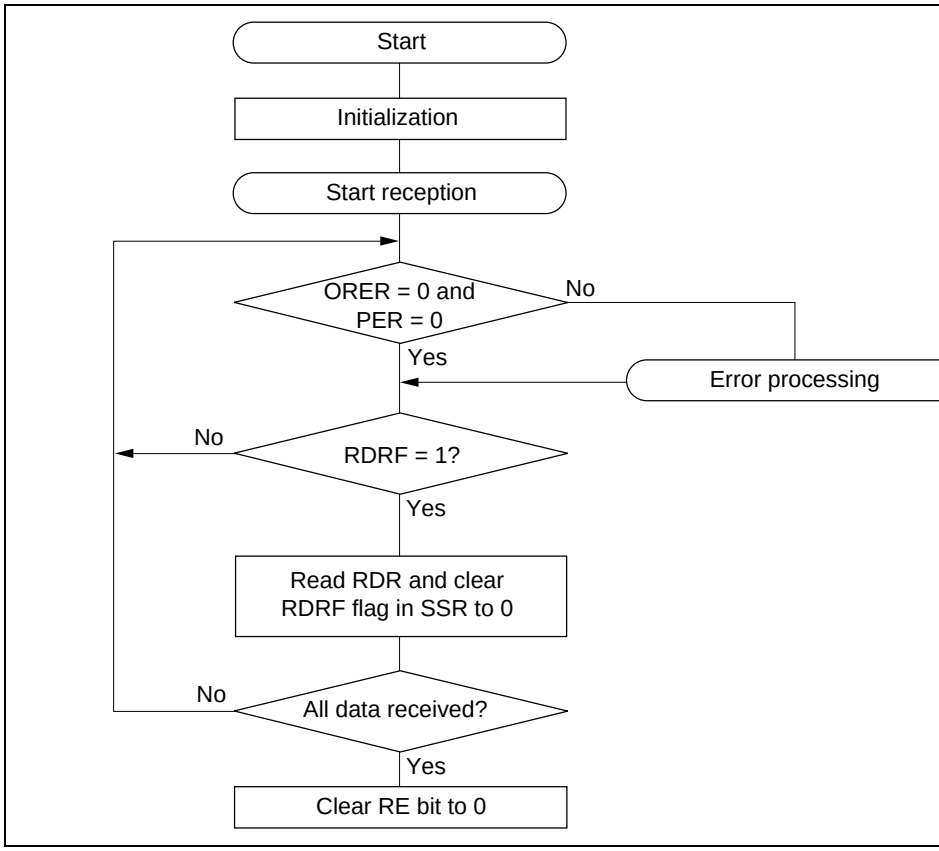


Figure 13.7 Example of Reception Processing Flow

For details, see Interrupt Operation and Data Transfer Operation by DTC below.

If a parity error occurs during reception and the PER is set to 1, the received data is sti
transferred to RDR, and therefore this data can be read.

Mode Switching Operation: When switching from receive mode to transmit mode, f
that the receive operation has been completed, then start from initialization, clearing R
and setting TE bit to 1. The RDRF flag or the PER and ORER flags can be used to che
receive operation has been completed.

When switching from transmit mode to receive mode, first confirm that the transmit o
been completed, then start from initialization, clearing TE bit to 0 and setting RE bit t
TEND flag can be used to check that the transmit operation has been completed.

Fixing Clock Output Level: When the GM bit in SMR is set to 1, the clock output le
fixed with bits CKE1 and CKE0 in SCR. At this time, the minimum clock pulse widt
made the specified width.

Figure 13.8 shows the timing for fixing the clock output level. In this example, GM is
CKE1 is cleared to 0, and the CKE0 bit is controlled.

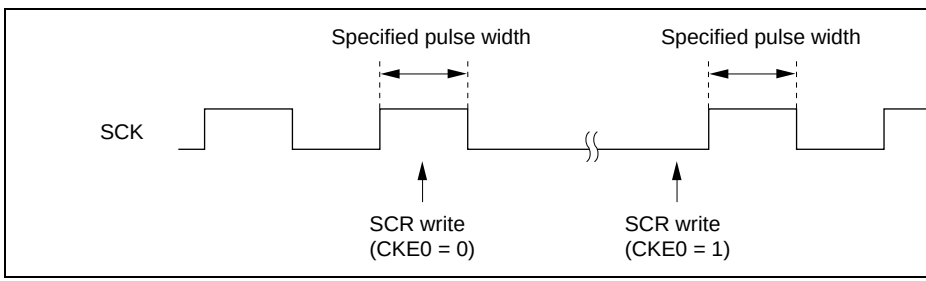


Figure 13.8 Timing for Fixing Clock Output Level

The relationship between the operating states and interrupt sources is shown in table 13

Table 13.8 Smart Card Mode Operating States and Interrupt Sources

Operating State		Flag	Enable Bit	Interrupt Source	DTC A
Transmit Mode	Normal operation	TEND	TIE	TXI	Possibl
	Error	ERS	RIE	ERI	Not pos
Receive Mode	Normal operation	RDRF	RIE	RXI	Possibl
	Error	PER, ORER	RIE	ERI	Not pos

Data Transfer Operation by DTC: In smart card mode, as with the normal SCI, trans carried out using the DTC. In a transmit operation, the TDRE flag is also set to 1 at the as the TEND flag in SSR, and a TXI interrupt is generated. If the TXI request is design beforehand as a DTC activation source, the DTC will be activated by the TXI request, a of the transmit data will be carried out. The TDRE and TEND flags are automatically c when data transfer is performed by the DTC. In the event of an error, the SCI retransmi data automatically. However, the ERS flag is not cleared automatically when an error o so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in of an error, and the ERS flag will be cleared.

When performing transfer using the DTC, it is essential to set and enable the DTC befo out SCI setting. For details of the DTC setting procedures, see section 7, Data Transfer

In a receive operation, an RXI interrupt request is generated when the RDRF flag in SS 1. If the RXI request is designated beforehand as a DTC activation source, the DTC wi activated by the RXI request, and transfer of the receive data will be carried out. The R cleared to 0 automatically when data transfer is performed by the DTC. If an error occu flag is set but the RDRF flag is not. The DTC is not activated, but instead, an ERI inter is sent to the CPU. Therefore, the error flag should be cleared.

[2] Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transmi
operation. At the same time, set the CKE1 bit to the value for the fixed output stat
software standby mode.

[3] Write 0 to the CKE0 bit in SCR to halt the clock.

[4] Wait for one serial clock period.

During this interval, clock output is fixed at the specified level, with the duty prese

[5] Write H'00 to SMR and SCMR.

[6] Make the transition to the software standby state.

- When returning to smart card interface mode from software standby mode

[7] Exit the software standby state.

[8] Set the CKE1 bit in SCR to the value for the fixed output state (current SCK pin st
software standby mode is initiated.

[9] Set smart card interface mode and output the clock. Signal generation is started wi
normal duty.

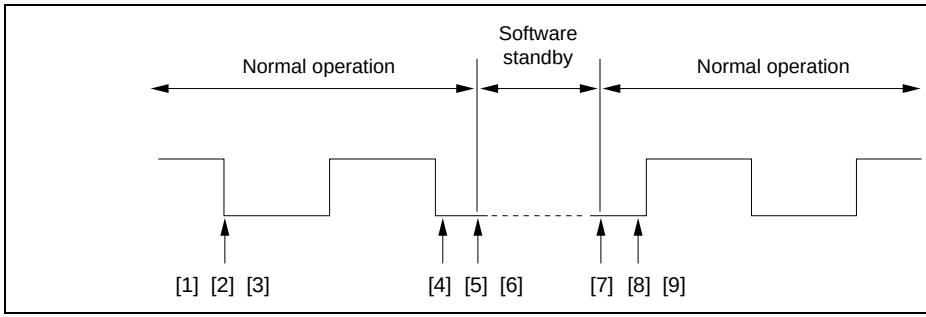


Figure 13.9 Clock Halt and Restart Procedure

13.4 Usage Note

The following points should be noted when using the SCI as a smart card interface.

Receive Data Sampling Timing and Reception Margin in Smart Card Interface M
smart card interface mode, the SCI operates on a basic clock with a frequency of 372 ti
transfer rate.

In reception, the SCI samples the falling edge of the start bit using the basic clock, and
internal synchronization. Receive data is latched internally at the rising edge of the 186
the basic clock. This is illustrated in figure 13.10.

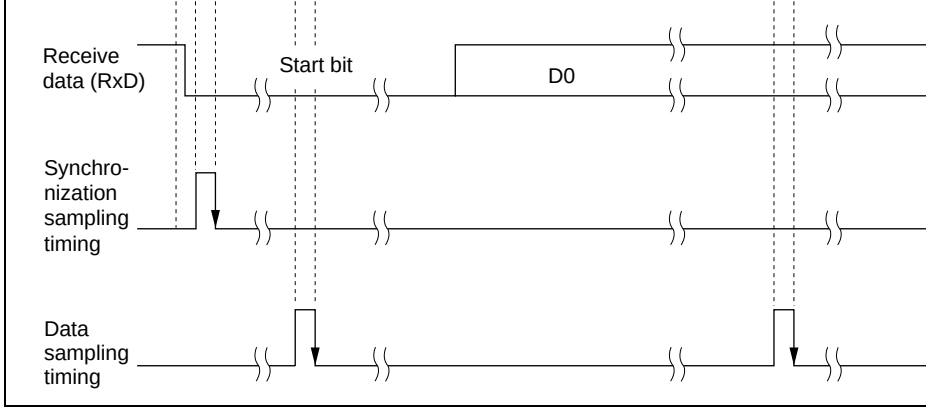


Figure 13.10 Receive Data Sampling Timing in Smart Card Mode

Thus the reception margin in smart card interface mode is given by the following form

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}\,(1 + F) \right| \times 100\%$$

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 372)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in the above formula, the reception margin forn
follows.

When D = 0.5 and F = 0,

$$M = (0.5 - 1/2 \times 372) \times 100\%$$
$$= 49.866\%$$

[2] The RDRF bit in SSR is not set for a frame in which an error has occurred.

[3] If no error is found when the received parity bit is checked, the PER bit in SSR is n

[4] If no error is found when the received parity bit is checked, the receive operation is have been completed normally, and the RDRF flag in SSR is automatically set to 1. bit in SCR is enabled at this time, an RXI interrupt request is generated.

If DTC data transfer by an RXI source is enabled, the contents of RDR can be read automatically. When the RDR data is read by the DTC, the RDRF flag is automatic to 0.

[5] When a normal frame is received, the pin retains the high-impedance state at the tin error signal transmission.

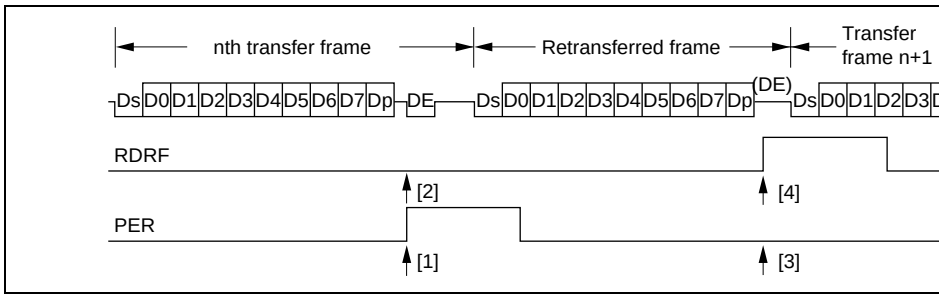


Figure 13.11 Retransfer Operation in SCI Receive Mode

is received.

[8] If an error signal is not sent back from the receiving end, the ERS bit in SSR is not

[9] If an error signal is not sent back from the receiving end, transmission of one fram a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1 bit in SCR is enabled at this time, a TXI interrupt request is generated.

If data transfer by the DTC by means of the TXI source is enabled, the next data c to TDR automatically. When data is written to TDR by the DTC, the TDRE bit is automatically cleared to 0.

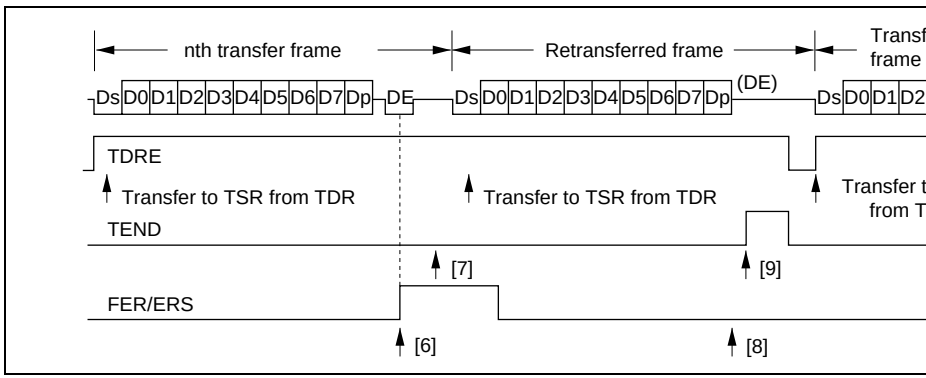


Figure 13.12 Retransfer Operation in SCI Transmit Mode

14.1.1 Features

A/D converter features are listed below

- 10-bit resolution
- Eight input channels
- Settable analog conversion voltage range
  - — Conversion of analog voltages with the reference voltage pin (V_{ref}) as the analo voltage
- High-speed conversion
  - — Minimum conversion time: 6.7 µs per channel (at 20-MHz operation)
- Choice of single mode or scan mode
  - — Single mode: Single-channel A/D conversion
  - — Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  - — Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  - — Choice of software or timer conversion start trigger (TPU or 8-bit timer), or $\overline{\text{AD}}$
- A/D conversion end interrupt generation
  - — A/D conversion end interrupt (ADI) request can be generated at the end of A/D
- Module stop mode can be set
  - — As the initial setting, A/D converter operation is halted. Register access is enab exiting module stop mode

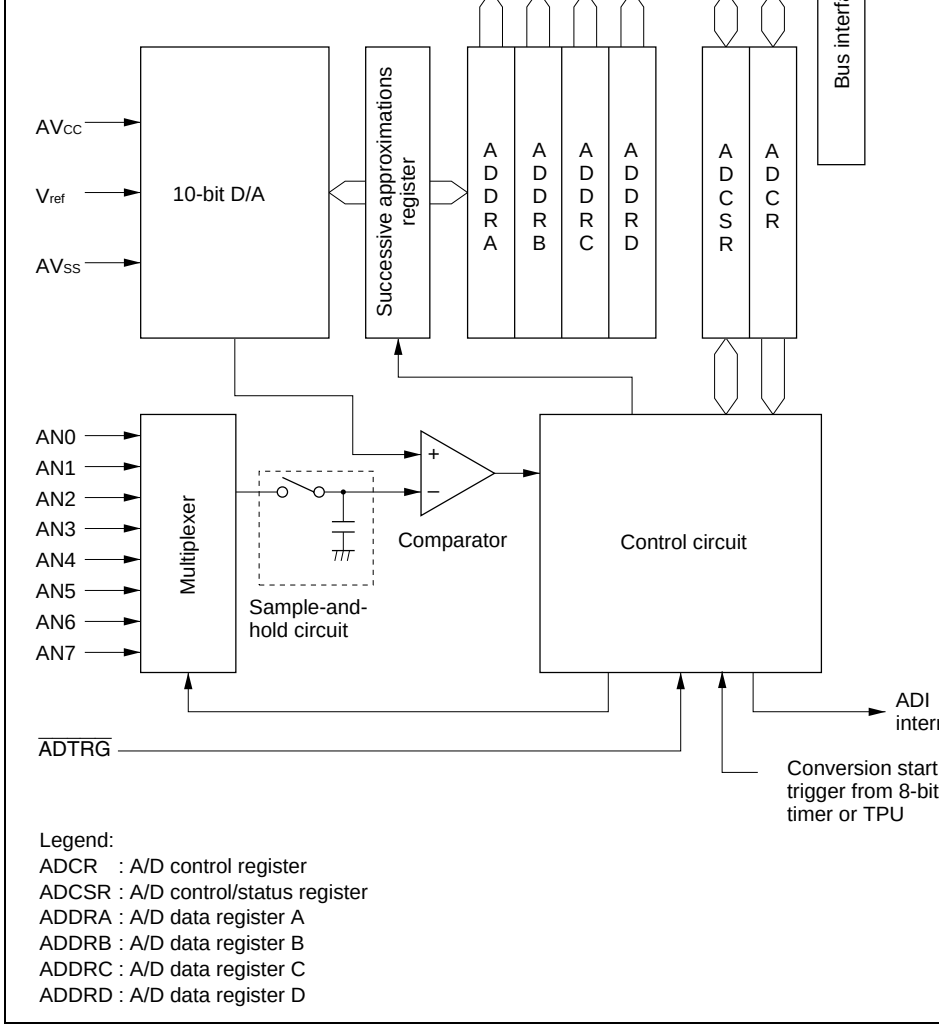


Figure 14.1 Block Diagram of A/D Converter

Table 14.1 A/D Converter Pins

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog block power supply
Analog ground pin	AV_{SS}	Input	Analog block ground and A/D reference voltage
Reference voltage pin	V_{ref}	Input	A/D conversion reference volt
Analog input pin 0	AN0	Input	Group 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Group 1 analog inputs
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for start conversion

A/D data register BH	ADDRBH	R	H'00	H'F
A/D data register BL	ADDRBL	R	H'00	H'F
A/D data register CH	ADDRCH	R	H'00	H'F
A/D data register CL	ADDRCL	R	H'00	H'F
A/D data register DH	ADDRDH	R	H'00	H'F
A/D data register DL	ADDRDL	R	H'00	H'F
A/D control/status register	ADCSR	R/(W)*2	H'00	H'F
A/D control register	ADCR	R/W	H'3F	H'F
Module stop control register	MSTPCR	R/W	H'3FFF	H'F

Notes: 1. Lower 16 bits of the address.
2. Bit 7 can only be written with 0 for flag clearing.

R/W . R R R R R R R R R R R R R R

There are four 16-bit read-only ADDR registers, ADDRA to ADDRD, used to store th
A/D conversion.

The 10-bit data resulting from A/D conversion is transferred to the ADDR register for
channel and stored there. The upper 8 bits of the converted data are transferred to the
(bits 15 to 8) of ADDR, and the lower 2 bits are transferred to the lower byte (bits 7 a
stored. Bits 5 to 0 are always read as 0.

The correspondence between the analog input channels and ADDR registers is shown
14.3.

ADDR can always be read by the CPU. The upper byte can be read directly, but for th
byte, data transfer is performed via a temporary register (TEMP). For details, see secti
Interface to Bus Master.

The ADDR registers are initialized to H'0000 by a reset, and in standby mode or modu
mode.

Table 14.3 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

Note: * Only 0 can be written to bit 7, to clear this flag.

ADCSR is an 8-bit readable/writable register that controls A/D conversion operations a
the status of the operation.

ADCSR is initialized to H'00 by a reset, and in hardware standby mode or module stop

Bit 7—A/D End Flag (ADF): Status flag that indicates the end of A/D conversion.

Bit 7

ADF	Description
0	[Clearing conditions] (• When 0 is written to the ADF flag after reading ADF = 1 • When the DTC is activated by an ADI interrupt and ADDR is read
1	[Setting conditions] • Single mode: When A/D conversion ends • Scan mode: When A/D conversion ends on all specified channels

Bit 6—A/D Interrupt Enable (ADIE): Selects enabling or disabling of interrupt (ADI
at the end of A/D conversion.

Bit 6

ADIE	Description
0	A/D conversion end interrupt (ADI) request disabled (
1	A/D conversion end interrupt (ADI) request enabled

0	A/D conversion stopped
1	• Single mode: A/D conversion is started. Cleared to 0 automatically wh conversion on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues sequer selected channels until ADST is cleared to 0 by softwar a transition to standby mode or module stop mode.

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode as the A/D conversio mode. See section 14.4, Operation, for single mode and scan mode operation. Only se bit while conversion is stopped.

Bit 4

SCAN	Description
0	Single mode
1	Scan mode

Bit 3—Clock Select (CKS): Sets the A/D conversion time. Only change the conversi while ADST = 0.

Bit 3

CKS	Description
0	Conversion time = 266 states (max.)
1	Conversion time = 134 states (max.)

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): Together with the SCAN bit, thes the analog input channels.

Only set the input channel while conversion is stopped.

1	0	0	AN4	AN4
		1	AN5	AN4, AN5
	1	0	AN6	AN4 to AN6
		1	AN7	AN4 to AN7

14.2.3 A/D Control Register (ADCR)

Bit	:	7	6	5	4	3	2	1
		TRGS1	TRGS0	—	—	—	—	—
Initial value	:	0	0	1	1	1	1	1
R/W	:	R/W	R/W	—	—	R/W	—	—

ADCR is an 8-bit readable/writable register that enables or disables external triggering conversion operations.

ADCR is initialized to H'3F by a reset, and in standby mode or module stop mode.

Bits 7 and 6—Timer Trigger Select 1 and 0 (TRGS1, TRGS0): Select enabling or d the start of A/D conversion by a trigger signal. Only set bits TRGS1 and TRGS0 while is stopped.

Bit 7	Bit 6	
TRGS1	**TRGS0**	**Description**
0	0	A/D conversion start by external trigger is disabled (I
	1	A/D conversion start by external trigger (TPU) is enabled
1	0	A/D conversion start by external trigger (8-bit timer) is enabled
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enable

Bits 5 to 0—Reserved: These bits are reserved; write as 1 in a write.

MSTPCR is a 16-bit readable/writable register that performs module stop mode contro

When the MSTP9 bit in MSTPCR is set to 1, A/D converter operation stops at the end cycle and a transition is made to module stop mode. Registers cannot be read or writte module stop mode. For details, see section 19.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not in software standby mode.

Bit 9—Module Stop (MSTP9): Specifies the A/D converter module stop mode.

Bit 9	
MSTP9	**Description**
0	A/D converter module stop mode cleared
1	A/D converter module stop mode set

When reading ADDR. always read the upper byte before the lower byte. It is possible
the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 14.2 shows the data flow for ADDR access.

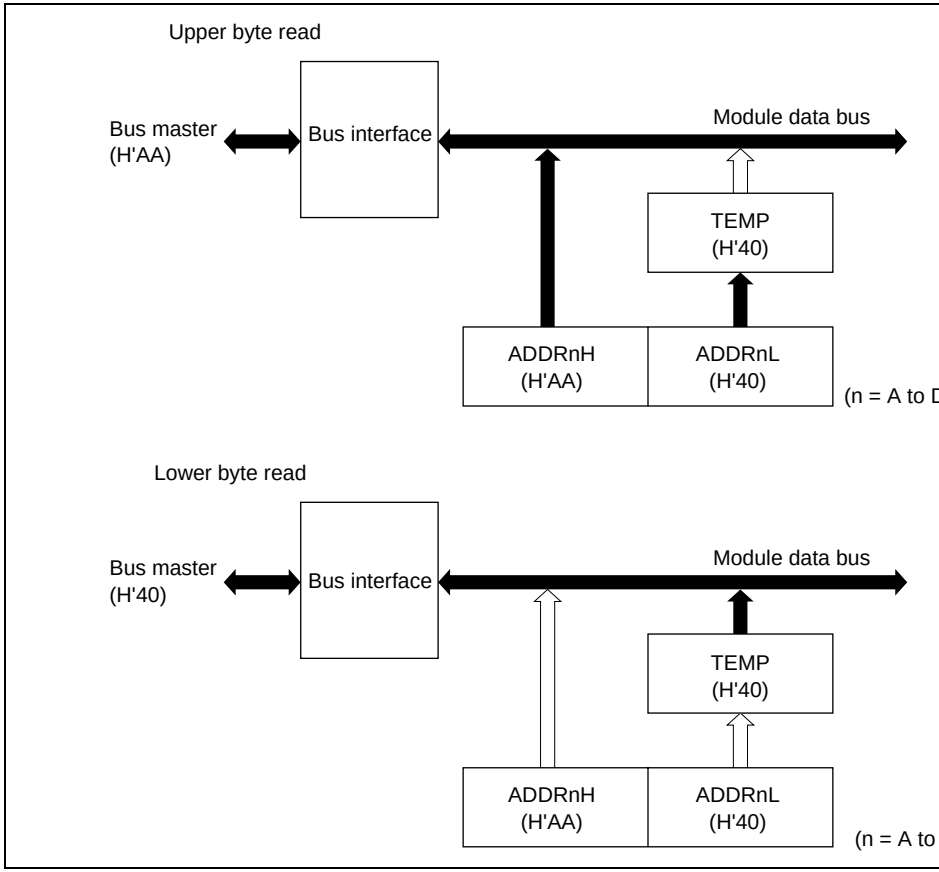


Figure 14.2 ADDR Access Operation (Reading H'AA40)

conversion is started when the ADST bit is set to 1, according to the software or exter
input. The ADST bit remains set to 1 during A/D conversion, and is automatically cle
when conversion ends.

On completion of conversion, the ADF flag is set to 1. If the ADIE bit is set to 1 at thi
ADI interrupt request is generated. The ADF flag is cleared by writing 0 after reading

When the operating mode or analog input channel must be changed during analog con
prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conve
making the necessary changes, set the ADST bit to 1 to start A/D conversion again. T
bit can be set at the same time as the operating mode or input channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described nex
14.3 shows a timing diagram for this example.

[1] Single mode is selected (SCAN = 0), input channel AN1 is selected (CH2 = 0, CH
CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started

[2] When A/D conversion is completed, the result is transferred to ADDRB. At the sa
ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes

[3] Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.

[4] The A/D interrupt handling routine starts.

[5] The routine reads ADCSR, then writes 0 to the ADF flag.

[6] The routine reads and processes the connection result (ADDRB).

[7] Execution of the A/D interrupt handling routine ends. After that, if the ADST bit i
A/D conversion starts again and steps [2] to [7] are repeated.

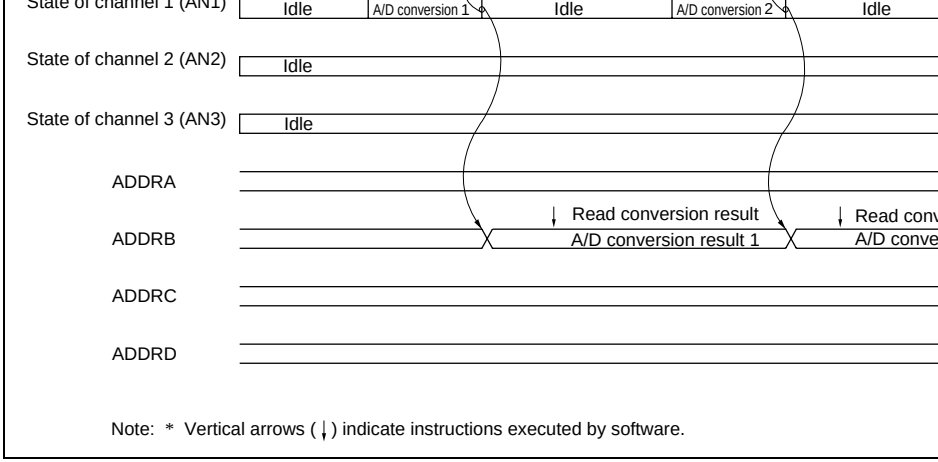


Figure 14.3 Example of A/D Converter Operation (Single Mode, Channel 1 S

When the operating mode or analog input channel must be changed during analog con
prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conve
making the necessary changes, set the ADST bit to 1 to start A/D conversion again. T
bit can be set at the same time as the operating mode or input channel is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are d
next. Figure 14.4 shows a timing diagram for this example.

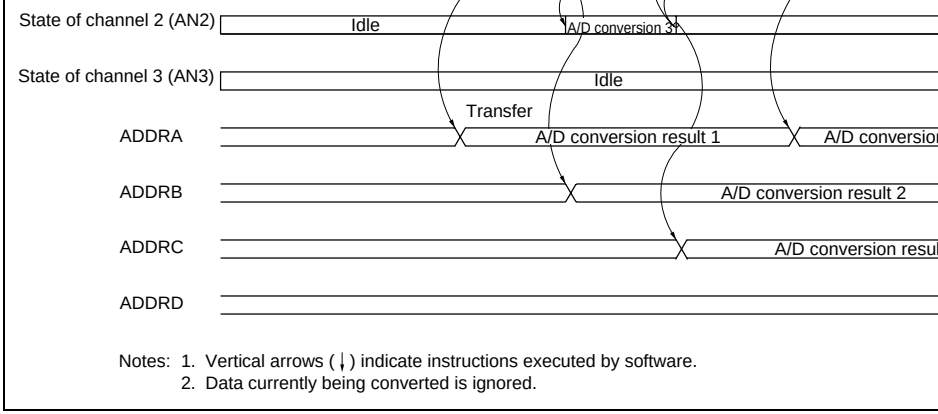
[1] Scan mode is selected (SCAN = 1), scan group 0 is selected (CH2 = 0), analog inp
AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (AD

[2] When A/D conversion of the first channel (AN0) is completed, the result is transfe
ADDRA. Next, conversion of the second channel (AN1) starts automatically.

[3] Conversion proceeds in the same way through the third channel (AN2).

[4] When conversion of all the selected channels (AN0 to AN2) is completed, the AD
to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set t
time, an ADI interrupt is requested after A/D conversion ends.

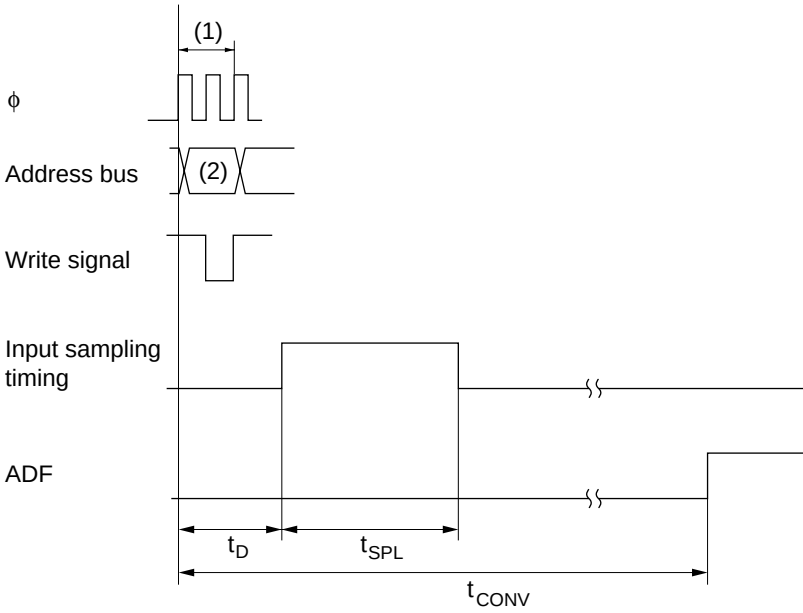
[5] Steps [2] to [4] are repeated as long as the ADST bit remains set to 1. When the A
cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D con
starts again from the first channel (AN0).



Notes: 1. Vertical arrows (↓) indicate instructions executed by software.
2. Data currently being converted is ignored.

Figure 14.4 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

In scan mode, the values given in table 14.4 apply to the first conversion time. In the subsequent conversions the conversion time is fixed at 256 states when CKS = 0 or 12 when CKS = 1.



Legend:
(1) : ADCSR write cycle
(2) : ADCSR address
t_D : A/D conversion start delay
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Figure 14.5 A/D Conversion Timing

14.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set
ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADT}}$
the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both singl
modes, are the same as if the ADST bit has been set to 1 by software. Figure 14.6 shov
timing.

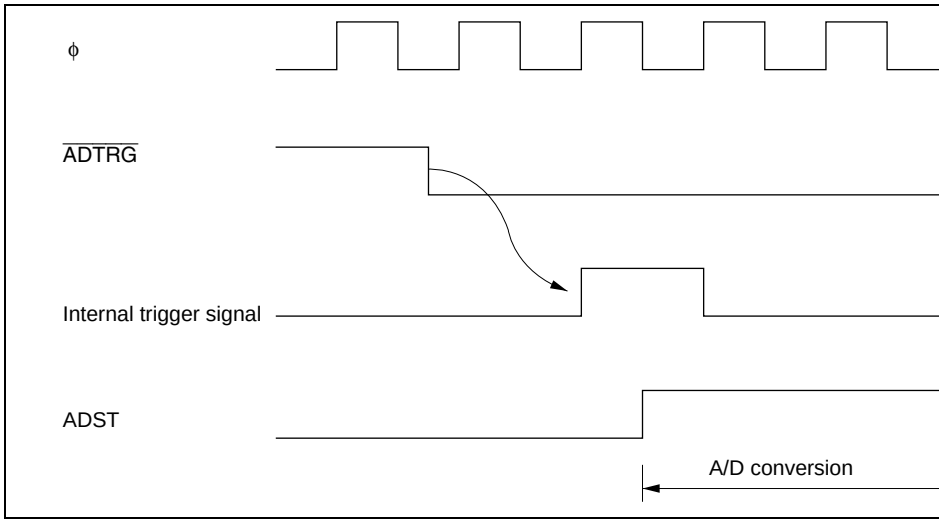


Figure 14.6 External Trigger Input Timing

The A/D converter interrupt source is shown in table 14.5.

Table 14.5 A/D Converter Interrupt Source

Interrupt Source	Description	DTC Activation
ADI	Interrupt due to end of conversion	Possible

14.6 Usage Notes

The following points should be noted when using the A/D converter.

Setting Range of Analog Power Supply and Other Pins:

(1) Analog input voltage range

The voltage applied to analog input pins AN0 to AN7 during A/D conversion shou
range $AV_{SS} \leq ANn \leq V_{ref}$.

(2) Relation between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}

As the relationship between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}, set, $AV_{CC} = V_{CC}$ and $AV_{SS} =$
A/D converter is not used, the AV_{CC} and AV_{SS} pins must on no account be left oper

(3) V_{ref} input range

The analog reference voltage input at the V_{ref} pin set in the range $V_{ref} \leq AV_{CC}$.

If conditions (1), (2), and (3) above are not met, the reliability of the device may be ad
affected.

Notes on Board Design: In board design, digital circuitry and analog circuitry should
mutually isolated as possible, and layout in which digital circuit signal lines and analo
signal lines cross or are in close proximity should be avoided as far as possible. Failu
may result in incorrect operation of the analog circuitry due to inductance, adversely a
conversion values.

14.7.

Also, the bypass capacitors connected to AV_{CC} and V_{ref} and the filter capacitor connecte to AN7 must be connected to AV_{SS}.

If a filter capacitor is connected as shown in figure 14.7, the input currents at the analog (AN0 to AN7) are averaged, and so an error may arise. Also, when A/D conversion is frequently, as in scan mode, if the current charged and discharged by the capacitance of sample-and-hold circuit in the A/D converter exceeds the current input via the input im (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefo when deciding the circuit constants.

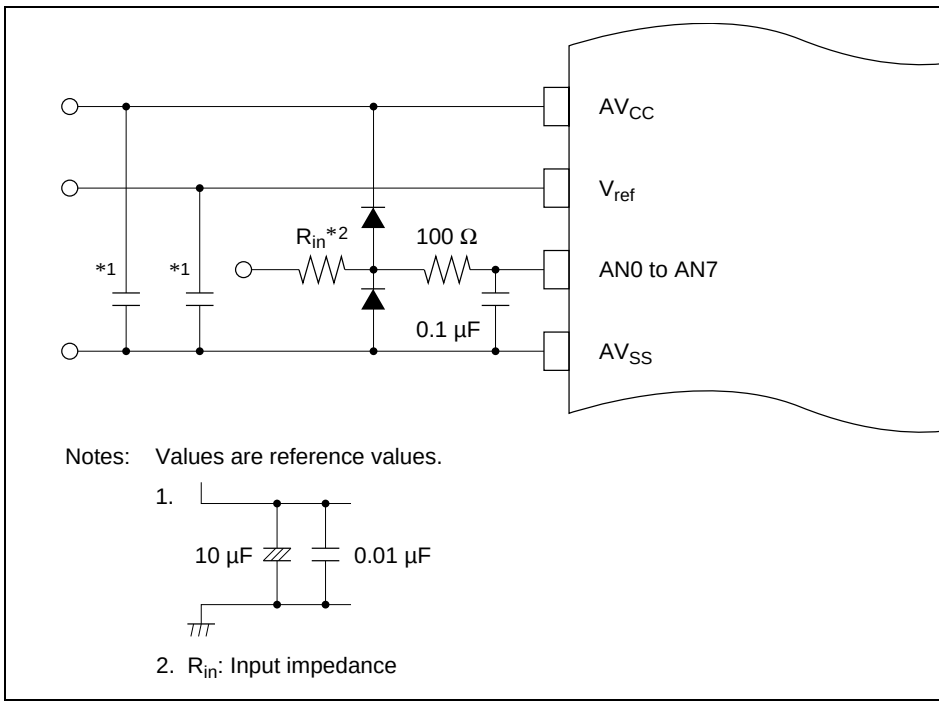
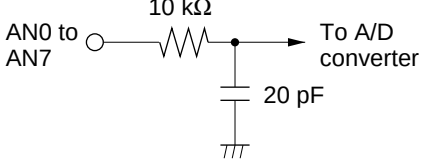


Figure 14.7 Example of Analog Input Protection Circuit



Note: Values are reference values.

Figure 14.8 Analog Input Pin Equivalent Circuit

A/D Conversion Precision Definitions: H8S/2345 Group A/D conversion precision d
are given below.

- Resolution

  The number of A/D converter digital output codes
- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion cha
  when the digital output changes from the minimum voltage value B'0000000000 (
  B'0000000001 (H'001) (see figure 14.10).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion cha
  when the digital output changes from B'1111111110 (H'3FE) to B'1111111111 (H
  figure 14.10).
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 14.9).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero
  the full-scale voltage. Does not include the offset error, full-scale error, or quantiz
- Absolute precision

  The deviation between the digital value and the analog input value. Includes the o
  full-scale error, quantization error, and nonlinearity error.

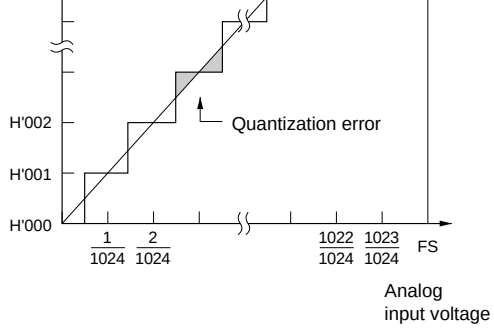


Figure 14.9 A/D Conversion Precision Definitions (1)

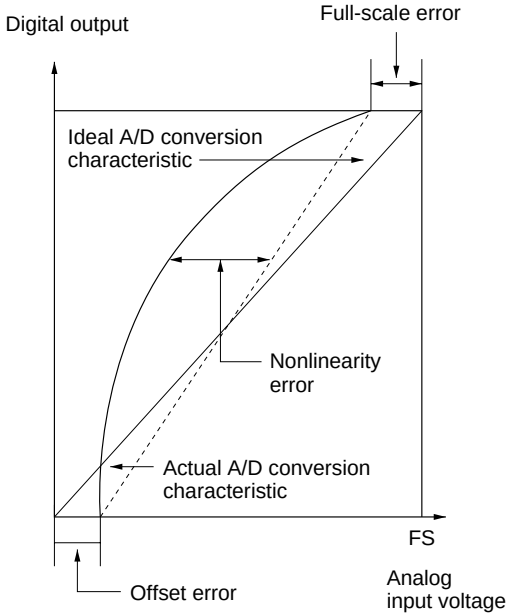


Figure 14.10 A/D Conversion Precision Definitions (2)

only the internal input resistance of 10 kΩ, and the signal source impedance is ignored

However, since a low-pass filter effect is obtained in this case, it may not be possible t
analog signal with a large differential coefficient (e.g., 5 mV/µs or greater).

When converting a high-speed analog signal, a low-impedance buffer should be insert

Influences on Absolute Precision: Adding capacitance results in coupling with GND
therefore noise in GND may adversely affect absolute precision. Be sure to make the
to an electrically stable GND such as AV_{SS}.

Care is also required to insure that filter circuits do not communicate with digital signa
mounting board, so acting as antennas.

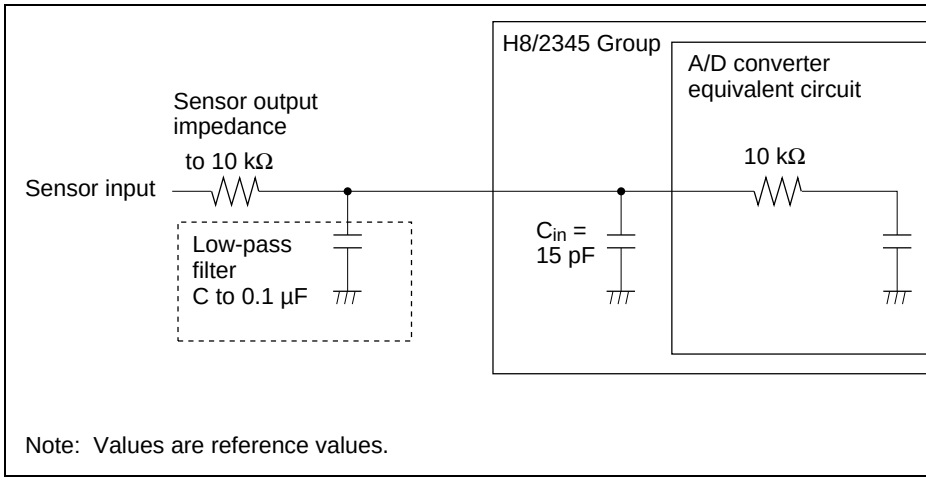


Figure 14.11 Example of Analog Input Circuit

D/A converter features are listed below

- 8-bit resolution
- Two output channels
- Maximum conversion time of 10 µs (with 20 pF load)
- Output voltage of 0 V to V_{ref}
- D/A output hold function in software standby mode

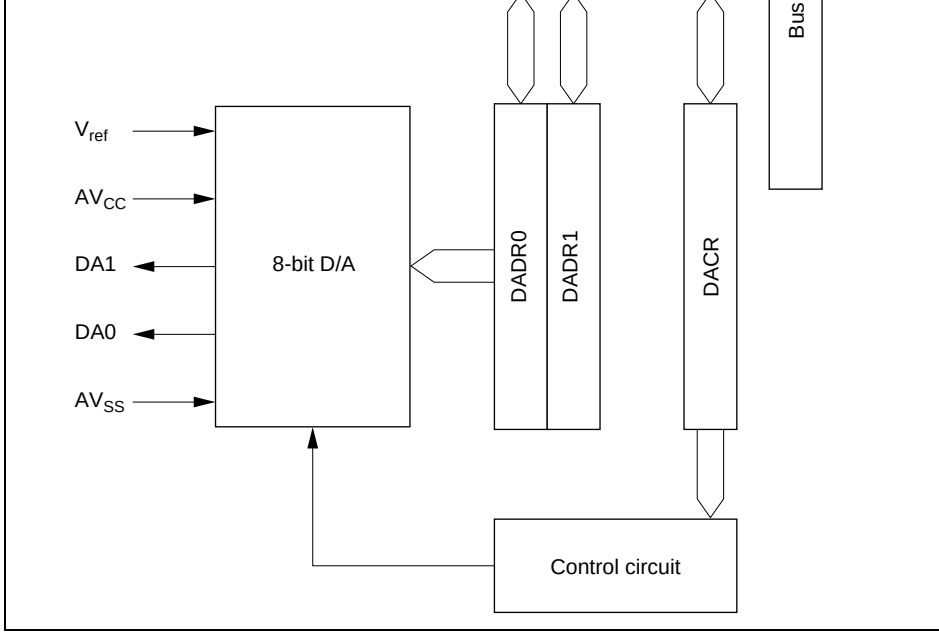


Figure 15.1 Block Diagram of D/A Converter

Analog output pin 0	DA0	Output	Channel 0 analog output
Analog output pin 1	DA1	Output	Channel 1 analog output
Reference voltage pin	V_{ref}	Input	Analog reference voltage

15.1.4 Register Configuration

Table 15.2 summarizes the registers of the D/A converter.

Table 15.2 D/A Converter Registers

Name	Abbreviation	R/W	Initial Value	A
D/A data register 0	DADR0	R/W	H'00	H
D/A data register 1	DADR1	R/W	H'00	H
D/A control register	DACR	R/W	H'1F	H
Module stop control register	MSTPCR	R/W	H'3FFF	H

Note: * Lower 16 bits of the address.

R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DADR0 and DADR1 are 8-bit readable/writable registers that store data for conversion

Whenever output is enabled, the values in DADR0 and DADR1 are converted and outp
analog output pins.

DADR0 and DADR1 are each initialized to H'00 by a reset and in hardware standby m

15.2.2 D/A Control Register (DACR)

Bit	:	7	6	5	4	3	2	1
		DAOE1	DAOE0	DAE	—	—	—	—
Initial value	:	0	0	0	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 conv

DACR is initialized to H'1F by a reset and in hardware standby mode.

Bit 7—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output
1.

Bit 7

DAOE1	Description
0	Analog output DA1 is disabled
1	Channel 1 D/A conversion is enabled; analog output DA1 is enabled

Bit 5—D/A Enable (DAE): The DAOE0 and DAOE1 bits both control D/A conversi
the DAE bit is cleared to 0, the channel 0 and 1 D/A conversions are controlled indepe
When the DAE bit is set to 1, the channel 0 and 1 D/A conversions are controlled toge

Output of resultant conversions is always controlled independently by the DAOE0 and
bits.

Bit 7	Bit 6	Bit 5	
DAOE1	**DAOE0**	**DAE**	**Description**
0	0	*	Channel 0 and 1 D/A conversions disabled
	1	0	Channel 0 D/A conversion enabled Channel 1 D/A conversion disabled
		1	Channel 0 and 1 D/A conversions enabled
1	0	0	Channel 0 D/A conversion disabled Channel 1 D/A conversion enabled
		1	Channel 0 and 1 D/A conversions enabled
	1	*	Channel 0 and 1 D/A conversions enabled

If the H8S/2345 Group enters software standby mode when D/A conversion is enable
output is held and the analog power current is the same as during D/A conversion. Wh
necessary to reduce the analog power current in software standby mode, clear the DA
and DAOE1 bits to 0 to disable D/A output.

Bits 4 to 0—Reserved: Read-only bits, always read as 1.

MSTPCR is a 16-bit readable/writable register that performs module stop mode control

When the MSTP10 bit in MSTPCR is set to 1, D/A converter operation stops at the end cycle and a transition is made to module stop mode. Registers cannot be read or written module stop mode. For details, see section 19.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not ini software standby mode.

Bit 10—Module Stop (MSTP10): Specifies the D/A converter module stop mode.

Bit 10

MSTP10	Description
0	D/A converter module stop mode cleared
1	D/A converter module stop mode set (I

The operation example described in this section concerns D/A conversion on channel
15.2 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 beco
output pin. The conversion result is output after the conversion time has elapsed. T
value is expressed by the following formula:

$$\frac{\text{DADR contents}}{256} \times V_{ref}$$

The conversion results are output continuously until DADR0 is written to again or
bit is cleared to 0.

[3] If DADR0 is written to again, the new data is immediately converted. The new co
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.

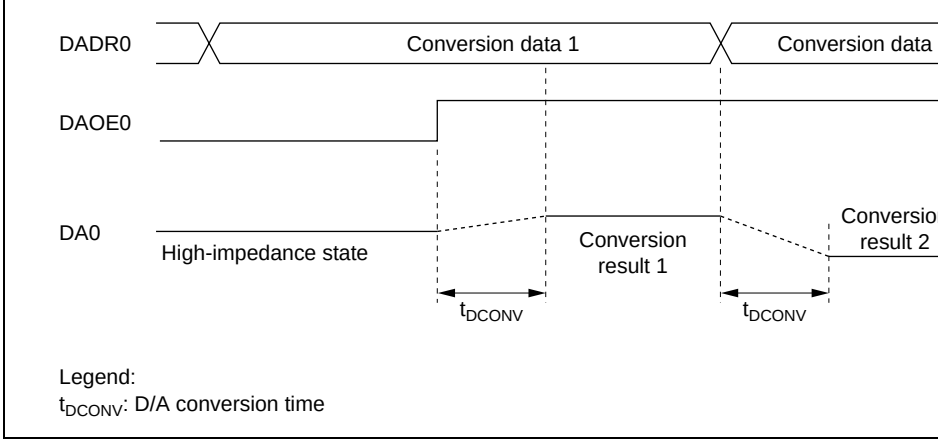


Figure 15.2 Example of D/A Converter Operation

15.4 Usage Notes

Setting range for pins other than analog power pin

(1) Relationship between AV_{CC}, V_{CC}, AV_{SS}, and Vss

The relationship between AV_{CC}, V_{CC}, AV_{SS}, and V_{SS} is $AV_{CC} = V_{CC}$ and $AV_{SS} = V_{SS}$. A AV_{CC} and AV_{SS} pins should never be left open, even if the D/A converter is not used

(2) Vref setting range

The setting range for the reference voltage from the Vref pin is Vref ≤ AV_{CC}.

Note: Failure to observe (1) and (2) above could have an adverse effect on the reliabi LSI.

perform fast word data transfer.

The on-chip RAM of the H8S/2345 and H8S/2344 is allocated addresses H'EC00 to H kbytes) in the normal modes (modes 1 to 3)*, and addresses H'FFEC00 to H'FFFBFF the advanced modes (modes 4 to 7).

The on-chip RAM of the H8S/2343, H8S/2341, and H8S/2340 is allocated addresses H H'FBFF (2 kbytes) in the normal modes (modes 1 to 3)*, and addresses H'FFF400 to H kbytes) in the advanced modes (modes 4 to 7).

The on-chip RAM can be enabled or disabled by means of the RAM enable bit (RAM system control register (SYSCR).

Note: * ZTAT, mask ROM, and ROMless versions only.

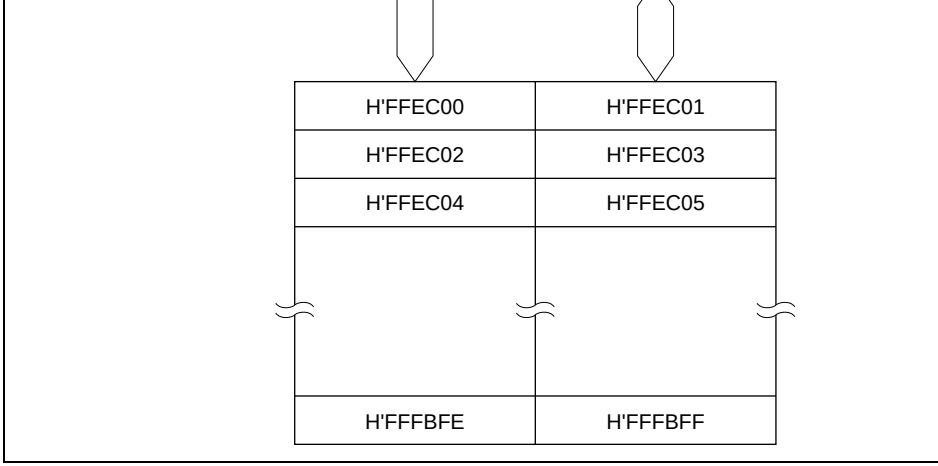


Figure 16.1 Block Diagram of RAM (H8S/2345, Advanced Mode)

16.1.2 Register Configuration

The on-chip RAM is controlled by SYSCR. Table 16.1 shows the address and initial v
SYSCR.

Table 16.1 RAM Register

Name	Abbreviation	R/W	Initial Value	A
System control register	SYSCR	R/W	H'01	H'

Note: * Lower 16 bits of the address.

R/W . R/W 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 c
SYSCR, see section 3.2.2, System Control Register (SYSCR).

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

Bit 0

RAME	Description
0	On-chip RAM is disabled
1	On-chip RAM is enabled

16.3 Operation

When the RAME bit is set to 1, accesses to addresses H'FFEC00 to H'FFFBFF (in the
H8S/2345 and H8S/2344) or addresses H'FFF400 to H'FFFBFF (in the case of the H8
H8S/2341, and H8S/2340) are directed to the on-chip RAM. When the RAME bit is c
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
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
at an even address.

16.4 Usage Note

DTC register information can be located in addresses H'FFF800 to H'FFFBFF. When
used, the RAME bit must not be cleared to 0.

connected to the H8S/2000 CPU by a 16-bit data bus. The CPU accesses both byte da
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 (MD_2, MD_1, and M
EAE in BCRL.

The flash memory versions of the H8S/2345 Group can be erased and programmed on
well as with a PROM programmer.

The PROM version of the H8S/2345 Group can be programmed with a PROM progra
setting PROM mode.

17.1.1 Block Diagram

Figure 17.1 shows a block diagram of the on-chip ROM.

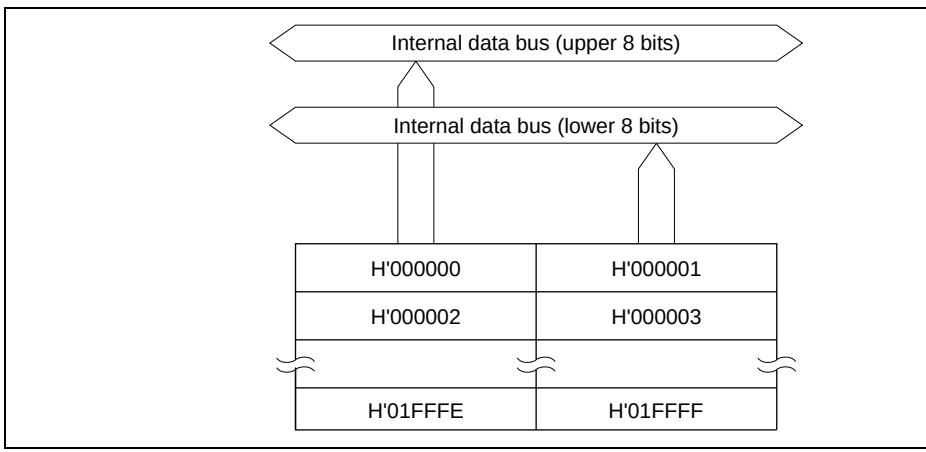


Figure 17.1 Block Diagram of ROM (H8S/2345)

Bus control register L	BCRL	R/W	Undefined	H'FE

Note: * Lower 16 bits of the address.

17.2 Register Descriptions

17.2.1 Mode Control Register (MDCR)

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

Note: * Determined by pins MD_2 to MD_0.

MDCR is an 8-bit read-only register that indicates the current operating mode of the H8 Group.

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 level MD_2 to MD_0 (the current operating mode). Bits MDS2 to MDS0 correspond to pins MD MDS2 to MDS0 are read-only bits, and cannot be written to. The mode pin (MD_2 to M levels are latched into these bits when MDCR is read. These latches are canceled by a p reset, but are retained after a manual reset.

Enabling or disabling of part of the H8S/2345 on-chip ROM area can be selected by
the EAE bit in BCRL. For details of the other bits in BCRL, see section 6.2.5, Bus Co
Register L (BCRL).

Bit 5—External Address Enable (EAE): Selects whether addresses H'010000 to H'0
to be internal addresses or external addresses.

Bit 5 EAE	Description
0	Addresses H'010000 to H'01FFFF are in on-chip ROM (H8S/2345).
	Addresses H'010000 to H'017FFF are in on-chip ROM and addresses H'0 H'01FFFF are a reserved area (in the H8S/2344).
	Addresses H'010000 to H'01FFFF are a reserved area (in the H8S/2343 a H8S/2341).
1	Addresses H'010000 to H'01FFFF are external addresses (external expan or a reserved area* (single-chip mode).

Note: * Reserved areas should not be accessed.

17.3 Operation

The on-chip ROM is connected to the CPU by a 16-bit data bus, and both byte and wo
be accessed in one state. Even addresses are connected to the upper 8 bits, and odd ad
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 (MD_2, MD_1, and M
EAE in BCRL. These settings are shown in tables 17.2 and 17.3.

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	0	Enabled (12
						1	Enabled (64
Mode 7	Advanced single-chip mode				1	0	Enabled (12
						1	Enabled (64
Mode 8	—	1	0	0	0	—	—
Mode 9					1		
Mode 10	Boot mode (advanced expanded mode with on-chip ROM enabled)*3			1	0	0	Enabled (12
						1	Enabled (64
Mode 11	Boot mode (advanced single-chip mode)*4				1	0	Enabled (12
						1	Enabled (64
Mode 12	—		1	0	0	—	—
Mode 13					1		
Mode 14	User program mode (advanced expanded mode with on-chip ROM enabled)*3			1	0	0	Enabled (12
						1	Enabled (64
Mode 15	User program mode (advanced single-chip mode)*4				1	0	Enabled (12
						1	Enabled (64

Notes: 1. Note that in modes 6, 7, 14, and 15, the on-chip ROM that can be used after on reset is the 64-kbyte area from H'000000 to H'00FFFF.
2. Note that in the mode 10 and mode 11 boot modes, the on-chip ROM that ca immediately after all flash memory is erased by the boot program is the 64-k from H'000000 to H'00FFFF.
3. Apart from the fact that flash memory can be erased and programmed, opera same as in advanced expanded mode with on-chip ROM enabled.
4. Apart from the fact that flash memory can be erased and programmed, opera same as in advanced single-chip mode.

	mode with on-chip ROM enabled							
Mode 3[*1]	Normal single-chip mode			1		Enabled (56 kbytes)	Enabled (56 kbytes)	Enabled (56 kbyte
Mode 4	Advanced expanded mode with on-chip ROM disabled	1	0	0	—	Disabled	Disabled	Disabled
Mode 5	Advanced expanded mode with on-chip ROM disabled			1				
Mode 6[*1]	Advanced expanded mode with on-chip ROM enabled		1	0	0	Enabled (128 kbytes)[*2]	Enabled[*2] (96 kbytes)	Enabled (64 kbyte
					1	Enabled (64 kbytes)	Enabled (64 kbytes)	
Mode 7[*1]	Advanced single-chip mode			1	0	Enabled (128 kbytes)[*2]	Enabled[*2] (96 kbytes)	
					1	Enabled (64 kbytes)	Enabled (64 kbytes)	

Notes: 1. Not used on ROMless version.

2. In H8S/2345 modes 6 and 7, the on-chip ROM available after a power-on r
64-kbyte area comprising addresses H'000000 to H'00FFFF.

a 100-pin/32-pin socket adapter enables programming with a commercial PROM progr

Note that the PROM programmer should not be set to page mode as the H8S/2345 does
support page programming.

Table 17.4 shows how PROM mode is selected.

Table 17.4 Selecting PROM Mode

Pin Names	Setting
MD_2, MD_1, MD_0	Low
$\overline{STBY}$	
PA_2, PA_1	High

17.4.2 Socket Adapter and Memory Map

Programs can be written and verified by attaching a socket adapter to the PROM progra
convert from a 100-pin arrangement to a 32-pin arrangement. Table 17.5 gives ordering
information for the socket adapter, and figure 17.2 shows the wiring of the socket adap
17.3 shows the memory map in PROM mode.

28	30	PD_5	EO_5	19
29	31	PD_6	EO_6	20
30	32	PD_7	EO_7	21
32	34	PC_0	EA_0	12
33	35	PC_1	EA_1	11
34	36	PC_2	EA_2	10
35	37	PC_3	EA_3	9
36	38	PC_4	EA_4	8
37	39	PC_5	EA_5	7
38	40	PC_6	EA_6	6
39	41	PC_7	EA_7	5
41	43	PB_0	EA_8	27
63	65	NMI	EA_9	26
43	45	PB_2	EA_{10}	23
44	46	PB_3	EA_{11}	25
45	47	PB_4	EA_{12}	4
46	48	PB_5	EA_{13}	28
47	49	PB_6	EA_{14}	29
48	50	PB_7	EA_{15}	3
50	52	PA_0	EA_{16}	2
74	76	PF_2	$\overline{CE}$	22
42	44	PB_1	$\overline{OE}$	24
75	77	PF_1	$\overline{PGM}$	31
40, 65, 98	42, 67, 100	V_{CC}	V_{CC}	32
77	79	AV_{CC}		
78	80	V_{ref}		
51	53	PA_1		
52	54	PA_2		
7, 18, 31, 49, 68, 88	9, 20, 33 51, 70, 90	V_{SS}	V_{SS}	16
87	89	AV_{SS}		
64	66	$\overline{STBY}$		
57	59	MD_0		
58	60	MD_1		
61	63	MD_2		

Legend:
V_{PP} : Programming power supply (12.5 V)
EO_7 to EO_0 : Data input/output
EA_{16} to EA_0 : Address input
$\overline{OE}$: Output enable
$\overline{CE}$: Chip enable
$\overline{PGM}$: Program

Note: Pins not shown in this figure should be left open.

Figure 17.2 Wiring of 100-Pin/32-Pin Socket Adapter

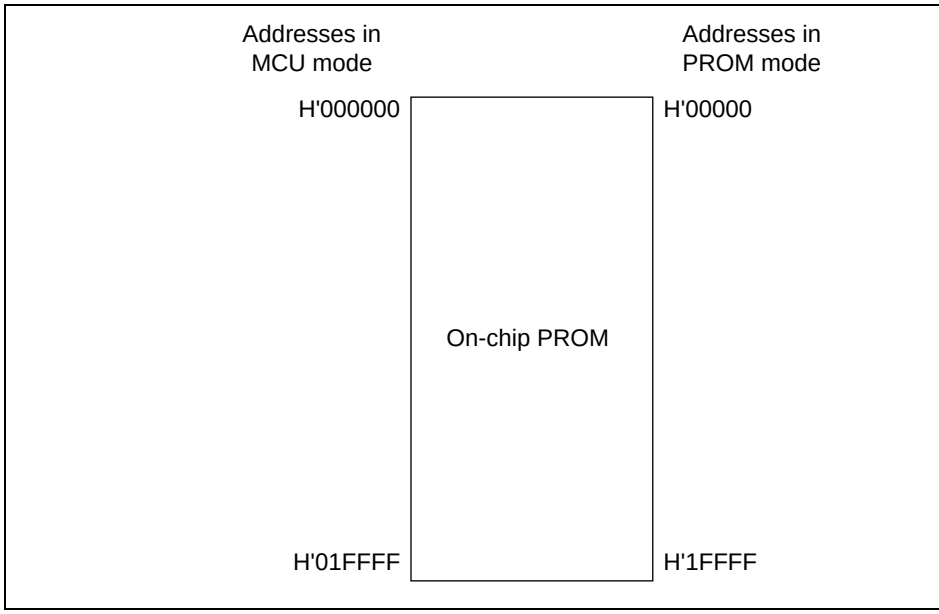


Figure 17.3 Memory Map in PROM Mode

	Pins						
Mode	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{PGM}$**	**V_{PP}**	**V_{CC}**	**EO_7 to EO_0**	**EA_{16}**
Program	L	H	L	V_{PP}	V_{CC}	Data input	Addr
Verify	L	L	H	V_{PP}	V_{CC}	Data output	Addr
Program-inhibit	L	L	L	V_{PP}	V_{CC}	High impedance	Addr
	L	H	H				
	H	L	L				
	H	H	H				

Legend:
L: Low voltage level
H: High voltage level
V_{PP}: V_{PP} voltage level
V_{CC}: V_{CC} voltage level

Programming and verification should be carried out using the same specifications as f
standard HN27C101 EPROM.

However, do not set the PROM programmer to page mode, as the H8S/2345 does not
programming. A PROM programmer that only supports page programming cannot be
choosing a PROM programmer, check that it supports high-speed programming in byt
Always set addresses within the range H'00000 to H'1FFFF.

17.5.2 Programming and Verification

An efficient, high-speed programming procedure can be used to program and verify P
This procedure writes data quickly without subjecting the chip to voltage stress or sac
reliability. It leaves the data H'FF in unused addresses. Figure 17.4 shows the basic h
programming flowchart. Tables 17.7 and 17.8 list the electrical characteristics of the
programming. Figure 17.5 shows a timing chart.

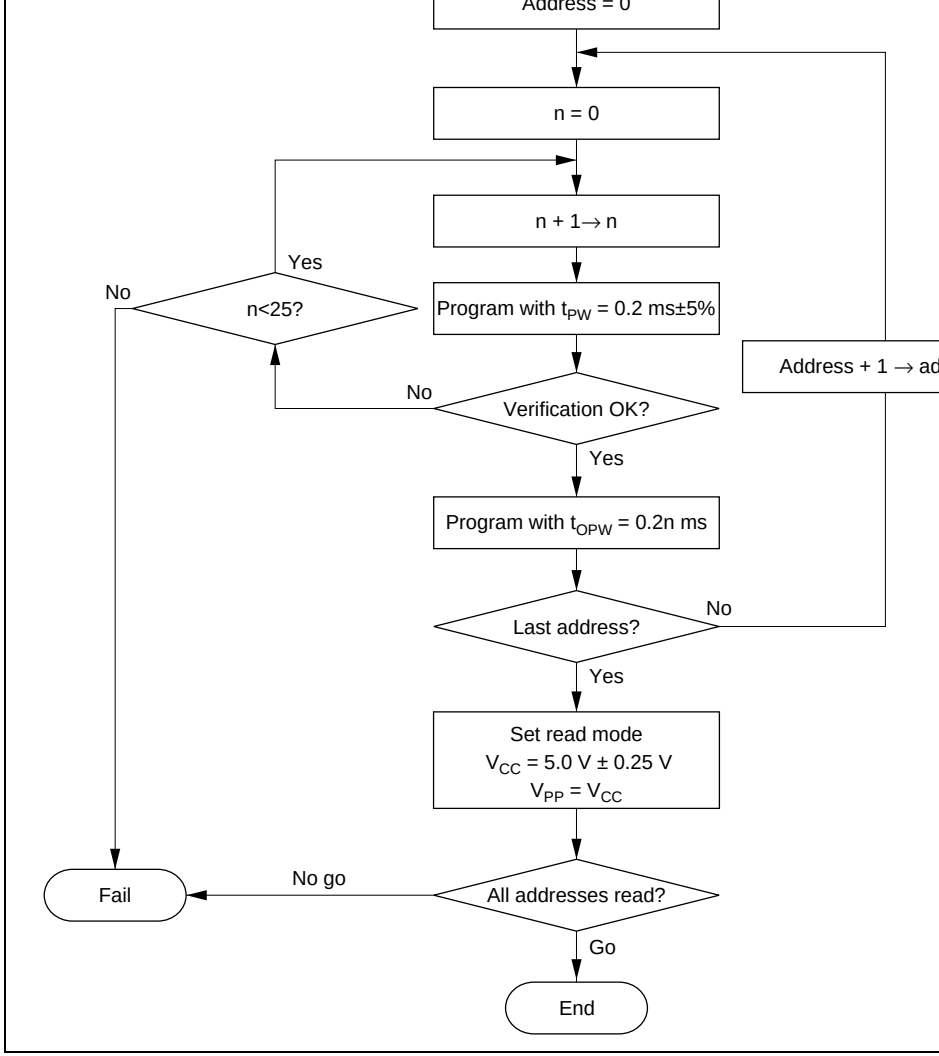


Figure 17.4 High-Speed Programming Flowchart

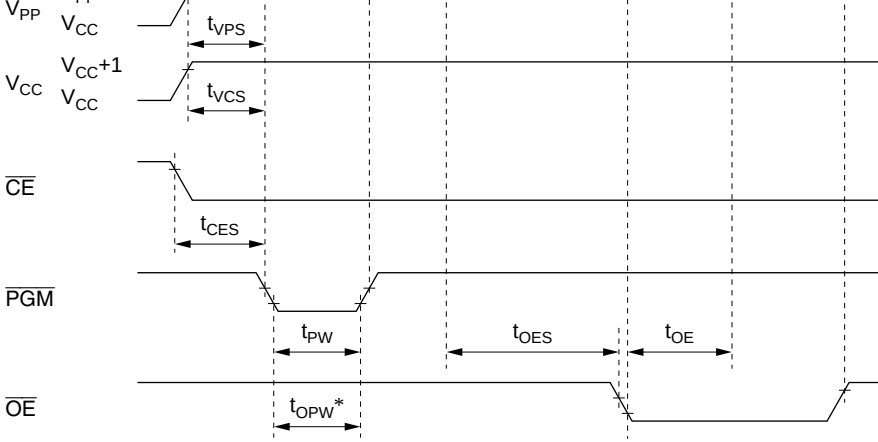
	EA_{16} to EA_0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$						
Output high voltage	EO_7 to EO_0	V_{OH}	2.4	—	—	V	I_O
Output low voltage	EO_7 to EO_0	V_{OL}	—	—	0.45	V	I_O
Input leakage current	EO_7 to EO_0, EA_{16} to EA_0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	$\lvert I_{LI} \rvert$	—	—	2	µA	V 5.
V_{CC} current		I_{CC}	—	—	40	mA	
V_{PP} current		I_{PP}	—	—	40	mA	

Address hold time	t_{AH}	0	—	—	µs
Data hold time	t_{DH}	2	—	—	µs
Data output disable time	t_{DF} *2	—	—	130	ns
V_{PP} setup time	t_{VPS}	2	—	—	µs
Programming pulse width	t_{PW}	0.19	0.20	0.21	ms
$\overline{PGM}$ pulse width for overwrite programming	t_{OPW} *3	0.19	—	5.25	ms
V_{CC} setup time	t_{VCS}	2	—	—	µs
$\overline{CE}$ setup time	t_{CES}	2	—	—	µs
Data output delay time	t_{OE}	0	—	150	ns

Notes: 1. Input pulse level: 0.8 V to 2.2 V
Input rise time and fall time ≤ 20 ns
Timing reference levels: Input: 1.0 V, 2.0 V
Output: 0.8 V, 2.0 V

2. t_{DF} is defined to be when output has reached the open state, and the output l longer be referenced.

3. t_{OPW} is defined by the value shown in the flowchart.



Note: * t_{OPW} is defined by the value shown in the flowchart.

Figure 17.5 PROM Programming/Verification Timing

17.5.3 Programming Precautions

- Program using the specified voltages and timing.
  The programming voltage (V_{pp}) in PROM mode is 12.5 V.
  If the PROM programmer is set to Renesas Technology HN27C101 specifications
  12.5 V. Applied voltages in excess of the specified values can permanently destroy
  Be particularly careful about the PROM programmer's overshoot characteristics.

- Before programming, check that the MCU is correctly mounted in the PROM prog
  Overcurrent damage to the MCU can result if the index marks on the PROM progr
  socket adapter, and MCU are not correctly aligned.

- Do not touch the socket adapter or MCU while programming. Touching either of t
  cause contact faults and programming errors.

An effective way to assure the data retention characteristics of the programmed chips is them at 150°C, then screen them for data errors. This procedure quickly eliminates chip PROM memory cells prone to early failure.

Figure 17.6 shows the recommended screening procedure.

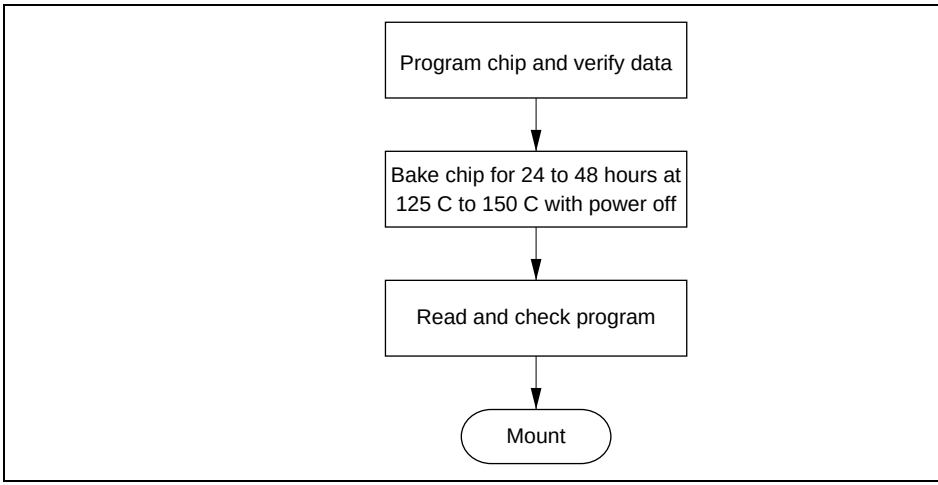


Figure 17.6 Recommended Screening Procedure

If a series of programming errors occurs while the same PROM programmer is being us programming and check the PROM programmer and socket adapter for defects.

Please inform Renesas Technology of any abnormal conditions noted during or after pr or in screening of program data after high-temperature baking.

— Erase mode
— Program-verify mode
— Erase-verify mode

- Programming/erase methods

  The flash memory is programmed 32 bytes at a time. Erasing is performed by bloc
  single-block units). When erasing multiple blocks, the individual blocks must be e
  sequentially. Block erasing can be performed as required on 1-kbyte, 8-kbyte, 16-k
  kbyte, and 32-kbyte blocks.

- Programming/erase times (5 V version)

  The flash memory programming time is 10 ms (typ.) for simultaneous 32-byte pro
  equivalent to 300 µs (typ.) per byte, and the erase time is 100 ms (typ.) per block.

- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.

- On-board programming modes

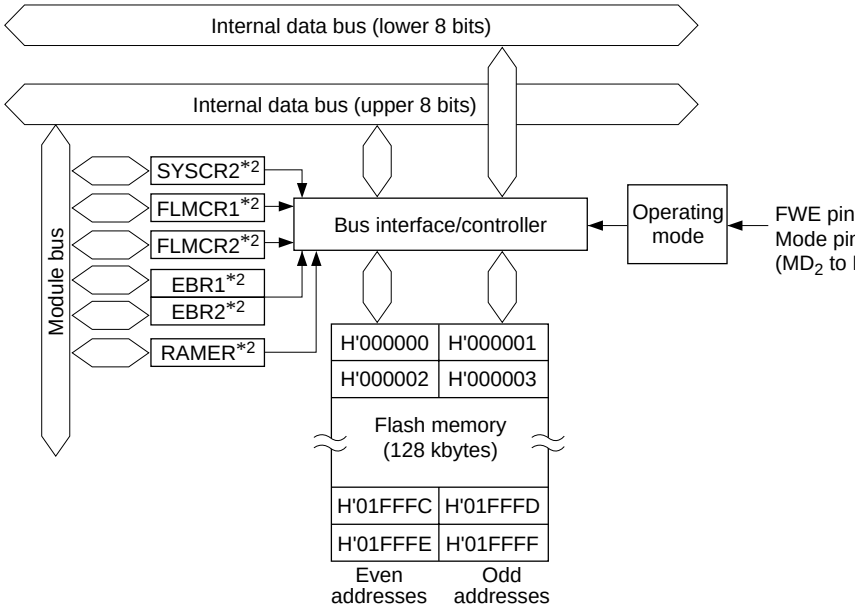
  There are two modes in which flash memory can be programmed/erased/verified o
  — Boot mode
  — User program mode

- Automatic bit rate adjustment

  With data transfer in boot mode, the bit rate of the H8S/2345 Group chip can be au
  adjusted to match the transfer bit rate of the host. (9600 bps, 4800 bps)

- Flash memory emulation by RAM

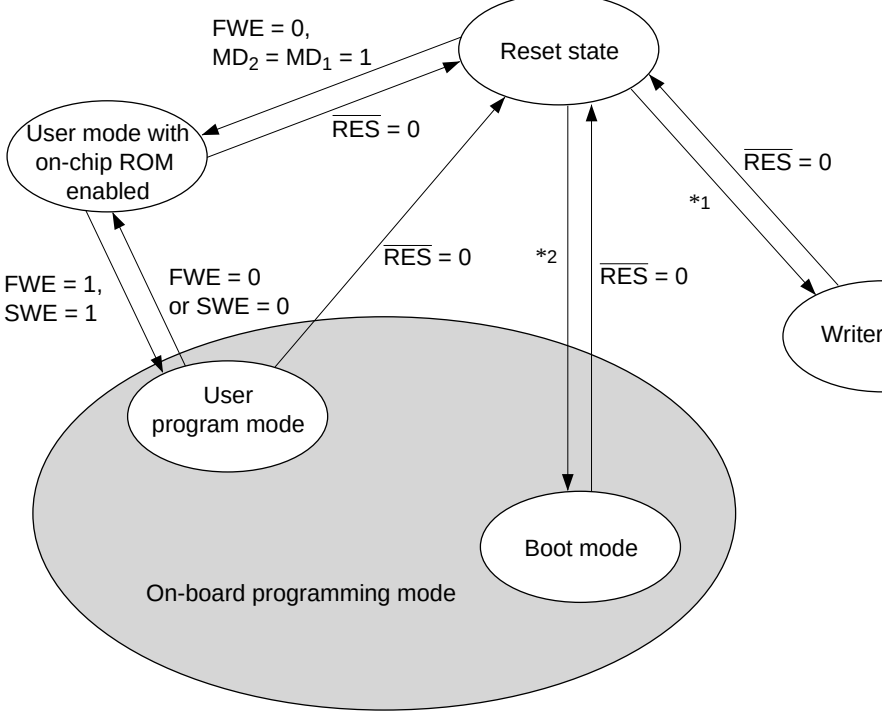
  Part of the RAM area can be overlapped onto flash memory, to emulate flash mem
  in real time.



Legend:
SYSCR2: System control register 2
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

Notes: 1. Functions as FWE pin on F-ZTAT version. Functions as $\overline{\text{WDTOVF}}$ pin on ZTAT, mask ROM, and ROMless versions.
2. The flash memory control registers (SYSCR2, FLMCR1, FLMCR2, EBR1, EBR2, RAMER) are enabled on the F-ZTAT version only. They do not exist on the ZTAT mask ROM, and ROMless versions, so an undefined value will be returned if they are read, and it is not possible to write to these registers.

Figure 17.7 Block Diagram of Flash Memory

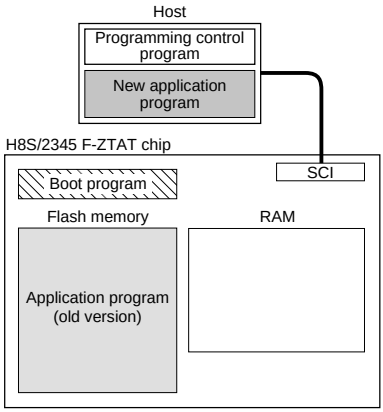


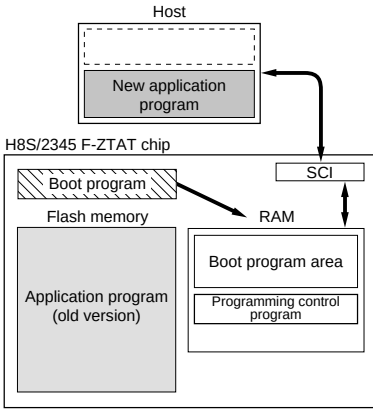
Notes: Only make a transition between user mode and user program mode when th
not accessing the flash memory.

1. NMI = 1, $MD_2 = MD_1 = MD_0 = 0$, $PF_2 = 1$, $PF_1 = PF_0 = 0$
2. NMI = 1, FWE = 1, $MD_2 = 0$, $MD_1 = 1$

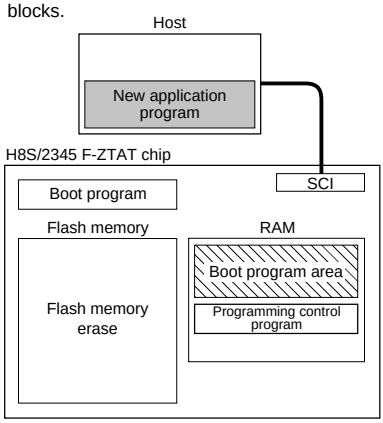
Figure 17.8 Flash Memory Mode Transitions

to the RAM boot program area.





3. Flash memory initialization
The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, entire flash memory erasure is performed, without regard to blocks.



4. Writing new application program
The programming control program transferre from the host to RAM is executed, and the n application program in the host is written into flash memory.

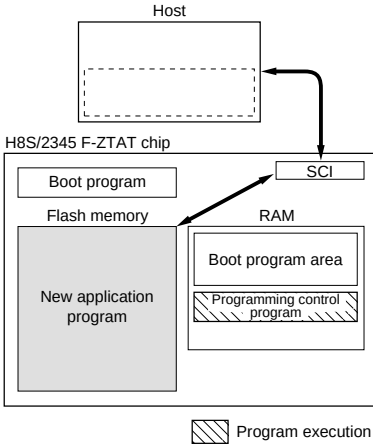
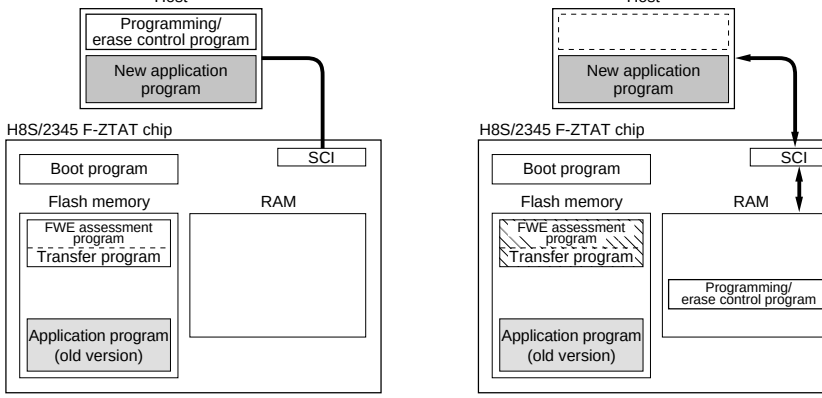


Figure 17.9 Boot Mode



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 l written into the erased flash memory bloc not write to unerased blocks.

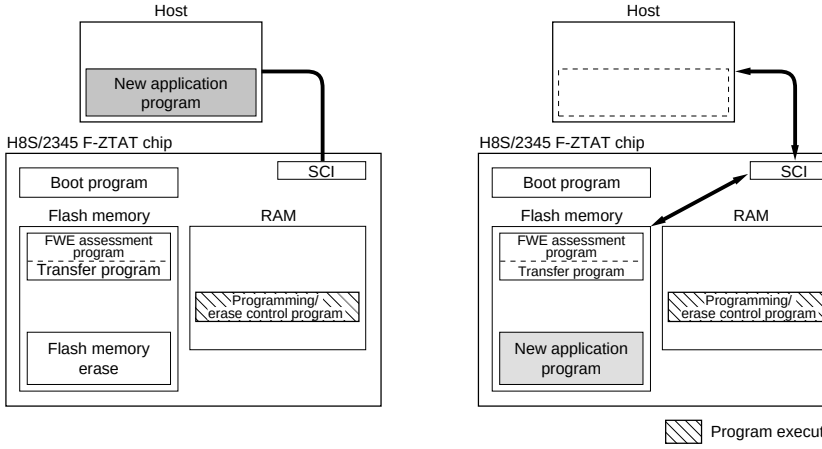


Figure 17.10 User Program Mode (Example)

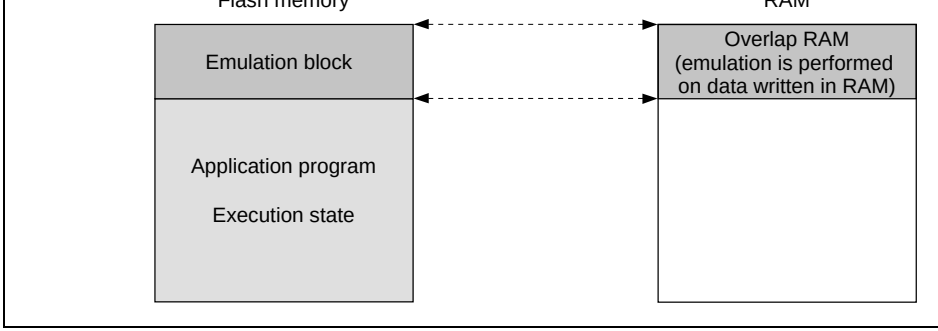


Figure 17.11 Reading Overlap Data in User Mode and User Program Mo

- Writing Overlap Data in User Program Mode

  When overlap RAM data is confirmed, the RAMS bit is cleared, RAM overlap is re
  writes should actually be performed to the flash memory.

  When the programming control program is transferred to RAM, ensure that the tran
  destination and the overlap RAM do not overlap, as this will cause data in the overl
  be rewritten.

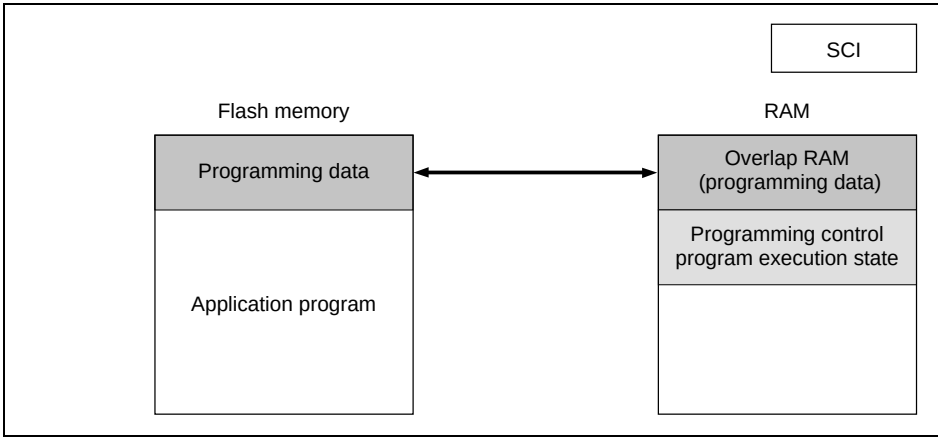


Figure 17.12 Writing Overlap Data in User Program Mode

Block Configuration: The flash memory is divided into four 1-kbyte blocks, one 28-k
one 16-kbyte block, two 8-kbyte blocks, and two 32-kbyte blocks.

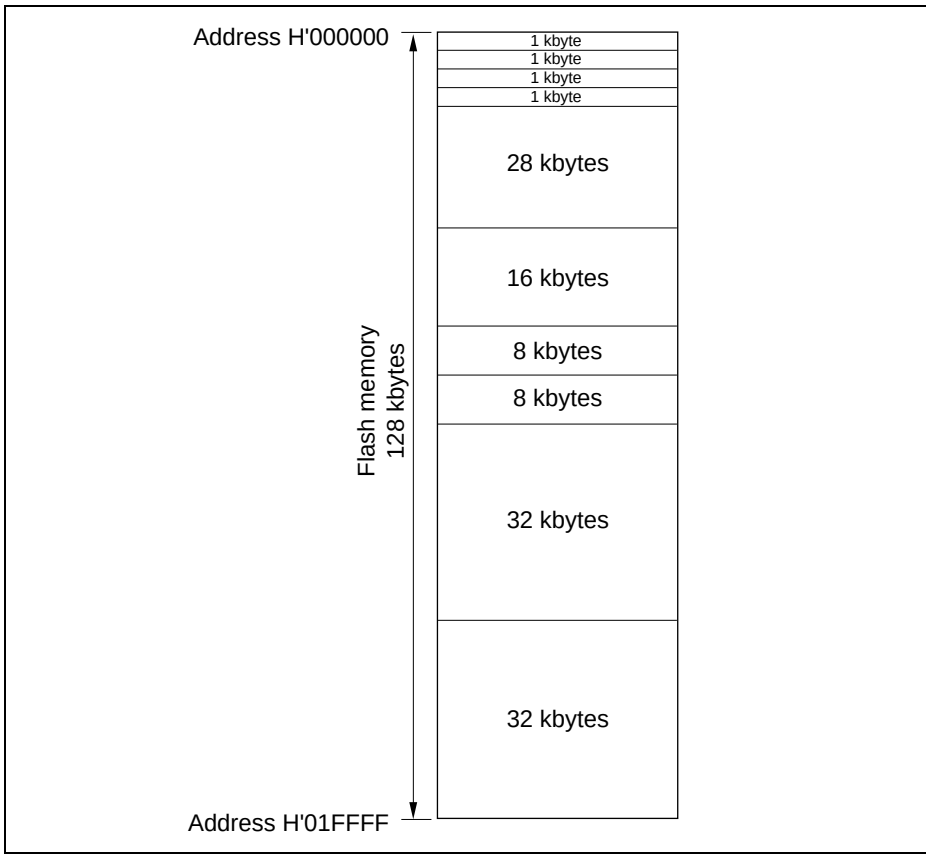


Figure 17.13 Flash Memory Block Configuration

Mode 2	MD_2	Input	Sets MCU operating mode
Mode 1	MD_1	Input	Sets MCU operating mode
Mode 0	MD_0	Input	Sets MCU operating mode
Port F_2	PF_2	Input	Sets MCU operating mode in write
Port F_1	PF_1	Input	Sets MCU operating mode in write
Port F_0	PF_0	Input	Sets MCU operating mode in write
Transmit data	TxD1	Output	Serial transmit data output
Receive data	RxD1	Input	Serial receive data input

Note: * FWE pin functions as $\overline{\text{WDTOVF}}$ pin on ZTAT, mask ROM, and ROMless ver

Flash memory control register 2	FLMCR2*6	R/W*3	H'00*5	H'F
Erase block register 1	EBR1*6	R/W*3	H'00*5	H'F
Erase block register 2	EBR2*6	R/W*3	H'00*5	H'F
System control register 2	SYSCR2	R/W	H'00	H'F
RAM emulation register	RAMER	R/W	H'00	H'F

Notes: The registers listed in table 7.11 are enabled on the F-ZTAT version only. They on the ZTAT, mask ROM, and ROMless versions, so an undefined value will b they are read, and it is not possible to write to these registers.

1. Lower 16 bits of the address.
2. Flash memory registers share addresses with other registers. Register sele performed by the FLSHE bit in the system control register 2 (SYSCR2).
3. In modes in which the on-chip flash memory is disabled (modes 4 and 5), a return H'00, and writes are invalid. Writes are also disabled when the FWE to 0 in FLMCR1.
4. When a high level is input to the FWE pin, the initial value is H'80.
5. When a low level is input to the FWE pin, or if a high level is input and the S FLMCR1 is not set, these registers are initialized to H'00.
6. FLMCR1, FLMCR2, EBR1, and EBR2 are 8-bit registers. Only byte access for these registers, the access requiring 2 states.

Read/Write R 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-v or erase-verify mode is entered by setting SWE to 1 when FWE = 1. Program mode is e setting SWE to 1 when FWE = 1, then setting the PSU bit in FLMCR2, and finally sett bit. Erase mode is entered by setting SWE to 1 when FWE = 1, then setting the ESU bi FLMCR2, and finally setting the E bit. FLMCR1 is initialized by a reset, and in hardwa mode and software standby mode. Its initial value is H'80 when a high level is input to pin, and H'00 when a low level is input. When on-chip flash memory is disabled (mode a read will return H'00, and writes are invalid.

Writes to the SWE bit in FLMCR1 are enabled only when FWE = 1; writes to the EV a only when FWE=1 and SWE=1; writes to the E bit only when FWE = 1, SWE = 1, and and writes to the P bit only when FWE = 1, SWE = 1, and PSU = 1.

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memor programming/erasing. See section 17.14, Flash Memory Programming and Erasing Pre before using this bit.

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

	[Setting condition] When FWE = 1

Bit 5 and 4—Reserved: Read-only bits, always read as 0.

Bit 3—Erase-Verify (EV): Selects erase-verify mode transition or clearing. Do not se
ESU, PSU, PV, E, or P bit at the same time.

Bit 3 EV	Description
0	Erase-verify mode cleared
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1

Bit 2—Program-Verify (PV): Selects program-verify mode transition or clearing. De
SWE, ESU, PSU, EV, E, or P bit at the same time.

Bit 2 PV	Description
0	Program-verify mode cleared
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1

	When FWE = 1, SWE = 1, and ESU = 1

Bit 0—Program (P): Selects program mode transition or clearing. Do not set the SWE PSU, EV, PV, or E bit at the same time.

Bit 0 P	Description
0	Program mode cleared (
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU = 1

17.7.2 Flash Memory Control Register 2 (FLMCR2)

Bit	7	6	5	4	3	2	1
	FLER	—	—	—	—	—	ESU
Initial value	0	0	0	0	0	0	0
Read/Write	R	—	—	—	—	—	R/W

FLMCR2 is an 8-bit register that monitors the presence or absence of flash memory pro protection (error protection) and performs setup for flash memory program/erase mode is initialized to H'00 by a reset, and in hardware standby mode. The ESU and PSU bits to 0 in software standby mode, hardware protect mode, and software protect mode.

When on-chip flash memory is disabled, a read will return H'00.

	[Clearing condition]
	Reset or hardware standby mode
1	An error has occurred during flash memory programming/erasing
	Flash memory program/erase protection (error protection) is enabled
	[Setting condition]
	See section 17.10.3, Error Protection

Bits 6 to 2—Reserved: Read-only bits, always read as 0.

Bit 1—Erase Setup (ESU): Prepares for a transition to erase mode. Set this bit to 1 b
the E bit to 1 in FLMCR1. Do not set the SWE, PSU, EV, PV, E, or P bit at the same

Bit 1 ESU	Description
0	Erase setup cleared
1	Erase setup
	[Setting condition]
	When FWE = 1, and SWE = 1

Bit 0—Program Setup (PSU): Prepares for a transition to program mode. Set this bit
setting the P bit to 1 in FLMCR1. Do not set the SWE, ESU, EV, PV, E, or P bit at th

Bit 0 PSU	Description
0	Program setup cleared
1	Program setup
	[Setting condition]
	When FWE = 1, and SWE = 1

EBR2	EB7	EB6	EB5	EB4	EB3	EB2	EB1
Initial value	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EBR1 and EBR2 are registers that specify the flash memory erase area block by block; 2 in EBR1 and bits 7 to 0 in EBR2 are readable/writable bits. EBR1 and EBR2 are each to H'00 by a reset, in hardware standby mode and software standby mode, when a low l input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit i is cleared to 0. When a bit in EBR1 or EBR2 is set, the corresponding block can be eras blocks are erase-protected. Blocks are erased separately (in one-block units), so set only EBR1 or EBR2 (more than one bit cannot be set to 1). To erase all blocks, erase one bl time, once after another in sequence. Then on-chip flash memory is disabled (modes 4 read with return H'00, and writes are disabled.

The flash memory block configuration is shown in table 17.12.

Table 17.12 Flash Memory Erase Blocks

Block (Size)	Address
EB0 (1 kbyte)	H'000000 to H'0003FF
EB1 (1 kbyte)	H'000400 to H'0007FF
EB2 (1 kbyte)	H'000800 to H'000BFF
EB3 (1 kbyte)	H'000C00 to H'000FFF
EB4 (28 kbytes)	H'001000 to H'007FFF
EB5 (16 kbytes)	H'008000 to H'00BFFF
EB6 (8 kbytes)	H'00C000 to H'00DFFF
EB7 (8 kbytes)	H'00E000 to H'00FFFF
EB8 (32 kbytes)	H'010000 to H'017FFF
EB9 (32 kbytes)	H'018000 to H'01FFFF

SYSCR2 is an 8-bit readable/writable register that controls on-chip flash memory (in F-ZTAT versions).

SYSCR2 is initialized to H'00 by a reset and in hardware standby mode.

SYSCR2 is available only in the F-ZTAT version. In the mask ROM and ZTAT versi register cannot be written to and will return an undefined value if read.

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

Bit 3—Flash Memory Control Register Enable (FLSHE): Controls CPU access to memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2). Setting the FLSH enables read/write access to the flash memory control registers. If FLSHE is cleared to memory control registers are deselected. In this case, the flash memory control registe are retained.

Bit 3 FLSHE	Description
0	Flash control registers deselected in area H'FFFFC8 to H'FFFFCB
1	Flash control registers selected in area H'FFFFC8 to H'FFFFCB

Bits 2 to 0—Reserved: Read-only bits, always read as 0.

RAMER specifies the area of flash memory to be overlapped with part of RAM when e
real-time flash memory programming. RAMER is initialized to H'00 by a reset and in h
standby mode. It is not initialized in software standby mode. RAMER settings should b
user mode or user program mode.

Flash memory area divisions are shown in table 17.13. To ensure correct operation of t
emulation function, the ROM for which RAM emulation is performed should not be ac
immediately after this register has been modified. Normal execution of an access imme
after register modification is not guaranteed.

Bits 7 to 3—Reserved: These bits are always read as 0.

Bit 2—RAM Select (RAMS): Specifies selection or non-selection of flash memory em
RAM. When RAMS = 1, all flash memory block are program/erase-protected.

Bit 2 RAMS	Description	
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	

Bits 1 and 0—Flash Memory Area Selection (RAM1, RAM0): These bits are used to
with bit 2 to select the flash memory area to be overlapped with RAM. (See table 17.13

H'000800–H'000BFF	EB2 (1 kbyte)	1	1	0
H'000C00–H'000FFF	EB3 (1 kbyte)	1	1	1

Notes: To use RAM for flash memory emulation, set the RAME bit of SYSCR to 1.
*: Don't care

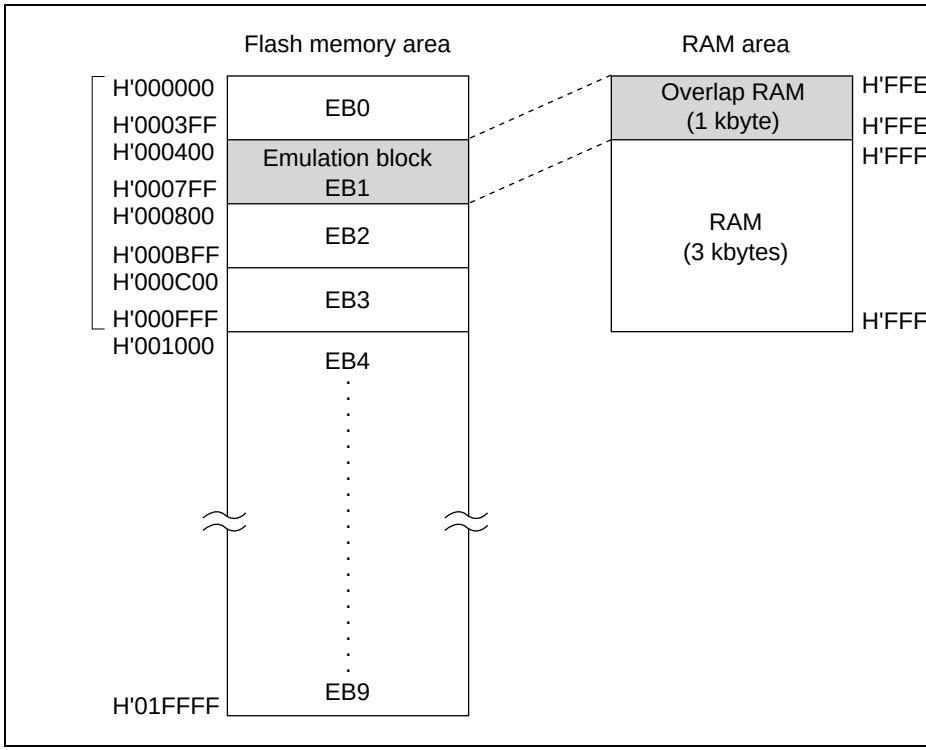


Figure 17.14 Example of Overlap Between Flash Memory Area and RAM (When RAMS = 1, RAM1 = 0, and RAM0 = 1)

Mode

Mode Name		CPU Operating Mode	FWE	MD_2	MD_1
Boot mode	Mode 10	Advanced expanded mode with on-chip ROM enabled	1	0	1
	Mode 11	Advanced single-chip mode			
User program mode[*1]	Mode 14	Advanced expanded mode with on-chip ROM enabled	1[*2]	1	1
	Mode 15	Advanced single-chip mode			

Notes: 1. Normally, user mode (modes 6 and 7) should be used. Set FWE to 1 to mak transition to user program mode (modes 14 and 15) before performing a program/erase/verify operation.

2. Refer to section 17.14, Flash Memory Programming and Erasing Precaution information on programming and clearing FWE.

RAM. After the transfer is completed, control branches to the start address of the prog
control program area and the programming control program execution state is entered
memory programming is performed).

The transferred programming control program must therefore include coding that follo
programming algorithm given later.

The system configuration in boot mode is shown in figure 17.15, and the boot progran
execution procedure in figure 17.16.

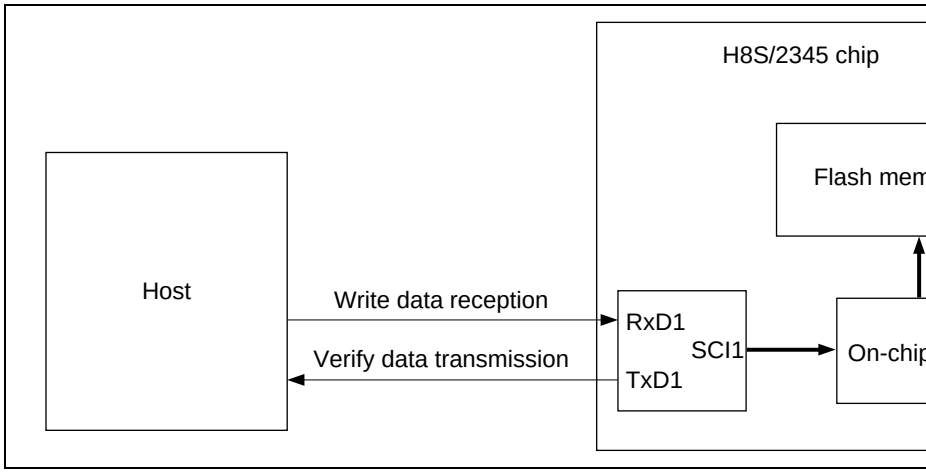
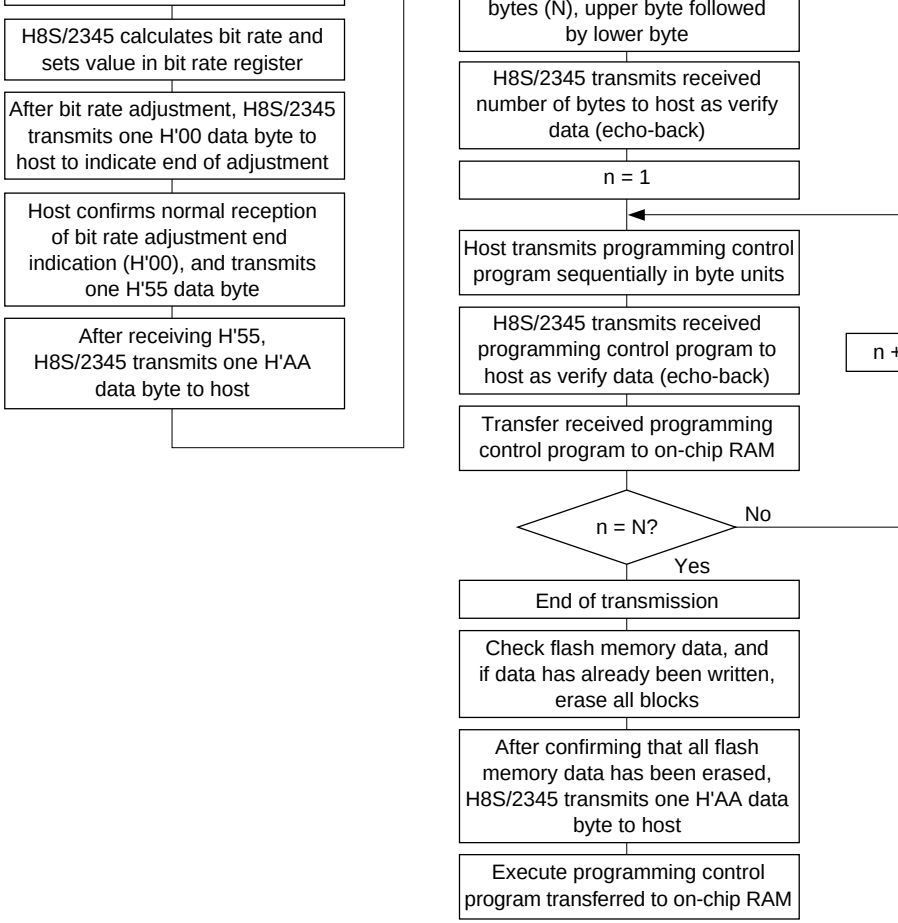


Figure 17.15 System Configuration in Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byt
transmitted as an erase error, and the erase operation and subsequent operatio
are halted.

Figure 17.16 Boot Mode Execution Procedure

Figure 17.17 Measurement of Low Period of Host Transmission Data

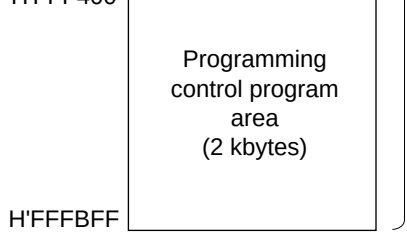
When boot mode is initiated, the H8S/2345 MCU measures the low period of the asyn
SCI communication data (H'00) transmitted continuously from the host, see figure 17.
transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The M
calculates the bit rate of the transmission from the host from the measured low period,
transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host
confirm that this adjustment end indication (H'00) has been received normally, and tra
H'55 byte to the MCU. If reception cannot be performed normally, initiate boot mode
(reset), and repeat the above operations. Depending on the host's transmission bit rate
MCU's system clock frequency, there will be a discrepancy between the bit rates of th
the MCU. To ensure correct SCI operation, the host's transfer bit rate should be set to
9600) bps.

Table 17.15 shows typical host transfer bit rates and system clock frequencies for whi
adjustment of the MCU's bit rate is possible. The boot program should be executed w
system clock range.

**Table 17.15 System Clock Frequencies for which Automatic Adjustment of H8S/
Rate is Possible**

Host Bit Rate	System Clock Frequency for which Automatic Adjustr of H8S/2345 Bit Rate is Possible
9600 bps	8 MHz to 20 MHz
4800 bps	4 MHz to 20 MHz

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the 2 kbytes area from
to H'FFF3FF is reserved for use by the boot program, as shown in figure 17.18. The ar
the programming control program is transferred is H'FFF400 to H'FFFBFF. The boot
area can be used when the programming control program transferred into RAM enters
execution state. A stack area should be set up as required.



Note: * The boot program area cannot be used until a transition is made to the execu
for the programming control program transferred to RAM. Note that the boot p
remains stored in this area after a branch is made to the programming contro

Figure 17.18 RAM Areas in Boot Mode

Notes on Use of User Mode:

- When the chip comes out of reset in boot mode, it measures the low-level period of the SCI's RxD1 pin. The reset should end with RxD1 high. After the reset ends, it t approximately 100 states before the chip is ready to measure the low-level period o pin.
- In boot mode, if any data has been programmed into the flash memory (if all data i flash memory blocks are erased. Boot mode is for use when user program mode is u such as the first time on-board programming is performed, or if the program activat program mode is accidentally erased.
- Interrupts cannot be used while the flash memory is being programmed or erased.
- The RxD1 and TxD1 pins should be pulled up on the board.
- Before branching to the programming control program (RAM area H'FFF400), the terminates transmit and receive operations by the on-chip SCI (channel 1) (by clear and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. The tra output pin, TxD1, goes to the high-level output state (P31DDR = 1, P31DR = 1).

To change from boot mode to another mode (user mode, etc.), the microcomputer's boot mode status must first be cleared by inputting a reset using the $\overline{RES}$ pin*[1]. In $\overline{RES}$ pin must be kept low (t_{RESW}) for at least 20 states. (See figure 17.38.)

- Do not change the FWE pin and mode pin input levels in boot mode, and do not dr pin low while the boot program is being executed or while flash memory is being p or erased.*[2]

- If the FWE pin or mode pin input levels are changed (for example, from low to hig reset, the state of ports with multiplexed address functions and bus control output p $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$) will change according to the change in the microcomputer's ope mode*[3].

  Therefore, care must be taken to make pin settings to prevent these pins from beco signal pins during a reset, or to prevent collision with signals outside the microcon

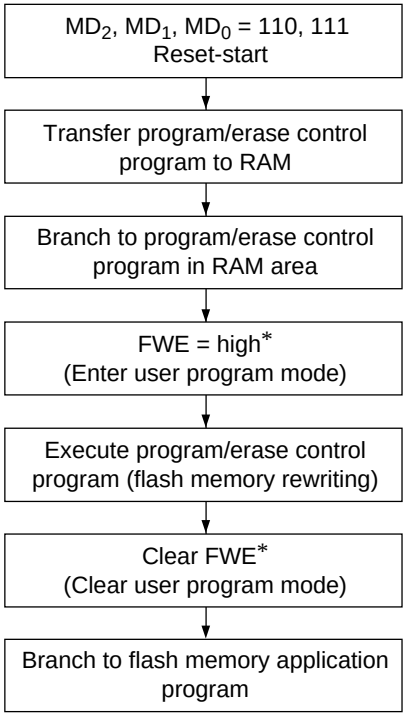
Notes: 1. FWE pin and mode pin input must satisfy the mode programming setup tim 200 ns) with respect to the reset release timing, as shown in figures 17.36 t
2. For further information on FWE application and disconnection, see section Flash Memory Programming and Erasing Precautions.
3. See appendix D, Pin States.

17.8.2 User Program Mode

When set to user program mode, the chip can program and erase its flash memory by e user program/erase control program. Therefore, on-board reprogramming of the on-ch memory can be carried out by providing on-board means of FWE control and supply o programming data, and storing a program/erase control program in part of the program necessary.

To select user program mode, select a mode that enables the on-chip flash memory (m and apply a high level to the FWE pin. In this mode, on-chip supporting modules othe memory operate as they normally would in modes 6 and 7, see figures 17.37 and 17.3

and transfer program (and the program/erase control program if necessary) to flash memory beforehand



Notes: Do not apply a constant high level to the FWE pin. Apply a high level to the FW only when the flash memory is programmed or erased. Also, while a high level 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 Flash Memory Programming and Erasing Precautions.

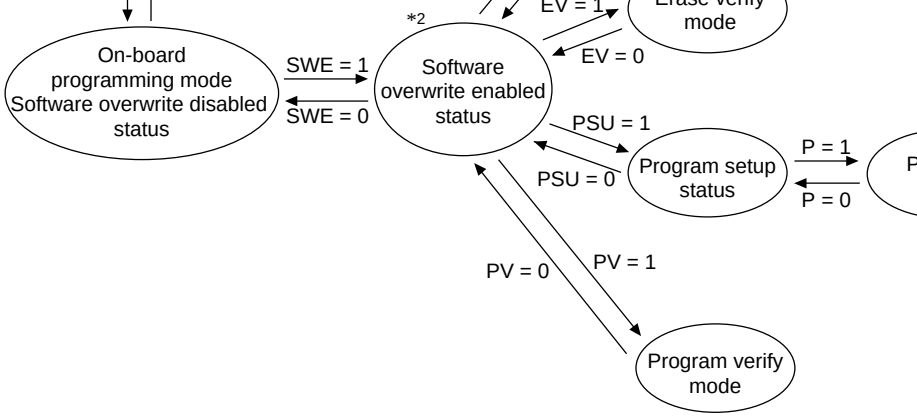
Figure 17.19 User Program Mode Execution Procedure

controls flash memory programming/erasing (the programming control program) shou
located and executed in on-chip RAM or external memory.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, EV, PV, E, and
FLMCR1, and the ESU and PSU bits in FLMCR2, is executed by a progra
memory.

2. When programming or erasing, set FWE to 1 (programming/erasing will n
executed if FWE = 0).

3. Perform programming in the erased state. Do not perform additional progr
previously programmed addresses.



Notes: 1. : user mode, : on-board programming mode
2. Do not make transitions by setting or clearing multiple bits simultaneously.
3. After changing from erase mode to erase setup status, do not change back to erase mo via software overwrite enable status.
4. After changing from program mode to program setup status, do not change back to pro except via software overwrite enable status.

Figure 17.20 Mode Transitions Using Settings of Bits in FLMCR1 and FLM

17.9.1 Program Mode

Follow the procedure shown in the program/program-verify flowchart in figure 17.21 t or programs to flash memory. Performing program operations according to this flowcha enable data or programs to be written to flash memory without subjecting the device to stress or sacrificing program data reliability. Programming should be carried out 32 byt time.

The wait times (x, y, z, α, ß, γ, ε, η) after bits are set or cleared in flash memory contro and 2 (FLMCR1, FLMCR2) and the maximum number of programming operations (N in table 20.10.

Next, the watchdog timer is set to prevent overprogramming in the event of program
Set a value greater than (y + z + α + ß) µs as the WDT overflow period. After this, pre
program mode (program setup) is carried out by setting the PSU bit in FLMCR2, and
elapse of (y) µs or more, the operating mode is switched to program mode by setting t
FLMCR1. The time during which the P bit is set is the flash memory programming tir
program setting so that the time for one programming operation is within the range of

17.9.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it
correctly written in the flash memory.

After the elapse of a given programming time, the programming mode is exited (the P
FLMCR1 is cleared to 0, then the PSU bit in FLMCR2 is cleared to 0 at least (α) µs la
the watchdog timer is cleared after the elapse of (y + z + α + β) µs or more, and the op
mode is switched to program-verify mode by setting the PV bit in FLMCR1. Before re
program-verify mode, a dummy write of H'FF data should be made to the addresses to
The dummy write should be executed after the elapse of (γ) µs or more. When the flas
read in this state (verify data is read in 16-bit units), the data at the latched address is r
least (ε) µs after the dummy write before performing this read operation. Next, the ori
written data is compared with the verify data, and reprogram data is computed (see fig
and transferred to the reprogram data area. After 32 bytes of data have been verified, e
verify mode, wait for at least (η) µs, then clear the SWE bit in FLMCR1 to 0. If repro
necessary, set program mode again, and repeat the program/program-verify sequence
However, ensure that the program/program-verify sequence is not repeated more than
the same bits.

Note: An area in RAM for storing write data (32 bytes) and an area for storing rewri
bytes) are required.

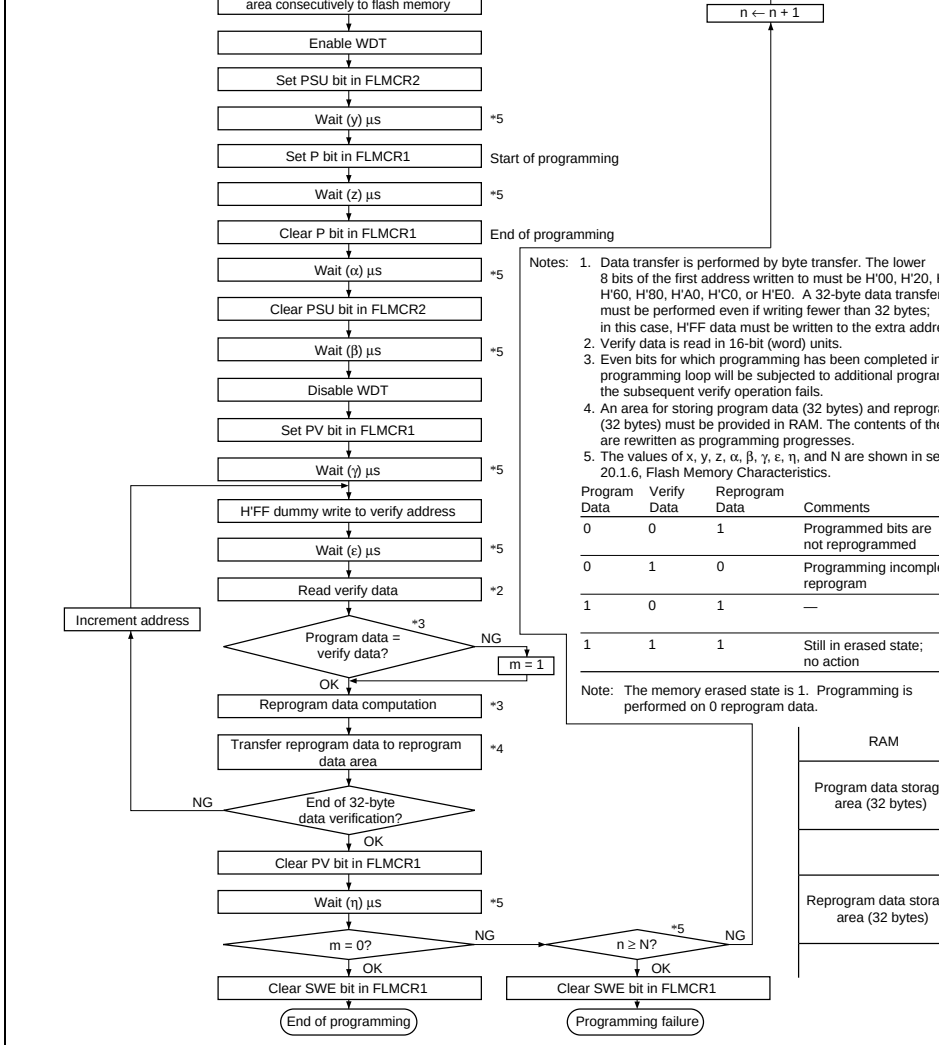


Figure 17.21 Program/Program-Verify Flowchart

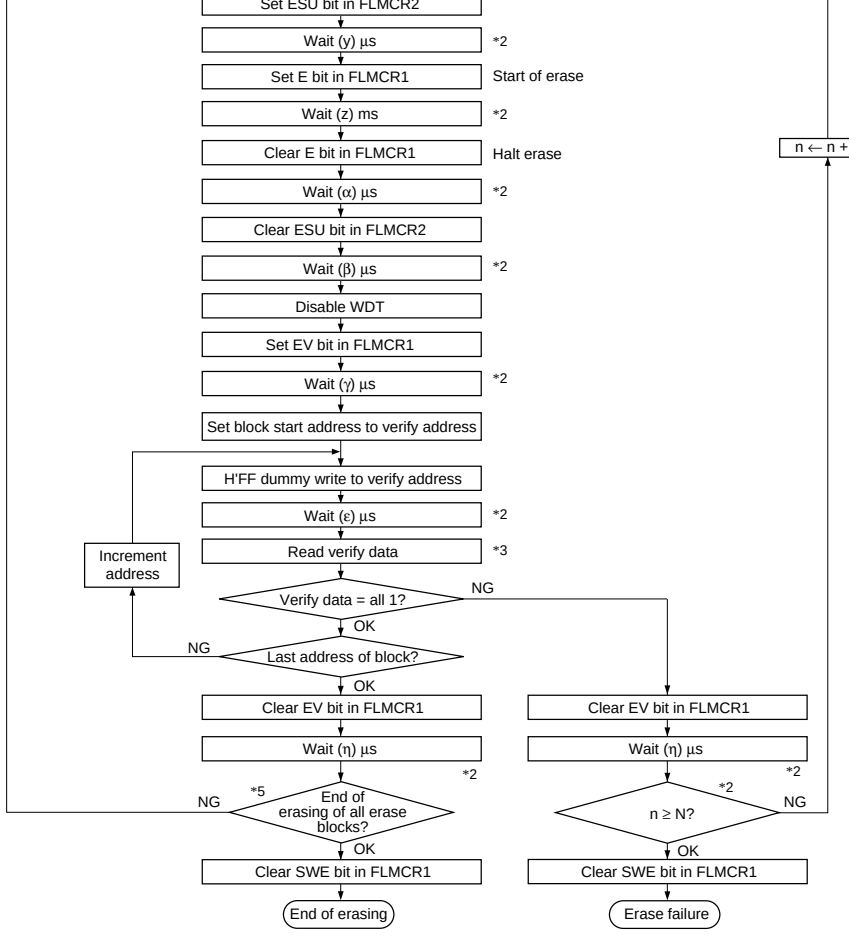
To perform data or program erasure, make a 1 bit setting for the flash memory area to
EBR1 or EBR2 at least (x) µs after setting the SWE bit to 1 in FLMCR1. Next, the wa
is set to prevent overerasing in the event of program runaway, etc. Set a value greater
α + ß) µs as the WDT overflow period. After this, preparation for erase mode (erase s
carried out by setting the ESU bit in FLMCR2, and after the elapse of (y) µs or more,
mode is switched to erase mode by setting the E bit in FLMCR1. The time during whi
is set is the flash memory erase time. Ensure that the erase time does not exceed (z) m

Note: With flash memory erasing, preprogramming (setting all data in the memory t
to 0) is not necessary before starting the erase procedure.

17.9.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it ha
correctly erased.

After the elapse of the erase time, erase mode is exited (the E bit in FLMCR1 is cleare
the ESU bit in FLMCR2 is cleared to 0 at least (α) µs later), the watchdog timer is cle
elapse of (β) µs or more, and the operating mode is switched to erase-verify mode by
EV bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data
made to the addresses to be read. The dummy write should be executed after the elaps
more. When the flash memory is read in this state (verify data is read in 16-bit units),
the latched address is read. Wait at least (ε) µs after the dummy write before performi
operation. If the read data has been erased (all 1), a dummy write is performed to the n
and erase-verify is performed. If the read data has not been erased, set erase mode aga
repeat the erase/erase-verify sequence in the same way. However, ensure that the erase
verify sequence is not repeated more than (N) times. When verification is completed,
verify mode, and wait for at least (η) µs. If erasure has been completed on all the erase
clear the SWE bit in FLMCR1 to 0. If there are any unerased blocks, make a 1 bit set
or EBR2 for the flash memory area to be erased, and repeat the erase/erase-verify sequ
same way.



Notes: 1. Preprogramming (setting erase block data to all 0) is not necessary.
2. The values of x, y, z, α, β, γ, ε, η, and N are shown in section 20.1.6, Flash Memory Characteristics.
3. Verify data is read in 16-bit (word) units.
4. Set only one bit in EBR1or EBR2. More than one bit cannot be set.
5. Erasing is performed in block units. To erase a number of blocks, the individual blocks must be erased sequentially.

Figure 17.22 Erase/Erase-Verify Flowchart (Single-Block Erase)

disabled or aborted. Hardware protection is reset by settings in flash memory control r
and 2 (FLMCR1, FLMCR2) and erase block registers 1 and 2 (EBR1, EBR2). (See tab

Table 17.16 Hardware Protection

		Functio	
Item	**Description**	**Program**	**Erase**
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1, FLMCR2 (excluding the FLER bit), EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	No	No
Reset/standby protection	• In a reset (including a WDT overflow 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 (t_{RESW}) specified in the AC Characteristics section.	No	No

Note: * Program verify and erase verify modes.

Item	Description	Functions	
		Program	Erase
SWE bit protection	• Clearing the SWE bit to 0 in FLMCR1 sets the program/erase-protected state for all blocks. (Execute in on-chip RAM or external memory.)	No	No
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block registers 1 and 2 (EBR1, EBR2). However, write protection is disabled. • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	No
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	No	No

Note: * Program verify and erase verify modes.

FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, a
settings are retained, but program mode or erase mode is aborted at the point at which
occurred. When the FLER bit is set to 1, it is not possible to re-enter the program mod
mode by resetting the P and E bits of FLMCR1. However, setting of the PV and EV b
FLMCR1 is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

- When flash memory is read during programming/erasing (including a vector read
  fetch)
- Immediately after exception handling (excluding a reset) during programming/eras
- When a SLEEP instruction (including software standby) is executed during
  programming/erasing
- When the CPU loses the bus during programming/erasing

Error protection is released only by a reset and in hardware standby mode.

Figure 17.23 shows the flash memory state transition diagram.

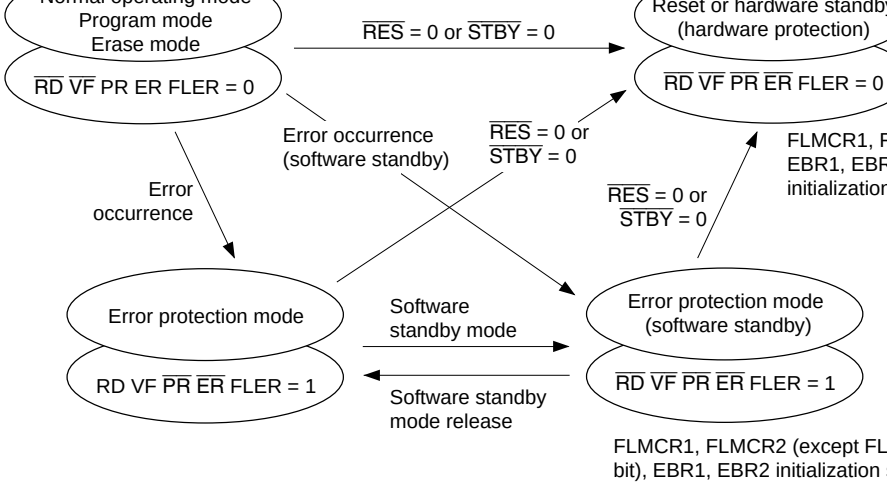


Figure 17.23 Flash Memory State Transitions

The error protect function has no effect on illegal operations unrelated to the setting co the FLER bit. Also, if a significant amount of time has elapsed before the transition to t status, there is a possibility that the data in flash memory may already have become cor Consequently, this function is not able to provide complete protection against corruptio data in flash memory.

For this reason, it is necessary to run program and erase algorithms correctly while flas enable (FWE) is being applied and to monitor the internal operation of the microcompu abnormalities using a watchdog timer, or the like, in order to prevent illegal operations mentioned above. Also, at the point when the transition is made to the protect mode, in the flash memory may be in an erroneously programmed or erased status, or the progra erasing may be incomplete due to a forced shutdown. In such a case, make sure to force

by overwriting parameters and other data in real time. In such cases, making a setting
emulation register (RAMER) enables part of RAM to be overlapped onto the flash me
so that data to be written to flash memory can be emulated in RAM in real time. After
RAMER setting has been made, accesses can be made from the flash memory area or
area overlapping flash memory. Emulation can be performed in user mode and user pr
mode. Figure 17.24 shows an example of emulation of real-time flash memory progra

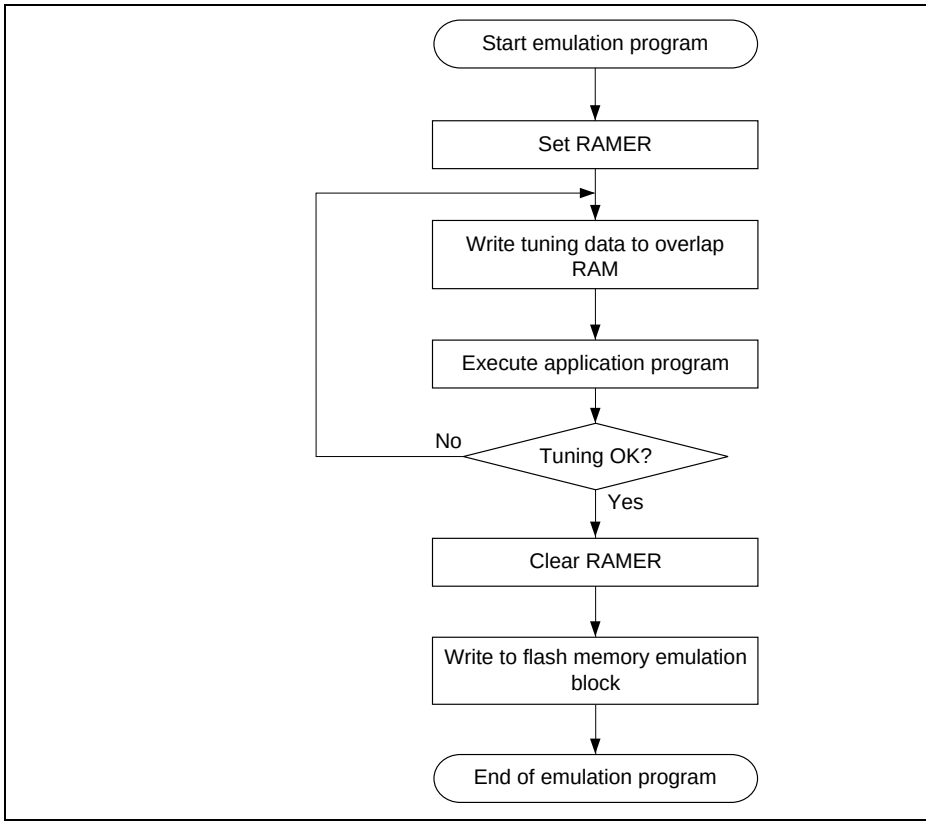


Figure 17.24 Flowchart for Flash Memory Emulation in RAM

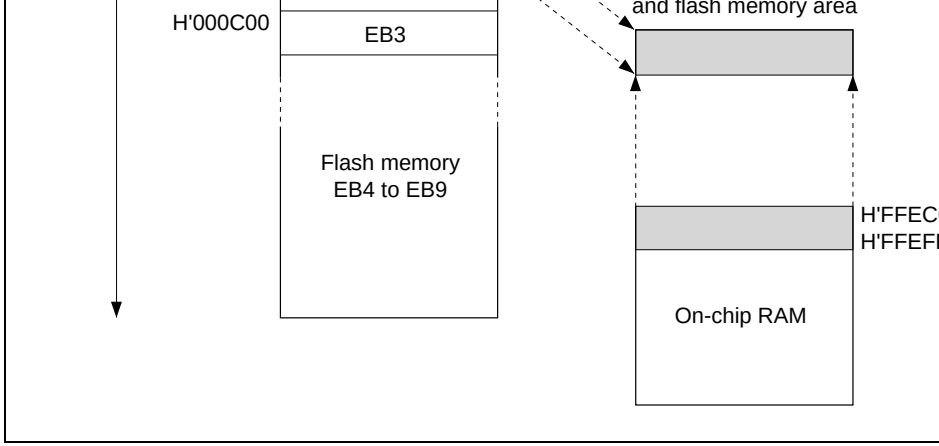


Figure 17.25 Example of RAM Overlap Operation

Example in Which Flash Memory Block Area (EB1) is Overlapped

1. Set bits RAMS, RAM1, and RAM0 in RAMER to 1, 0, 1, to overlap part of RAM o area (EB1) 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
4. The data written in the overlapping RAM is written into the flash memory space (E

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all b regardless of the value of RAM1 and RAM0 (emulation protection). In this setting the P or E bit in flash memory control register 1 (FLMCR1) will not transition to program mode or erase mode. When actually programming a fl memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 includes the vector table. When performing RAM emulation vector table is needed by the overlap RAM.

2. In the interrupt exception handling sequence during programming or erasing, the v not be read correctly*[2], possibly resulting in MCU runaway.
3. If interrupt occurred during boot program execution, it would not be possible to ex normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disab interrupt, as an exception to the general rule. However, this provision does not guaran erasing and programming or MCU operation. All requests, including NMI interrupt, m therefore be restricted inside and outside the MCU when programming or erasing flash NMI interrupt is also disabled in the error-protection state while the P or E bit remains FLMCR1.

Notes: 1. Interrupt requests must be disabled inside and outside the MCU until the p control program has completed programming.
2. The vector may not be read correctly in this case for the following two reas
   - If flash memory is read while being programmed or erased (while the P set in FLMCR1), correct read data will not be obtained (undetermined v be returned).
   - If the interrupt entry in the vector table has not been programmed yet, i exception handling will not be executed correctly.

read mode are supported with this device type. In auto-program mode, auto-erase mod
status read mode, a status polling procedure is used, and in status read mode, detailed i
signals are output after execution of an auto-program or auto-erase operation.

Table 17.18 shows writer mode pin settings.

Table 17.18 Writer Mode Pin Settings

Pin Names	Settings/External Circuit Connection
Mode pins: MD_2, MD_1, MD_0	Low-level input
Mode setting pins: PF_2, PF_1, PF_0	High-level input to PF_2, low-level input to PF_1
FWE pin	High-level input (in auto-program and auto-e modes)
$\overline{STBY}$ pin	High-level input (do not select hardware stan
$\overline{RES}$ pin	Power-on reset circuit
NMI pin	High-level input (for power-on reset)
XTAL, EXTAL pins	Oscillator circuit
Other pins requiring setting: $P2_3$, $P2_5$	High-level input to $P2_3$ and $P2_5$

in writer mode. For pin names in writer mode, see section 1.3.2, Pin Functions in Each Mode.

Table 17.19 Socket Adapter Name

		Socket Adapter Name	
Product Model	**Package Name**	**Minato Electronics**	**Data I/O**
HD64F2345	100-pin TQFP (TFP-100B)	ME2345ESNF1H	HF234BT
	100-pin TQFP (TFP-100G)	ME2345ESMF1H	HF234GT
	100-pin QFP (FP-100A)	ME2345ESFF1H	HF234AQ
	100-pin QFP (FP-100B)	ME2345ESHF1H	HF234BQ

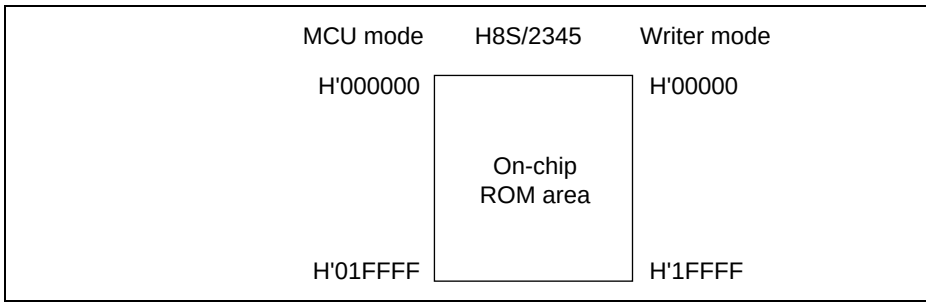


Figure 17.26 Memory Map in Writer Mode

Auto-program mode supports programming of 128 bytes at a time. Status polling is confirm the end of auto-programming.

- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status poll to confirm the end of auto-erasing.

- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal terminati confirmed by reading the FO_6 signal. In status read mode, error information is outpu error occurs.

Table 17.20 Settings for Each Operating Mode in Writer Mode

	Pin Names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**FO_0 to FO_7**	**FA**
Read	H or L	L	L	H	Data output	Ai
Output disable	H or L	L	H	H	Hi-Z	X
Command write	H or L*3	L	H	L	Data input	Ai
Chip disable*1	H or L	H	X	X	Hi-Z	X

Legend:

H: High level

L: Low level

X: Don't care

Hi-Z: High impedance

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 when making a transition to auto-program or auto-eras input a high level to the FWE pin.

Legend:

RA: Read address

WA: Program address (Write address)

Dout: Read data

Din: Program data

Notes: 1. In memory read mode, the number of cycles depends on the number of ad
cycles (n).

2. In auto-program mode. 129 cycles are required for command writing by a s
128-byte write.

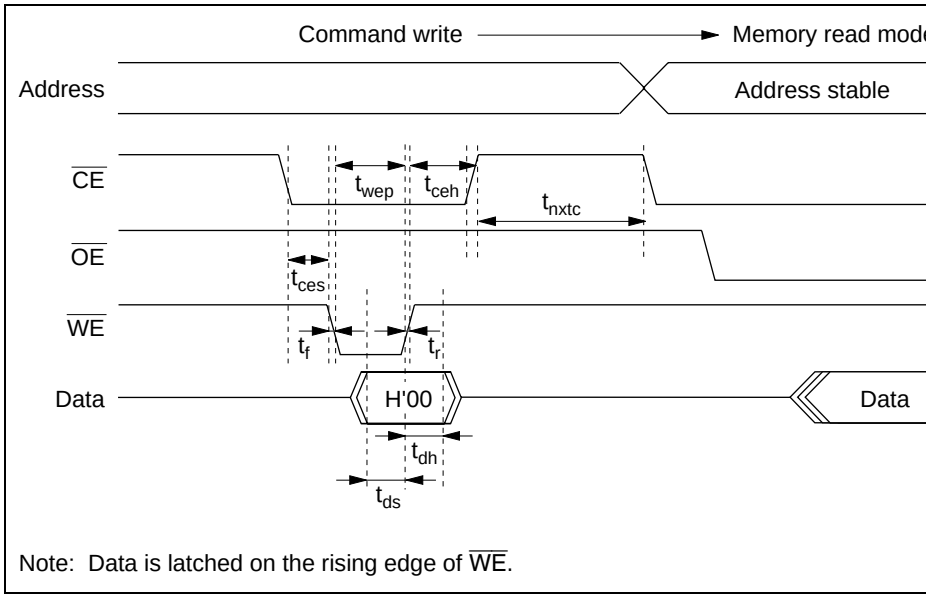
voltage							
Schmitt trigger input voltage	$\overline{OE}$, $\overline{CE}$, $\overline{WE}$	V_{T-}	1.0	—	2.5	V	
		V_{T+}	2.0	—	3.5	V	
		V_{T+}–V_{T-}	0.4	—	—	V	
Output high-level voltage	FO_7–FO_0	V_{OH}	V_{CC}– 0.5	—	—	V	I_O
Output low-level voltage	FO_7–FO_0	V_{OL}	—	—	0.4	V	I_O
Input leak current	FO_7–FO_0, FA_{16}–FA_0	$\lvert I_{LI} \rvert$	—	—	2	µA	
V_{CC} current	During read	I_{CC}	—	60	89	mA	
	During programming	I_{CC}	—	70	89	mA	
	During erasing	I_{CC}	—	70	89	mA	

Note: Refer to the maximum rating for the F-ZTAT version “20.1.1 Absolute Maximum
If the maximum rating is exceeded, the LSI may be damaged permanently.

17.13.4 Memory Read Mode

- After the end of an auto-program, auto-erase, or status read operation, the command is entered. To read memory contents, a transition must be made to memory read mo means of a command write before the read is executed.
- Command writes can be performed in memory read mode, just as in the command w
- Once memory read mode has been entered, consecutive reads can be performed.
- After power-on, memory read mode is entered.

$\overline{CE}$ 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: Data is latched on the rising edge of $\overline{WE}$.

Figure 17.27 Memory Read Mode Transition Timing Waveforms

Output disable delay time	t_{df}			100	ns
Data output hold time	t_{oh}	5			ns

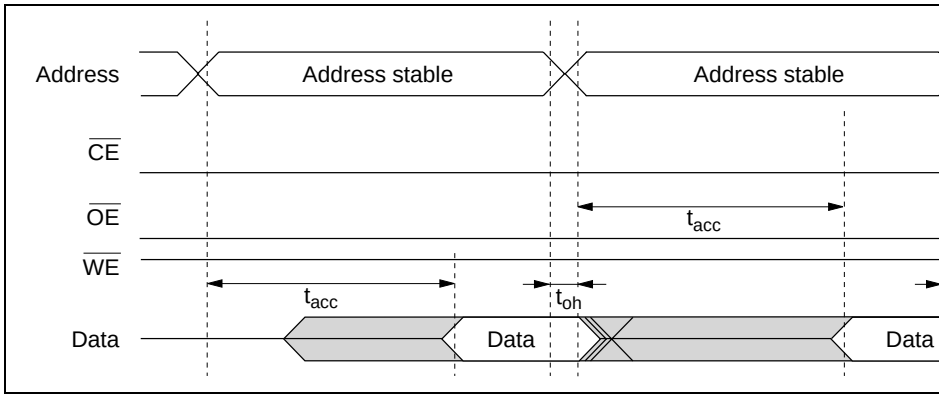


Figure 17.28 Timing Waveforms for $\overline{CE}$/$\overline{OE}$ Enable State Read

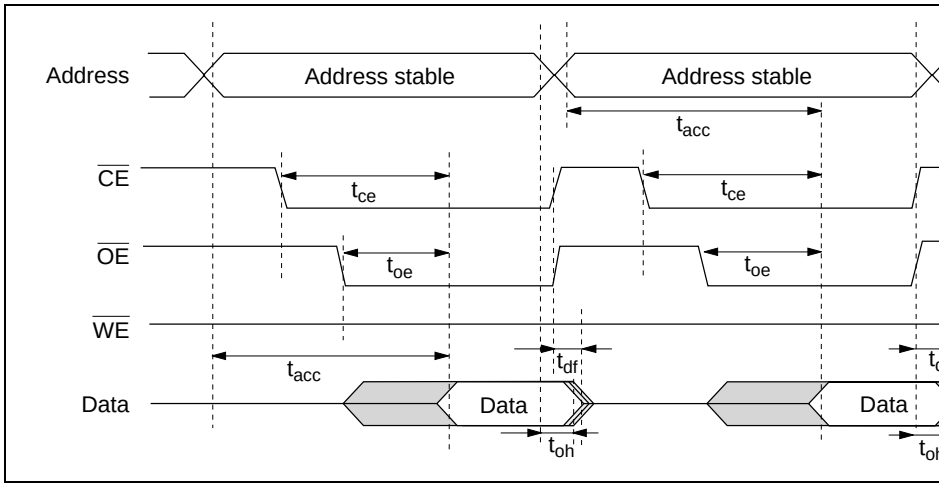


Figure 17.29 Timing Waveforms for $\overline{CE}$/$\overline{OE}$ Clocked Read

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

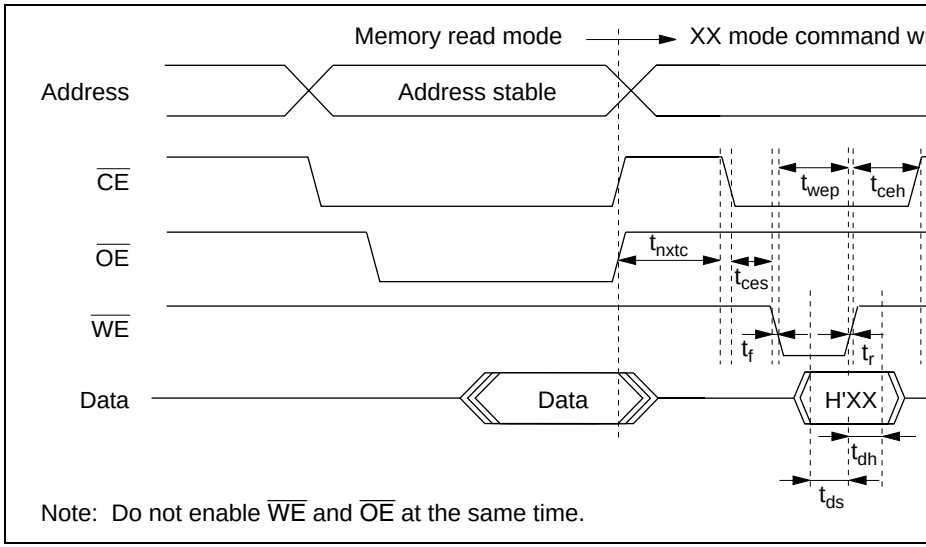


Figure 17.30 Timing Waveforms when Entering Another Mode from Memory

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
$\overline{WE}$ rise time	t_{r}		30	ns
$\overline{WE}$ fall time	t_{f}		30	ns
Write setup time	t_{pns}	100		ns
Write end setup time	t_{pnh}	100		ns

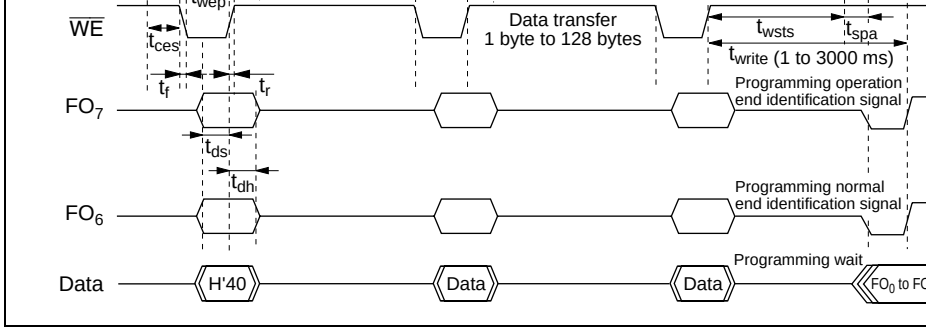


Figure 17.31 Auto-Program Mode Timing Waveforms

Notes on Use of Auto-Program Mode

- In auto-program mode, 128 bytes are programmed simultaneously. This should be by executing 128 consecutive byte transfers.
- A 128-byte data transfer is necessary even when programming fewer than 128 byt case, H'FF data must be written to the extra addresses.
- The lower 8 bits of the transfer address must be H'00 or H'80. If a value other than address is input, processing will switch to a memory write operation but a write er flagged.
- Memory address transfer is performed in the second cycle (figure 17.31). Do not p memory address transfer after the second cycle.
- Do not perform a command write during a programming operation.
- Perform one auto-programming operation for a 128-byte block for each address. Characteristics are not guaranteed for two or more programming operations.
- Confirm normal end of auto-programming by checking FO_6. Alternatively, status r can also be used for this purpose (FO_7 status polling uses the auto-program operati identification pin).
- The status polling FO_6 and FO_7 pin information is retained until the next command the next command write is performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

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	
$\overline{WE}$ rise time	t_{r}		30	ns	
$\overline{WE}$ fall time	t_{f}		30	ns	
Erase setup time	t_{ens}	100		ns	
Erase end setup time	t_{enh}	100		ns	

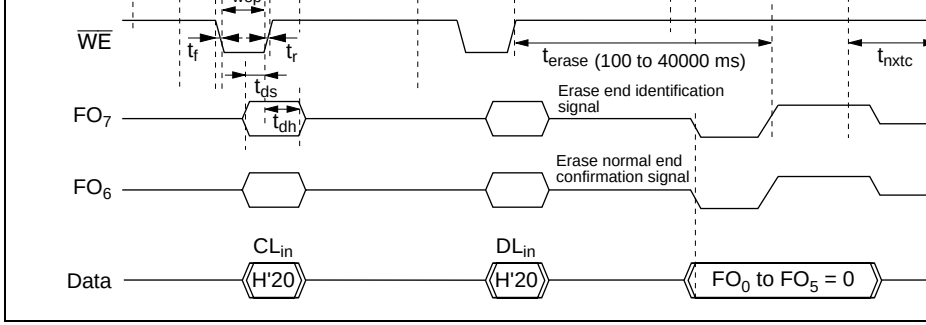


Figure 17.32 Auto-Erase Mode Timing Waveforms

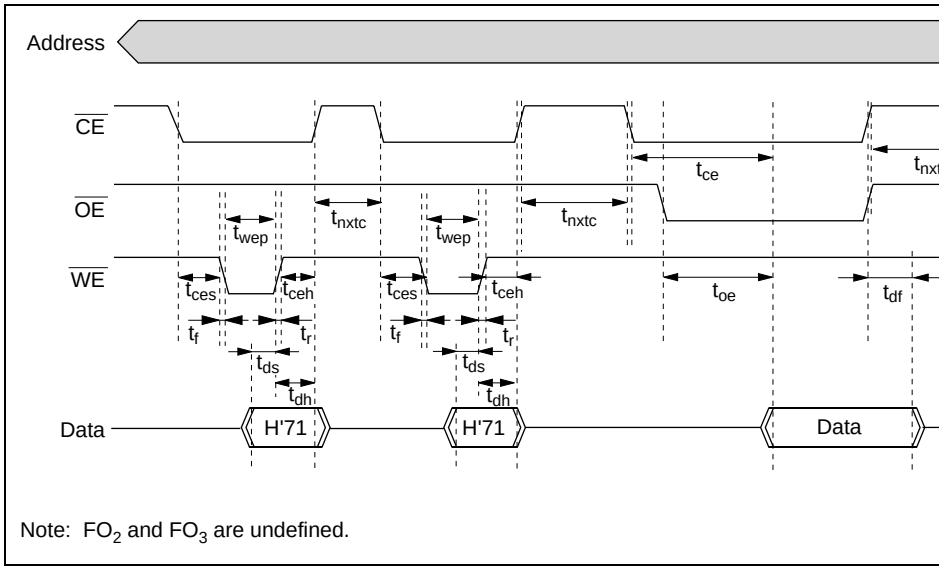
Notes on Use of Erase-Program Mode

- Auto-erase mode supports only entire memory erasing.
- Do not perform a command write during auto-erasing.
- Confirm normal end of auto-erasing by checking FO_6. Alternatively, status read m be used for this purpose (FO_7 status polling uses the auto-erase operation end ident pin).
- The status polling FO_6 and FO_7 pin information is retained until the next command the next command write is performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

17.13.7 Status Read Mode

- Status read mode is used to identify what type of abnormal end has occurred. Use when an abnormal end occurs in auto-program mode or auto-erase mode.
- The return code is retained until a command write for other than status read mode performed.

$\overline{CE}$ 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: FO_2 and FO_3 are undefined.

Figure 17.33 Status Read Mode Timing Waveforms

	end: 0 Abnormal end: 1	error: 1 Otherwise: 0	ming error: 1 Otherwise: 0	error: 1 Otherwise: 0	exceeded: 1 Otherwise: 0

Note: * FO_2 and FO_3 are undefined.

Status Read Mode Usage Note: After the auto-program mode or auto-erase mode has make sure to enter the status read mode before powering off the system.

The return commands are undefined immediately after power-on or if the system has b powered off once.

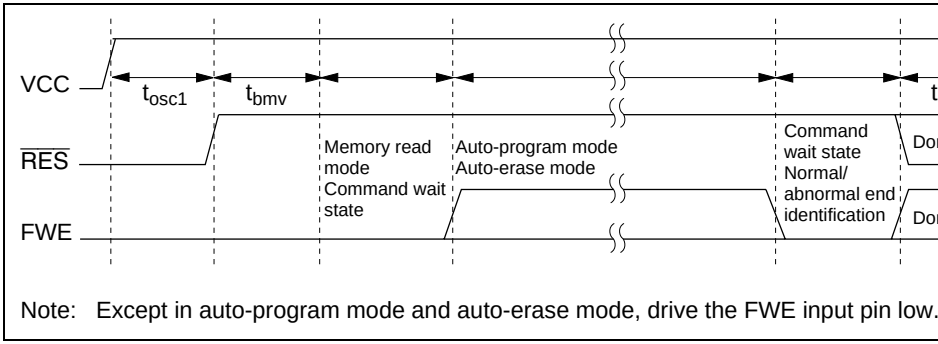
17.13.8 Status Polling

- The FO_7 status polling flag indicates the operating status in auto-program or auto-e
- The FO_6 status polling flag indicates a normal or abnormal end in auto-program or mode.

Table 17.30 Status Polling Output Truth Table

Pin Names	Internal Operation in Progress	Abnormal End	—	Norma
FO_7	0	1	0	1
FO_6	0	0	1	1
FO_0 to FO_5	0	0	0	0

stabilization time)				
Writer mode setup time	t_{bmv}	10	—	ms
V_{CC} hold time	t_{dwn}	0	—	ms



Note: Except in auto-program mode and auto-erase mode, drive the FWE input pin low.

Figure 17.34 Oscillation Stabilization Time, Writer Mode Setup Time, and Pow Fall Sequence

Notes: 1. The flash memory is initially in the erased state when the device is shipped
Technology. For other chips for which the erasure history is unknown, it is
recommended that auto-erasing be executed to check and supplement the i
(erase) level.

2. Auto-programming in the writer mode should be performed only once for
byte write unit block.

It is not possible to write additional data to a 128-byte write unit block tha
been programmed. To reprogram a block, first use the auto-erase mode
the auto-program mode.

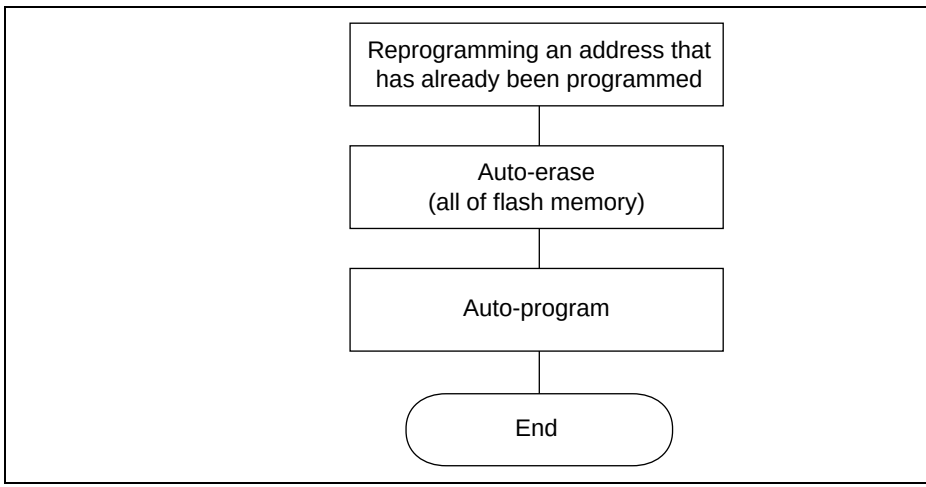


Figure 17.35 Reprogramming an Address that has Already Been Progra

Do not select the HN28F101 setting for the PROM programmer, and only use the speci
adapter. Incorrect use will result in damaging the device.

Powering on and off (see figures 17.36 to 17.38): Do not apply a high level to the FW
V_{CC} has stabilized. Also, drive the FWE pin low before turning off V_{CC}.

When applying or disconnecting V_{CC}, fix the FWE pin low and place the flash memory
hardware protection state.

The power-on and power-off timing requirements should also be satisfied in the event o
failure and subsequent recovery. If these timing requirements are not satisfied, the micr
experience program runaway, possibly resulting in excessive programming and erasing
cause the memory cell to no longer operate properly.

FWE pin application/disconnection (see figures 17.36 to 17.38): FWE pin applicatio
carried out when MCU operation is in a stable condition. If MCU operation is not stabl
FWE pin low and set the protection state.

The following points must be observed concerning FWE pin application and disconnec
prevent unintentional programming or erasing of flash memory:

- Apply the FWE pin when the V_{CC} voltage has stabilized within its rated voltage rang
the FWE pin when oscillation has stabilized (after the oscillation settling time t_{OSC1} h
elapsed).
- In boot mode, apply and disconnect the FWE pin during a reset.
- In user program mode, the FWE pin can be switched between high and low level re
the reset state. FWE pin input can also be switched during program execution in fla
- Do not apply the FWE pin if program runaway has occurred.
- Disconnect the FWE pin only when the SWE, ESU, PSU, EV, PV, P, and E bits in
and FLMCR2 are cleared.
- Make sure that the SWE, ESU, PSU, EV, PV, P, and E bits are not set by mistake w
applying or disconnecting the FWE pin.

recommended algorithm enables programming and erasing to be carried out without
device to voltage stress or sacrificing program data reliability. When setting the P or E
FLMCR1, the watchdog timer should be set beforehand as a precaution against progra
etc.

Do not set or clear the SWE bit during program execution in flash memory: Clea
bit before executing a program or reading data in flash memory. When the SWE bit is
flash memory can be rewritten, but flash memory should only be accessed for verify o
(verification during programming/erasing). Similarly, when using the RAM emulation
while a high level is being input to the FWE pin, the SWE bit must be cleared before e
program or reading data in flash memory. However, the RAM area overlapping flash
space can be read and written to regardless of whether the SWE bit is set or cleared.

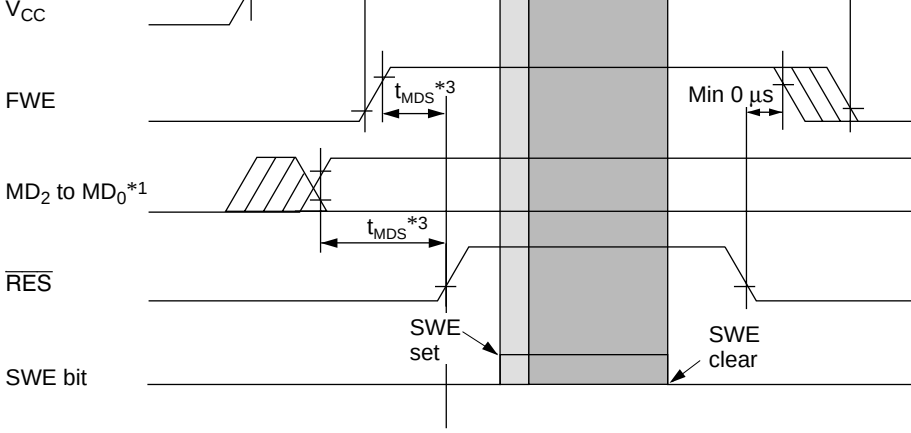
Do not use interrupts while flash memory is being programmed or erased: All int
requests, including NMI, should be disabled during FWE application to give priority t
program/erase operations (including when RAM is being used to emulate flash memo

Also, it is necessary to prohibit release of the bus.

Do not perform additional programming. Erase the memory before reprogramm
board programming, perform only one programming operation on a 32-byte programm
block. In writer mode, too, perform only one programming operation on a 128-byte pr
unit block. Programming should be carried out with the entire programming unit bloc

Before programming, check that the chip is correctly mounted in the PROM pro
Overcurrent damage to the device can result if the index marks on the PROM program
socket adapter, and chip are not correctly aligned.

Do not touch the socket adapter or chip during programming: Touching either of
cause contact faults and write errors.

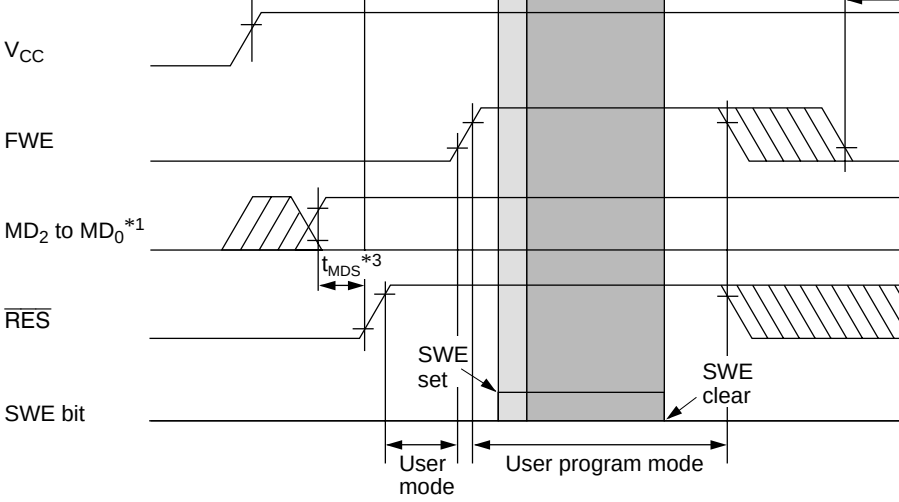


Flash memory access disabled period
(x: Wait time after SWE setting)*2

Flash memory reprogrammable period
(Flash memory program execution and data read, other than verify, are disabled.)

Notes: 1. Always fix the level by pulling down or pulling up the mode pins (MD_2 to MD_0) u
powering off, except for mode switching.
2. See section 20.1.6, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns.

Figure 17.36 Powering On/Off Timing (Boot Mode)

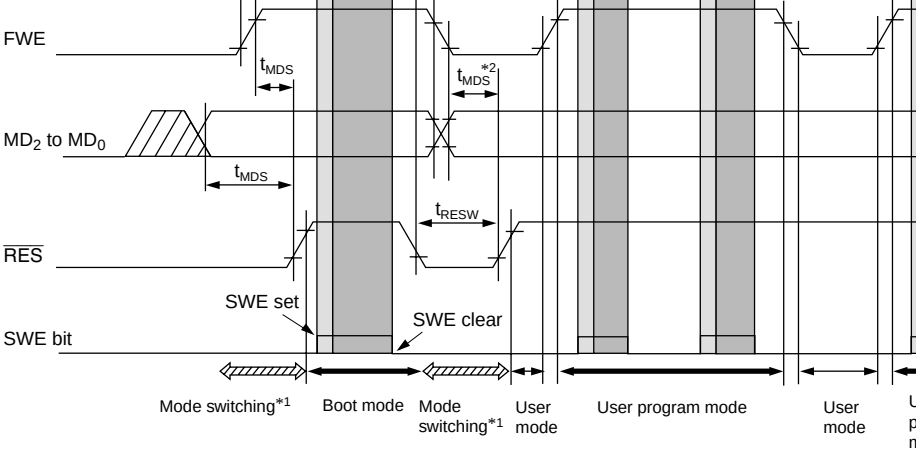


Flash memory access disabled period
(x: Wait time after SWE setting)*2

Flash memory reprogrammable period
(Flash memory program execution and data read, other than verify, are disabled.)

Notes: 1. Always fix the level by pulling down or pulling up the mode pins (MD_2 to MD_0) u
powering off, except for mode switching.
2. See section 20.1.6, Flash Memory Characteristics.
3. Mode programming setup time t_{MDS} (min) = 200 ns.

Figure 17.37 Powering On/Off Timing (User Program Mode)



Flash memory access disabled period
(x: Wait time after SWE setting)*3

Flash memory reprogammable period
(Flash memory program execution and data read, other than verify, are disabled.)

Notes: 1. In transition to the boot mode and transition from the boot mode to another mode, mode switching input is necessary.
During this switching period (period during which a low level is input to the RES pin), the state of t dual port and bus control output signals (AS, RD, HUR, LWR) changes.
Therefore, do not use these pins as output signals during this switching period.
2. When making a transition from the boot mode to another mode, the mode programming setup tim t_{MDS} (min) = 200 ns relative to the RES clear timing is necessary.
3. See section 20.1.6, Flash Memory Characteristics.

Figure 17.38 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode

		Status	
Register	**Bit**	**F–ZTAT Version**	**Mask-ROM Version**
FLMCR1	FWE	0: Application software running	0: Is not read out
		1: Programming	1: Application softwa

Note: This difference applies to all the F-ZTAT versions and all the mask-ROM versio different ROM size.

The clock pulse generator consists of an oscillator circuit, a duty adjustment circuit, a speed clock divider, and a bus master clock selection circuit.

18.1.1 Block Diagram

Figure 18.1 shows a block diagram of the clock pulse generator.

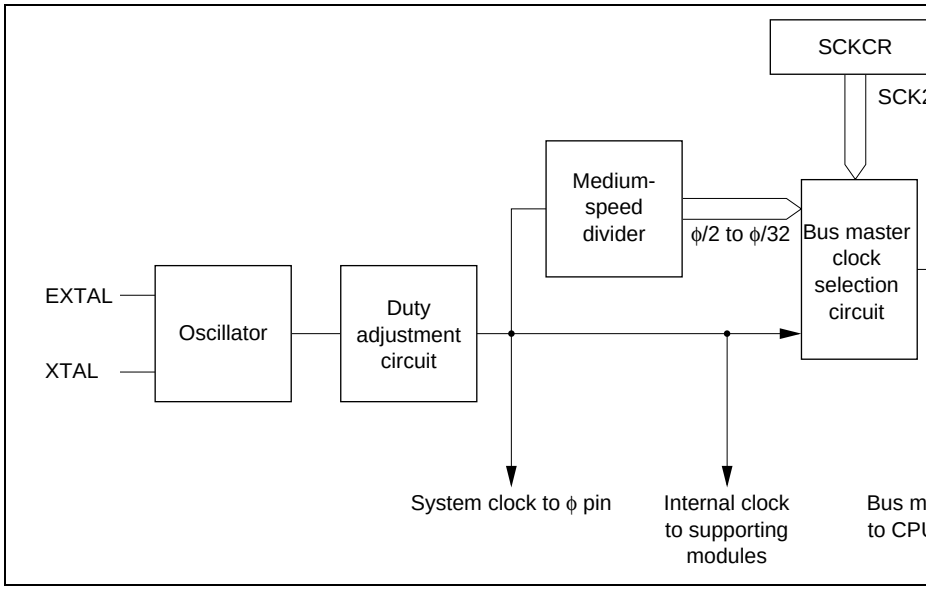


Figure 18.1 Block Diagram of Clock Pulse Generator

18.2 Register Descriptions

18.2.1 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1
		PSTOP	—	—	—	—	SCK2	SCK1
Initial value	:	0	0	0	0	0	0	0
R/W	:	R/W	R/W	—	—	—	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ϕ clock output control and r
speed mode control.

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

Bit 7—ϕ Clock Output Disable (PSTOP): Controls ϕ output.

	Description			
Bit 7				
PSTOP	**Normal Operation**	**Sleep Mode**	**Software Standby Mode**	**Hardwa Standby**
0	ϕ output (initial value)	ϕ output	Fixed high	High imp
1	Fixed high	Fixed high	Fixed high	High imp

Bit 6—Reserved: This bit can be read or written to, but only 0 should be written.

Bits 5 to 3—Reserved: Read-only bits, always read as 0.

		1	Medium-speed clock is ϕ/8
1	0	0	Medium-speed clock is ϕ/16ϕ
		1	Medium-speed clock is ϕ/32
	1	—	—

18.3 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an exter

18.3.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the exampl 18.2. Select the damping resistance R_d according to table 18.2. An AT-cut parallel-res crystal should be used.

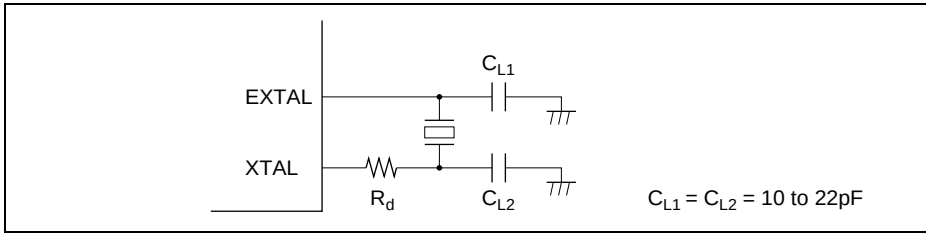


Figure 18.2 Connection of Crystal Resonator (Example)

Table 18.2 Damping Resistance Value

Frequency (MHz)	2	4	8	10	12	16
R_d (Ω)	1k	500	200	100	0	0

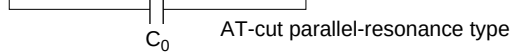


Figure 18.3 Crystal Resonator Equivalent Circuit

Table 18.3 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	10	12	16
R_S max (Ω)	500	120	80	60	60	50
C_0 max (pF)	7	7	7	7	7	7

Note on Board Design: When a crystal resonator is connected, the following points sh
noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction
interfering with correct oscillation. See figure 18.4.

When designing the board, place the crystal resonator and its load capacitors as close a
to the XTAL and EXTAL pins.

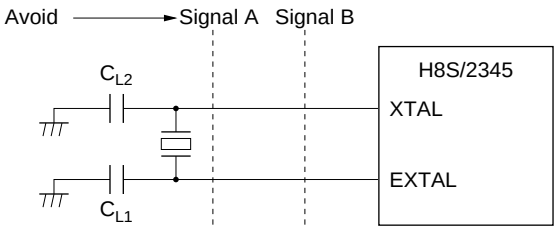


Figure 18.4 Example of Incorrect Board Design

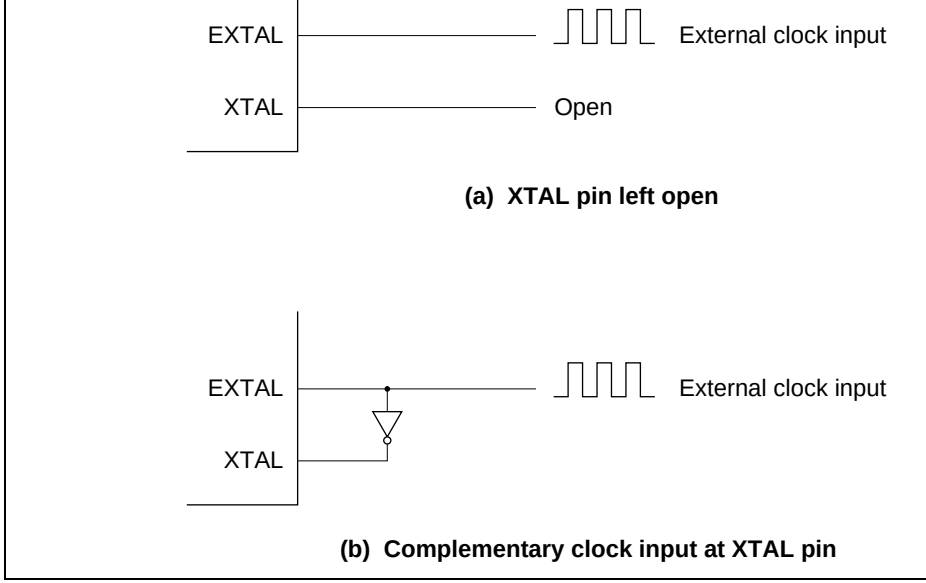


Figure 18.5 External Clock Input (Examples)

Item	Symbol	Min	Max	Min	Max	Unit	Test Conditions
External clock input low pulse width	t_{EXL}	40	—	20	—	ns	Figure 18.6
External clock input high pulse width	t_{EXH}	40	—	20	—	ns	
External clock rise time	t_{EXr}	—	10	—	5	ns	
External clock fall time	t_{EXf}	—	10	—	5	ns	
Clock low pulse width level	t_{CL}	0.4	0.6	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz
		80	—	80	—	ns	$\phi < 5$ MHz
Clock high pulse width level	t_{CH}	0.4	0.6	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz
		80	—	80	—	ns	$\phi < 5$ MHz

Note: * ZTAT, mask ROM, and ROMless versions only.

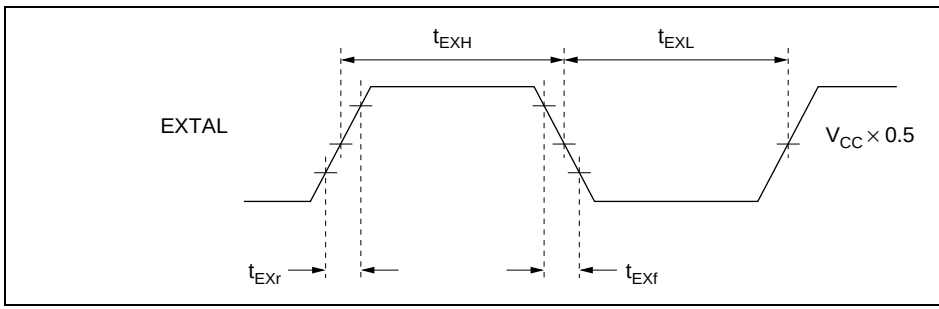


Figure 18.6 External Clock Input Timing

18.6 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the system clock (ϕ) or one of the mediu
clocks (ϕ/2, ϕ/4, or ϕ/8, ϕ/16, and ϕ/32) to be supplied to the bus master, according to
of the SCK2 to SCK0 bits in SCKCR.

modules, and so on.

The H8S/2345 Group 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 a CPU mode, medium-spee
CPU and bus master mode, and module stop mode is an on-chip supporting module m
(including bus masters other than the CPU). A combination of these modes can be set.

After a reset, the H8S/2345 Group is in high-speed mode.

Table 19.1 shows the conditions for transition to the various modes, the status of the C
supporting modules, etc., and the method of clearing each mode.

mode						speed*1	
Sleep mode	Instruction	Interrupt	Functions	Halted	Retained	High speed	Functions
Module stop mode	Control register		Functions	High/ medium speed	Functions	Halted	Retained/ reset*2
Software standby mode	Instruction	External interrupt	Halted	Halted	Retained	Halted	Retained/ reset*2
Hardware standby mode	Pin		Halted	Halted	Undefined	Halted	Reset

Notes: 1. The bus master operates on the medium-speed clock, and other on-chip sup modules on the high-speed clock.

2. The SCI and A/D converter are reset, and other on-chip supporting modules state.

19.1.1 Register Configuration

Power-down modes are controlled by the SBYCR, SCKCR, and MSTPCR registers. Ta summarizes these registers.

Table 19.2 Power-Down Mode Registers

Name	Abbreviation	R/W	Initial Value	Ad
Standby control register	SBYCR	R/W	H'08	H'
System clock control register	SCKCR	R/W	H'00	H'
Module stop control register H	MSTPCRH	R/W	H'3F	H'
Module stop control register L	MSTPCRL	R/W	H'FF	H'

Note: * Lower 16 bits of the address.

R/W . R/W R/W R/W R/W R/W — —

SBYCR is an 8-bit readable/writable register that performs software standby mode co

SBYCR is initialized to H'08 by a reset and in hardware standby mode. It is not initial
software standby mode.

Bit 7—Software Standby (SSBY): Specifies a transition to software standby mode. I
to 1 when software standby mode is released by an external interrupt, and a transition
normal operation. The SSBY bit should be cleared by writing 0 to it.

Bit 7

SSBY	Description
0	Transition to sleep mode after execution of SLEEP instruction
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
waits for the clock to stabilize when software standby mode is cleared by an external i
With crystal oscillation, refer to table 19.4 and make a selection according to the oper
frequency so that the standby time is at least 8 ms (the oscillation stabilization time). W
external clock, any selection* can be made.

Note: * The 16-state standby time cannot be used in the F-ZTAT version; a standb
8192 states or longer should be used.

	1	Standby time = 262144 states
1	0	Reserved
	1	Standby time = 16 states*

Note: * Not used on the F-ZTAT version.

Bit 3—Output Port Enable (OPE): Specifies whether the output of the address bus a control signals ($\overline{CS0}$ to $\overline{CS3}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$) is retained or set to the high-imped in software standby mode.

Bit 3

OPE	Description
0	In software standby mode, address bus and bus control signals are high-im
1	In software standby mode, address bus and bus control signals retain outp (

Bits 2 and 1—Reserved: Read-only bits, always read as 0.

Bit 0—Reserved: This bit can be read or written to, but only 0 should be written.

SCKCR is an 8-bit readable/writable register that performs φ clock output control and
speed mode control.

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

Bit 7—φ Clock Output Disable (PSTOP): Controls φ output.

Bit 7	Description			
PSTOP	Normal Operating Mode	Sleep Mode	Software Standby Mode	Hardwar Mode
0	φ output (initial value)	φ output	Fixed high	High imp
1	Fixed high	Fixed high	Fixed high	High imp

Bits 6—Reserved: This bit can be read or written to, but only 0 should be written.

Bits 5 to 3—Reserved: Read-only bits, always read as 0.

Bits 2 to 0—System Clock Select (SCK2 to SCK0): These bits select the clock for t
master.

Bit 2	Bit 1	Bit 0	
SCK2	SCK1	SCK0	Description
0	0	0	Bus master in high-speed mode
		1	Medium-speed clock is φ/2
	1	0	Medium-speed clock is φ/4
		1	Medium-speed clock is φ/8
1	0	0	Medium-speed clock is φ/16
		1	Medium-speed clock is φ/32
	1	—	—

MSTPCR is a 16-bit readable/writable register that performs module stop mode control

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not ini
software standby mode.

Bits 15 to 0—Module Stop (MSTP 15 to MSTP 0): These bits specify module stop m
table 19.3 for the method of selecting on-chip supporting modules.

Bits 15 to 0

MSTP15 to MSTP0	Description
0	Module stop mode cleared
1	Module stop mode set

19.3 Medium-Speed Mode

When the SCK2 to SCK0 bits in SCKCR are set to 1, the operating mode changes to m
speed mode as soon as the current bus cycle ends. In medium-speed mode, the CPU op
the operating clock (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) specified by the SCK2 to SCK0 bits. T
masters other than the CPU (the DTC) also operate in medium-speed mode. On-chip su
modules other than the bus masters always operate on the high-speed clock (ϕ).

In medium-speed mode, a bus access is executed in the specified number of states with
the bus master operating clock. For example, if ϕ/4 is selected as the operating clock, o
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
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 bit in SBYCR is cleared to 0, a tran
made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode

Figure 19.1 shows the timing for transition to and clearance of medium-speed mode.

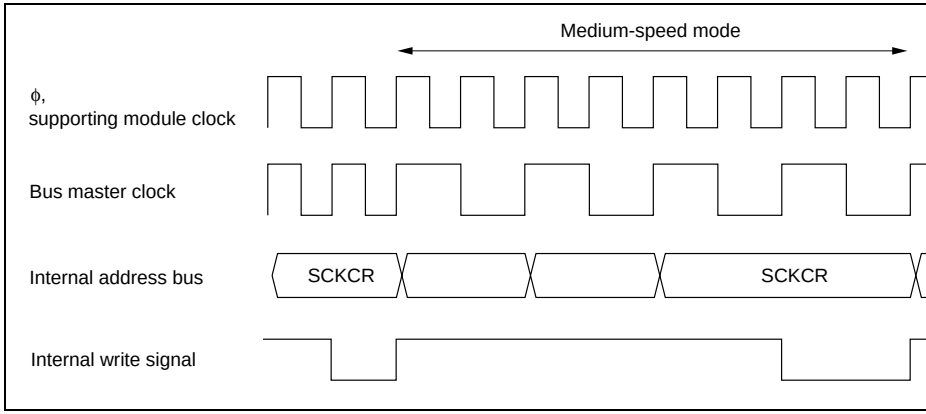


Figure 19.1 Medium-Speed Mode Transition and Clearance Timing

19.4 Sleep Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, the C sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's interna are retained. Other supporting modules do not stop.

Sleep mode is cleared by a reset or any interrupt, and the CPU returns to the normal pr execution state via the exception handling state. Sleep mode is not cleared if interrupts disabled, or if interrupts other than NMI are masked by the CPU.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

independently.

Table 19.3 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 th
starts operating at the end of the bus cycle. In module stop mode, the internal states of
other than the SCI and A/D converter are retained.

After reset clearance, all modules other than DTC are in module stop mode.

When an on-chip supporting module is in module stop mode, read/write access to its re
disabled.

If a transition is made to sleep mode when all modules are stopped (MSTPCR = H'FFF
modules other than the 8-bit timers are stopped (MSTPCR = H'EFFF), operation of the
controller and I/O ports is also halted, enabling current dissipation to be further reduce

	MSTP10	D/A converter
	MSTP9	A/D converter
	MSTP8	—
MSTPCRL	MSTP7	—
	MSTP6	Serial communication interface (SCI) channel 1
	MSTP5	Serial communication interface (SCI) channel 0
	MSTP4	—
	MSTP3	—
	MSTP2	—
	MSTP1	—
	MSTP0	—

Note: Bits 15, 11, 8, 7, and 4 to 0 can be read or written to, but do not affect operation

19.5.2 Usage Notes

DTC Module Stop: Depending on the operating status of the DTC, the MSTP14 bit
set to 1. Setting of the DTC module stop mode should be carried out only when the re
module is not activated.

For details, refer to section 7, Data Transfer Controller.

On-Chip Supporting Module Interrupt: Relevant interrupt operations cannot be pe
module stop mode. Consequently, if module stop mode is entered when an interrupt h
requested, it will not be possible to clear the CPU interrupt source or the DTC activati
Interrupts should therefore be disabled before entering module stop mode.

Writing to MSTPCR: MSTPCR should only be written to by the CPU.

address bus and bus control signals are placed in the high-impedance state or retain the state can be specified by the OPE bit in SBYCR.

In this mode the oscillator stops, and therefore power dissipation is significantly reduce

19.6.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IR}}$ means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an NMI or IRQ0 to IRQ2 interrupt request signal is input, clock oscillation st after the elapse of the time set in bits STS2 to STS0 in SYSCR, stable clocks are su the entire H8S/2345 Group chip, software standby mode is cleared, and interrupt ex handling is started.

  When clearing software standby mode with an IRQ0 to IRQ2 interrupt, set the corre enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ is generated. Software standby mode cannot be cleared if the interrupt has been mas CPU side or has been designated as a DTC activation source.

- Clearing with the $\overline{\text{RES}}$ pin

  When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as clo oscillation starts, clocks are supplied to the entire H8S/2345 Group chip. Note that pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin goes high 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.

Table 19.4 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz
0	0	0	8192 states	0.41	0.51	0.68	0.8	1.0	1.3	2.0
		1	16384 states	0.82	1.0	1.3	1.6	2.0	2.7	4.1
	1	0	32768 states	1.6	2.0	2.7	3.3	4.1	5.5	8.2
		1	65536 states	3.3	4.1	5.5	6.6	8.2	10.9	16.4
1	0	0	131072 states	6.6	8.2	10.9	13.1	16.4	21.8	32.8
		1	262144 states	13.1	16.4	21.8	26.2	32.8	43.6	65.6
	1	0	Reserved	—	—	—	—	—	—	—
		1	16 states	0.8	1.0	1.3	1.6	2.0	2.7	4.0

: Recommended time setting

Using an External Clock: Any value can be set. Normally, use of the minimum time recommended.*

Note: * The 16-state standby time cannot be used in the F-ZTAT version; a standby 8192 states or longer should be used.

19.6.4 Software Standby Mode Application Example

Figure 19.2 shows an example in which a transition is made to software standby mode falling edge on the NMI pin, and software standby mode is cleared at the rising edge o pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSB to 1, and a SLEEP instruction is executed, causing a transition to software standby mo

Software standby mode is then cleared at the rising edge on the NMI pin.

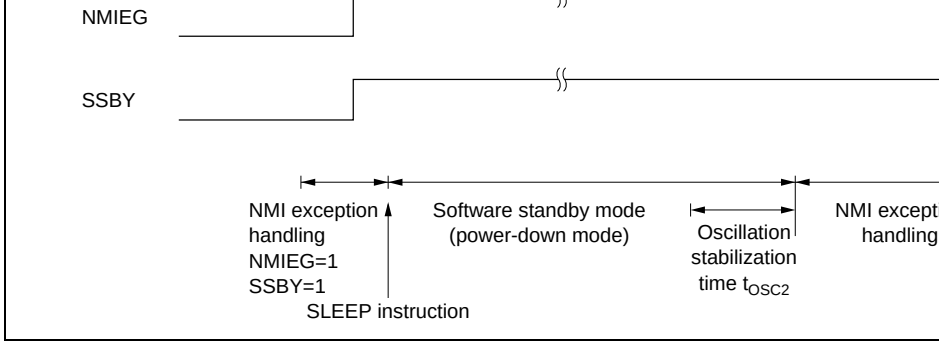


Figure 19.2 Software Standby Mode Application Example

19.6.5 Usage Notes

I/O Port Status: In software standby mode, I/O port states are retained. If the OPE bit the address bus and bus control signal output is also retained. Therefore, there is no red current dissipation for the output current when a high-level signal is output.

Current Dissipation during Oscillation Stabilization Wait Period: Current dissipati increases during the oscillation stabilization wait period.

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 driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD_2 to MD_0) while the H8S/2345 Group is standby mode.

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation i Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (at least 8 ms—oscillation stabilization time—when using a crystal oscillator). When the $\overline{\text{RES}}$ pin is s driven high, a transition is made to the program execution state via the reset exception state.

19.7.2 Hardware Standby Mode Timing

Figure 19.3 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin for the oscillation stabilization time, then changing the $\overline{\text{RES}}$ pin from low to high.

Figure 19.3 Hardware Standby Mode Timing (Example)

19.8 ϕ Clock Output Disabling Function

Output of the ϕ clock can be controlled by means of the PSTOP bit in SCKCR, and DD
corresponding port. When the PSTOP bit is set to 1, the ϕ clock stops at the end of the
and ϕ output goes high. ϕ clock output is enabled when the PSTOP bit is cleared to 0. W
for the corresponding port is cleared to 0, ϕ clock output is disabled and input port mod
Table 19.5 shows the state of the ϕ pin in each processing state.

Table 19.5 ϕ Pin State in Each Processing State

DDR	0	1	
PSTOP	—	0	
Hardware standby mode	High impedance		
Software standby mode	High impedance	Fixed high	
Sleep mode	High impedance	ϕ output	Fixe
Normal operating state	High impedance	ϕ output	Fixe

Table 20.1 Absolute Maximum Ratings

Item	Symbol	Value
Power supply voltage	V_{CC}	–0.3 to +7.0
Input voltage (FWE)*1	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (except port 4)	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (port 4)	V_{in}	–0.3 to AV_{CC} +0.3
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3
Analog power supply voltage	AV_{CC}	–0.3 to +7.0
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	Regular specifications: –20 to +75*2
		Wide-range specifications: –40 to +85*
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the chip may result if absolute maximum rating are ex

Notes: 1. Never apply 12 V to any of the pins. Doing so could permanently damage t
2. The operating temperature range for flash memory programming/erase ope
T_a = 0 to +75°C (regular specifications), T_a = 0 to +85°C (wide-range specifi

Item		Symbol	Min	Typ	Max	Unit	Test Co
Schmitt trigger input voltage	Port 2, $\overline{IRQ0}$ to $\overline{IRQ7}$	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0, FWE	V_{IH}	$V_{CC} - 0.7$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Ports 1, 3, A to G		2.0	—	$V_{CC} + 0.3$	V	
	Port 4		2.0	—	$AV_{CC} + 0.3$	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0, FWE	V_{IL}	–0.3	—	0.5	V	
	NMI, EXTAL, Ports 1, 3, 4, A to G		–0.3	—	0.8	V	
Output high voltage	All output pins	V_{OH}	$V_{CC} - 0.5$	—	—	V	I_{OH} = –2
			3.5	—	—	V	I_{OH} = –1
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6
	Ports 1, A to C		—	—	1.0	V	I_{OL} = 10
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	V_{in} = 0.5 to V
	$\overline{STBY}$, NMI, MD_2 to MD_0, FWE		—	—	1.0	µA	
	Port 4		—	—	1.0	µA	V_{in} = 0.5 to A
Three-state leakage current (off state)	Ports 1 to 3, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V

	and NMI						
Current dissipation*2	Normal operation	I_{CC}*4	—	60 (5.0 V)	89	mA	f = 20
	Sleep mode		—	40 (5.0 V)	73	mA	f = 20
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq 50$
			—	—	20		50°C <
	During flash memory programming/ erase		—	70 (5.0 V)	89	mA	0°C ≤ f = 20
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	0.8 (5.0 V)	2.0	mA	
	Idle		—	0.01	5.0	µA	
Reference current	During A/D and D/A conversion	AI_{CC}	—	1.9 (5.0 V)	3.0	mA	V_{ref} = 5
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, AV_{SS}, an open.
Connect AV_{CC} and V_{ref} to V_{CC}, and connect AV_{SS} to V_{SS}.

2. Current dissipation values are for V_{IH} min = V_{CC} –0.5 V and V_{IL} max = 0.5V w pins unloaded and the on-chip pull-up transistors in the off state.

3. The values are for $V_{RAM} \leq V_{CC} < 4.5V$, 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) + 0.80 (mA/(MHz × V)) × V_{CC} × f [normal mode]
I_{CC} max = 1.0 (mA) + 0.65 (mA/(MHz × V)) × V_{CC} × f [sleep mode]

Permissible output low current (total)	Total of 28 pins including ports 1 and A to C	$\sum I_{OL}$	—	—	80
	Total of all output pins, including the above		—	—	120
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0
Permissible output high current (total)	Total of all output pins	$\sum -I_{OH}$	—	—	40

Notes: 1. To protect chip reliability, do not exceed the output current values in table 20

2. When driving a darlington pair or LED directly, always insert a current-limitin the output line, as show in figures 20.1 and 20.2.

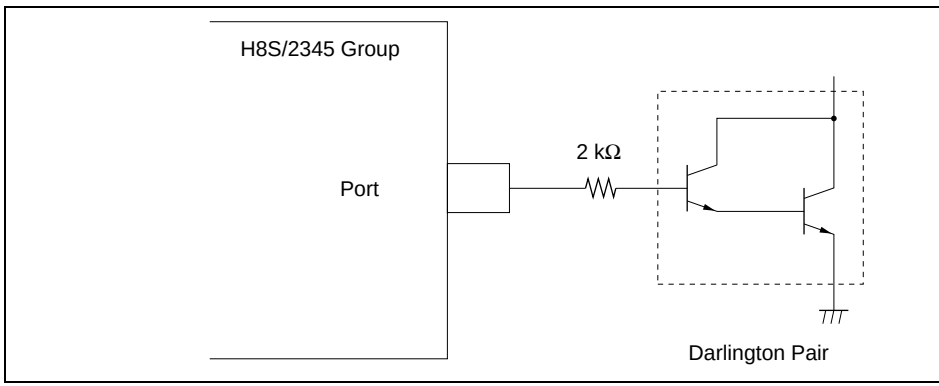


Figure 20.1 Darlington Pair Drive Circuit (Example)

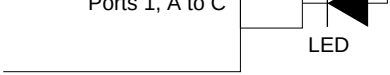


Figure 20.2 LED Drive Circuit (Example)

20.1.3 AC Characteristics

Figure 20.3 show, the test conditions for the AC characteristics.

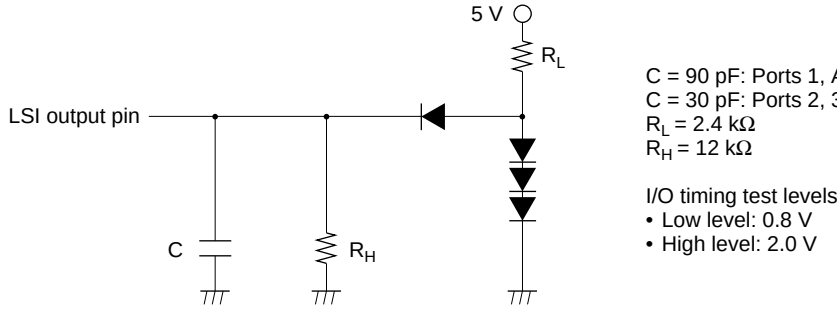


Figure 20.3 Output Load Circuit

Clock cycle time	t_{cyc}	50	500	ns	
Clock high pulse width	t_{CH}	20	—	ns	Figure 2
Clock low pulse width	t_{CL}	20	—	ns	
Clock rise time	t_{Cr}	—	5	ns	
Clock fall time	t_{Cf}	—	5	ns	
Clock oscillator setting time at reset (crystal)	t_{OSC1}	10	—	ms	Figure 2
Clock oscillator setting time in software standby (crystal)	t_{OSC2}	8	—	ms	Figure 2
External clock output stabilization delay time	t_{DEXT}	500	—	µs	Figure 2

$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	ns	Figure
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI reset setup time	t_{NMIRS}	200	—	ns	
NMI reset hold time	t_{NMIRH}	200	—	ns	
Mode programming setup time	t_{MDS}	200	—	ns	
NMI setup time	t_{NMIS}	150	—	ns	Figure
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{\text{IRQ}}$ setup time	t_{IRQS}	150	—	ns	
$\overline{\text{IRQ}}$ hold time	t_{IRQH}	10	—	ns	
$\overline{\text{IRQ}}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

Address delay time	t_{AD}	—	20	ns	Figure 2
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 15$	—	ns	Figure 2
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	20	ns	
$\overline{AS}$ delay time	t_{ASD}	—	20	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	20	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	20	ns	
$\overline{CAS}$ delay time	t_{CASD}	—	20	ns	
Read data setup time	t_{RDS}	15	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time 1	t_{ACC1}	—	$1.0 \times t_{cyc} - 25$	ns	
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 25$	ns	
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc} - 25$	ns	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 25$	ns	
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 25$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	20	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	20	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 20$	—	ns	
Write data delay time	t_{WDD}	—	30	ns	
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 20$	—	ns	
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	30	—	ns	Figure 2
$\overline{WAIT}$ hold time	t_{WTH}	5	—	ns	
$\overline{BREQ}$ setup time	t_{BRQS}	30	—	ns	Figure 2
$\overline{BACK}$ delay time	t_{BACD}	—	15	ns	
Bus-floating time	t_{BZD}	—	50	ns	

Item			Symbol	Min	Max	Unit	T C
I/O port	Output data delay time		t_{PWD}	—	50	ns	F
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
TPU	Timer output delay time		t_{TOCD}	—	50	ns	F
	Timer input setup time		t_{TICS}	30	—		
	Timer clock input setup time		t_{TCKS}	30	—	ns	F
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
8-bit timer	Timer output delay time		t_{TMOD}	—	50	ns	F
	Timer reset input setup time		t_{TMRS}	30	—	ns	F
	Timer clock input setup time		t_{TMCS}	30	—	ns	F
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—		
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	F
		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}	—	50	ns	F
	Receive data setup time (synchronous)		t_{RXS}	50	—	ns	
	Receive data hold time (synchronous)		t_{RXH}	50	—	ns	
A/D converter	Trigger input setup time		t_{TRGS}	30	—	ns	F

T_a = −40 to +85°C (wide-range specifications)

Item	Min	Typ	Max	Un
Resolution	10	10	10	bit
Conversion time	—	—	6.7	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	10*1	kΩ
	—	—	5*2	
Nonlinearity error	—	—	±3.5	LS
Offset error	—	—	±3.5	LS
Full-scale error	—	—	±3.5	LS
Quantization	—	—	±0.5	LS
Absolute accuracy	—	—	±4.0	LS

Notes: 1. $\phi \leq 12$ MHz
2. $\phi > 12$ MHz

T_a = −40 to +85°C (Wide-range specifications)

Item	Min	Typ	Max	Unit	Test Con
Resolution	8	8	8	bit	
Conversion time	—	—	10	µs	20-pF cap
Absolute accuracy	—	±1.0	±1.5	LSB	2-MΩ resi
	—	—	±1.0	LSB	4-MΩ resi

wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Programming time*1 *2 *4		t_P	—	10	200	ms/ 32 bytes
Erase time*1 *3 *5		t_E	—	100	1200	ms/block
Reprogramming count		N_{WEC}	—	—	100	Times
Programming	Wait time after setting SWE bit*1	x	10	—	—	µs
	Wait time after setting PSU bit*1	y	50	—	—	µs
	Wait time after setting P bit*1 *4	z	150	—	200	µs
	Wait time after clearing P bit*1	α	10	—	—	µs
	Wait time after clearing PSU bit*1	β	10	—	—	µs
	Wait time after setting PV bit*1	γ	4	—	—	µs
	Wait time after H'FF dummy write*1	ε	2	—	—	µs
	Wait time after clearing PV bit*1	η	4	—	—	µs
	Max. number of programmings*1 *4	N	—	—	1000*5	Times
Erase	Wait time after setting SWE bit*1	x	10	—	—	µs
	Wait time after setting ESU bit*1	y	200	—	—	µs
	Wait time after setting E bit*1 *6	z	5	—	10	µs
	Wait time after clearing E bit*1	α	10	—	—	µs
	Wait time after clearing ESU bit*1	β	10	—	—	µs
	Wait time after setting EV bit*1	γ	20	—	—	µs
	Wait time after H'FF dummy write*1	ε	2	—	—	µs
	Wait time after clearing EV bit*1	η	5	—	—	µs
	Max. number of erases*1 *6	N	120	—	240	Times

Notes: 1. Time settings should be made in accordance with the programming/erase al
2. Programming time per 32 bytes. (Indicates the total time the P bit in the flash control register 1 (FLMCR1) is set. The program verification time is not inclu
3. Time to erase one block. (Indicates the total time the E bit in FLMCR1 is set. verification time is not included.)

When z = 10 [ms], N = 120 times

Item	Symbol	Value
Power supply voltage	V_{CC}	–0.3 to +7.0
Programming voltage*	V_{PP}	–0.3 to +13.5
Input voltage (except port 4)	V_{in}	–0.3 to V_{CC} +0.3
Input voltage (port 4)	V_{in}	–0.3 to AV_{CC} +0.3
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3
Analog power supply voltage	AV_{CC}	–0.3 to +7.0
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3
Operating temperature	T_{opr}	Regular specifications: –20 to +75
		Wide-range specifications: –40 to +85
Storage temperature	T_{stg}	–55 to +125

Caution: Permanent damage to the chip may result if absolute maximum rating are exc

Note: * ZTAT version only.

Item		Symbol	Min	Typ	Max	Unit	Test
Schmitt trigger input voltage	Port 2, $\overline{IRQ0}$ to $\overline{IRQ7}$	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0	V_{IH}	$V_{CC} - 0.7$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	V	
	Ports 1, 3, A to G		2.0	—	$V_{CC} + 0.3$	V	
	Port 4		2.0	—	$AV_{CC} + 0.3$	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0	V_{IL}	–0.3	—	0.5	V	
	NMI, EXTAL, Ports 1, 3, 4, A to G		–0.3	—	0.8	V	
Output high voltage	All output pins	V_{OH}	$V_{CC} - 0.5$	—	—	V	I_{OH} =
			3.5	—	—	V	I_{OH} =
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} =
	Ports 1, A to C		—	—	1.0	V	I_{OL} =
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	V_{in} = 0.5 to
	$\overline{STBY}$, NMI, MD_2 to MD_0		—	—	1.0	µA	
	Port 4		—	—	1.0	µA	V_{in} = 0.5 to V
Three-state leakage current (off state)	Ports 1 to 3, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to

	and NMI						
Current dissipation*2	Normal operation	I_{CC}*4	—	60 (5.0 V)	89	mA	f = 20
	Sleep mode		—	40 (5.0 V)	73	mA	f = 20
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq 5$
			—	—	20		50°C <
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	0.8 (5.0 V)	2.0	mA	
	Idle		—	0.01	5.0	µA	
Reference current	During A/D and D/A conversion	AI_{CC}	—	1.9 (5.0 V)	3.0	mA	V_{ref} = 5
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, AV_{SS}, and open.
Connect AV_{CC} and V_{ref} to V_{CC}, and connect AV_{SS} to V_{SS}.

2. Current dissipation values are for V_{IH} min = V_{CC} –0.5 V and V_{IL} max = 0.5V wi pins unloaded and the on-chip pull-up transistors in the off state.
3. The values are for $V_{RAM} \leq V_{CC} < 4.5V$, 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) + 0.80 (mA/(MHz × V)) × V_{CC} × f [normal mode]
I_{CC} max = 1.0 (mA) + 0.65 (mA/(MHz × V)) × V_{CC} × f [sleep mode]

voltage		$V_T^+ - V_T^-$	$V_{CC} \times 0.07$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_2 to MD_0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Ports 1, 3, A to G		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 4		$V_{CC} \times 0.7$	—	AV_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_2 to MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, Ports 1, 3, 4, A to G		–0.3	—	$V_{CC} \times 0.2$	V	V_{CC} <
					0.8		V_{CC} =
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –
			V_{CC} – 1.0	—	—	V	I_{OH} = –
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1
	Ports 1, A to C		—	—	1.0	V	$V_{CC} \leq$ I_{OL} = 5 4.0 < I_{OL} = 1
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	V_{in} = 0.5 to
	$\overline{STBY}$, NMI, MD_2 to MD_0		—	—	1.0	µA	
	Port 4		—	—	1.0	µA	V_{in} = 0.5 to
Three-state leakage current (off state)	Ports 1 to 3, A to G	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to

	and NMI						
Current dissipation*2	Normal operation	I_{CC}*4	—	18 (3.0 V)	45	mA	f = 10 M
	Sleep mode		—	11 (3.0 V)	37	mA	f = 10 M
	Standby mode*3		—	0.01	5.0	µA	$T_a \leq 50$
			—	—	20		50°C <
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	0.2 (3.0 V)	2.0	mA	
	Idle		—	0.01	5.0	µA	
Reference current	During A/D and D/A conversion	AI_{CC}	—	1.2 (3.0 V)	3.0	mA	V_{ref} = 3
	Idle		—	0.01	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. If the A/D and D/A converters are not used, do not leave the AV_{CC}, AV_{SS}, and open.
Connect AV_{CC} and V_{ref} to V_{CC}, and connect AV_{SS} to V_{SS}.

2. Current dissipation values are for V_{IH} min = V_{CC} –0.5 V and V_{IL} max = 0.5V wi pins unloaded and the on-chip pull-up transistors 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 = 0.3V

4. I_{CC} depends on V_{CC} and f as follows:
I_{CC} max = 1.0 (mA) + 0.80 (mA/(MHz × V)) × V_{CC} × f [normal mode]
I_{CC} max = 1.0 (mA) + 0.65 (mA/(MHz × V)) × V_{CC} × f [sleep mode]

Permissible output low current (total)	Total of 28 pins including port 1 and A to C	$\sum I_{OL}$	—	—	80
	Total of all output pins, including the above		—	—	120
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2.0
Permissible output high current (total)	Total of all output pins	$\sum -I_{OH}$	—	—	40

Notes: 1. To protect chip reliability, do not exceed the output current values in table 2

2. When driving a darlington pair or LED directly, always insert a current-limiti the output line, as show in figures 20.4 and 20.5.

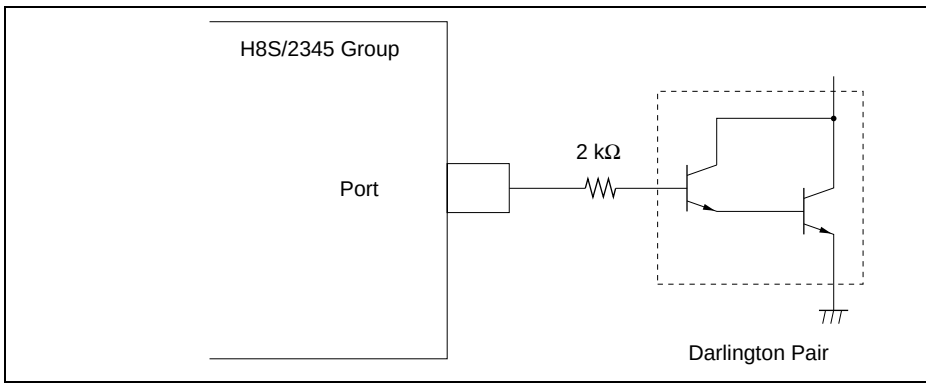


Figure 20.4 Darlington Pair Drive Circuit (Example)

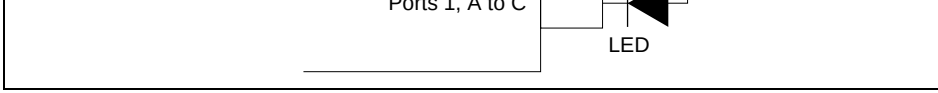


Figure 20.5 LED Drive Circuit (Example)

20.2.3 AC Characteristics

Figure 20.6 show, the test conditions for the AC characteristics.

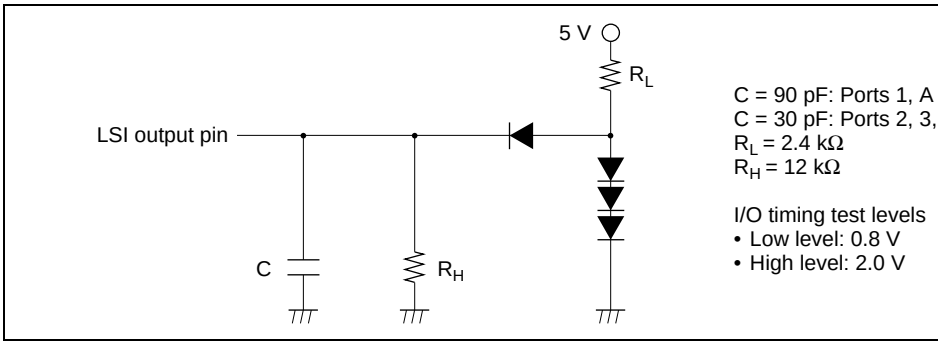


Figure 20.6 Output Load Circuit

ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test**
Clock cycle time	t_{cyc}	100	500	50	500	ns	Figure
Clock high pulse width	t_{CH}	35	—	20	—	ns	
Clock low pulse width	t_{CL}	35	—	20	—	ns	
Clock rise time	t_{Cr}	—	15	—	5	ns	
Clock fall time	t_{Cf}	—	15	—	5	ns	
Clock oscillator setting time at reset (crystal)	t_{OSC1}	20	—	10	—	ms	Figure
Clock oscillator setting time in software standby (crystal)	t_{OSC2}	8	—	8	—	ms	Figure
External clock output stabilization delay time	t_{DEXT}	500	—	500	—	µs	Figure

ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test C**
$\overline{\text{RES}}$ setup time	t_{RESS}	200	—	200	—	ns	Figure
$\overline{\text{RES}}$ pulse width	t_{RESW}	20	—	20	—	t_{cyc}	
NMI reset setup time	t_{NMIRS}	250	—	200	—	ns	
NMI reset hold time	t_{NMIRH}	200	—	200	—	ns	
NMI setup time	t_{NMIS}	250	—	150	—	ns	Figure
NMI hold time	t_{NMIH}	10	—	10	—	ns	
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	200	—	ns	
$\overline{\text{IRQ}}$ setup time	t_{IRQS}	250	—	150	—	ns	
$\overline{\text{IRQ}}$ hold time	t_{IRQH}	10	—	10	—	ns	
$\overline{\text{IRQ}}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	200	—	ns	

ϕ= 2 to 20 MHz, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)

		Condition A		Condition B			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test**
Address delay time	t_{AD}	—	40	—	20	ns	Figu
Address setup time	t_{AS}	0.5 × t_{cyc} – 30	—	0.5 × t_{cyc} – 15	—	ns	Figu
Address hold time	t_{AH}	0.5 × t_{cyc} – 20	—	0.5 × t_{cyc} – 10	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	40	—	20	ns	
$\overline{AS}$ delay time	t_{ASD}	—	40	—	20	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	40	—	20	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	40	—	20	ns	
$\overline{CAS}$ delay time	t_{CASD}	—	40	—	20	ns	
Read data setup time	t_{RDS}	30	—	15	—	ns	
Read data hold time	t_{RDH}	0	—	0	—	ns	
Read data access time 1	t_{ACC1}	—	1.0 × t_{cyc} – 50	—	1.0 × t_{cyc} – 25	ns	
Read data access time 2	t_{ACC2}	—	1.5 × t_{cyc} – 50	—	1.5 × t_{cyc} – 25	ns	
Read data access time 3	t_{ACC3}	—	2.0 × t_{cyc} – 50	—	2.0 × t_{cyc} – 25	ns	
Read data access time 4	t_{ACC4}	—	2.5 × t_{cyc} – 50	—	2.5 × t_{cyc} – 25	ns	
Read data access time 5	t_{ACC5}	—	3.0 × t_{cyc} – 50	—	3.0 × t_{cyc} – 25	ns	

Write data delay time	t_{WDD}	—	60	—	30	ns	
Write data setup time	t_{WDS}	0.5 × t_{cyc} – 40	—	0.5 × t_{cyc} – 20	—	ns	
Write data hold time	t_{WDH}	0.5 × t_{cyc} – 20	—	0.5 × t_{cyc} – 10	—	ns	
$\overline{\text{WAIT}}$ setup time	t_{WTS}	60	—	30	—	ns	Figure
$\overline{\text{WAIT}}$ hold time	t_{WTH}	10	—	5	—	ns	
$\overline{\text{BREQ}}$ setup time	t_{BRQS}	60	—	30	—	ns	Figure
$\overline{\text{BACK}}$ delay time	t_{BACD}	—	30	—	15	ns	
Bus-floating time	t_{BZD}	—	100	—	50	ns	

Condition B: V_{CC} = AV_{CC} = 5.0 V ± 10%, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

				Condition A		Condition B			
Item			**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test**
I/O port	Output data delay time		t_{PWD}	—	100	—	50	ns	Figur
	Input data setup time		t_{PRS}	50	—	30	—		
	Input data hold time		t_{PRH}	50	—	30	—		
TPU	Timer output delay time		t_{TOCD}	—	100	—	50	ns	Figur
	Timer input setup time		t_{TICS}	50	—	30	—		
	Timer clock input setup time		t_{TCKS}	50	—	30	—	ns	Figur
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—	2.5	—		
8-bit timer	Timer output delay time		t_{TMOD}	—	100	—	50	ns	Figur
	Timer reset input setup time		t_{TMRS}	50	—	30	—	ns	Figur
	Timer clock input setup time		t_{TMCS}	50	—	30	—	ns	Figur
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—	2.5	—		

	Input clock pulse width	t_{SCKW}	0.4	0.6	0.4	0.6	t_{Scyc}	
	Input clock rise time	t_{SCKr}	—	1.5	—	1.5	t_{cyc}	
	Input clock fall time	t_{SCKf}	—	1.5	—	1.5		
	Transmit data delay time	t_{TXD}	—	100	—	50	ns	Figure
	Receive data setup time (synchronous)	t_{RXS}	100	—	50	—	ns	
	Receive data hold time (synchronous)	t_{RXH}	100	—	50	—	ns	
A/D converter	Trigger input setup time	t_{TRGS}	50	—	30	—	ns	Figure

T_a = –40 to +85°C (wide-range specifications)

Condition B: V_{CC} = AV_{CC} = 5.0 V ± 10%, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)

	Condition A			Condition B		
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
Resolution	10	10	10	10	10	10
Conversion time	—	—	13.4	—	—	6.7
Analog input capacitance	—	—	20	—	—	20
Permissible signal-source impedance	—	—	10*1	—	—	10*
	—	—	5*2	—	—	5*4
Nonlinearity error	—	—	±7.5	—	—	±3.5
Offset error	—	—	±7.5	—	—	±3.5
Full-scale error	—	—	±7.5	—	—	±3.5
Quantization	—	—	±0.5	—	—	±0.5
Absolute accuracy	—	—	±8.0	—	—	±4.0

Notes: 1. 4.0 V ≤ AV_{CC} ≤ 5.5 V
2. 2.7 V ≤ AV_{CC} < 4.0 V
3. ϕ ≤ 12 MHz
4. ϕ > 12 MHz

Condition B: V_{CC} = AV_{CC} = 5.0 V ± 10%, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ϕ = 2 to 20 MHz, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)

	Condition A			Condition B				
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**	**Test C**
Resolution	8	8	8	8	8	8	bit	
Conversion time	—	—	10	—	—	10	µs	20-pF load
Absolute accuracy	—	±2.0	±3.0	—	±1.0	±1.5	LSB	2-MΩ load
	—	—	±2.0	—	—	±1.0	LSB	4-MΩ load

System Clock Timing: Figure 20.7 shows the system clock timing.

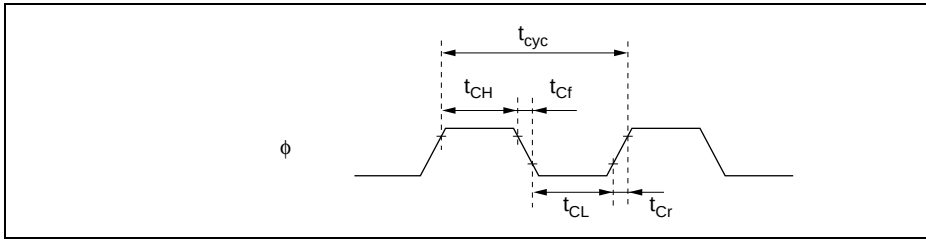


Figure 20.7 System Clock Timing

Oscillator Settling Timing: Figure 20.8 shows the oscillator settling timing.

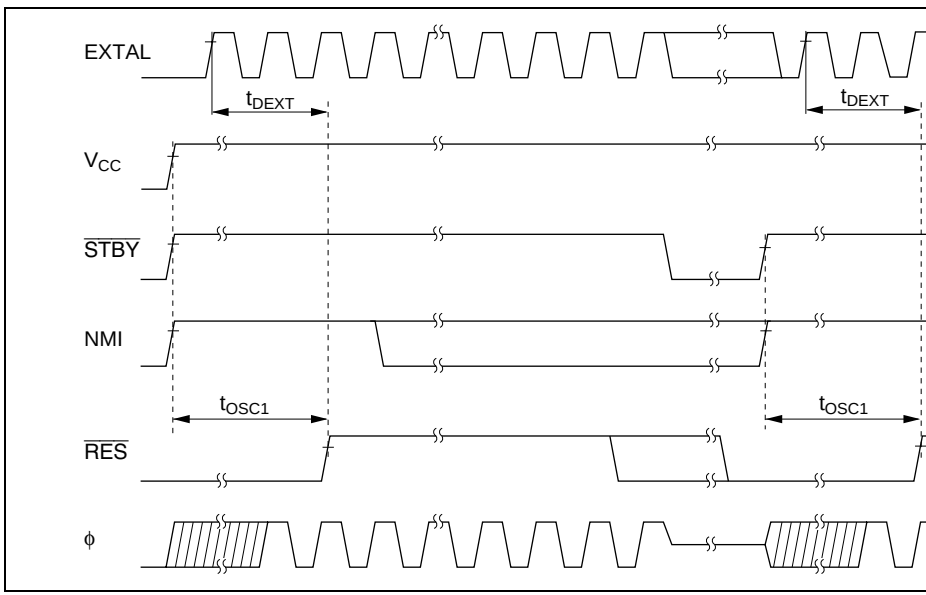


Figure 20.8 Oscillator Settling Timing

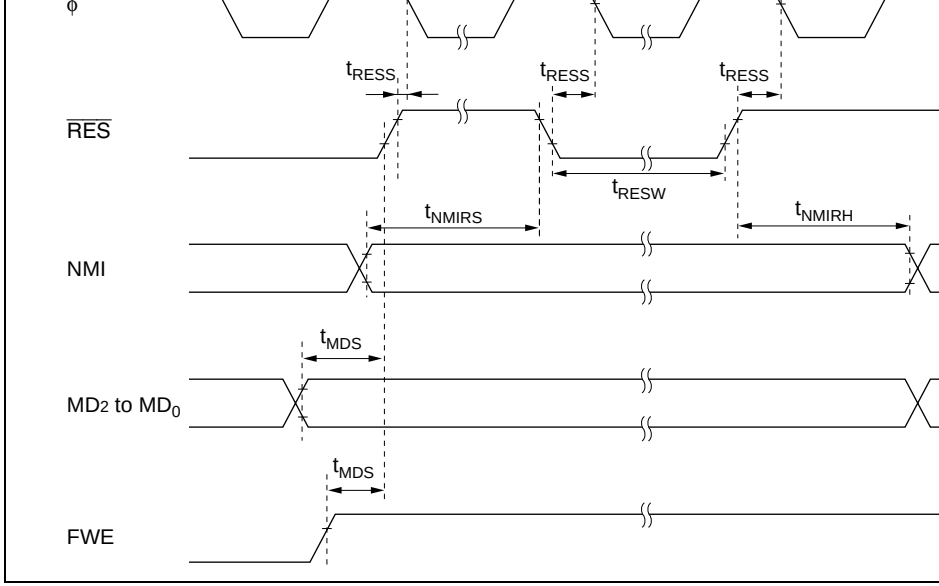


Figure 20.9 Reset Input Timing

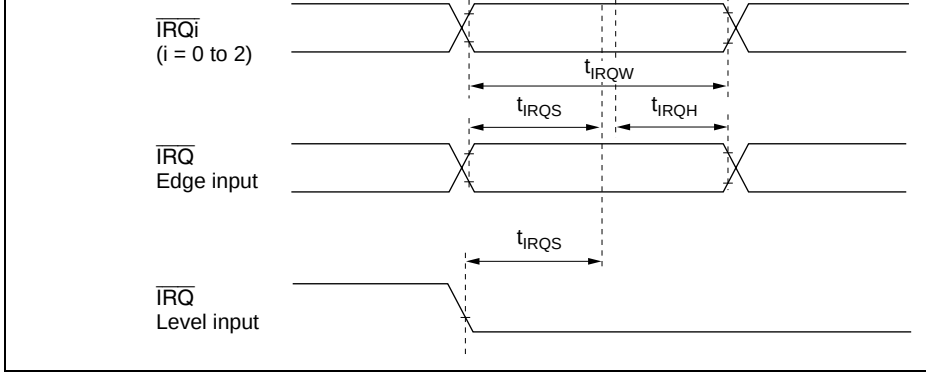


Figure 20.10 Interrupt Input Timing

20.3.3 Bus Timing

The bus timing is shown below.

Basic Bus Timing (Two-State Access): Figure 20.11 shows the basic bus timing for state access.

Basic Bus Timing (Three-State Access): Figure 20.12 shows the basic bus timing fo three-state access.

Basic Bus Timing (Three-State Access with One Wait State): Figure 20.13 shows t timing for external three-state access with one wait state.

Burst ROM Access Timing (Two-State Access): Figure 20.14 shows the burst ROM timing for two-state access.

Burst ROM Access Timing (One-State Access): Figure 20.15 shows the burst ROM timing for one-state access.

External Bus Release Timing: Figure 20.16 shows the external bus release timing.

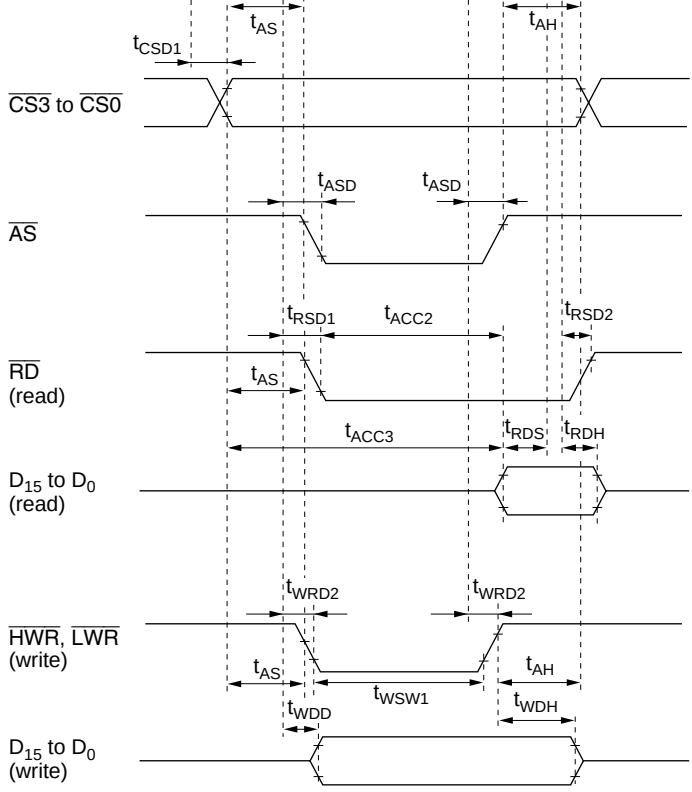


Figure 20.11 Basic Bus Timing (Two-State Access)

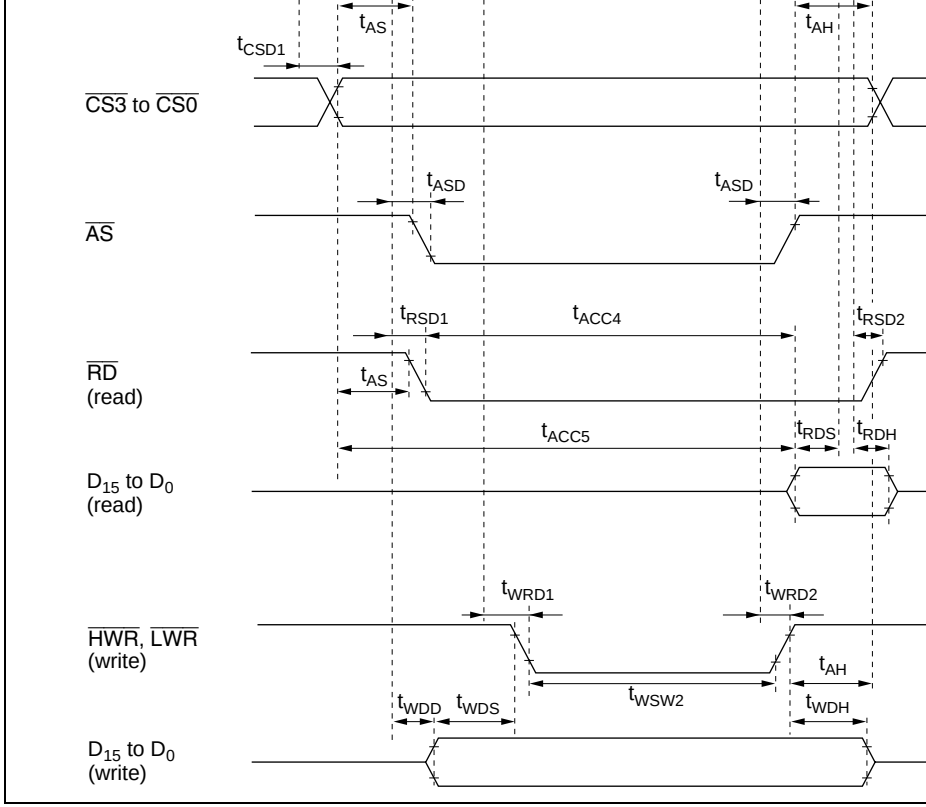


Figure 20.12 Basic Bus Timing (Three-State Access)

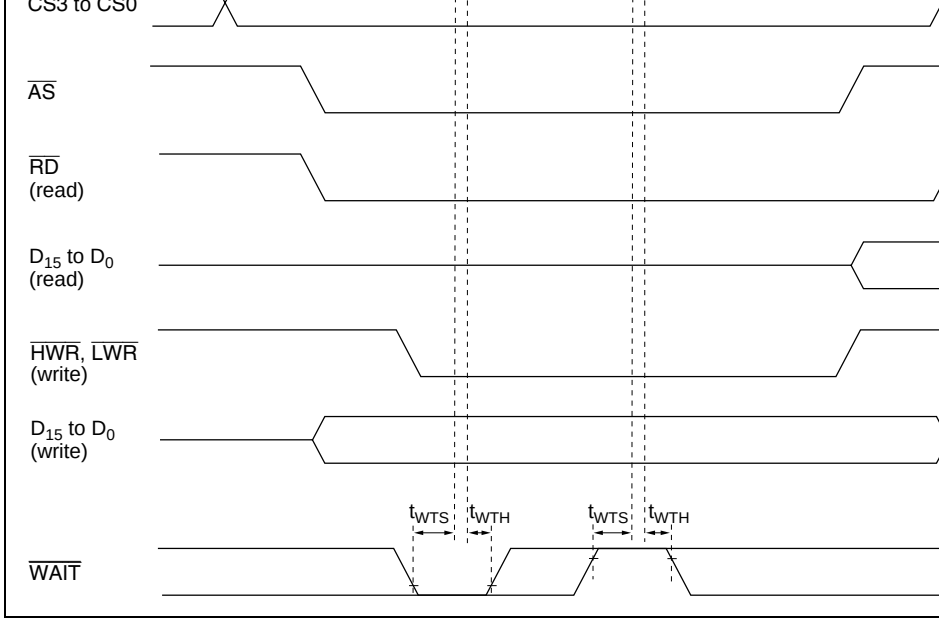


Figure 20.13 Basic Bus Timing (Three-State Access with One Wait Stat

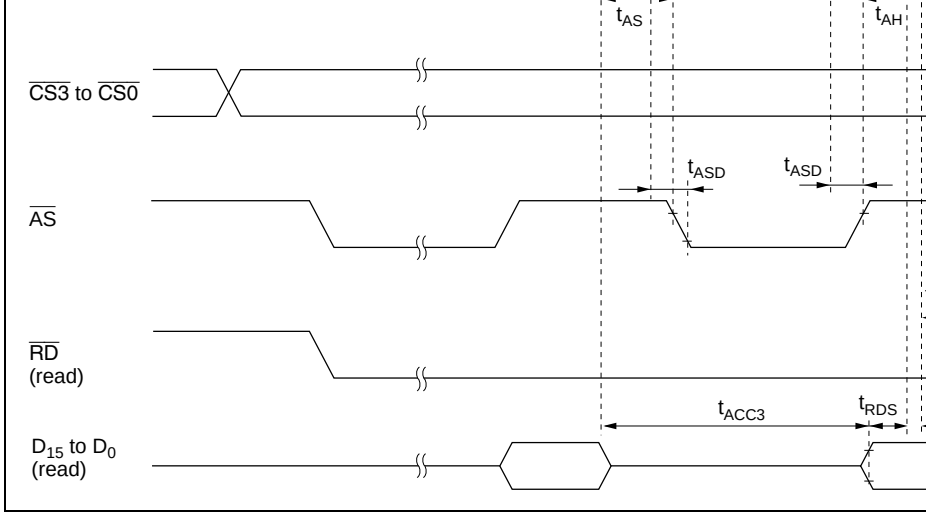


Figure 20.14 Burst ROM Access Timing (Two-State Access)

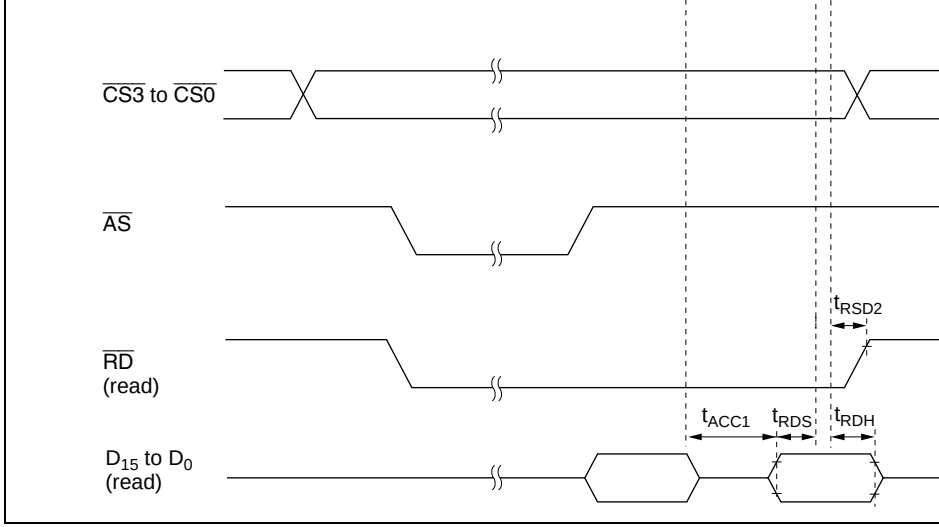


Figure 20.15 Burst ROM Access Timing (One-State Access)

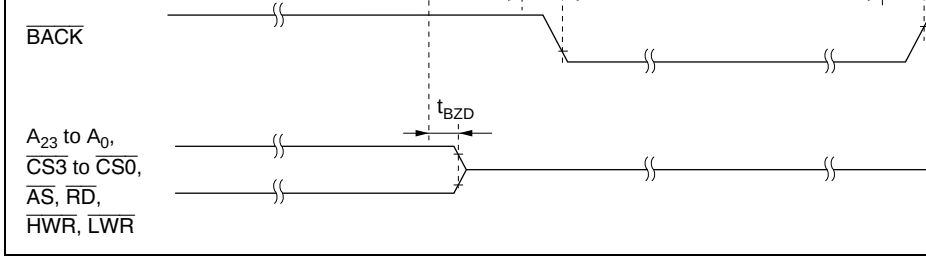


Figure 20.16 External Bus Release Timing

20.3.4 Timing for On-Chip Supporting Modules

Figure 20.17 to figure 20.26 show the timings for on-chip peripheral modules.

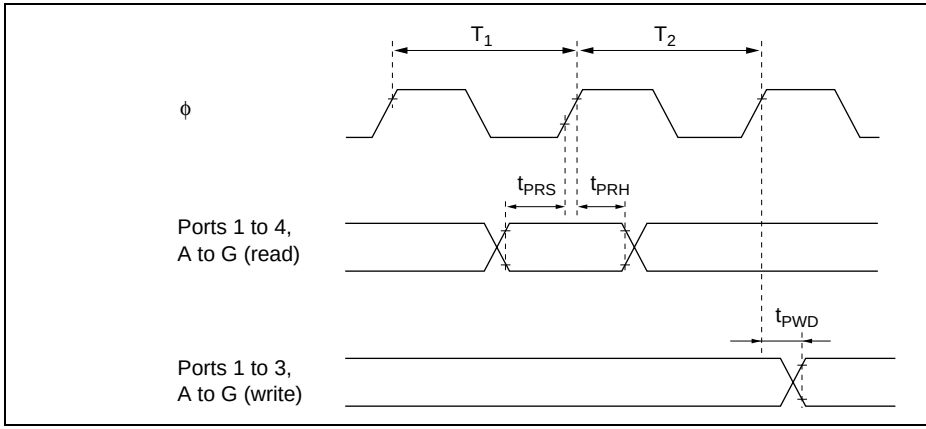
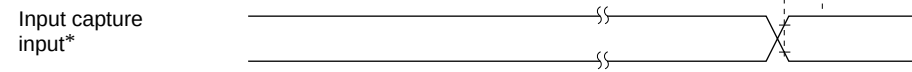


Figure 20.17 I/O Port Input/Output Timing



Note: * TIOCA0 to TIOCA5, TIOCB0 to TIOCB5, TIOCC0, TIOCC3, TIOCD0, TIOCD3

Figure 20.18 TPU Input/Output Timing

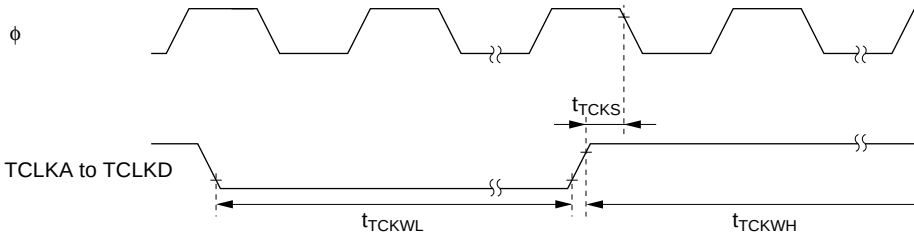


Figure 20.19 TPU Clock Input Timing

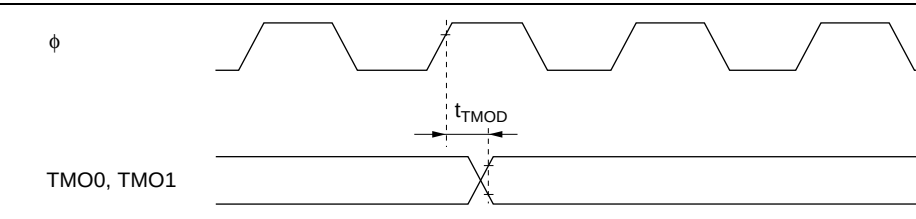


Figure 20.20 8-Bit Timer Output Timing

Figure 20.21 8-Bit Timer Clock Input Timing

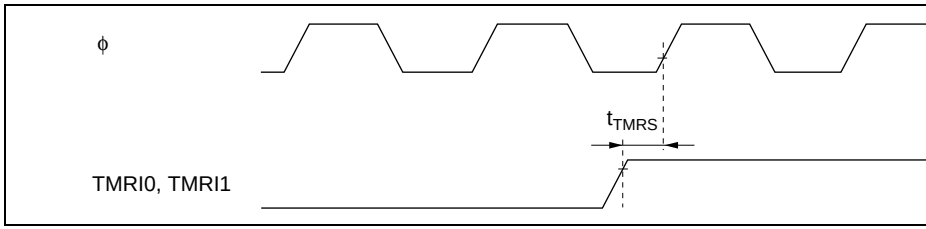


Figure 20.22 8-Bit Timer Reset Input Timing

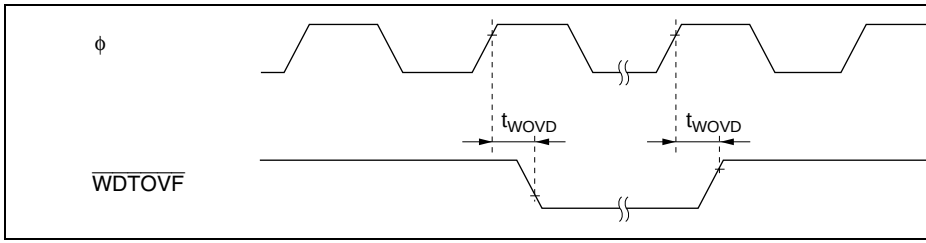


Figure 20.23 WDT Output Timing
(ZTAT Version, Mask ROM Version, and ROMless Version Only)

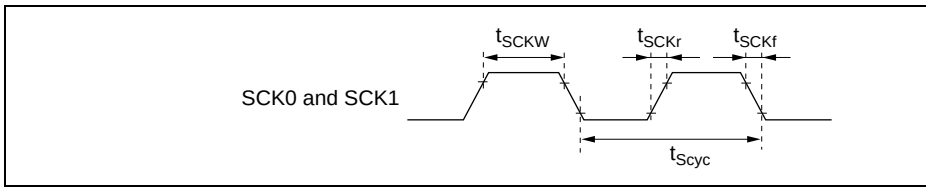


Figure 20.24 SCK Clock Input Timing

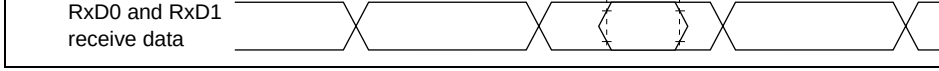


Figure 20.25 SCI Input/Output Timing (Clock Synchronous Mode)

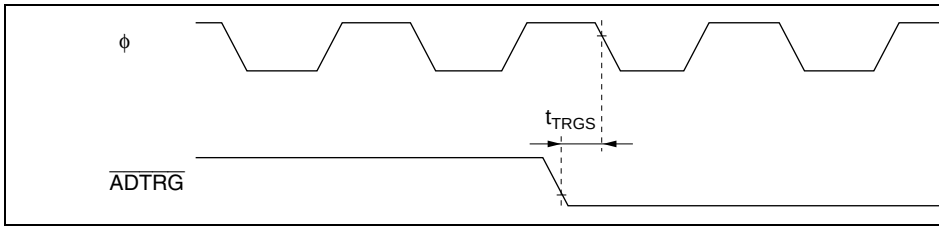


Figure 20.26 A/D Converter External Trigger Input Timing

20.4 Usage Note

Although the F-ZTAT, ZTAT, mask ROM, and ROMless versions fully meet the electr
specifications listed in this manual, due to differences in the fabrication process, the on
ROM, and the layout patterns, there will be differences in the actual values of the electr
characteristics, the operating margins, the noise margins, and other aspects.

Therefore, if a system is estimated using the F-ZTAT or ZTAT version, a similar evalu
should also be performed using the mask ROM version.

Rs	General register (source)
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 ri 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 r
to R7, E0 to E7), and 32-bit registers (ER0 to ER7).
2. The MAC register cannot be used in the H8S/2345 Group.

(1) Data Transfer Instructions

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code			
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z
MOV	MOV.B #xx:8,Rd	B	2							#xx:8→Rd8	—	—	↕	↕
	MOV.B Rs,Rd	B		2						Rs8→Rd8	—	—	↕	↕
	MOV.B @ERs,Rd	B			2					@ERs→Rd8	—	—	↕	↕
	MOV.B @(d:16,ERs),Rd	B				4				@(d:16,ERs)→Rd8	—	—	↕	↕
	MOV.B @(d:32,ERs),Rd	B				8				@(d:32,ERs)→Rd8	—	—	↕	↕
	MOV.B @ERs+,Rd	B					2			@ERs→Rd8,ERs32+1→ERs32	—	—	↕	↕
	MOV.B @aa:8,Rd	B						2		@aa:8→Rd8	—	—	↕	↕
	MOV.B @aa:16,Rd	B						4		@aa:16→Rd8	—	—	↕	↕
	MOV.B @aa:32,Rd	B						6		@aa:32→Rd8	—	—	↕	↕
	MOV.B Rs,@ERd	B			2					Rs8→@ERd	—	—	↕	↕
	MOV.B Rs,@(d:16,ERd)	B				4				Rs8→@(d:16,ERd)	—	—	↕	↕
	MOV.B Rs,@(d:32,ERd)	B				8				Rs8→@(d:32,ERd)	—	—	↕	↕
	MOV.B Rs,@-ERd	B					2			ERd32-1→ERd32,Rs8→@ERd	—	—	↕	↕
	MOV.B Rs,@aa:8	B						2		Rs8→@aa:8	—	—	↕	↕
	MOV.B Rs,@aa:16	B						4		Rs8→@aa:16	—	—	↕	↕
	MOV.B Rs,@aa:32	B						6		Rs8→@aa:32	—	—	↕	↕
	MOV.W #xx:16,Rd	W	4							#xx:16→Rd16	—	—	↕	↕
	MOV.W Rs,Rd	W		2						Rs16→Rd16	—	—	↕	↕

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)							Operation	Condition Code	
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		
MOV	MOV.W @(d:16,ERs),Rd	W			4					@(d:16,ERs)→Rd16	— — ↕ ↕ 0 —
	MOV.W @(d:32,ERs),Rd	W			8					@(d:32,ERs)→Rd16	— — ↕ ↕ 0 —
	MOV.W @ERs+,Rd	W				2				@ERs→Rd16,ERs32+2→ERs32	— — ↕ ↕ 0 —
	MOV.W @aa:16,Rd	W					4			@aa:16→Rd16	— — ↕ ↕ 0 —
	MOV.W @aa:32,Rd	W					6			@aa:32→Rd16	— — ↕ ↕ 0 —
	MOV.W Rs,@ERd	W		2						Rs16→@ERd	— — ↕ ↕ 0 —
	MOV.W Rs,@(d:16,ERd)	W			4					Rs16→@(d:16,ERd)	— — ↕ ↕ 0 —
	MOV.W Rs,@(d:32,ERd)	W			8					Rs16→@(d:32,ERd)	— — ↕ ↕ 0 —
	MOV.W Rs,@-ERd	W				2				ERd32-2→ERd32,Rs16→@ERd	— — ↕ ↕ 0 —
	MOV.W Rs,@aa:16	W					4			Rs16→@aa:16	— — ↕ ↕ 0 —
	MOV.W Rs,@aa:32	W					6			Rs16→@aa:32	— — ↕ ↕ 0 —
	MOV.L #xx:32,ERd	L	6							#xx:32→ERd32	— — ↕ ↕ 0 —
	MOV.L ERs,ERd	L		2						ERs32→ERd32	— — ↕ ↕ 0 —
	MOV.L @ERs,ERd	L			4					@ERs→ERd32	— — ↕ ↕ 0 —
	MOV.L @(d:16,ERs),ERd	L				6				@(d:16,ERs)→ERd32	— — ↕ ↕ 0 —
	MOV.L @(d:32,ERs),ERd	L				10				@(d:32,ERs)→ERd32	— — ↕ ↕ 0 —
	MOV.L @ERs+,ERd	L					4			@ERs→ERd32,ERs32+4→@ERs32	— — ↕ ↕ 0 —
	MOV.L @aa:16,ERd	L						6		@aa:16→ERd32	— — ↕ ↕ 0 —
	MOV.L @aa:32,ERd	L						8		@aa:32→ERd32	— — ↕ ↕ 0 —

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H
MOV	MOV.L ERs,@ERd	L			4						ERs32→@ERd	—	—
	MOV.L ERs,@(d:16,ERd)	L				6					ERs32→@(d:16,ERd)	—	—
	MOV.L ERs,@(d:32,ERd)	L				10					ERs32→@(d:32,ERd)	—	—
	MOV.L ERs,@-ERd	L					4				ERd32–4→ERd32,ERs32→@ERd	—	—
	MOV.L ERs,@aa:16	L						6			ERs32→@aa:16	—	—
	MOV.L ERs,@aa:32	L						8			ERs32→@aa:32	—	—
POP	POP.W Rn	W								2	@SP→Rn16,SP+2→SP	—	—
	POP.L ERn	L								4	@SP→ERn32,SP+4→SP	—	—
PUSH	PUSH.W Rn	W								2	SP-2→SP,Rn16→@SP	—	—
	PUSH.L ERn	L								4	SP-4→SP,ERn32→@SP	—	—
LDM	LDM @SP+,(ERm-ERn)	L								4	(@SP→ERn32,SP+4→SP) Repeated for each register restored	—	—
STM	STM (ERm-ERn),@-SP	L								4	(SP-4→SP, ERn32→@SP) Repeated for each register saved	—	—
MOVFPE	MOVFPE @aa:16,Rd	Cannot be used in the H8S/2345 Group											
MOVTPE	MOVTPE Rs,@aa:16	Cannot be used in the H8S/2345 Group											

(2) Arithmetic Instructions

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code					
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C
ADD	ADD.B #xx:8,Rd	B 2									Rd8+#xx:8→Rd8	—	↕	↕	↕	↕	↕
	ADD.B Rs,Rd	B	2								Rd8+Rs8→Rd8	—	↕	↕	↕	↕	↕
	ADD.W #xx:16,Rd	W 4									Rd16+#xx:16→Rd16	—	[3]	↕	↕	↕	↕
	ADD.W Rs,Rd	W	2								Rd16+Rs16→Rd16	—	[3]	↕	↕	↕	↕
	ADD.L #xx:32,ERd	L 6									ERd32+#xx:32→ERd32	—	[4]	↕	↕	↕	↕
	ADD.L ERs,ERd	L	2								ERd32+ERs32→ERd32	—	[4]	↕	↕	↕	↕
ADDX	ADDX #xx:8,Rd	B 2									Rd8+#xx:8+C→Rd8	—	↕	↕	↕	↕	↕
	ADDX Rs,Rd	B	2								Rd8+Rs8+C→Rd8	—	↕	↕	↕	↕	↕
ADDS	ADDS #1,ERd	L	2								ERd32+1→ERd32	—	—	—	—	—	—
	ADDS #2,ERd	L	2								ERd32+2→ERd32	—	—	—	—	—	—
	ADDS #4,ERd	L	2								ERd32+4→ERd32	—	—	—	—	—	—
INC	INC.B Rd	B	2								Rd8+1→Rd8	—	—	↕	↕	↕	—
	INC.W #1,Rd	W	2								Rd16+1→Rd16	—	—	↕	↕	↕	—
	INC.W #2,Rd	W	2								Rd16+2→Rd16	—	—	↕	↕	↕	—
	INC.L #1,ERd	L	2								ERd32+1→ERd32	—	—	↕	↕	↕	—
	INC.L #2,ERd	L	2								ERd32+2→ERd32	—	—	↕	↕	↕	—
DAA	DAA Rd	B	2								Rd8 decimal adjust→Rd8	—	*	↕	↕	*	↕
SUB	SUB.B Rs,Rd	B	2								Rd8-Rs8→Rd8	—	↕	↕	↕	↕	↕
	SUB.W #xx:16,Rd	W 4									Rd16-#xx:16→Rd16	—	[3]	↕	↕	↕	↕

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H
SUB	SUB.W Rs,Rd	W	2								Rd16-Rs16→Rd16	—	[3]
	SUB.L #xx:32,ERd	L	6								ERd32-#xx:32→ERd32	—	[4]
	SUB.L ERs,ERd	L		2							ERd32-ERs32→ERd32	—	[4]
SUBX	SUBX #xx:8,Rd	B	2								Rd8-#xx:8-C→Rd8	—	[1]
	SUBX Rs,Rd	B		2							Rd8-Rs8-C→Rd8	—	[1]
SUBS	SUBS #1,ERd	L		2							ERd32-1→ERd32	—	—
	SUBS #2,ERd	L		2							ERd32-2→ERd32	—	—
	SUBS #4,ERd	L		2							ERd32-4→ERd32	—	—
DEC	DEC.B Rd	B		2							Rd8-1→Rd8	—	—
	DEC.W #1,Rd	W		2							Rd16-1→Rd16	—	—
	DEC.W #2,Rd	W		2							Rd16-2→Rd16	—	—
	DEC.L #1,ERd	L		2							ERd32-1→ERd32	—	—
	DEC.L #2,ERd	L		2							ERd32-2→ERd32	—	—
DAS	DAS Rd	B		2							Rd8 decimal adjust→Rd8	—	*
MULXU	MULXU.B Rs,Rd	B		2							Rd8×Rs8→Rd16 (unsigned multiplication)	—	—
	MULXU.W Rs,ERd	W		2							Rd16×Rs16→ERd32 (unsigned multiplication)	—	—
MULXS	MULXS.B Rs,Rd	B		4							Rd8×Rs8→Rd16 (signed multiplication)	—	—
	MULXS.W Rs,ERd	W		4							Rd16×Rs16→ERd32 (signed multiplication)	—	—

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code					
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C
DIVXU	DIVXU.B Rs,Rd	B 2								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	↕	↕	—	—
	DIVXU.W Rs,ERd	W 2								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	↕	↕	—	—
DIVXS	divxs.B Rs,Rd	B 4								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	↕	↕	—	—
	DIVXS.W Rs,ERd	W 4								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	↕	↕	—	—
CMP	CMP.B #xx:8,Rd	B 2								Rd8-#xx:8	—	↕	↕	↕	↕	↕
	CMP.B Rs,Rd	B	2							Rd8-Rs8	—	↕	↕	↕	↕	↕
	CMP.W #xx:16,Rd	W 4								Rd16-#xx:16	—	[3]	↕	↕	↕	↕
	CMP.W Rs,Rd	W	2							Rd16-Rs16	—	[3]	↕	↕	↕	↕
	CMP.L #xx:32,ERd	L 6								ERd32-#xx:32	—	[4]	↕	↕	↕	↕
	CMP.L ERs,ERd	L	2							ERd32-ERs32	—	[4]	↕	↕	↕	↕
NEG	NEG.B Rd	B	2							0-Rd8→Rd8	—	↕	↕	↕	↕	↕
	NEG.W Rd	W	2							0-Rd16→Rd16	—	↕	↕	↕	↕	↕
	NEG.L ERd	L	2							0-ERd32→ERd32	—	↕	↕	↕	↕	↕
EXTU	EXTU.W Rd	W	2							0→(<bit 15 to 8> of Rd16)	—	—	0	↕	0	—
	EXTU.L ERd	L	2							0→(<bit 31 to 16> of ERd32)	—	—	0	↕	0	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16)→ (<bit 15 to 8> of Rd16)	—	—
	EXTS.L ERd	L		2								(<bit 15> of ERd32)→ (<bit 31 to 16> of ERd32)	—	—
TAS	TAS @ERd	B			4							@ERd-0→CCR set, (1)→ (<bit 7> of @ERd)	—	—
MAC	MAC @ERn+, @ERm+	Cannot be used in the H8S/2345 Group												
CLRMAC	CLRMAC													
LDMAC	LDMAC ERs,MACH													
	LDMAC ERs,MACL													
STMAC	STMAC MACH,ERd													
	STMAC MACL,ERd													

RENESAS

(3) Logical Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H
AND	AND.B #xx:8,Rd	B	2									Rd8∧#xx:8→Rd8	—	—
	AND.B Rs,Rd	B		2								Rd8∧Rs8→Rd8	—	—
	AND.W #xx:16,Rd	W	4									Rd16∧#xx:16→Rd16	—	—
	AND.W Rs,Rd	W		2								Rd16∧Rs16→Rd16	—	—
	AND.L #xx:32,ERd	L	6									ERd32∧#xx:32→ERd32	—	—
	AND.L ERs,ERd	L		4								ERd32∧ERs32→ERd32	—	—
OR	OR.B #xx:8,Rd	B	2									Rd8∨#xx:8→Rd8	—	—
	OR.B Rs,Rd	B		2								Rd8∨Rs8→Rd8	—	—
	OR.W #xx:16,Rd	W	4									Rd16∨#xx:16→Rd16	—	—
	OR.W Rs,Rd	W		2								Rd16∨Rs16→Rd16	—	—
	OR.L #xx:32,ERd	L	6									ERd32∨#xx:32→ERd32	—	—
	OR.L ERs,ERd	L		4								ERd32∨ERs32→ERd32	—	—
XOR	XOR.B #xx:8,Rd	B	2									Rd8⊕#xx:8→Rd8	—	—
	XOR.B Rs,Rd	B		2								Rd8⊕Rs8→Rd8	—	—
	XOR.W #xx:16,Rd	W	4									Rd16⊕#xx:16→Rd16	—	—
	XOR.W Rs,Rd	W		2								Rd16⊕Rs16→Rd16	—	—
	XOR.L #xx:32,ERd	L	6									ERd32⊕#xx:32→ERd32	—	—
	XOR.L ERs,ERd	L		4								ERd32⊕ERs32→ERd32	—	—
NOT	NOT.B Rd	B		2								¬ Rd8→Rd8	—	—

(4) Shift Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code					
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C
SHAL	SHAL.B Rd	B		2									—	—	↕	↕	↕	↕
	SHAL.B #2,Rd	B		2									—	—	↕	↕	↕	↕
	SHAL.W Rd	W		2									—	—	↕	↕	↕	↕
	SHAL.W #2,Rd	W		2									—	—	↕	↕	↕	↕
	SHAL.L ERd	L		2									—	—	↕	↕	↕	↕
	SHAL.L #2,ERd	L		2									—	—	↕	↕	↕	↕
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	↕	↕
	SHAR.B #2,Rd	B		2									—	—	↕	↕	↕	↕
	SHAR.W Rd	W		2									—	—	↕	↕	↕	↕
	SHAR.W #2,Rd	W		2									—	—	↕	↕	↕	↕
	SHAR.L ERd	L		2									—	—	↕	↕	↕	↕
	SHAR.L #2,ERd	L		2									—	—	↕	↕	↕	↕
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	↕	↕
	SHLL.B #2,Rd	B		2									—	—	↕	↕	↕	↕
	SHLL.W Rd	W		2									—	—	↕	↕	↕	↕
	SHLL.W #2,Rd	W		2									—	—	↕	↕	↕	↕
	SHLL.L ERd	L		2									—	—	↕	↕	↕	↕
	SHLL.L #2,ERd	L		2									—	—	↕	↕	↕	↕

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code					
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—			I	H	N	Z	V	C
SHLR	SHLR.B Rd	B	2									0 → MSB → LSB → C	—	—	0	↕	0	↕
	SHLR.B #2,Rd	B	2										—	—	0	↕	0	↕
	SHLR.W Rd	W	2										—	—	0	↕	0	↕
	SHLR.W #2,Rd	W	2										—	—	0	↕	0	↕
	SHLR.L ERd	L	2										—	—	0	↕	0	↕
	SHLR.L #2,ERd	L	2										—	—	0	↕	0	↕
ROTXL	ROTXL.B Rd	B	2									C ← MSB ← LSB	—	—	↕	↕	0	↕
	ROTXL.B #2,Rd	B	2										—	—	↕	↕	0	↕
	ROTXL.W Rd	W	2										—	—	↕	↕	0	↕
	ROTXL.W #2,Rd	W	2										—	—	↕	↕	0	↕
	ROTXL.L ERd	L	2										—	—	↕	↕	0	↕
	ROTXL.L #2,ERd	L	2										—	—	↕	↕	0	↕
ROTXR	ROTXR.B Rd	B	2									MSB → LSB → C	—	—	↕	↕	0	↕
	ROTXR.B #2,Rd	B	2										—	—	↕	↕	0	↕
	ROTXR.W Rd	W	2										—	—	↕	↕	0	↕
	ROTXR.W #2,Rd	W	2										—	—	↕	↕	0	↕
	ROTXR.L ERd	L	2										—	—	↕	↕	0	↕
	ROTXR.L #2,ERd	L	2										—	—	↕	↕	0	↕

Mnemonic	Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code					
		#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—			I	H	U	N	Z	V
ROTL	ROTL.B Rd	B	2										—	—	—	↕	↕	0
	ROTL.B #2,Rd	B	2										—	—	—	↕	↕	0
	ROTL.W Rd	W	2										—	—	—	↕	↕	0
	ROTL.W #2,Rd	W	2										—	—	—	↕	↕	0
	ROTL.L ERd	L	2										—	—	—	↕	↕	0
	ROTL.L #2,ERd	L	2										—	—	—	↕	↕	0
ROTR	ROTR.B Rd	B	2										—	—	—	↕	↕	0
	ROTR.B #2,Rd	B	2										—	—	—	↕	↕	0
	ROTR.W Rd	W	2										—	—	—	↕	↕	0
	ROTR.W #2,Rd	W	2										—	—	—	↕	↕	0
	ROTR.L ERd	L	2										—	—	—	↕	↕	0
	ROTR.L #2,ERd	L	2										—	—	—	↕	↕	0

RENESAS

(5) Bit-Manipulation Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H
BSET	BSET #xx:3,Rd	B	2								(#xx:3 of Rd8)←1	—	—
	BSET #xx:3,@ERd	B		4							(#xx:3 of @ERd)←1	—	—
	BSET #xx:3,@aa:8	B						4			(#xx:3 of @aa:8)←1	—	—
	BSET #xx:3,@aa:16	B						6			(#xx:3 of @aa:16)←1	—	—
	BSET #xx:3,@aa:32	B						8			(#xx:3 of @aa:32)←1	—	—
	BSET Rn,Rd	B		2							(Rn8 of Rd8)←1	—	—
	BSET Rn,@ERd	B			4						(Rn8 of @ERd)←1	—	—
	BSET Rn,@aa:8	B						4			(Rn8 of @aa:8)←1	—	—
	BSET Rn,@aa:16	B						6			(Rn8 of @aa:16)←1	—	—
	BSET Rn,@aa:32	B						8			(Rn8 of @aa:32)←1	—	—
BCLR	BCLR #xx:3,Rd	B	2								(#xx:3 of Rd8)←0	—	—
	BCLR #xx:3,@ERd	B		4							(#xx:3 of @ERd)←0	—	—
	BCLR #xx:3,@aa:8	B						4			(#xx:3 of @aa:8)←0	—	—
	BCLR #xx:3,@aa:16	B						6			(#xx:3 of @aa:16)←0	—	—
	BCLR #xx:3,@aa:32	B						8			(#xx:3 of @aa:32)←0	—	—
	BCLR Rn,Rd	B	2								(Rn8 of Rd8)←0	—	—
	BCLR Rn,@ERd	B		4							(Rn8 of @ERd)←0	—	—
	BCLR Rn,@aa:8	B						4			(Rn8 of @aa:8)←0	—	—
	BCLR Rn,@aa:16	B						6			(Rn8 of @aa:16)←0	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code							
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	U	N	Z	V	C
BCLR	BCLR Rn,@aa:32	B						8			(Rn8 of @aa:32)←0	—	—	—	—	—	—	—
BNOT	BNOT #xx:3,Rd	B		2							(#xx:3 of Rd8)←[¬ (#xx:3 of Rd8)]	—	—	—	—	—	—	—
	BNOT #xx:3,@ERd	B			4						(#xx:3 of @ERd)←[¬ (#xx:3 of @ERd)]	—	—	—	—	—	—	—
	BNOT #xx:3,@aa:8	B						4			(#xx:3 of @aa:8)←[¬ (#xx:3 of @aa:8)]	—	—	—	—	—	—	—
	BNOT #xx:3,@aa:16	B						6			(#xx:3 of @aa:16)←[¬ (#xx:3 of @aa:16)]	—	—	—	—	—	—	—
	BNOT #xx:3,@aa:32	B						8			(#xx:3 of @aa:32)←[¬ (#xx:3 of @aa:32)]	—	—	—	—	—	—	—
	BNOT Rn,Rd	B		2							(Rn8 of Rd8)←[¬ (Rn8 of Rd8)]	—	—	—	—	—	—	—
	BNOT Rn,@ERd	B			4						(Rn8 of @ERd)←[¬ (Rn8 of @ERd)]	—	—	—	—	—	—	—
	BNOT Rn,@aa:8	B						4			(Rn8 of @aa:8)←[¬ (Rn8 of @aa:8)]	—	—	—	—	—	—	—
	BNOT Rn,@aa:16	B						6			(Rn8 of @aa:16)←[¬ (Rn8 of @aa:16)]	—	—	—	—	—	—	—
	BNOT Rn,@aa:32	B						8			(Rn8 of @aa:32)←[¬ (Rn8 of @aa:32)]	—	—	—	—	—	—	—
BTST	BTST #xx:3,Rd	B		2							¬ (#xx:3 of Rd8)→Z	—	—	—	—	↕	—	—
	BTST #xx:3,@ERd	B			4						¬ (#xx:3 of @ERd)→Z	—	—	—	—	↕	—	—
	BTST #xx:3,@aa:8	B						4			¬ (#xx:3 of @aa:8)→Z	—	—	—	—	↕	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H
BTST	BTST #xx:3,@aa:32	B					8				¬ (#xx:3 of @aa:32)→Z	—	—
	BTST Rn,Rd	B	2								¬ (Rn8 of Rd8)→Z	—	—
	BTST Rn,@ERd	B		4							¬ (Rn8 of @ERd)→Z	—	—
	BTST Rn,@aa:8	B					4				¬ (Rn8 of @aa:8)→Z	—	—
	BTST Rn,@aa:16	B					6				¬ (Rn8 of @aa:16)→Z	—	—
	BTST Rn,@aa:32	B					8				¬ (Rn8 of @aa:32)→Z	—	—
BLD	BLD #xx:3,Rd	B	2								(#xx:3 of Rd8)→C	—	—
	BLD #xx:3,@ERd	B		4							(#xx:3 of @ERd)→C	—	—
	BLD #xx:3,@aa:8	B					4				(#xx:3 of @aa:8)→C	—	—
	BLD #xx:3,@aa:16	B					6				(#xx:3 of @aa:16)→C	—	—
	BLD #xx:3,@aa:32	B					8				(#xx:3 of @aa:32)→C	—	—
BILD	BILD #xx:3,Rd	B	2								¬ (#xx:3 of Rd8)→C	—	—
	BILD #xx:3,@ERd	B		4							¬ (#xx:3 of @ERd)→C	—	—
	BILD #xx:3,@aa:8	B					4				¬ (#xx:3 of @aa:8)→C	—	—
	BILD #xx:3,@aa:16	B					6				¬ (#xx:3 of @aa:16)→C	—	—
	BILD #xx:3,@aa:32	B					8				¬ (#xx:3 of @aa:32)→C	—	—
BST	BST #xx:3,Rd	B	2								C→(#xx:3 of Rd8)	—	—
	BST #xx:3,@ERd	B		4							C→(#xx:3 of @ERd)	—	—
	BST #xx:3,@aa:8	B					4				C→(#xx:3 of @aa:8)	—	—

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H
BST	BST #xx:3,@aa:16	B						6			C→(#xx:3 of @aa:16)	—	—
	BST #xx:3,@aa:32	B						8			C→(#xx:3 of @aa:32)	—	—
BIST	BIST #xx:3,Rd	B		2							¬ C→(#xx:3 of Rd8)	—	—
	BIST #xx:3,@ERd	B			4						¬ C→(#xx:3 of @ERd)	—	—
	BIST #xx:3,@aa:8	B						4			¬ C→(#xx:3 of @aa:8)	—	—
	BIST #xx:3,@aa:16	B						6			¬ C→(#xx:3 of @aa:16)	—	—
	BIST #xx:3,@aa:32	B						8			¬ C→(#xx:3 of @aa:32)	—	—
BAND	BAND #xx:3,Rd	B		2							C∧(#xx:3 of Rd8)→C	—	—
	BAND #xx:3,@ERd	B			4						C∧(#xx:3 of @ERd)→C	—	—
	BAND #xx:3,@aa:8	B						4			C∧(#xx:3 of @aa:8)→C	—	—
	BAND #xx:3,@aa:16	B						6			C∧(#xx:3 of @aa:16)→C	—	—
	BAND #xx:3,@aa:32	B						8			C∧(#xx:3 of @aa:32)→C	—	—
BIAND	BIAND #xx:3,Rd	B		2							C∧[¬ (#xx:3 of Rd8)]→C	—	—
	BIAND #xx:3,@ERd	B			4						C∧[¬ (#xx:3 of @ERd)]→C	—	—
	BIAND #xx:3,@aa:8	B						4			C∧[¬ (#xx:3 of @aa:8)]→C	—	—
	BIAND #xx:3,@aa:16	B						6			C∧[¬ (#xx:3 of @aa:16)]→C	—	—
	BIAND #xx:3,@aa:32	B						8			C∧[¬ (#xx:3 of @aa:32)]→C	—	—
BOR	BOR #xx:3,Rd	B		2							C∨(#xx:3 of Rd8)→C	—	—
	BOR #xx:3,@ERd	B			4						C∨(#xx:3 of @ERd)→C	—	—

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H
BOR	BOR #xx:3,@aa:8	B					4				C∨(#xx:3 of @aa:8)→C	—	—
	BOR #xx:3,@aa:16	B						6			C∨(#xx:3 of @aa:16)→C	—	—
	BOR #xx:3,@aa:32	B						8			C∨(#xx:3 of @aa:32)→C	—	—
BIOR	BIOR #xx:3,Rd	B		2							C∨[¬ (#xx:3 of Rd8)]→C	—	—
	BIOR #xx:3,@ERd	B			4						C∨[¬ (#xx:3 of @ERd)]→C	—	—
	BIOR #xx:3,@aa:8	B						4			C∨[¬ (#xx:3 of @aa:8)]→C	—	—
	BIOR #xx:3,@aa:16	B						6			C∨[¬ (#xx:3 of @aa:16)]→C	—	—
	BIOR #xx:3,@aa:32	B						8			C∨[¬ (#xx:3 of @aa:32)]→C	—	—
BXOR	BXOR #xx:3,Rd	B		2							C⊕(#xx:3 of Rd8)→C	—	—
	BXOR #xx:3,@ERd	B			4						C⊕(#xx:3 of @ERd)→C	—	—
	BXOR #xx:3,@aa:8	B						4			C⊕(#xx:3 of @aa:8)→C	—	—
	BXOR #xx:3,@aa:16	B						6			C⊕(#xx:3 of @aa:16)→C	—	—
	BXOR #xx:3,@aa:32	B						8			C⊕(#xx:3 of @aa:32)→C	—	—
BIXOR	BIXOR #xx:3,Rd	B		2							C⊕[¬ (#xx:3 of Rd8)]→C	—	—
	BIXOR #xx:3,@ERd	B			4						C⊕[¬ (#xx:3 of @ERd)]→C	—	—
	BIXOR #xx:3,@aa:8	B						4			C⊕[¬ (#xx:3 of @aa:8)]→C	—	—
	BIXOR #xx:3,@aa:16	B						6			C⊕[¬ (#xx:3 of @aa:16)]→C	—	—
	BIXOR #xx:3,@aa:32	B						8			C⊕[¬ (#xx:3 of @aa:32)]→C	—	—

(6) Branch Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Branching Condition	Condition Code					
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C
Bcc	BRA d:8(BT d:8)	—							2			if condition is true then PC←PC+d else next;	Always	—	—	—	—	—	—
	BRA d:16(BT d:16)	—							4					—	—	—	—	—	—
	BRN d:8(BF d:8)	—							2				Never	—	—	—	—	—	—
	BRN d:16(BF d:16)	—							4					—	—	—	—	—	—
	BHI d:8	—							2				C∨Z=0	—	—	—	—	—	—
	BHI d:16	—							4					—	—	—	—	—	—
	BLS d:8	—							2				C∨Z=1	—	—	—	—	—	—
	BLS d:16	—							4					—	—	—	—	—	—
	BCC d:B(BHS d:8)	—							2				C=0	—	—	—	—	—	—
	BCC d:16(BHS d:16)	—							4					—	—	—	—	—	—
	BCS d:8(BLO d:8)	—							2				C=1	—	—	—	—	—	—
	BCS d:16(BLO d:16)	—							4					—	—	—	—	—	—
	BNE d:8	—							2				Z=0	—	—	—	—	—	—
	BNE d:16	—							4					—	—	—	—	—	—
	BEQ d:8	—							2				Z=1	—	—	—	—	—	—
	BEQ d:16	—							4					—	—	—	—	—	—
	BVC d:8	—							2				V=0	—	—	—	—	—	—
	BVC d:16	—							4					—	—	—	—	—	—

RENESAS

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code								
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	U	N	Z	V	C		
Bcc	BVS d:8	—							2			V=1	—	—	—	—	—	—	—		
	BVS d:16	—							4				—	—	—	—	—	—	—		
	BPL d:8	—							2			N=0	—	—	—	—	—	—	—		
	BPL d:16	—							4				—	—	—	—	—	—	—		
	BMI d:8	—							2			N=1	—	—	—	—	—	—	—		
	BMI d:16	—							4				—	—	—	—	—	—	—		
	BGE d:8	—							2			N⊕V=0	—	—	—	—	—	—	—		
	BGE d:16	—							4				—	—	—	—	—	—	—		
	BLT d:8	—							2			N⊕V=1	—	—	—	—	—	—	—		
	BLT d:16	—							4				—	—	—	—	—	—	—		
	BGT d:8	—							2			Z∨(N⊕V)=0	—	—	—	—	—	—	—		
	BGT d:16	—							4				—	—	—	—	—	—	—		
	BLE d:8	—							2			Z∨(N⊕V)=1	—	—	—	—	—	—	—		
	BLE d:16	—							4				—	—	—	—	—	—	—		

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H
JMP	JMP @ERn	—			2						PC←ERn	—	—
	JMP @aa:24	—						4			PC←aa:24	—	—
	JMP @@aa:8	—								2	PC←@aa:8	—	—
BSR	BSR d:8	—							2		PC→@-SP,PC←PC+d:8	—	—
	BSR d:16	—							4		PC→@-SP,PC←PC+d:16	—	—
JSR	JSR @ERn	—			2						PC→@-SP,PC←ERn	—	—
	JSR @aa:24	—						4			PC→@-SP,PC←aa:24	—	—
	JSR @@aa:8	—								2	PC→@-SP,PC←@aa:8	—	—
RTS	RTS	—									PC←@SP+	—	—
										2			

RENESAS

(7) System Control Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)								Operation	Condition Code					
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa		I	H	N	Z	V	C
TRAPA	TRAPA #xx:2	—									PC→@-SP,CCR→@-SP, EXR→@-SP,<vector>→PC	1	—	↕	↕	↕	↕
RTE	RTE	—									EXR←@SP+,CCR←@SP+, PC←@SP+	↕	↕	↕	↕	↕	↕
SLEEP	SLEEP	—									Transition to power-down state	—	—	—	—	—	—
LDC	LDC #xx:8,CCR	B	2								#xx:8→CCR	↕	↕	↕	↕	↕	↕
	LDC #xx:8,EXR	B	4								#xx:8→EXR	—	—	↕	↕	↕	↕
	LDC Rs,CCR	B		2							Rs8→CCR	↕	↕	↕	↕	↕	↕
	LDC Rs,EXR	B		2							Rs8→EXR	—	—	↕	↕	↕	↕
	LDC @ERs,CCR	W			4						@ERs→CCR	↕	↕	↕	↕	↕	↕
	LDC @ERs,EXR	W			4						@ERs→EXR	—	—	↕	↕	↕	↕
	LDC @(d:16,ERs),CCR	W				6					@(d:16,ERs)→CCR	↕	↕	↕	↕	↕	↕
	LDC @(d:16,ERs),EXR	W				6					@(d:16,ERs)→EXR	—	—	↕	↕	↕	↕
	LDC @(d:32,ERs),CCR	W				10					@(d:32,ERs)→CCR	↕	↕	↕	↕	↕	↕
	LDC @(d:32,ERs),EXR	W				10					@(d:32,ERs)→EXR	—	—	↕	↕	↕	↕
	LDC @ERs+,CCR	W					4				@ERs→CCR,ERs32+2→ERs32	↕	↕	↕	↕	↕	↕
	LDC @ERs+,EXR	W					4				@ERs→EXR,ERs32+2→ERs32	—	—	↕	↕	↕	↕
	LDC @aa:16,CCR	W						6			@aa:16→CCR	↕	↕	↕	↕	↕	↕
	LDC @aa:16,EXR	W						6			@aa:16→EXR	—	—	↕	↕	↕	↕

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H
STC	STC CCR,Rd	B	2									CCR→Rd8	—	—
	STC EXR,Rd	B	2									EXR→Rd8	—	—
	STC CCR,@ERd	W			4							CCR→@ERd	—	—
	STC EXR,@ERd	W			4							EXR→@ERd	—	—
	STC CCR,@(d:16,ERd)	W				6						CCR→@(d:16,ERd)	—	—
	STC EXR,@(d:16,ERd)	W				6						EXR→@(d:16,ERd)	—	—
	STC CCR,@(d:32,ERd)	W				10						CCR→@(d:32,ERd)	—	—
	STC EXR,@(d:32,ERd)	W				10						EXR→@(d:32,ERd)	—	—
	STC CCR,@-ERd	W					4					ERd32-2→ERd32,CCR→@ERd	—	—
	STC EXR,@-ERd	W					4					ERd32-2→ERd32,EXR→@ERd	—	—
	STC CCR,@aa:16	W						6				CCR→@aa:16	—	—
	STC EXR,@aa:16	W						6				EXR→@aa:16	—	—
	STC CCR,@aa:32	W						8				CCR→@aa:32	—	—
	STC EXR,@aa:32	W						8				EXR→@aa:32	—	—
ANDC	ANDC #xx:8,CCR	B	2									CCR∧#xx:8→CCR	↕	↕
	ANDC #xx:8,EXR	B	4									EXR∧#xx:8→EXR	—	—
ORC	ORC #xx:8,CCR	B	2									CCR∨#xx:8→CCR	↕	↕
	ORC #xx:8,EXR	B	4									EXR∨#xx:8→EXR	—	—
XORC	XORC #xx:8,CCR	B	2									CCR⊕#xx:8→CCR	↕	↕
	XORC #xx:8,EXR	B	4									EXR⊕#xx:8→EXR	—	—

(8) Block Transfer Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code	
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H
EEPMOV	EEPMOV.B	—									4	if R4L≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4L-1→R4L Until R4L=0 else next;	—	—
	EEPMOV.W	—									4	if R4≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4-1→R4 Until R4=0 else next;	—	—

Notes: 1. The number of states is the number of states required for execution when the instruction and its operand are located in on-chip memory.
2. 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/2345 Group.
[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.

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							

RENESAS

Instruction	Mnemonic	Size	Instruction Format							
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8
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	8
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 I
	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	r
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 I
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 I
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 I

Instruc-tion	Mnemonic	Size	Instruction Format																											
			1st byte				2nd byte				3rd byte				4th byte				5th byte				6th byte				7th 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
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
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
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
	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

Instruction	Mnemonic	Size	Instruction Format																													
			1st byte				2nd byte				3rd byte				4th byte				5th byte				6th byte				7th byte				8	
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	I		
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	I		
	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	r			
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	I		
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	I		
	BTST Rn,Rd	B	6	3	rn			rd																								

Instruction	Mnemonic	Size	Instruction Format 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th 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	r
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:I
CLRMAC	CLRMAC	—	Cannot be used in the H8S/2345 Group							
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				

Instruc-tion	Mnemonic	Size	Instruction Format																			
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th 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								
	LDC @(d:32,ERs),EXR	W	0	1	4	1	7	8	0 ers	0	6	B	2	0								
	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	disp											

Instruction	Mnemonic	Size	Instruction Format: 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th 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/2345 Group							
	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			

RENESAS

Instruction	Mnemonic	Size	Instruction Format: 1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th 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)*	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/2345 Group						
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					

RENESAS

Instruction	Mnemonic	Size	Instruction Format						
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th 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	8
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						

RENESAS

Instruction	Mnemonic	Size	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th 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	
	STC.W EXR,@(d:32,ERd)	W	0 1	4 1	7 8	0 erd 0	6 B	A 0	
	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					
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/2345 Group																	
	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	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										

Instruc-tion	Mnemonic	Size	Instruction Format							
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	
XORC	XORC #xx:8,CCR	B	0 5	IMM						
	XORC #xx:8,EXR	B	0 1	4 1	0 5	IMM				

Note: * Bit 7 of the 4th byte of the MOV.L ERs, @(d:32,ERd) instruction can be either 1 or 0.

Legend:

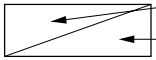
IMM:	Immediate data (2, 3, 8, 16, or 32 bits)
abs:	Absolute address (8, 16, 24, or 32 bits)
disp:	Displacement (8, 16, or 32 bits)
rs, rd, rn:	Register field (4 bits specifying an 8-bit or 16-bit register. The symbols rs, rd, and rn correspond to op
ers, erd, ern, erm:	Register field (3 bits specifying an address register or 32-bit register. The symbols ers, erd, ern, and e 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

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	
0	NOP	Table A.3(2)	STC / STMAC*	LDC / LDMAC*	ORC	XORC	ANDC	LDC	ADD		Table A.3(2)	Table A.3(2)	MOV		AD
1	Table A.3(2)	Table A.3(2)	Table A.3(2)	Table A.3(2)	OR	XOR	AND	Table A.3(2)	SUB		Table A.3(2)	Table A.3(2)	CMP		SU
2	MOV.B														
3															
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	B
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.3(2)	JMP			BSR	J	
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BIST	MOV		Table A.3(2)	MOV			
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD	MOV	Table A.3(2)	Table A.3(2)	EEPMOV	Table A.		
8	ADD														
9	ADDX														
A	CMP														
B	SUBX														
C	OR														
D	XOR														
E	AND														
F	MOV														

Note: * Cannot be used in the H8S/2345 Group.

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
01	MOV	LDM / STM			LDC / STC		MAC*		SLEEP		CLRMAC*		Table A.3(3)
0A	INC								ADD				
0B	ADDS					INC		INC	ADDS				
0F	DAA								MOV				
10	SHLL			SHLL				SHLL	SHAL			SHAL	
11	SHLR			SHLR				SHLR	SHAR			SHAR	
12	ROTXL			ROTXL				ROTXL	ROTL			ROTL	
13	ROTXR			ROTXR				ROTXR	ROTR			ROTR	
17	NOT			NOT		EXTU		EXTU	NEG			NEG	
1A	DEC								SUB				
1B	SUBS					DEC		DEC	SUBS				
1F	DAS								CMP				
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE
6A	MOV	Table A.3(4)	MOV	Table A.3(4)	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/2345 Group.

Instruction code

1st byte		2nd byte		3rd byte		4th byte	
AH	AL	BH	BL	CH	CL	DH	DL

Instruction whe
Instruction whe

AH AL BH BL CH \ CL	0	1	2	3	4	5	6	7	8	9	A	B	C
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.

RENESAS

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 whe
Instruction whe

EL / AHALBHBLCHCLDHDLEH	0	1	2	3	4	5	6	7	8	9	A	B	C
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 b
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL	GH	GL	HH

Instruction whe
Instruction whe

GL / AHALBHBL ... FHFLGH	0	1	2	3	4	5	6	7	8	9	A	B	C
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

I J K L M N

Examples: Advanced mode, program code and stack located in external memory, on-
supporting modules accessed in two states with 8-bit bus width, external devices acce
states with one wait state and 16-bit bus width.

1. BSET #0, @FFFFC7:8

   From table A.5:

   I = L = 2, J = K = M = N = 0

   From table A.4:

   $S_I = 4$, $S_L = 2$

   Number of states required for execution = $2 \times 4 + 2 \times 2 = 12$

2. JSR @@30

   From table A.5:

   I = J = K = 2, L = M = N = 0

   From table A.4:

   $S_I = S_J = S_K = 4$

   Number of states required for execution = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Instruction fetch	S_I	2	4	2	4	6 + 2m	2
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

Legend:

m: Number of wait states inserted into external device access

	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		

	BLT d:8	2		
	BGT d:8	2		
	BLE d:8	2		
	BRA d:16 (BT d:16)	2		
	BRN d:16 (BF d:16)	2		
	BHI d:16	2		
	BLS d:16	2		
	BCC d:16 (BHS d:16)	2		
	BCS d:16 (BLO d:16)	2		
	BNE d:16	2		
	BEQ d:16	2		
	BVC d:16	2		
	BVS d:16	2		
	BPL d:16	2		
	BMI d:16	2		
	BGE d:16	2		
	BLT d:16	2		
	BGT d:16	2		
	BLE d:16	2		
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

	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

	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	
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

	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/2345 Group	
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	
	DIVXS.W Rs,ERd	2	
DIVXU	DIVXU.B Rs,Rd	1	
	DIVXU.W Rs,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				
	JMP @@aa:8	2	2			
JSR	JSR @ERn	2		2		
	JSR @aa:24	2		2		
	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
	LDC @ERs+,EXR	2				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

	(ERn-ERn+3)				
LDMAC	LDMAC ERs,MACH	Cannot be used in the H8S/2345 Group			
	LDMAC ERs,MACL				
MAC	MAC @ERn+,@ERm+	Cannot be used in the H8S/2345 Group			
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	
	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	
	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
	MOV.W @aa:16,Rd	2			1
	MOV.W @aa:32,Rd	3			1
	MOV.W Rs,@ERd	1			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
	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
	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/2345 Group	
MOVTPE	MOVTPE Rs,@:aa:16		
MULXS	MULXS.B Rs,Rd	2	
	MULXS.W Rs,ERd	2	
MULXU	MULXU.B Rs,Rd	1	
	MULXU.W Rs,ERd	1	
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	

	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
	POP.L ERn	2		2
PUSH	PUSH.W Rn	1		1
	PUSH.L ERn	2		2
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		

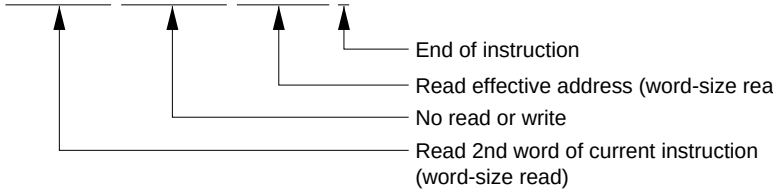
	ROTXR.L ERd	1	
	ROTXR.L #2,ERd	1	
RTE	RTE	2	2/3[*1]
RTS	RTS	2	2
SHAL	SHAL.B Rd	1	
	SHAL.B #2,Rd	1	
	SHAL.W Rd	1	
	SHAL.W #2,Rd	1	
	SHAL.L ERd	1	
	SHAL.L #2,ERd	1	
SHAR	SHAR.B Rd	1	
	SHAR.B #2,Rd	1	
	SHAR.W Rd	1	
	SHAR.W #2,Rd	1	
	SHAR.L ERd	1	
	SHAR.L #2,ERd	1	
SHLL	SHLL.B Rd	1	
	SHLL.B #2,Rd	1	
	SHLL.W 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	

	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
	STC.W EXR,@-ERd	2			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	
	STM.L (ERn-ERn+2), @-SP	2		6	
	STM.L (ERn-ERn+3), @-SP	2		8	
STMAC	STMAC MACH,ERd	Cannot be used in the H8S/2345 Group			
	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	2			2
TRAPA	TRAPA #x:2 Advanced	2	2	2/3*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.

Instruction	1	2	3	4	5	6	7
JMP@aa:24	R:W 2nd	Internal operation 1 state	R:W EA				



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

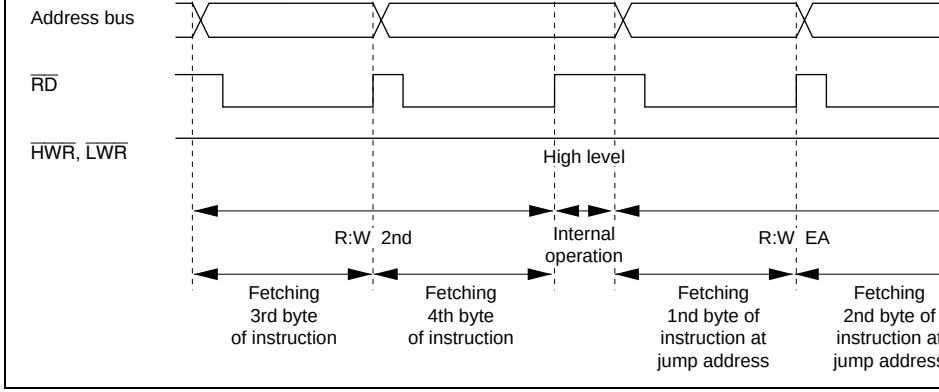


Figure A.1 Address Bus, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ Timing (8-Bit Bus, Three-State Access, No Wait States)

Instruction	1	2	3	4	5	6	7
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					

Instruction	1	2	3	4	5	6	7
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			

Instruction	1	2	3	4	5	6	7
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			

Instruction	1	2	3	4	5	6	7
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
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/2345 Group						
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						

Instruction	1	2	3	4	5	6	7
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		

Instruction	1	2	3	4	5	6	7
LDM.L @SP+,(ERn–ERn+2)	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3		
LDM.L @SP+,(ERn–ERn+3)	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3		
LDMAC ERs,MACH	Cannot be used in the H8S/2345 Group						
LDMAC ERs,MACL							
MAC @ERn+,@ERm+							
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				

Instruction	1	2	3	4	5	6	7
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			
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/2345 Group						
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						

Instruction	1	2	3	4	5	6	7
OR.W #xx:16,Rd	R:W 2nd	R:W NEXT					
OR.W Rs,Rd	R:W NEXT						
OR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT				
OR.L ERs,ERd	R:W 2nd	R:W NEXT					
ORC #xx:8,CCR	R:W NEXT						
ORC #xx:8,EXR	R:W 2nd	R:W NEXT					
POP.W Rn	R:W NEXT	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						

Instruction	1	2	3	4	5	6	7
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						
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			

Instruction	1	2	3	4	5	6	7
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			
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/2345 Group						
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	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
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				

Instruction	1	2	3	4	5	6	7
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*5			
Interrupt exception handling	R:W*6	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+2

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 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 regist
4. Start address after return.
5. Start address of the program.
6. Prefetch address, equal to two plus the PC value pushed onto the stack. In recovery from sleep mode or softw operation is replaced by an internal operation.
7. Start address of the interrupt-handling routine.

	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

ADDS	—	—	—	—	—	
ADDX	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{\text{Rm–4}}$ + Sm–4 · $\overline{\text{Rm}}$
						N = Rm
						Z = Z' · $\overline{\text{Rm}}$ · ····· · $\overline{\text{R0}}$
						V = Sm · Dm · $\overline{\text{Rm}}$ + $\overline{\text{Sm}}$ · $\overline{\text{Dm}}$ · Rm
						C = Sm · Dm + Dm · $\overline{\text{Rm}}$ + Sm · $\overline{\text{Rm}}$
AND	—	↕	↕	0	—	N = Rm
						Z = $\overline{\text{Rm}}$ · $\overline{\text{Rm–1}}$ · ····· · $\overline{\text{R0}}$
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' · $\overline{\text{Dn}}$
BILD	—	—	—	—	↕	C = $\overline{\text{Dn}}$
BIOR	—	—	—	—	↕	C = C' + $\overline{\text{Dn}}$
BIST	—	—	—	—	—	
BIXOR	—	—	—	—	↕	C = C' · Dn + $\overline{\text{C'}}$ · $\overline{\text{Dn}}$
BLD	—	—	—	—	↕	C = Dn
BNOT	—	—	—	—	—	
BOR	—	—	—	—	↕	C = C' + Dn
BSET	—	—	—	—	—	
BSR	—	—	—	—	—	
BST	—	—	—	—	—	
BTST	—	—	↕	—	—	Z = $\overline{\text{Dn}}$
BXOR	—	—	—	—	↕	C = C' · $\overline{\text{Dn}}$ + $\overline{\text{C'}}$ · Dn

DAA	*	↕	↕	*	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C: decimal arithmetic carry
DAS	*	↕	↕	*	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C: decimal arithmetic borrow
DEC	—	↕	↕	↕	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = Dm · $\overline{Rm}$
DIVXS	—	↕	↕	—	—	N = Sm · $\overline{Dm}$ + $\overline{Sm}$ · Dm Z = $\overline{Sm} \cdot \overline{Sm-1} \cdot \cdots\cdots \cdot \overline{S0}$
DIVXU	—	↕	↕	—	—	N = Sm Z = $\overline{Sm} \cdot \overline{Sm-1} \cdot \cdots\cdots \cdot \overline{S0}$
EEPMOV	—	—	—	—	—	
EXTS	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
EXTU	—	0	↕	0	—	Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
INC	—	↕	↕	↕	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Dm}$ · Rm
JMP	—	—	—	—	—	
JSR	—	—	—	—	—	
LDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
LDM	—	—	—	—	—	

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)

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 = $Dm \cdot Dm-1 + \overline{Dm} \cdot \overline{Dm-1}$ (1-bit shift) V = $Dm \cdot Dm-1 \cdot Dm-2 \cdot \overline{Dm} \cdot \overline{Dm-1} \cdot \overline{Dm-2}$ (2 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/2345 Group

SUBX	↕	↕	↕	↕	↕	H = $\overline{Sm–4}$ · Dm–4 + Dm–4 · $\overline{Rm–4}$ + $\overline{Sm–4}$ · Rm 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.

H'FBFF	SAR									
		MRB	CHNE	DISEL	—	—	—	—	—	—
		DAR								
		CRA								
		CRB								
H'FE80	TCR3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU3
H'FE81	TMDR3	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
H'FE82	TIOR3H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FE83	TIOR3L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
H'FE84	TIER3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
H'FE85	TSR3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
H'FE86	TCNT3									
H'FE87										
H'FE88	TGR3A									
H'FE89										
H'FE8A	TGR3B									
H'FE8B										
H'FE8C	TGR3C									
H'FE8D										
H'FE8E	TGR3D									
H'FE8F										

H'FE97										
H'FE98	TGR4A									
H'FE99										
H'FE9A	TGR4B									
H'FE9B										
H'FEA0	TCR5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU5
H'FEA1	TMDR5	—	—	—	—	MD3	MD2	MD1	MD0	
H'FEA2	TIOR5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FEA4	TIER5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FEA5	TSR5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FEA6	TCNT5									
H'FEA7										
H'FEA8	TGR5A									
H'FEA9										
H'FEAA	TGR5B									
H'FEAB										
H'FEB0	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	Port
H'FEB1	P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR	
H'FEB2	P3DDR	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR	
H'FEB9	PADDR	—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR	
H'FEBA	PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR	
H'FEBB	PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR	
H'FEBC	PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR	
H'FEBD	PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR	
H'FEBE	PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR	PF0DDR	
H'FEBF	PGDDR	—	—	—	PG4DDR	PG3DDR	PG2DDR	PG1DDR	PG0DDR	

H'FECA	IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECB	IPRH	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECC	IPRI	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECD	IPRJ	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	
H'FECE	IPRK	—	IPR6	IPR5	IPR4	—	—	—	—	
H'FED0	ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	Bus contr
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	—	EAE	—	—	—	—	WAITE	
H'FEDB	RAMER	—	—	—	—	—	RAMS	RAM1	RAM0	
H'FF2C	ISCRH	IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	Interrupt controller
H'FF2D	ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA	
H'FF2E	IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
H'FF2F	ISR	IRQ7F	IRQ6F	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F	
H'FF30 to H'FF34	DTCER	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0	DTC
H'FF37	DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0	
H'FF38	SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—	Power-do mode
H'FF39	SYSCR	—	—	INTM1	INTM0	NMIEG	—	—	RAME	MCU
H'FF3A	SCKCR	PSTOP	—	—	—	—	SCK2	SCK1	SCK0	Clock pul generator
H'FF3B	MDCR	—	—	—	—	—	MDS2	MDS1	MDS0	MCU
H'FF3C	MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	Power-do mode
H'FF3D	MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0	
H'FF42	SYSCR2	—	—	—	—	FLSHE	—	—	—	MCU

H'FF59	PORTA	—	—	—	—	PA3	PA2	PA1	PA0
H'FF5A	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
H'FF5B	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
H'FF5C	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
H'FF5D	PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
H'FF5E	PORTF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
H'FF5F	PORTG	—	—	—	PG4	PG3	PG2	PG1	PG0
H'FF60	P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
H'FF61	P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
H'FF62	P3DR	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
H'FF69	PADR	—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR
H'FF6A	PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
H'FF6B	PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
H'FF6C	PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
H'FF6D	PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
H'FF6E	PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	PF2DR	PF1DR	PF0DR
H'FF6F	PGDR	—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	PG0DR
H'FF70	PAPCR	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR
H'FF71	PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
H'FF72	PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
H'FF73	PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
H'FF74	PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
H'FF76	P3ODR	—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR
H'FF77	PAODR	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR

H'FF7D	RDR0									
H'FF7E	SCMR0	—	—	—	—	SDIR	SINV	—	SMIF	
H'FF80	SMR1	C/$\overline{A}$/ GM*2	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI1, Smart ca
H'FF81	BRR1									interface
H'FF82	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FF83	TDR1									
H'FF84	SSR1	TDRE	RDRF	ORER	FER/ ERS*3	PER	TEND	MPB	MPBT	
H'FF85	RDR1									
H'FF86	SCMR1	—	—	—	—	SDIR	SINV	—	SMIF	
H'FF90	ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D conv
H'FF91	ADDRAL	AD1	AD0	—	—	—	—	—	—	
H'FF92	ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FF93	ADDRBL	AD1	AD0	—	—	—	—	—	—	
H"FF94	ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FF95	ADDRCL	AD1	AD0	—	—	—	—	—	—	
H'FF96	ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'FF97	ADDRDL	AD1	AD0	—	—	—	—	—	—	
H'FF98	ADCSR	ADF	ADIE	ADST	SCAN	CKS	—	CH1	CH0	
H'FF99	ADCR	TRGS1	TRGS0	—	—	—	—	—	—	
H'FFA4	DADR0									D/A conv
H'FFA5	DADR1									
H'FFA6	DACR	DAOE1	DAOE0	DAE	—	—	—	—	—	

H'FFB6	TCORB0									
H'FFB7	TCORB1									
H'FFB8	TCNT0									
H'FFB9	TCNT1									
H'FFBC (read)	TCSR	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	WDT
H'FFBD (read)	TCNT									
H'FFBF (read)	RSTCSR	WOVF	RSTE	RSTS	—	—	—	—	—	
H'FFC0	TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU
H'FFC1	TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
H'FFC8	FLMCR1	FWE	SWE	—	—	EV	PV	E	P	FLASH
H'FFC9	FLMCR2	FLER	—	—	—	—	—	ESU	PSU	
H'FFCA	EBR1	—	—	—	—	—	—	EB9	EB8	
H'FFCB	EBR2	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0	
H'FFD0	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU0
H'FFD1	TMDR0	—	—	BFB	BFA	MD3	MD2	MD1	MD0	
H'FFD2	TIOR0H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFD3	TIOR0L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0	
H'FFD4	TIER0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA	
H'FFD5	TSR0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA	
H'FFD6	TCNT0									
H'FFD7										
H'FFD8	TGR0A									
H'FFD9										
H'FFDA	TGR0B									
H'FFDB										

H'FFE2	TIOR1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFE4	TIER1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FFE5	TSR1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FFE6	TCNT1									
H'FFE7										
H'FFE8	TGR1A									
H'FFE9										
H'FFEA	TGR1B									
H'FFEB										
H'FFF0	TCR2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU2
H'FFF1	TMDR2	—	—	—	—	MD3	MD2	MD1	MD0	
H'FFF2	TIOR2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0	
H'FFF4	TIER2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA	
H'FFF5	TSR2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA	
H'FFF6	TCNT2									
H'FFF7										
H'FFF8	TGR2A									
H'FFF9										
H'FFFA	TGR2B									
H'FFFB										

Notes: 1. Located in on-chip RAM. The bus width is 32 bits when the DTC accesses register information, and 16 bits otherwise.
2. Functions as C/$\overline{A}$ for SCI use, and as GM for smart card interface use.
3. Functions as FER for SCI use, and as ERS for smart card interface use.

DTC D
Transf

0	By tra
1	Wo tra

DTC Transfer Mode Sel

0	Destination side is area or block area
1	Source side is rep or block area

DTC Mode

0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

Destination Address Mode

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

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)

Reserved
Only 0 should be written t

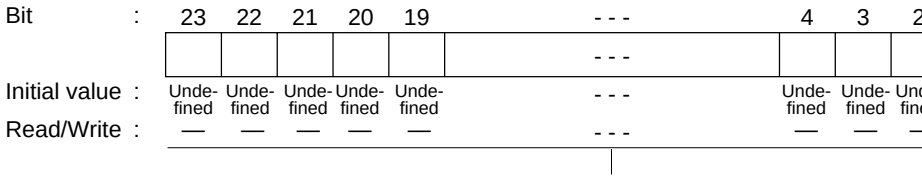
DTC Interrupt Select

0	After a data transfer ends, the CPU interrupt is disabled unless the transfer counter is 0
1	After a data transfer ends, the CPU interrupt is en

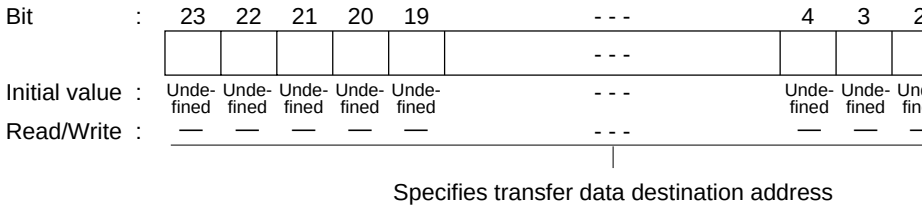
DTC Chain Transfer Enable

0	End of DTC data transfer
1	DTC chain transfer

SAR—DTC Source Address Register **H'F800—H'FBFF**



DAR—DTC Destination Address Register **H'F800—H'FBFF**



Specifies the number of DTC data transfers

CRB—DTC Transfer Count Register B **H'F800—H'FBFF**

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
Initial value	:	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fine
Read/Write	:	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Specifies the number of DTC block data transfers

0	0	0	Internal clock: counts on φ/
		1	Internal clock: counts on φ/
	1	0	Internal clock: counts on φ/
		1	Internal clock: counts on φ/
1	0	0	External clock: counts on T
		1	Internal clock: counts on φ/
	1	0	Internal clock: counts on φ/
		1	Internal clock: counts on φ/

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Counter Clear

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.

0	0	0	0	Normal op
			1	Reserved
		1	0	PWM mod
			1	PWM mod
	1	0	0	Phase cou
			1	Phase cou
		1	0	Phase cou
			1	Phase cou
1	*	*	*	—

Notes: 1. MD3 is a reserved bi
it should always be w
2. Phase counting mod
set for channels 0 ar
case, 0 should alway
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

					Output disabled	
			1	TGR3A is output compare register	Initial output is 0 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c
1	0	0	0	TGR3A is input capture register	Capture input source is TIOCA3 pin	Input capture at ris
			1			Input capture at fa
		1	*			Input capture at bo
	1	*	*		Capture input source is channel 4/count clock	Input capture at T count-down

TGR3B I/O Control

0	0	0	0	TGR3B 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	TGR3B is input capture register	Capture input source is TIOCB3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down*1

*: Don't care

Note: 1. If bits TPSC2 to TPSC0 in TCR4 are set to B'000, and φ/1 is used as the TCNT4 count clock, this setting will be invalid and input capture will not occur.

			1	is output compare register	Initial output is 0 output	0 output at compare mat
		1	0			1 output at compare mat
			1			Toggle output at compar
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare mat
		1	0			1 output at compare mat
			1			Toggle output at compar
1	0	0	0	TGR3C is input capture register	Capture input source is TIOCC3 pin	Input capture at rising ed
			1			Input capture at falling ed
		1	*			Input capture at both edg
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-down

*

Note: When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a bu this setting is invalid and input capture/output compare is not gener

TGR3D I/O Control

0	0	0	0	TGR3D 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	TGR3D is input capture register	Capture input source is TIOCD3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down*1

*: Don't care

Notes: When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and φ/1 is used as the TCNT4 count clock, this setting is invalid and input capture 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.

0	Interrupt re by TGFA bi
1	Interrupt re by TGFA bi

TGR Interrupt Enable

0	Interrupt request by TGFB bit disa
1	Interrupt request by TGFB bit ena

TGR Interrupt Enable C

0	Interrupt requests (TGIC) b TGFC bit disabled
1	Interrupt requests (TGIC) b 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

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

	DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after re
1	[Setting conditions] • When TCNT=TGRA while TGRA ing as output compare register • When TCNT value is transferred input capture signal while TGRA input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt v of MRB in DTC is 0 • When 0 is written to TGFB after reading TC
1	[Setting conditions] • When TCNT = TGRB while TGRB is functi output compare register • When TCNT value is transferred to TGRB capture signal while TGRB is functioning a register

Input Capture/Output Compare Flag C

0	[Clearing conditions] • When DTC is activated by TGIC interrupt while DISEL b DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as ou register • When TCNT value is transferred to TGRC by input cap 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 M is 0 • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output cor • When TCNT value is transferred to TGRD by input capture sig 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.

TGR3A—Timer General Register 3A	**H'FE88**
TGR3B—Timer General Register 3B	**H'FE8A**
TGR3C—Timer General Register 3C	**H'FE8C**
TGR3D—Timer General Register 3D	**H'FE8E**

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

0	0	0	Internal clock: counts on φ/1
		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/1
		1	Internal clock: counts on φ/6
1	0	0	External clock: counts on TC
		1	External clock: counts on TC
	1	0	Internal clock: counts on φ/1
		1	Counts on TCNT5 overflow/

Note: This setting is ignored when channe counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is ignored when channel 4 is in phase counting mode.

Counter Clear

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*

Note: * Synchronous operating setting is performed by setting the SYNC bit TSYR to 1.

0	0	0	0	Normal ope
			1	Reserved
		1	0	PWM mode
			1	PWM mode
	1	0	0	Phase cou
			1	Phase cou
		1	0	Phase cou
			1	Phase cou
1	*	*	*	—

Note: MD3 is a reserved bit. In
it should always be writt

			1	is output compare register	Initial output is 0 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c
1	0	0	0	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at ris
			1			Input capture at fa
		1	*			Input capture at bo
	1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at ge TGR3A compare m capture

TGR4B I/O Control

0	0	0	0	TGR4B 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	TGR4B is input capture register	Capture input source is TIOCB4 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR3C compare match/ input capture	Input capture at generation of TGR3C compare match/input capture

*: Don't care

	by TGFA bit disable
1	Interrupt requests (T by TGFA bit enable

TGR Interrupt Enable B

0	Interrupt requests (TGIB) TGFB bit disabled
1	Interrupt requests (TGIB) 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

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

	DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after rea
1	[Setting conditions] • When TCNT = TGRA while TGRA i as output compare register • When TCNT value is transferred to input capture signal while TGRA is as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt bit of MRB in DTC is 0 • When 0 is written to TGFB after reading T
1	[Setting conditions] • When TCNT = TGRB while TGRB is funct output compare register • When TCNT value is transferred to TGRB capture signal while TGRB is functioning a 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.

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase co
mode or when performing overflow/underflow counting on another chan
other cases it functions as an up-counter.

TGR4A—Timer General Register 4A **H'FE98**
TGR4B—Timer General Register 4B **H'FE9A**

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

		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TCL
		1	External clock: counts on TCL
	1	0	Internal clock: counts on φ/25
		1	External clock: counts on TCL

Note: This setting is ignored when channel 5 counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is ignored when channel 5 is in phase counting mode.

Counter Clear

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*

Note: * Synchronous operating setting is performed by setting the SYNC bit TSYR to 1.

0	0	0	0	Normal o
			1	Reserved
		1	0	PWM mo
			1	PWM mo
	1	0	0	Phase co
			1	Phase co
		1	0	Phase co
			1	Phase co
1	*	*	*	—

Note: MD3 is a reserved bit.
it should always be wri

			1	is output compare register	Initial output is 0 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compa
		1	0			1 output at compa
			1			Toggle output at c
1	*	0	0	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at ri
			1			Input capture at fa
		1	*			Input capture at b

TGR5B I/O Control

0	0	0	0	TGR5B 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	TGR5B is input capture register	Capture input source is TIOCB5 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

	by TGFA bit disable
1	Interrupt requests (T by TGFA bit enable

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

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

	DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after rea
1	[Setting conditions] • When TCNT = TGRA while TGRA as output compare register • When TCNT value is transferred input capture signal while TGRA as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrup bit of MRB in DTC is 0 • When 0 is written to TGFB after reading
1	[Setting conditions] • When TCNT = TGRB while TGRB is fun output compare register • When TCNT value is transferred to TGR capture signal while TGRB is functioning 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.

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase co
mode or when performing overflow/underflow counting on another chan
other cases it functions as an up-counter.

TGR5A—Timer General Register 5A **H'FEA8**
TGR5B—Timer General Register 5B **H'FEAA**

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

P1DDR—Port 1 Data Direction Register **H'FEB0**

Bit	:	7	6	5	4	3	2	1
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specify input or output for individual port 1 pins

Specify input or output for individual port 2 pins

P3DDR—Port 3 Data Direction Register H'FEB2

Bit	:	7	6	5	4	3	2	1
		—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR
Initial value	:	Undefined	Undefined	0	0	0	0	0
Read/Write	:	—	—	W	W	W	W	W

Specify input or output for individual port 3 p

PADDR—Port A Data Direction Register H'FEB9

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	PA3DDR	PA2DDR	PA1DDR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0
Read/Write	:	—	—	—	—	W	W	W

Specify input or output for individual

Specify input or output for individual port B pins

PCDDR—Port C Data Direction Register H'FEBB

Bit	:	7	6	5	4	3	2	1
		PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DD
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specify input or output for individual port C pins

PDDDR—Port D Data Direction Register H'FEBC

Bit	:	7	6	5	4	3	2	1
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specify input or output for individual port D pins

Specify input or output for individual port E pins

PFDDR—Port F Data Direction Register H'FEBE

Bit	:	7	6	5	4	3	2	1
		PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	PF2DDR	PF1DDR
Modes 1, 2, 4 to 6								
Initial value	:	1	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W
Modes 3, 7								
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	W	W	W	W	W	W	W

Specify input or output for individual port F pins

Modes 2, 3, 6, 7

Initial value	:	Undefined	Undefined	Undefined	0	0	0	0
Read/Write	:	—	—	—	W	W	W	W

Specify input or output for individual p

Bit	:	7	6	5	4	3	2	1	
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1	
Initial value	:	0	1	1	1	0	1	1	
Read/Write	:	—	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	WDT	—*
IPRE	—*	A/D converter
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRI	8-bit timer channel 0	8-bit timer channel 1
IPRJ	—*	SCI channel 0
IPRK	SCI channel 1	—*

Note: * Reserved bits. May be read or written, but the setting is ignore

Initial value :	0	0	0	0	0	0	0
Read/Write :	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
1	Area n is designated for 8-bit a

ASTCR—Access State Control Register **H'FED1** **Bu**

Bit :	7	6	5	4	3	2	1
	AST7	AST6	AST5	AST4	AST3	AST2	AST1
Initial value :	1	1	1	1	1	1	1
Read/Write :	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 disab
1	Area n is designated for 3-state access Wait state insertion in area n external space is enab

Area 4 Wait Control

0	0	Program wait not ins
	1	1 program wait state
1	0	2 program wait state
	1	3 program wait state

Area 5 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 6 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 7 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 0 Wait Control

0	0	Program wait not i
	1	1 program wait sta
1	0	2 program wait sta
	1	3 program wait sta

Area 1 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 2 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Area 3 Wait Control

0	0	Program wait not inserted
	1	1 program wait state inserted
1	0	2 program wait states inserted
	1	3 program wait states inserted

Only 0 should
to these bits

Burst Cycle Select 0

0	Max. 4 words in burst ac
1	Max. 8 words in burst ac

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

Only 0 should be written to thi

WAIT Pin Enable

0	Wait input by $\overline{\text{WAIT}}$ pin
1	Wait input by $\overline{\text{WAIT}}$ pin

Reserved
Only 1 should be written to these bit

External Addresses H'010000 to H'01FFFF Enable

0	On-chip ROM (H8S/2345) or reserved area* (H
1	External addresses (in external expansion mod reserved area (in single-chip mode)

Note: * Do not access a reserved area.

Reserved
Only 0 should be written to this bit

Bus Release Enable

0	External bus release is disabled
1	External bus release is enabled

RAM Select, Flash Memory Area Select

RAMS	RAM1	RAM0	RAM Are
0	*	*	H'FFEC00–H'FF
1	0	0	H'000000–H'000
		1	H'000400–H'000
	1	0	H'000800–H'000
		1	H'000C00–H'00

*: Don’t care

Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ7 to IRQ4 Sense Control

ISCRL

Bit	:	7	6	5	4	3	2	1
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SC
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ3 to IRQ0 Sense Control

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

0	IRQn interrupt disabled
1	IRQn interrupt enabled

(n = 7 to 0)

ISR—IRQ Status Register **H'FF2F** **Interrupt**

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

Indicate the status of IRQ7 to IRQ0 interrupt requests

Note: * Can only be written with 0 for flag clearing.

0	DTC activation by this interrupt is disabled [Clearing conditions] • When the DISEL bit is 1 and data transfer h • When the specified number of transfers hav
1	DTC activation by this interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified nur transfers have not ended

Correspondence between Interrupt Sources and DTCER

	Bits						
Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**
DTCERA	IRQ0	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6
DTCERB	—	ADI	TGI0A	TGI0B	TGI0C	TGI0D	TGI1
DTCERC	TGI2A	TGI2B	TGI3A	TGI3B	TGI3C	TGI3D	TGI4
DTCERD	—	—	TGI5A	TGI5B	CMIA0	CMIB0	CMIA
DTCERE	—	—	—	—	RXI0	TXI0	RXI1

Sets vector number for DTC software activation

DTC Software Activation Enable

0	DTC software activation is disabled [Clearing condition] When the DISEL bit is 0 and the specified number of transfers ha not ended
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

Note: * A value of 1 can always be written to the SWDTE bit, but 0 can only be writt
is read.

Only 0 should be writte

Output Port Enable

0	In software standby mode, address bus control signals are high-impeda
1	In software standby mode, address bus control signals retain output sta

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	Reserved
		1	Standby time = 16 states

Software Standby

0	Transition to sleep mode after execution of SLEEP instruction
1	Transition to software standby mode after execution of SLEEP i

Only 0 should be writte
to this bit

RAM Enable

0	On-chip RAM
1	On-chip RAM

NMI Input Edge Select

0	Falling edge
1	Rising edge

Interrupt Control Mode Selection

0	0	Interrupt control mode 0
	1	—
1	0	Interrupt control mode 2
	1	—

Reserved
Only 0 should be written to this bit

Bus Master Clock Select

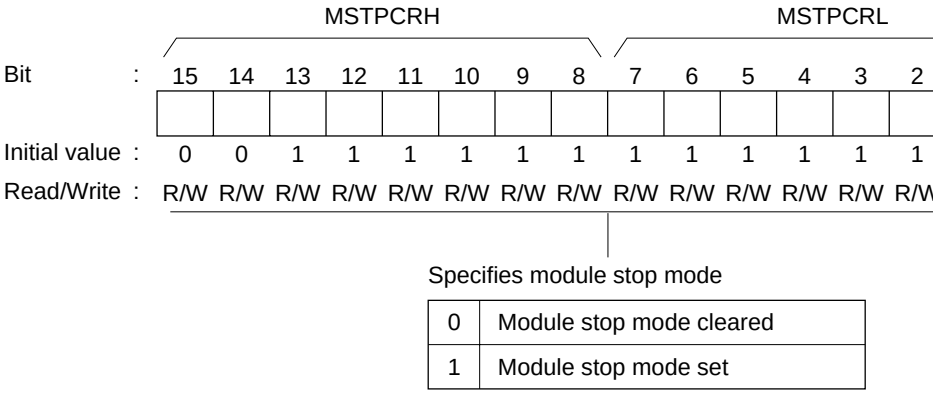
0	0	0	Bus master is in high-spe
		1	Medium-speed clock is φ
	1	0	Medium-speed clock is φ
		1	Medium-speed clock is φ
1	0	0	Medium-speed clock is φ
		1	Medium-speed clock is φ
	1	—	—

φ Clock Output Control

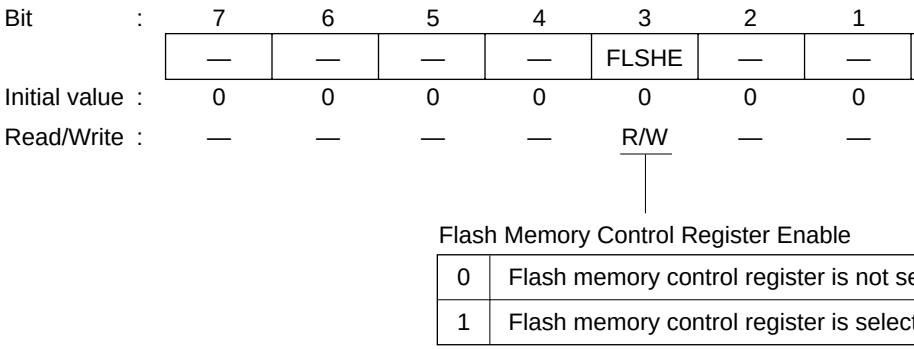
PSTOP	Normal Operation	Sleep Mode	Software Standby Mode	Hardw Standby
0	φ output	φ output	Fixed high	High imp
1	Fixed high	Fixed high	Fixed high	High imp

Note: * Determined by pins MD_2 to MD_0

MSTPCRH—Module Stop Control Register H **H'FF3C** **Power-**
MSTPCRL—Module Stop Control Register L **H'FF3D** **Power-**



SYSCR2 — System Control Register 2 **H'FF42**



Reserved
Only 0 should be written to these

Reserved Register **H'FF45**

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

Reserved

PORT1—Port 1 Register **H'FF50**

Bit	:	7	6	5	4	3	2	1
		P17	P16	P15	P14	P13	P12	P11
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port 1 pins

Note: * Determined by the state of pins $P1_7$ to $P1_0$.

State of port 2 pins

Note: * Determined by the state of pins $P2_7$ to $P2_0$.

PORT3—Port 3 Register **H'FF52**

Bit	:	7	6	5	4	3	2	1
		—	—	P35	P34	P33	P32	P31
Initial value	:	Undefined	Undefined	—*	—*	—*	—*	—*
Read/Write	:	—	—	R	R	R	R	R

State of port 3 pins

Note: * Determined by the state of pins $P3_5$ to $P3_0$.

PORT4—Port 4 Register **H'FF53**

Bit	:	7	6	5	4	3	2	1
		P47	P46	P45	P44	P43	P42	P41
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port 4 pins

Note: * Determined by the state of pins $P4_7$ to $P4_0$.

State of port A pin

Note: * Determined by the state of pins PA_3 to PA_0.

PORTB—Port B Register **H'FF5A**

Bit	:	7	6	5	4	3	2	1
		PB7	PB6	PB5	PB4	PB3	PB2	PB1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port B pins

Note: * Determined by the state of pins PB_7 to PB_0.

PORTC—Port C Register **H'FF5B**

Bit	:	7	6	5	4	3	2	1
		PC7	PC6	PC5	PC4	PC3	PC2	PC1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port C pins

Note: * Determined by the state of pins PC_7 to PC_0.

State of port D pins

Note: * Determined by the state of pins PD_7 to PD_0.

PORTE—Port E Register **H'FF5D**

Bit	:	7	6	5	4	3	2	1
		PE7	PE6	PE5	PE4	PE3	PE2	PE1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port E pins

Note: * Determined by the state of pins PE_7 to PE_0.

PORTF—Port F Register **H'FF5E**

Bit	:	7	6	5	4	3	2	1
		PF7	PF6	PF5	PF4	PF3	PF2	PF1
Initial value	:	—*	—*	—*	—*	—*	—*	—*
Read/Write	:	R	R	R	R	R	R	R

State of port F pins

Note: * Determined by the state of pins PF_7 to PF_0.

State of port G pins

Note: * Determined by the state of pins PG_4 to PG_0.

P1DR—Port 1 Data Register **H'FF60**

Bit	:	7	6	5	4	3	2	1
		P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 1 pins ($P1_7$ to $P1_0$)

P2DR—Port 2 Data Register **H'FF61**

Bit	:	7	6	5	4	3	2	1
		P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 2 pins ($P2_7$ to $P2_0$)

Stores output data for port 5 pins ($P5_5$ to $P5_0$)

PADR—Port A Data Register **H'FF69**

Bit :	7	6	5	4	3	2	1	
	—	—	—	—	PA3DR	PA2DR	PA1DR	
Initial value :	Undefined	Undefined	Undefined	Undefined	0	0	0	
Read/Write :	—	—	—	—	R/W	R/W	R/W	

Stores output data for port A pins (PA

PBDR—Port B Data Register **H'FF6A**

Bit :	7	6	5	4	3	2	1	
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	
Initial value :	0	0	0	0	0	0	0	
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Stores output data for port B pins (PB_7 to PB_0)

Stores output data for port C pins (PC_7 to PC_0)

PDDR—Port D Data Register **H'FF6C**

Bit	:	7	6	5	4	3	2	1
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port D pins (PD_7 to PD_0)

PEDR—Port E Data Register **H'FF6D**

Bit	:	7	6	5	4	3	2	1
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port E pins (PE_7 to PE_0)

Stores output data for port F pins (PF_7 to PF_0)

PGDR—Port G Data Register H'FF6F

Bit	:	7	6	5	4	3	2	1	
		—	—	—	PG4DR	PG3DR	PG2DR	PG1DR	
Initial value	:	Undefined	Undefined	Undefined	0	0	0	0	
Read/Write	:	—	—	—	R/W	R/W	R/W	R/W	

Stores output data for port G pins (PG

PAPCR—Port A MOS Pull-Up Control Register H'FF70

Bit	:	7	6	5	4	3	2	1	
		—	—	—	—	PA3PCR	PA2PCR	PA1PCR	P
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	
Read/Write	:	—	—	—	—	R/W	R/W	R/W	

Controls the MOS input pull-up function incorporated into port A on a bit-l

Controls the MOS input pull-up function incorporated into port B on a bit

PCPCR—Port C MOS Pull-Up Control Register H'FF72

Bit	:	7	6	5	4	3	2	1
		PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	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

PDPCR—Port D MOS Pull-Up Control Register H'FF73

Bit	:	7	6	5	4	3	2	1
		PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	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

Controls the MOS input pull-up function incorporated into port E on a bit-l

P3ODR—Port 3 Open Drain Control Register H'FF76

Bit	:	7	6	5	4	3	2	1
		—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR
Initial value	:	Undefined	Undefined	0	0	0	0	0
Read/Write	:	—	—	R/W	R/W	R/W	R/W	R/W

Controls the PMOS on/off status for each port 3 pin (P

PAODR—Port A Open Drain Control Register H'FF77

Bit	:	7	6	5	4	3	2	1
		—	—	—	—	PA3ODR	PA2ODR	PA1ODR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0
Read/Write	:	—	—	—	—	R/W	R/W	R/W

Controls the PMOS on/off status for each port A pin (PA

0	0	φ clock
	1	φ/4 clo
1	0	φ/16 c
	1	φ/64 c

Multiprocessor Mode

0	Multiprocessor functi
1	Multiprocessor forma

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 tra

Asynchronous Mode/Synchronous Mode Select

0	Asynchronous mode
1	Synchronous mode

0	0	φ clock
	1	φ/4 clock
1	0	φ/16 cloc
	1	φ/64 cloc

Multiprocessor Mode

0	Multiprocessor function
1	Setting prohibited

Stop Bit Length

0	Setting prohibited
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Setting prohibited
1	Parity bit addition and checking enabled

Character Length

0	8-bit data
1	Setting prohibited

GSM Mode

0	Normal smart card interface mode operation • TEND flag generated 12.5 etu after beginning of start bit • Clock output on/off control only
1	GSM mode smart card interface mode operation • TEND flag generated 11.0 etu after beginning of start bit • Fixed high/low-level control possible (set in SCR) in addition to clock output on/off c

Note: etu (Elementary Time Unit): Time for transfer of one bit

Sets the serial transfer bit rate

Note: See section 12.2.8, Bit Rate Register (BRR), for details.

		Synchronous mode	Internal clock/SCK p as serial clock outpu
	1	Asynchronous mode	Internal clock/SCK p as clock output*1
		Synchronous mode	Internal clock/SCK p as serial clock outpu
1	0	Asynchronous mode	External clock/SCK as clock input*2
		Synchronous mode	External clock/SCK as serial clock input
	1	Asynchronous mode	External clock/SCK as clock input*2
		Synchronous mode	External clock/SCK as serial clock input

Notes: 1. Outputs a clock of the same frequency
2. Inputs a clock with a frequency 16 tim

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 [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 setting of 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

0	See SCI specificatio			
1	0	0	0	Operat pin
1	0	0	1	Clock o output
1	1	0	0	Fixed-l SCK ou
1	1	0	1	Clock o output
1	1	1	0	Fixed-h SCK ou
1	1	1	1	Clock o output

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 [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 (requests, and setting of the RDRF, FER, and ORER flag SSR are disabled until data with the multiprocessor bit se 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

	Stores data for serial transmission

0	[Clearing condition] When data with a 0 multiprocessor bit is rece
1	[Setting condition] When data with a 1 multiprocessor bit is rece

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writ
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-b 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 does not match the parity setting (even or odd) specified by the O/$\overline{E}$

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 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 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: * Can only be written with 0 for flag clearing.

0	[Setting 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 and writes data
1	[Setting conditions] • On reset, or in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is 0 • When TDRE = 1 and ERS = 0 (normal transmission) 2.5 etu after a 1-byte serial character is transmitted when GM = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after a 1-byte serial character is transmitted when GM = 1

Note: etu: Elementary Time Unit (the time taken to transmit one bit)

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 pa does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bi

Error Signal Status

0	[Clearing conditions] • On reset, or in standby mode or module stop mode • When 0 is written to ERS after reading ERS = 1
1	[Setting condition] When the error signal is sampled at the low level

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its

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 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 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: * Can only be written with 0 for flag clearing.

Stores received serial data

SCMR0—Smart Card Mode Register 0 H'FF7E SCI0, Smart Card

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
Read/Write	:	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Smart Card interface function is disabled
1	Smart Card interface function is enabled

Smart Card Data Invert

0	TDR contents are transmitted a Receive data is stored in RDR
1	TDR contents are inverted bef being transmitted Receive data is stored in RDR in inverted form

Smart Card Data 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

0	0	φ clock
	1	φ/4 cloc
1	0	φ/16 clo
	1	φ/64 clo

Multiprocessor Mode

0	Multiprocessor functio
1	Multiprocessor format

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 tran

Asynchronous Mode/Synchronous Mode Select

0	Asynchronous mode
1	Synchronous mode

0	0	φ clock
	1	φ/4 cloc
1	0	φ/16 clo
	1	φ/64 clo

Multiprocessor Mode

0	Multiprocessor functio
1	Setting prohibited

Stop Bit Length

0	Setting prohibited
1	2 stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Setting prohibited
1	Parity bit addition and checking enabled

Character Length

0	8-bit data
1	Setting prohibited

GSM Mode

0	Normal smart card interface mode operation • TEND flag generated 12.5 etu after beginning of start bit • Clock output on/off control only
1	GSM mode smart card interface mode operation • TEND flag generated 11.0 etu after beginning of start bit • Fixed high/low-level control possible (set in SCR) in addition to clock output on/off

Note: etu (Elementary Time Unit): Time for transfer of one bit

Sets the serial transfer bit rate

Note: See section 12.2.8, Bit Rate Register (BRR), for details.

		Synchronous mode	Internal clock/SCK as serial clock out
	1	Asynchronous mode	Internal clock/SCK as clock output*1
		Synchronous mode	Internal clock/SCK as serial clock out
1	0	Asynchronous mode	External clock/SC as clock input*2
		Synchronous mode	External clock/SC as serial clock inp
	1	Asynchronous mode	External clock/SC as clock input*2
		Synchronous mode	External clock/SC as serial clock inp

Notes: 1. Outputs a clock of the same frequer
2. Inputs a clock with a frequency 16 t

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 [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 (ER requests, and setting of the RDRF, FER, and ORER flags i SSR are disabled until data with the multiprocessor bit set t 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

0	See SCI specification			
1	0	0	0	Operates pin
1	0	0	1	Clock ou output pi
1	1	0	0	Fixed-lov SCK outp
1	1	0	1	Clock ou output pi
1	1	1	0	Fixed-hig SCK outp
1	1	1	1	Clock ou output pi

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 [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 (E requests, and setting of the RDRF, FER, and ORER flags SSR are disabled until data with the multiprocessor bit set 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

Stores data for serial transmission

0	[Clearing condition] When data with a 0 multiprocessor bit is receiv
1	[Setting condition] When data with a 1 multiprocessor bit is receiv

Transmit End

0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and write
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-by 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 th does not match the parity setting (even or odd) specified by the O/

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 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 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: * Can only be written with 0 for flag clearing.

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 and writes d
1	[Setting conditions] • On reset, or in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is 0 • When TDRE = 1 and ERS = 0 (normal transmission) 2.5 e after a 1-byte serial character is transmitted when GM = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 e after a 1-byte serial character is transmitted when GM = 1

Note: etu: Elementary Time Unit (Time for transfer of one bit)

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 par does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit i

Error Signal Status

0	[Clearing conditions] • On reset, or in standby mode or module stop mode • When 0 is written to ERS after reading ERS =1
1	[Setting condition] When the error signal is sampled at the low level

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its pr

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 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 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: * Can only be written with 0 for flag clearing.

Stores received serial data

SCMR1—Smart Card Mode Register 1 **H'FF86** **SCI1, Smart Card**

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
Read/Write	:	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Smart Card interface function is disabled
1	Smart Card interface function is enabled

Smart Card Data Invert

0	TDR contents are transmitted Receive data is stored in RDI
1	TDR contents are inverted be being transmitted Receive data is stored in RDI in inverted form

Smart Card Data 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	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2
		AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	:	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Stores the results of A/D conversion

Analog Input Channel		A/D Data Register
Group 0	Group 1	
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6	ADDRC
AN3	AN7	ADDRD

Group select	Channel select		Single Mode	Sca
CH2	CH1	CH0		
0	0	0	AN0	AN0
		1	AN1	AN0
	1	0	AN2	AN0
		1	AN3	AN0
1	0	0	AN4	AN4
		1	AN5	AN4
	1	0	AN6	AN4
		1	AN7	AN4

Group Select

0	Conversion time= 266 states (max.)
1	Conversion time= 134 states (max.)

Scan Mode

0	Single mode
1	Scan mode

A/D Start

0	A/D conversion stopped
1	• Single mode: A/D conversion is started. Cleared to 0 automatically when conversion ends • Scan mode: A/D conversion is started. Conversion cor sequentially on the selected channels until ADST is cle 0 by software, a reset, or transition to standby mode or module stop mode

A/D Interrupt Enable

0	A/D conversion end interrupt (ADI) request disabled
1	A/D conversion end interrupt (ADI) request enabled

A/D End Flag

0	[Clearing conditions] • When 0 is written to the ADF flag after reading ADF = 1 • When the DTC is activated by an ADI interrupt, and ADDR is read
1	[Setting conditions] • Single mode: When A/D conversion ends • Scan mode: When one round of conversion has been performed on all specified channels

Note: * Can only be written with 0 for flag clearing.

Reserved bit. Write as 1.

Timer Trigger Select

TRGS1	TRGS0	Description
0	0	A/D conversion start by external trigger is disabled
	1	A/D conversion start by external trigger (TPU) is enabled
1	0	A/D conversion start by external trigger (8-bit timer) is enab
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is ena

DADR0—D/A Data Register 0 **H'FFA4**
DADR1—D/A Data Register 1 **H'FFA5**

Bit :	7	6	5	4	3	2	1
Initial value :	0	0	0	0	0	0	0
Read/Write :	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores data for D/A conversion

D/A Output Enable 0

0	Analog output DA0 is disabled
1	Channel 0 D/A conversion is enabled Analog output DA0 is enabled

D/A Output Enable 1

0	Analog output DA1 is disabled
1	Channel 1 D/A conversion is enabled Analog output DA1 is enabled

D/A Conversion Control

DAOE1	DAOE0	DAE	Description
0	0	*	Channel 0 and 1 D/A conversion disabled
	1	0	Channel 0 D/A conversion enabled Channel 1 D/A conversion disabled
		1	Channel 0 and 1 D/A conversions enabled
1	0	0	Channel 0 D/A conversion disabled Channel 1 D/A conversion enabled
		1	Channel 0 and 1 D/A conversion enabled
	1	*	Channel 0 and 1 D/A conversion enabled

*: Don't car

0	0	0	Clock input disabled
		1	Internal clock: counted at f of ϕ/8
	1	0	Internal clock: counted at f of ϕ/64
		1	Internal clock: counted at f of ϕ/8192
1	0	0	For channel 0: Count at TCNT1 overflow s For channel 1: Count at TCNT0 compare
		1	External clock: counted at
	1	0	External clock: counted at
		1	External clock: counted at falling edges

Note: * If the count input of channel 0 is the T signal and that of channel 1 is the TC match signal, no incrementing clock is Do not use this setting.

Counter Clear

0	0	Clear is disabled
	1	Clear by compare match A
1	0	Clear by compare match B
	1	Clear by rising edge of external reset input

Timer Overflow Interrupt Enable

0	OVF interrupt requests (OVI) are disabled
1	OVF interrupt requests (OVI) are enabled

Compare Match Interrupt Enable A

0	CMFA interrupt requests (CMIA) are disabled
1	CMFA interrupt requests (CMIA) are enabled

Compare Match Interrupt Enable B

0	CMFB interrupt requests (CMIB) are disabled
1	CMFB interrupt requests (CMIB) are enabled

	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS
Initial value :	0	0	0	1	0	0	0	0
Read/Write :	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/

Output Select

0	0	No change whe match A occurs
	1	0 is output whe match A occurs
1	0	1 is output whe match A occurs
	1	Output is invert compare match occurs (toggle

Output Select

0	0	No change when compare matc
	1	0 is output when compare matc
1	0	1 is output when compare matc
	1	Output is inverted when compar occurs (toggle output)

A/D Trigger Enable (TCSR0 only)

0	A/D converter start requests by compare match A are
1	A/D converter start requests by compare match A are

Timer Overflow Flag

0	[Clearing condition] Cleared by reading OVF when OVF = 1, then writing 0 to OVF
1	[Setting condition] Set when TCNT overflows (changes from H'FF to H'00)

Compare Match Flag A

0	[Clearing conditions] • Cleared by reading CMFA when CMFA = 1, then writing 0 to CMFA • When the DTC is activated by a CMIA interrupt, while DISEL bit of MRB in DT
1	[Setting condition] Set when TCNT matches TCORA

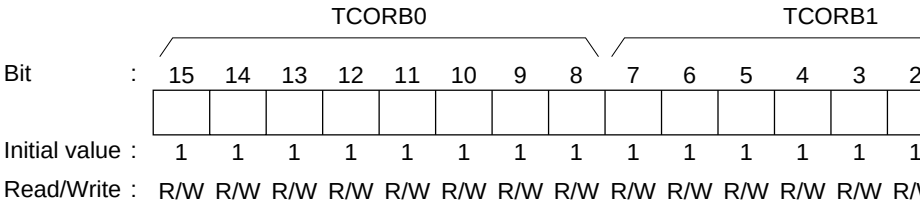
Compare Match Flag B

0	[Clearing conditions] • Cleared by reading CMFB when CMFB = 1, then writing 0 to CMFB • When the DTC is activated by a CMIB interrupt, while DISEL bit of MRB in DTC is 0.
1	[Setting condition] Set when TCNT matches TCORB

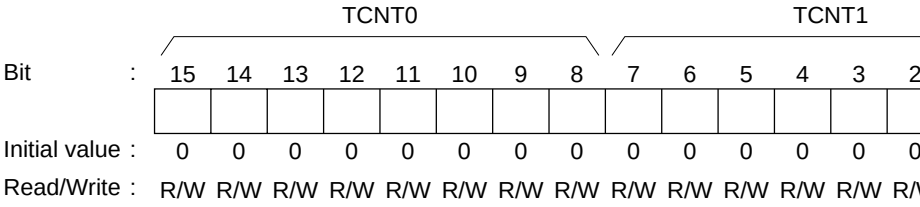
Note: * Only 0 can be written to bits 7 to 5, to clear these flags.

Read/Write : R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/

TCORB0—Time Constant Register B0 **H'FFB6** **8-Bit Time**
TCORB1—Time Constant Register B1 **H'FFB7** **8-Bit Time**



TCNT0—Timer Counter 0 **H'FFB8** **8-Bit Time**
TCNT1—Timer Counter 1 **H'FFB9** **8-Bit Time**



CKS2	CKS1	CKS0	Clock	(when φ
0	0	0	φ/2 (initial value)	25.6µs
		1	φ/64	819.2µs
	1	0	φ/128	1.6ms
		1	φ/512	6.6ms
1	0	0	φ/2048	26.2ms
		1	φ/8192	104.9m
	1	0	φ/32768	419.4m
		1	φ/131072	1.68s

Note: * The overflow period is the time from starts counting up from H'00 until ov

Timer Enable

0	TCNT is initialized to H'00 and halted
1	TCNT counts

Timer Mode Select

0	Interval timer mode: Sends the CPU an interval timer interrupt reque (WOVI) when TCNT overflows
1	Watchdog timer mode: Generates the $\overline{\text{WDTOVF}}$ signal when TCNT overflows

Overflow Flag

0	[Clearing condition] Cleared by reading TCSR when OVF = 1, then writing 0 to OVF
1	[Setting condition] Set when TCNT overflows from H'FF to H'00 in interval timer mode

The method for writing to TCSR is different from that for general registers to prevent inadvertent ov For details see section 11.2.4, Notes on Register Access.

Note: * Can only be written with 0 for flag clearing.

overwriting. For details, see section 11.2.4, Notes on Register Access.

Reset Select

0	Power-on reset
1	Manual reset

Reset Enable

0	Reset signal is not generated if TCNT overflows*
1	Reset signal is generated if TCNT overflows

Note: * The modules H8S/2345 Group are not reset, but T
and TCSR in WDT are reset.

Watchdog Timer Overflow Flag

0	[Clearing condition] Cleared by reading TCSR when WOVF = 1, then writing 0 to WO
1	[Setting condition] Set when TCNT overflows (changed from H'FF to H'00) during watchdog timer operation

Note: * Can only be written with 0 for flag clearing.

The method for writing to RSTCSR is different from that for general registers to pr
inadvertent overwriting. For details see section 11.2.4, Notes on Register Access

Counter Start

0	TCNTn count operation
1	TCNTn performs count

Note: If 0 is written to the CST bit during operation with the TIOC pin designated for c the counter stops but the TIOC pin output compare output level is retained. If is written to when the CST bit is cleared to 0, the pin output level will be chang the set initial output value.

TSYR—Timer Synchro Register **H'FFC1**

Bit :	7	6	5	4	3	2	1
	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1
Initial value :	0	0	0	0	0	0	0
Read/Write :	—	—	R/W	R/W	R/W	R/W	R/W

Timer Synchronization

0	TCNTn operates independently (TCNT pre clearing is unrelated to other channels)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronou is possible

Notes: 1. To set synchronous operation, the SYNC bits for at least two channels m be set to 1.
2. To set synchronous clearing, in addition to the SYNC bit , the TCNT clea source must also be set by means of bits CCLR2 to CCLR0 in TCR.

Program

0	Program mode cl
1	Transition to prog [Setting condition] When FWE = 1, S and PSU = 1

Erase

0	Erase mode cleared
1	Transition to erase mo [Setting condition] When FWE = 1, SWE and ESU = 1

Software Write Enable

0	Writes disabled
1	Writes enabled [Setting condition] When FWE = 1

Program-Verify

0	Program-verify mode cleare
1	Transition to program-verify [Setting condition] When FWE = 1 and SWE =

Erase-Verify

0	Erase-verify mode cleared
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1

Flash Write Enable

0	When a low level is input to the FWE pin (hardware-protected st
1	When a high level is input to the FWE pin

Note: * Determined by the state of the FWE pin.

Program Setup

0	Program setup clea
1	Program setup [Setting condition] When FWE = 1, an

Erase Setup

0	Erase setup cleared
1	Erase setup [Setting condition] When FWE = 1, and SW

Flash Memory Error

0	Flash memory is operating normally Flash memory program/erase protection (error protection) is dis [Clearing condition] Reset or hardware standby mode
1	An error has occurred during flash memory programming/erasin Flash memory program/erase protection (error protection) is ena [Setting condition] See section 17.10.3, Error Protection

Bit	:	7	6	5	4	3	2	1
EBR2		EB7	EB6	EB5	EB4	EB3	EB2	EB1
Initial value	:	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Flash Memory Erase Blocks

Block (Size)	Address
EB0 (1 kbyte)	H'000000 to H'0003FF
EB1 (1 kbyte)	H'000400 to H'0007FF
EB2 (1 kbyte)	H'000800 to H'000BFF
EB3 (1 kbyte)	H'000C00 to H'000FFF
EB4 (28 kbytes)	H'001000 to H'007FFF
EB5 (16 kbytes)	H'008000 to H'00BFFF
EB6 (8 kbytes)	H'00C000 to H'00DFFF
EB7 (8 kbytes)	H'00E000 to H'00FFFF
EB8 (32 kbytes)	H'010000 to H'017FFF
EB9 (32 kbytes)	H'018000 to H'01FFFF

0	0	0	Internal clock: counts on φ/
		1	Internal clock: counts on φ/
	1	0	Internal clock: counts on φ/
		1	Internal clock: counts on φ/
1	0	0	External clock: counts on T
		1	External clock: counts on T
	1	0	External clock: counts on T
		1	External clock: counts on T

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	*	Count at both edges

*: Don't care

Counter Clear

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.

Mode

0	0	0	0	Normal opera
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counti
			1	Phase counti
		1	0	Phase counti
			1	Phase counti
1	*	*	*	—

*

Notes: 1. MD3 is a reserved bit. should always be writt
2. Phase counting mode set for channels 0 and case, 0 should always MD2.

TGRA Buffer Operation

0	TGRA operates normally
1	TGRA and TGRC used together for buffer operation

TGRB Buffer Operation

0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

			1	is output compare register	Initial output is 0 output	0 output at compar	
		1	0			1 output at compar	
			1			Toggle output at co	
	1	0	0		Output disabled		
			1		Initial output is 1 output	0 output at compar	
		1	0			1 output at compar	
			1			Toggle output at co	
1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at risi	
			1			Input capture at fall	
		1	*			Input capture at bo	
	1	*	*		Capture input source is channel 1/count clock	Input capture at TC count-down	

TGR0B I/O Control

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 0 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 compare 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	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down*1

*: Don't care

Note: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000, and ϕ/1 is used as the TCNT1 count clock, this set input capture is not generated.

		1	0	register	0 output	1 output at compare m
			1			Toggle output at comp
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare m
		1	0			1 output at compare m
			1			Toggle output at comp
1	0	0	0	TGR0C is input capture register *1	Capture input source is TIOCC0 pin	Input capture at rising
			1			Input capture at falling
		1	*			Input capture at both e
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT count-down

Note: 1. When the BFA bit in TMDR0 is set to 1 and TGR0C is used a this setting is invalid and input capture/output compare is not

TGR0D I/O Control

0	0	0	0	TGR0D 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	TGR0D is input capture register *2	Capture input source is TIOCD0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down*1

*: Don't care

Notes: 1. When bits TPSC2 to TPSC0 in TCR1 are set to B'000, and ϕ/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.
2. 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 bu

0	Interrupt r by TGFA
1	Interrupt r by TGFA

TGR Interrupt Enab

0	Interrupt requ by TGFB bit
1	Interrupt requ by TGFB bit

TGR Interrupt Enable C

0	Interrupt requests (TGI TGFC bit disabled
1	Interrupt requests (TGI 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 disable
1	Interrupt requests (TCIV) by TCFV enable

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

	• When 0 is written to TGFA after TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGR as output compare register • When TCNT value is transferred input capture signal while TGRA as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt of MRB in DTC is 0 • When 0 is written to TGFB after reading T
1	[Setting conditions] • When TCNT = TGRB while TGRB is func output compare register • When TCNT value is transferred to TGRB capture signal while TGRB is functioning register

Input Capture/Output Compare Flag C

0	[Clearing conditions] • When DTC is activated by TGIC interrupt while DISEL DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as ou register • When TCNT value is transferred to TGRC by input cap 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 M is 0 • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output co • When TCNT value is transferred to TGRD by input capture sig 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.

TGR0A—Timer General Register 0A	**H'FFD8**
TGR0B—Timer General Register 0B	**H'FFDA**
TGR0C—Timer General Register 0C	**H'FFDC**
TGR0D—Timer General Register 0D	**H'FFDE**

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

0	0	0	Internal clock: counts on φ/1
		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/16
		1	Internal clock: counts on φ/64
1	0	0	External clock: counts on TC
		1	External clock: counts on TC
	1	0	Internal clock: counts on φ/25
		1	Counts on TCNT2 overflow/u

Note: This setting is ignored when channe counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	*	Count at both edges

*: Don't care

Note: This setting is ignored when channel 1 is in phase counting mode.

Counter Clear

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*

Note: * Synchronous operating setting is performed by setting the SYNC bit in TSYR to 1.

0	0	0	0	Normal op
			1	Reserved
		1	0	PWM mod
			1	PWM mod
	1	0	0	Phase cou
			1	Phase cou
		1	0	Phase cou
			1	1Phase co
1	*	*	*	—

Note: MD3 is a reserved bit.
it should always be wri

			1	is output compare register	Initial output is 0 output	0 output at compare
		1	0			1 output at compare
			1			Toggle output at com
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare
		1	0			1 output at compare
			1			Toggle output at com
1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at risin
			1			Input capture at fallin
		1	*			Input capture at both
	1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at gene channel 0/TGR0A co input capture

TGR1B I/O Control

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 is TIOCB1 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at generation of TGR0B compare match/input capture

*: Don't care

	by TGFA bit disa
1	Interrupt request by TGFA bit ena

TGR Interrupt Enable B

0	Interrupt requests (T by TGFB bit disable
1	Interrupt requests (T by TGFB bit enable

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

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

	DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGR as output compare register • When TCNT value is transferred input capture signal while TGRA as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrup bit of MRB in DTC is 0 • When 0 is written to TGFB after reading
1	[Setting conditions] • When TCNT = TGRB while TGRB is fur output compare register • When TCNT value is transferred to TGF capture signal while TGRB is functionin 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.

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase co
mode or when performing overflow/underflow counting on another chan
other cases it functions as an up-counter.

TGR1A—Timer General Register 1A **H'FFE8**
TGR1B—Timer General Register 1B **H'FFEA**

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

0	0	0	Internal clock: counts on φ/1
		1	Internal clock: counts on φ/4
	1	0	Internal clock: counts on φ/1
		1	Internal clock: counts on φ/6
1	0	0	External clock: counts on TC
		1	External clock: counts on TC
	1	0	External clock: counts on TC
		1	Internal clock: counts on φ/1

Note: This setting is ignored when channe counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	*	Count at both edges

*: Don't care

Note: This setting is ignored when channel 2 is in phase counting mode.

Counter Clear

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*

Note: * Synchronous operating setting is performed by setting the SYNC bit TSYR to 1.

Mode

0	0	0	0	Normal op
			1	Reserved
		1	0	PWM mo
			1	PWM mo
	1	0	0	Phase co
			1	Phase co
		1	0	Phase co
			1	Phase co
1	*	*	*	—

Note: MD3 is a reserved bit
it should always be w

			1	is output compare register	Initial output is 0 output	0 output at compare
		1	0			1 output at compare
			1			Toggle output at co
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare
		1	0			1 output at compare
			1			Toggle output at co
1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at risi
			1			Input capture at falli
		1	*			Input capture at both

TGR2B I/O Control

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 is TIOCB2 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

0	Interrupt re by TGFA bi
1	Interrupt re by TGFA bi

TGR Interrupt Enable

0	Interrupt reques by TGFB bit dis
1	Interrupt reques by TGFB bit ena

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

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC • When 0 is written to TGFA after TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA functioning as output compare • When TCNT value is transferred input capture signal while TGRA as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt 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 as capture signal while TGRB is functioning as 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.

Up/down-counter*

Note: * This timer counter can be used as an up/down-counter only in phase counti mode or when performing overflow/underflow counting on another channel. other cases it functions as an up-counter.

TGR2A—Timer General Register 2A **H'FFF8**
TGR2B—Timer General Register 2B **H'FFFA**

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

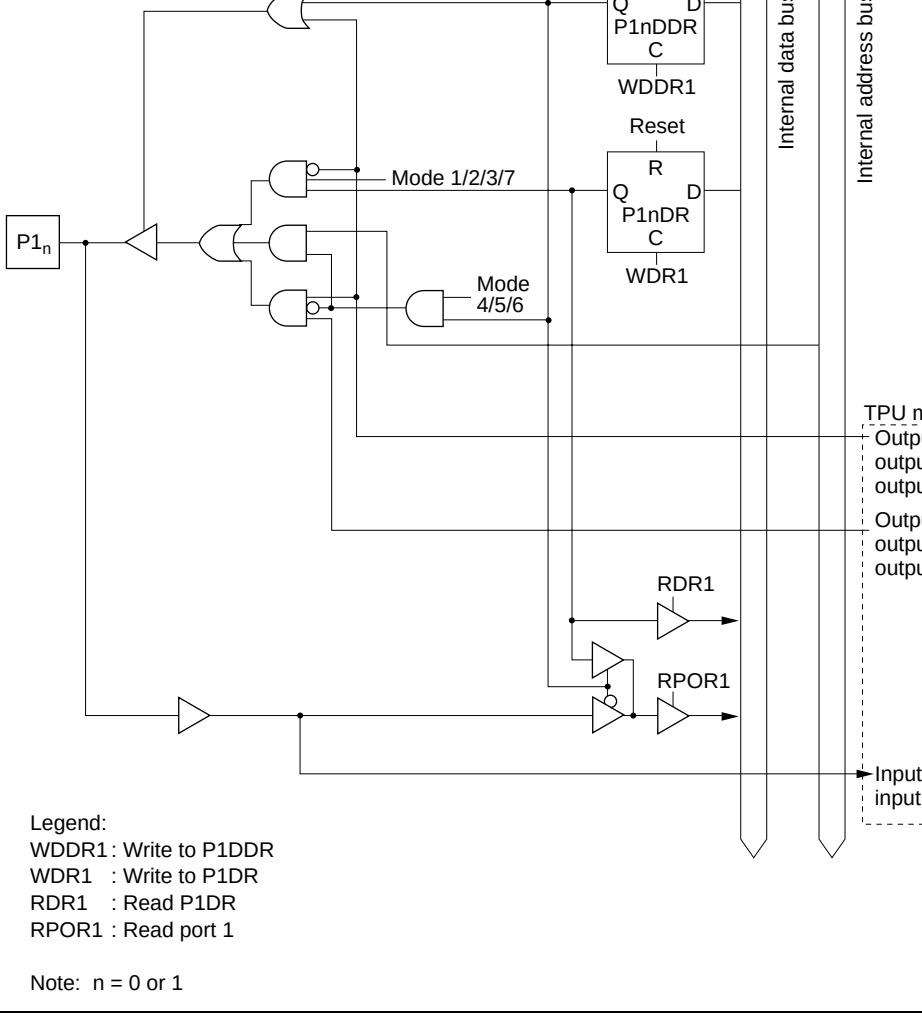


Figure C.1 (a) Port 1 Block Diagram (Pins $P1_0$ and $P1_1$)

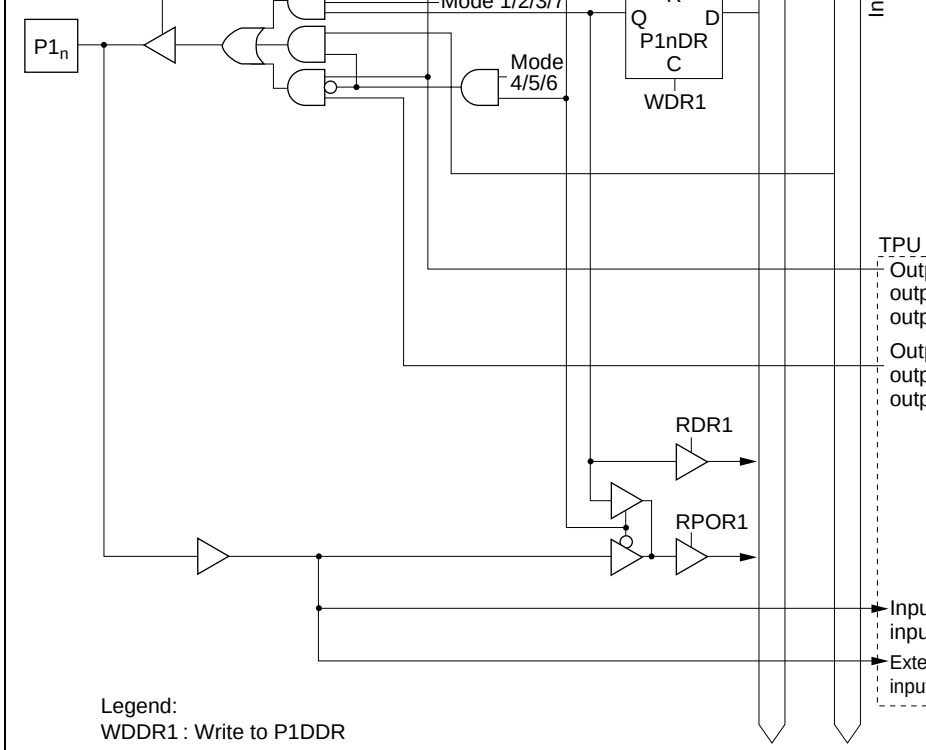


Figure C.1 (b) Port 1 Block Diagram (Pins $P1_2$ and $P1_3$)

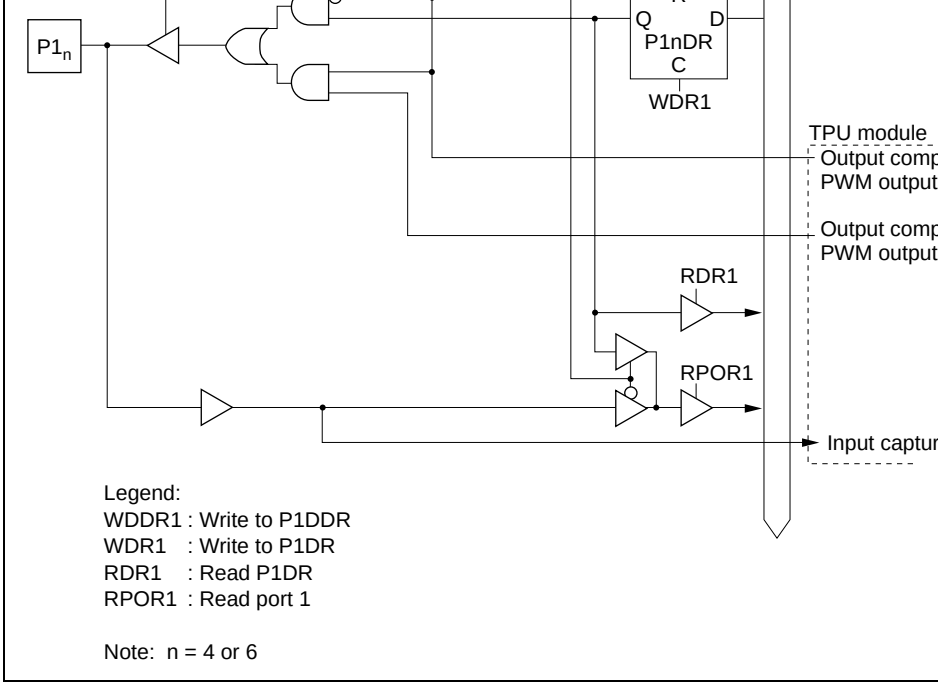
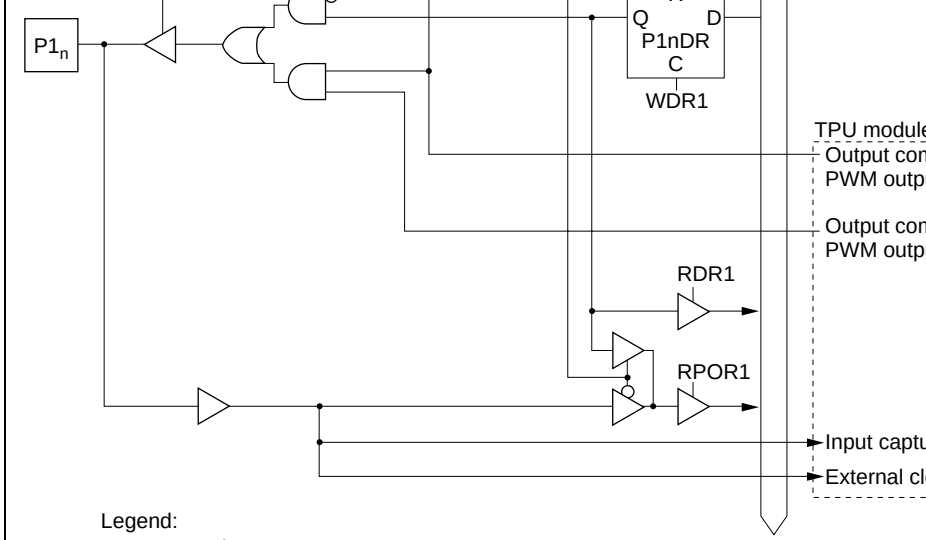


Figure C.1 (c) Port 1 Block Diagram (Pins $P1_4$ and $P1_6$)



Legend:
WDDR1 : Write to P1DDR
WDR1 : Write to P1DR
RDR1 : Read P1DR
RPOR1 : Read port 1

Note: n = 5 or 7

Figure C.1 (d) Port 1 Block Diagram (Pins $P1_5$ and $P1_7$)

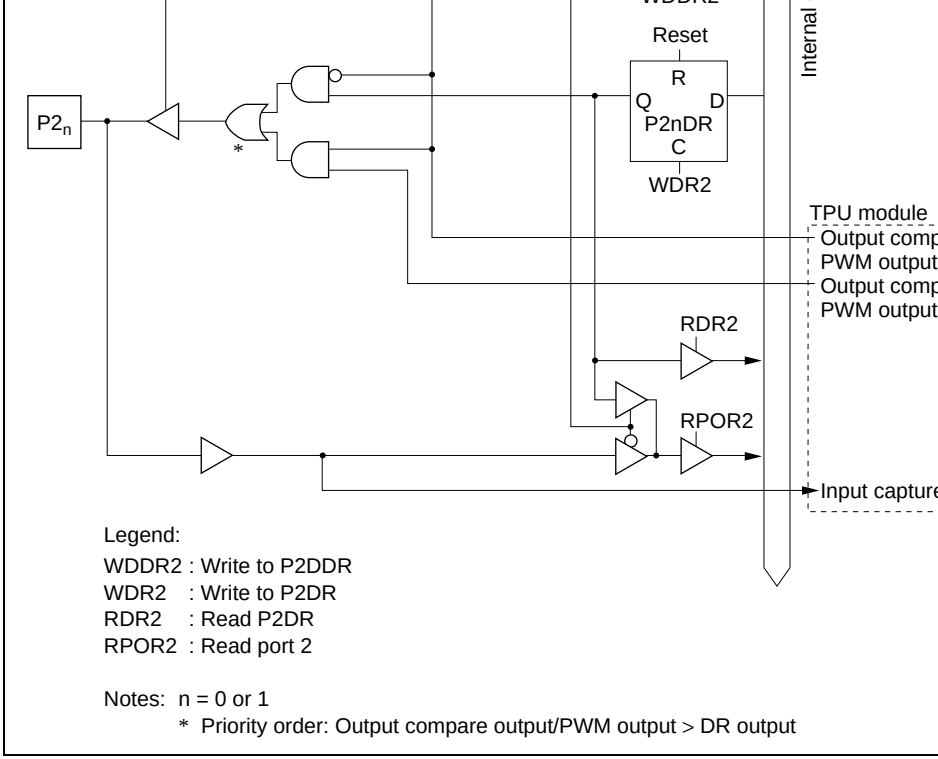
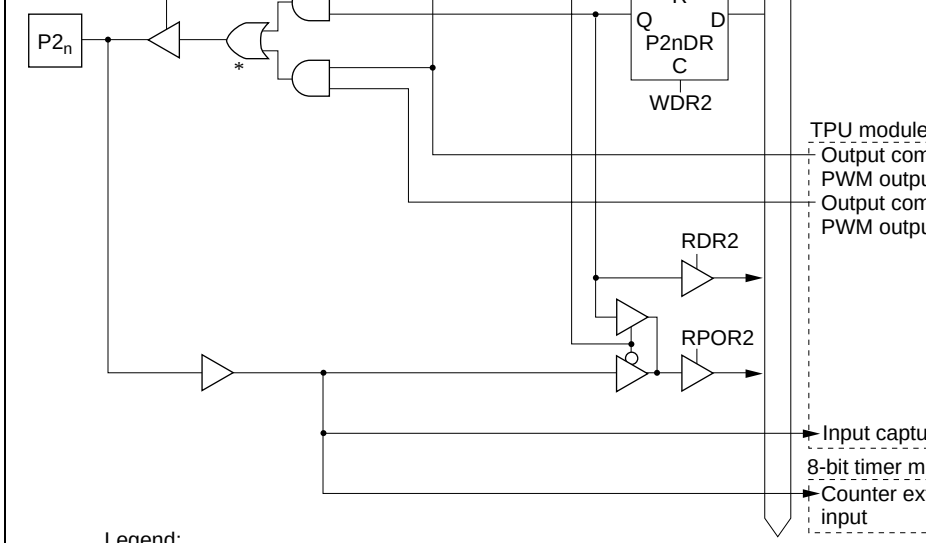


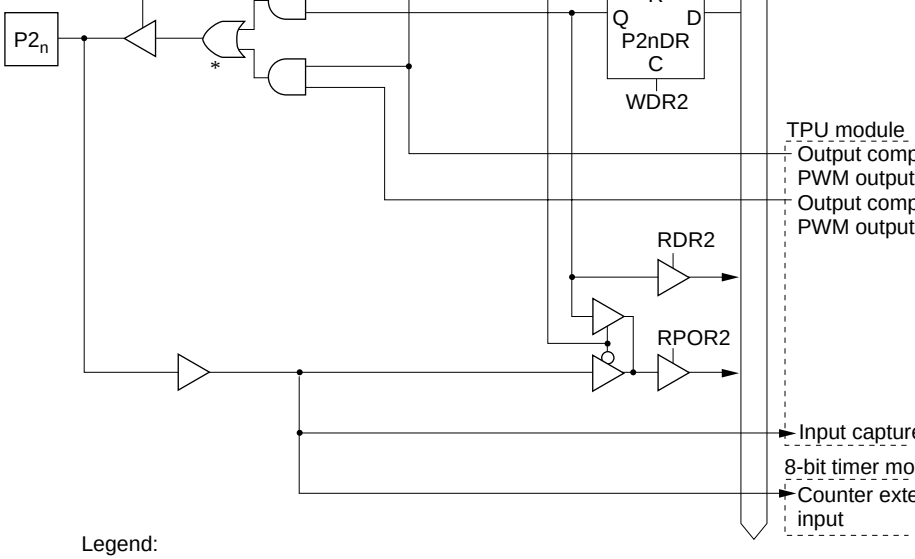
Figure C.2 (a) Port 2 Block Diagram (Pins $P2_0$ and $P2_1$)



Legend:
WDDR2 : Write to P2DDR
WDR2 : Write to P2DR
RDR2 : Read P2DR
RPOR2 : Read port 2

Notes: n = 2 or 4
* Priority order: Output compare output/PWM output > DR output

Figure C.2 (b) Port 2 Block Diagram (Pins $P2_2$ and $P2_4$)

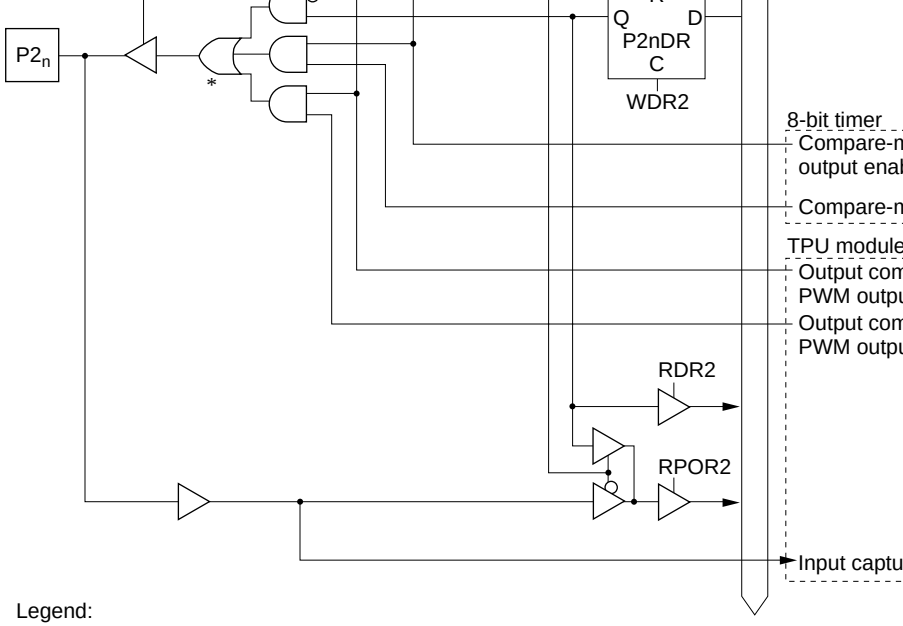


Legend:

WDDR2 : Write to P2DDR
WDR2 : Write to P2DR
RDR2 : Read P2DR
RPOR2 : Read port 2

Notes: n = 3 or 5
* Priority order: Output compare output/PWM output > DR output

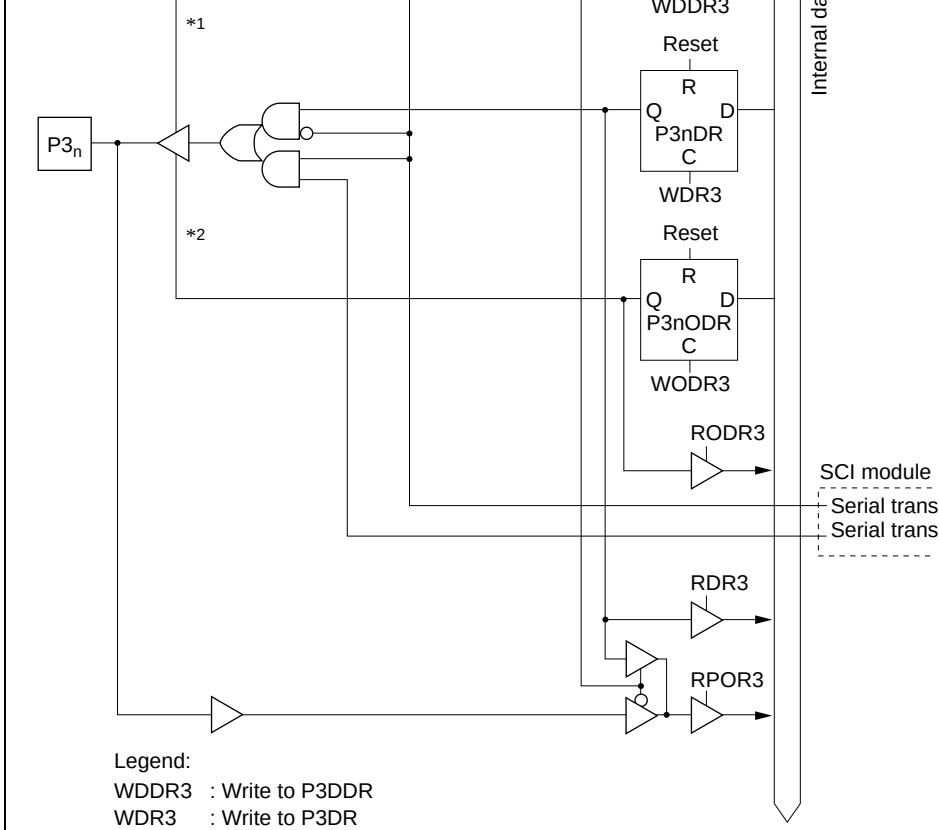
Figure C.2 (c) Port 2 Block Diagram (Pins $P2_3$ and $P2_5$)



Legend:
WDDR2 : Write to P2DDR
WDR2 : Write to P2DR
RDR2 : Read P2DR
RPOR2 : Read port 2

Notes: n = 6 or 7
* Priority order: Output compare output/PWM output > compare match output > DR ou

Figure C.2 (d) Port 2 Block Diagram (Pins $P2_6$ and $P2_7$)



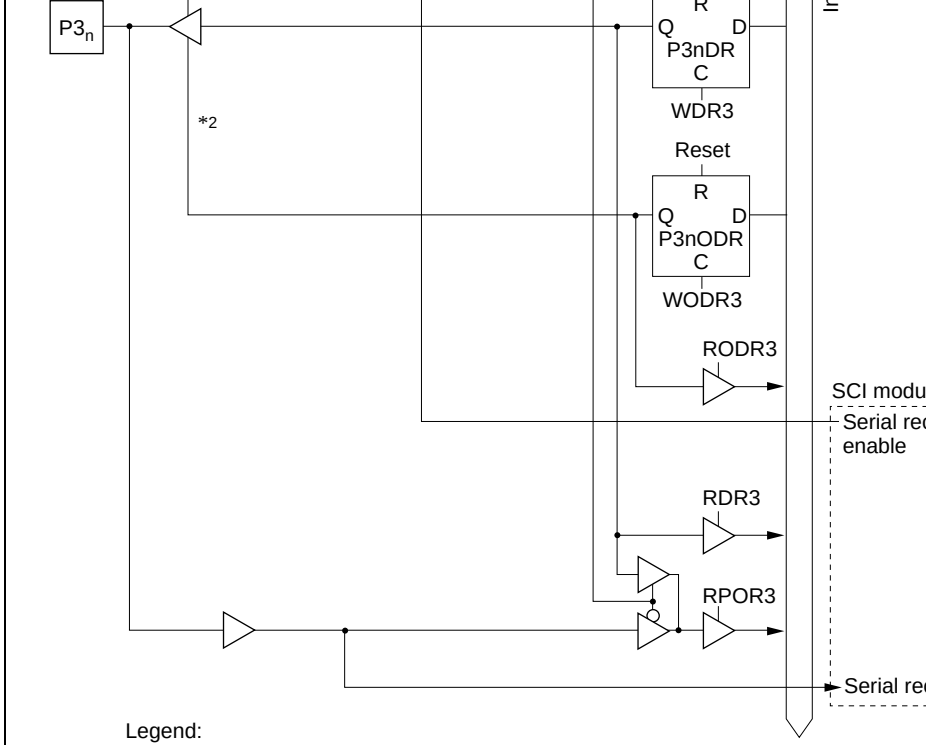
Legend:

WDDR3 : Write to P3DDR
WDR3 : Write to P3DR
WODR3 : Write to P3ODR
RDR3 : Read P3DR
RPOR3 : Read port 3
RODR3 : Read P3ODR

Notes: n = 0 or 1

1. Output enable signal
2. Open drain control signal

Figure C.3 (a) Port 3 Block Diagram (Pins $P3_0$ and $P3_1$)



Legend:

WDDR3 : Write to P3DDR
WDR3 : Write to P3DR
WODR3 : Write to P3ODR
RDR3 : Read P3DR
RPOR3 : Read port 3
RODR3 : Read P3ODR

Notes: n = 2 or 3
1. Output enable signal
2. Open drain control signal

Figure C.3 (b) Port 3 Block Diagram (Pins $P3_2$ and $P3_3$)

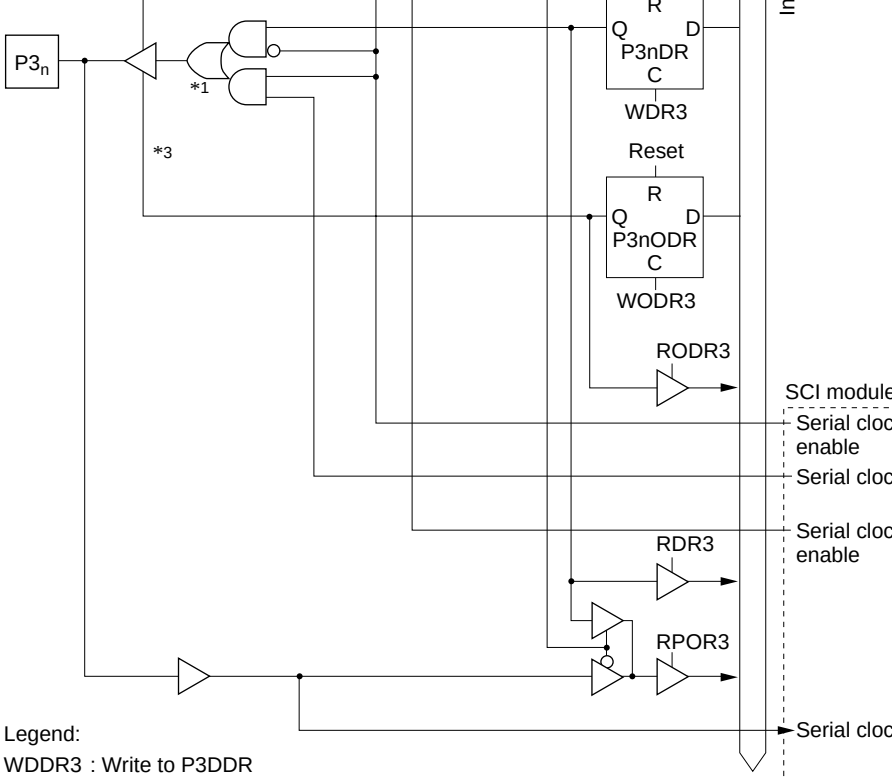
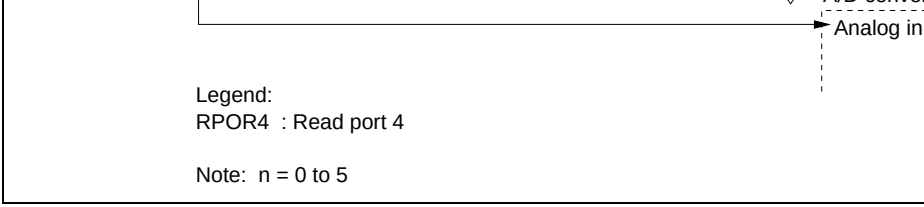


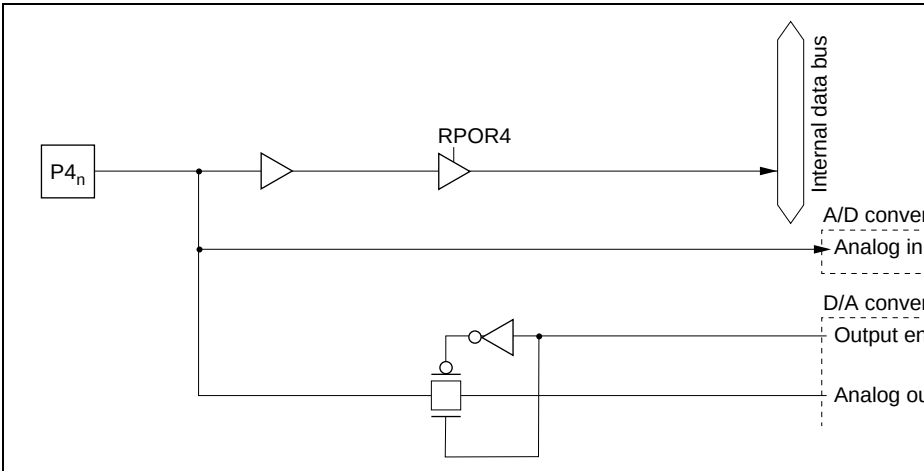
Figure C.3 (c) Port 3 Block Diagram (Pins $P3_4$ and $P3_5$)



Legend:
RPOR4 : Read port 4

Note: n = 0 to 5

Figure C.4 (a) Port 4 Block Diagram (Pins $P4_0$ to $P4_5$)



Legend:
RPOR4 : Read port 4

Note: n = 6 or 7

Figure C.4 (b) Port 4 Block Diagram (Pins $P4_6$ and $P4_7$)

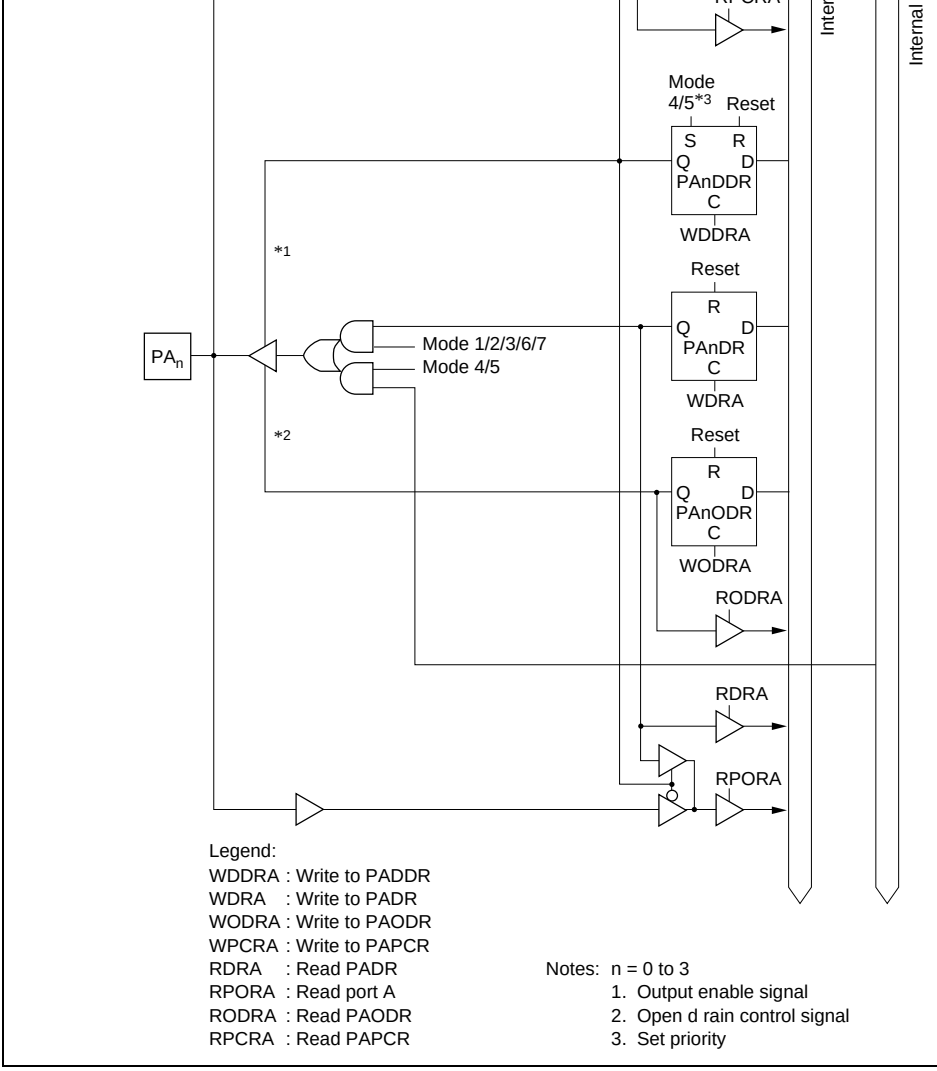


Figure C.5 Port A Block Diagram (Pins PA_0 to PA_3)

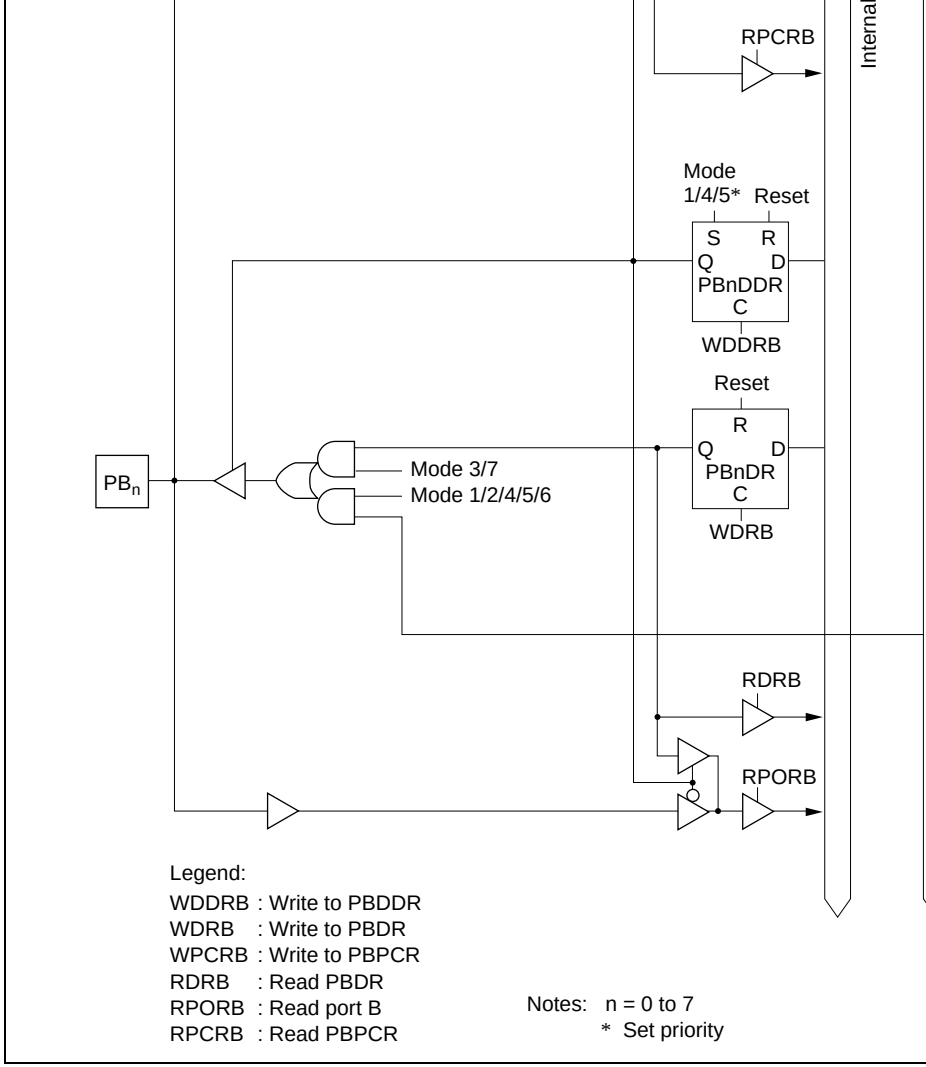


Figure C.6 Port B Block Diagram (Pin PB_n)

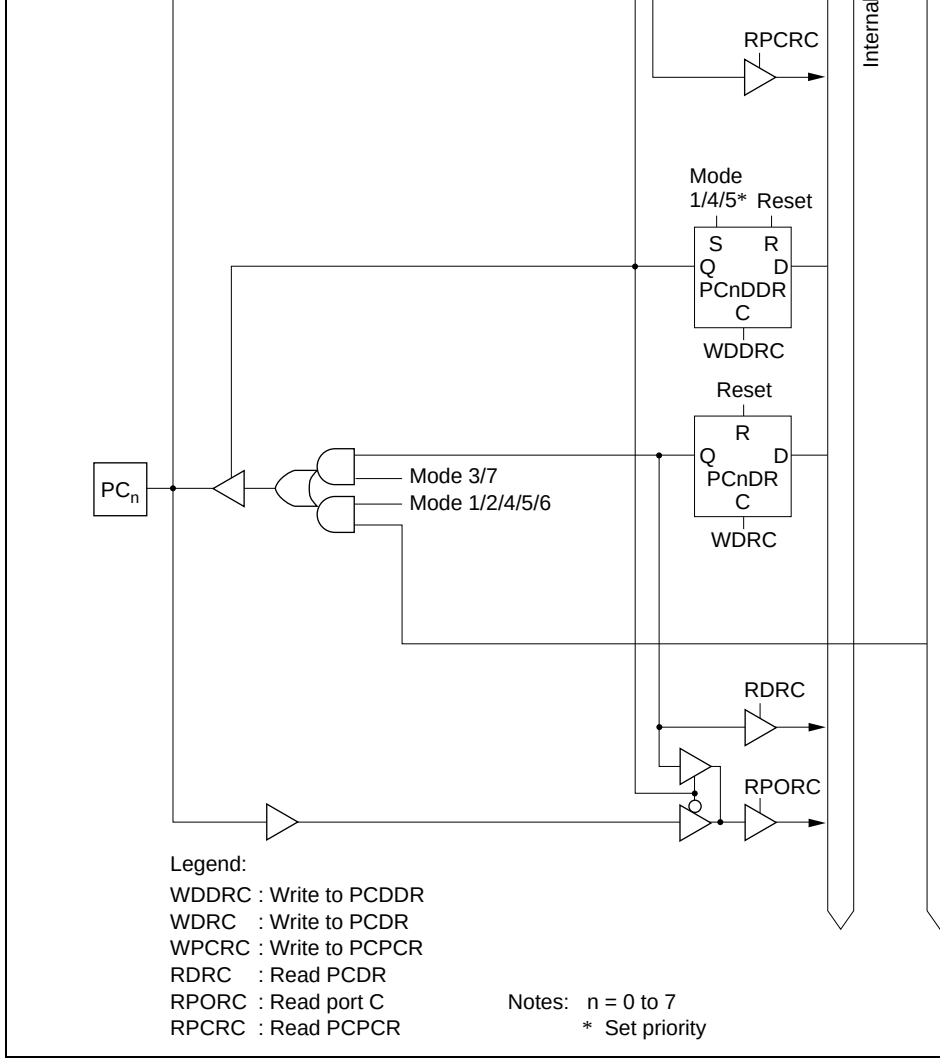


Figure C.7 Port C Block Diagram (Pin PC_n)

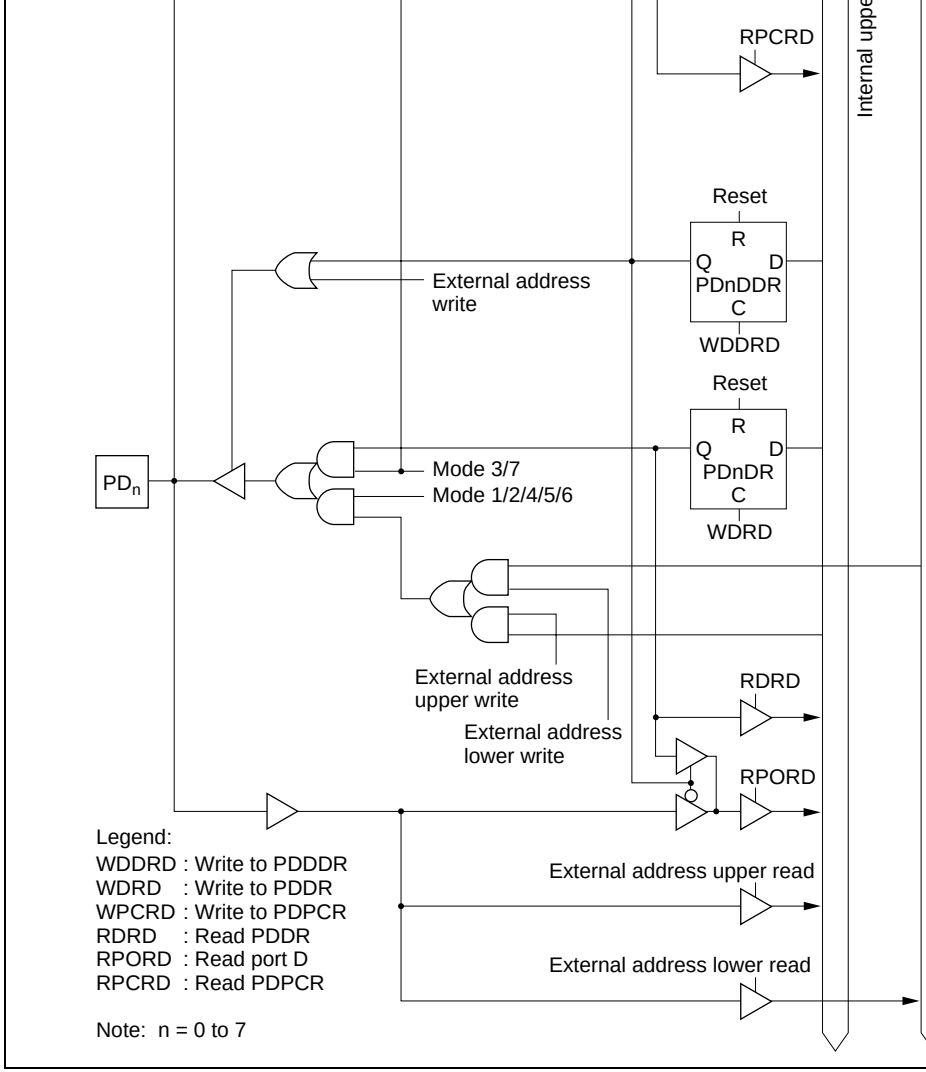


Figure C.8 Port D Block Diagram (Pin PD_n)

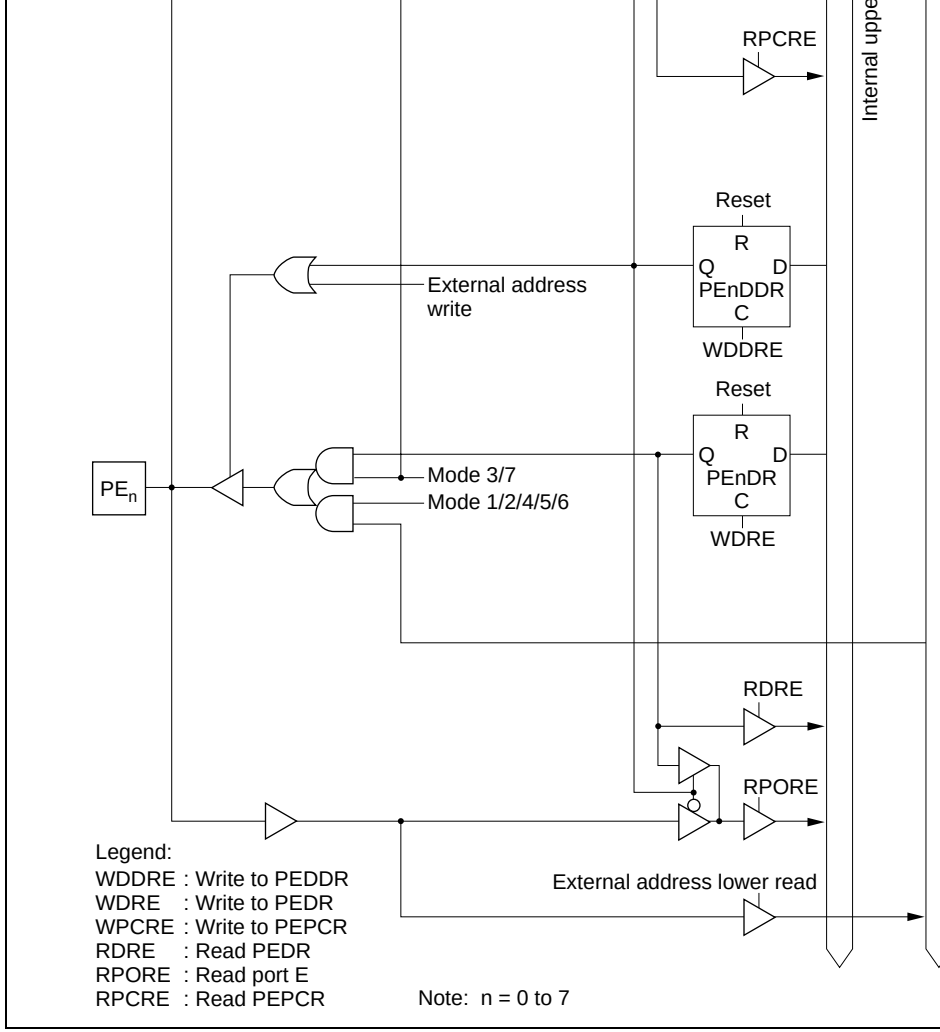


Figure C.9 Port E Block Diagram (Pin PE_n)

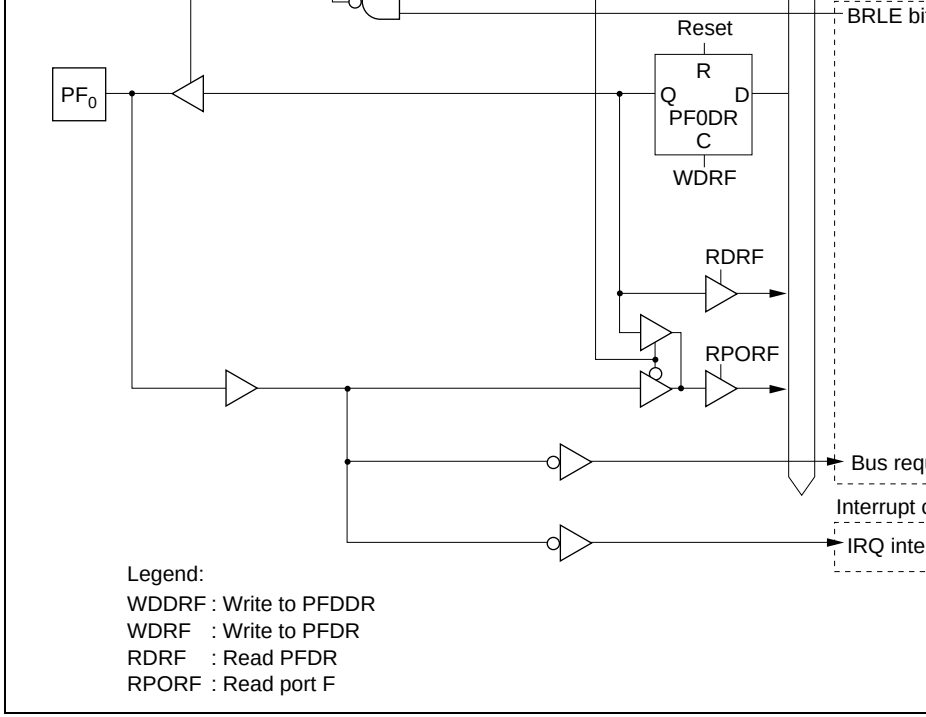


Figure C.10 (a) Port F Block Diagram (Pin PF_0)

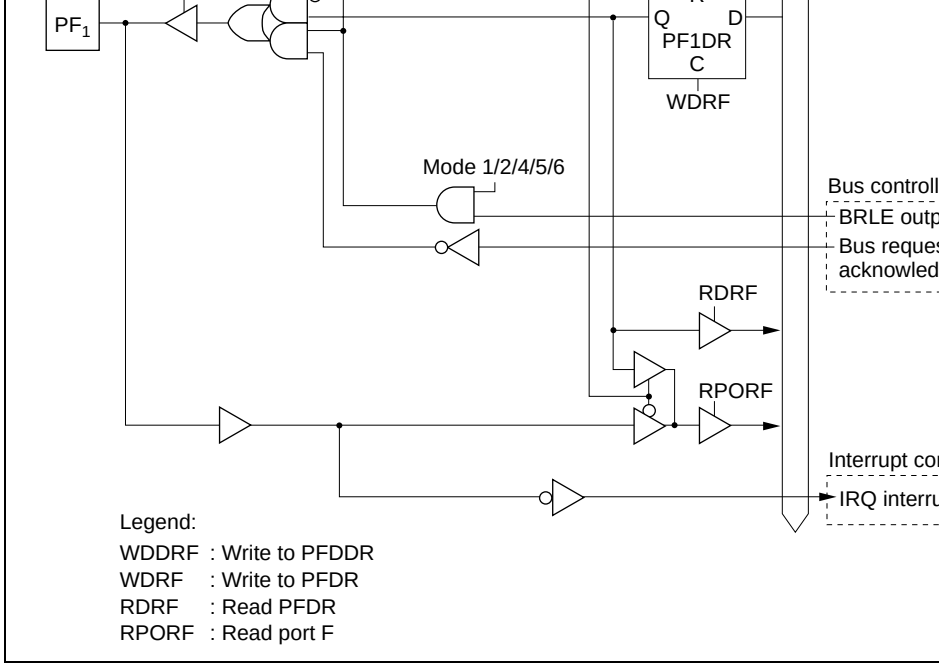


Figure C.10 (b) Port F Block Diagram (Pin PF_1)

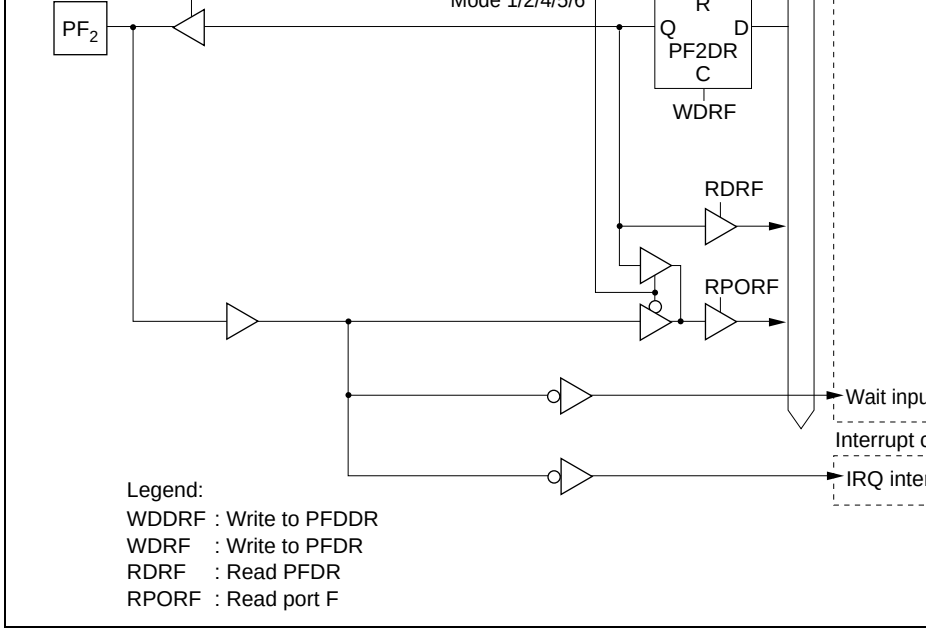


Figure C.10 (c) Port F Block Diagram (Pin PF_2)

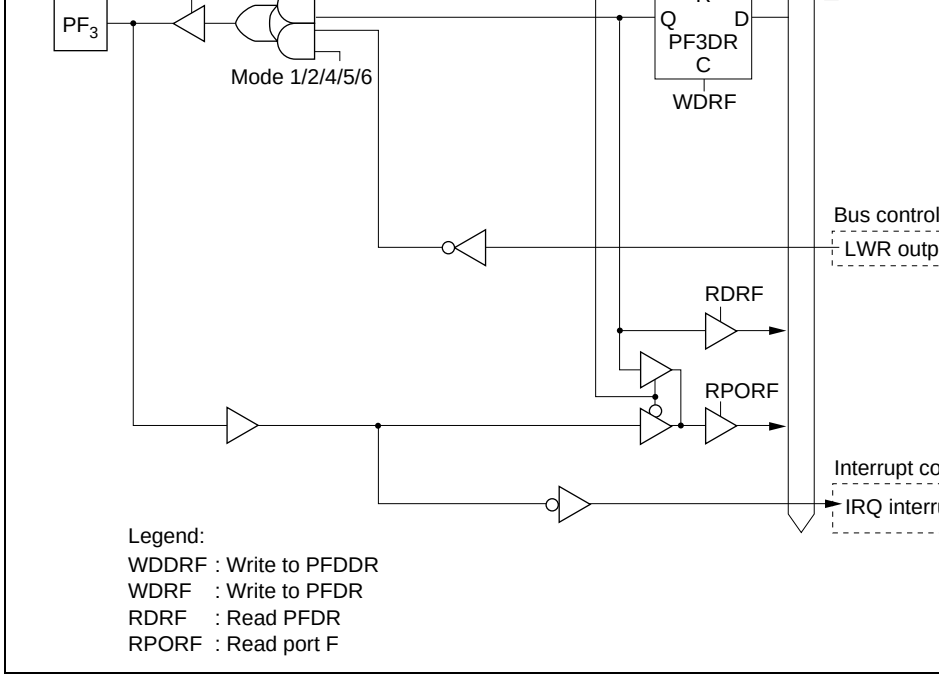


Figure C.10 (d) Port F Block Diagram (Pin PF_3)

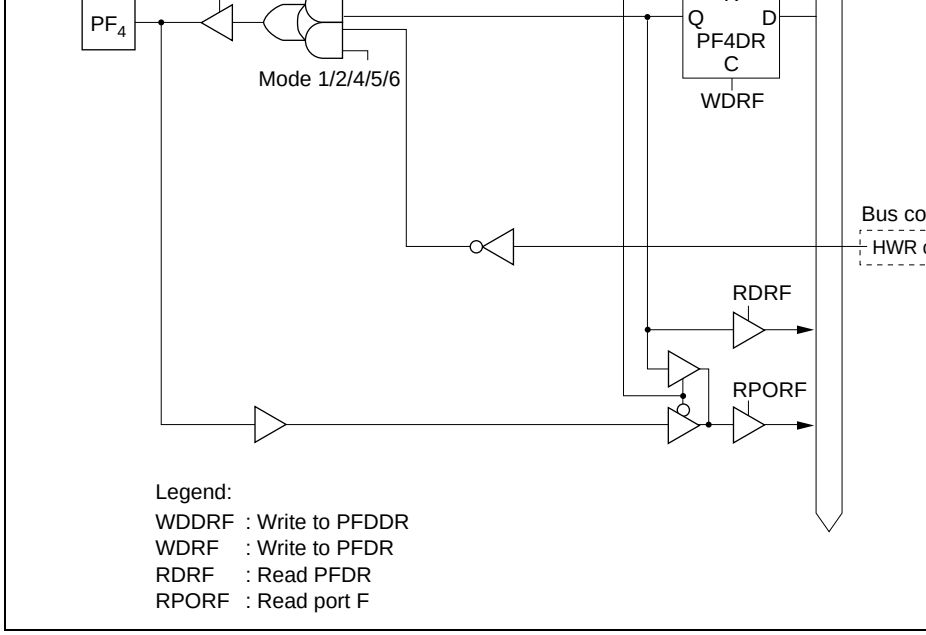


Figure C.10 (e) Port F Block Diagram (Pin PF_4)

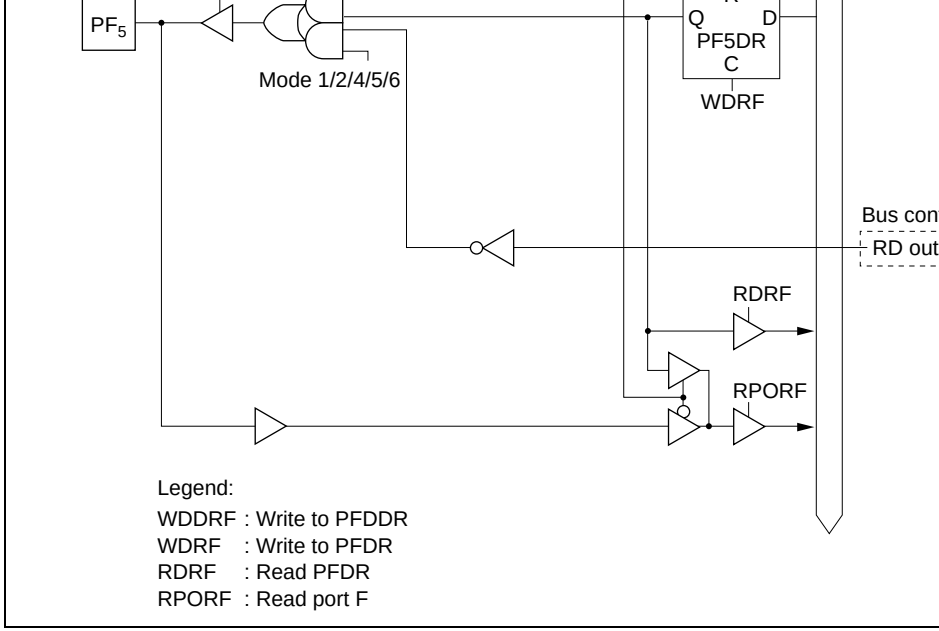


Figure C.10 (f) Port F Block Diagram (Pin PF_5)

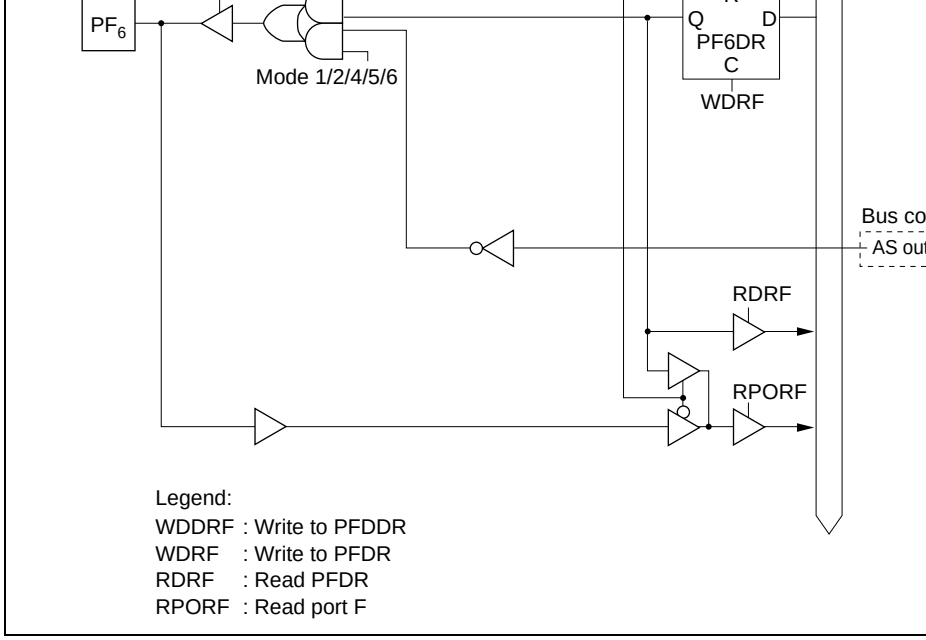


Figure C.10 (g) Port F Block Diagram (Pin PF_6)

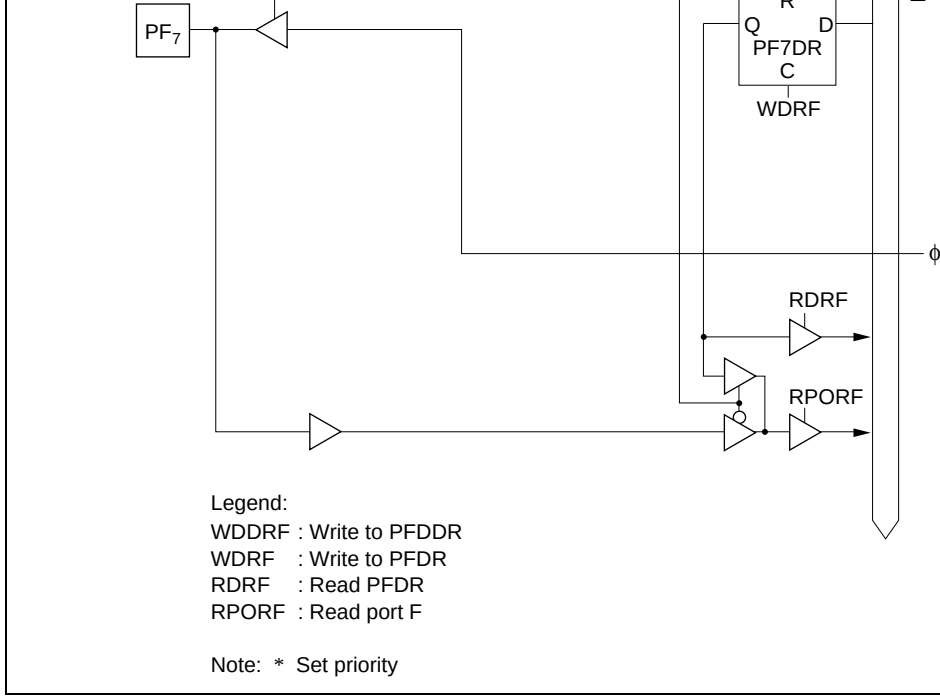


Figure C.10 (h) Port F Block Diagram (Pin PF_7)

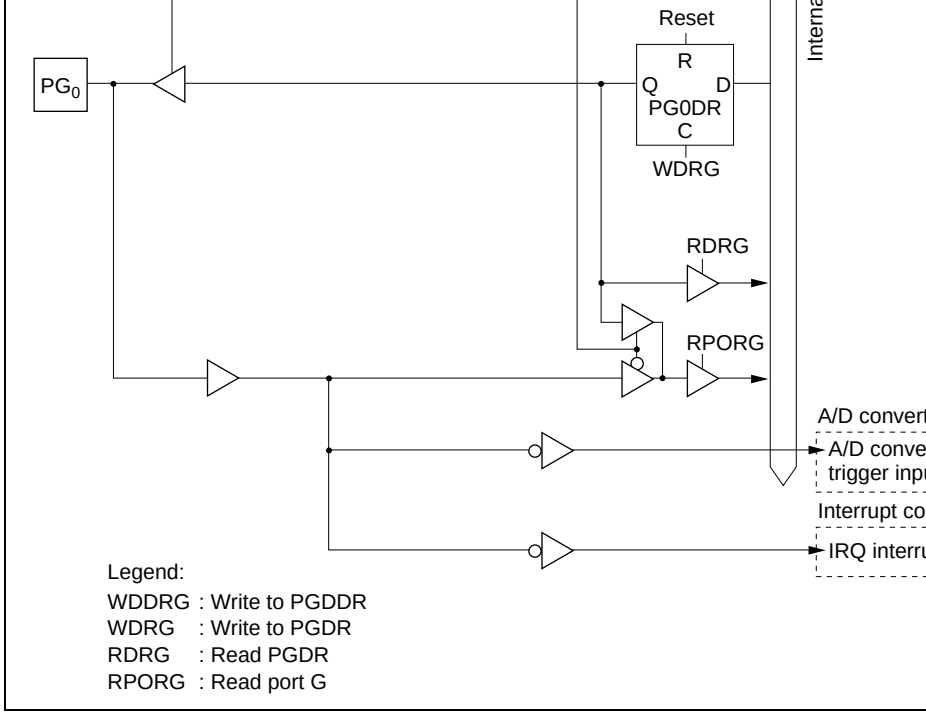


Figure C.11 (a) Port G Block Diagram (Pin PG_0)

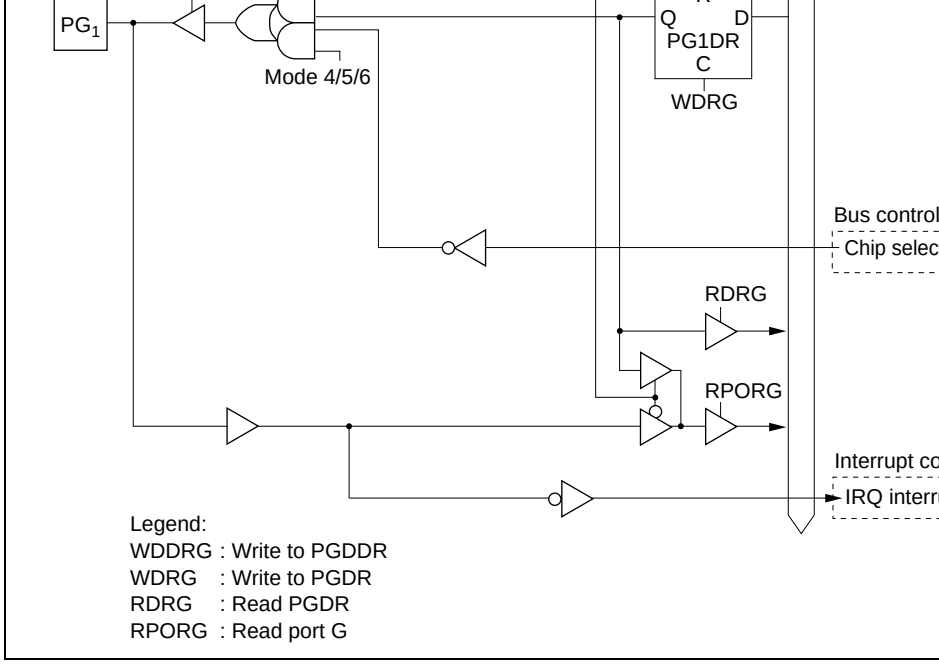
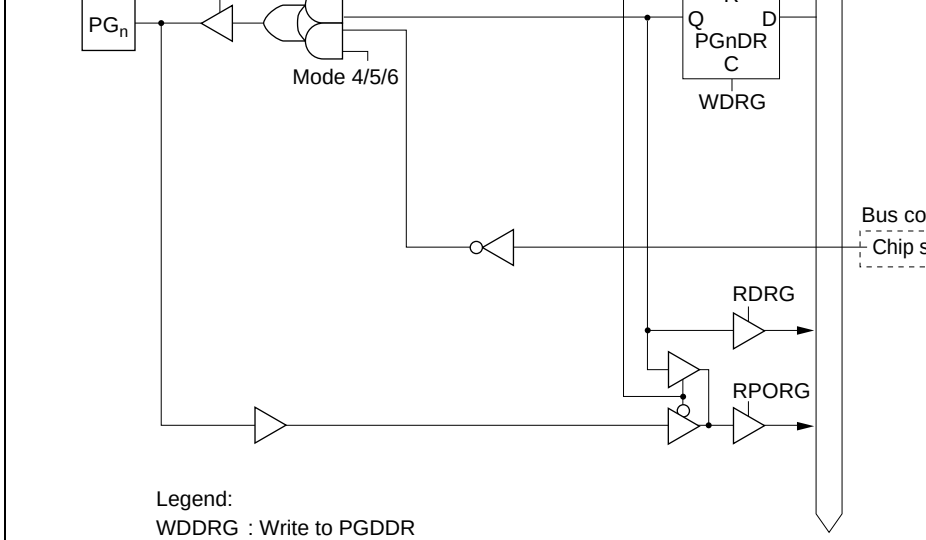


Figure C.11 (b) Port G Block Diagram (Pin PG_1)



Legend:
WDDRG : Write to PGDDR
WDRG : Write to PGDR
RDRG : Read PGDR
RPORG : Read port G

Note: n = 2 or 3

Figure C.11 (c) Port G Block Diagram (Pins PG_2 and PG_3)

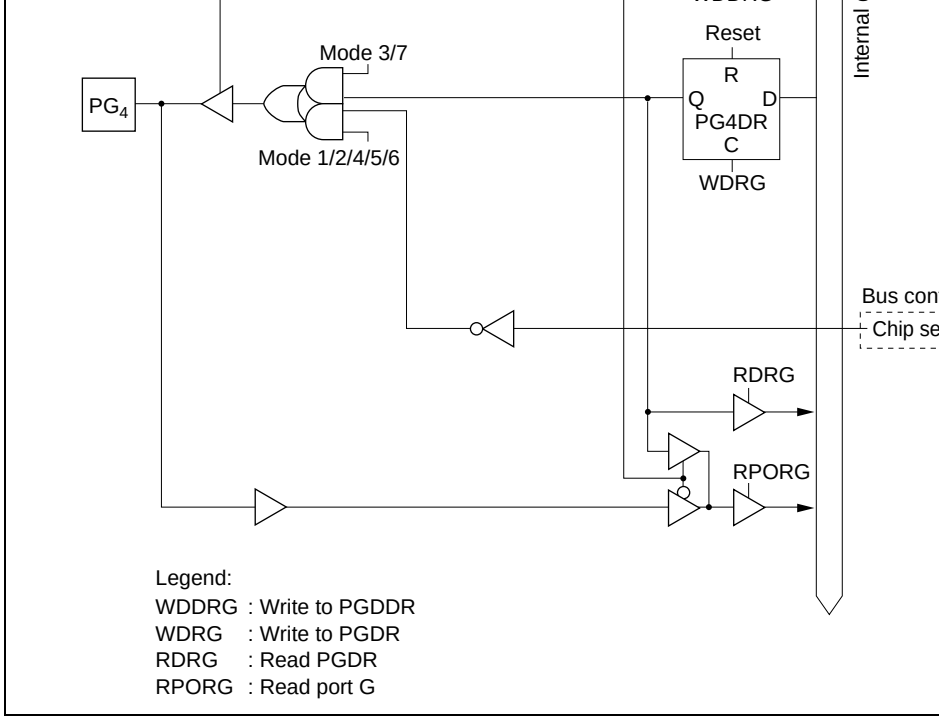


Figure C.11 (d) Port G Block Diagram (Pin PG_4)

Port Name Pin Name	Operating Mode	On Reset	Manual Reset	Standby Mode	Standby Mode	Release State
$P1_7$/TIOCB2/ TCLKD	1 to 7	T	kept	T	kept	kept
$P1_6$/TIOCA2						
$P1_5$/TIOCB1/ TCLKC						
$P1_4$/TIOCA1						
$P1_3$/TIOCD0/ TCLKB/A_{23}	1 to 3, 7	T	kept	T	kept	kept
	4 to 6	T	kept	T	[DDR · OPE = 0] T	T
$P1_2$/TIOCC0/ TCLKA/A_{22}					[DDR · OPE = 1] kept	
$P1_1$/TIOCB0/ A_{21}						
$P1_0$/TIOCA0/ A_{20}						
Port 2	1 to 7	T	kept	T	kept	kept
Port 3	1 to 7	T	kept	T	kept	kept
$P4_7$/DA1	1 to 7	T	T	T	[DAOE1 = 1] kept	kept
					[DAOE1 = 0] T	
$P4_6$/DA0	1 to 7	T	T	T	[DAOE0 = 1] kept	kept
					[DAOE0 = 0] T	
$P4_5$ to $P4_0$	1 to 7	T	T	T	T	T

	6	T	kept	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T
Port B	1, 4, 5	L	kept	T	[OPE = 0] T [OPE = 1] kept	T
	2, 6	T	kept	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T
	3, 7	T	kept	T	kept	kept
Port C	1, 4, 5	L	kept	T	[OPE = 0] T [OPE = 1] kept	T
	2, 6	T	kept	T	[DDR · OPE = 0] T [DDR · OPE = 1] kept	T
	3, 7	T	kept	T	kept	kept
Port D	1, 2, 4 to 6	T	T*	T	T	T
	3, 7	T	kept	T	kept	kept
Port E	1, 2, 4 to 6 (8 bit bus)	T	kept	T	kept	kept
	1, 2, 4 to 6 (16 bit bus)	T	T*	T	T	T
	3, 7	T	kept	T	kept	kept

	3, 7	T	kept	T	[DDR = 0] Input port [DDR = 1] H	[DDR = 0] Input port [DDR = 1] Clock output
$PF_6/\overline{AS}$ $PF_5/\overline{RD}$ $PF_4/\overline{HWR}$	1, 2, 4 to 6	H	H*	T	[OPE = 0] T [OPE = 1] H	T
$PF_3/\overline{LWR}/\overline{IRQ3}$	3, 7	T	kept	T	kept	kept
$PF_2/\overline{WAIT}/\overline{IRQ2}$	1, 2, 4 to 6	T	[WAITE = 0] kept [WAITE = 1] T	T	[WAITE = 0] kept [WAITE = 1] T	[WAITE = 0] kept [WAITE = 1] T
	3, 7	T	kept	T	kept	kept
$PF_1/\overline{BACK}/\overline{IRQ1}$	1, 2, 4 to 6	T	[BRLE = 0] kept [BRLE = 1] $\overline{BACK}$	T	[BRLE = 0] kept [BRLE = 1] H	L
	3, 7	T	kept	T	kept	kept
$PF_0/\overline{BREQ}/\overline{IRQ0}$	1, 2, 4 to 6	T	[BRLE = 0] kept [BRLE = 1] $\overline{BREQ}$	T	[BRLE = 0] kept [BRLE = 1] T	T
	3, 7	T	kept	T	kept	kept
$PG_4/\overline{CS0}$	1, 4, 5	H	[DDR = 0] T [DDR = 1] H*	T	[DDR · OPE = 0] T [DDR · OPE = 1] H	T
	2, 6	T				
	3, 7	T	kept	T	kept	kept

$\overline{PG_0}$/ADTRG/ $\overline{IRQ6}$	1 to 3, 7	T	kept	T	kept	kept
	4 to 6	T	kept	T	kept	T

Legend:

H : High level
L : Low level
T : High impedance
kept : 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: * Indicates the state after completion of the executing bus cycle.

STBY signal goes low (delay from STBY low to RES high: 0 ns or more).

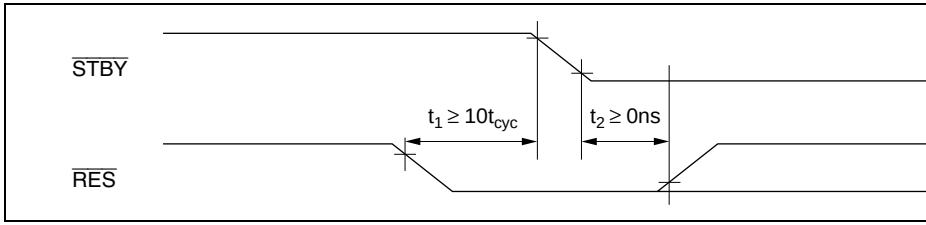


Figure E.1 Timing of Transition to Hardware Standby Mode

(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM not need to be retained, RES does not have to be driven low as in (1).

Timing of Recovery from Hardware Standby Mode

Drive the RES signal low and the NMI signal high approximately 100 ns or more befo goes high to execute a power-on reset.

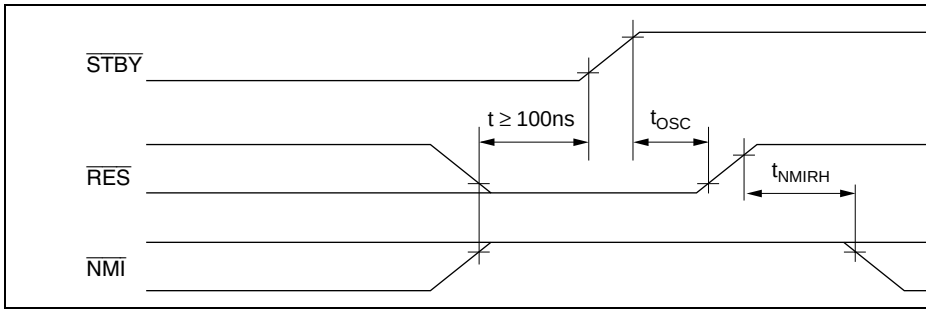


Figure E.2 Timing of Recovery from Hardware Standby Mode

			HD6432345(***)FA	100-pin QFP (FP
	ZTAT™	HD6472345	HD6472345TE	100-pin TQFP (T
			HD6472345TF	100-pin TQFP (T
			HD6472345F	100-pin QFP (FP
			HD6472345FA	100-pin QFP (FP
	F-ZTAT™	HD64F2345	HD64F2345TE	100-pin TQFP (T
			HD64F2345TF	100-pin TQFP (T
			HD64F2345F	100-pin QFP (FP
			HD64F2345FA	100-pin QFP (FP
H8S/2344	Mask ROM	HD6432344	HD6432344(***)TE	100-pin TQFP (T
			HD6432344(***)TF	100-pin TQFP (T
			HD6432344(***)F	100-pin QFP (FP
			HD6432344(***)FA	100-pin QFP (FP
H8S/2343	Mask ROM	HD6432343	HD6432343(***)TE	100-pin TQFP (T
			HD6432343(***)TF	100-pin TQFP (T
			HD6432343(***)F	100-pin QFP (FP
			HD6432343(***)FA	100-pin QFP (FP
H8S/2341	Mask ROM	HD6432341	HD6432341(***)TE	100-pin TQFP (T
			HD6432341(***TF	100-pin TQFP (T
			HD6432341(***)F	100-pin QFP (FP
			HD6432341(***)FA	100-pin QFP (FP
H8S/2340	ROMless	HD6412340	HD6412340TE	100-pin TQFP (T
			HD6412340TF	100-pin TQFP (T
			HD6412340F	100-pin QFP (FP
			HD6412340FA	100-pin QFP (FP

Note: (***) indicates the ROM code.

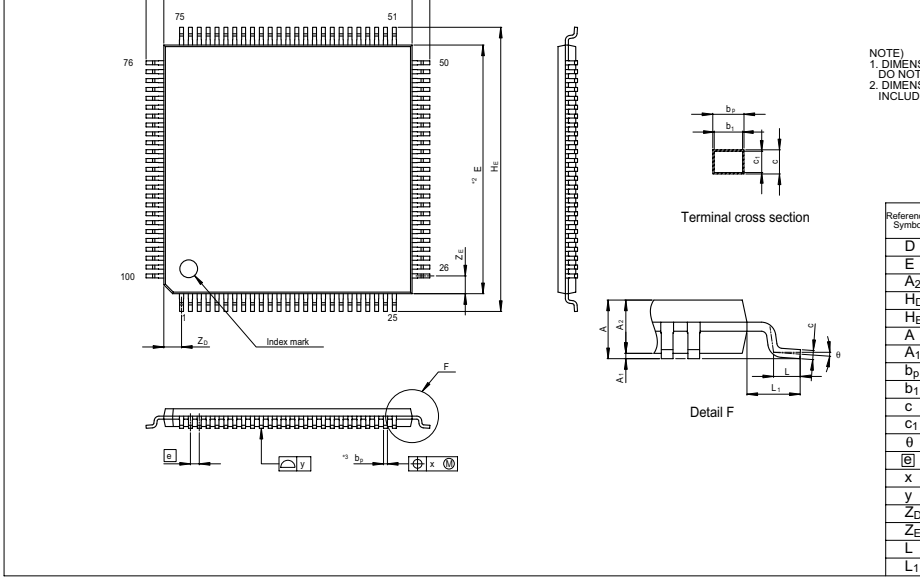
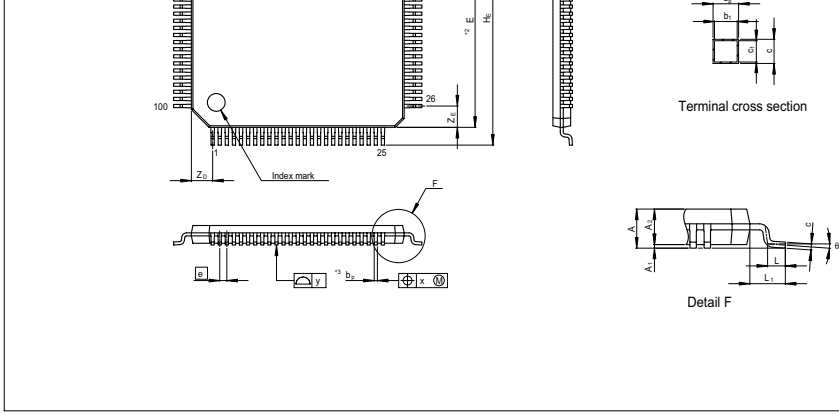
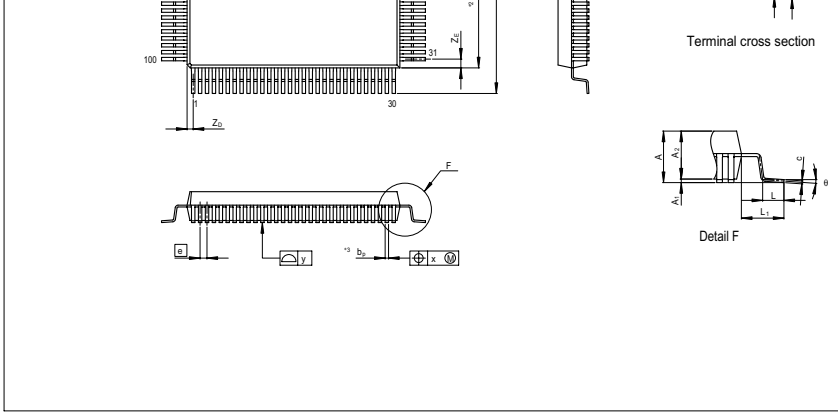


Figure G.1 TFP-100B Package Dimensions



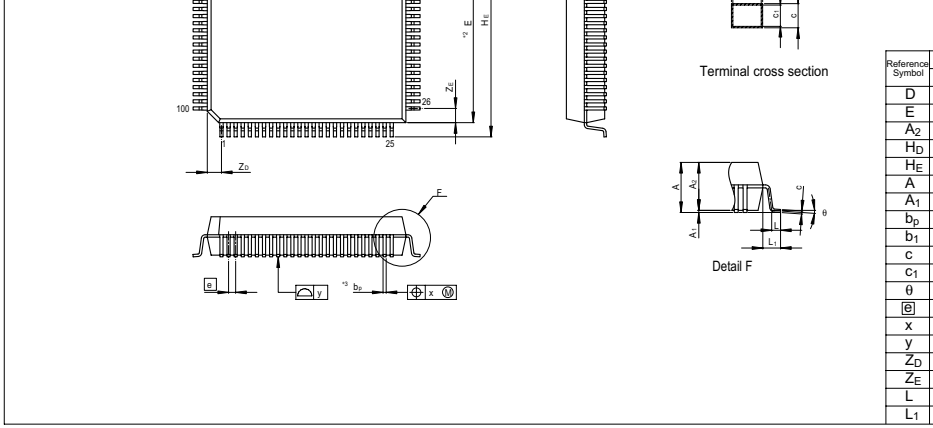
Reference Symbol
D
E
A_2
H_D
H_E
A
A_1
b_p
b_1
c
c_1
θ
e
x
y
Z_D
Z_E
L
L_1

Figure G.2 TFP-100G Package Dimensions



Referen Symbo
D
E
A_2
H_D
H_E
A
A_1
b_p
b_1
c
c_1
θ
e
x
y
Z_D
Z_E
L
L_1

Figure G.3 FP-100A Package Dimensions



Reference Symbol
D
E
A_2
H_D
H_E
A
A_1
b_p
b_1
c
c_1
θ
e
x
y
Z_D
Z_E
L
L_1

Figure G.4 FP-100B Package Dimensions

Renesas 16-Bit Single-Chip Microcomputer
Hardware Manual
H8S/2345 Group, H8S/2345 F-ZTAT™

Publication Date: 1st Edition, September 1997
Rev.4.00, February 15, 2006
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

RENESAS SALES OFFICES

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehs
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8S/2345 Group, H8S/2345 F-ZTAT™ Hardware Manual

Renesas Electronics Corporation
1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ09B0291-0400

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for renesas ***manufacturer:***

Other Similar products are found below :

EL4511CUZ-T7 PYB15-Q24-S5-H-U PQA30-D24-S24-DH PQA30-D48-S12-TH PYB30-Q24-T312-H-U PYB15-Q24-S5-H-T PYB15-Q24-S12-H-T V7815-500-SMT PYB20-Q48-S12-H-T PQZ6-Q24-S15-D PYB20-Q48-S5-H-T PYB20-Q24-S12-H-T VLED15-120-350 VGS-75-12 PYB15-Q24-S12-H-U VGS-50-15 VGS-50-24 R5F100GFAFB#V0 VGS-25-24 VGS-50-5 VGS-100-12 M30620FCAFP#U3 ETSA120500UD-P5P-SZ PDQ2-D24-S12-S PDS1-S12-D12-M PDS1-S12-D15-M PYB15-Q24-S12-T PYB15-Q24-S24 PYB20-Q24-S24-DIN PYB20-Q48-S12 R0K33062PS000BE R0K505220S000BE R0K561664S000BE R0K570865S000BE HC55185AIMZ R5F52108CDFP#30 R5F72145BDFA#V1 R5S72631P200FP R7S721001VCBGAC0 EPSA050250UB-P5P-EJ HS0005PUU01H IS82C55A-5 ISL29035EVAL1Z ISL55110IVZ ISL6730AEVAL1Z ISL68200DEMO1Z ISL78235EVAL2Z ISL78268EVAL1Z ISL91107IRA-EVZ ISL9220IRTZEVAL1Z