Hitachi Single-Chip Microcomputer
H8S/2128 Series
H8S/2124 Series
H8S/2128 F-ZTAT™

H8S/2128 HD6432128W, HD6432128,
HD64F2128, HD64F2128V
H8S/2127 HD6432127RW, HD6432127R
H8S/2126 HD6432126RW, HD6432126R
H8S/2124 HD6432124
H8S/2123 HD6432123
H8S/2122 HD6432122
H8S/2120 HD6432120

Hardware Manual

HITACHI

ADE-602-114A
Rev. 2.0
3/12/99

Notice

When using this document, keep the following in mind:

1. This document may, wholly or partially, be subject to change without notice.
2. All rights are reserved: No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without Hitachi's permission.
3. Hitachi will not be held responsible for any damage to the user that may result from accidents or any other reasons during operation of the user's unit according to this document.
4. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Hitachi's semiconductor products. Hitachi assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.
5. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi, Ltd.
6. MEDICAL APPLICATIONS: Hitachi's products are not authorized for use in MEDICAL APPLICATIONS without the written consent of the appropriate officer of Hitachi's sales company. Such use includes, but is not limited to, use in life support systems. Buyers of Hitachi's products are requested to notify the relevant Hitachi sales offices when planning to use the products in MEDICAL APPLICATIONS.

Contents

Preface

The H8S/2128 Series and H8S/2124 Series comprise high-performance microcomputers with a 32-bit H8S/2000 CPU core, and a set of on-chip supporting functions required for system configuration.

The H8S/2000 CPU can execute basic instructions in one state, and is provided with sixteen internal 16-bit general registers with a 32-bit configuration, and a concise and optimized instruction set. The CPU can handle a 16-Mbyte linear address space (architecturally 4 Gbytes). Programs based on the high-level language C can also be run efficiently.

Single-power-supply flash memory (F-ZTAT™*) and mask ROM versions are available, providing a quick and flexible response to conditions from ramp-up through full-scale volume production, even for applications with frequently changing specifications.

On-chip peripheral functions include a 16-bit free-running timer module (FRT), 8-bit timer module (TMR), watchdog timer module (WDT), two PWM timers (PWM and PWMX), a serial communication interface (SCI), A/D converter (ADC), and I/O ports. An I^2C bus interface (IIC) can also be incorporated as an option.

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

The H8S/2128 Series has all the above on-chip supporting functions, and can also be provided with an IIC module as an options. The H8S/2124 Series comprises reduced-function versions, with fewer TMR, and no PWM, IIC, or DTC modules.

Use of the H8S/2128 or H8S/2124 Series enables compact, high-performance systems to be implemented easily. The various timer functions and their interconnectability (timer connection), plus the interlinked operation of the I^2C bus interface and data transfer controller (DTC), in particular, make these devices ideal for use in PC monitors. In addition, the combination of F-ZTAT™ and reduced-function versions is ideal for system applications in which on-chip program memory is essential to meet performance requirements, product start-up times are short, and program modifications may be necessary after end-product assembly.

This manual describes the hardware of the H8S/2128 Series and H8S/2124 Series. Refer to the *H8S/2600 Series and H8S/2000 Series Programming Manual* for a detailed description of the instruction set.

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

On-Chip Supporting Modules

Series	H8S/2128 Series	H8S/2124 Series
Product names	H8S/2128, 2127	H8S/2124, 2123, 2122, 2120
Bus controller (BSC)	Available (8 bits)	Available (8 bits)
Data transfer controller (DTC)	Available	—
8-bit PWM timer (PWM)	×16	—
14-bit PWM timer (PWMX)	×2	—
16-bit free-running timer (FRT)	×1	×1
8-bit timer (TMR)	×4	×3
Timer connection	Available	—
Watchdog timer (WDT)	×2	×2
Serial communication interface (SCI)	×2	×2
I^2C bus interface (IIC)	×2 (option)	—
A/D converter	×8 (analog inputs) ×8 (expansion A/D inputs)	×8 (analog inputs) ×8 (expansion A/D inputs)

Section 1 Overview

1.1 Overview

The H8S/2128 Series and H8S/2124 Series comprise microcomputers (MCUs) built around the H8S/2000 CPU, employing Hitachi's proprietary architecture, and equipped with supporting modules on-chip.

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

On-chip supporting modules required for system configuration include a data transfer controller (DTC) bus master, ROM and RAM memory, a16-bit free-running timer module (FRT), 8-bit timer module (TMR), watchdog timer module (WDT), two PWM timers (PWM and PWMX), serial communication interface (SCI), A/D converter (ADC), and I/O ports. An I^2C bus interface (IIC) can also be incorporated as an option.

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

Three operating modes, modes 1 to 3, are provided, and there is a choice of address space and single-chip mode or externally expanded modes.

The features of the H8S/2128 Series and H8S/2124 Series are shown in Table 1.1.

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

Table 1.1 Overview

Item	Specifications
CPU	• General-register architecture — Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers) • High-speed operation suitable for real-time control — Maximum operating frequency: 20 MHz/5 V, 10 MHz/3 V — High-speed arithmetic and logic operations 8/16/32-bit register-register add/subtract: 50 ns (20 MHz operation) 16 × 16-bit register-register multiply: 1000 ns (20 MHz operation) 32 ÷ 16-bit register-register divide: 1000 ns (20 MHz operation) • Instruction set suitable for high-speed operation — Sixty-five basic instructions — 8/16/32-bit transfer/arithmetic and logic instructions — Unsigned/signed multiply and divide instructions — Powerful bit-manipulation instructions • Two CPU operating modes — Normal mode: 64-kbyte address space — Advanced mode: 16-Mbyte address space
Operating modes	• Three MCU operating modes
Bus controller	• 2-state or 3-state access space can be designated for external expansion areas • Number of program wait states can be set for external expansion areas

				External Data Bus	
Mode	**CPU Operating Mode**	**Description**	**On-Chip ROM**	**Initial Value**	**Maximum Value**
1	Normal	Expanded mode with on-chip ROM disabled	Disabled	8 bits	8 bits
2	Advanced	Expanded mode with on-chip ROM enabled	Enabled	8 bits	8 bits
		Single-chip mode		None	
3	Normal	Expanded mode with on-chip ROM enabled	Enabled	8 bits	8 bits
		Single-chip mode		None	

Table 1.1 Overview (cont)

Item	Specifications
Data transfer controller (DTC) (H8S/2128 Series)	• Can be activated by internal interrupt or software • Multiple transfers or multiple types of transfer possible for one activation source • Transfer possible in repeat mode, block transfer mode, etc. • Request can be sent to CPU for interrupt that activated DTC
16-bit free-running timer module (FRT: 1 channel)	• One 16-bit free-running counter (also usable for external event counting) • Two output compare outputs • Four input capture inputs (with buffer operation capability)
8-bit timer module (2 channels: TMR0, TMR1)	Each channel has: • One 8-bit up-counter (also usable for external event counting) • Two timer constant registers • The two channels can be connected
Timer connection and 8-bit timer module (2 channels: TMRX, TMRY) (Timer connection and TMRX provided in H8S/2128 Series)	Input/output and FRT, TMR1, TMRX, TMRY can be interconnected • Measurement of input signal or frequency-divided waveform pulse width and cycle (FRT, TMR1) • Output of waveform obtained by modification of input signal edge (FRT, TMR1) • Determination of input signal duty cycle (TMRX) • Output of waveform synchronized with input signal (FRT, TMRX, TMRY) • Automatic generation of cyclical waveform (FRT, TMRY)
Watchdog timer module (WDT: 2 channels)	• Watchdog timer or interval timer function selectable • Subclock operation capability (channel 1 only)
8-bit PWM timer module (PWM) (H8S/2128 Series)	• Up to 16 outputs • Pulse duty cycle settable from 0 to 100% • Resolution: 1/256 • 1.25 MHz maximum carrier frequency (20 MHz operation)
14-bit PWM timer module (PWMX) (H8S/2128 Series)	• Up to 2 outputs • Resolution: 1/16384 • 312.5 kHz maximum carrier frequency (20 MHz operation)
Serial communication interface (SCI: 2 channels, SCI0, SCI1)	• Asynchronous mode or synchronous mode selectable • Multiprocessor communication function

Table 1.1 Overview (cont)

Item	Specifications
A/D converter	• Resolution: 10 bits • Input: 8 channels (dedicated analog input pins) 8 channels (expansion A/D input pins) • High-speed conversion: 6.7 µs minimum conversion time (20 MHz operation) • Single or scan mode selectable • Sample-and-hold function • A/D conversion can be activated by external trigger or timer trigger
I/O ports	• 43 input/output pins (including 24 with LED drive capability) • 8 input-only pins
Memory	• Flash memory or mask ROM • High-speed static RAM
Interrupt controller	• Four external interrupt pins (NMI, $\overline{IRQ0}$ to $\overline{IRQ2}$) • 33 internal interrupt sources • Three priority levels settable
Power-down state	• Medium-speed mode • Sleep mode • Module stop mode • Software standby mode • Hardware standby mode • Subclock operation
Clock pulse generator	• Built-in duty correction circuit
Packages	• 64-pin plastic DLP (DP-64S) • 64-pin plastic QFP (FP-64A) • 80-pin plastic TQFP (TFP-80C)
I^2C bus interface (IIC: 2 channels) (option in H8S/2128 Series)	• Conforms to Philips I^2C bus interface standard • Single master mode/slave mode • Arbitration lost condition can be identified • Supports two slave addresses

Product Name	ROM	RAM
H8S/2124, H8S/2128	128 kbytes	4 kbytes
H8S/2123	96 kbytes	4 kbytes
H8S/2122, H8S/2127	64 kbytes	2 kbytes
H8S/2120, H8S/2126	32 kbytes	2 kbytes

Table 1.1 Overview (cont)

Item	Specifications			
Product lineup (preliminary)	Product Code			
	Series	Mask ROM Versions	F-ZTAT™ Versions	ROM/RAM (Bytes)	Packages
	H8S/2128	HD6432128 HD6432128W*	HD64F2128	128 k/4 k	DP-64S, FP-64A, TFP-80C
		HD6432127R HD6432127RW*	—	64 k/2 k	
		HD6432126R HD6432126RW*	—	32 k/2 k	
	H8S/2124	HD6432124	—	128 k/4 k	
		HD6432123	—	96 k/4 k	
		HD6432122	—	64 k/2 k	
		HD6432120	—	32 k/2 k	
	Note: * “W” indicates the I^2C bus option.				

1.2 Internal Block Diagram

An internal block diagram of the H8S/2128 Series is shown in figure 1.1, and an internal block diagram of the H8S/2124 Series in figure 1.2.

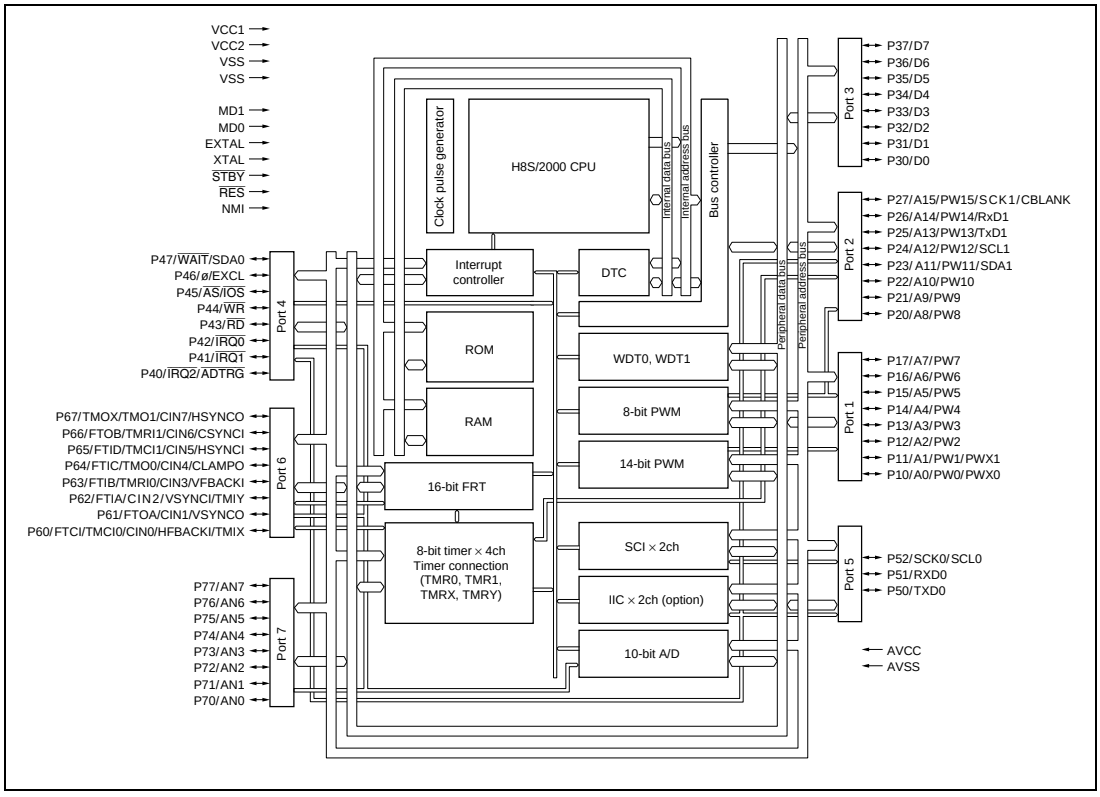


Figure 1.1 Internal Block Diagram of H8S/2128 Series

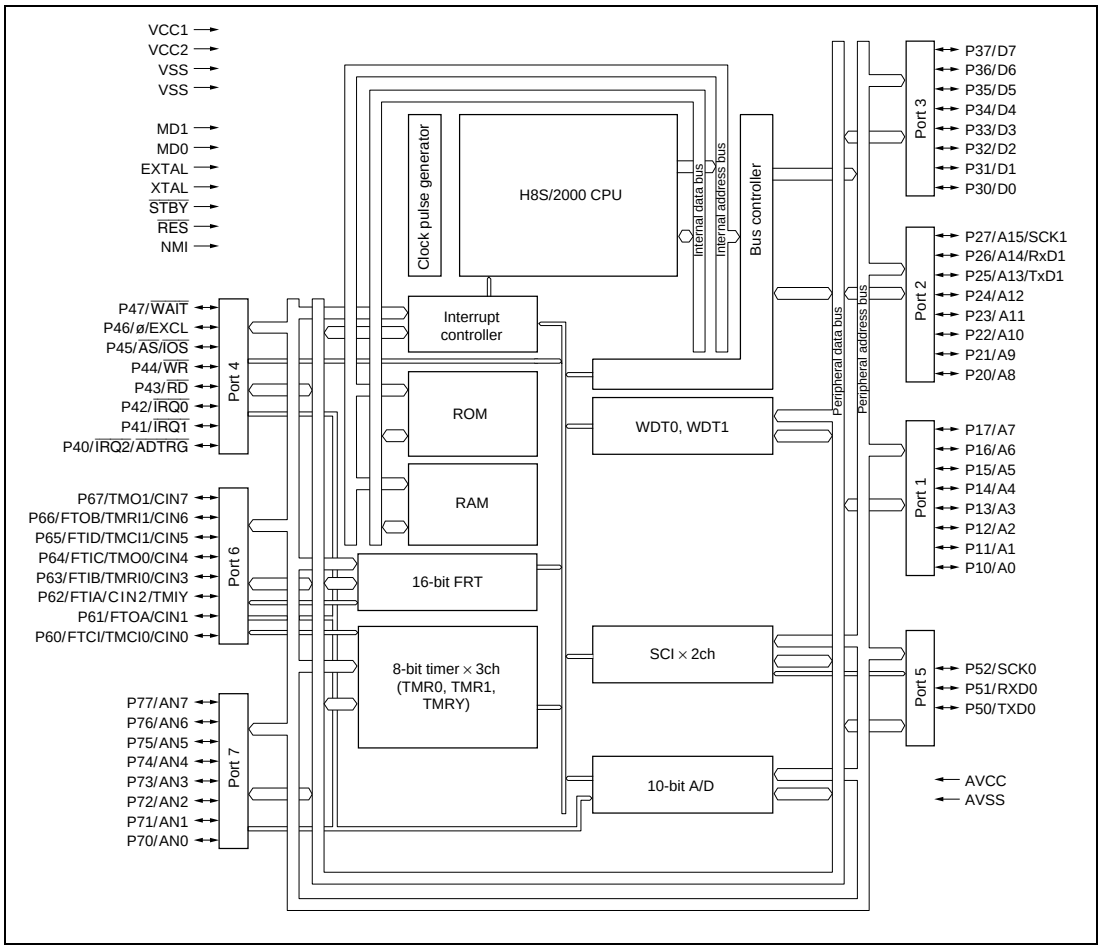


Figure 1.2 Internal Block Diagram of H8S/2124 Series

1.3 Pin Arrangement and Functions

1.3.1 Pin Arrangement

The pin arrangement of the H8S/2128 Series is shown in figures 1.3 to 1.5, and the pin arrangement of the H8S/2124 Series in figures 1.6 to 1.8.

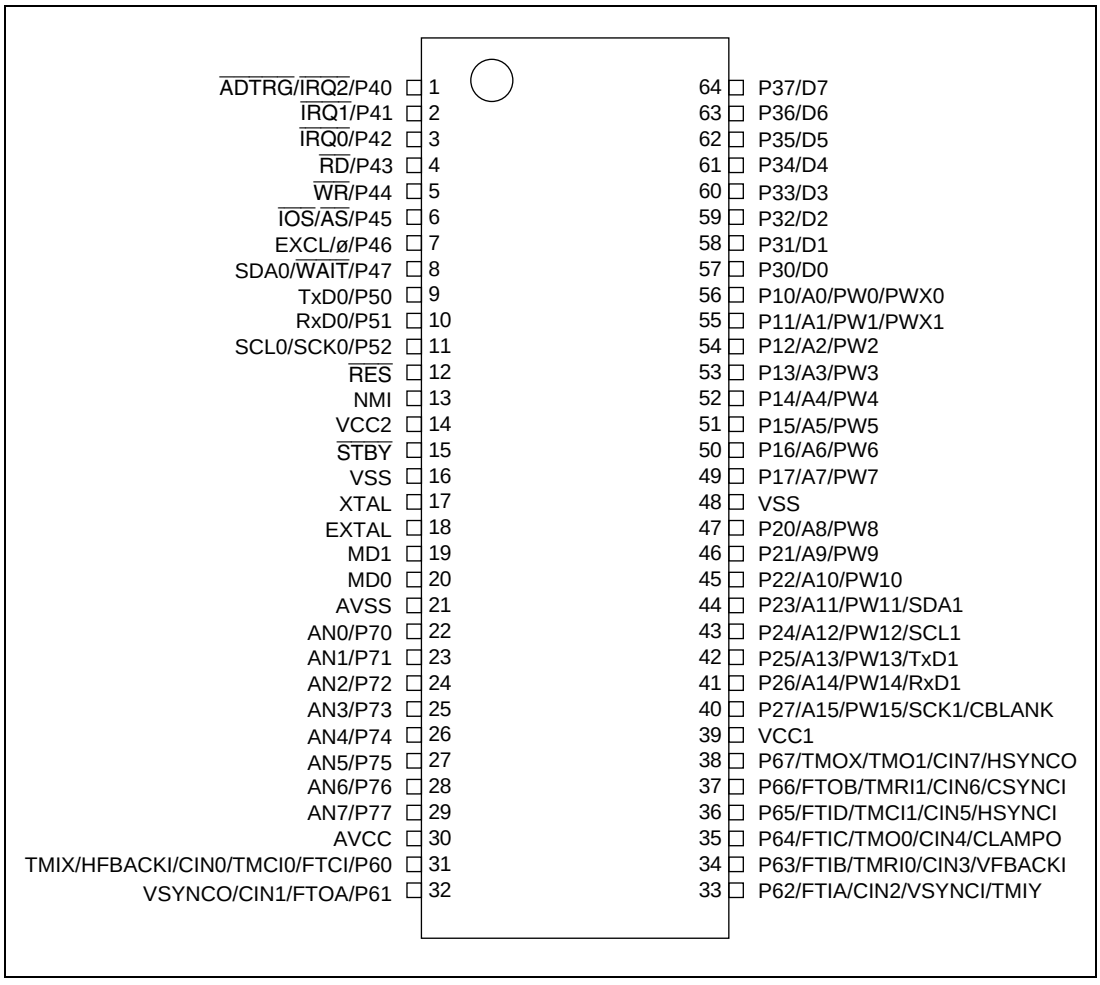


Figure 1.3 Pin Arrangement of H8S/2128 Series (DP-64S: Top View)

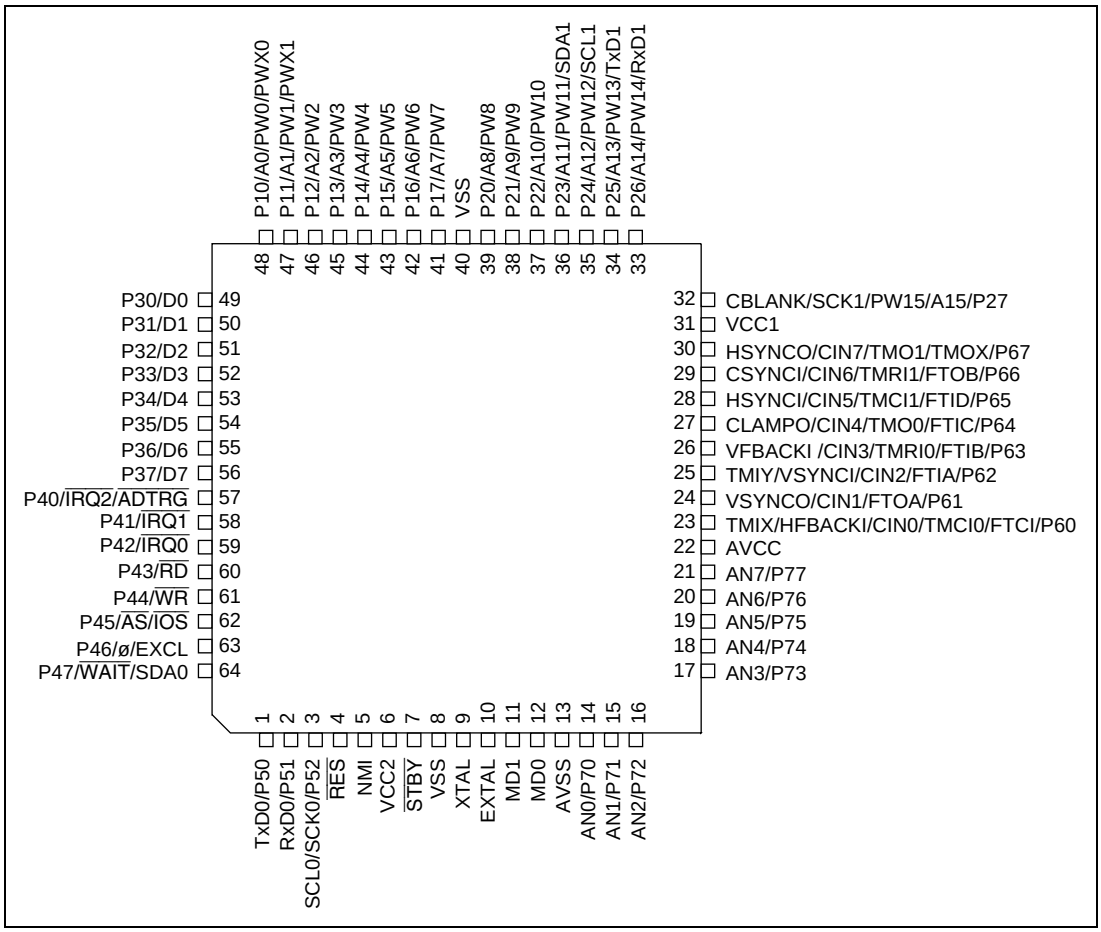


Figure 1.4 Pin Arrangement of H8S/2128 Series (FP-64A: Top View)

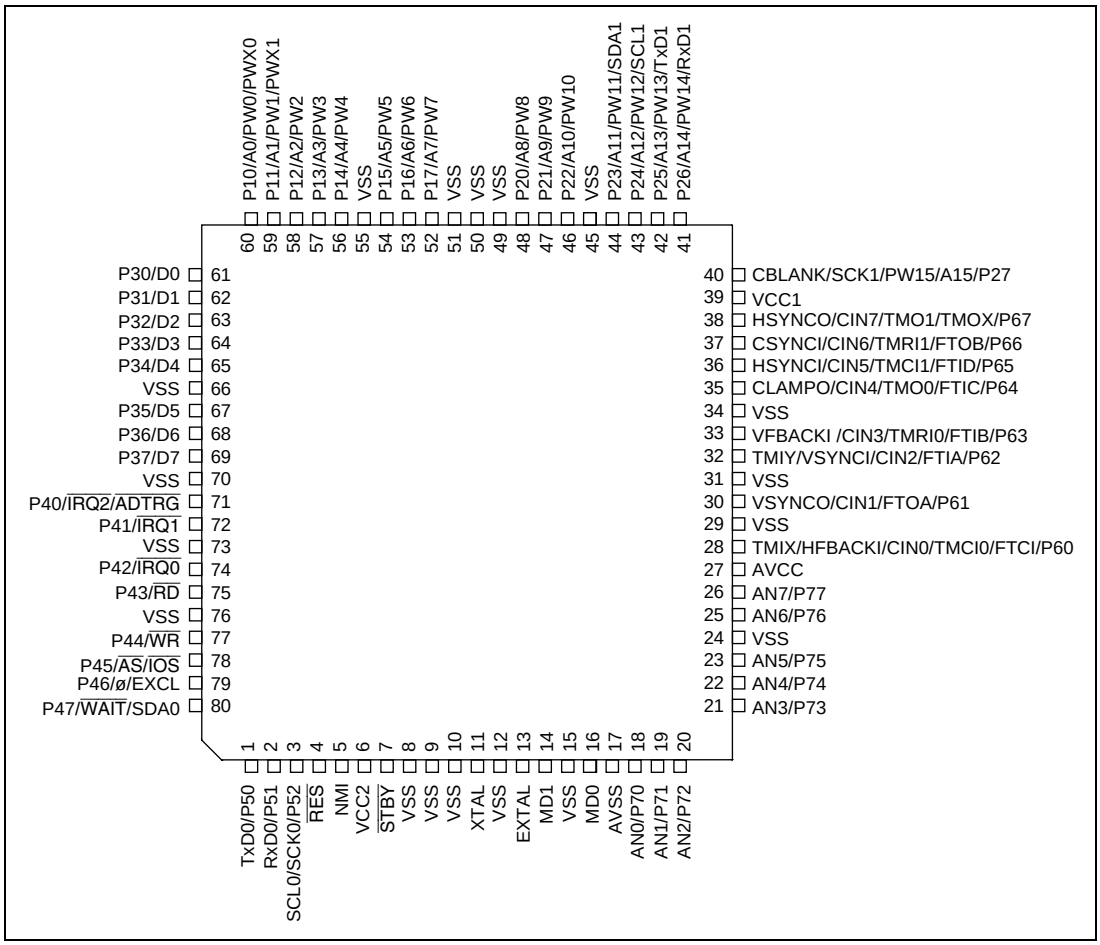


Figure 1.5 Pin Arrangement of H8S/2128 Series (TFP-80C: Top View)

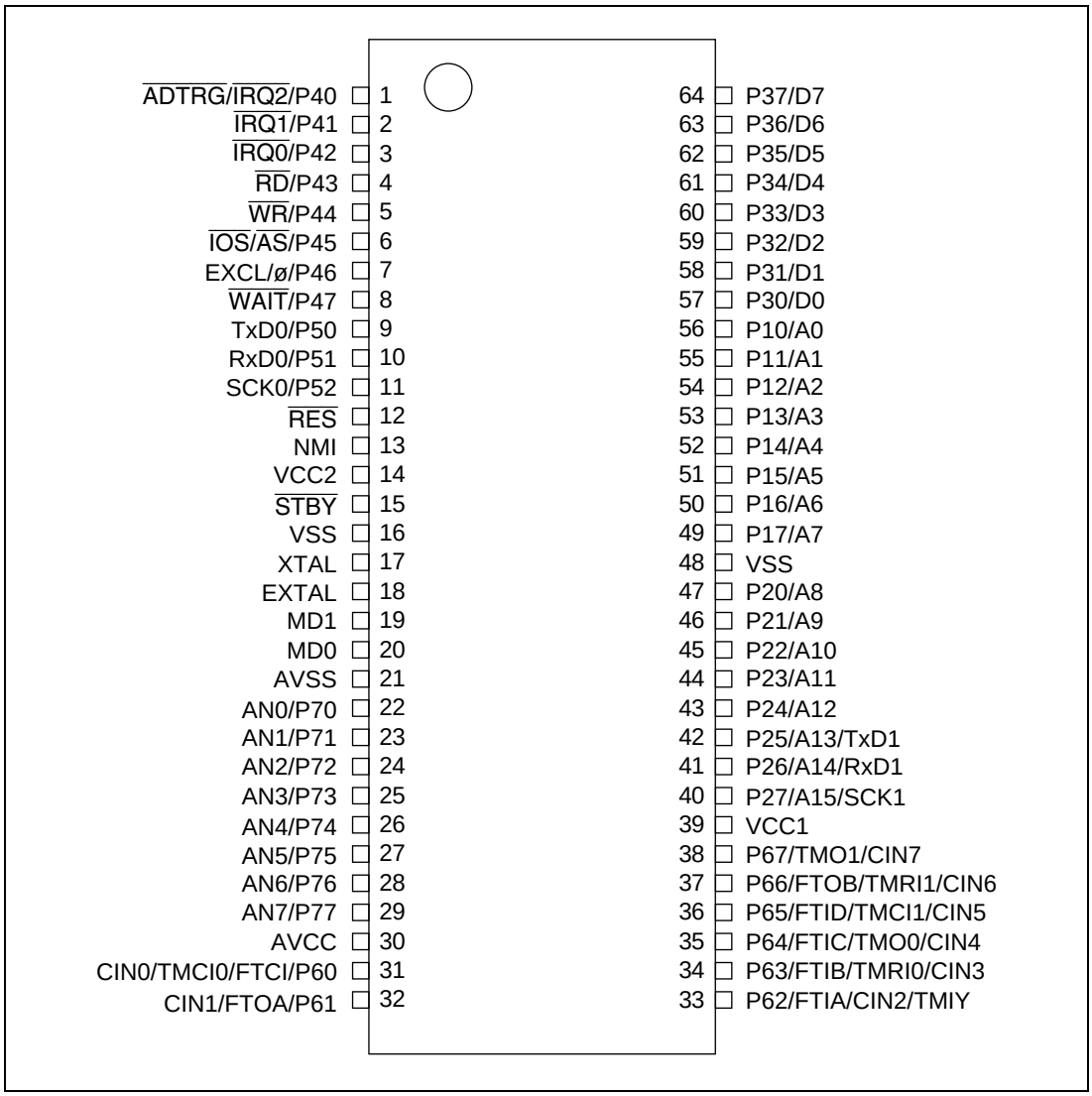


Figure 1.6 Pin Arrangement of H8S/2124 Series (DP-64S: Top View)

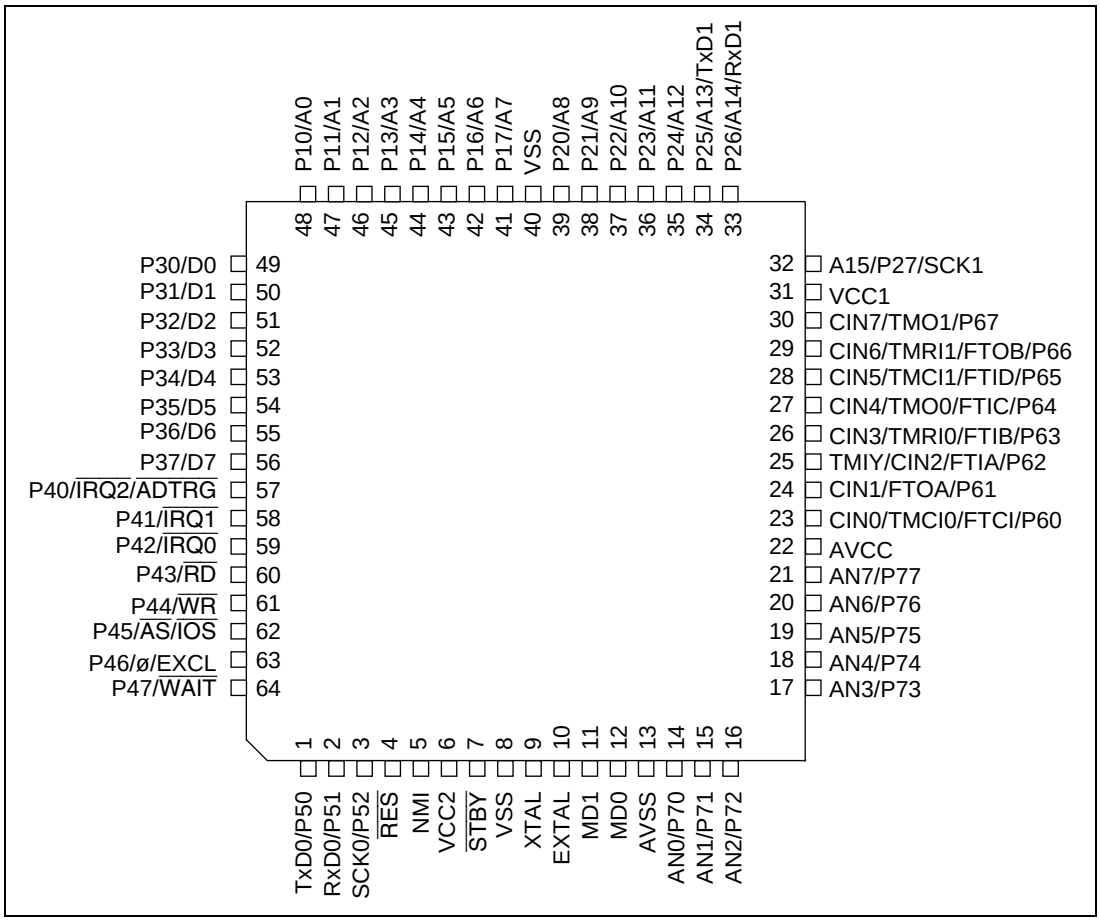


Figure 1.7 Pin Arrangement of H8S/2124 Series (FP-64A: Top View)

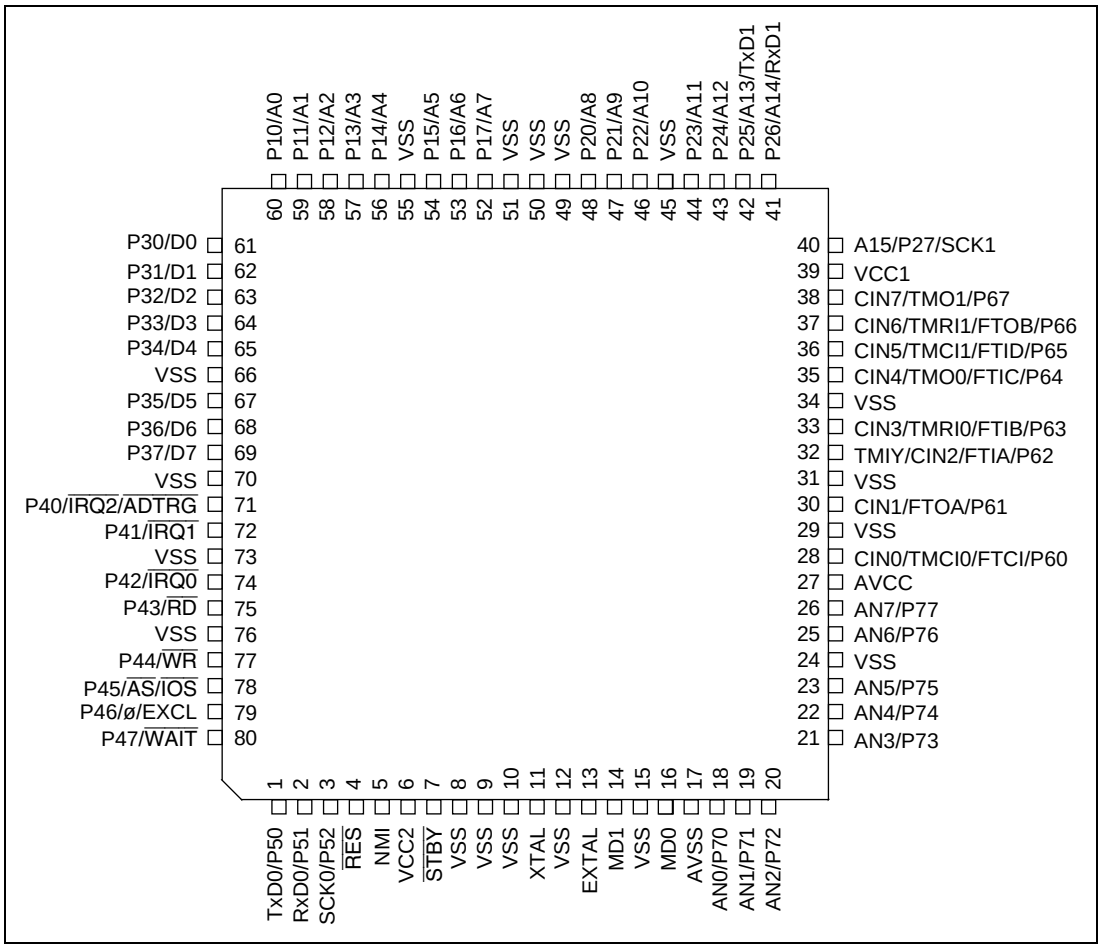


Figure 1.8 Pin Arrangement of H8S/2124 Series (TFP-80C: Top View)

1.3.2 Pin Functions in Each Operating Mode

Tables 1.2 and 1.3 show the pin functions of the H8S/2128 Series and H8S/2124 Series in each of the operating modes.

Table 1.2 H8S/2128 Series Pin Functions in Each Operating Mode

Pin No.			Pin Name			
			Expanded Modes		Single-Chip Modes	Flash Memory Writer Mode
DP-64S	FP-64A	TFP-80C	Mode 1	Mode 2 (EXPE = 1) Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0) Mode 3 (EXPE = 0)	
1	57	71	P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	VCC
2	58	72	P41/$\overline{\text{IRQ1}}$	P41/$\overline{\text{IRQ1}}$	P41/$\overline{\text{IRQ1}}$	VCC
—	—	73	VSS	VSS	VSS	VSS
3	59	74	P42/$\overline{\text{IRQ0}}$	P42/$\overline{\text{IRQ0}}$	P42/$\overline{\text{IRQ0}}$	VSS
4	60	75	$\overline{\text{RD}}$	$\overline{\text{RD}}$	P43	$\overline{\text{WE}}$
—	—	76	VSS	VSS	VSS	VSS
5	61	77	$\overline{\text{WR}}$	$\overline{\text{WR}}$	P44	FA15
6	62	78	$\overline{\text{AS}}$/$\overline{\text{IOS}}$	$\overline{\text{AS}}$/$\overline{\text{IOS}}$	P45	FA16
7	63	79	ø/P46/EXCL	P46/ø/EXCL	P46/ø/EXCL	NC
8	64	80	P47/$\overline{\text{WAIT}}$/SDA0	P47/$\overline{\text{WAIT}}$/SDA0	P47/SDA0	VCC
9	1	1	P50/TxD0	P50/TxD0	P50/TxD0	NC
10	2	2	P51/RxD0	P51/RxD0	P51/RxD0	FA17
11	3	3	P52/SCK0/SCL0	P52/SCK0/SCL0	P52/SCK0/SCL0	NC
12	4	4	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
13	5	5	NMI	NMI	NMI	FA9
14	6	6	VCC2	VCC2	VCC2	VCC
15	7	7	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	VCC
16	8	8	VSS	VSS	VSS	VSS
—	—	9	VSS	VSS	VSS	VSS
—	—	10	VSS	VSS	VSS	VSS
17	9	11	XTAL	XTAL	XTAL	XTAL
—	—	12	VSS	VSS	VSS	VSS
18	10	13	EXTAL	EXTAL	EXTAL	EXTAL
19	11	14	MD1	MD1	MD1	VSS

Table 1.2 H8S/2128 Series Pin Functions in Each Operating Mode (cont)

Pin No.			Pin Name			
			Expanded Modes		Single-Chip Modes	Flash Memory Writer Mode
DP-64S	FP-64A	TFP-80C	Mode 1	Mode 2 (EXPE = 1) Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0) Mode 3 (EXPE = 0)	
—	—	15	VSS	VSS	VSS	VSS
20	12	16	MD0	MD0	MD0	VSS
21	13	17	AVSS	AVSS	AVSS	VSS
22	14	18	P70/AN0	P70/AN0	P70/AN0	NC
23	15	19	P71/AN1	P71/AN1	P71/AN1	NC
24	16	20	P72/AN2	P72/AN2	P72/AN2	NC
25	17	21	P73/AN3	P73/AN3	P73/AN3	NC
26	18	22	P74/AN4	P74/AN4	P74/AN4	NC
27	19	23	P75/AN5	P75/AN5	P75/AN5	NC
—	—	24	VSS	VSS	VSS	VSS
28	20	25	P76/AN6	P76/AN6	P76/AN6	NC
29	21	26	P77/AN7	P77/AN7	P77/AN7	NC
30	22	27	AVCC	AVCC	AVCC	VCC
31	23	28	P60/FTCI/TMCI0/ CIN0/HFBACKI/ TMIX	P60/FTCI/TMCI0/ CIN0/HFBACKI/ TMIX	P60/FTCI/TMCI0/ CIN0/HFBACKI/ TMIX	NC
—	—	29	VSS	VSS	VSS	VSS
32	24	30	P61/FTOA/CIN1/ VSYNCO	P61/FTOA/CIN1/ VSYNCO	P61/FTOA/CIN1/ VSYNCO	NC
—	—	31	VSS	VSS	VSS	VSS
33	25	32	P62/FTIA/CIN2/ VSYNCI/TMIY	P62/FTIA/CIN2/ VSYNCI/TMIY	P62/FTIA/CIN2/ VSYNCI/TMIY	NC
34	26	33	P63/FTIB/TMRI0/ CIN3/VFBACKI	P63/FTIB/TMRI0/ CIN3/VFBACKI	P63/FTIB/TMRI0/ CIN3/VFBACKI	NC
—	—	34	VSS	VSS	VSS	VSS
35	27	35	P64/FTIC/TMO0/ CIN4/CLAMPO	P64/FTIC/TMO0/ CIN4/CLAMPO	P64/FTIC/TMO0/ CIN4/CLAMPO	NC
36	28	36	P65/FTID/TMCI1/ CIN5/HSYNCI	P65/FTID/TMCI1/ CIN5/HSYNCI	P65/FTID/TMCI1/ CIN5/HSYNCI	NC
37	29	37	P66/FTOB/TMRI1/ CIN6/CSYNCI	P66/FTOB/TMRI1/ CIN6/CSYNCI	P66/FTOB/TMRI1/ CIN6/CSYNCI	NC

Table 1.2 H8S/2128 Series Pin Functions in Each Operating Mode (cont)

Pin No.			Pin Name			
			Expanded Modes		Single-Chip Modes	Flash Memory Writer Mode
DP-64S	FP-64A	TFP-80C	Mode 1	Mode 2 (EXPE = 1) Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0) Mode 3 (EXPE = 0)	
38	30	38	P67/TMOX/ TMO1/CIN7/ HSYNCO	P67/TMOX/TMO1/ CIN7/HSYNCO	P67/TMO1/TMOX/ CIN7/HSYNCO	VSS
39	31	39	VCC1	VCC1	VCC1	VCC
40	32	40	A15	A15/P27/PW15/ SCK1/CBLANK	P27/PW15/ SCK1/CBLANK	$\overline{CE}$
41	33	41	A14	A14/P26/PW14/ RxD1	P26/PW14/ RxD1	FA14
42	34	42	A13	A13/P25/PW13/ TxD1	P25/PW13/ TxD1	FA13
43	35	43	A12	A12/P24/PW12/ SCL1	P24/PW12/SCL1	FA12
44	36	44	A11	A11/P23/PW11/ SDA1	P23/PW11/SDA1	FA11
—	—	45	VSS	VSS	VSS	VSS
45	37	46	A10	A10/P22/PW10	P22 /PW10	FA10
46	38	47	A9	A9 /P21/PW9	P21/PW9	$\overline{OE}$
47	39	48	A8	A8 /P20 /PW8	P20/PW8	FA8
—	—	49	VSS	VSS	VSS	VSS
48	40	50	VSS	VSS	VSS	VSS
—	—	51	VSS	VSS	VSS	VSS
49	41	52	A7	A7/P17/PW7	P17/PW7	FA7
50	42	53	A6	A6/P16/PW6	P16/PW6	FA6
51	43	54	A5	A5/P15/PW5	P15/PW5	FA5
—	—	55	VSS	VSS	VSS	VSS
52	44	56	A4	A4/P14/PW4	P14/PW4	FA4
53	45	57	A3	A3/P13/PW3	P13/PW3	FA3
54	46	58	A2	A2/P12/PW2	P12/PW2	FA2
55	47	59	A1	A1/P11/PW1/PWX 1	P11/PW1/PWX1	FA1
56	48	60	A0	A0/P10/PW0/PWX 0	P10/PW0/PWX0	FA0

Table 1.2 H8S/2128 Series Pin Functions in Each Operating Mode (cont)

Pin No.			Pin Name			
			Expanded Modes		Single-Chip Modes	Flash Memory Writer Mode
DP-64S	FP-64A	TFP-80C	Mode 1	Mode 2 (EXPE = 1) Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0) Mode 3 (EXPE = 0)	
57	49	61	D0	D0	P30	FO0
58	50	62	D1	D1	P31	FO1
59	51	63	D2	D2	P32	FO2
60	52	64	D3	D3	P33	FO3
61	53	65	D4	D4	P34	FO4
—	—	66	VSS	VSS	VSS	VSS
62	54	67	D5	D5	P35	FO5
63	55	68	D6	D6	P36	FO6
64	56	69	D7	D7	P37	FO7
—	—	70	VSS	VSS	VSS	VSS

Table 1.3 H8S/2124 Series Pin Functions in Each Operating Mode

Pin No.			Pin Name			
			Expanded Modes		Single-Chip Modes	Flash Memory
DP-64S	FP-64A	TFP-80C	Mode 1	Mode 2 (EXPE = 1) Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0) Mode 3 (EXPE = 0)	Writer Mode
1	57	71	P40/IRQ2/ADTRG	P40/IRQ2/ADTRG	P40/IRQ2/ADTRG	VCC
2	58	72	P41/IRQ1	P41/IRQ1	P41/IRQ1	VCC
—	—	73	VSS	VSS	VSS	VSS
3	59	74	P42/IRQ0	P42/IRQ0	P42/IRQ0	VSS
4	60	75	RD	RD	P43	WE
—	—	76	VSS	VSS	VSS	VSS
5	61	77	WR	WR	P44	FA15
6	62	78	AS/IOS	AS/IOS	P45	FA16
7	63	79	P46/ø/EXCL	P46/ø/EXCL	P46/ø/EXCL	NC
8	64	80	P47/WAIT	P47/WAIT	P47	VCC
9	1	1	P50/TxD0	P50/TxD0	P50/TxD0	NC
10	2	2	P51/RxD0	P51/RxD0	P51/RxD0	FA17
11	3	3	P52/SCK0	P52/SCK0	P52/SCK0	NC
12	4	4	RES	RES	RES	RES
13	5	5	NMI	NMI	NMI	FA9
14	6	6	VCC2	VCC2	VCC2	VCC
15	7	7	STBY	STBY	STBY	VCC
16	8	8	VSS	VSS	VSS	VSS
—	—	9	VSS	VSS	VSS	VSS
—	—	10	VSS	VSS	VSS	VSS
17	9	11	XTAL	XTAL	XTAL	XTAL
—	—	12	VSS	VSS	VSS	VSS
18	10	13	EXTAL	EXTAL	EXTAL	EXTAL
19	11	14	MD1	MD1	MD1	VSS
—	—	15	VSS	VSS	VSS	VSS
20	12	16	MD0	MD0	MD0	VSS
21	13	17	AVSS	AVSS	AVSS	VSS
22	14	18	P70/AN0	P70/AN0	P70/AN0	NC

Table 1.3 H8S/2124 Series Pin Functions in Each Operating Mode (cont)

Pin No.			Pin Name			
			Expanded Modes		Single-Chip Modes	Flash Memory Writer Mode
DP-64S	FP-64A	TFP-80C	Mode 1	Mode 2 (EXPE = 1) Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0) Mode 3 (EXPE = 0)	
23	15	19	P71/AN1	P71/AN1	P71/AN1	NC
24	16	20	P72/AN2	P72/AN2	P72/AN2	NC
25	17	21	P73/AN3	P73/AN3	P73/AN3	NC
26	18	22	P74/AN4	P74/AN4	P74/AN4	NC
27	19	23	P75/AN5	P75/AN5	P75/AN5	NC
—	—	24	VSS	VSS	VSS	VSS
28	20	25	P76/AN6	P76/AN6	P76/AN6	NC
29	21	26	P77/AN7	P77/AN7	P77/AN7	NC
30	22	27	AVCC	AVCC	AVCC	VCC
31	23	28	P60/FTCI/TMCI0/ CIN0	P60/FTCI/TMCI0/ CIN0	P60/FTCI/TMCI0/ CIN0	NC
—	—	29	VSS	VSS	VSS	VSS
32	24	30	P61/FTOA/CIN1	P61/FTOA/CIN1	P61/FTOA/CIN1	NC
—	—	31	VSS	VSS	VSS	VSS
33	25	32	P62/FTIA/CIN2/ TMIY	P62/FTIA/CIN2/ TMIY	P62/FTIA/CIN2/ TMIY	NC
34	26	33	P63/FTIB/TMRI0/ CIN3	P63/FTIB/TMRI0/ CIN3	P63/FTIB/TMRI0/ CIN3	NC
—	—	34	VSS	VSS	VSS	VSS
35	27	35	P64/FTIC/TMO0/ CIN4	P64/FTIC/TMO0/ CIN4	P64/FTIC/TMO0/ CIN4	NC
36	28	36	P65/FTID/TMCI1/ CIN5	P65/FTID/TMCI1/ CIN5	P65/FTID/TMCI1/ CIN5	NC
37	29	37	P66/FTOB/TMRI1/ CIN6	P66/FTOB/TMRI1/ CIN6	P66/FTOB/TMRI1/ CIN6	NC
38	30	38	P67/TMO1/CIN7	P67/TMO1/CIN7	P67/TMO1/CIN7	VSS
39	31	39	VCC1	VCC1	VCC1	VCC
40	32	40	A15	A15/P27/SCK1	P27/SCK1	$\overline{CE}$
41	33	41	A14	A14/P26/RxD1	P26/RxD1	FA14

Table 1.3 H8S/2124 Series Pin Functions in Each Operating Mode (cont)

Pin No.			Pin Name			
			Expanded Modes		Single-Chip Modes	Flash Memory Writer Mode
DP-64S	FP-64A	TFP-80C	Mode 1	Mode 2 (EXPE = 1) Mode 3 (EXPE = 1)	Mode 2 (EXPE = 0) Mode 3 (EXPE = 0)	
42	34	42	A13	A13/P25/TxD1	P25/TxD1	FA13
43	35	43	A12	A12/P24	P24	FA12
44	36	44	A11	A11/P23	P23	FA11
—	—	45	VSS	VSS	VSS	VSS
45	37	46	A10	A10/P22	P22	FA10
46	38	47	A9	A9 /P21	P21	$\overline{OE}$
47	39	48	A8	A8 /P20	P20	FA8
—	—	49	VSS	VSS	VSS	VSS
48	40	50	VSS	VSS	VSS	VSS
—	—	51	VSS	VSS	VSS	VSS
49	41	52	A7	A7/P17	P17	FA7
50	42	53	A6	A6/P16	P16	FA6
51	43	54	A5	A5/P15	P15	FA5
—	—	55	VSS	VSS	VSS	VSS
52	44	56	A4	A4/P14	P14	FA4
53	45	57	A3	A3/P13	P13	FA3
54	46	58	A2	A2/P12	P12	FA2
55	47	59	A1	A1/P11	P11	FA1
56	48	60	A0	A0/P10	P10	FA0
57	49	61	D0	D0	P30	FO0
58	50	62	D1	D1	P31	FO1
59	51	63	D2	D2	P32	FO2
60	52	64	D3	D3	P33	FO3
61	53	65	D4	D4	P34	FO4
—	—	66	VSS	VSS	VSS	VSS
62	54	67	D5	D5	P35	FO5
63	55	68	D6	D6	P36	FO6
64	56	69	D7	D7	P37	FO7
—	—	70	VSS	VSS	VSS	VSS

1.3.3 Pin Functions

Table 1.4 summarizes the functions of the H8S/2128 Series and H8S/2124 Series pins.

Table 1.4 Pin Functions

		Pin No.				
Type	**Symbol**	**DP-64S**	**FP-64A**	**TFP-80C**	**I/O**	**Name and Function**
Power supply	VCC1, VCC2	14, 39	6, 31	6, 39	Input	**Power supply:** For connection to the power supply. All VCC1 and VCC2 pins should be connected to the system power supply.
	VSS	16, 48	8, 40	8, 9, 10, 12, 15, 24, 29, 31, 34, 45, 49, 50, 51, 55, 66, 70, 73, 76	Input	**Ground:** For connection to the power supply (0 V). All VSS pins should be connected to the system power supply (0 V).
Clock	XTAL	17	9	11	Input	Connected to a crystal oscillator. See section 21, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator and external clock input.
	EXTAL	18	10	13	Input	Connected to a crystal oscillator. The EXTAL pin can also input an external clock. See section 21, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator and external clock input.
	ø	7	63	79	Output	**System clock:** Supplies the system clock to external devices.
	EXCL	7	63	79	Input	**External subclock input:** Input a 32.768 kHz external subclock.

Table 1.4 Pin Functions (cont)

Type	Symbol	Pin No. DP-64S	FP-64A	TFP-80C	I/O	Name and Function
Operating mode control	MD1 MD0	19 20	11 12	14 16	Input	**Mode pins:** These pins set the operating mode. The relation between the settings of pins MD1 and MD0 and the operating mode is shown below. These pins should not be changed while the MCU is operating.

MD1	MD0	Operating Mode	Description
0	1	Mode 1	Normal Expanded mode with on-chip ROM disabled
1	0	Mode 2	Advanced Expanded mode with on-chip ROM enabled Single-chip mode
1	1	Mode 3	Normal Expanded mode with on-chip ROM enabled Single-chip mode

Type	Symbol	DP-64S	FP-64A	TFP-80C	I/O	Name and Function
System control	$\overline{\text{RES}}$	12	4	4	Input	**Reset input:** When this pin is driven low, the chip is reset.
	$\overline{\text{STBY}}$	15	7	7	Input	**Standby:** When this pin is driven low, a transition is made to hardware standby mode.
Address bus	A15 to A0	40 to 47, 49 to 56	32 to 39, 41 to 48	40 to 44, 46 to 48, 52 to 54, 56 to 60	Output	**Address bus:** These pins output an address.
Data bus	D7 to D0	64 to 57	56 to 49	69 to 67, 65 to 61	Input/ output	**Data bus:** These pins constitute a bidirectional data bus.

Table 1.4 Pin Functions (cont)

Type	Symbol	Pin No. DP-64S	FP-64A	TFP-80C	I/O	Name and Function
Bus control	$\overline{WAIT}$	8	64	80	Input	**Wait:** Requests insertion of a wait state in the bus cycle when accessing external 3-state address space.
	$\overline{RD}$	4	60	75	Output	**Read:** When this pin is low, it indicates that the external address space is being read.
	$\overline{WR}$	5	61	77	Output	**Write:** When this pin is low, it indicates that the external address space is being written to.
	$\overline{AS}/\overline{IOS}$	6	62	78	Output	**Address strobe:** When this pin is low, it indicates that address output on the address bus is valid.
Interrupt signals	NMI	13	5	5	Input	**Nonmaskable interrupt:** Requests a nonmaskable interrupt.
	$\overline{IRQ0}$ to $\overline{IRQ2}$	1 to 3	57 to 59	71, 72, 74	Input	**Interrupt request 0 to 2:** These pins request a maskable interrupt
16-bit free-running timer (FRT)	FTCI	31	23	28	Input	**FRT counter clock input:** Input pin for an external clock signal for the free-running counter (FRC).
	FTOA	32	24	30	Output	**FRT output compare A output:** The output compare A output pin.
	FTOB	37	29	37	Output	**FRT output compare B output:** The output compare B output pin.
	FTIA	33	25	32	Input	**FRT input capture A input:** The input capture A input pin.
	FTIB	34	26	33	Input	**FRT input capture B input:** The input capture B input pin.
	FTIC	25	27	35	Input	**FRT input capture C input:** The input capture C input pin.
	FTID	36	28	36	Input	**FRT input capture D input:** The input capture D input pin.

Table 1.4 Pin Functions (cont)

Type	Symbol	Pin No. DP-64S	FP-64A	TFP-80C	I/O	Name and Function
8-bit timer (TMR0, TMR1, TMRX, TMRY)	TMO0 TMO1 TMOX	35 38 38	27 30 30	35 38 38	Output	**Compare-match output:** TMR0, TMR1, and TMRX compare-match output pins.
	TMCI0 TMCI1	31 36	23 28	28 36	Input	**Counter external clock input:** TMR0 and TMR1 input pins for the external clock input to the counter.
	TMRI0 TMRI1	34 37	26 29	33 37	Input	**Counter external reset input:** TMR0 and TMR1 counter reset input pins.
	TMIX TMIY	31 33	23 25	28 32	Input	**Counter external clock input and reset input:** TMRX and TMRY counter clock input pins and reset input pins.
Serial communication interface (SCI0, SCI1)	TxD0	9	1	1	Output	**Transmit data:** Data output pins.
	TxD1	42	34	42		
	RxD0	10	2	2	Input	**Receive data:** Data input pins.
	RxD1	41	33	41		
	SCK0	11	3	3	Input/ output	**Serial clock:** Clock input/output pins. The SCK0 output type is NMOS push-pull. (H8S/2128 Series only)
	SCK1	40	32	40		
A/D converter	AN7 to AN0	29 to 22	21 to 14	26, 25, 23 to 18	Input	**Analog 7 to 0:** Analog input pins.
	CIN0 to CIN7	31 to 38	23 to 30	28, 30, 32 to 33, 35 to 38	Input	**Expansion A/D input:** Expansion A/D input pins can be connected to the A/D converter, but as they are also used as digital I/O pins, precision falls to the equivalent of 6-bit resolution.
	$\overline{\text{ADTRG}}$	1	57	71	Input	**A/D conversion external trigger input:** Pin for input of an external trigger to start A/D conversion.

Table 1.4 Pin Functions (cont)

Type	Symbol	Pin No. DP-64S	FP-64A	TFP-80C	I/O	Name and Function
A/D converter	AVCC	30	22	27	Input	**Analog power supply:** The reference power supply pin for the A/D converter. When the A/D converter is not used, this pin should be connected to the system power supply (+5 V or +3 V).
	AVSS	21	13	17	Input	**Analog ground:** The ground pin for the A/D converter. This pin should be connected to the system power supply (0 V).
PWM timer (PWM)	PW15 to PW0	40 to 47, 49 to 56	32 to 39, 41 to 48	40 to 44, 46 to 48, 52 to 54, 56 to 60	Output	**PWM timer output:** PWM timer pulse output pins.
14-bit PWM timer (PWMX)	PWX0 PWX1	56 55	48 47	60 59	Output	**PWMX timer output:** PWM D/A pulse output pins.
Timer connection	VSYNCI HSYNCI CSYNCI VFBACKI HFBACKI	33 36 37 34 31	25 28 29 26 23	32 36 37 33 28	Input	**Timer connection input:** Timer connection synchronous signal input pins.
	VSYNCO HSYNCO CLAMPO CBLANK	32 38 35 40	24 30 27 32	30 38 35 40	Output	**Timer connection output:** Timer connection synchronous signal output pins.
I^2C bus interface (IIC) (option)	SCL0 SCL1	11 43	3 35	3 43	Input/ output	**I^2C clock input/output (channels 0 and 1):** I^2C clock I/O pins. These pins have a bus drive function. The SCL0 output form is NMOS open-drain.
	SDA0 SDA1	8 44	64 36	80 44	Input/ output	**I^2C clock input/output (channels 0 and 1):** I^2C clock I/O pins. These pins have a bus drive function. The SDA0 output form is NMOS open-drain.

Table 1.4 Pin Functions (cont)

Type	Symbol	Pin No. DP-64S	FP-64A	TFP-80C	I/O	Name and Function
I/O ports	P17 to P10	49 to 56	41 to 48	52 to 54, 56 to 60	Input/ output	**Port 1:** Eight input/output pins. The data direction of each pin can be selected in the port 1 data direction register (P1DDR). These pins have built-in MOS input pull-ups, and also have LED drive capability.
	P27 to P20	40 to 47	32 to 39	40 to 44, 46 to 48	Input/ output	**Port 2:** Eight input/output pins. The data direction of each pin can be selected in the port 2 data direction register (P2DDR). These pins have built-in MOS input pull-ups, and also have LED drive capability.
	P37 to P30	64 to 57	56 to 49	69 to 67, 65 to 61	Input/ output	**Port 3:** Eight input/output pins. The data direction of each pin can be selected in the port 3 data direction register (P3DDR). These pins have built-in MOS input pull-ups, and also have LED drive capability.
	P47 to P40	8 to 1	64 to 57	80 to 77, 75, 74, 72, 71	Input/ output	**Port 4:** Eight input/output pins. The data direction of each pin can be selected in the port 4 data direction register (P4DDR). (Except P46) P47 is an NMOS push-pull output. (H8S/2128 Series only)
	P52 to P50	11 to 9	3 to 1	3 to 1	Input/ output	**Port 5:** Three input/output pins. The data direction of each pin can be selected in the port 5 data direction register (P5DDR). P52 is an NMOS push-pull output. (H8S/2128 Series only)
	P67 to P60	38 to 31	30 to 23	38 to 35, 33, 32, 30, 28	Input/ output	**Port 6:** Eight input/output pins. The data direction of each pin can be selected in the port 6 data direction register (P6DDR).
	P77 to P70	29 to 22	21 to 14	26, 25, 23 to 18	Input	**Port 7:** Eight input pins.

Section 2 CPU

2.1 Overview

The H8S/2000 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2000 CPU has sixteen 16-bit general registers, can address a 16-Mbyte (architecturally 4-Gbyte) linear address space, and is ideal for realtime control.

2.1.1 Features

The H8S/2000 CPU has the following features.

- Upward-compatible with H8/300 and H8/300H CPUs
  - Can execute H8/300 and H8/300H object programs
- General-register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)
- Sixty-five basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes (4 Gbytes architecturally)

- High-speed operation
  - All frequently-used instructions execute in one or two states
  - Maximum clock rate: 20 MHz
  - 8/16/32-bit register-register add/subtract: 50 ns
  - 8 × 8-bit register-register multiply: 600 ns
  - 16 ÷ 8-bit register-register divide: 600 ns
  - 16 × 16-bit register-register multiply: 1000 ns
  - 32 ÷ 16-bit register-register divide: 1000 ns
- Two CPU operating modes
  - Normal mode
  - 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 shown below.

- Register configuration

  The MAC register is supported only by the H8S/2600 CPU.
- Basic instructions

  The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by the H8S/2600 CPU.
- Number of execution states

  The number of execution states of the MULXU and MULXS instructions differ as follows.

		Number of Execution States	
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, EXR register functions, power-down state, etc., depending on the product.

2.1.3 Differences from H8/300 CPU

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

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

2.1.4 Differences from H8/300H CPU

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

- Additional control register
  - One 8-bit 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.

2.2 CPU Operating Modes

The H8S/2000 CPU has two operating modes: normal and advanced. Normal mode supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte total address space (architecturally the maximum total address space is 4 Gbytes, with a maximum of 16 Mbytes for the program area and a maximum of 4 Gbytes for the data area). The mode is selected by the mode pins of the microcontroller.

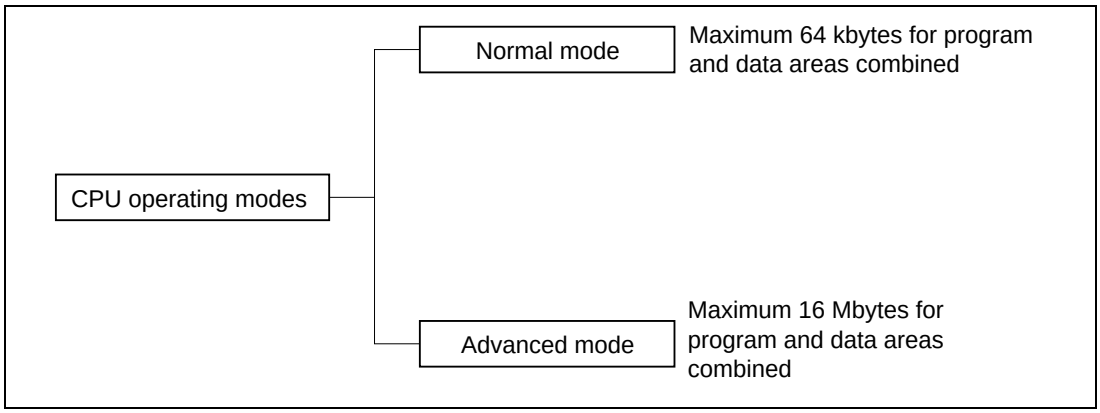


Figure 2.1 CPU Operating Modes

(1) Normal Mode

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

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

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers. When En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. If the general register is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corresponding extended register (En) will be affected.

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

Exception Vector Table and Memory Indirect Branch Addresses: In normal mode the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits. The configuration of the exception vector table in normal mode is shown in figure 2.2. For details of the exception vector table, see section 4, Exception Handling.

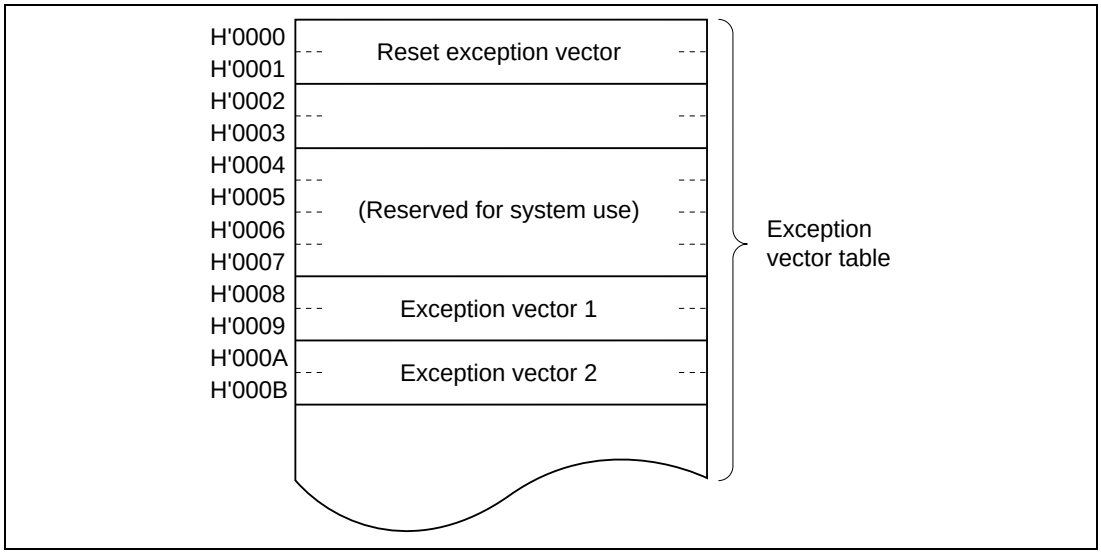


Figure 2.2 Exception Vector Table (Normal Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode the operand is a 16-bit word operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.

Stack Structure: When the program counter (PC) is pushed onto the stack in a subroutine call, and the PC and condition-code register (CCR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.3. The extended control register (EXR) is not pushed onto the stack. For details, see section 4, Exception Handling.

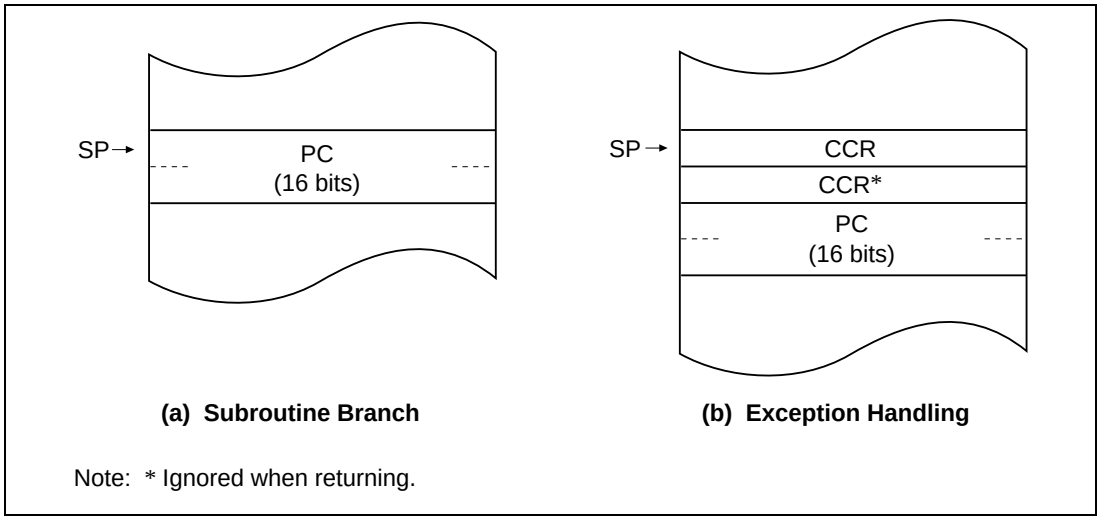


Figure 2.3 Stack Structure in Normal Mode

(2) Advanced Mode

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

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

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

Exception Vector Table and Memory Indirect Branch Addresses: In advanced mode the top area starting at H'00000000 is allocated to the exception vector table in units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2.4). For details of the exception vector table, see section 4, Exception Handling.

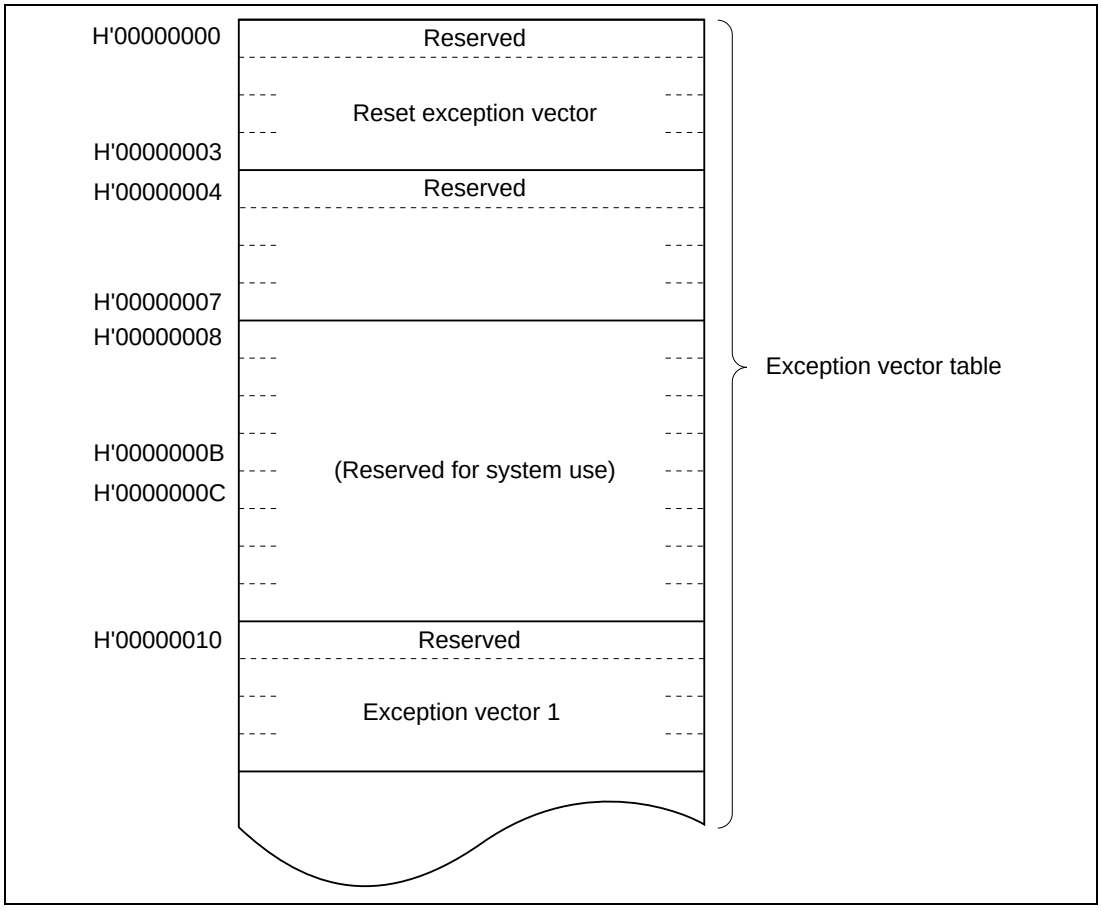


Figure 2.4 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In advanced mode the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper 8 bits of these 32 bits are a reserved area that is regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. Note that the first part of this range is also the exception vector table.

Stack Structure: In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC and condition-code register (CCR) are pushed onto the stack in exception handling, they are stored as shown in figure 2.5. The extended control register (EXR) is not pushed onto the stack. For details, see section 4, Exception Handling.

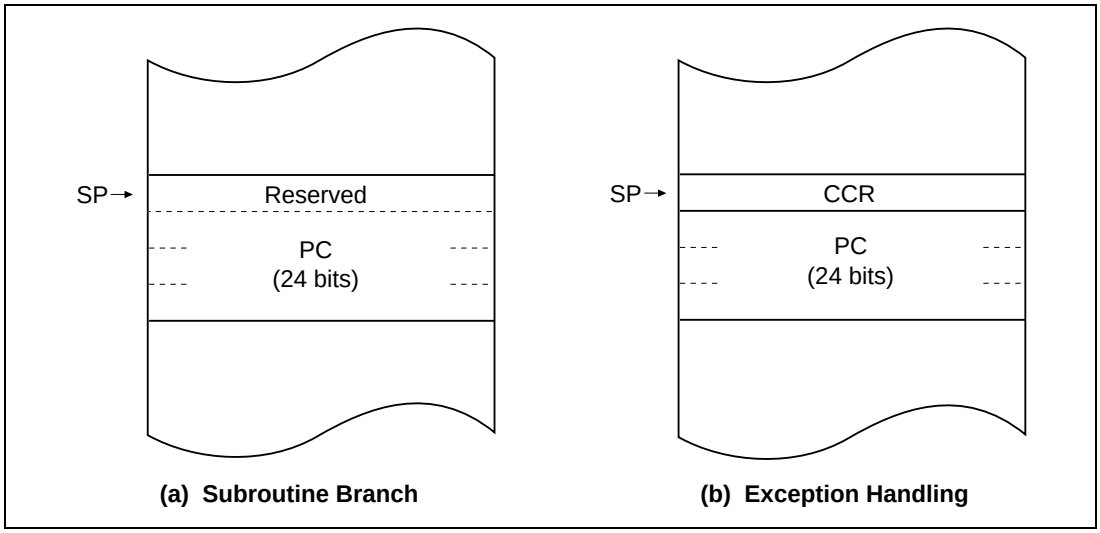


Figure 2.5 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2.6 shows a memory map of the H8S/2000 CPU. The H8S/2000 CPU provides linear access to a maximum 64-kbyte address space in normal mode, and a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode.

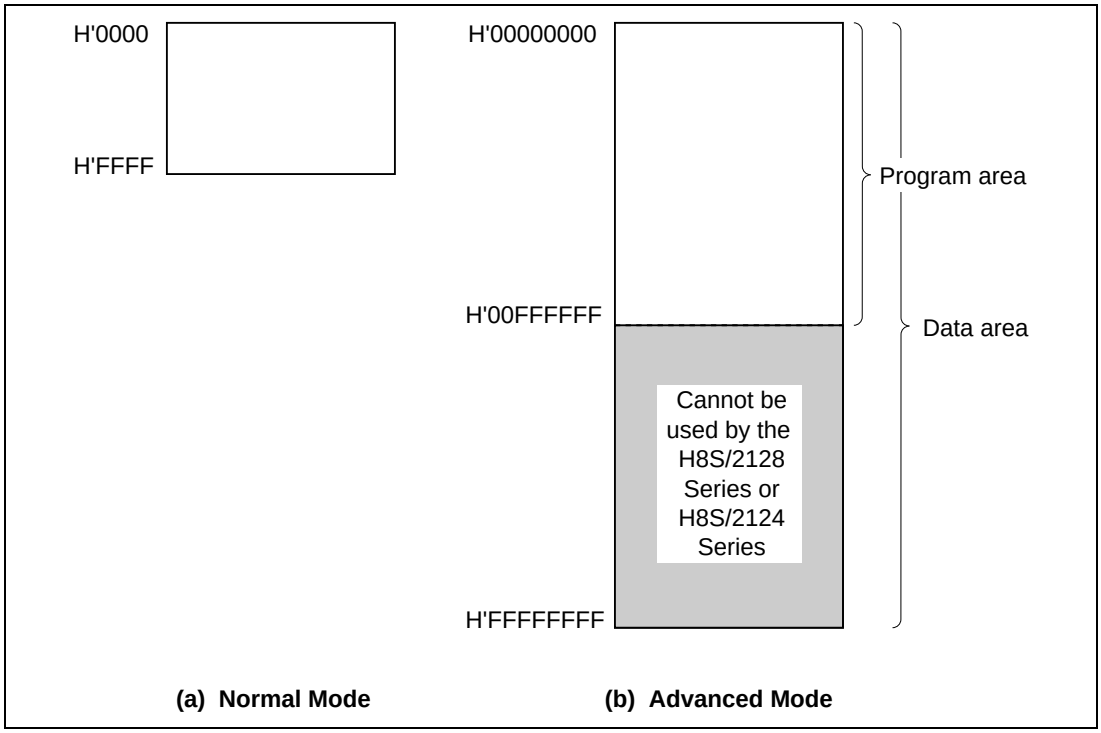


Figure 2.6 Memory Map

2.4 Register Configuration

2.4.1 Overview

The CPU has the internal registers shown in figure 2.7. There are two types of registers: general registers and control registers.

General Registers (Rn) and Extended Registers (En)

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

Control Registers (CR)

23 0
PC

	7	6	5	4	3	2	1	0
EXR*	T	—	—	—	—	I2	I1	I0

	7	6	5	4	3	2	1	0
CCR	I	UI	H	U	N	Z	V	C

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: * Does not affect operation in the H8S/2128 Series and H8S/2124 Series.

Figure 2.7 CPU Registers

2.4.2 General Registers

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

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

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

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

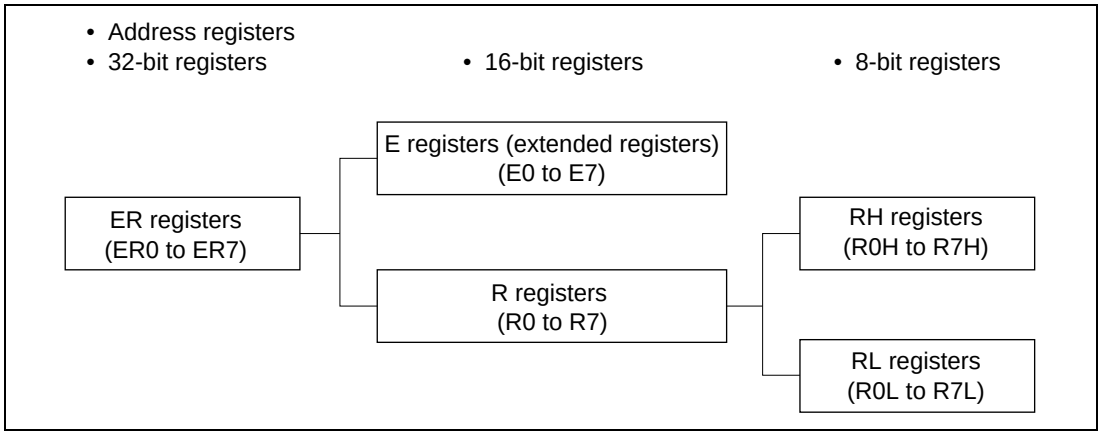


Figure 2.8 Usage of General Registers

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

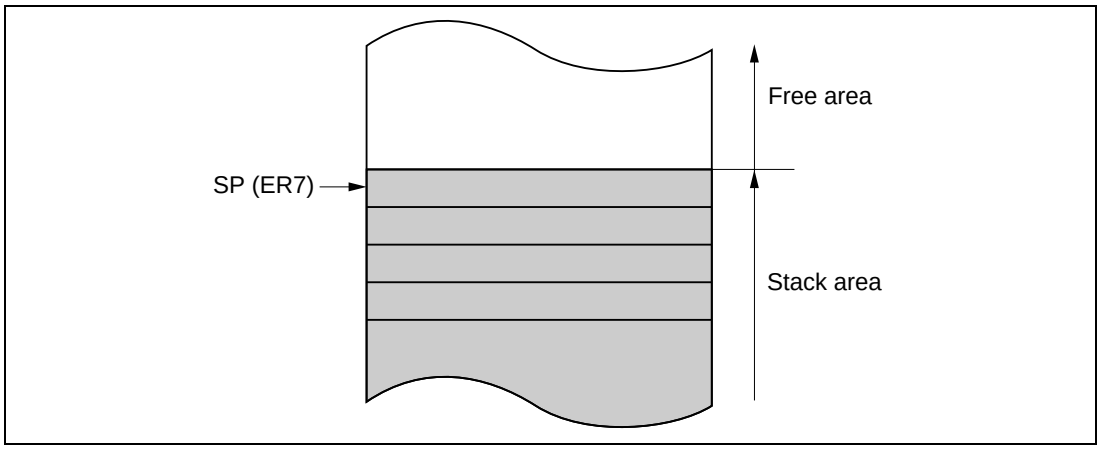


Figure 2.9 Stack

2.4.3 Control Registers

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

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

(2) Extended Control Register (EXR): An 8-bit register. In the H8S/2128 Series and H8S/2124 Series, this register does not affect operation.

Bit 7—Trace Bit (T): This bit is reserved. In the H8S/2128 Series and H8S/2124 Series, this bit does not affect operation.

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 are reserved. In the H8S/2128 Series and H8S/2124 Series, these bits do not affect operation.

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

Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. (NMI is accepted regardless of the I bit setting.) The I bit is set to 1 by hardware at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an interrupt mask bit. For details, refer to section 5, Interrupt Controller.

Bit 5—Half-Carry Flag (H): When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.

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

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

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

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

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

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

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

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

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

2.4.4 Initial Register Values

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

2.5 Data Formats

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

2.5.1 General Register Data Formats

Figure 2.10 shows the data formats in general registers.

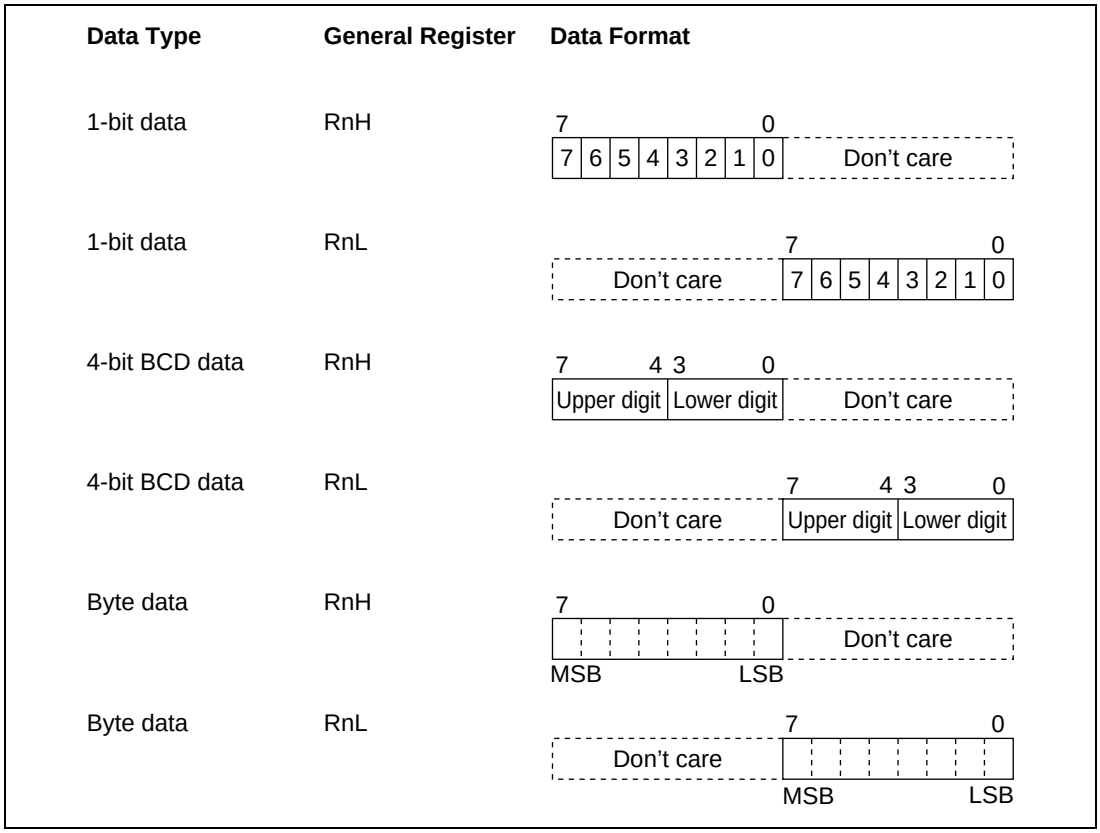


Figure 2.10 General Register Data Formats

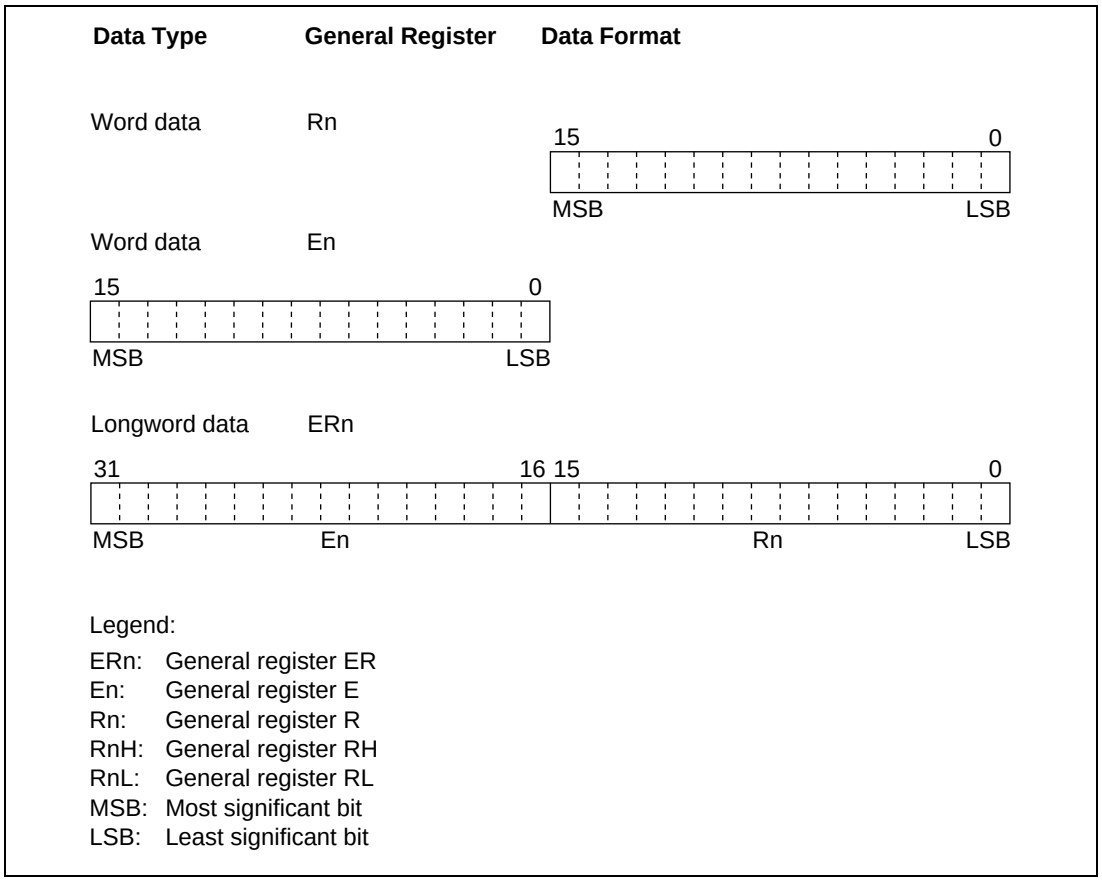


Figure 2.10 General Register Data Formats (cont)

2.5.2 Memory Data Formats

Figure 2.11 shows the data formats in memory. The CPU can access word data and longword data in memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

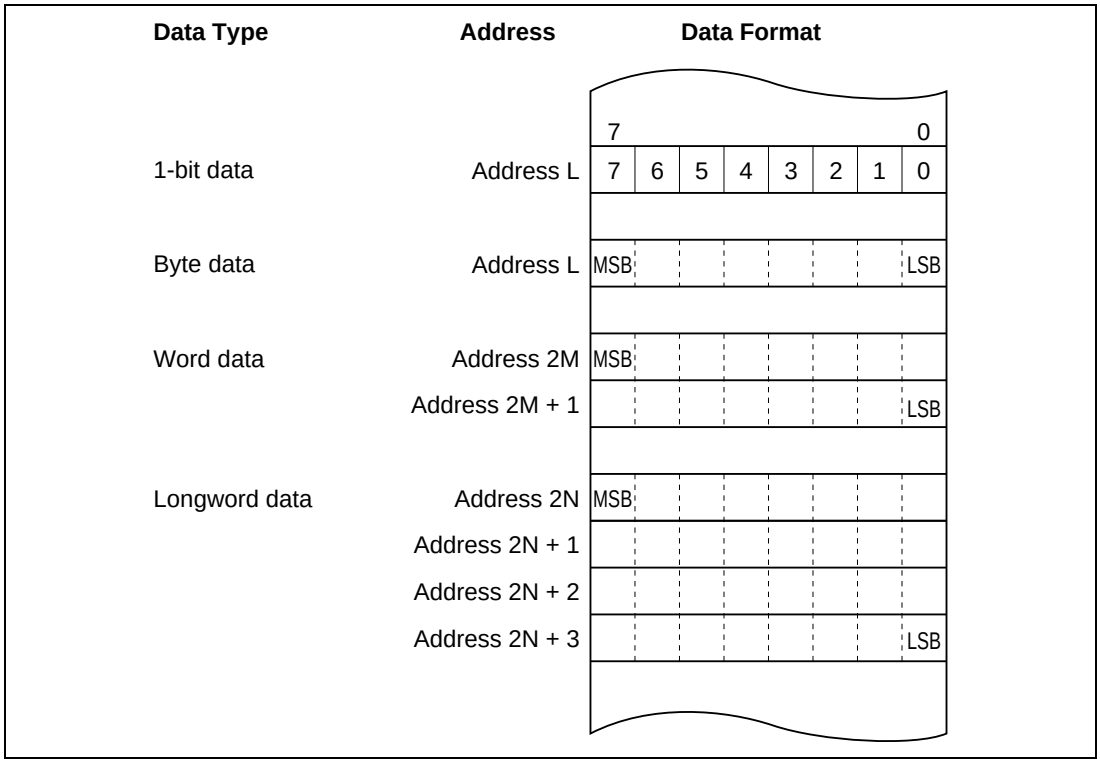


Figure 2.11 Memory Data Formats

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

2.6 Instruction Set

2.6.1 Overview

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

Table 2.1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	BWL	5
	POP*[1], PUSH*[1]	WL	
	LDM, STM	L	
	MOVFPE*[3], MOVTPE*[3]	B	
Arithmetic operations	ADD, SUB, CMP, EG	BWL	19
	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	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	BWL	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc*[2], JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1
			Total: 65 types

Notes: B: byte size; W: word size; L: longword size.

1. POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP.
   POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.
2. Bcc is the general name for conditional branch instructions.
3. Cannot be used in the H8S/2128 Series or H8S/2124 Series.

2.6.2 Instructions and Addressing Modes

Table 2.2 indicates the combinations of instructions and addressing modes that the H8S/2000 CPU can use.

Table 2.2 Combinations of Instructions and Addressing Modes

Function	Instruction	Addressing Modes													
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:32,ERn)	@–ERn/@ERn+	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC)	@@aa:8	—
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	—	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	—	WL
	LDM, STM	—	—	—	—	—	—	—	—	—	—	—	—	—	L
	MOVFPE*, 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	—	—	—	—	—	—	—	—	—	—	—	—	—	○

Note: * Cannot be used in the H8S/2128 Series or H8S/2124 Series.

Table 2.2 Combinations of Instructions and Addressing Modes (cont)

Function	Instruction	Addressing Modes													
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:32,ERn)	@–ERn/@ERn+	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC)	@@aa:8	—
System control	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	RTE	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	LDC	B	B	W	W	W	W	—	W	—	W	—	—	—	—
	STC	—	B	W	W	W	W	—	W	—	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—	—	—	○
Block data transfer		—	—	—	—	—	—	—	—	—	—	—	—	—	BW

Legend:

B: Byte

W: Word

L: Longword

2.6.3 Table of Instructions Classified by Function

Table 2.3 summarizes the instructions in each functional category. The notation used in table 2.3 is defined below.

Operation Notation

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

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

Table 2.3 Instructions Classified by Function

Type	Instruction	Size*	Function
Data transfer	MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
	MOVFPE	B	Cannot be used in the H8S/2128 Series or H8S/2124 Series.
	MOVTPE	B	Cannot be used in the H8S/2128 Series or H8S/2124 Series.
	POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn.
	PUSH	W/L	Rn → @–SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @–SP.
	LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
	STM	L	Rn (register list) → @–SP Pushes two or more general registers onto the stack.

Table 2.3 Instructions Classified by Function (cont)

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

Table 2.3 Instructions Classified by Function (cont)

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

Table 2.3 Instructions Classified by Function (cont)

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

Table 2.3 Instructions Classified by Function (cont)

Type	Instruction	Size*	Function
Bit-manipulation instructions	BSET	B	1 → (<bit-No.> of <EAd>) Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIAND	B	C ∧ ¬ (<bit-No.> of <EAd>) → C ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
	BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIOR	B	C ∨ ¬ (<bit-No.> of <EAd>) → C ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

Table 2.3 Instructions Classified by Function (cont)

Type	Instruction	Size*	Function
Bit-manipulation instructions	BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIXOR	B	C ⊕ ¬ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
	BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in a general register or memory operand to the carry flag.
	BILD	B	¬ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data.
	BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in a general register or memory operand.
	BIST	B	¬ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.

Table 2.3 Instructions Classified by Function (cont)

Type	Instruction	Size*	Function		
Branch instructions	Bcc	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.		
			Mnemonic	**Description**	**Condition**
			BRA(BT)	Always (true)	Always
			BRN(BF)	Never (false)	Never
			BHI	High	$C \vee Z = 0$
			BLS	Low or same	$C \vee Z = 1$
			BCC(BHS)	Carry clear (high or same)	$C = 0$
			BCS(BLO)	Carry set (low)	$C = 1$
			BNE	Not equal	$Z = 0$
			BEQ	Equal	$Z = 1$
			BVC	Overflow clear	$V = 0$
			BVS	Overflow set	$V = 1$
			BPL	Plus	$N = 0$
			BMI	Minus	$N = 1$
			BGE	Greater or equal	$N \oplus V = 0$
			BLT	Less than	$N \oplus V = 1$
			BGT	Greater than	$Z \vee (N \oplus V) = 0$
			BLE	Less or equal	$Z \vee (N \oplus V) = 1$
	JMP	—	Branches unconditionally to a specified address.		
	BSR	—	Branches to a subroutine at a specified address.		
	JSR	—	Branches to a subroutine at a specified address.		
	RTS	—	Returns from a subroutine		

Table 2.3 Instructions Classified by Function (cont)

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

Table 2.3 Instructions Classified by Function (cont)

Type	Instruction	Size*	Function
Block data transfer instructions	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;
			Block transfer instruction. Transfers the number of data bytes specified by R4L or R4 from locations starting at the address indicated by ER5 to locations starting at the address indicated by ER6. After the transfer, the next instruction is executed.

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

2.6.4 Basic Instruction Formats

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

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

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

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

Condition Field: Specifies the branching condition of Bcc instructions.

Figure 2.12 shows examples of instruction formats.

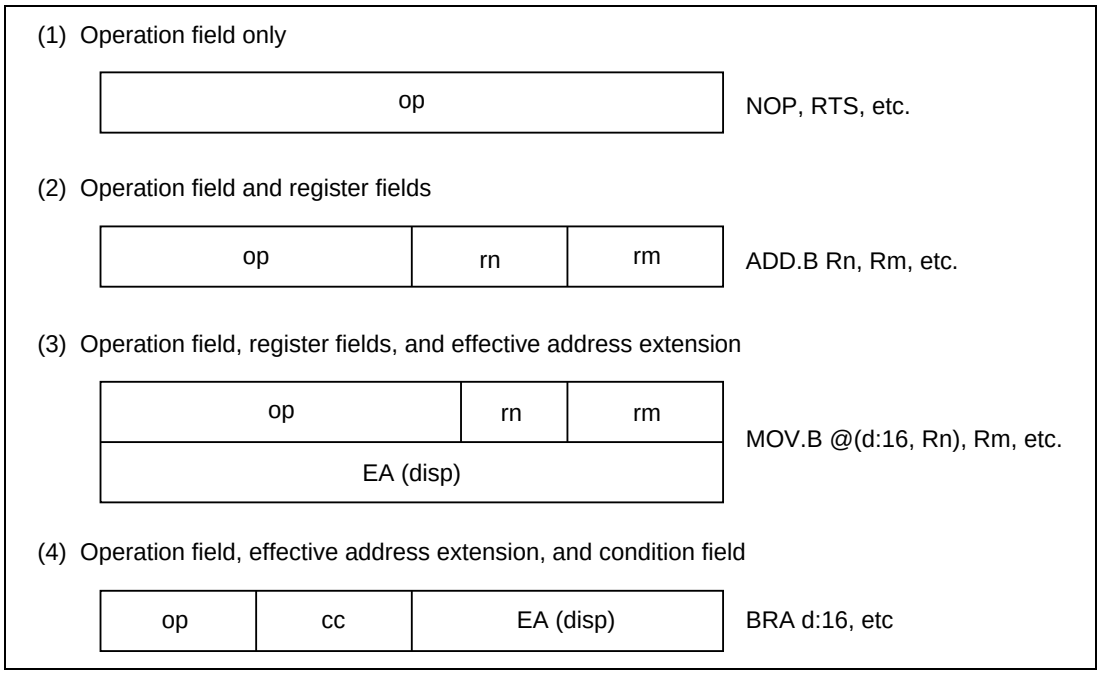


Figure 2.12 Instruction Formats (Examples)

2.6.5 Notes on Use of Bit-Manipulation Instructions

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

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

2.7 Addressing Modes and Effective Address Calculation

2.7.1 Addressing Mode

The CPU supports the eight addressing modes listed in table 2.4. Each instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit-manipulation instructions use register direct, register indirect, or absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2.4 Addressing Modes

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

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

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

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

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

- Register indirect with post-increment—@ERn+

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

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

Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32: The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long (@aa:32).

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

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

Table 2.5 indicates the accessible absolute address ranges.

Table 2.5 Absolute Address Access Ranges

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

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

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

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

Memory Indirect—@@aa:8: This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF in normal mode, H'000000 to H'0000FF in advanced mode). In normal mode the memory operand is a word operand and the branch address is 16 bits long. In advanced mode the memory operand is a longword operand, the first byte of which is assumed to be all 0 (H'00).

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

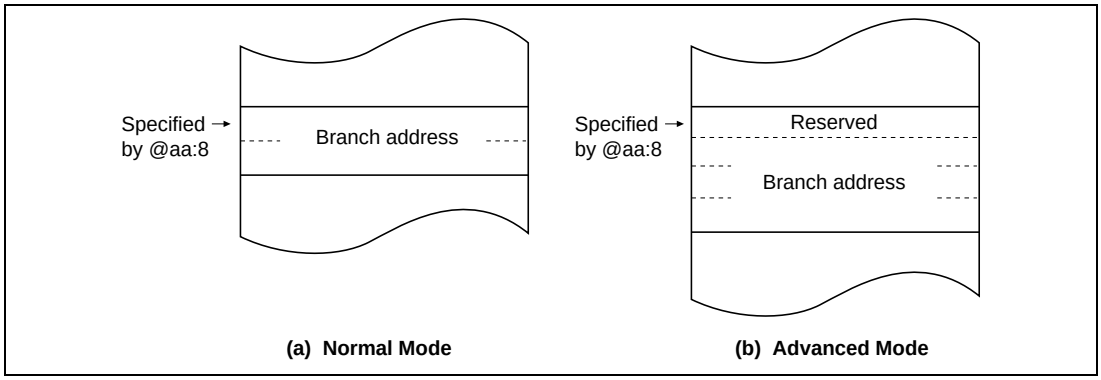


Figure 2.13 Branch Address Specification in Memory Indirect Mode

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

2.7.2 Effective Address Calculation

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

Table 2.6 Effective Address Calculation

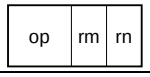
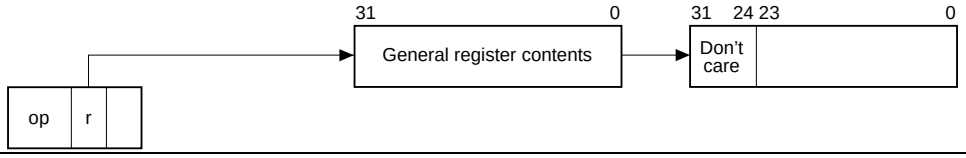
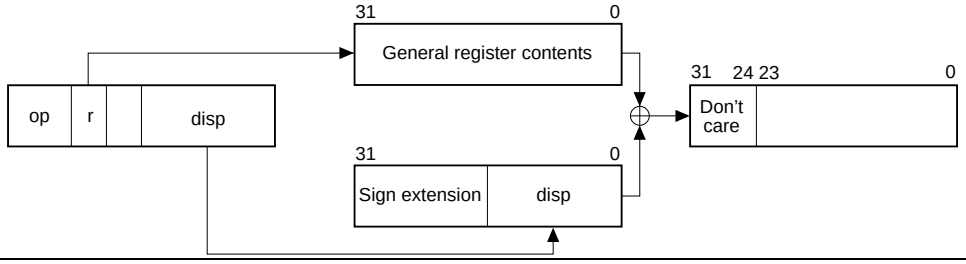
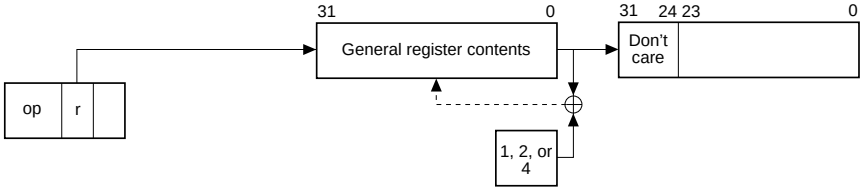
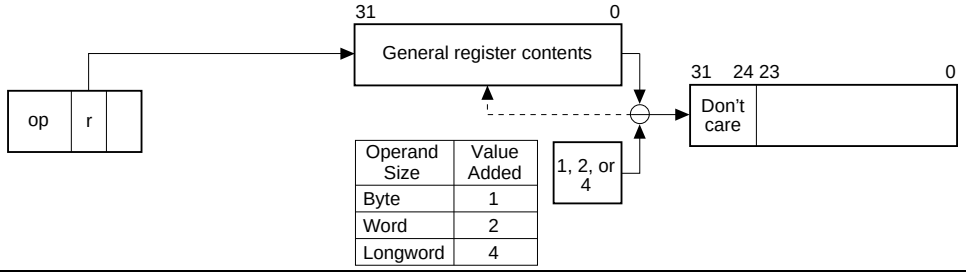
No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct (Rn) 		Operand is general register contents.
2	Register indirect (@ERn) 		
3	Register indirect with displacement @(d:16, ERn) or @(d:32, ERn) 		
4	Register indirect with post-increment or pre-decrement • Register indirect with post-increment @ERn+  • Register indirect with pre-decrement @-ERn  Operand Size / Value Added Byte 1 Word 2 Longword 4		

Table 2.6 Effective Address Calculation (cont)

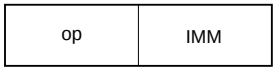
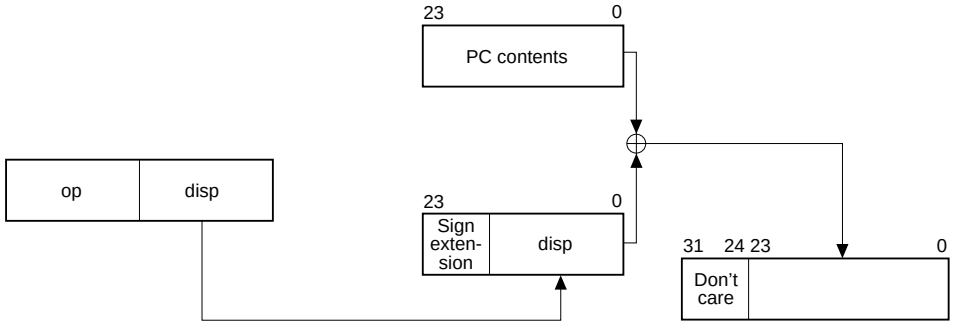
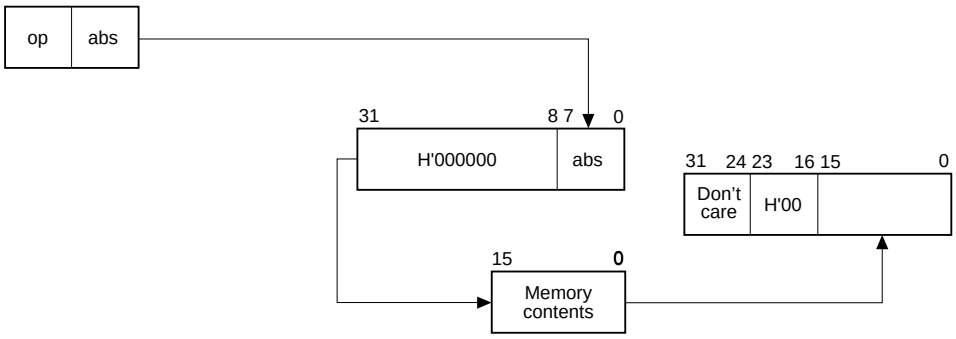
No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	Absolute address @aa:8  @aa:16  @aa:24  @aa:32 		
6	Immediate #xx:8/#xx:16/#xx:32 		Operand is immediate data.
7	Program-counter relative @(d:8, PC)/@(d:16, PC) 		

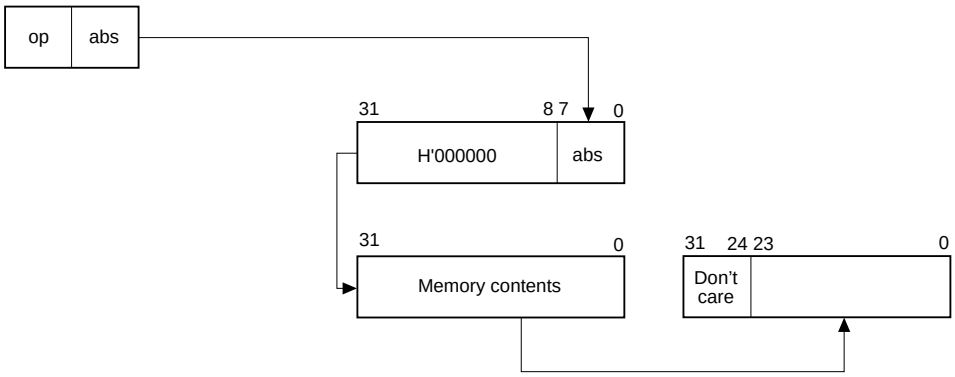
Table 2.6 Effective Address Calculation (cont)

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
8	Memory indirect @@aa:8		

- Normal mode



- Advanced mode



2.8 Processing States

2.8.1 Overview

The CPU has five main processing states: the reset state, exception-handling state, program execution state, bus-released state, and power-down state. Figure 2.14 shows a diagram of the processing states. Figure 2.15 indicates the state transitions.

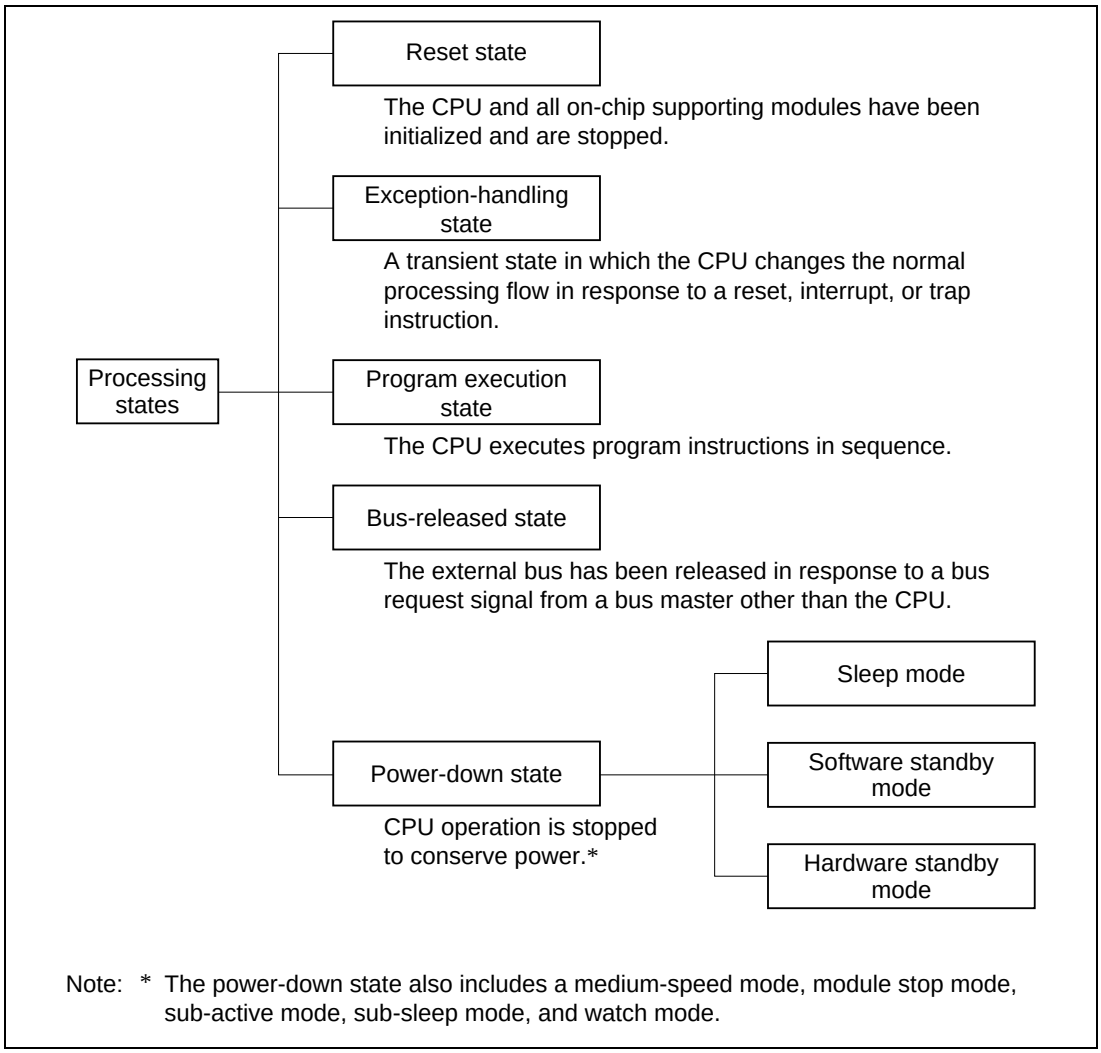
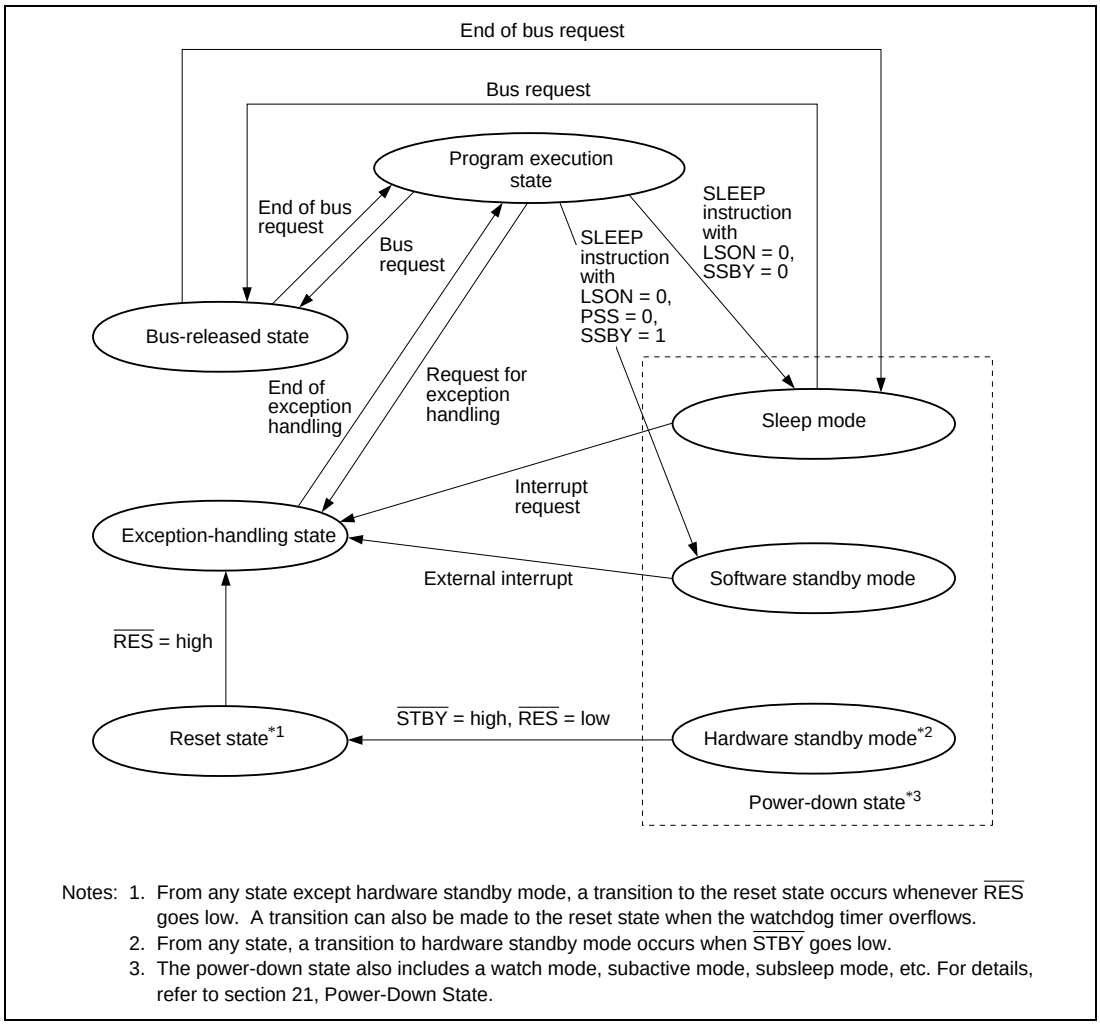


Figure 2.14 Processing States



Notes: 1. From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.
2. From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.
3. The power-down state also includes a watch mode, subactive mode, subsleep mode, etc. For details, refer to section 21, Power-Down State.

Figure 2.15 State Transitions

2.8.2 Reset State

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

The reset state can also be entered by a watchdog timer overflow. For details, refer to section 14, Watchdog Timer.

2.8.3 Exception-Handling State

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

Types of Exception Handling and Their Priority: Exception handling is performed for resets, interrupts, and trap instructions. Table 2.7 indicates the types of exception handling and their priority. Trap instruction exception handling is always accepted in the program execution state.

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

Table 2.7 Exception Handling Types and Priority

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

Notes: 1. Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC instructions, or immediately after reset exception handling.
2. Trap instruction exception handling is always accepted in the program execution state.

Reset Exception Handling: After the $\overline{\text{RES}}$ pin has gone low and the reset state has been entered, when $\overline{\text{RES}}$ goes high again, reset exception handling starts. When reset exception handling starts the CPU fetches a start address (vector) from the exception vector table and starts program execution from that address. All interrupts, including NMI, are disabled during reset exception handling and after it ends.

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

Figure 2.16 shows the stack after exception handling ends.

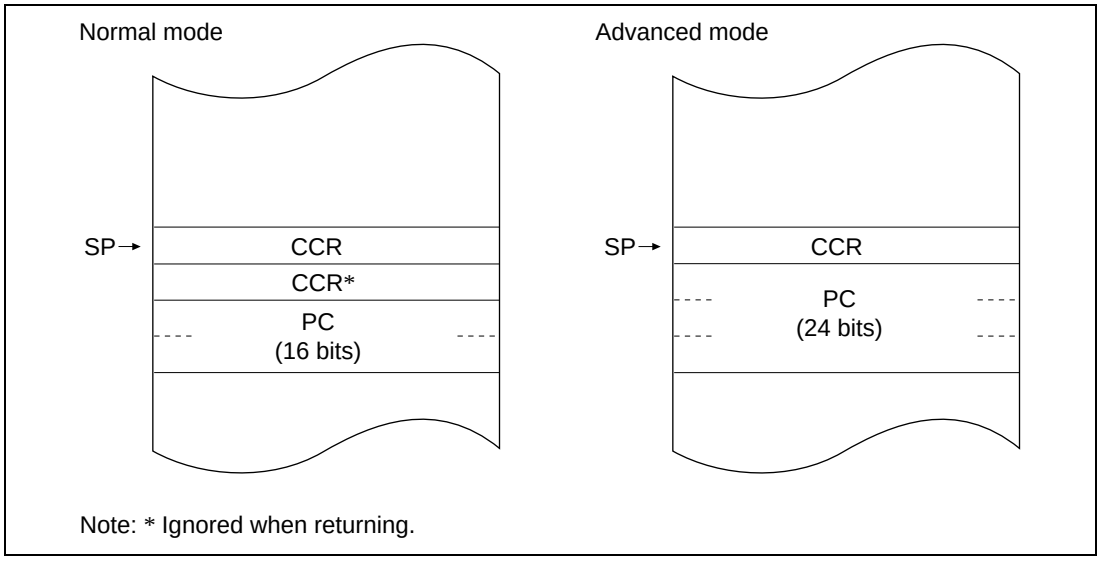


Figure 2.16 Stack Structure after Exception Handling (Examples)

2.8.4 Program Execution State

In this state the CPU executes program instructions in sequence.

2.8.5 Bus-Released State

This is a state in which the bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts except for internal operations.

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

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

2.8.6 Power-Down State

The power-down state includes both modes in which the CPU stops operating and modes in which the CPU does not stop. There are five modes in which the CPU stops operating: sleep mode, software standby mode, hardware standby mode, subsleep mode, and watch mode. There are also three other power-down modes: medium-speed mode, module stop mode, and subactive mode. In medium-speed mode, the CPU and other bus masters operate on a medium-speed clock. Module stop mode permits halting of the operation of individual modules, other than the CPU. Subactive mode, subsleep mode, and watch mode are power-down modes that use subclock input. For details, refer to section 21, Power-Down State.

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

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

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

2.9 Basic Timing

2.9.1 Overview

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

2.9.2 On-Chip Memory (ROM, RAM)

On-chip memory is accessed in one state. The data bus is 16 bits wide, permitting both byte and word transfer instruction. Figure 2.17 shows the on-chip memory access cycle. Figure 2.18 shows the pin states.

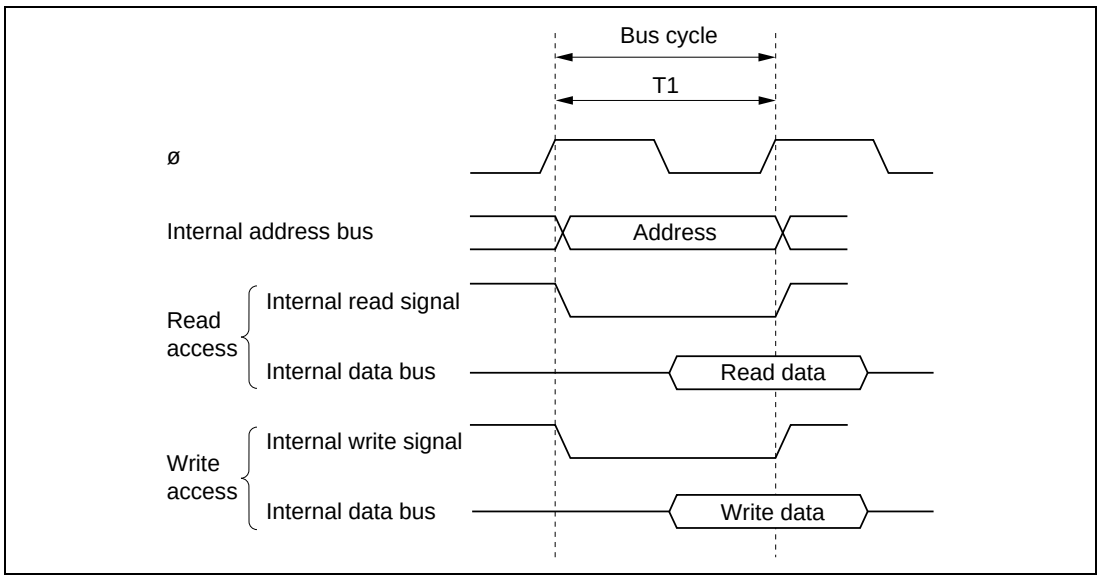


Figure 2.17 On-Chip Memory Access Cycle

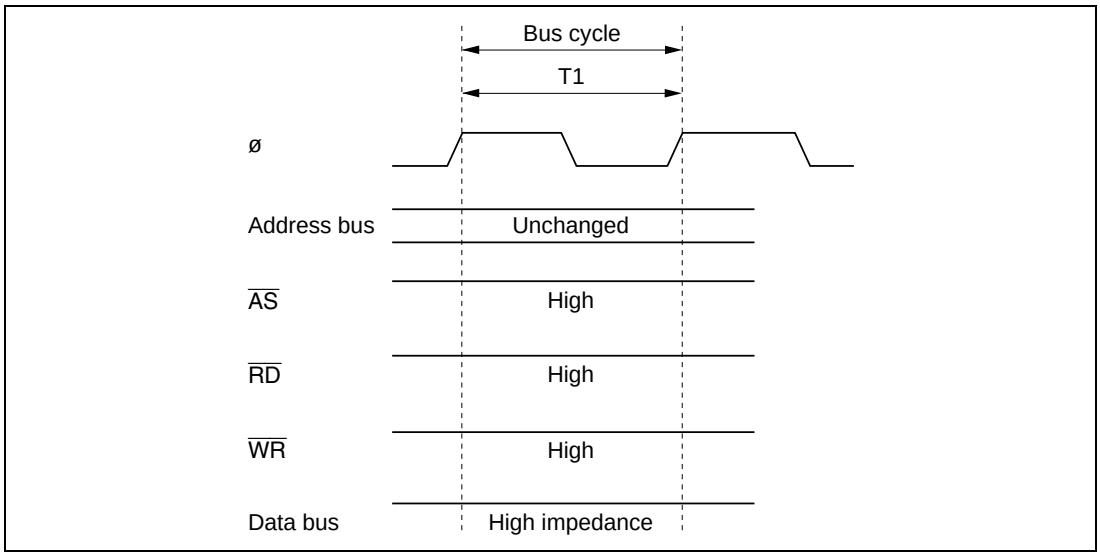


Figure 2.18 Pin States during On-Chip Memory Access

2.9.3 On-Chip Supporting Module Access Timing

The on-chip supporting modules are accessed in two states. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed. Figure 2.19 shows the access timing for the on-chip supporting modules. Figure 2.20 shows the pin states.

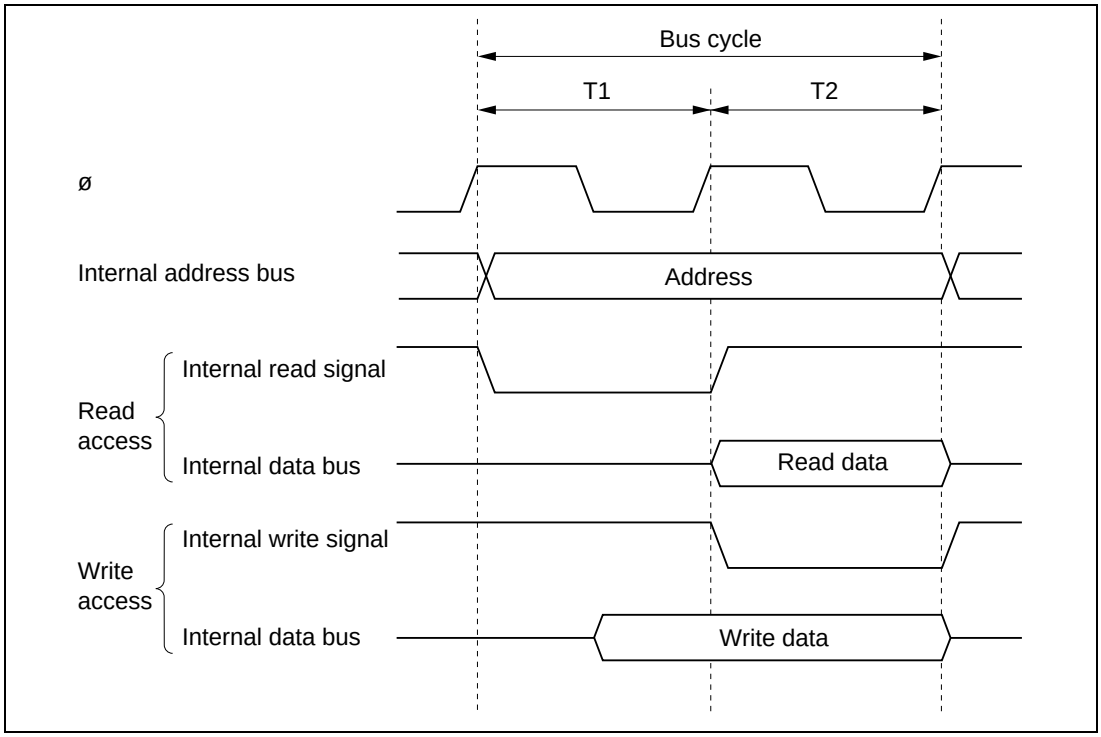


Figure 2.19 On-Chip Supporting Module Access Cycle

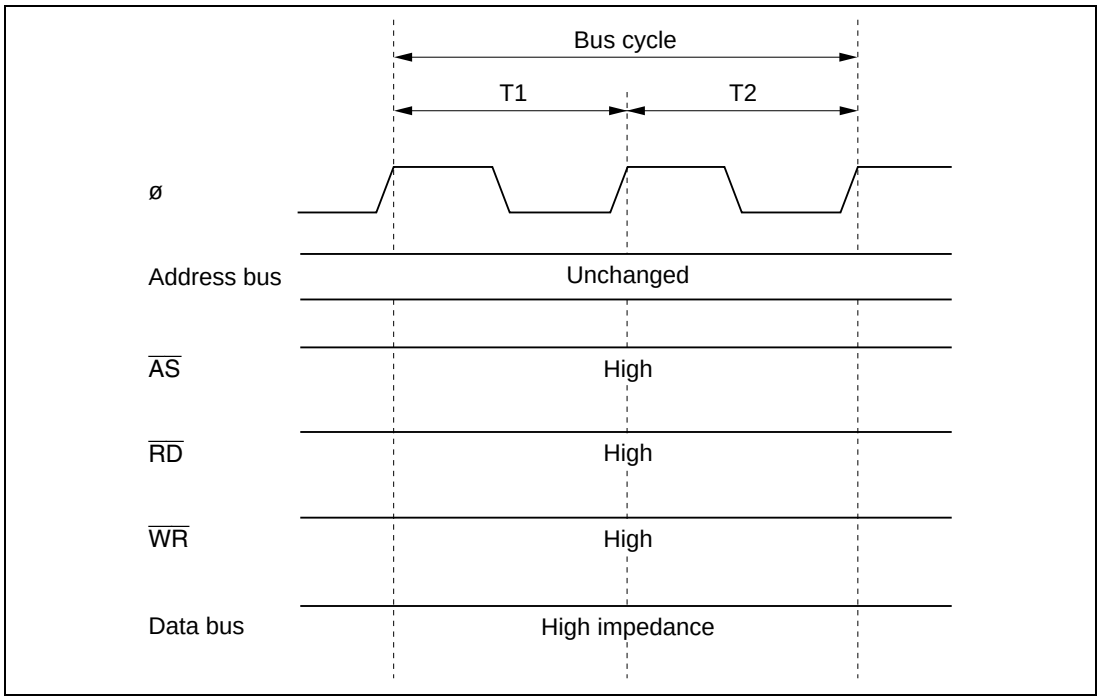


Figure 2.20 Pin States during On-Chip Supporting Module Access

2.9.4 External Address Space Access Timing

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

Section 3 MCU Operating Modes

3.1 Overview

3.1.1 Operating Mode Selection

The H8S/2128 Series and H8S/2124 Series have three operating modes (modes 1 to 3). These modes enable selection of the CPU operating mode and enabling/disabling of on-chip ROM, by setting the mode pins (MD1 and MD0).

Table 3.1 lists the MCU operating modes.

Table 3.1 MCU Operating Mode Selection

MCU Operating Mode	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM
0	0	0	—	—	—
1		1	Normal	Expanded mode with on-chip ROM disabled	Disabled
2	1	0	Advanced	Expanded mode with on-chip ROM enabled Single-chip mode	Enabled
3		1	Normal	Expanded mode with on-chip ROM enabled Single-chip mode	

The CPU's architecture allows for 4 Gbytes of address space, but the H8S/2128 Series and H8S/2124 Series actually access a maximum of 16 Mbytes. However, as there are 16 external address output pins, advanced mode is enabled only in single-chip mode or in expanded mode with on-chip ROM enabled when a specific area in the external address space is accessed using $\overline{\text{IOS}}$. The external data bus width is 8 bits.

Mode 1 is an externally expanded mode that allows access to external memory and peripheral devices. With modes 2 and 3, operation begins in single-chip mode after reset release, but a transition can be made to external expansion mode by setting the EXPE bit in MDCR.

The H8S/2128 Series and H8S/2124 Series can only be used in modes 1 to 3. These means that the mode pins must select one of these modes. Do not changes the inputs at the mode pins during operation.

3.1.2 Register Configuration

The H8S/2128 Series and H8S/2124 Series have a mode control register (MDCR) that indicates the inputs at the mode pins (MD1 and MD0), a system control register (SYSCR) and bus control register (BCR) that control the operation of the MCU, and a serial/timer control register (STCR) that controls the operation of the supporting modules. Table 3.2 summarizes these registers.

Table 3.2 MCU Registers

Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R/W	Undetermined	H'FFC5
System control register	SYSCR	R/W	H'09	H'FFC4
Bus control register	BCR	R/W	H'D7	H'FFC6
Serial/timer control register	STCR	R/W	H'00	H'FFC3

Note: *Lower 16 bits of the address.

3.2 Register Descriptions

3.2.1 Mode Control Register (MDCR)

Bit	7	6	5	4	3	2	1	0
	EXPE	—	—	—	—	—	MDS1	MDS0
Initial value	—*	0	0	0	0	0	—*	—*
Read/Write	R/W*	—	—	—	—	—	R	R

Note: * Determined by the MD1 and MD0 pins.

MDCR is an 8-bit read-only register that indicates the operating mode setting and the current operating mode of the MCU.

The EXPE bit is initialized in coordination with the mode pin states by a reset and in hardware standby mode.

Bit 7—Expanded Mode Enable (EXPE): Sets expanded mode. In mode 1, this bit is fixed at 1 and cannot be modified. In modes 2 and 3, this bit has an initial value of 0, and can be read and written.

Bit 7

EXPE	Description
0	Single chip mode is selected
1	Expanded mode is selected

Bits 6 to 2—Reserved: These bits cannot be modified and are always read as 0.

Bits 1 and 0—Mode Select 1 and 0 (MDS1, MDS0): These bits indicate the input levels at pins MD1 and MD0 (the current operating mode). Bits MDS1 and MDS0 correspond to MD1 and MD0. MDS1 and MDS0 are read-only bits—they cannot be written to. The mode pin (MD1 and MD0) input levels are latched into these bits when MDCR is read.

3.2.2 System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
	CS2E	IOSE	INTM1	INTM0	XRST	NMIEG	HIE	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R	R/W	R	R/W	R/W	R/W

SYSCR is an 8-bit readable/writable register that performs selection of system pin functions, reset source monitoring, interrupt control mode selection, NMI detected edge selection, supporting module register access control, and RAM address space control.

Only bits 7, 6, 3, 1, and 0 are described here. For a detailed description of these bits, refer also to the description of the relevant modules (bus controller, watchdog timer, RAM, etc.). For information on bits 5, 4, and 2, see section 5.2.1, System Control Register (SYSCR).

SYSCR is initialized to H'09 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Chip Select 2 Enable (CS2E): Specifies the location of the host interface control pin. As these series do not include an on-chip host interface, this bit should not be set to 1.

Bit 6—IOS Enable (IOSE): Controls the function of the $\overline{AS}/\overline{IOS}$ pin in expanded mode.

Bit 6

IOSE	Description
0	The $\overline{AS}/\overline{IOS}$ pin functions as the address strobe pin (Low output when accessing an external area) (Initial value)
1	The $\overline{AS}/\overline{IOS}$ pin functions as the I/O strobe pin (Low output when accessing a specified address from H'(FF)F000 to H'(FF)FE4F)

Bit 3—External Reset (XRST): Indicates the reset source. When the watchdog timer is used, a reset can be generated by watchdog timer overflow as well as by external reset input. XRST is a read-only bit. It is set to 1 by an external reset and cleared to 0 by watchdog timer overflow.

Bit 3

XRST	Description
0	A reset is generated by watchdog timer overflow
1	A reset is generated by an external reset (Initial value)

Bit 1—Host Interface Enable (HIE): Enables or disables CPU access to on-chip supporting function registers.

This bit controls CPU access to the 8-bit timer (channel X and Y) data registers and control registers (TCRX/TCRY, TCSRX/TCSRY, TICRR/TCORAY, TICRF/TCORBY, TCNTX/TCNTY, TCORC/TISR, TCORAX, and TCORBX), and the timer connection control registers (TCONRI, TCONRO, TCONRS, and SEDGR).

Bit 1

HIE	Description
0	In areas H'(FF)FFF0 to H'(FF)FFF7 and H'(FF)FFFC to H'(FF)FFFF, CPU access to 8-bit timer (channel X and Y) data registers and control registers, and timer connection control registers, is permitted (Initial value)
1	In areas H'(FF)FFF0 to H'(FF)FFF7 and H'(FF)FFFC to H'(FF)FFFF, CPU access to 8-bit timer (channel X and Y) data registers and control registers, and timer connection control registers, is not permitted

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

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

3.2.3 Bus Control Register (BCR)

Bit	7	6	5	4	3	2	1	0
	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0
Initial value	1	1	0	1	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCR is an 8-bit readable/writable register that specifies the external memory space access mode, and the I/O area range when the $\overline{AS}$ pin is designated for use as the I/O strobe. For details on bits 7 to 2, see section 6.2.1, Bus Control Register (BCR).

BCR is initialized to H'D7 by a reset and in hardware standby mode.

Bits 1 and 0—IOS Select 1 and 0 (IOS1, IOS0): These bits specify the addresses for which the $\overline{AS}$/$\overline{IOS}$ pin output goes low when IOSE = 1.

	BCR	
Bit 1	**Bit 0**	
IOS1	**IOS0**	**Description**
0	0	The $\overline{AS}$/$\overline{IOS}$ pin output goes low in accesses to addresses H'(FF)F000 to H'(FF)F03F
	1	The $\overline{AS}$/$\overline{IOS}$ pin output goes low in accesses to addresses H'(FF)F000 to H'(FF)F0FF
1	0	The $\overline{AS}$/$\overline{IOS}$ pin output goes low in accesses to addresses H'(FF)F000 to H'(FF)F3FF
	1	The $\overline{AS}$/$\overline{IOS}$ pin output goes low in accesses to addresses H'(FF)F000 to H'(FF)FE4F (Initial value)

3.2.4 Serial Timer Control Register (STCR)

Bit	7	6	5	4	3	2	1	0
	—	IICX1	IICX0	IICE	FLSHE	—	ICKS1	ICKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

STCR is an 8-bit readable/writable register that controls register access, the IIC operating mode (when the on-chip IIC option is included), an on-chip flash memory (in F-ZTAT versions), and also selects the TCNT input clock. For details of functions other than register access control, see the descriptions of the relevant modules. If a module controlled by STCR is not used, do not write 1 to the corresponding bit.

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

Bits 7 to 5—I^2C Control (IICS, IICX1, IICX0): These bits control the operation of the I^2C bus interface when the on-chip IIC option is included. For details, see section 16, I^2C Bus Interface.

Bit 4—I^2C Master Enable (IICE): Controls CPU access to the I^2C bus interface data registers and control registers (ICCR, ICSR, ICDR/SARX, and ICMR/SAR), the PWMX data registers and control registers (DADRAH/DACR, DADRAL, DADRBH/DACNTH, and DADRBL/DACNTL), and the SCI control registers (SMR, BRR, and SCMR).

Bit 4

IICE	Description	
0	Addresses H'(FF)FF88 and H'(FF)89, and H'(FF)FF8E and H'(FF)FF8F, are used for SCI1 control register access	(Initial value)
	Addresses H'(FF)FFD8 and H'(FF)FFD9, and H'(FF)FFDE and H'(FF)FFDF, are used for SCI0 control register access	
1	Addresses H'(FF)FF88 and H'(FF)FF89, and H'(FF)FF8E and H'(FF)FF8F, are used for IIC1 data register and control register access	
	Addresses H'(FF)FFA0 and H'(FF)FFA1, and H'(FF)FFA6 and H'(FF)FFA7, are used for PWMX data register and control register access	
	Addresses H'(FF)FFD8 and H'(FF)FFD9, and H'(FF)FFDE and H'(FF)FFDF, are used for IIC0 data register and control register access	

Bit 3—Flash Memory Control Register Enable (FLSHE): Controls CPU access to the flash memory control registers (FLMCR1, FLMCR2, EBR1, and EBR2), the power-down mode control registers (SBYCR, LPWRCR, MSTPCRH, and MSTPCRL), and the supporting module control register (PCSR).

Bit 3

FLSHE	Description	
0	Addresses H'(FF)F80 to H'(FF)F87 are used for power-down mode control register and supporting module control register access	(Initial value)
1	Addresses H'(FF)FF80 to H'(FF)FF87 are used for flash memory control register access	

Bit 2—Reserved: Do not write 1 to this bit.

Bits 1 and 0—Internal Clock Source Select 1 and 0 (ICKS1, ICKS0): These bits, together with bits CKS2 to CKS0 in TCR, select the clock to be input to TCNT. For details, see section 12, 8-Bit Timers.

3.3 Operating Mode Descriptions

3.3.1 Mode 1

The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is disabled.

Ports 1 and 2 function as an address bus, port 3 function as a data bus, and part of port 4 carries bus control signals.

3.3.2 Mode 2

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled. After a reset, single-chip mode is set, and the EXPE bit in MDCR must be set to 1 in order to use external addresses. However, as these series have a maximum of 16 address outputs, an external address can be specified correctly only when the I/O strobe function of the $\overline{AS}$/$\overline{IOS}$ pin is used.

When the EXPE bit in MDCR is set to 1, ports 1 and 2 function as input ports after a reset. They can be set to output addresses by setting the corresponding bits in the data direction register (DDR) to 1. Port 3 function as a data bus, and part of port 4 carries bus control signals.

3.3.3 Mode 3

The CPU can access a 64-kbyte address space in normal mode. The on-chip ROM is enabled. After a reset, single-chip mode is set, and the EXPE bit in MDCR must be set to 1 in order to use external addresses.

When the EXPE bit in MDCR is set to 1, ports 1 and 2 function as input ports after a reset. They can be set to output addresses by setting the corresponding bits in the data direction register (DDR) to 1. Port 3 function as a data bus, and part of port 4 carries bus control signals.

In products with an on-chip ROM capacity of 64 kbytes or more, the amount of on-chip ROM that can be used is limited to 56 kbytes.

3.4 Pin Functions in Each Operating Mode

The pin functions of ports 1 to 4 vary depending on the operating mode. Table 3.3 shows their functions in each operating mode.

Table 3.3 Pin Functions in Each Mode

Port		Mode 1	Mode 2	Mode 3
Port 1		A	P*/A	P*/A
Port 2		A	P*/A	P*/A
Port 3		D	P*/D	P*/D
Port 4	P47	P*/C	P*/C	P*/C
	P46	C */P	P*/C	P*/C
	P45 to P43	C	P*/C	P*/C
	P42 to P40	P	P	P

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

3.5 Memory Map in Each Operating Mode

Figures 3.1 to 3.4 show memory maps for each of the operating modes.

The address space is 64 kbytes in modes 1 and 3 (normal modes), and 16 Mbytes in mode 2 (advanced mode).

The on-chip ROM capacity is 32 kbytes (H8S/2126 and H8S/2120), 64 kbytes (H8S/2127 and H8S/2122), 96 kbytes (H8S/2123), or 128 kbytes (H8S/2128 and H8S/2124), but for products with an on-chip ROM capacity of 64 kbytes or more, the amount of on-chip ROM that can be used is limited to 56 kbytes in mode 3 (normal mode).

For details, see section 6, Bus Controller.

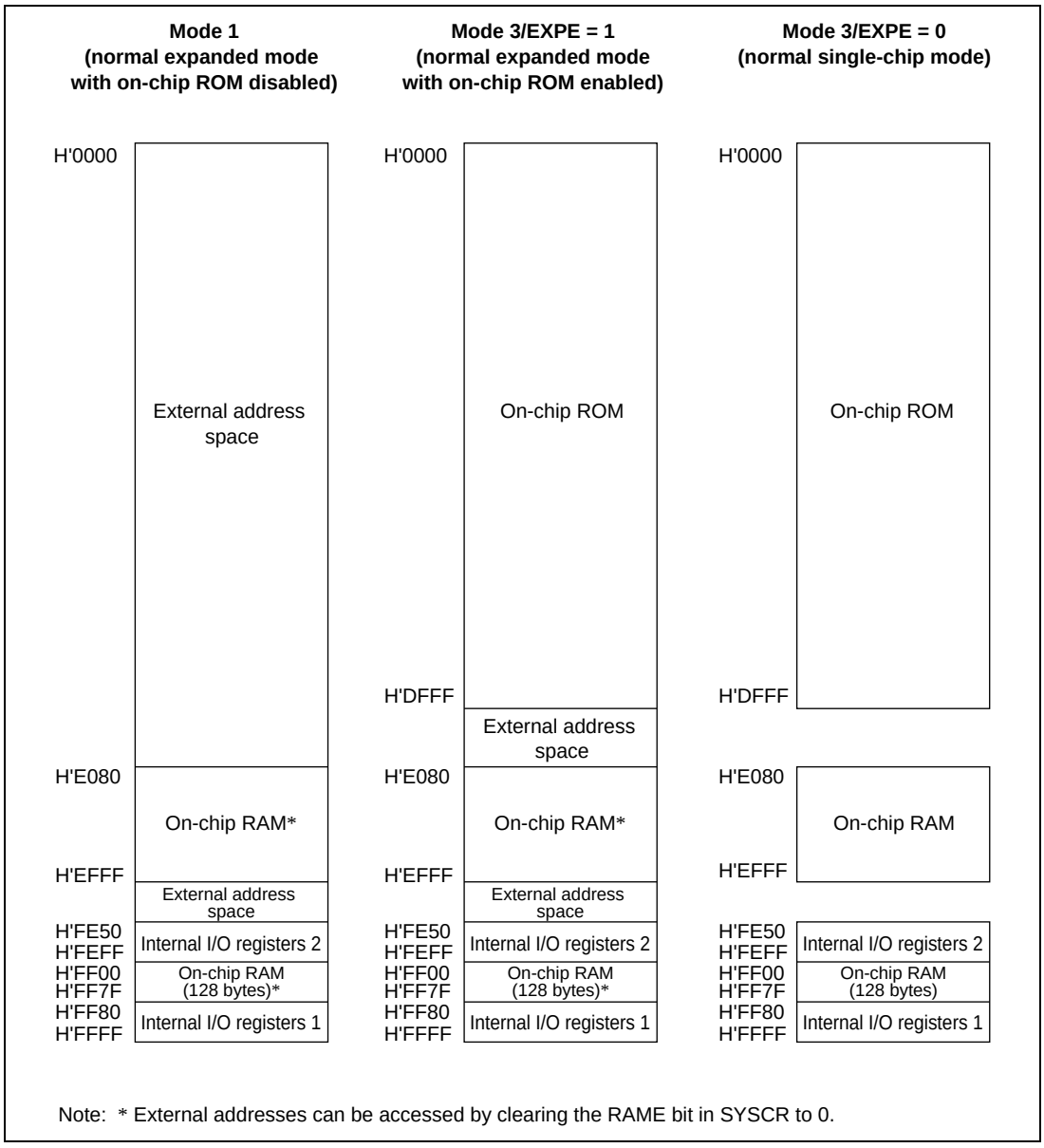
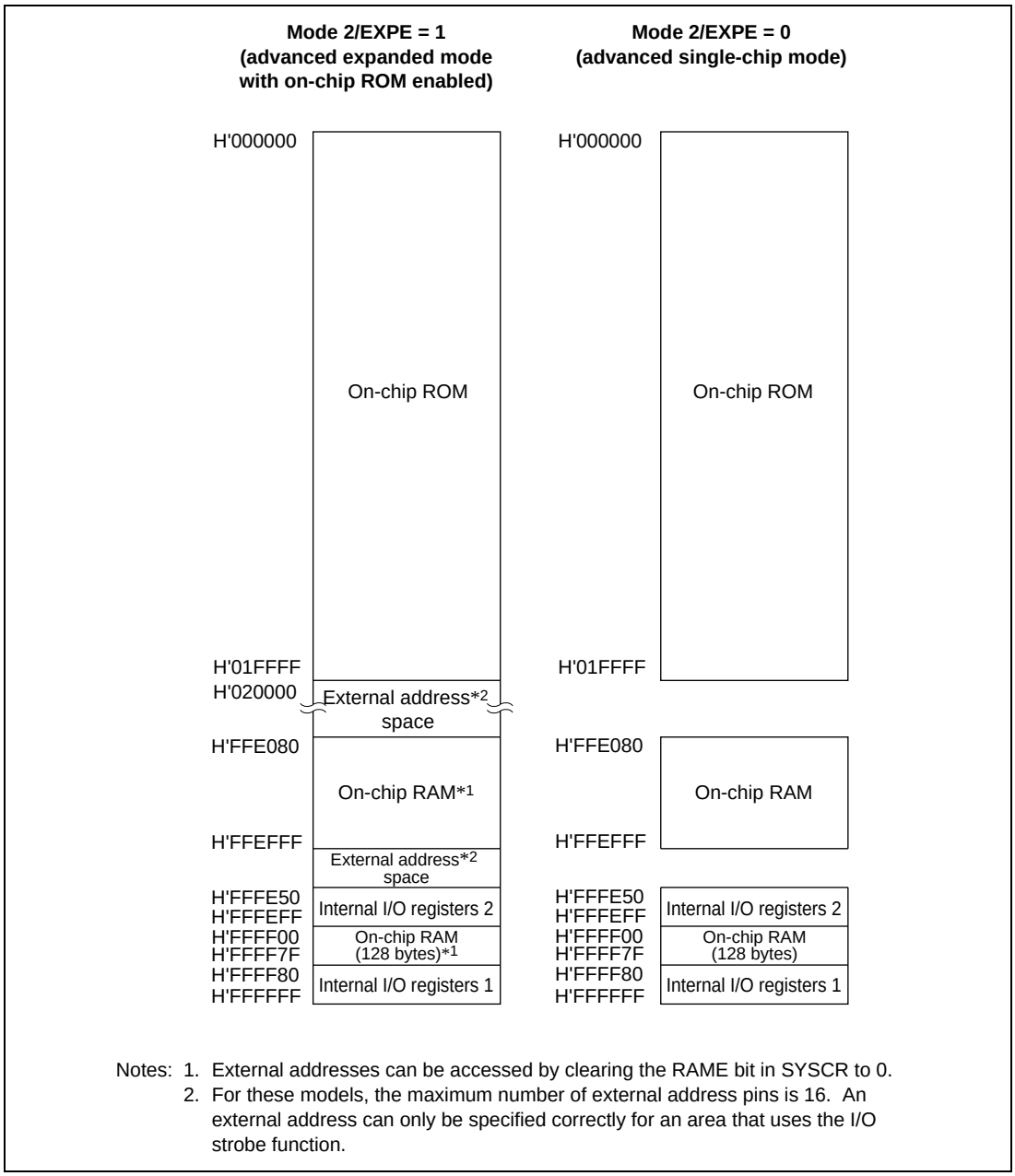


Figure 3.1 H8S/2128 and H8S/2124 Memory Map in Each Operating Mode



Notes: 1. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
2. For these models, the maximum number of external address pins is 16. An external address can only be specified correctly for an area that uses the I/O strobe function.

Figure 3.1 H8S/2128 and H8S/2124 Memory Map in Each Operating Mode (cont)

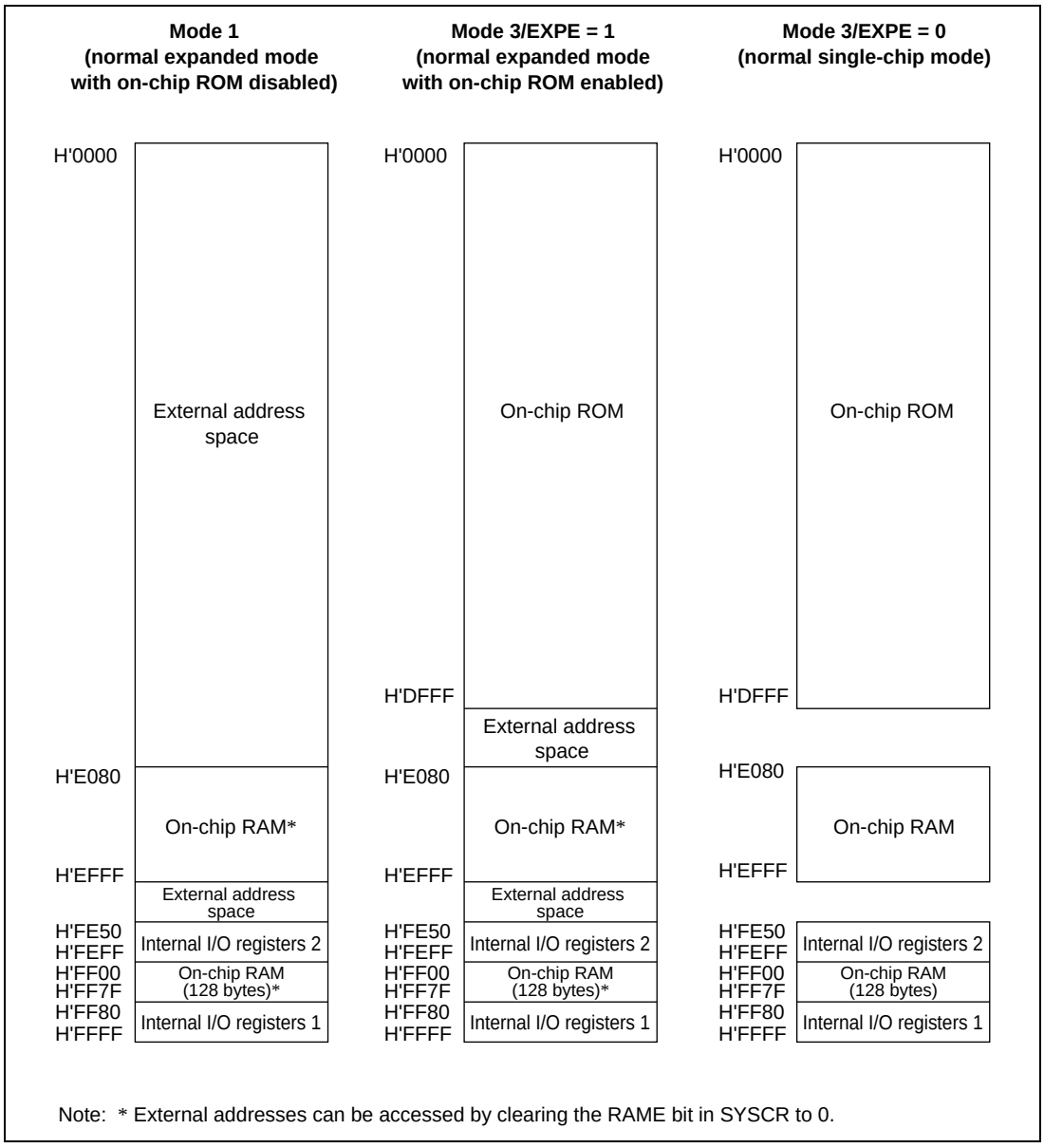
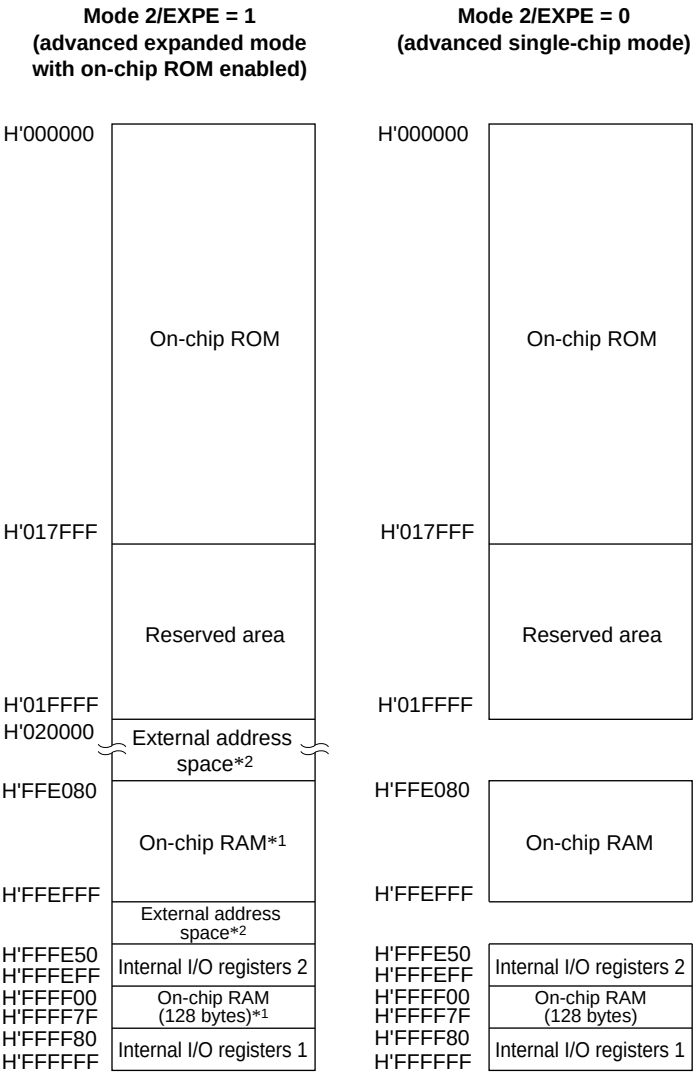


Figure 3.2 H8S/2123 Memory Map in Each Operating Mode



Notes: 1. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
2. For these models, the maximum number of external address pins is 16. An external address can only be specified correctly for an area that uses the I/O strobe function.

Figure 3.2 H8S/2123 Memory Map in Each Operating Mode (cont)

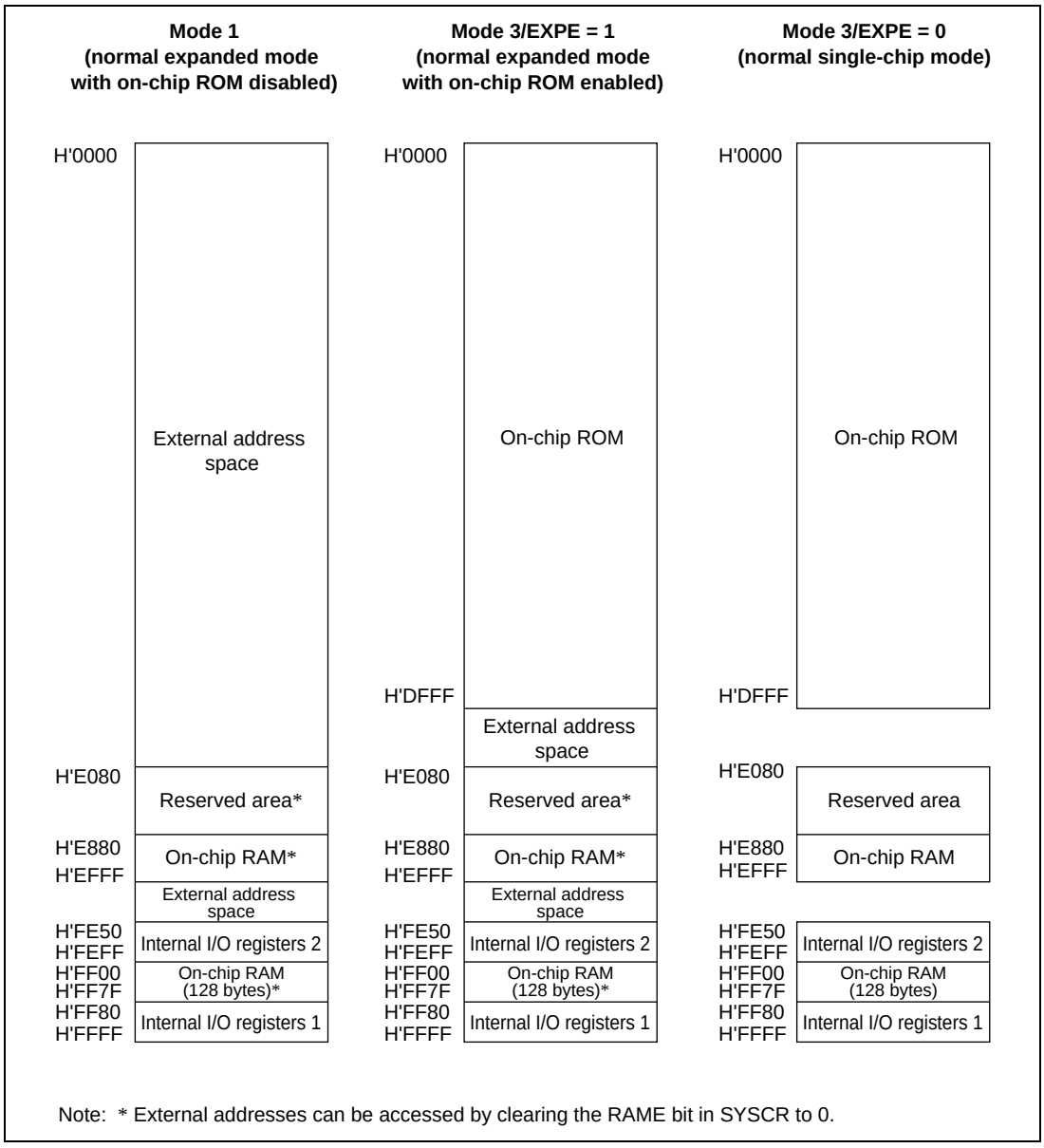
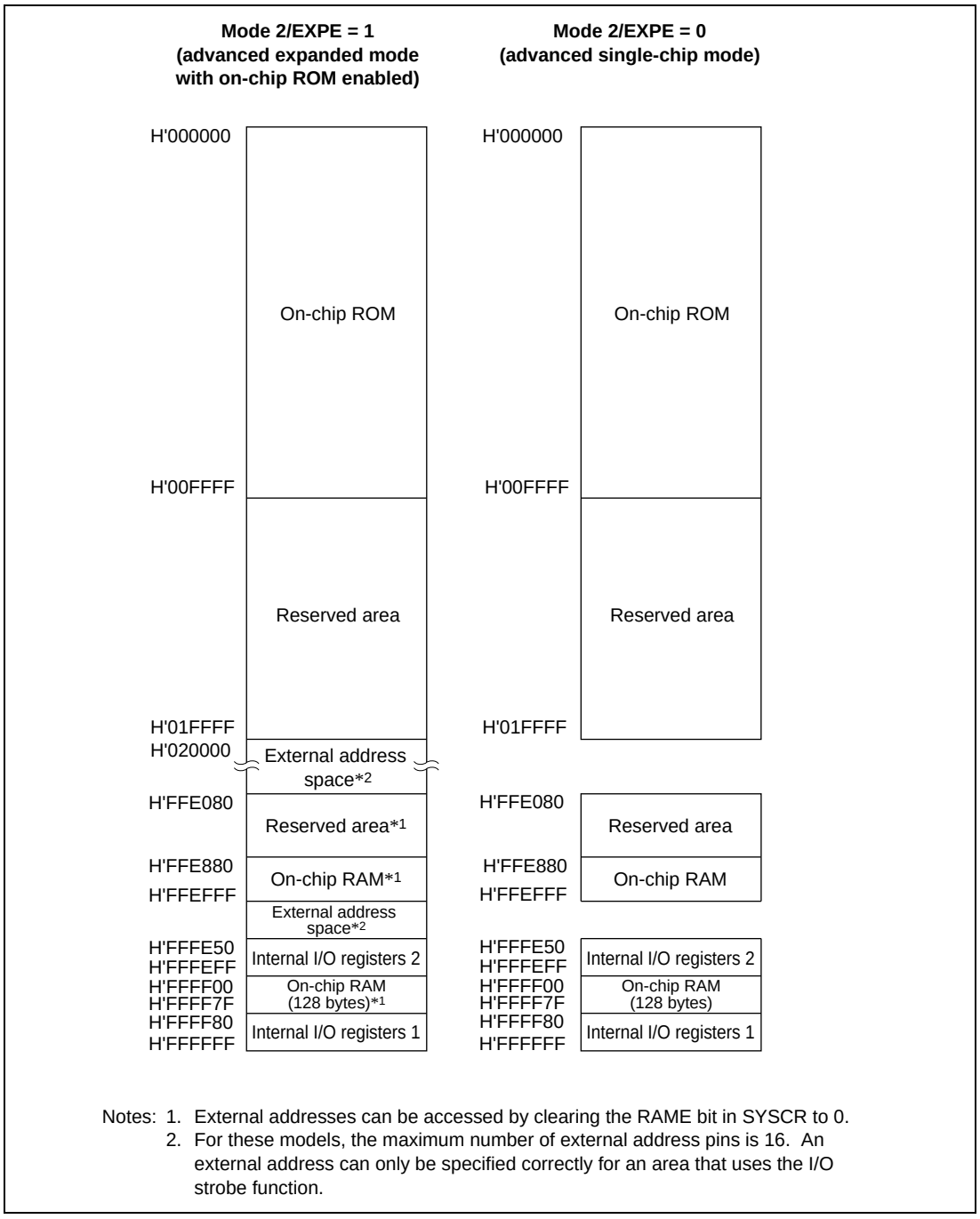


Figure 3.3 H8S/2127 and H8S/2122 Memory Map in Each Operating Mode



Notes: 1. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
2. For these models, the maximum number of external address pins is 16. An external address can only be specified correctly for an area that uses the I/O strobe function.

Figure 3.3 H8S/2127 and H8S/2122 Memory Map in Each Operating Mode (cont)

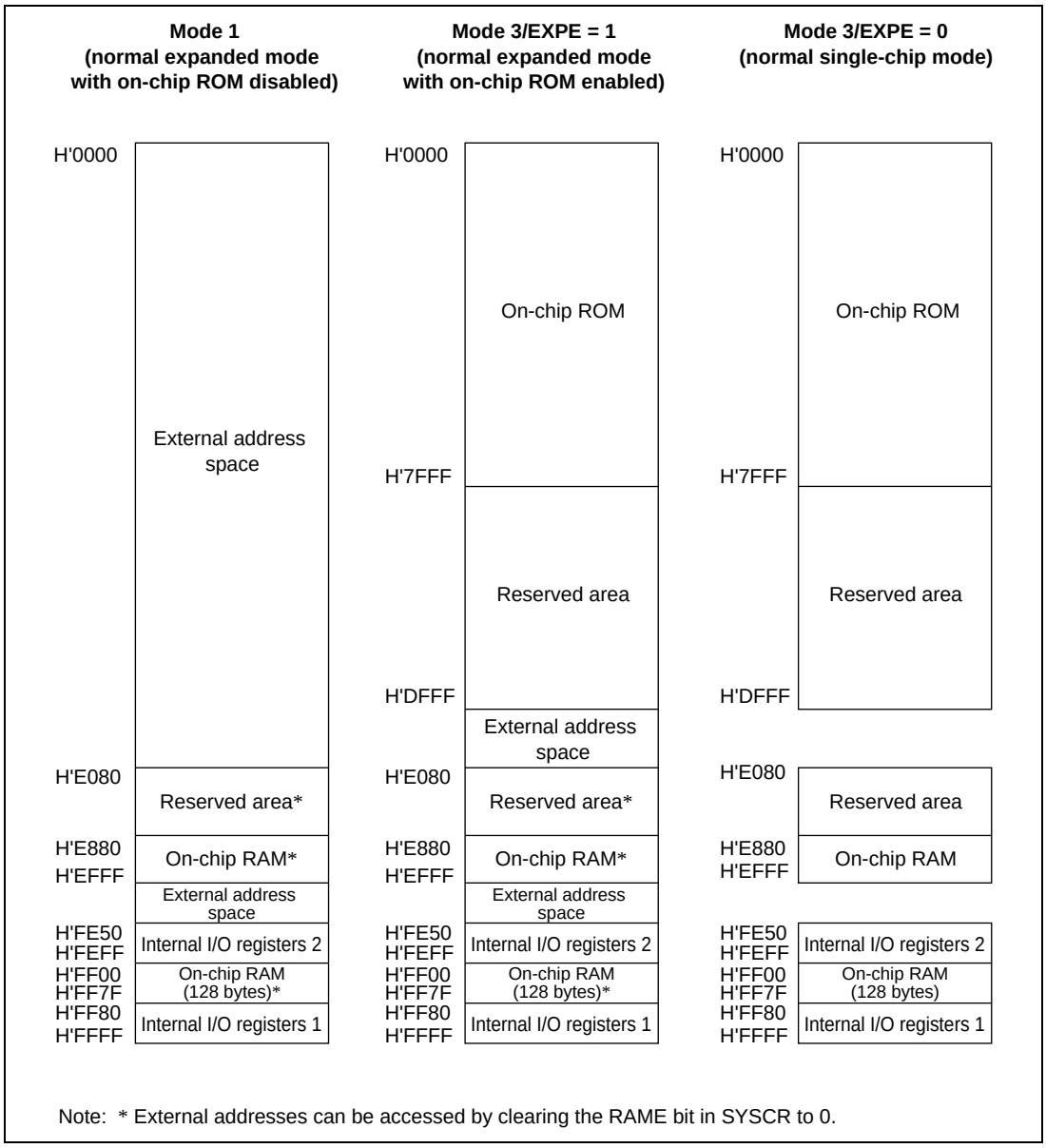
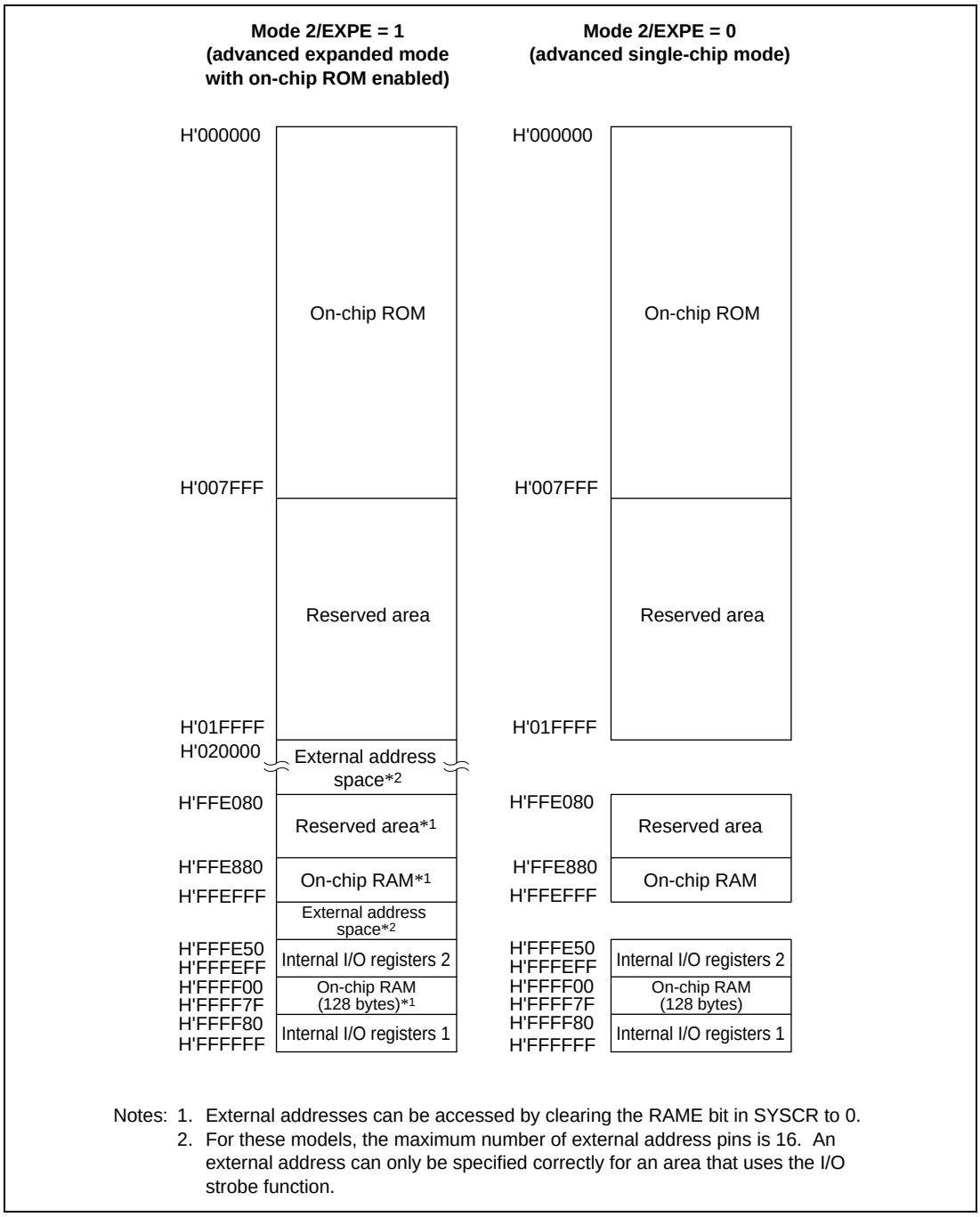


Figure 3.4 H8S/2126 and H8S/2120 Memory Map in Each Operating Mode



Notes: 1. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
2. For these models, the maximum number of external address pins is 16. An external address can only be specified correctly for an area that uses the I/O strobe function.

Figure 3.4 H8S/2126 and H8S/2120 Memory Map in Each Operating Mode (cont)

Section 4 Exception Handling

4.1 Overview

4.1.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, trap instruction, or interrupt. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority. Trap instruction exceptions are accepted at all times in the program execution state.

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

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows.
	Trace*[1]	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit is set to 1.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued.*[2]
	Direct transition	Started by a direct transition resulting from execution of a SLEEP instruction.
Low	Trap instruction (TRAPA)*[3]	Started by execution of a trap instruction (TRAPA).

Notes: 1. Traces are enabled only in interrupt control modes 2 and 3. (They cannot be used in the H8S/2128 Series or H8S/2124 Series.) Trace exception handling is not executed after execution of an RTE instruction.

2. Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.

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

4.1.2 Exception Handling Operation

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

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

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

4.1.3 Exception Sources and Vector Table

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

Table 4.2 lists the exception sources and their vector addresses.

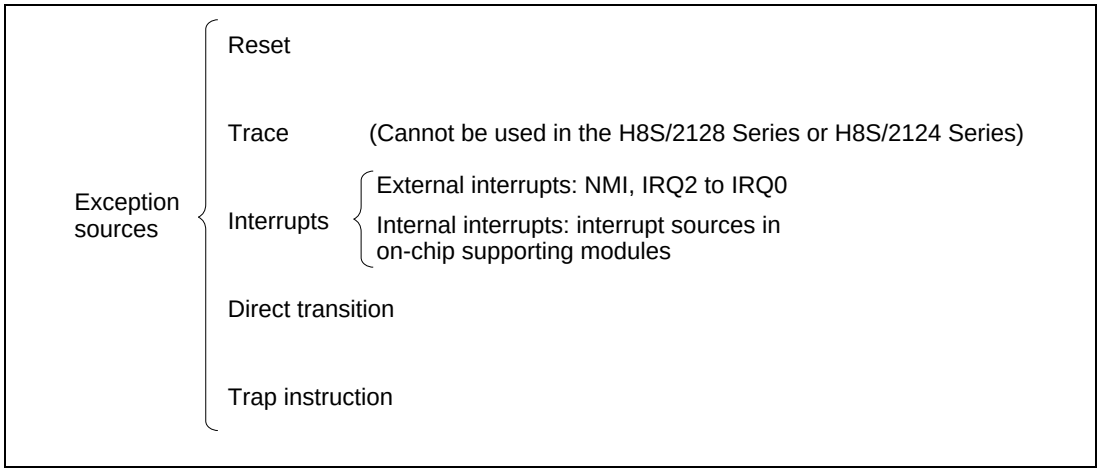


Figure 4.1 Exception Sources

Table 4.2 Exception Vector Table

Exception Source		Vector Number	Vector Address*[1] Normal Mode	Vector Address*[1] Advanced Mode
Reset		0	H'0000 to H'0001	H'0000 to H'0003
Reserved for system use		1	H'0002 to H'0003	H'0004 to H'0007
		2	H'0004 to H'0005	H'0008 to H'000B
		3	H'0006 to H'0007	H'000C to H'000F
		4	H'0008 to H'0009	H'0010 to H'0013
		5	H'000A to H'000B	H'0014 to H'0017
Direct transition		6	H'000C to H'000D	H'0018 to H'001B
External interrupt	NMI	7	H'000E to H'000F	H'001C to H'001F
Trap instruction (4 sources)		8	H'0010 to H'0011	H'0020 to H'0023
		9	H'0012 to H'0013	H'0024 to H'0027
		10	H'0014 to H'0015	H'0028 to H'002B
		11	H'0016 to H'0017	H'002C to H'002F
Reserved for system use		12	H'0018 to H'0019	H'0030 to H'0033
		13	H'001A to H'001B	H'0034 to H'0037
		14	H'001C to H'001D	H'0038 to H'003B
		15	H'001E to H'001F	H'003C to H'003F
External interrupt	IRQ0	16	H'0020 to H'0021	H'0040 to H'0043
	IRQ1	17	H'0022 to H'0023	H'0044 to H'0047
	IRQ2	18	H'0024 to H'0025	H'0048 to H'004B
Reserved		19	H'0026 to H'0027	H'004C to H'004F
		20	H'0028 to H'0029	H'0050 to H'0053
		21	H'002A to H'002B	H'0054 to H'0057
		22	H'002C to H'002D	H'0058 to H'005B
		23	H'002E to H'002F	H'005C to H'005F
Internal interrupt*[2]		24 | 103	H'0030 to H'0031 | H'00CE to H'00CF	H'0060 to H'0063 | H'019C to H'019F

Notes: 1. Lower 16 bits of the address.

2. For details on internal interrupt vectors, see section 5.3.3, Interrupt Exception Vector Table.

4.2 Reset

4.2.1 Overview

A reset has the highest exception priority.

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

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

H8S/2128 Series and H8S/2124 Series MCUs can also be reset by overflow of the watchdog timer. For details, see section 14, Watchdog Timer.

4.2.2 Reset Sequence

The MCU enters the reset state when the $\overline{\text{RES}}$ pin goes low.

To ensure that the chip is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms when powering on. To reset the chip during operation, hold the $\overline{\text{RES}}$ pin low for at least 20 states. For pin states in a reset, see Appendix D.1, Port States in Each Processing State.

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

[1] The internal state of the CPU and the registers of the on-chip supporting modules are initialized, and the I bit is set to 1 in CCR.

[2] The reset exception vector address is read and transferred to the PC, and program execution starts from the address indicated by the PC.

Figures 4.2 and 4.3 show examples of the reset sequence.

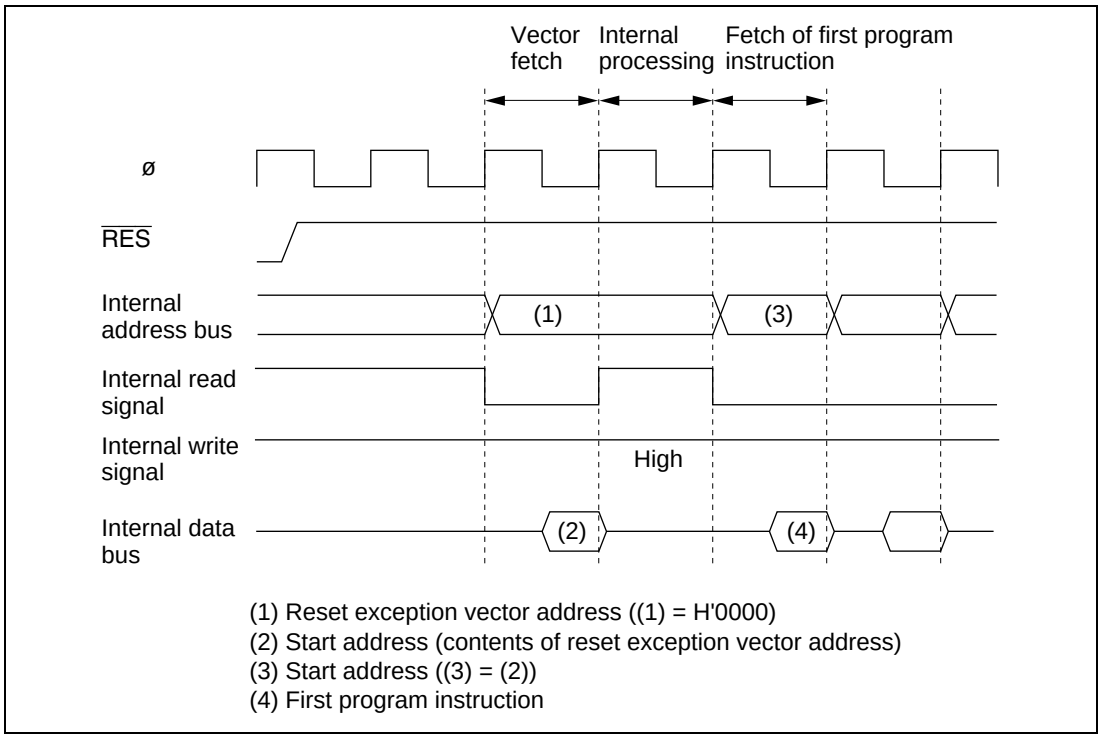
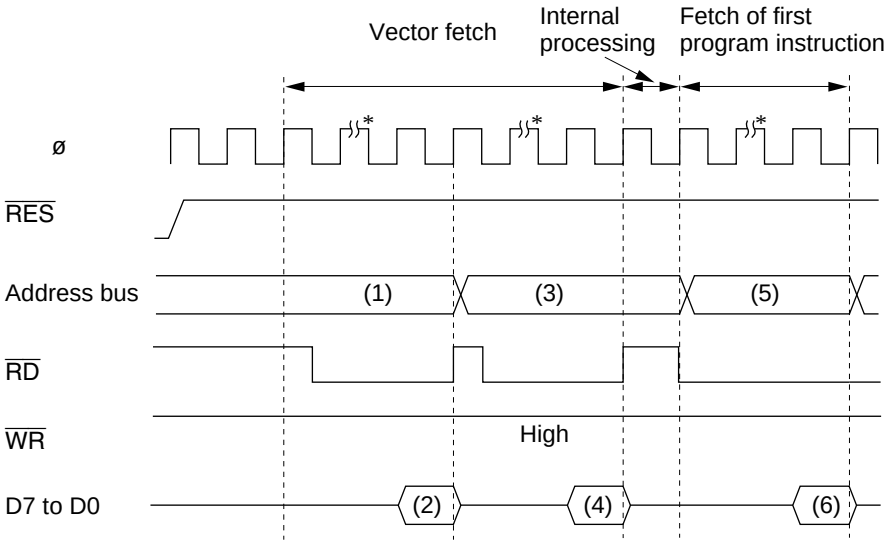


Figure 4.2 Reset Sequence (Mode 3)



(1) (3) Reset exception vector address ((1) = H'0000, (3) = H'0001)
(2) (4) Start address (contents of reset exception 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 1)

4.2.3 Interrupts after Reset

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

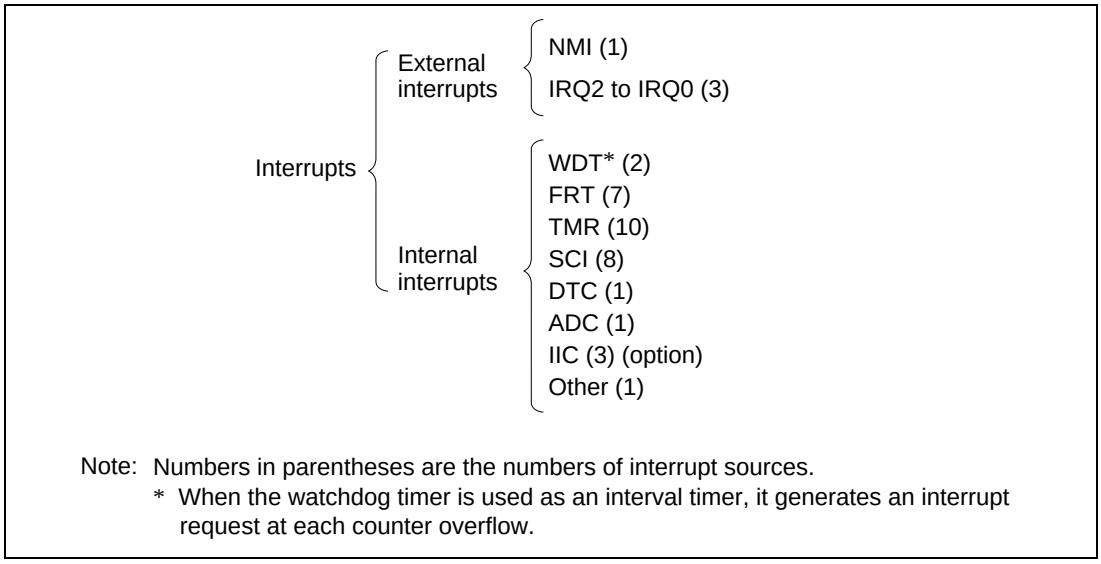
4.3 Interrupts

Interrupt exception handling can be requested by four external sources (NMI and IRQ2 to IRQ0), and internal sources in the on-chip supporting modules. Figure 4.4 shows the interrupt sources and the number of interrupts of each type.

The on-chip supporting modules that can request interrupts include the watchdog timer (WDT), 16-bit free-running timer (FRT), 8-bit timer (TMR), serial communication interface (SCI), data transfer controller (DTC), A/D converter (ADC), I^2C bus interface (option). Each interrupt source has a separate vector address.

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

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



Note: Numbers in parentheses are the numbers of interrupt sources.
* When the watchdog timer is used as an interval timer, it generates an interrupt request at each counter overflow.

Figure 4.4 Interrupt Sources and Number of Interrupts

4.4 Trap Instruction

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

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

Table 4.3 shows the status of CCR and EXR after execution of trap instruction exception handling.

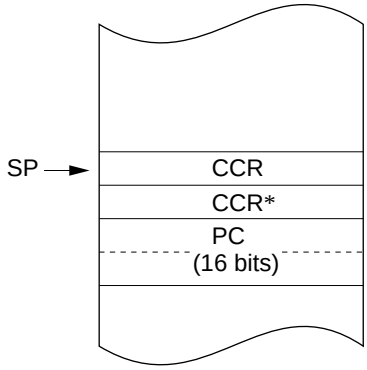
Table 4.3 Status of CCR and EXR after Trap Instruction Exception Handling

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

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

4.5 Stack Status after Exception Handling

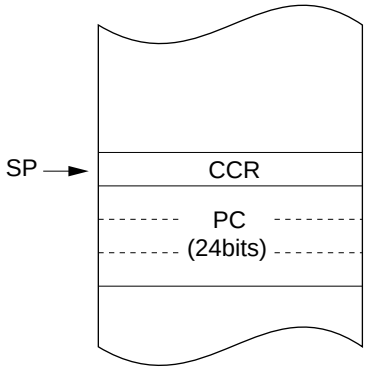
Figure 4.5 shows the stack after completion of trap instruction exception handling and interrupt exception handling.



Interrupt control modes 0 and 1

Note: * Ignored on return.

Figure 4.5 (1) Stack Status after Exception Handling (Normal Mode)



Interrupt control modes 0 and 1

Note: * Ignored on return.

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

4.6 Notes on Use of the Stack

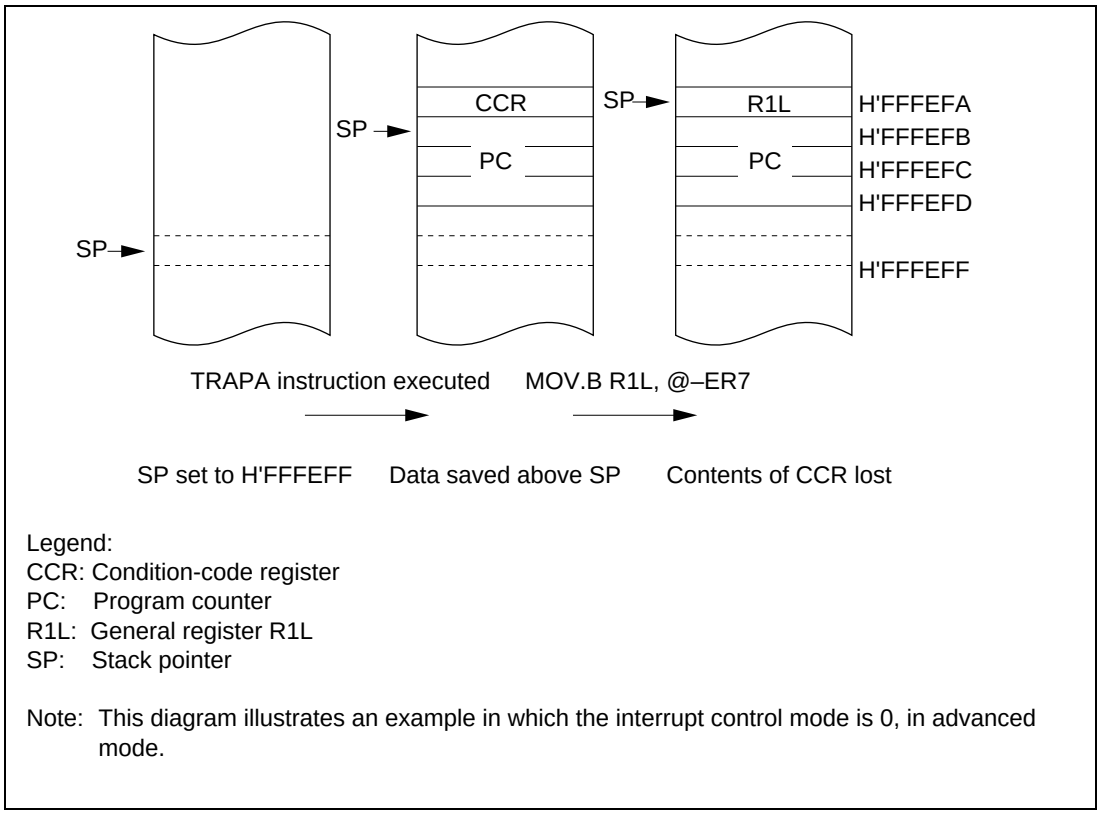
When accessing word data or longword data, the H8S/2128 Series or H8S/2124 Series chip assumes that the lowest address bit is 0. The stack should always be accessed by word transfer instruction or longword transfer instruction, and the value of the stack pointer (SP: ER7) should always be kept even. Use the following instructions to save registers:

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

Use the following instructions to restore registers:

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

Setting SP to an odd value may lead to a malfunction. Figure 4.6 shows an example of what 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 advanced mode.

Figure 4.6 Operation when SP Value is Odd

Section 5 Interrupt Controller

5.1 Overview

5.1.1 Features

H8S/2128 Series and H8S/2124 Series MCUs control interrupts by means of an interrupt controller. The interrupt controller has the following features:

- Two interrupt control modes
  - Either of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR).
- Priorities settable with ICR
  - An interrupt control register (ICR) is provided for setting interrupt priorities. Three priority levels can be set for each module for all interrupts except NMI.
- Independent vector addresses
  - All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Four external interrupt pins
  - NMI is the highest-priority interrupt, and is accepted at all times. A rising or falling edge at the NMI pin can be selected for the NMI interrupt.
  - Falling edge, rising edge, or both edge detection, or level sensing, at pins $\overline{\text{IRQ2}}$ to $\overline{\text{IRQ0}}$ can be selected for interrupts IRQ2 to IRQ0.
- DTC control
  - DTC activation is controlled by means of interrupts.

5.1.2 Block Diagram

A block diagram of the interrupt controller is shown in Figure 5.1.

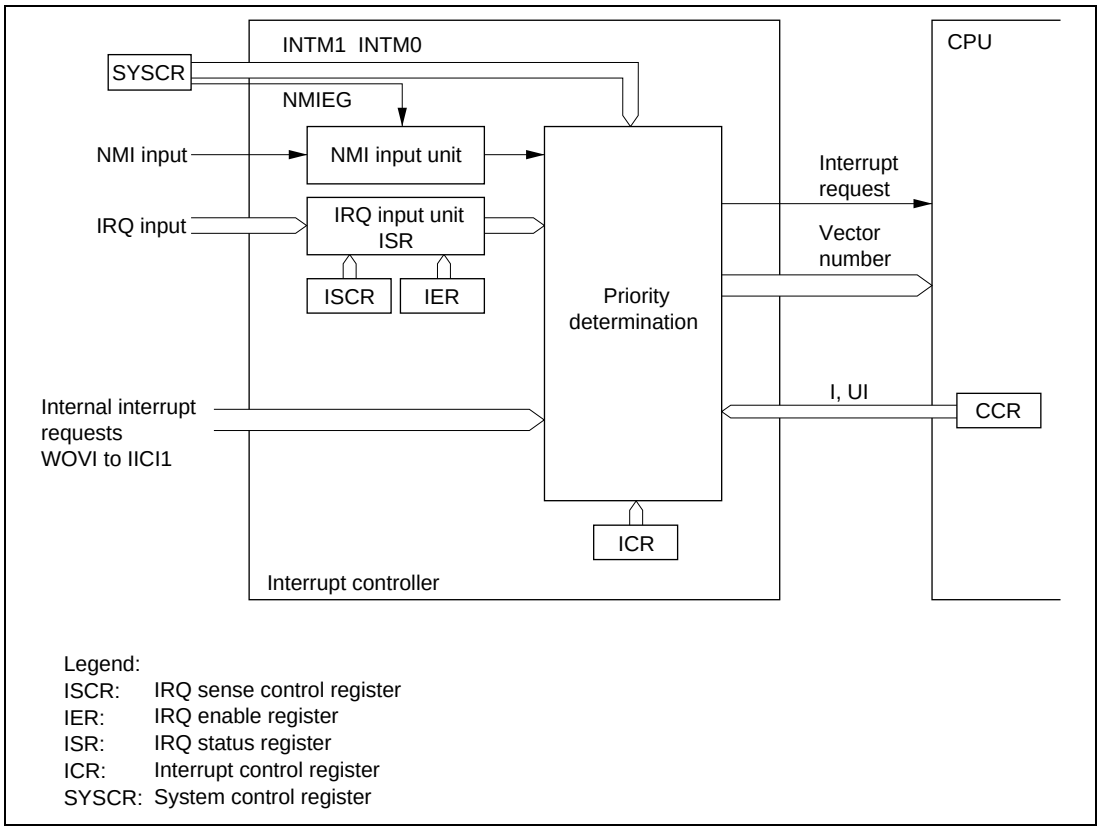


Figure 5.1 Block Diagram of Interrupt Controller

5.1.3 Pin Configuration

Table 5.1 summarizes the pins of the interrupt controller.

Table 5.1 Interrupt Controller Pins

Name	Symbol	I/O	Function
Nonmaskable interrupt	NMI	Input	Nonmaskable external interrupt; rising or falling edge can be selected
External interrupt requests 2 to 0	$\overline{\text{IRQ2}}$ to $\overline{\text{IRQ0}}$	Input	Maskable external interrupts; rising, falling, or both edges, or level sensing, can be selected

5.1.4 Register Configuration

Table 5.2 summarizes the registers of the interrupt controller.

Table 5.2 Interrupt Controller Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
System control register	SYSCR	R/W	H'09	H'FFC4
IRQ sense control register H	ISCRH	R/W	H'00	H'FEEC
IRQ sense control register L	ISCRL	R/W	H'00	H'FEED
IRQ enable register	IER	R/W	H'F8	H'FFC2
IRQ status register	ISR	R/(W)*[2]	H'00	H'FEEB
Interrupt control register A	ICRA	R/W	H'00	H'FEE8
Interrupt control register B	ICRB	R/W	H'00	H'FEE9
Interrupt control register C	ICRC	R/W	H'00	H'FEEA
Address break control register	ABRKCR	R/W	H'00	H'FEF4
Break address register A	BARA	R/W	H'00	H'FEF5
Break address register B	BARB	R/W	H'00	H'FEF6
Break address register C	BARC	R/W	H'00	H'FEF7

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written, for flag clearing.

5.2 Register Descriptions

5.2.1 System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
	CS2E	IOSE	INTM1	INTM0	XRST	NMIEG	HIE	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R	R/W	R	R/W	R/W	R/W

SYSCR is an 8-bit readable/writable register of which bits 5, 4, and 2 select the interrupt control mode and the detected edge for NMI.

Only bits 5, 4, and 2 are described here; for details on the other bits, see section 3.2.2, System Control Register (SYSCR).

SYSCR is initialized to H'09 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select one of four interrupt control modes for the interrupt controller. The INTM1 bit must not be set to 1.

Bit 5	Bit 4	Interrupt		
INTM1	**INTM0**	**Control Mode**	**Description**	
0	0	0	Interrupts are controlled by I bit	(Initial value)
	1	1	Interrupts are controlled by I and UI bits and ICR	
1	0	2	Cannot be used in the H8S/2128 Series or H8S/2124 Series	
	1	3	Cannot be used in the H8S/2128 Series or H8S/2124 Series	

Bit 2—NMI Edge Select (NMIEG): Selects the input edge for the NMI pin.

Bit 2		
NMIEG	**Description**	
0	Interrupt request generated at falling edge of NMI input	(Initial value)
1	Interrupt request generated at rising edge of NMI input	

5.2.2 Interrupt Control Registers A to C (ICRA to ICRC)

Bit	7	6	5	4	3	2	1	0
	ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The ICR registers are three 8-bit readable/writable registers that set the interrupt control level for interrupts other than NMI.

The correspondence between ICR settings and interrupt sources is shown in table 5.3.

The ICR registers are initialized to H'00 by a reset and in hardware standby mode.

Bit n—Interrupt Control Level (ICRn): Sets the control level for the corresponding interrupt source.

Bit n

ICRn	**Description**	
0	Corresponding interrupt source is control level 0 (non-priority)	(Initial value)
1	Corresponding interrupt source is control level 1 (priority)	

(n = 7 to 0)

Table 5.3 Correspondence between Interrupt Sources and ICR Settings

	Bits							
Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ICRA	IRQ0	IRQ1	IRQ2	—	—	DTC	Watchdog timer 0	Watchdog timer 1
ICRB	A/D converter	Free-running timer	—	—	8-bit timer channel 0	8-bit timer channel 1	8-bit timer channels X, Y	—
ICRC	SCI channel 0	SCI channel 1	—	IIC channel 0 (option)	IIC channel 1 (option)	—	—	—

5.2.3 IRQ Enable Register (IER)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	IRQ2E	IRQ1E	IRQ0E
Initial value	1	1	1	1	1	0	0	0
Read/Write	R	R	R	R	R	R/W	R/W	R/W

IER is an 8-bit readable/writable register that controls enabling and disabling of interrupt requests IRQ2 to IRQ0.

IER is initialized to H'F8 by a reset and in hardware standby mode.

Bits 7 to 3—Reserved: These bits cannot be modified and are always read as 0.

Bits 2 to 0—IRQ2 to IRQ0 Enable (IRQ2E to IRQ0E): These bits select whether IRQ2 to IRQ0 are enabled or disabled.

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

(n = 2 to 0)

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

- ISCRH

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

- ISCRL

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ISCRH and ISCRL are 8-bit readable/writable registers that select rising edge, falling edge, or both edge detection, or level sensing, for the input at pins $\overline{IRQ2}$ to $\overline{IRQ0}$.

Each of the ISCR registers is initialized to H'00 by a reset and in hardware standby mode.

ISCRH Bits 7 to 0, ISCRL Bits 7 and 6—Reserved: Do not write 1 to this bit.

ISCRL Bits 5 to 0—IRQ2 Sense Control A and B (IRQ2SCA, IRQ2SCB) to IRQ0 Sense Control A and B (IRQ0SCA, IRQ0SCB)

ISCRL Bits 5 to 0		
IRQ2SCB to IRQ0SCB	**IRQ2SCA to IRQ0SCA**	**Description**
0	0	Interrupt request generated at $\overline{IRQ2}$ to $\overline{IRQ0}$ input low level (Initial value)
	1	Interrupt request generated at falling edge of $\overline{IRQ2}$ to $\overline{IRQ0}$ input
1	0	Interrupt request generated at rising edge of $\overline{IRQ2}$ to $\overline{IRQ0}$ input
	1	Interrupt request generated at both falling and rising edges of $\overline{IRQ2}$ to $\overline{IRQ0}$ input

5.2.5 IRQ Status Register (ISR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	IRQ2F	IRQ1F	IRQ0F
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R/(W)*	R/(W)*	R/(W)*

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

ISR is an 8-bit readable/writable register that indicates the status of IRQ2 to IRQ0 interrupt requests.

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

Bits 7 to 3—Reserved

Bits 2 to 0—IRQ2 to IRQ0 Flags (IRQ2F to IRQ0F): These bits indicate the status of IRQ2 to IRQ0 interrupt requests.

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

(n = 2 to 0)

5.2.6 Address Break Control Register (ABRKCR)

Bit	7	6	5	4	3	2	1	0
	CMF	—	—	—	—	—	—	BIE
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	—	—	—	—	—	—	R/W

ABRKCR is an 8-bit readable/writable register that performs address break control.

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

Bit 7—Condition Match Flag (CMF): This is the address break source flag, used to indicate that the address set by BAR has been prefetched. When the CMF flag and BIE flag are both set to 1, an address break is requested.

Bit 7

CMF	Description	
0	[Clearing condition] When address break interrupt exception handling is executed	(Initial value)
1	[Setting condition] When address set by BARA to BARC is prefetched while BIE = 1	

Bits 6 to 1—Reserved: These bits cannot be modified and are always read as 0.

Bit 0—Break Interrupt Enable (BIE): Selects address break enabling or disabling.

Bit 0

BIE	Description	
0	Address break disabled	(Initial value)
1	Address break enabled	

5.2.7 Break Address Registers A, B, C (BARA, BARB, BARC)

Bit	7	6	5	4	3	2	1	0
BARA	A23	A22	A21	A20	A19	A18	A17	A16
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
BARB	A15	A14	A13	A12	A11	A10	A9	A8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
BARC	A7	A6	A5	A4	A3	A2	A1	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

BAR consists of three 8-bit readable/writable registers (BARA, BARB, and BARC), and is used to specify the address at which an address break is to be executed.

Each of the BAR registers is initialized to H'00 by a reset and in hardware standby mode. They are not initialized in software standby mode.

BARA Bits 7 to 0—Address 23 to 16 (A23 to A16)
BARB Bits 7 to 0—Address 15 to 8 (A15 to A8)
BARC Bits 7 to 1—Address 7 to 1 (A7 to A1)

These bits specify the address at which an address break is to be executed. BAR bits A23 to A1 are compared with internal address bus lines A23 to A1, respectively.

The address at which the first instruction byte is located should be specified as the break address. Occurrence of the address break condition may not be recognized for other addresses.

In normal mode, no comparison is made with address lines A23 to A16.

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

5.3 Interrupt Sources

Interrupt sources comprise external interrupts (NMI and IRQ2 to IRQ0) and internal interrupts.

5.3.1 External Interrupts

There are four external interrupt sources: NMI, and $\overline{\text{IRQ2}}$ to $\overline{\text{IRQ0}}$. NMI, and IRQ2 to IRQ0 can be used to restore the H8S/2128 Series or H8S/2124 Series chip from software standby mode.

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

The vector number for NMI interrupt exception handling is 7.

IRQ2 to IRQ0 Interrupts: Interrupts IRQ2 to IRQ0 are requested by an input signal at pins $\overline{\text{IRQ2}}$ to $\overline{\text{IRQ0}}$. Interrupts IRQ2 to IRQ0 have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins $\overline{\text{IRQ2}}$ to $\overline{\text{IRQ0}}$.
- Enabling or disabling of interrupt requests IRQ2 to IRQ0 can be selected with IER.
- The interrupt control level can be set with ICR.
- The status of interrupt requests IRQ2 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

A block diagram of interrupts IRQ2 to IRQ0 is shown in figure 5.2.

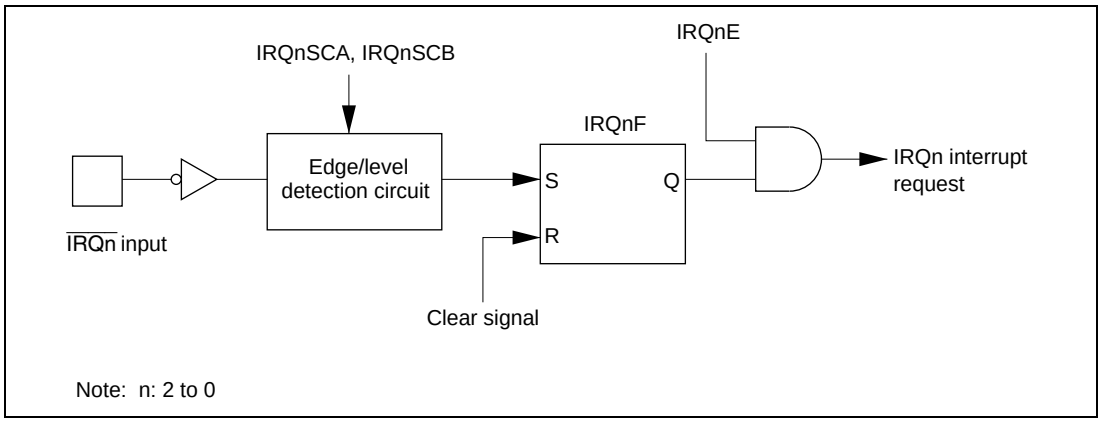


Figure 5.2 Block Diagram of Interrupts IRQ2 to IRQ0

Figure 5.3 shows the timing of IRQnF setting.

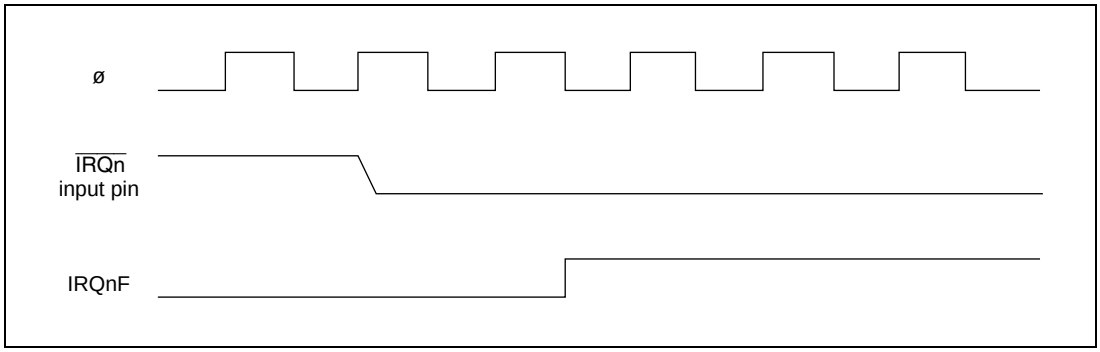


Figure 5.3 Timing of IRQnF Setting

The vector numbers for IRQ2 to IRQ0 interrupt exception handling are 18 to 16.

Detection of IRQ2 to IRQ0 interrupts does not depend on whether the relevant pin has been set for input or output. Therefore, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR bit to 0 and use the pin as an I/O pin for another function.

As interrupt request flags IRQ2F to IRQ0F are set when the setting condition is met, regardless of the IER setting, only the necessary flags should be referenced.

5.3.2 Internal Interrupts

There are 32 sources for internal interrupts from on-chip supporting modules, plus one software interrupt source (PC break).

- For each on-chip supporting module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. If any one of these is set to 1, an interrupt request is issued to the interrupt controller.
- The interrupt control level can be set by means of ICR.
- The DTC can be activated by an FRT, TMR, SCI, or other interrupt request. When the DTC is activated by an interrupt, the interrupt control mode and interrupt mask bits have no effect.

5.3.3 Interrupt Exception Vector Table

Table 5.4 shows interrupt exception handling sources, vector addresses, and interrupt priorities. For default priorities, the lower the vector number, the higher the priority.

Priorities among modules can be set by means of ICR. The situation when two or more modules are set to the same priority, and priorities within a module, are fixed as shown in table 5.4.

Table 5.4 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address		ICR	Priority
			Normal Mode	Advanced Mode		
NMI	External pin	7	H'000E	H'00001C		High
IRQ0		16	H'0020	H'000040	ICRA7	
IRQ1		17	H'0022	H'000044	ICRA6	
IRQ2		18	H'0024	H'000048	ICRA5	
Reserved	—	19 to 23	H'0026 to H'002E	H'00004C to H'00005C		
SWDTEND (software activation interrupt end)	DTC	24	H'0030	H'000060	ICRA2	
WOVI0 (interval timer)	Watchdog timer 0	25	H'0032	H'000064	ICRA1	
WOVI1 (interval timer)	Watchdog timer 1	26	H'0034	H'000068	ICRA0	
PC break	—	27	H'0036	H'00006C		
ADI (A/D conversion end)	A/D	28	H'0038	H'000070	ICRB7	
Reserved	—	29 to 47	H'003A to H'005E	H'000074 to H'0000BC		
ICIA (input capture A)	Free-running timer	48	H'0060	H'0000C0	ICRB6	
ICIB (input capture B)		49	H'0062	H'0000C4		
ICIC (input capture C)		50	H'0064	H'0000C8		
ICID (input capture D)		51	H'0066	H'0000CC		
OCIA (output compare A)		52	H'0068	H'0000D0		
OCIB (output compare B)		53	H'006A	H'0000D4		
FOVI (overflow)		54	H'006C	H'0000D8		
Reserved		55	H'006E	H'0000DC		
Reserved	—	56 to 63	H'0070 to H'007E	H'0000E0 to H'0000FC		Low

Table 5.4 Interrupt Sources, Vector Addresses, and Interrupt Priorities (cont)

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address: Normal Mode	Vector Address: Advanced Mode	ICR	Priority
CMIA0 (compare-match A) CMIB0 (compare-match B) OVI0 (overflow) Reserved	8-bit timer channel 0	64 65 66 67	H'0080 H'0082 H'0084 H'0086	H'000100 H'000104 H'000108 H'00010C	ICRB3	High
CMIA1 (compare-match A) CMIB1 (compare-match B) OVI1 (overflow) Reserved	8-bit timer channel 1	68 69 70 71	H'0088 H'008A H'008C H'008E	H'000110 H'000114 H'000118 H'00011C	ICRB2	
CMIAY (compare-match A) CMIBY (compare-match B) OVIY (overflow) ICIX (input capture X)	8-bit timer channels Y, X	72 73 74 75	H'0090 H'0092 H'0094 H'0096	H'000120 H'000124 H'000128 H'00012C	ICRB1	
Reserved	—	76 to 79	H'0098 to H'009E	H'000130 to H'00013C		
ERI0 (receive error 0) RXI0 (reception completed 0) TXI0 (transmit data empty 0) TEI0 (transmission end 0)	SCI channel 0	80 81 82 83	H'00A0 H'00A2 H'00A4 H'00A6	H'000140 H'000144 H'000148 H'00014C	ICRC7	
ERI1 (receive error 1) RXI1 (reception completed 1) TXI1 (transmit data empty 1) TEI1 (transmission end 1)	SCI channel 1	84 85 86 87	H'00A8 H'00AA H'00AC H'00AE	H'000150 H'000154 H'000158 H'00015C	ICRC6	
Reserved	—	84 to 91	H'00B0 to H'00B6	H'000160 to H'00016C		
IICI0 (1-byte transmission/ reception completed) DDCSWI (format switch)	IIC channel 0 (option)	92 93	H'00B8 H'00BA	H'000170 H'000174	ICRC4	
IICI1 (1-byte transmission/ reception completed) Reserved	IIC channel 1 (option)	94 95	H'00BC H'00BE	H'000178 H'00017C	ICRC3	
Reserved	—	96 to 103	H'00C0 to H'00CE	H'000180 to H'00019C		Low

5.4 Address Breaks

5.4.1 Features

With the H8S/2128 Series and H8S/2124 Series, it is possible to identify the prefetch of a specific address by the CPU and generate an address break interrupt, using the ABRKCR and BAR registers. When an address break interrupt is generated, address break interrupt exception handling is executed.

This function can be used to detect the beginning of execution of a bug location in the program, and branch to a correction routine.

5.4.2 Block Diagram

A block diagram of the address break function is shown in figure 5.4.

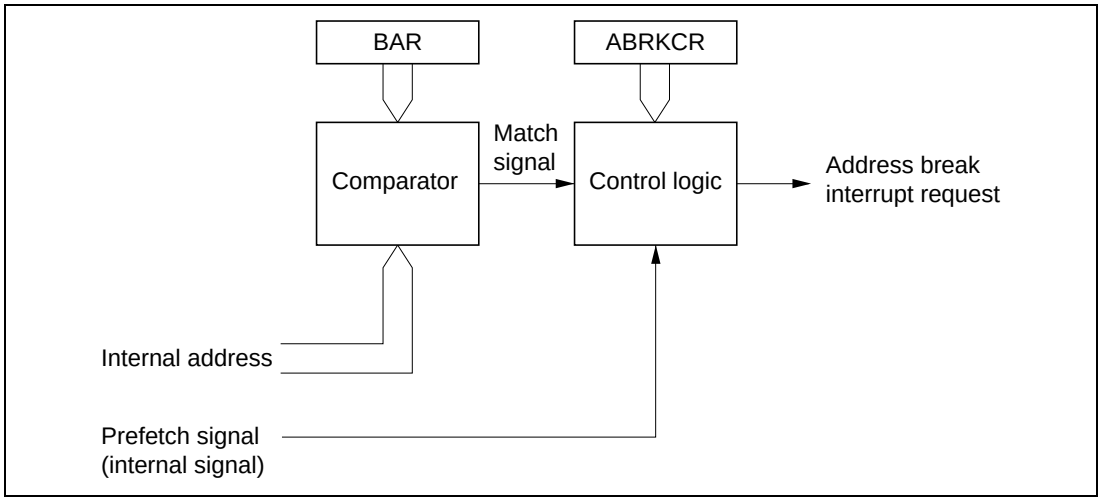


Figure 5.4 Block Diagram of Address Break Function

5.4.3 Operation

ABRKCR and BAR settings can be made so that an address break interrupt is generated when the CPU prefetches the address set in BAR. This address break function issues an interrupt request to the interrupt controller when the address is prefetched, and the interrupt controller determines the interrupt priority. When the interrupt is accepted, interrupt exception handling is started on completion of the currently executing instruction. With an address break interrupt, interrupt mask control by the I and UI bits in the CPU's CCR is ineffective.

The register settings when the address break function is used are as follows.

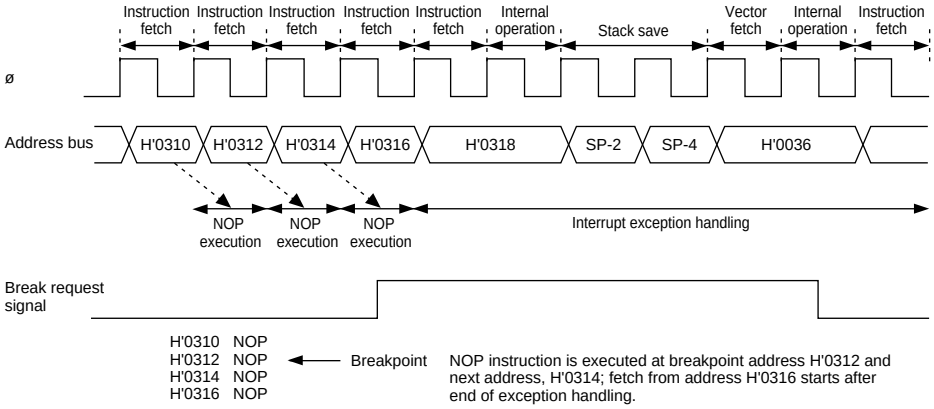
1. Set the break address in bits A23 to A1 in BAR.
2. Set the BIE bit in ABRKCR to 1 to enable address breaks. An address break will not be requested if the BIE bit is cleared to 0.

When the setting condition occurs, the CMF flag in ABRKCR is set to 1 and an interrupt is requested. If necessary, the source should be identified in the interrupt handling routine.

5.4.4 Usage Notes

- With the address break function, the address at which the first instruction byte is located should be specified as the break address. Occurrence of the address break condition may not be recognized for other addresses.
- In normal mode, no comparison is made with address lines A23 to A16.
- If a branch instruction (Bcc, BSR), jump instruction (JMP, JSR), RTS instruction, or RTE instruction is located immediately before the address set in BAR, execution of this instruction will output a prefetch signal for that address, and an address break may be requested. This can be prevented by not making a break address setting for an address immediately following one of these instructions, or by determining within the interrupt handling routine whether interrupt handling was initiated by a genuine condition occurrence.
- As an address break interrupt is generated by a combination of the internal prefetch signal and address, the timing of the start of interrupt exception handling depends on the content and execution cycle of the instruction at the set address and the preceding instruction. Figure 5.5 shows some address timing examples.

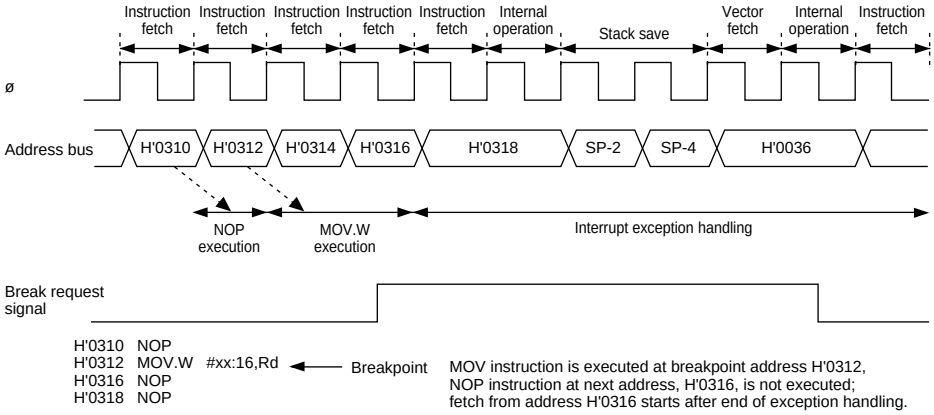
• Program area in on-chip memory, 1-state execution instruction at specified break address



H'0310 NOP
H'0312 NOP ← Breakpoint
H'0314 NOP
H'0316 NOP

NOP instruction is executed at breakpoint address H'0312 and next address, H'0314; fetch from address H'0316 starts after end of exception handling.

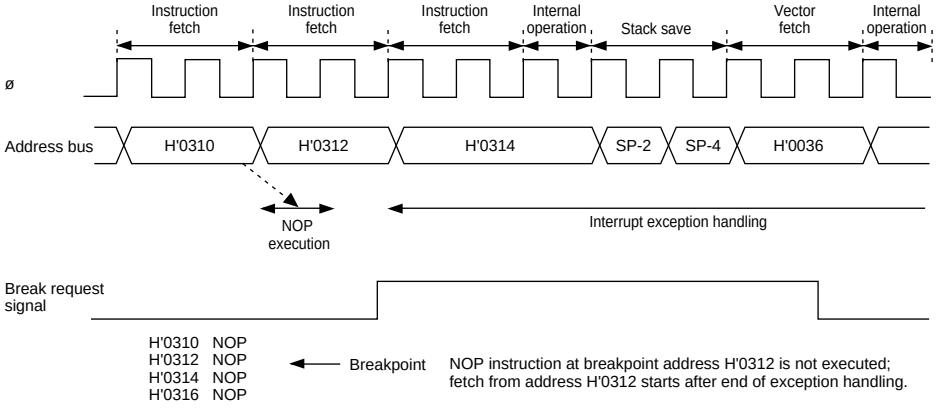
• Program area in on-chip memory, 2-state execution instruction at specified break address



H'0310 NOP
H'0312 MOV.W #xx:16,Rd ← Breakpoint
H'0316 NOP
H'0318 NOP

MOV instruction is executed at breakpoint address H'0312, NOP instruction at next address, H'0316, is not executed; fetch from address H'0316 starts after end of exception handling.

• Program area in external memory (2-state access, 16-bit-bus access), 1-state execution instruction at specified break address



H'0310 NOP
H'0312 NOP ← Breakpoint
H'0314 NOP
H'0316 NOP

NOP instruction at breakpoint address H'0312 is not executed; fetch from address H'0312 starts after end of exception handling.

Figure 5.5 Examples of Address Break Timing

5.5 Interrupt Operation

5.5.1 Interrupt Control Modes and Interrupt Operation

Interrupt operations in the H8S/2128 Series and H8S/2124 Series differ depending on the interrupt control mode.

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

Table 5.5 shows the interrupt control modes.

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

Table 5.5 Interrupt Control Modes

Interrupt Control Mode	SYSCR		Priority Setting Register	Interrupt Mask Bits	Description
	INTM1	INTM0			
0	0	0	ICR	I	Interrupt mask control is performed by the I bit Priority can be set with ICR
1		1	ICR	I, UI	3-level interrupt mask control is performed by the I and UI bits Priority can be set with ICR

Figure 5.6 shows a block diagram of the priority decision circuit.

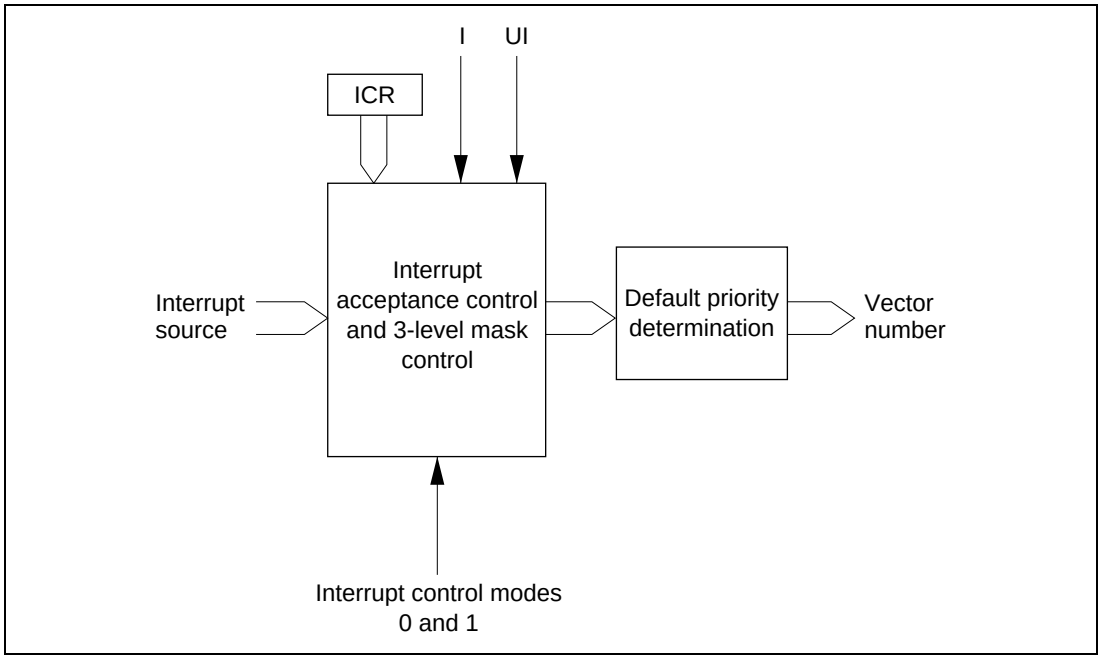


Figure 5.6 Block Diagram of Interrupt Control Operation

Interrupt Acceptance Control and 3-Level Control: In interrupt control modes 0 and 1, interrupt acceptance control and 3-level mask control is performed by means of the I and UI bits in CCR, and ICR (control level).

Table 5.6 shows the interrupts selected in each interrupt control mode.

Table 5.6 Interrupts Selected in Each Interrupt Control Mode

	Interrupt Mask Bits		
Interrupt Control Mode	**I**	**UI**	**Selected Interrupts**
0	0	*	All interrupts (control level 1 has priority)
	1	*	NMI interrupts
1	0	*	All interrupts (control level 1 has priority)
	1	0	NMI and control level 1 interrupts
		1	NMI interrupts

Legend:
*: Don’t care

Default Priority Determination: The priority is determined for the selected interrupt, and a vector number is generated.

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

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

Table 5.7 shows operations and control signal functions in each interrupt control mode.

Table 5.7 Operations and Control Signal Functions in Each Interrupt Control Mode

Interrupt Control Mode	Setting		Interrupt Acceptance Control 3-Level Control				Default Priority Determination	T (Trace)
	INTM1	INTM0		I	UI	ICR		
0	0	0	O	IM	—	PR	O	—
1		1	O	IM	IM	PR	O	—

Legend:
O: Interrupt operation control performed
IM: Used as interrupt mask bit
PR: Sets priority
—: Not used

5.5.2 Interrupt Control Mode 0

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

Figure 5.7 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. When interrupt requests are sent to the interrupt controller, a control level 1 interrupt, according to the control level set in ICR, has priority for selection, and other interrupt requests are held pending. If a number of interrupt requests with the same control level setting are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.4 is selected.
3. The I bit is then referenced. If the I bit is cleared to 0, the interrupt request is accepted. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending.
4. When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This disables all interrupts except NMI.
7. A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.

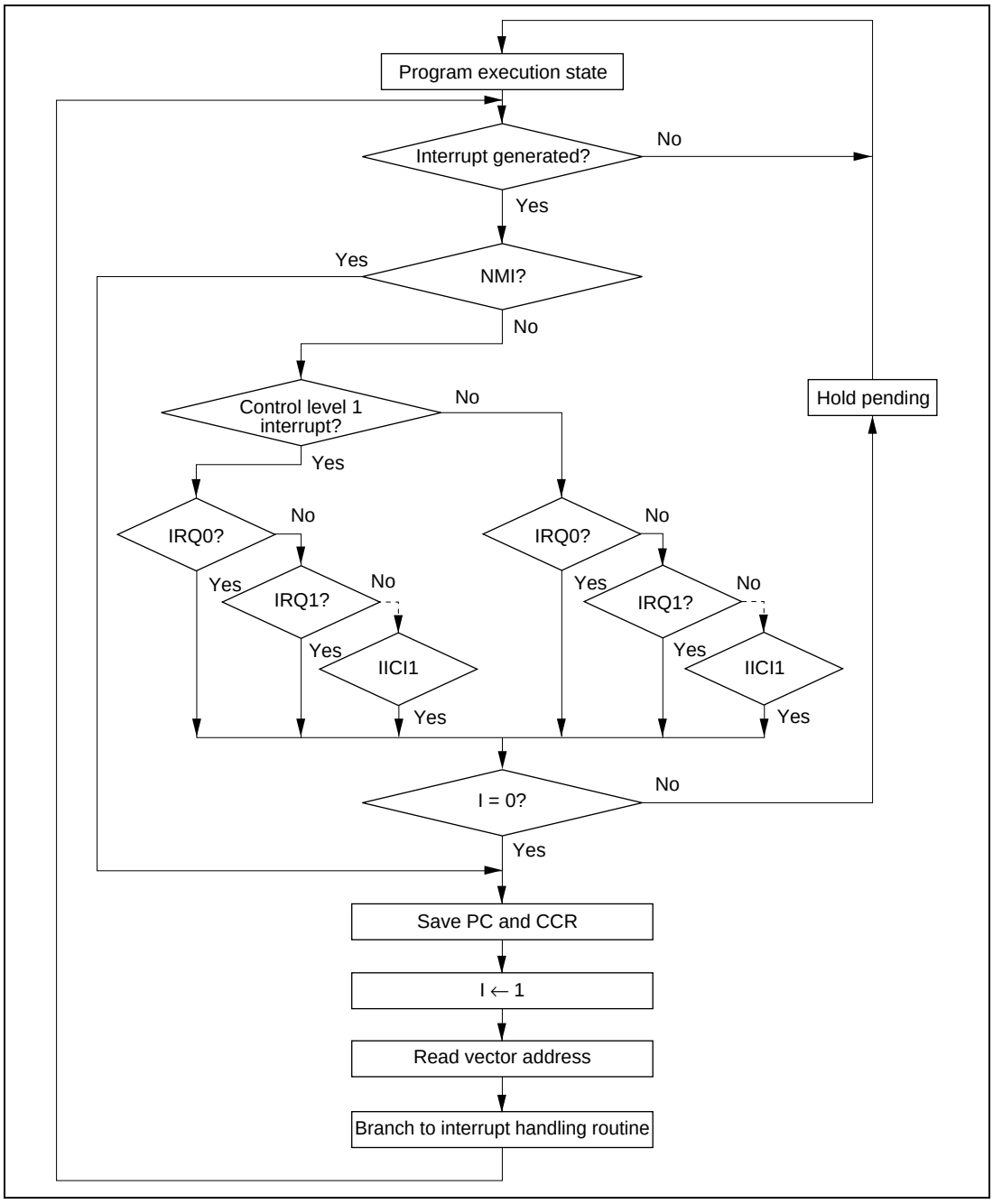


Figure 5.7 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

5.5.3 Interrupt Control Mode 1

Three-level masking is implemented for IRQ interrupts and on-chip supporting module interrupts by means of the I and UI bits in the CPU's CCR, and ICR.

- Control level 0 interrupt requests are enabled when the I bit is cleared to 0, and disabled when set to 1.
- Control level 1 interrupt requests are enabled when the I bit or UI bit is cleared to 0, and disabled when both the I bit and the UI bit are set to 1.

For example, if the interrupt enable bit for an interrupt request is set to 1, and H'20, H'00, and H'00 are set in ICRA, ICRB, and ICRC, respectively, (i.e. IRQ2 interrupts are set to control level 1 and other interrupts to control level 0), the situation is as follows:

- When I = 0, all interrupts are enabled
  (Priority order: NMI > IRQ2 > IRQ0 > IRQ1 ...)
- When I = 1 and UI = 0, only NMI, and IRQ2 interrupts are enabled
- When I = 1 and UI = 1, only NMI interrupts are enabled

Figure 5.8 shows the state transitions in these cases.

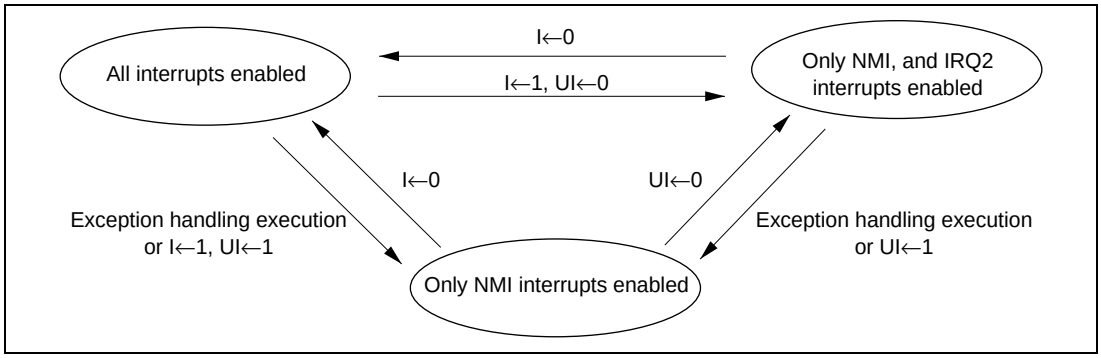


Figure 5.8 Example of State Transitions in Interrupt Control Mode 1

Figure 5.9 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. When interrupt requests are sent to the interrupt controller, a control level 1 interrupt, according to the control level set in ICR, has priority for selection, and other interrupt requests are held pending. If a number of interrupt requests with the same control level setting are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.4 is selected.
3. The I bit is then referenced. If the I bit is cleared to 0, the UI bit has no effect.

   An interrupt request set to interrupt control level 0 is accepted when the I bit is cleared to 0. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending.

   An interrupt request set to interrupt control level 1 has priority over an interrupt request set to interrupt control level 0, and is accepted if the I bit is cleared to 0, or if the I bit is set to 1 and the UI bit is cleared to 0.

   When both the I bit and the UI bit are set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending.
4. When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I and UI bits in CCR are set to 1. This disables all interrupts except NMI.
7. A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.

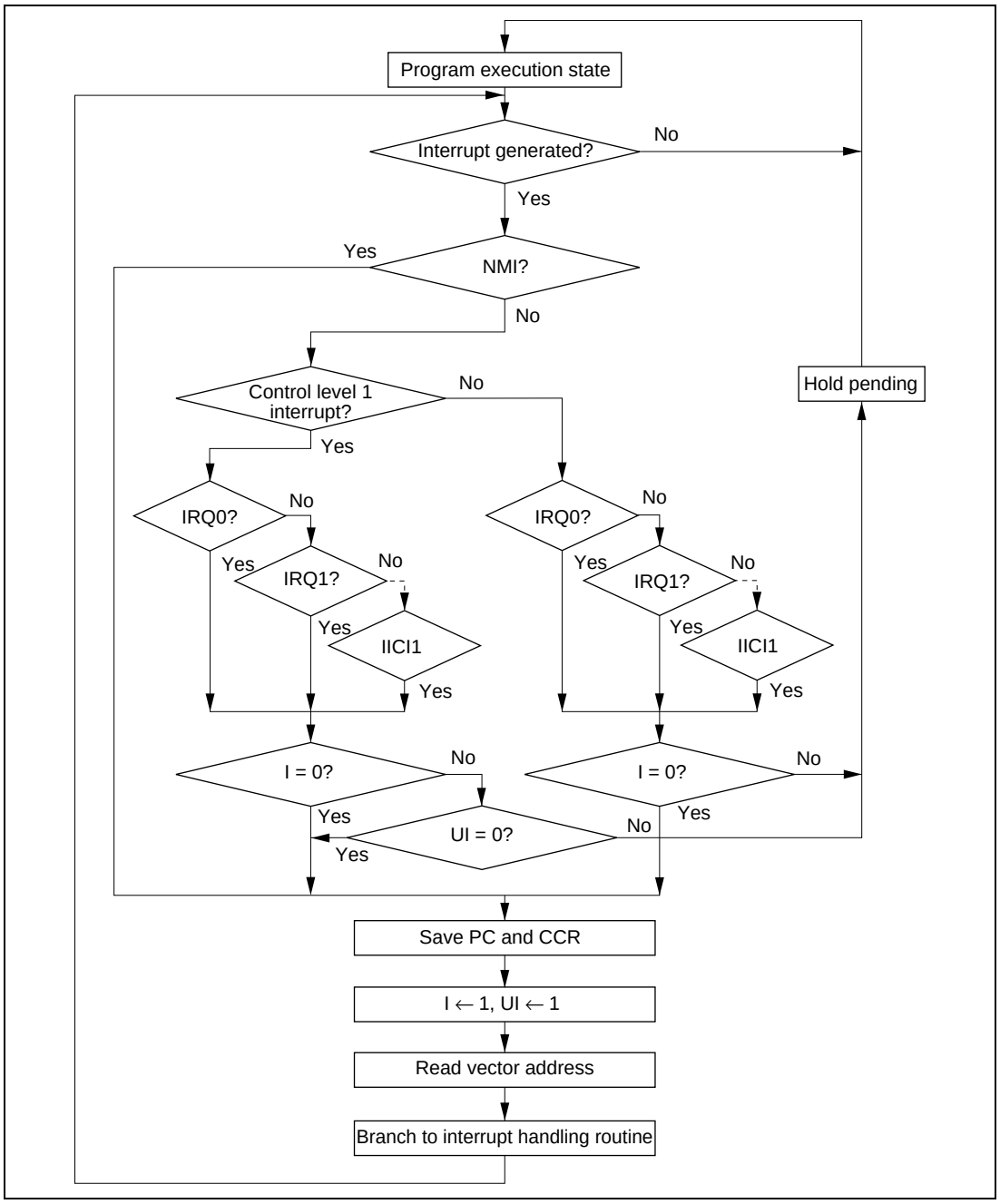


Figure 5.9 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 1

5.5.4 Interrupt Exception Handling Sequence

Figure 5.10 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0 is set in advanced mode, and the program area and stack area are in on-chip memory.

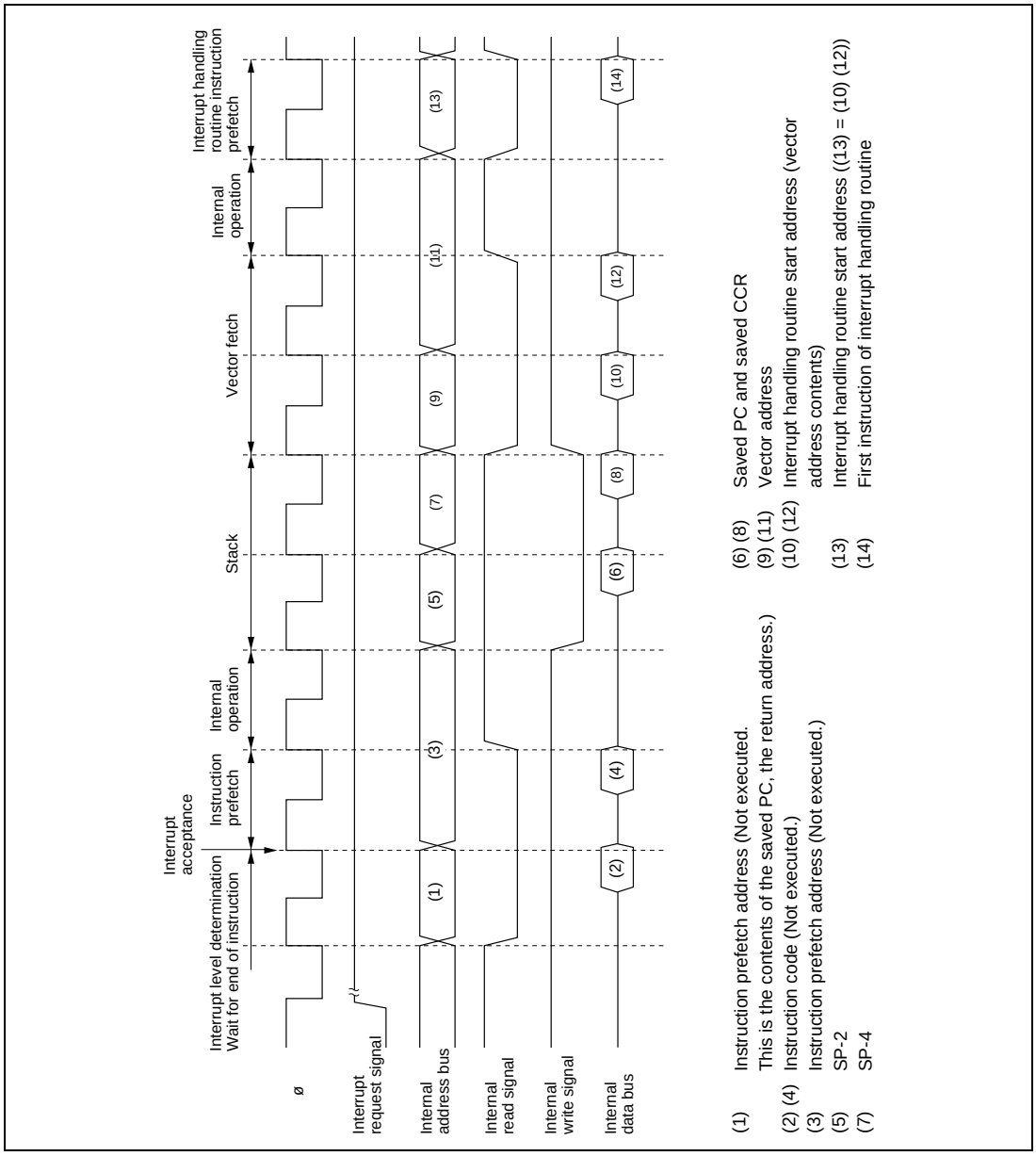


Figure 5.10 Interrupt Exception Handling

5.5.5 Interrupt Response Times

The H8S/2128 Series and H8S/2124 Series are capable of fast word access to on-chip memory, and high-speed processing can be achieved by providing the program area in on-chip ROM and the stack area in on-chip RAM.

Table 5.8 shows interrupt response times—the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine. The symbols used in table 5.8 are explained in table 5.9.

Table 5.8 Interrupt Response Times

		Number of States	
No.	**Item**	**Normal Mode**	**Advanced Mode**
1	Interrupt priority determination*[1]	3	3
2	Number of wait states until executing instruction ends*[2]	1 to 19+2·S_I	1 to 19+2·S_I
3	PC, CCR stack save	2·S_K	2·S_K
4	Vector fetch	S_I	2·S_I
5	Instruction fetch*[3]	2·S_I	2·S_I
6	Internal processing*[4]	2	2
Total (using on-chip memory)		11 to 31	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 vector fetch.

Table 5.9 Number of States in Interrupt Handling Routine Execution

		Object of Access		
			External Device	
			8-Bit Bus	
	Symbol	**Internal Memory**	**2-State Access**	**3-State Access**
Instruction fetch	S_I	1	4	6+2m
Branch address read	S_J			
Stack manipulation	S_K			

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

5.6 Usage Notes

5.6.1 Contention between Interrupt Generation and Disabling

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

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

The same also applies when an interrupt source flag is cleared to 0.

Figure 5.11 shows an example in which the CMIEA bit in 8-bit timer register TCR is cleared to 0.

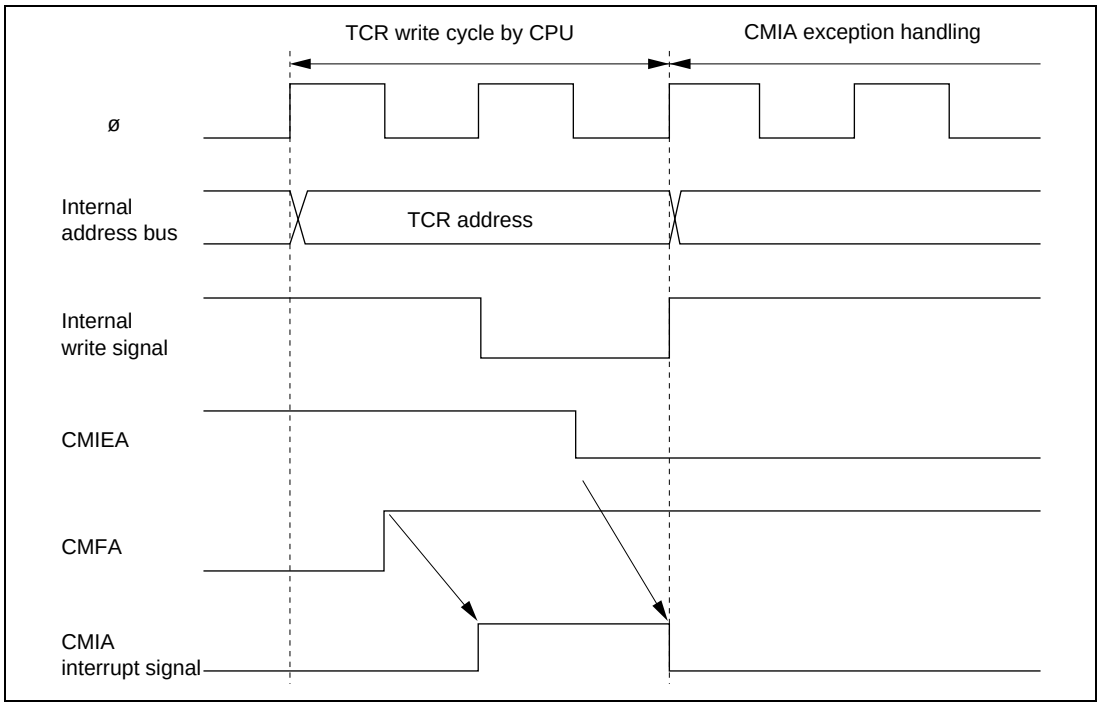


Figure 5.11 Contention between Interrupt Generation and Disabling

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

5.6.2 Instructions that Disable Interrupts

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

5.6.3 Interrupts during Execution of EEPMOV Instruction

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

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

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

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

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

5.7 DTC Activation by Interrupt

5.7.1 Overview

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

- Interrupt request to CPU
- Activation request to DTC
- Both of the above

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

5.7.2 Block Diagram

Figure 5.12 shows a block diagram of the DTC and interrupt controller.

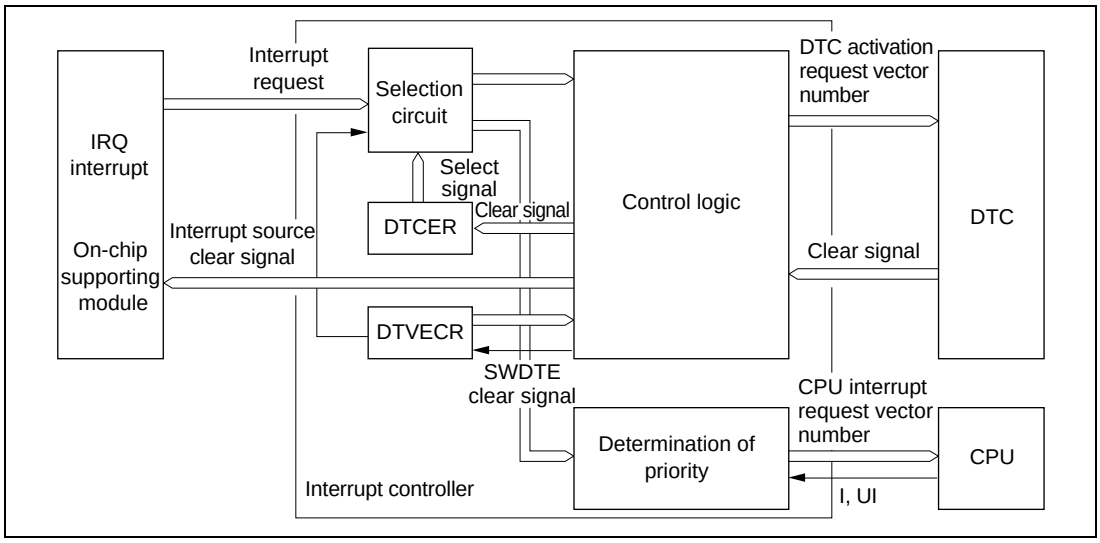


Figure 5.12 Interrupt Control for DTC

5.7.3 Operation

The interrupt controller has three main functions in DTC control.

Selection of Interrupt Source: It is possible to select DTC activation request or CPU interrupt request with the DTCE bit of DTCERA to DTCERE in the DTC.

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

When the DTC performs the specified number of data transfers and the transfer counter reaches 0, following the DTC data transfer the DTCE bit is cleared to 0 and an interrupt request is sent to the CPU.

Determination of Priority: The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels. See section 7.3.3, DTC Vector Table, for the respective priorities.

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

Table 5.10 summarizes interrupt source selection and interrupt source clearance control according to the settings of the DTCE bit of DTCERA to DTCERE in the DTC and the DISEL bit of MRB in the DTC.

Table 5.10 Interrupt Source Selection and Clearing Control

Settings			
DTC		Interrupt Source Selection/Clearing Control	
DTCE	DISEL	DTC	CPU
0	*	×	(
1	0	(	×
	1	○	(

Legend

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

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

×: The relevant bit cannot be used.

*: Don't care

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

Section 6 Bus Controller

6.1 Overview

The H8S/2128 Series and H8S/2124 Series have a built-in bus controller (BSC) that allows external address space bus specifications, such as bus width and number of access states, to be set.

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

6.1.1 Features

The features of the bus controller are listed below.

- Basic bus interface
  - 2-state access or 3-state access can be selected
  - Program wait states can be inserted
- Burst ROM interface
  - External space can be designated as ROM interface space
  - 1-state or 2-state burst access can be selected
- Idle cycle insertion
  - An idle cycle can be inserted when an external write cycle immediately follows an external read cycle
- Bus arbitration function
  - Includes a bus arbiter that arbitrates bus mastership between the CPU and DTC

6.1.2 Block Diagram

Figure 6.1 shows a block diagram of the bus controller.

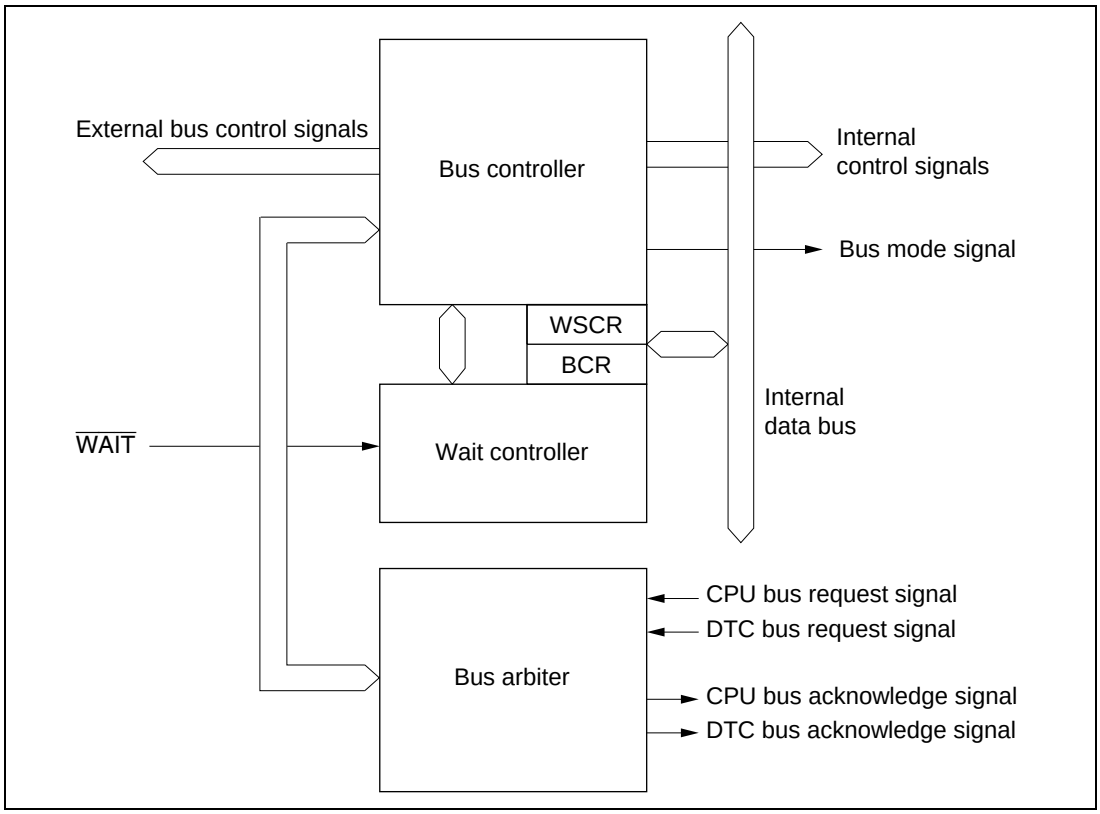


Figure 6.1 Block Diagram of Bus Controller

6.1.3 Pin Configuration

Table 6.1 summarizes the pins of the bus controller.

Table 6.1 Bus Controller Pins

Name	Symbol	I/O	Function
Address strobe	$\overline{AS}$	Output	Strobe signal indicating that address output on address bus is enabled (when IOSE bit is 0)
I/O select	$\overline{IOS}$	Output	I/O select signal (when IOSE bit is 1)
Read	$\overline{RD}$	Output	Strobe signal indicating that external space is being read
Write	$\overline{WR}$	Output	Strobe signal indicating that external space is being written to, and that data bus is enabled
Wait	$\overline{WAIT}$	Input	Wait request signal when external 3-state access space is accessed

6.1.4 Register Configuration

Table 6.2 summarizes the registers of the bus controller.

Table 6.2 Bus Controller Registers

Name	Abbreviation	R/W	Initial Value	Address*
Bus control register	BCR	R/W	H'D7	H'FFC6
Wait state control register	WSCR	R/W	H'33	H'FFC7

Note: *Lower 16 bits of the address.

6.2 Register Descriptions

6.2.1 Bus Control Register (BCR)

Bit	7	6	5	4	3	2	1	0
	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0
Initial value	1	1	0	1	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCR is an 8-bit readable/writable register that specifies the external memory space access mode, and the extent of the I/O area when the I/O strobe function has been selected for the $\overline{AS}$ pin.

BCR is initialized to H'D7 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Idle Cycle Insert 1 (ICIS1): Reserved. Do not write 0 to this bit.

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

Bit 6

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

Bit 5—Burst ROM Enable (BRSTRM): Selects whether external space is designated as a burst ROM interface space. The selection applies to the entire external space .

Bit 5

BRSTRM	Description
0	Basic bus interface (Initial value)
1	Burst ROM interface

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

Bit 4

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

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

Bit 3

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

Bit 2—Reserved: Do not write 0 to this bit.

Bits 1 and 0—IOS Select 1 and 0 (IOS1, IOS0): See table 6.4.

6.2.2 Wait State Control Register (WSCR)

Bit	7	6	5	4	3	2	1	0
	RAMS	RAM0	ABW	AST	WMS1	WMS0	WC1	WC0
Initial value	0	0	1	1	0	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WSCR is an 8-bit readable/writable register that specifies the data bus width, number of access states, wait mode, and number of wait states for external memory space. The on-chip memory and internal I/O register bus width and number of access states are fixed, irrespective of the WSCR settings.

WSCR is initialized to H'33 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—RAM Select (RAMS)/Bit 6—RAM Area Setting (RAM0): Reserved bits.

Bit 5—Bus Width Control (ABW): Specifies whether the external memory space is 8-bit access space or 16-bit access space.

However, a 16-bit access space cannot be specified for these series, and therefore 0 should not be written to this bit.

Bit 5		
ABW	**Description**	
0	External memory space is designated as 16-bit access space (A 16-bit access space cannot be specified for these series)	
1	External memory space is designated as 8-bit access space	(Initial value)

Bit 4—Access State Control (AST): Specifies whether the external memory space is 2-state access space or 3-state access space, and simultaneously enables or disables wait state insertion.

Bit 4		
AST	**Description**	
0	External memory space is designated as 2-state access space Wait state insertion in external memory space accesses is disabled	
1	External memory space is designated as 3-state access space Wait state insertion in external memory space accesses is enabled	(Initial value)

Bits 3 and 2—Wait Mode Select 1 and 0 (WMS1, WMS0): These bits select the wait mode when external memory space is accessed while the AST bit is set to 1.

Bit 3	Bit 2		
WMS1	**WMS0**	**Description**	
0	0	Program wait mode	(Initial value)
	1	Wait-disabled mode	
1	0	Pin wait mode	
	1	Pin auto-wait mode	

Bits 1 and 0—Wait Count 1 and 0 (WC1, WC0): These bits select the number of program wait states when external memory space is accessed while the AST bit is set to 1.

Bit 1	Bit 0		
WC1	**WC0**	**Description**	
0	0	No program wait states are inserted	
	1	1 program wait state is inserted in external memory space accesses	
1	0	2 program wait states are inserted in external memory space accesses	
	1	3 program wait states are inserted in external memory space accesses	(Initial value)

6.3 Overview of Bus Control

6.3.1 Bus Specifications

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

The bus width and number of access states for on-chip memory and internal I/O registers are fixed, and are not affected by the bus controller.

Bus Width: A bus width of 8 or 16 bits can be selected with the ABW bit. A 16-bit access space cannot be specified for these series.

Number of Access States: Two or three access states can be selected with the AST bit.

When 2-state access space is designated, wait insertion is disabled. The number of access states on the burst ROM interface is determined without regard to the AST bit setting.

Wait Mode and Number of Program Wait States: When 3-state access space is designated by the AST bit, the wait mode and the number of program wait states to be inserted automatically is selected with WMS1, WMS0, WC1, and WC0. From 0 to 3 program wait states can be selected.

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

Table 6.3 Bus Specifications for Each Area (Basic Bus Interface)

						Bus Specifications (Basic Bus Interface)		
ABW	**AST**	**WMS1**	**WMS0**	**WC1**	**WC0**	**Bus Width**	**Access States**	**Program Wait States**
0	0	—	—	—	—	Cannot be used in the H8S/2128 Series or H8S/2124 Series.		
1	0	—	—	—	—	8	2	0
	1	0	1	—	—	8	3	0
		—*	—*	0	0		3	0
					1			1
				1	0			2
					1			3

Note: * Except when WMS1 = 0 and WMS0 = 1

6.3.2 Advanced Mode

The H8S/2128 and H8S/2124 have 16 address output pins, so there are no pins for output of the upper address bits (A16 to A23) in advanced mode. H'FFF000 to H'FFFE4F can be accessed by designating the $\overline{AS}$ pin as an I/O strobe pin. The accessible external space is therefore H'FFF000 to H'FFFE4F even when expanded mode with ROM enabled is selected in advanced mode.

The initial state of the external space is basic bus interface, three-state access space. In ROM-enabled expanded mode, the space excluding the on-chip ROM, on-chip RAM, and internal I/O registers is external space. The on-chip RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space becomes external space.

6.3.3 Normal Mode

The initial state of the external memory space is basic bus interface, three-state access space. In ROM-disabled expanded mode, the space excluding the on-chip RAM and internal I/O registers is external space. In ROM-enabled expanded mode, the space excluding the on-chip ROM, on-chip RAM, and internal I/O registers is external space. The on-chip RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space becomes external space.

6.3.4 I/O Select Signal

In the H8S/2128 Series and H8S/2124 Series, an I/O select signal ($\overline{IOS}$) can be output, with the signal output going low when the designated external space is accessed.

Figure 6.2 shows an example of $\overline{IOS}$ signal output timing.

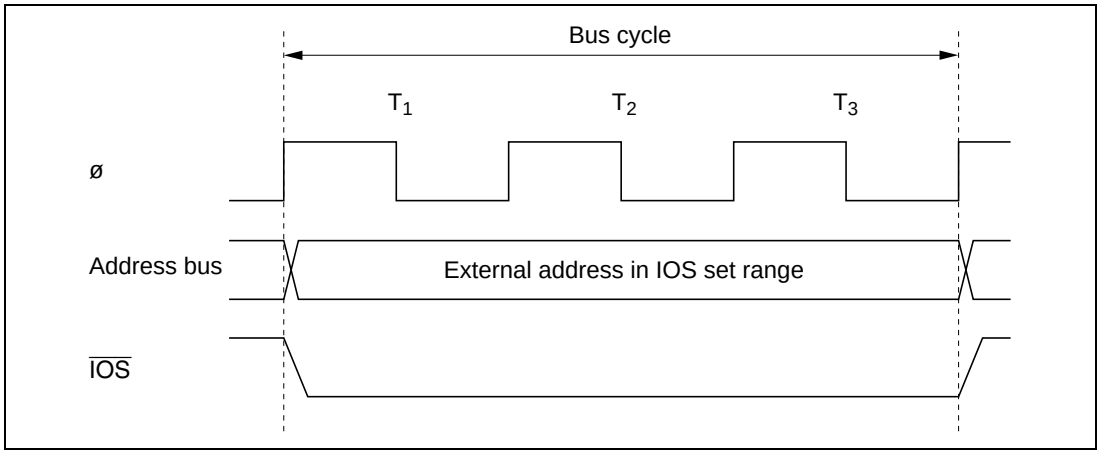


Figure 6.2 $\overline{IOS}$ Signal Output Timing

Enabling or disabling of $\overline{IOS}$ signal output is controlled by the setting of the IOSE bit in SYSCR. In expanded mode, this pin operates as the $\overline{AS}$ output pin after a reset, and therefore the IOSE bit in SYSCR must be set to 1 in order to use this pin as the $\overline{IOS}$ signal output. See section 8, I/O Ports, for details.

The range of addresses for which the $\overline{IOS}$ signal is output can be set with bits IOS1 and IOS0 in BCR. The $\overline{IOS}$ signal address ranges are shown in table 6.4.

Table 6.4 $\overline{IOS}$ Signal Output Range Settings

IOS1	IOS0	$\overline{IOS}$ Signal Output Range	
0	0	H'(FF)F000 to H'(FF)F03F	
	1	H'(FF)F000 to H'(FF)F0FF	
1	0	H'(FF)F000 to H'(FF)F3FF	
	1	H'(FF)F000 to H'(FF)FE4F	(Initial value)

6.4 Basic Bus Interface

6.4.1 Overview

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

The bus specifications can be selected with the AST bit, and the WMS1, WMS0, WC1, and WC0 bits (see table 6.3).

6.4.2 Data Size and Data Alignment

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

These series only have an upper data bus, and only 8-bit access space alignment is used. In these series, the upper data bus pins are designated D7 to D0.

8-Bit Access Space: Figure 6.3 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word access is performed as two byte accesses, and a longword access, as four byte accesses.

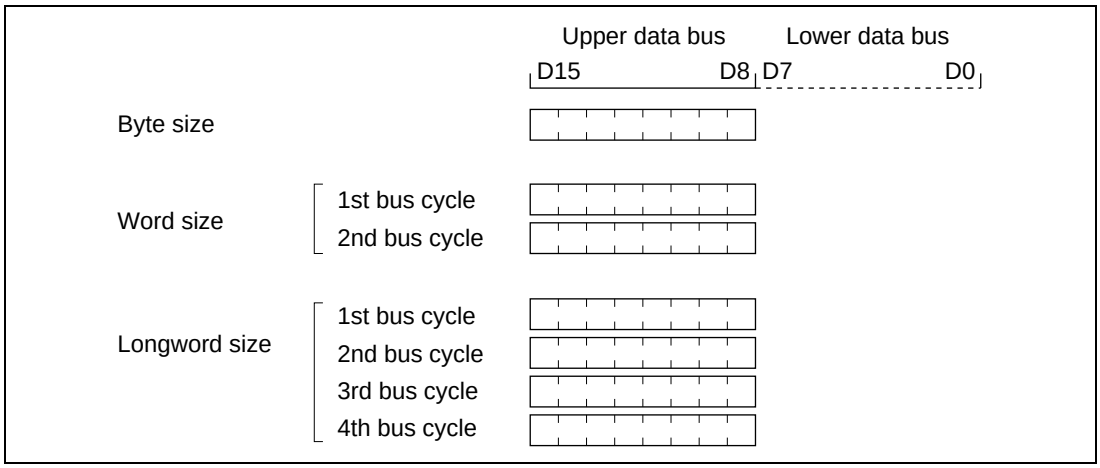


Figure 6.3 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-Bit Access Space (Cannot be Used in the H8S/2128 Series or H8S/2124 Series): Figure 6.4 illustrates data alignment control for the 16-bit access space. With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to D0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword access is executed as two word accesses.

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

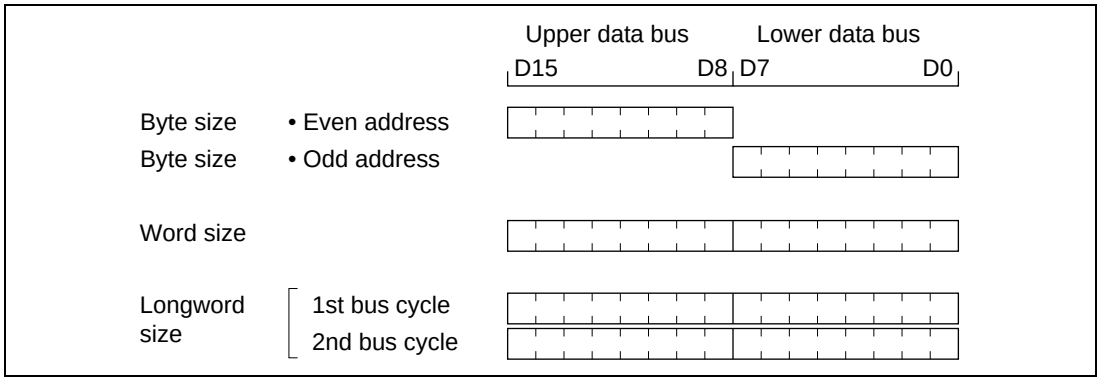


Figure 6.4 Access Sizes and Data Alignment Control (16-Bit Access Space)

6.4.3 Valid Strobes

Table 6.5 shows the data buses used and valid strobes for the access spaces.

In a read, the $\overline{RD}$ signal is valid without discrimination between the upper and lower halves of the data bus.

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

These series only have an upper data bus, and only the $\overline{RD}$ and $\overline{HWR}$ signals are valid. In these series, the $\overline{HWR}$ signal pin is designated $\overline{WR}$.

Table 6.5 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)*[1]	Lower Data Bus (D7 to D0)*[3]
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Port, etc.
		Write	—	$\overline{HWR}$*[2]		Port, etc.
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
(Cannot be used in the H8S/2128 Series or H8S/2124 Series)			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Undefined
			Odd	$\overline{LWR}$	Undefined	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

Notes: Undefined: Undefined data is output.

Invalid: Input state; input value is ignored.

Port, etc.: Pins are used as port or on-chip supporting module input/output pins, and not as data bus pins.

1. The pin names in these series are D7 to D0.
2. The pin name in these series is $\overline{WR}$.
3. There are no lower data bus pins in these series.

6.4.4 Basic Timing

8-Bit 2-State Access Space: Figure 6.5 shows the bus timing for an 8-bit 2-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used.

Wait states cannot be inserted.

These series have no lower data bus (D7 to D0) pins or $\overline{\text{LWR}}$ pin. In these series, the upper data bus (D15 to D8) pins are designated D7 to D0, and the $\overline{\text{HWR}}$ signal pin is designated $\overline{\text{WR}}$.

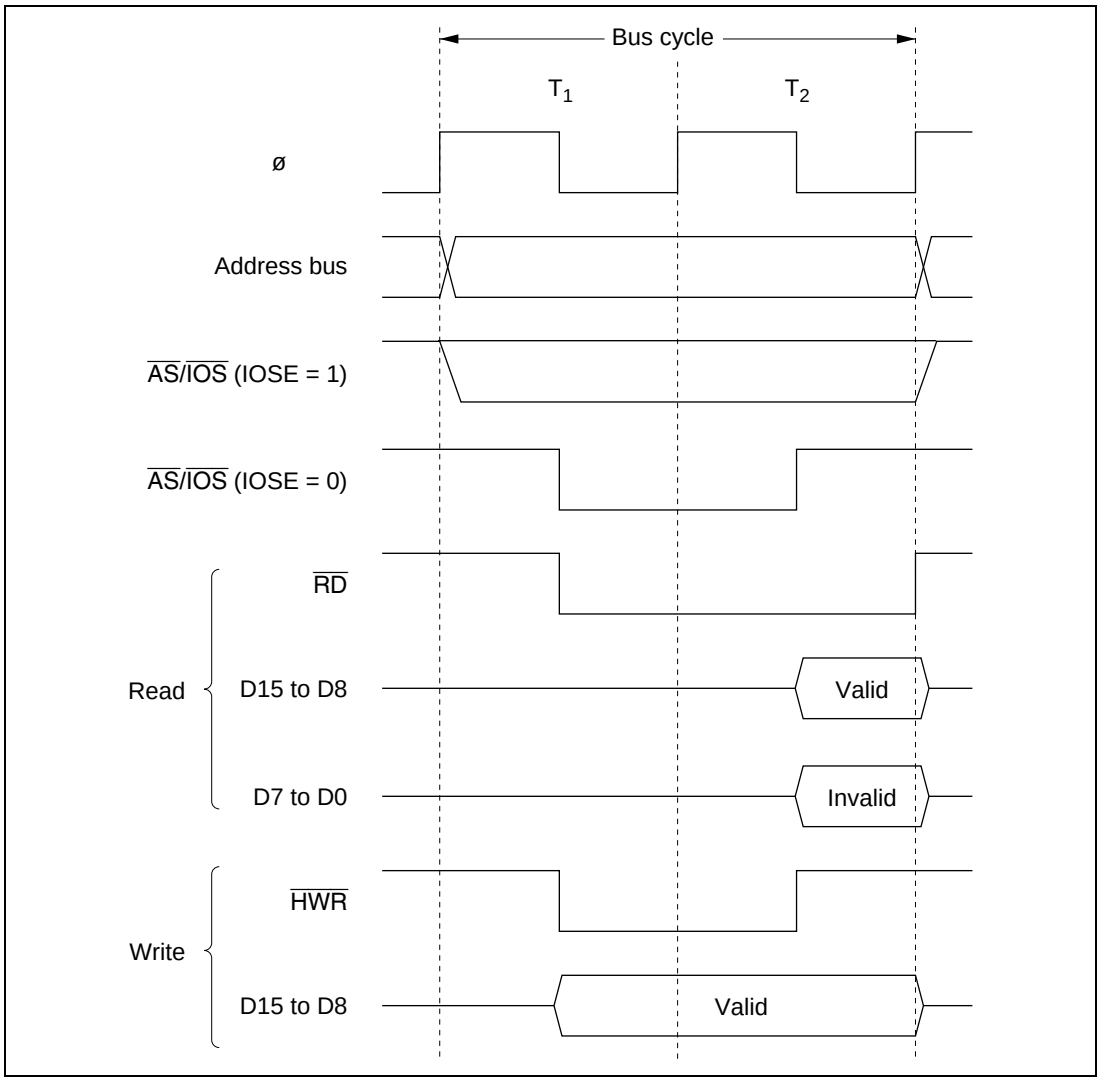


Figure 6.5 Bus Timing for 8-Bit 2-State Access Space

8-Bit 3-State Access Space: Figure 6.6 shows the bus timing for an 8-bit 3-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used.

Wait states can be inserted.

These series have no lower data bus (D7 to D0) pins or $\overline{\text{LWR}}$ pin. In these series, the upper data bus (D15 to D8) pins are designated D7 to D0, and the $\overline{\text{HWR}}$ signal pin is designated $\overline{\text{WR}}$.

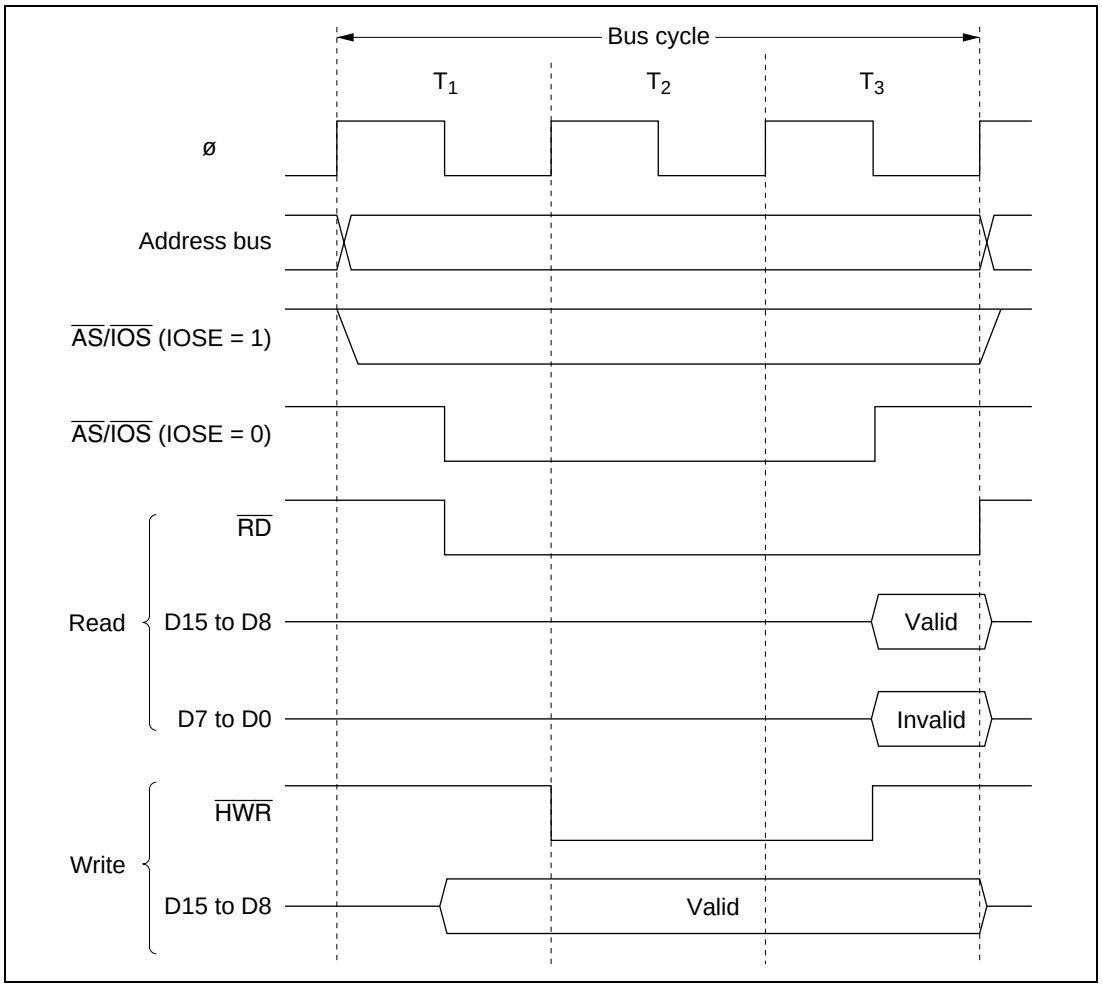


Figure 6.6 Bus Timing for 8-Bit 3-State Access Space

6.4.5 Wait Control

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

Program Wait Mode

In program wait mode, the number of T_w states specified by bits WC1 and WC0 are always inserted between the T_2 and T_3 states when external space is accessed.

Pin Wait Mode

In pin wait mode, the number of T_w states specified by bits WC1 and WC0 are always inserted between the T_2 and T_3 states when external space is accessed. If the $\overline{\text{WAIT}}$ pin is low at the fall of ø in the last T_2 or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

Pin wait mode is useful for inserting four or more wait states, or for changing the number of T_w states for different external devices.

Pin Auto-Wait Mode

In pin auto-wait mode, if the $\overline{\text{WAIT}}$ pin is low at the fall of the system clock in the T_2 state, the number of T_w states specified by bits WC1 and WC0 are inserted between the T_2 and T_3 states when external space is accessed. No additional T_w states are inserted even if the $\overline{\text{WAIT}}$ pin remains low. Pin auto-wait mode can be used for an easy interface to low-speed memory, simply by routing the chip select signal to the $\overline{\text{WAIT}}$ pin.

Figure 6.7 shows an example of wait state insertion timing.

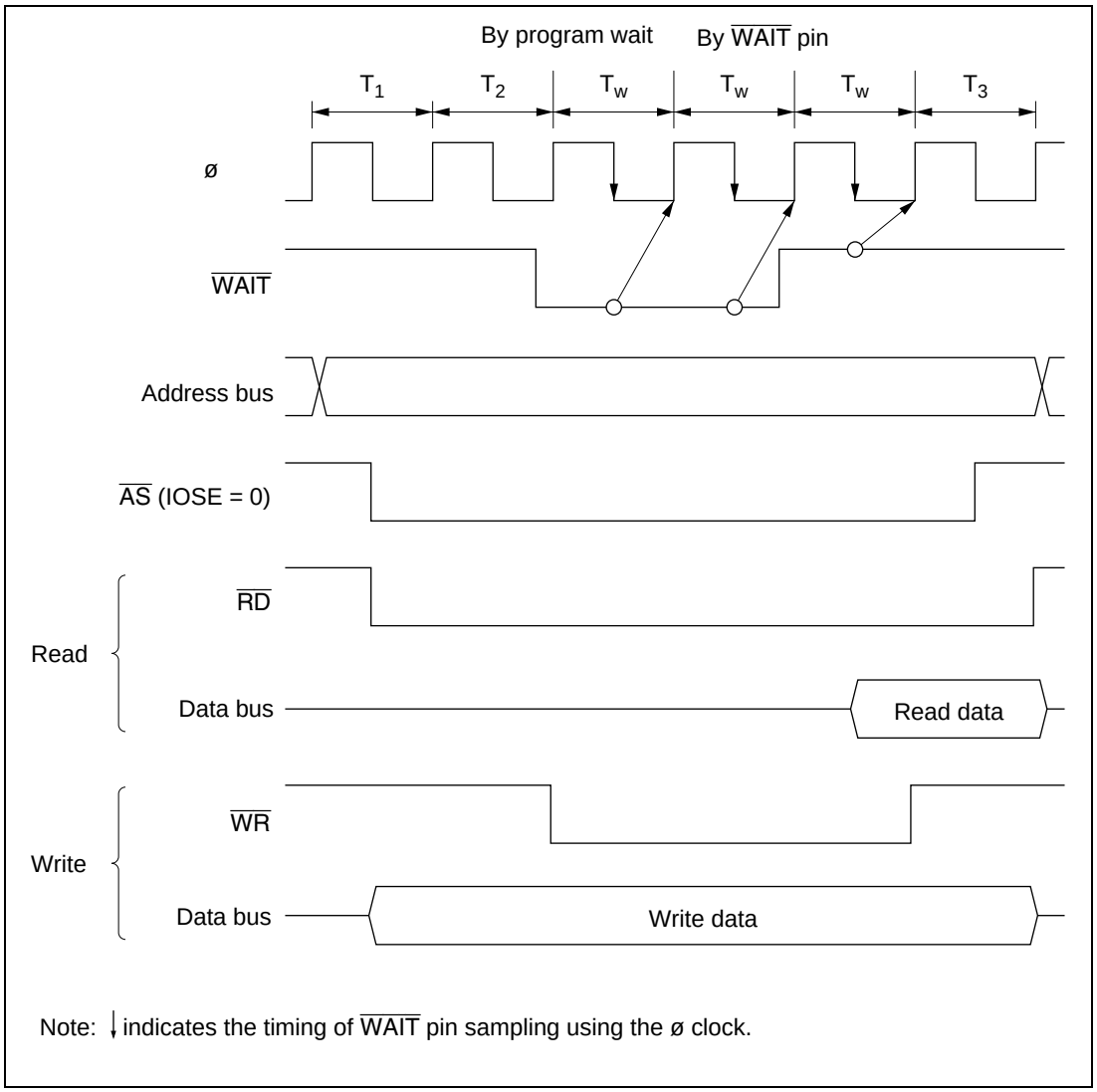


Figure 6.7 Example of Wait State Insertion Timing

The settings after a reset are: 3-state access, insertion of 3 program wait states, and WAIT input disabled.

6.5 Burst ROM Interface

6.5.1 Overview

With the H8S/2128 Series and H8S/2124 Series, external space area 0 can be designated as burst ROM space, and burst ROM interfacing can be performed.

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

6.5.2 Basic Timing

The number of states in the initial cycle (full access) of the burst ROM interface is in accordance with the setting of the AST bit. Also, when the AST 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 BCR. Wait states cannot be inserted.

When the BRSTS0 bit in BCR 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 figure 6.8 (a) and (b). The timing shown in figure 6.8 (a) is for the case where the AST and BRSTS1 bits are both set to 1, and that in figure 6.8 (b) is for the case where both these bits are cleared to 0.

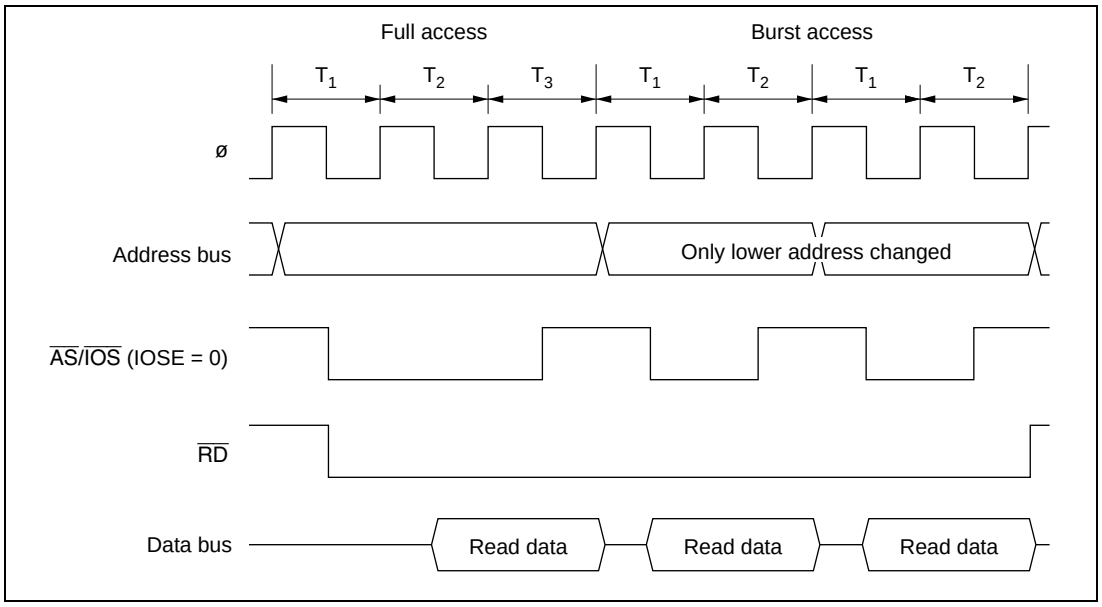


Figure 6.8 (a) Example of Burst ROM Access Timing (When AST = BRSTS1 = 1)

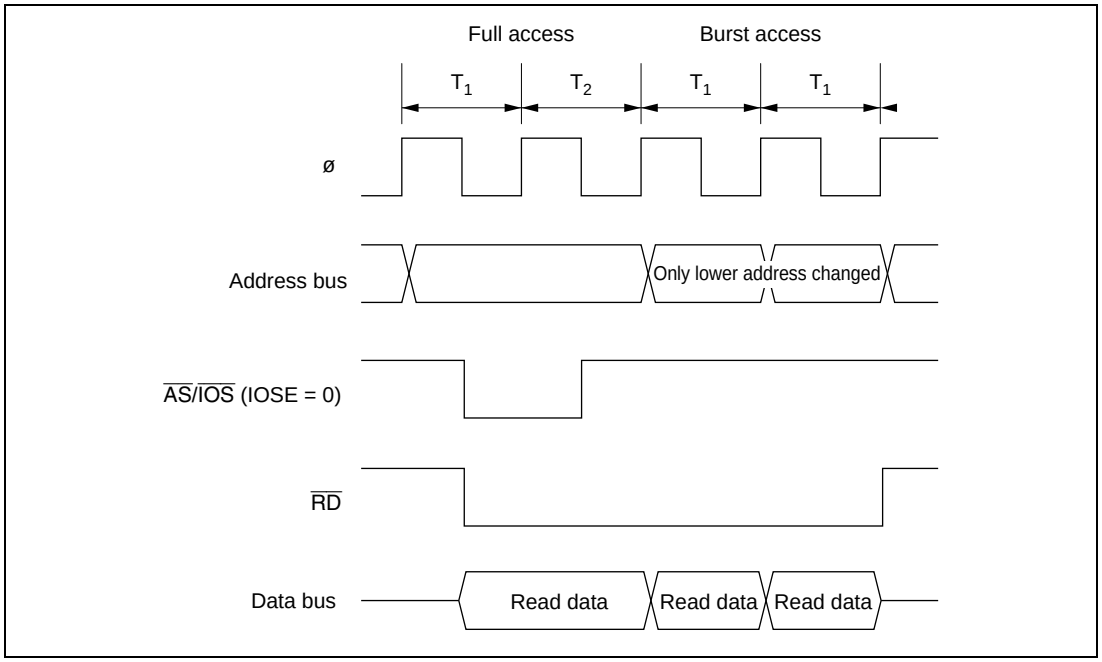


Figure 6.8 (b) Example of Burst ROM Access Timing (When AST = BRSTS1 = 0)

6.5.3 Wait Control

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

Wait states cannot be inserted in a burst cycle.

6.6 Idle Cycle

6.6.1 Operation

When the H8S/2128 Series or H8S/2124 Series chip accesses external space, it can insert a 1-state idle cycle (T_I) between bus cycles when a write cycle occurs immediately after a read cycle. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, with a long output floating time, and high-speed memory, I/O interfaces, and so on.

If an external write occurs after an external read while the ICIS0 bit in BCR is set to 1, an idle cycle is inserted at the start of the write cycle. This is enabled in advanced mode and normal mode.

Figure 6.9 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a CPU write cycle. In (a), an idle cycle is not inserted, and a collision occurs in cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.

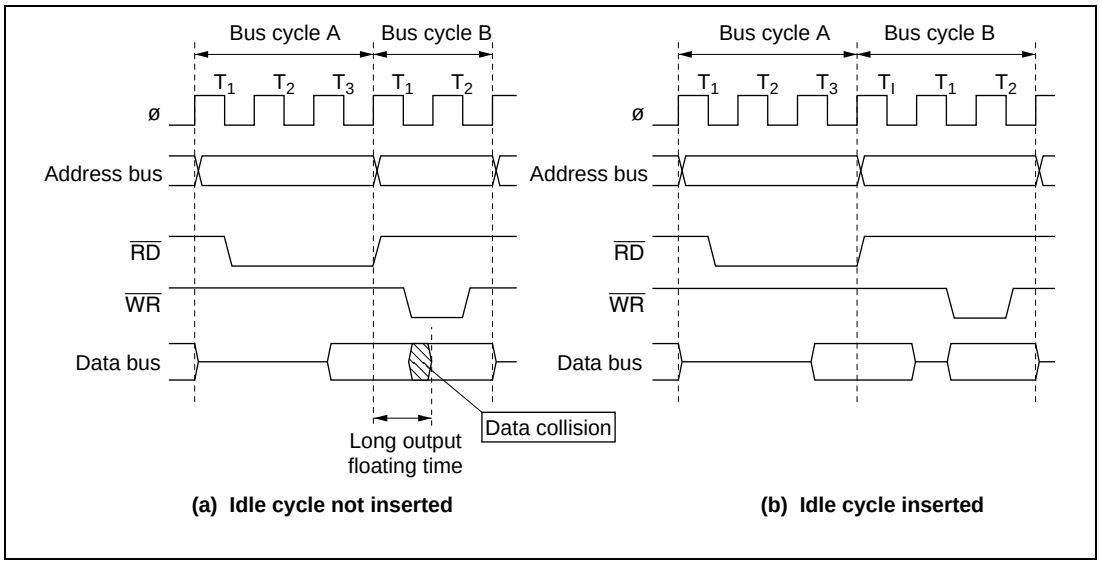


Figure 6.9 Example of Idle Cycle Operation

6.6.2 Pin States in Idle Cycle

Table 6.5 shows pin states in an idle cycle.

Table 6.5 Pin States in Idle Cycle

Pins	Pin State
A15 to A0, $\overline{IOS}$	Contents of next bus cycle
D7 to D0	High impedance
$\overline{AS}$	High
$\overline{RD}$	High
$\overline{WR}$	High

6.7 Bus Arbitration

6.7.1 Overview

The H8S/2128 Series and H8S/2124 Series have a bus arbiter that arbitrates bus master operations.

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

6.7.2 Operation

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

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

(High) DTC > CPU (Low)

6.7.3 Bus Transfer Timing

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

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

- The bus is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the operations.

  See appendix A.5, Bus States during Instruction Execution, for timings at which the bus is not transferred.
- If the CPU is executing a multiply or divide instruction or similar internal operation, it transfers the bus immediately.
- If the CPU is in sleep mode, it transfers the bus immediately.

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

The DTC does not release the bus until it has completed a series of processing operations.

Section 7 Data Transfer Controller [H8S/2128 Series]

Provided in the H8S/2128 Series; not provided in the H8S/2124 Series.

7.1 Overview

The H8S/2128 Series includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

7.1.1 Features

- 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 transfer source and destination addresses can be selected
- Direct specification of 16-Mbyte address space possible
  - — 24-bit transfer source and destination addresses can be specified
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
  - — An interrupt request can be issued to the CPU after one data transfer ends
  - — An interrupt request can be issued to the CPU after all specified data transfers have ended
- Activation by software is possible
- Module stop mode can be set
  - — The initial setting enables DTC registers to be accessed. DTC operation is halted by setting module stop mode

7.1.2 Block Diagram

Figure 7.1 shows a block diagram of the DTC.

The DTC's register information is stored in the on-chip RAM*. A 32-bit bus connects the DTC to the on-chip RAM (1 kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

Note: * When the DTC is used, the RAME bit in SYSCR must be set to 1.

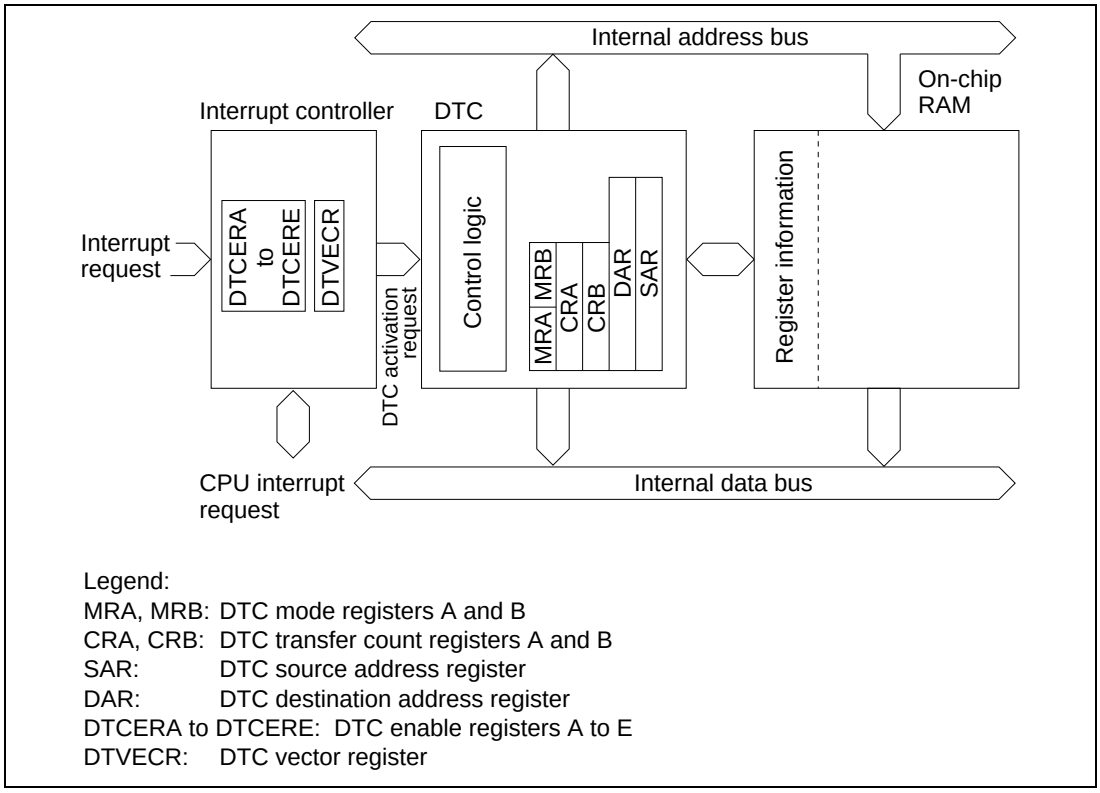


Figure 7.1 Block Diagram of DTC

7.1.3 Register Configuration

Table 7.1 summarizes the DTC registers.

Table 7.1 DTC Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
DTC mode register A	MRA	—*[2]	Undefined	—*[3]
DTC mode register B	MRB	—*[2]	Undefined	—*[3]
DTC source address register	SAR	—*[2]	Undefined	—*[3]
DTC destination address register	DAR	—*[2]	Undefined	—*[3]
DTC transfer count register A	CRA	—*[2]	Undefined	—*[3]
DTC transfer count register B	CRB	—*[2]	Undefined	—*[3]
DTC enable registers	DTCER*[4]	R/W	H'00	H'FEEE to H'FEF2
DTC vector register	DTVECR*[4]	R/W	H'00	H'FEF3
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

Notes: 1. Lower 16 bits of the address.

2. Registers within the DTC cannot be read or written to directly.
3. Allocated to on-chip RAM addresses H'EC00 to H'EFFF as register information.They cannot be located in external memory space.

   When the DTC is used, do not clear the RAME bit in SYSCR to 0.
4. The H8S/2124 Series does not include an on-chip DTC, and therefore the DTCER and DTVECR register addresses should not be accessed by the CPU.

7.2 Register Descriptions

7.2.1 DTC Mode Register A (MRA)

Bit	7	6	5	4	3	2	1	0
	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz
Initial value	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
Read/Write	—	—	—	—	—	—	—	—

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

Bits 7 and 6—Source Address Mode 1 and 0 (SM1, SM0): These bits specify whether SAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 7	Bit 6	
SM1	**SM0**	**Description**
0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by 1 when Sz = 0; by 2 when Sz = 1)
	1	SAR is decremented after a transfer (by 1 when Sz = 0; by 2 when Sz = 1)

Bits 5 and 4—Destination Address Mode 1 and 0 (DM1, DM0): These bits specify whether DAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 5	Bit 4	
DM1	**DM0**	**Description**
0	—	DAR is fixed
1	0	DAR is incremented after a transfer (by 1 when Sz = 0; by 2 when Sz = 1)
	1	DAR is decremented after a transfer (by 1 when Sz = 0; by 2 when Sz = 1)

Bits 3 and 2—DTC Mode (MD1, MD0): These bits specify the DTC transfer mode.

Bit 3	Bit 2	
MD1	**MD0**	**Description**
0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

Bit 1—DTC Transfer Mode Select (DTS): Specifies whether the source side or the destination side is set to be a repeat area or block area, in repeat mode or block transfer mode.

Bit 1	
DTS	**Description**
0	Destination side is repeat area or block area
1	Source side is repeat area or block area

Bit 0—DTC Data Transfer Size (Sz): Specifies the size of data to be transferred.

Bit 0	
Sz	**Description**
0	Byte-size transfer
1	Word-size transfer

7.2.2 DTC Mode Register B (MRB)

Bit	7	6	5	4	3	2	1	0
	CHNE	DISEL	—	—	—	—	—	—
Initial value	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
Read/Write	—	—	—	—	—	—	—	—

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

Bit 7—DTC Chain Transfer Enable (CHNE): Specifies chain transfer. In chain transfer, multiple data transfers can be performed consecutively in response to a single transfer request. With data transfer for which CHNE is set to 1, there is no determination of the end of the specified number of transfers, clearing of the interrupt source flag, or clearing of DTCER.

Bit 7

CHNE	Description
0	End of DTC data transfer (activation waiting state is entered)
1	DTC chain transfer (new register information is read, then data is transferred)

Bit 6—DTC Interrupt Select (DISEL): Specifies whether interrupt requests to the CPU are disabled or enabled after a data transfer.

Bit 6

DISEL	Description
0	After a data transfer ends, the CPU interrupt is disabled unless the transfer counter is 0 (the DTC clears the interrupt source flag of the activating interrupt to 0)
1	After a data transfer ends, the CPU interrupt is enabled (the DTC does not clear the interrupt source flag of the activating interrupt to 0)

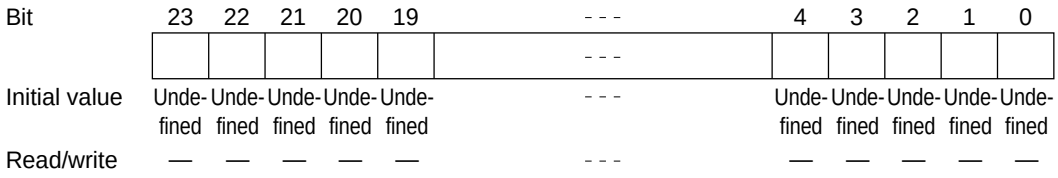
Bits 5 to 0—Reserved: In the H8S/2128 Series these bits have no effect on DTC operation, and should always be written with 0.

7.2.3 DTC Source Address Register (SAR)

Bit	23	22	21	20	19	- - -	4	3	2	1	0
						- - -					
Initial value	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined	- - -	Unde-fined	Unde-fined	Unde-fined	Unde-fined	Unde-fined
Read/write	—	—	—	—	—	- - -	—	—	—	—	—

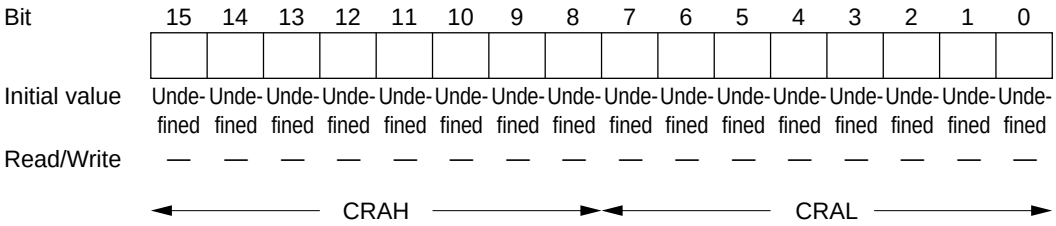
SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

7.2.4 DTC Destination Address Register (DAR)



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

7.2.5 DTC Transfer Count Register A (CRA)



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

In normal mode, the entire CRA register functions as a 16-bit transfer counter (1 to 65,536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

In repeat mode or block transfer mode, CRA is divided into two parts: the upper 8 bits (CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are transferred when the count reaches H'00. This operation is 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	1	0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65,536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

7.2.7 DTC Enable Registers (DTCER)

Bit	7	6	5	4	3	2	1	0
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

Bit n—DTC Activation Enable (DTCEn)

Bit n DTCEn	Description	
0	DTC activation by interrupt is disabled [Clearing conditions] • When data transfer ends with the DISEL bit set to 1 • When the specified number of transfers end	(Initial value)
1	DTC activation by interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified number of transfers have not ended	

(n = 7 to 0)

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

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

7.2.8 DTC Vector Register (DTVECR)

Bit	7	6	5	4	3	2	1	0
	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * A value of 1 can always be written to the SWDTE bit, but 0 can only be written after 1 is read.

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

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

Bit 7—DTC Software Activation Enable (SWDTE): Specifies enabling or disabling of DTC software activation. To clear the SWDTE bit by software, read SWDTE when set to 1, then write 0 in the bit.

Bit 7

SWDTE	Description	
0	DTC software activation is disabled	(Initial value)
	[Clearing condition]	
	When the DISEL bit is 0 and the specified number of transfers have not ended	
1	DTC software activation is enabled	
	[Holding conditions]	
	• When data transfer ends with the DISEL bit set to 1	
	• When the specified number of transfers end	
	• During software-activated data transfer	

Bits 6 to 0—DTC Software Activation Vectors 6 to 0 (DTVEC6 to DTVEC0): These bits specify a vector number for DTC software activation.

The vector address is H'0400 + (vector number) << 1 (where << 1 indicates a 1-bit left shift). For example, if DTVEC6 to DTVEC0 = H'10, the vector address is H'0420.

7.2.9 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	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	R/W	R/W

MSTPCR, comprising two 8-bit readable/writable registers, performs module stop mode control.

When the MSTP14 bit in MSTPCR is set to 1, the DTC operation stops at the end of the bus cycle and a transition is made to module stop mode. Note that 1 cannot be written to the MSTP14 bit when the DTC is being activated. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 6—Module Stop (MSTP14): Specifies the DTC module stop mode.

MSTPCRH Bit 6

MSTP14	Description	
0	DTC module stop mode is cleared	(Initial value)
1	DTC module stop mode is set	

7.3 Operation

7.3.1 Overview

When activated, the DTC reads register information that is already stored in memory and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to memory. Pre-storage of register information in memory makes it possible to transfer data over any required number of channels. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation.

Figure 7.2 shows a flowchart of DTC operation.

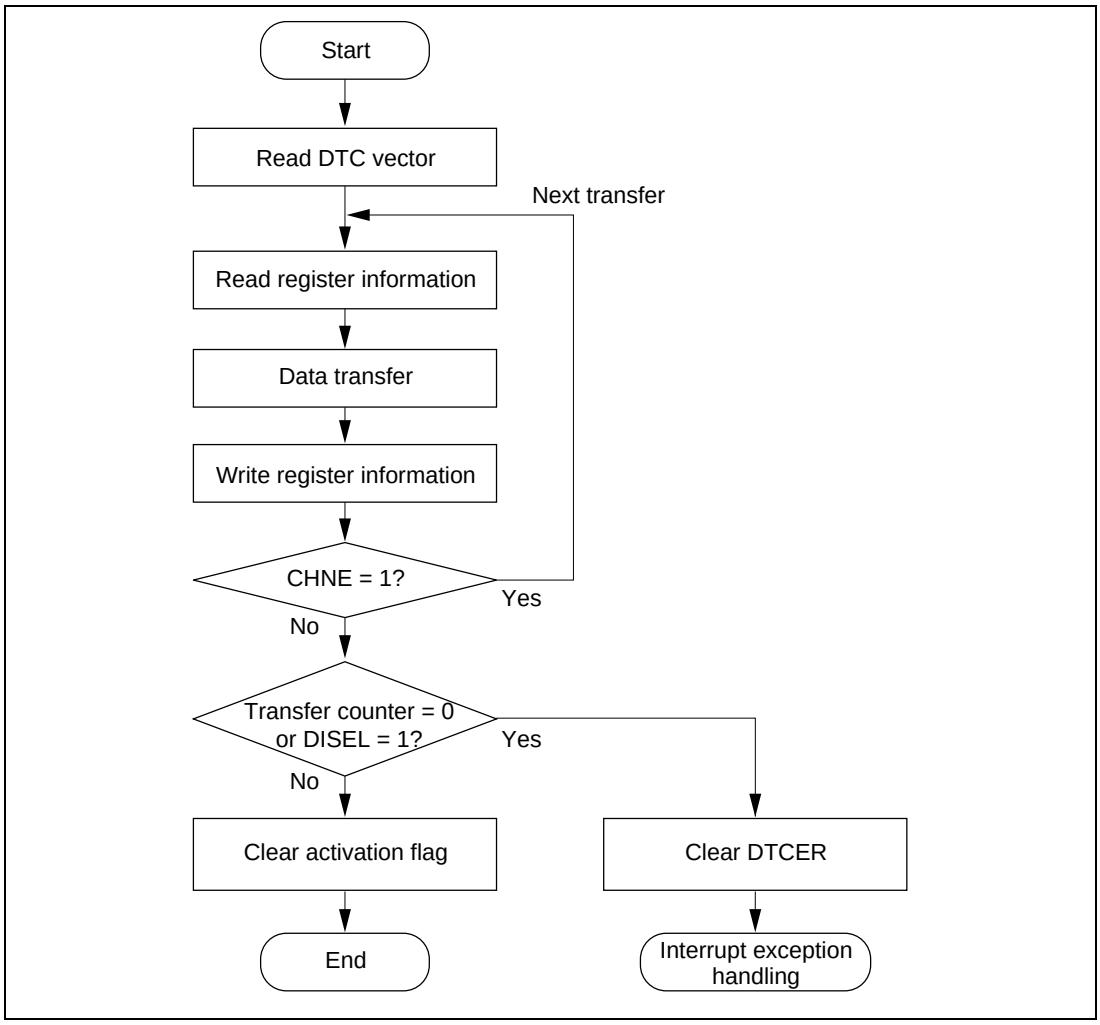


Figure 7.2 Flowchart of DTC Operation

The DTC transfer mode can be normal mode, repeat mode, or block transfer mode.

The 24-bit SAR designates the DTC transfer source address and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed.

Table 7.2 outlines the functions of the DTC.

Table 7.2 DTC Functions

		Address Registers	
Transfer Mode	**Activation Source**	**Transfer Source**	**Transfer Destination**
• Normal mode — One transfer request transfers one byte or one word — Memory addresses are incremented or decremented by 1 or 2 — Up to 65,536 transfers possible • Repeat mode — One transfer request transfers one byte or one word — Memory addresses are incremented or decremented by 1 or 2 — After the specified number of transfers (1 to 256), the initial state resumes and operation continues • Block transfer mode — One transfer request transfers a block of the specified size — Block size is from 1 to 256 bytes or words — Up to 65,536 transfers possible — A block area can be designated at either the source or destination	• IRQ • FRT ICI, OCI • 8-bit timer CMI • SCI TXI or RXI • A/D converter ADI • IIC IICI • Software	24 bits	24 bits

7.3.2 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by software (software activation). An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. The interrupt request is directed to the DTC when the corresponding bit is set to 1, and to the CPU when the bit is cleared to 0.

At the end of one data transfer (or the last of the consecutive transfers in the case of chain transfer) the interrupt source or the corresponding DTCER bit is cleared. Table 7.3 shows activation sources and DTCER clearing.

The interrupt source flag for RXI0, for example, is the RDRF flag in SCI0.

Table 7.3 Activation Sources and DTCER Clearing

Activation Source	When DISEL Bit Is 0 and Specified Number of Transfers Have Not Ended	When DISEL Bit Is 1 or Specified Number of Transfers Have Ended
Software activation	SWDTE bit cleared to 0	• SWDTE bit held at 1 • Interrupt request sent to CPU
Interrupt activation	• Corresponding DTCER bit held at 1 • Activation source flag cleared to 0	• Corresponding DTCER bit cleared to 0 • Activation source flag held at 1 • Activation source interrupt request sent to CPU

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

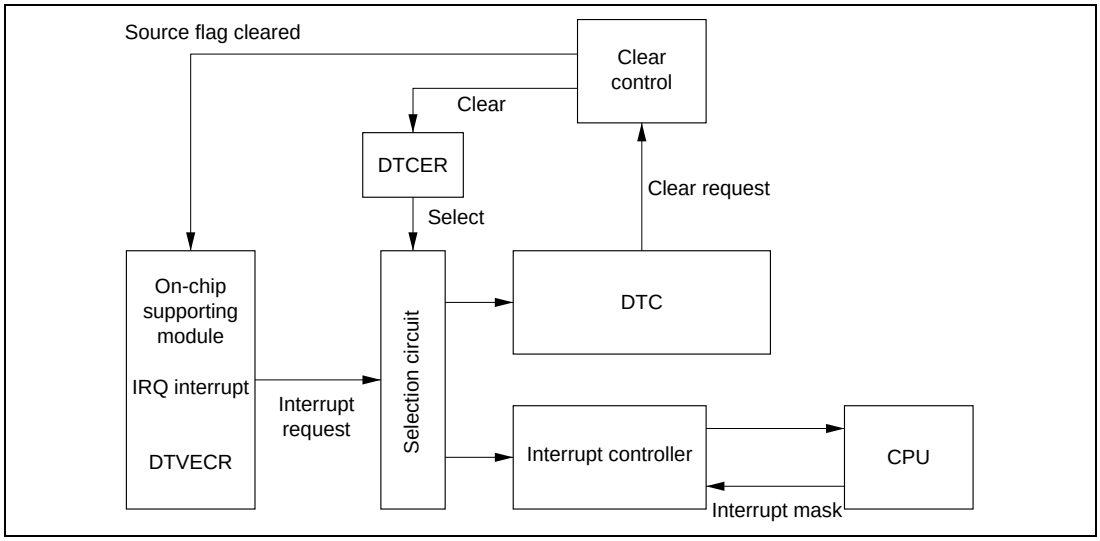


Figure 7.3 Block Diagram of DTC Activation Source Control

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

7.3.3 DTC Vector Table

Figure 7.4 shows the correspondence between DTC vector addresses and register information.

Table 7.4 shows the correspondence between activation sources, vector addresses, and DTCER bits. When the DTC is activated by software, the vector address is obtained from: H'0400 + DTVECR[6:0] << 1 (where << 1 indicates a 1-bit left shift). For example, if DTVECR is H'10, the vector address is H'0420.

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

The configuration of the vector address is the same in both normal and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the address in the on-chip RAM.

Table 7.4 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE*	Priority
Write to DTVECR	Software	DTVECR	H'0400 + DTVECR [6:0] << 1	—	High
IRQ0	External pin	16	H'0420	DTCEA7	
IRQ1		17	H'0422	DTCEA6	
IRQ2		18	H'0424	DTCEA5	
IRQ3		19	H'0426	DTCEA4	
ADI (A/D conversion end)	A/D	28	H'0438	DTCEA3	
ICIA (FRT input capture A)	FRT	48	H'0460	DTCEA2	
ICIB (FRT input capture B)		49	H'0462	DTCEA1	
OCIA (FRT output compare A)		52	H'0468	DTCEA0	
OCIB (FRT output compare B)		54	H'046A	DTCEB7	
CMIA0 (TMR0 compare-match A)	TMR0	64	H'0480	DTCEB2	
CMIB0 (TMR0 compare-match B)		65	H'0482	DTCEB1	
CMIA1 (TMR1 compare-match A)	TMR1	68	H'0488	DTCEB0	
CMIB1 (TMR1 compare-match B)		69	H'048A	DTCEC7	
CMIAY (TMRY compare-match A)	TMRY	72	H'0490	DTCEC6	
CMIBY (TMRY compare-match B)		73	H'0492	DTCEC5	
RXI0 (reception completed 0)	SCI channel 0	81	H'04A2	DTCEC2	
TXI0 (transmit data empty 0)		82	H'04A4	DTCEC1	
RXI1 (reception completed 1)	SCI channel 1	85	H'04AA	DTCEC0	
TXI1 (transmit data empty 1)		86	H'04AC	DTCED7	
IICI0 (IIC0 1-byte transmission/ reception completed)	IIC0 (option)	92	H'04B8	DTCED4	
IICI1 (IIC1 1-byte transmission/ reception completed)	IIC1 (option)	94	H'04BC	DTCED3	Low

Note: *DTCE bits with no corresponding interrupt are reserved, and should be written with 0.

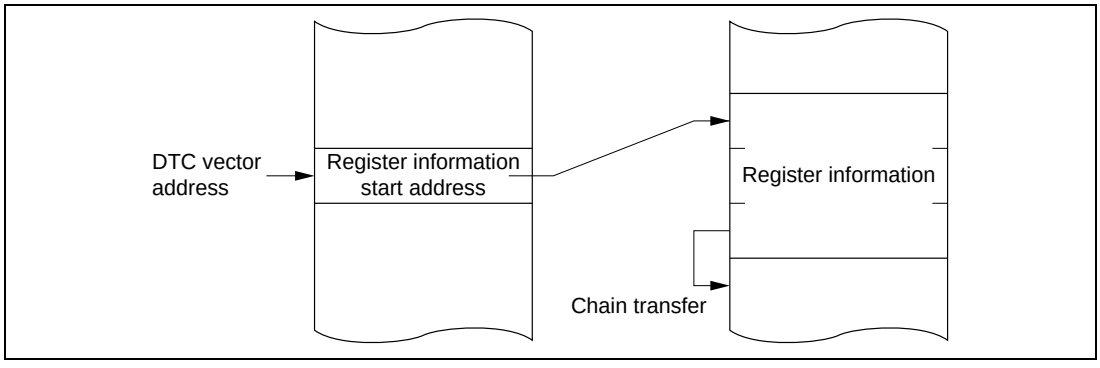


Figure 7.4 Correspondence between DTC Vector Address and Register Information

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 start address of the register information (vector address contents). In chain transfer, locate the register information in consecutive areas.

Locate the register information in the on-chip RAM (addresses: H'FFEC00 to H'FFEFFF).

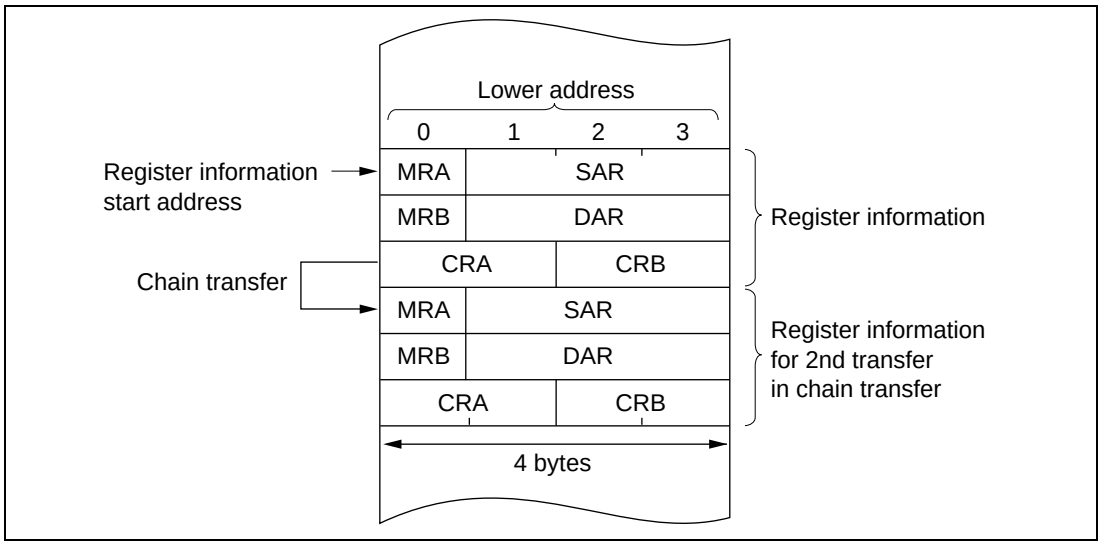


Figure 7.5 Location of DTC Register Information in Address Space

7.3.5 Normal Mode

In normal mode, one operation transfers one byte or one word of data.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt can be requested.

Table 7.5 lists the register information in normal mode and figure 7.6 shows memory mapping in normal mode.

Table 7.5 Register Information in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination address
DTC transfer count register A	CRA	Transfer count
DTC transfer count register B	CRB	Not used

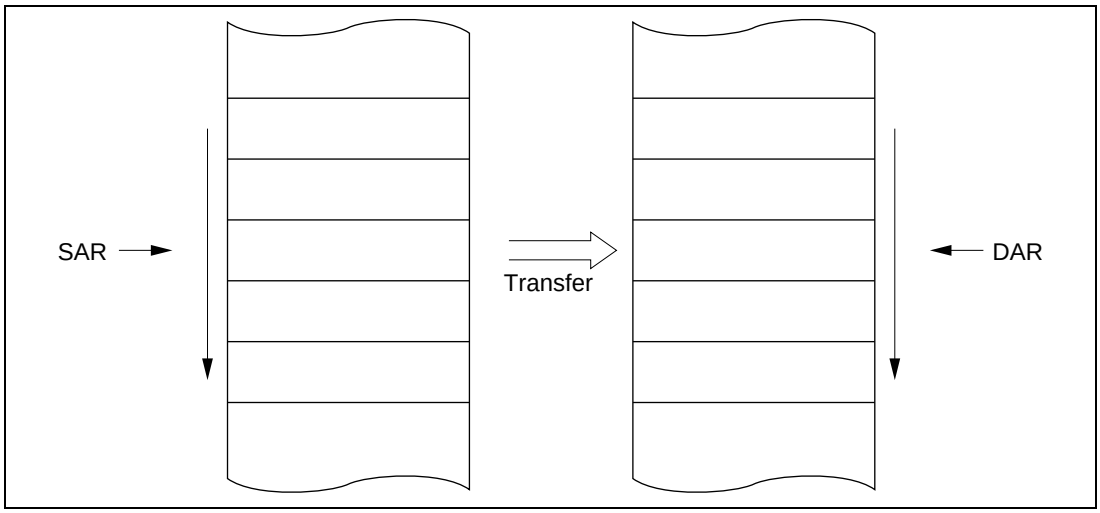


Figure 7.6 Memory Mapping in Normal Mode

7.3.6 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data.

From 1 to 256 transfers can be specified. Once the specified number of transfers have ended, the initial address register state specified by the transfer counter and repeat area resumes and transfer is repeated. In repeat mode the transfer counter does not reach H'00, and therefore CPU interrupts cannot be requested when DISEL = 0.

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

Table 7.6 Register Information in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination address
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Transfer count
DTC transfer count register B	CRB	Not used

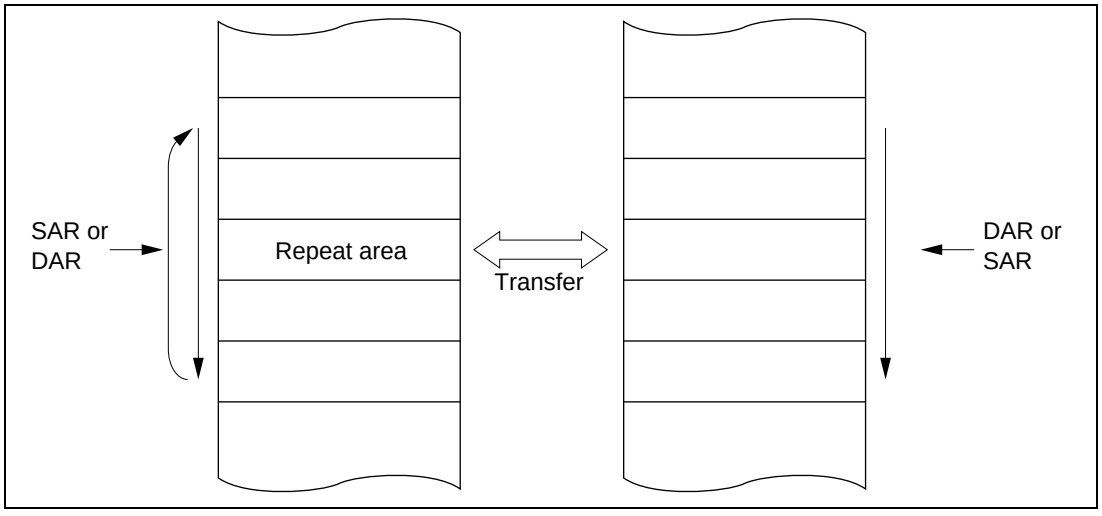


Figure 7.7 Memory Mapping in Repeat Mode

7.3.7 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is specified as a block area.

The block size is 1 to 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified in the block area is restored. The other address register is successively incremented or decremented, or left fixed.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt is requested.

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

Table 7.7 Register Information in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Transfer source address
DTC destination address register	DAR	Transfer destination address
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Block size count
DTC transfer count register B	CRB	Transfer counter

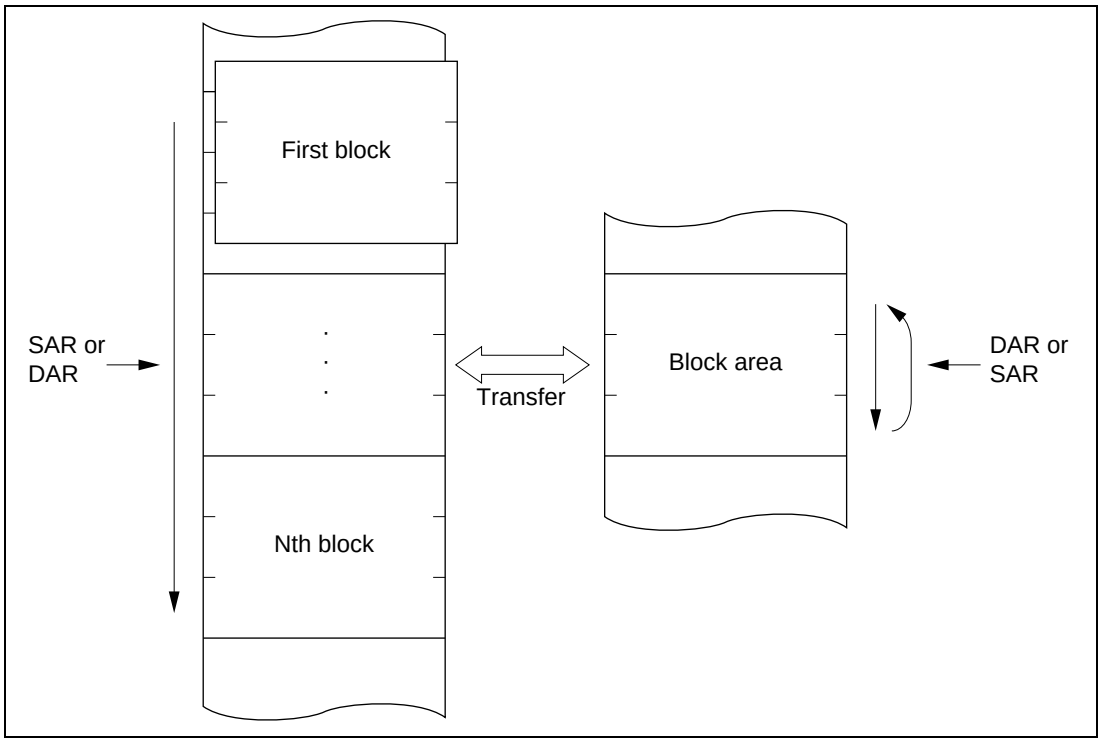


Figure 7.8 Memory Mapping in Block Transfer Mode

7.3.8 Chain Transfer

Setting the CHNE bit to 1 enables a number of data transfers to be performed consecutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 7.9 shows memory mapping for chain transfer.

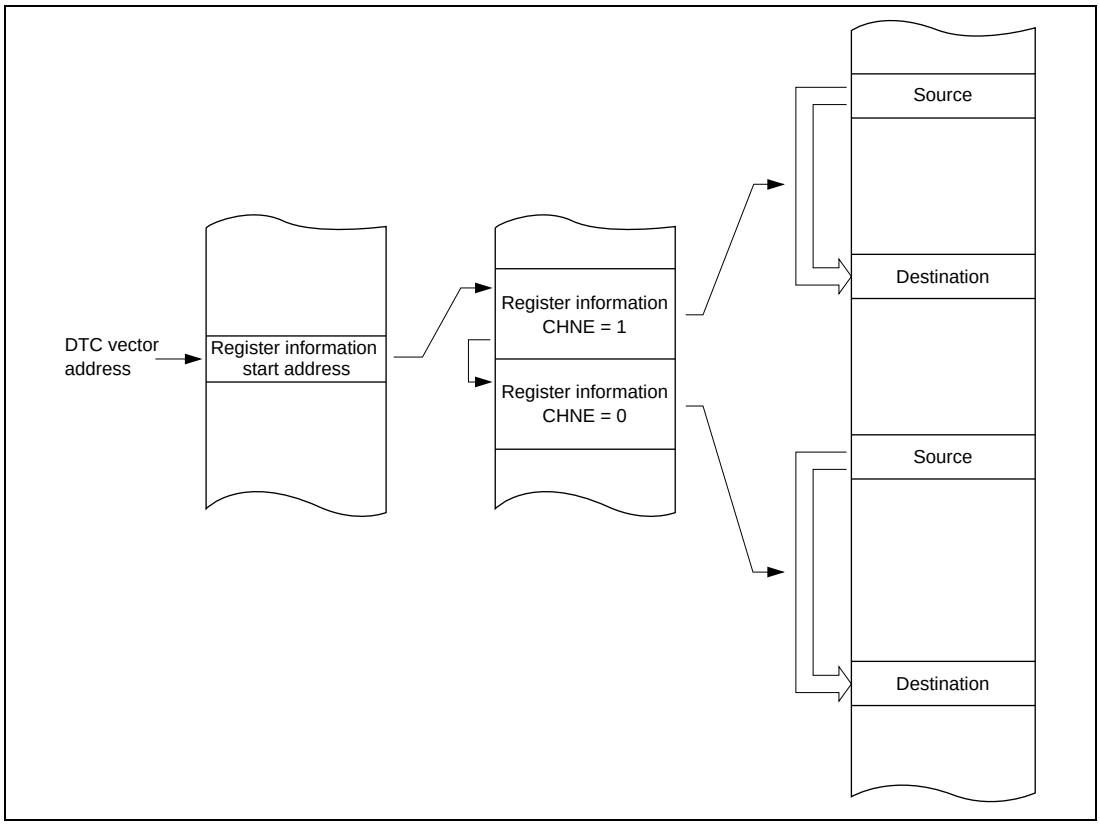


Figure 7.9 Memory Mapping in Chain Transfer

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

7.3.9 Operation Timing

Figures 7.10 to 7.12 show examples of DTC operation timing.

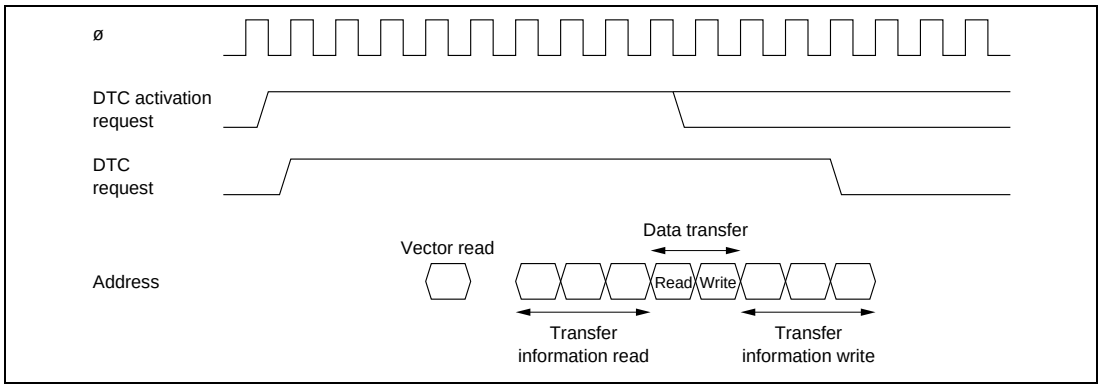


Figure 7.10 DTC Operation Timing (Normal Mode or Repeat Mode)

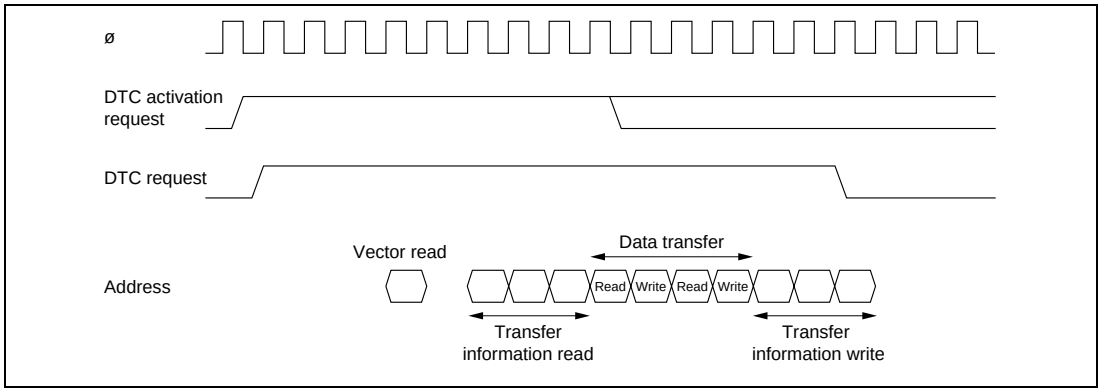


Figure 7.11 DTC Operation Timing (Block Transfer Mode, with Block Size of 2)

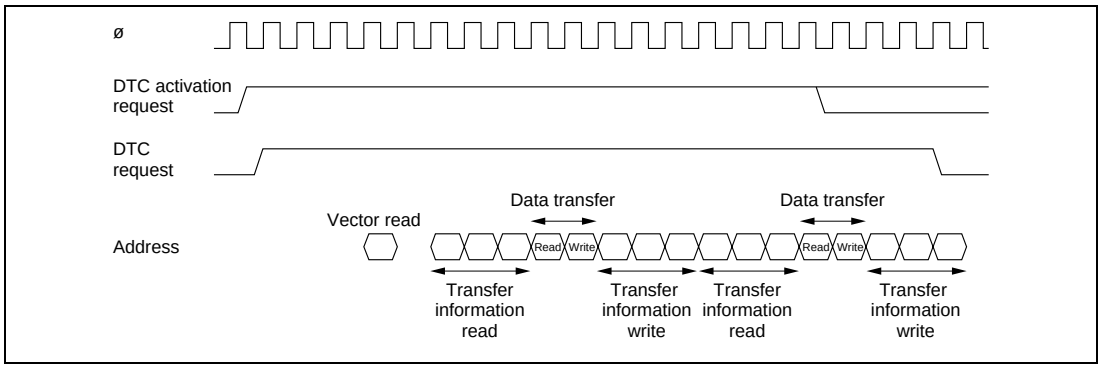


Figure 7.12 DTC Operation Timing (Chain Transfer)

7.3.10 Number of DTC Execution States

Table 7.8 lists execution phases for a single DTC data transfer, and table 7.9 shows the number of states required for each execution phase.

Table 7.8 DTC Execution Phases

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operation M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

N: Block size (initial setting of CRAH and CRAL)

Table 7.9 Number of States Required for Each Execution Phase

Object of Access			On-Chip RAM	On-Chip ROM	Internal I/O Registers		External Devices	
Bus width			32	16	8	16	8	8
Access states			1	1	2	2	2	3
Execution phase	Vector read	S_I	—	1	—	—	4	6+2m
	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	1	1	1	1	1

The number of execution states is calculated from the formula below. Note that Σ means the sum of all transfers activated by one activation event (the number for which the CHNE bit is set to one, plus 1).

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

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

7.3.11 Procedures for Using the DTC

Activation by Interrupt: The procedure for using the DTC with interrupt activation is as follows:

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.
5. After the end of one data transfer, or after the specified number of data transfers have ended, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

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

1. Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.
2. Set the start address of the register information in the DTC vector address.
3. Check that the SWDTE bit is 0.
4. Write 1 in the SWDTE bit and the vector number to DTVECR.
5. Check the vector number written to DTVECR.
6. After the end of one data transfer, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data transfers have ended, the SWDTE bit is held at 1 and a CPU interrupt is requested.

7.3.12 Examples of Use of the DTC

Normal Mode: An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

1. Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the SCI RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.
2. Set the start address of the register information at the DTC vector address.
3. Set the corresponding bit in DTCER to 1.
4. Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.
5. Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.
6. When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine should perform wrap-up processing.

Software Activation: An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

1. Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.
2. Set the start address of the register information at the DTC vector address (H'04C0).
3. Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.
4. Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.
5. Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.

6. If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.
7. After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform other wrap-up processing.

7.4 Interrupts

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

In the case of activation by software, a software-activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers have ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine should clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

7.5 Usage Notes

Module Stop: When the MSTP14 bit in MSTPCR is set to 1, the DTC clock stops, and the DTC enters the module stop state. However, 1 cannot be written in the MSTP14 bit while the DTC is operating. When the DTC is placed in the module stop state, the DTCER registers must all be in the cleared state when the MSTP14 bit is set to 1.

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

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

Section 8 I/O Ports

8.1 Overview

The H8S/2128 Series and H8S/2124 Series have six I/O ports (ports 1 to 6), and one input-only port (port 7).

Tables 8.1 and 8.2 summarize the port functions. The pins of each port also have other functions.

Each port includes a data direction register (DDR) that controls input/output (not provided for the input-only port) and data registers (DR) that store output data.

Ports 1 to 3 have a built-in MOS input pull-up function. Ports 1 to 3 have a MOS input pull-up control register (PCR), in addition to DDR and DR, to control the on/off status of the MOS input pull-ups.

Ports 1 to 6 can drive a single TTL load and 30 pF capacitive load. All the I/O ports can drive a Darlington transistor when in output mode. Ports 1 to 3 can drive an LED (10 mA sink current).

In the H8S/2128 Series, P52 in port 5 and P47 in port 4 are NMOS push-pull outputs.

Note that the H8S/2124 Series has subset specifications that do not include some supporting modules. For differences in pin functions, see table 8.1, H8S/2128 Series Port Functions, and table 8.2, H8S/2124 Series Port Functions.

Table 8.1 H8S/2128 Series Port Functions

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 1	• 8-bit I/O port • Built-in MOS input pull-ups • LED drive capability	P17 to P10/ A7 to A0/ PW7 to PW0/ PWX1, PWX0	Lower address output (A7 to A0)	When DDR = 0 (after reset): input port When DDR = 1: lower address output (A7 to A0) or PWM timer output (PW7 to PW0, PWX1, PWX0)	I/O port also functioning as PWM timer output (PW7 to PW0, PWX1, PWX0)
Port 2	• 8-bit I/O port • Built-in MOS input pull-ups • LED drive capability	P27/A15/PW15/ SCK1/CBLANK P26/A14/PW14/ RxD1 P25/A13/PW13/ TxD1 P24/A12/PW12/ SCL1 P23/A11/PW11/ SDA1 P22/A10/PW10 P21/A9/PW9 P20/A8/PW8	Upper address output (A15 to A8)	When DDR = 0 (after reset): input port or timer connection output (CBLANK) When DDR = 1: upper address output (A15 to A8), PWM timer output (PW15 to PW8), or timer connection output (CBLANK), or output ports (P27 to P24)	I/O port also functioning as PWM timer output (PW15 to PW8), SCI1 I/O pins (TxD1, RxD1, SCK1) and timer connection output (CBLANK), I^2C bus interface 1 (option) I/O pins (SCL1, SDA1), and I/O port
Port 3	• 8-bit I/O port • Built-in MOS input pull-ups • LED drive capability	P37 to P30/ D7 to D0	Data bus input/output (D7 to D0)		I/O port

Table 8.1 H8S/2128 Series Port Functions (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 4	• 8-bit I/O port	P47/$\overline{\text{WAIT}}$/SDA0	I/O port also functioning as expanded data bus control input ($\overline{\text{WAIT}}$) and I²C bus interface 0 (option) input/output (SDA0)		I/O port also functioning as I²C bus interface 0 (option) input/output (SDA0)
		P46/ø/EXCL	When DDR = 0: input port or EXCL input When DDR = 1 (after reset): ø output	When DDR = 0 (after reset): input port or EXCL input When DDR = 1: ø output	
		P45/$\overline{\text{AS}}$/$\overline{\text{IOS}}$ P44/$\overline{\text{WR}}$ P43/$\overline{\text{RD}}$	Expanded data bus control output ($\overline{\text{AS}}$/$\overline{\text{IOS}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$)		I/O port
		P42/$\overline{\text{IRQ0}}$ P41/$\overline{\text{IRQ1}}$	I/O port also functioning as external interrupt input ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$)		
		P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	I/O port also functioning as external interrupt input ($\overline{\text{IRQ2}}$), and A/D converter external trigger input ($\overline{\text{ADTRG}}$)		
Port 5	• 3-bit I/O port	P52/SCK0/SCL0 P51/RxD0 P50/TxD0	I/O port also functioning as SCI0 input/output (TxD0, RxD0, SCK0) and I²C bus interface 0 (option) input/output (SCL0)		

Table 8.1 H8S/2128 Series Port Functions (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 6	• 8-bit I/O port	P67/TMOX/ TMO1/CIN7/ HSYNCO P66/FTOB/ TMRI1/CIN6/ CSYNCI P65/FTID/TMCI1/ CIN5/HSYNCI P64/FTIC/TMO0/ CIN4/CLAMPO P63/FTIB/TMRI0/CI N3/VFBACKI P62/FTIA/TMIY/ CIN2/VSYNCI P61/FTOA/CIN1/ VSYNCO P60/FTCI/TMIX/ TMCI0/CIN0/ HFBACKI	I/O port also functioning as FRT input/output (FTCI, FTOA, FTIA, FTIB, FTIC, FTID, FTOB), 8-bit timer 0 and 1 input/output (TMCI0, TMRI0, TMO0, TMCI1, TMRI1, TMO1), 8-bit timer X and Y input/output (TMOX, TMIX, TMIY), timer connection input/output (HSYNCO, CSYNCI, HSYNCI, CLAMPO, VFBACKI, VSYNCI, VSYNCO, HFBACKI), and expansion A/D converter input (CIN7 to CIN0)		
Port 7	• 8-bit input port	P77/AN7 P76/AN6 P75/AN5 P74/AN4 P73/AN3 P72/AN2 P71/AN1 P70/AN0	Input port also functioning as A/D converter analog input (AN7 to AN0)		

Table 8.2 H8S/2124 Series Port Functions

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2, Mode 3 (EXPE = 1)**	**Mode 2, Mode 3 (EXPE = 0)**
Port 1	• 8-bit I/O port • Built-in MOS input pull-ups • LED drive capability	P17 to P10/ A7 to A0	Lower address output (A7 to A0)	When DDR = 0 (after reset): input port When DDR = 1: lower address output (A7 to A0)	I/O port
Port 2	• 8-bit I/O port • Built-in MOS input pull-ups • LED drive capability	P27/A15/SCK1 P26/A14/RxD1 P25/A13/TxD1 P24/A12 P23/A11 P22/A10 P21/A9 P20/A8	Upper address output (A15 to A8)	When DDR = 0 (after reset): input port When DDR = 1: upper address output (A15 to A8), or output ports (P27 to P24)	I/O port also functioning as SCI1 I/O pins (TxD1, RxD1, SCK1)
Port 3	• 8-bit I/O port • Built-in MOS input pull-ups • LED drive capability	P37 to P30/ D7 to D0	Data bus input/output (D7 to D0)		I/O port
Port 4	• 8-bit I/O port	P47/$\overline{\text{WAIT}}$	I/O port also functioning as expanded data bus control input ($\overline{\text{WAIT}}$)		I/O port
		P46/ø/EXCL	When DDR = 0: input port or EXCL input When DDR = 1 (after reset): ø output	When DDR = 0 (after reset): input port or EXCL input When DDR = 1: ø output	
		P45/$\overline{\text{AS}}$/$\overline{\text{IOS}}$ P44/$\overline{\text{WR}}$ P43/$\overline{\text{RD}}$	Expanded data bus control output($\overline{\text{AS}}$/$\overline{\text{IOS}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$)		I/O port
		P42/$\overline{\text{IRQ}}$0 P41/$\overline{\text{IRQ}}$1	I/O port also functioning as external interrupt input ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$)		

Table 8.2 H8S/2124 Series Port Functions (cont)

Port	Description	Pins	Expanded Modes: Mode 1	Expanded Modes: Mode 2, Mode 3 (EXPE = 1)	Single-Chip Mode: Mode 2, Mode 3 (EXPE = 0)
Port 4	• 8-bit I/O port	P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$	I/O port also functioning as external interrupt input ($\overline{\text{IRQ2}}$), and A/D converter external trigger input ($\overline{\text{ADTRG}}$)		I/O port also functioning as external interrupt input ($\overline{\text{IRQ2}}$) and A/D converter external trigger input ($\overline{\text{ADTRG}}$)
Port 5	• 3-bit I/O port	P52/SCK0 P51/RxD0 P50/TxD0	I/O port also functioning as SCI0 input/output (TxD0, RxD0, SCK0)		
Port 6	• 8-bit I/O port	P67/TMO1/CIN7 P66/FTOB/TMRI1/CIN6 P65/FTID/TMCI1/CIN5 P64/FTIC/TMO0/CIN4 P63/FTIB/TMRI0/CIN3 P62/FTIA/TMIY/CIN2 P61/FTOA/CIN1 P60/FTCI/TMCI0/CIN0	I/O port also functioning as FRT input/output (FTCI, FTOA, FTIA, FTIB, FTIC, FTID, FTOB), 8-bit timer 0 and 1 input/output (TMCI0, TMRI0, TMO0, TMCI1, TMRI1, TMO1), 8-bit timer Y input (TMIY), and expansion A/D converter input (CIN7 to CIN0)		
Port 7	• 8-bit input port	P77/AN7 P76/AN6 P75/AN5 P74/AN4 P73/AN3 P72/AN2 P71/AN1 P70/AN0	Input port also functioning as A/D converter analog input (AN7 to AN0)		

8.2 Port 1

8.2.1 Overview

Port 1 is an 8-bit I/O port. Port 1 pins also function as address bus output pins as 8-bit PWM output pins (PW7 to PW0) (H8S/2128 Series only), and as 14-bit PWM output pins (PWX1 to PWX0) (H8S/2128 Series only). Port 1 functions change according to the operating mode. Port 1 has a built-in MOS input pull-up function that can be controlled by software.

Figure 8.1 shows the port 1 pin configuration.

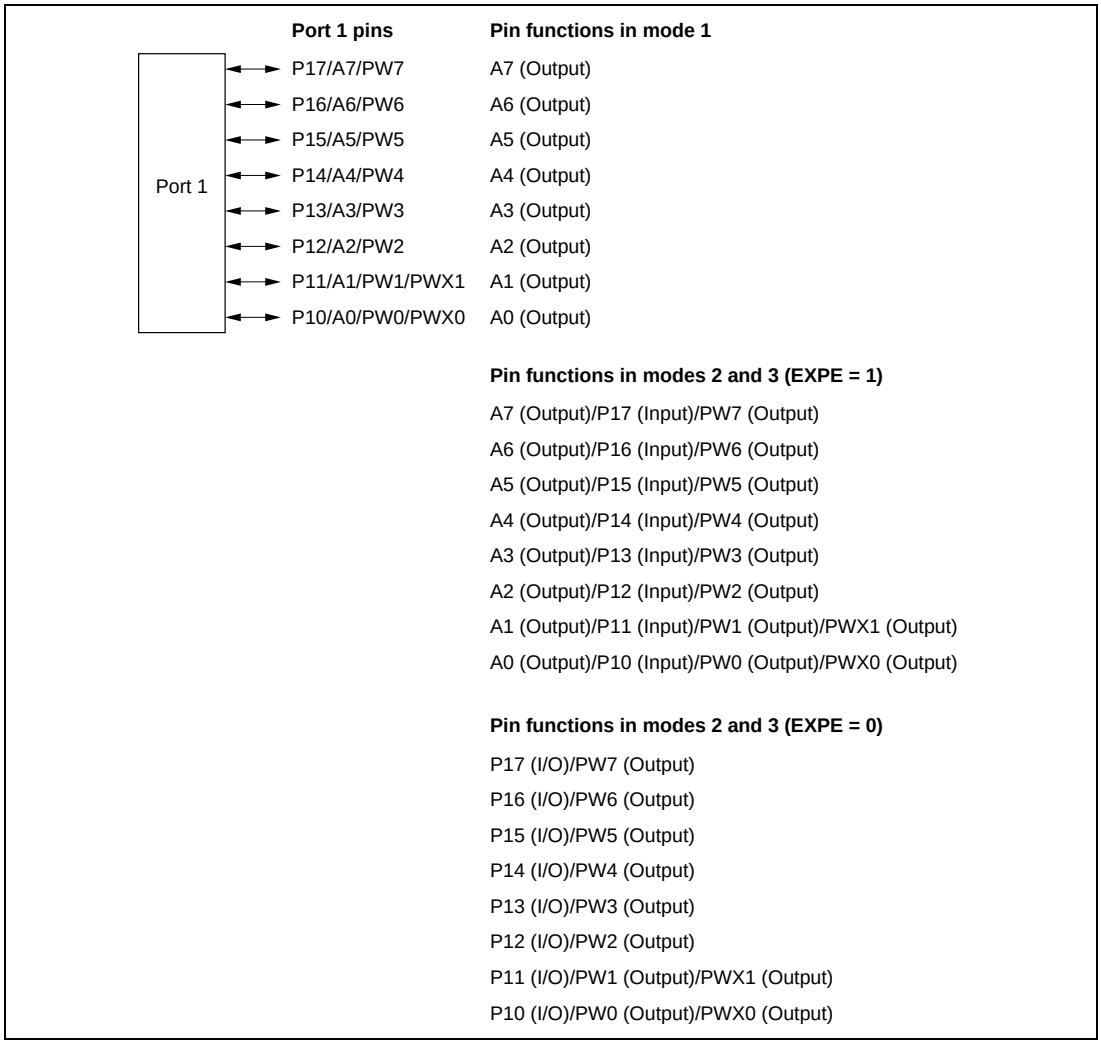


Figure 8.1 Port 1 Pin Functions

8.2.2 Register Configuration

Table 8.3 shows the port 1 register configuration.

Table 8.3 Port 1 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 1 data direction register	P1DDR	W	H'00	H'FFB0
Port 1 data register	P1DR	R/W	H'00	H'FFB2
Port 1 MOS pull-up control register	P1PCR	R/W	H'00	H'FFAC

Note: *Lower 16 bits of the address.

Port 1 Data Direction Register (P1DDR)

Bit	7	6	5	4	3	2	1	0
	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

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

P1DDR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode. The address output pins maintain their output state in a transition to software standby mode.

- Mode 1

  The corresponding port 1 pins are address outputs, regardless of the P1DDR setting.

  In hardware standby mode, the address outputs go to the high-impedance state.
- Modes 2 and 3 (EXPE = 1)

  The corresponding port 1 pins are address outputs or PWM outputs when P1DDR bits are set to 1, and input ports when cleared to 0.

  P10 and P11 can be designated as PWMX outputs regardless of P1DDR, but to ensure normal execution of external space accesses, this designation should not be used.
- Modes 2 and 3 (EXPE = 0)

  The corresponding port 1 pins are output ports or PWM outputs when P1DDR bits are set to 1, and input ports when cleared to 0.

  P10 and P11 can be designated as PWMX outputs regardless of P1DDR.

Port 1 Data Register (P1DR)

Bit	7	6	5	4	3	2	1	0
	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1DR is an 8-bit readable/writable register that stores output data for the port 1 pins (P17 to P10). If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read directly, regardless of the actual pin states. If a port 1 read is performed while P1DDR bits are cleared to 0, the pin states are read.

P1DR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

Port 1 MOS Pull-Up Control Register (P1PCR)

Bit	7	6	5	4	3	2	1	0
	P17PCR	P16PCR	P15PCR	P14PCR	P13PCR	P12PCR	P11PCR	P10PCR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1PCR is an 8-bit readable/writable register that controls the port 1 built-in MOS input pull-ups on a bit-by-bit basis.

In modes 2 and 3, the MOS input pull-up is turned on when a P1PCR bit is set to 1 while the corresponding P1DDR bit is cleared to 0 (input port setting).

P1PCR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

8.2.3 Pin Functions in Each Mode

Mode 1: In mode 1, port 1 pins automatically function as address outputs. The port 1 pin functions are shown in figure 8.2.

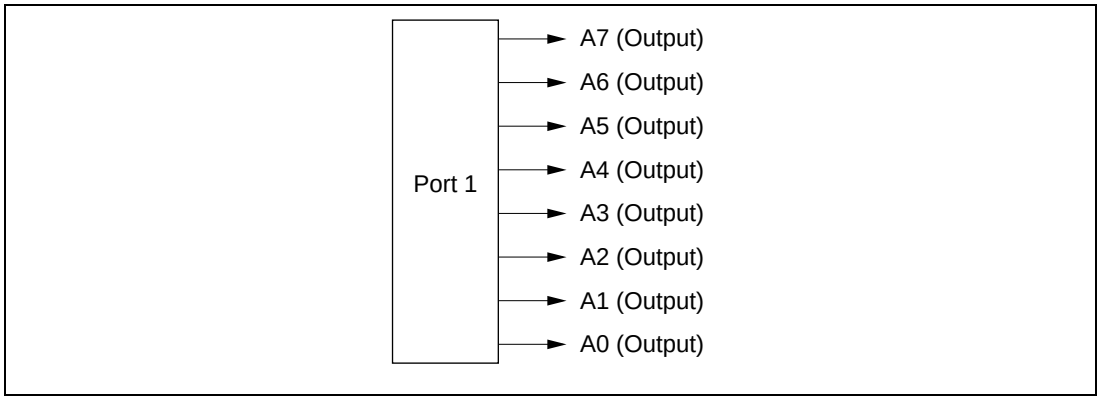


Figure 8.2 Port 1 Pin Functions (Mode 1)

Modes 2 and 3 (EXPE = 1): In modes 2 and 3 (when EXPE = 1), port 1 pins function as address outputs, PWM outputs, or input ports, and input or output can be specified on a bit-by-bit basis. When a bit in P1DDR is set to 1, the corresponding pin functions as an address output or PWM output, and when cleared to 0, as an input port. P10 and P11 can be designated as PWMX outputs regardless of P1DDR, but to ensure normal execution of external space accesses, this designation should not be used.

The port 1 pin functions are shown in figure 8.3.

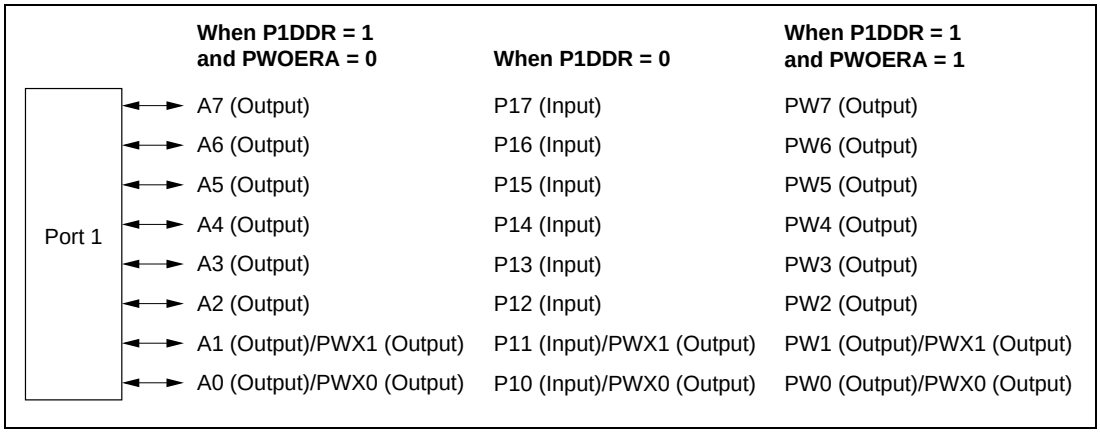


Figure 8.3 Port 1 Pin Functions (Modes 2 and 3 (EXPE = 1))

Modes 2 and 3 (EXPE = 0): In modes 2 and 3 (when EXPE = 0), port 1 pins function as PWM outputs or I/O ports, and input or output can be specified on a bit-by-bit basis. When a bit in P1DDR is set to 1, the corresponding pin functions as a PWM output or output port, and when cleared to 0, as an input port. P10 and P11 can be designated as PWMX outputs regardless of P1DDR.

The port 1 pin functions are shown in figure 8.4.

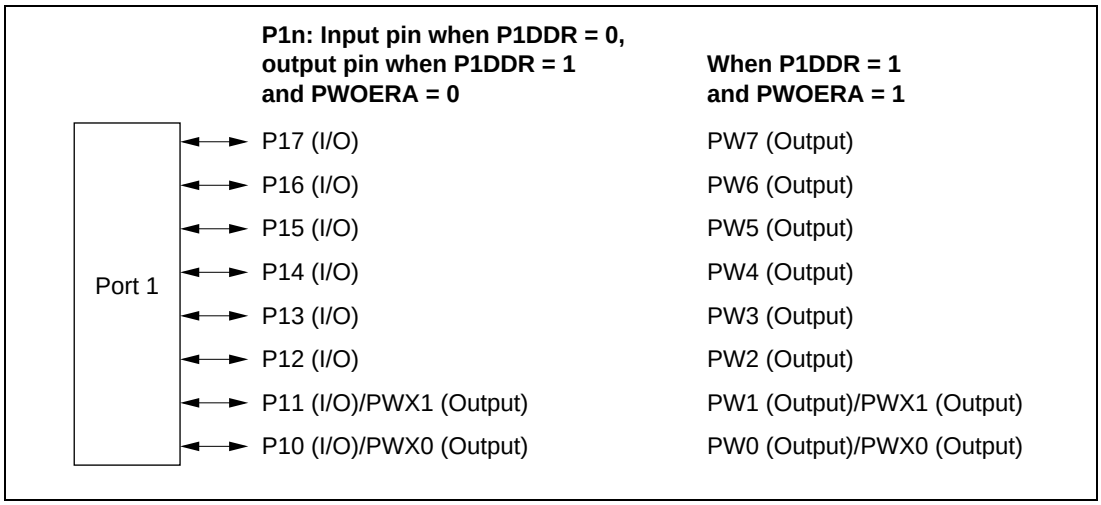


Figure 8.4 Port 1 Pin Functions (Modes 2 and 3 (EXPE = 0))

8.2.4 MOS Input Pull-Up Function

Port 1 has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 2 and 3, and can be specified as on or off on a bit-by-bit basis.

When a P1DDR bit is cleared to 0 in mode 2 or 3, setting the corresponding P1PCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset and in hardware standby mode. The prior state is retained in software standby mode.

Table 8.4 summarizes the MOS input pull-up states.

Table 8.4 MOS Input Pull-Up States (Port 1)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1	Off	Off	Off	Off
2, 3	Off	Off	On/Off	On/Off

Legend:
Off: MOS input pull-up is always off.
On/Off: On when P1DDR = 0 and P1PCR = 1; otherwise off.

8.3 Port 2

8.3.1 Overview

Port 2 is an 8-bit I/O port. Port 2 pins also function as address bus output pins, 8-bit PWM output pins (PW15 to PW8) (H8S/2128 Series only), the timer connection output pin (CBLANK) (H8S/2128 Series only), IIC1 I/O pins (SCL1, SDA1) (option in H8S/2128 Series only), and SCI1 I/O pins (SCK1, RxD1, TxD1). Port 2 functions change according to the operating mode. Port 2 has a built-in MOS input pull-up function that can be controlled by software.

Figure 8.5 shows the port 2 pin configuration.

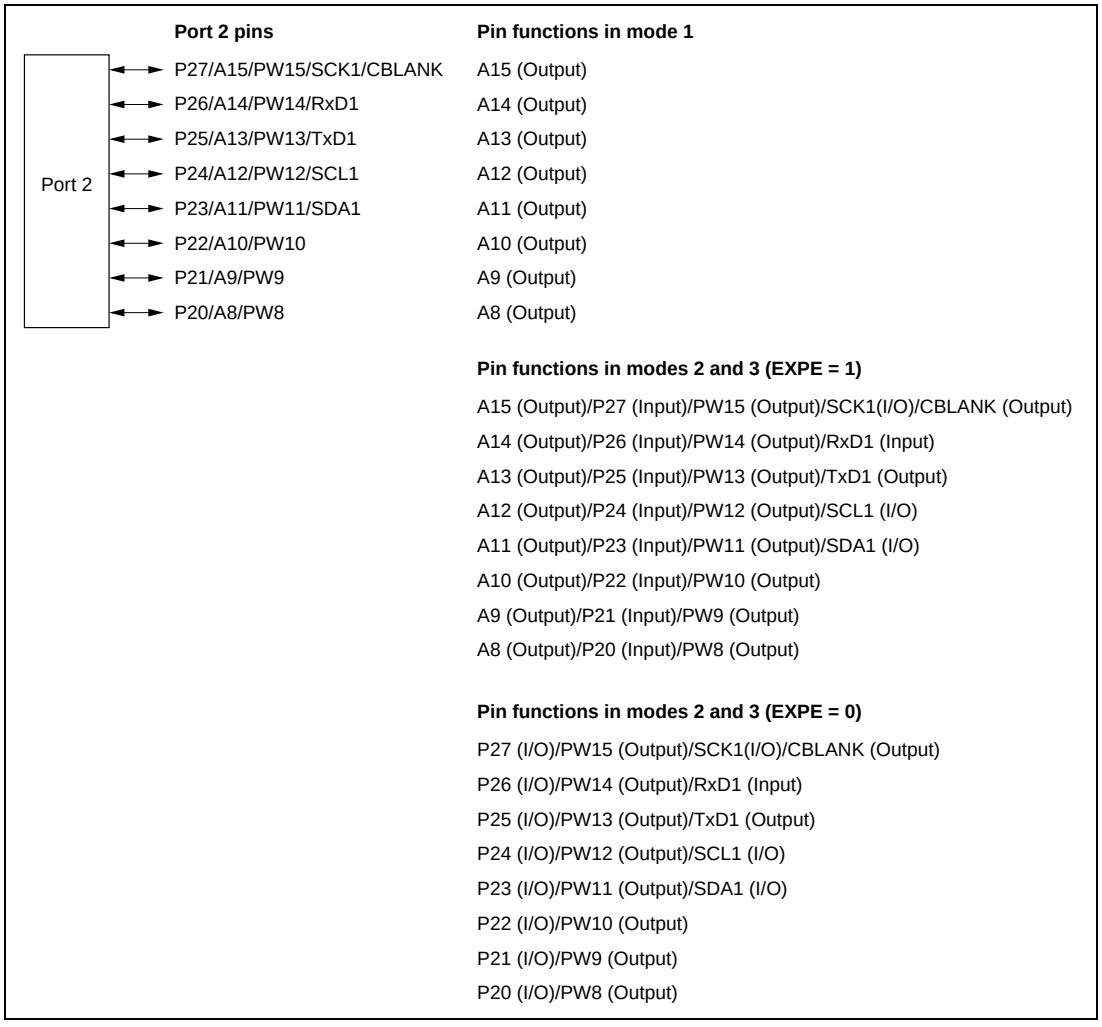


Figure 8.5 Port 2 Pin Functions

8.3.2 Register Configuration

Table 8.5 shows the port 2 register configuration.

Table 8.5 Port 2 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 2 data direction register	P2DDR	W	H'00	H'FFB1
Port 2 data register	P2DR	R/W	H'00	H'FFB3
Port 2 MOS pull-up control register	P2PCR	R/W	H'00	H'FFAD

Note: *Lower 16 bits of the address.

Port 2 Data Direction Register (P2DDR)

Bit	7	6	5	4	3	2	1	0
	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

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

P2DDR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode. The address output pins maintain their output state in a transition to software standby mode.

- Mode 1

  The corresponding port 2 pins are address outputs, regardless of the P2DDR setting.

  In hardware standby mode, the address outputs go to the high-impedance state.
- Modes 2 and 3 (EXPE = 1)

  The corresponding port 2 pins are address outputs or PWM outputs when P2DDR bits are set to 1, and input ports when cleared to 0. P27 to P24 are switched from address outputs to output ports by setting the IOSE bit to 1.

  P27 to P23 can be used as an on-chip supporting module output pin regardless of the P2DDR setting, but to ensure normal access to external space, P27 should not be set as an on-chip supporting module output pin when port 2 pins are used as address output pins.
- Modes 2 and 3 (EXPE = 0)

  The corresponding port 2 pins are output ports or PWM outputs when P2DDR bits are set to 1, and input ports when cleared to 0.

P27 to P23 can be used as an on-chip supporting module output pin regardless of the P2DDR setting.

Port 2 Data Register (P2DR)

Bit	7	6	5	4	3	2	1	0
	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2DR is an 8-bit readable/writable register that stores output data for the port 2 pins (P27 to P20). If a port 2 read is performed while P2DDR bits are set to 1, the P2DR values are read directly, regardless of the actual pin states. If a port 2 read is performed while P2DDR bits are cleared to 0, the pin states are read.

P2DR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

Port 2 MOS Pull-Up Control Register (P2PCR)

Bit	7	6	5	4	3	2	1	0
	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PCR
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2PCR is an 8-bit readable/writable register that controls the port 2 built-in MOS input pull-ups on a bit-by-bit basis.

In modes 2 and 3, the MOS input pull-up is turned on when a P2PCR bit is set to 1 while the corresponding P2DDR bit is cleared to 0 (input port setting).

P2PCR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

8.3.3 Pin Functions in Each Mode

Mode 1: In mode 1, port 2 pins automatically function as address outputs. The port 2 pin functions are shown in figure 8.6.

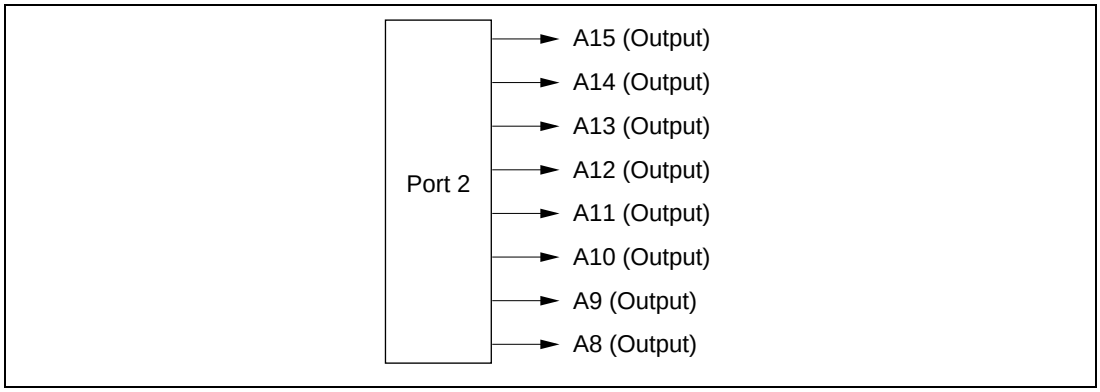


Figure 8.6 Port 2 Pin Functions (Mode 1)

Modes 2 and 3 (EXPE = 1): In modes 2 and 3 (when EXPE = 1), port 2 pins function as address outputs, PWM outputs, or I/O ports, and input or output can be specified on a bit-by-bit basis. When a bit in P2DDR is set to 1, the corresponding pin functions as an address output or PWM output, and when cleared to 0, as an input port. P27 to P24 are switched from address outputs to output ports by setting the IOSE bit to 1. P27 to P23 can be used as an on-chip supporting module output pin regardless of the P2DDR setting, but to ensure normal access to external space, P27 should not be set as an on-chip supporting module output pin when port 2 pins are used as address output pins.

The port 2 pin functions are shown in figure 8.7.

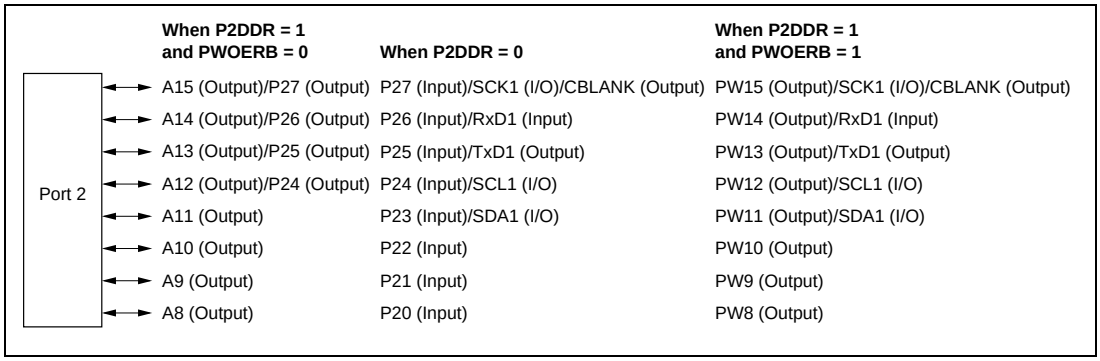


Figure 8.7 Port 2 Pin Functions (Modes 2 and 3 (EXPE = 1))

Modes 2 and 3 (EXPE = 0): In modes 2 and 3 (when EXPE = 0), port 2 pins function as PWM outputs, the timer connection output (CBLANK), IIC1 I/O pins (SCL1, SDA1), SCI1 I/O pins (SCK1, RxD1, TxD1), or I/O ports, and input or output can be specified on a bit-by-bit basis. When a bit in P2DDR is set to 1, the corresponding pin functions as a PWM output or output port, and when cleared to 0, as an input port. P27 to P23 can be used as an on-chip supporting module output pin regardless of the P2DDR setting.

The port 2 pin functions are shown in figure 8.8.

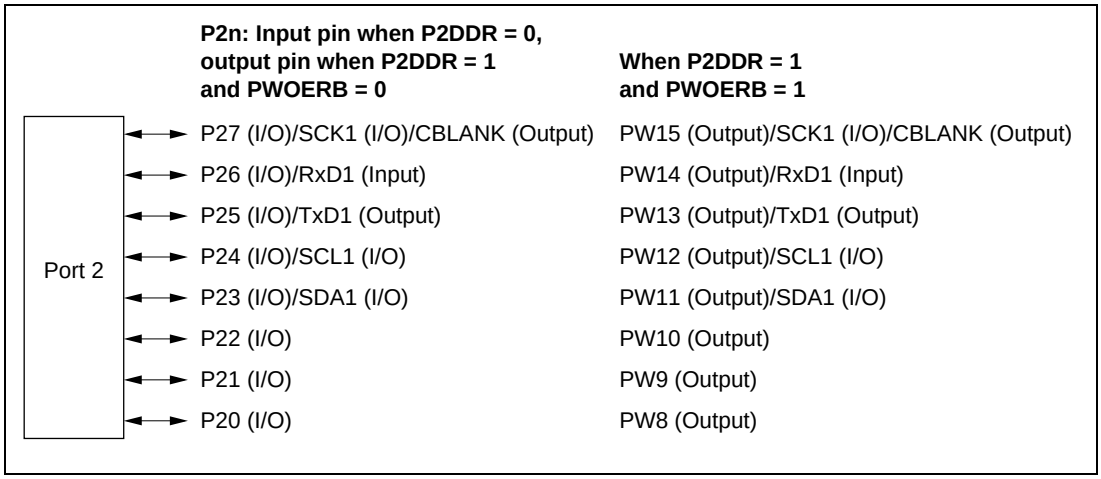


Figure 8.8 Port 2 Pin Functions (Modes 2 and 3 (EXPE = 0))

8.3.4 MOS Input Pull-Up Function

Port 2 has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 2 and 3, and can be specified as on or off on a bit-by-bit basis.

When a P2DDR bit is cleared to 0 in mode 2 or 3, setting the corresponding P2PCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset and in hardware standby mode. The prior state is retained in software standby mode.

Table 8.6 summarizes the MOS input pull-up states.

Table 8.6 MOS Input Pull-Up States (Port 2)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1	Off	Off	Off	Off
2, 3	Off	Off	On/Off	On/Off

Legend:
Off: MOS input pull-up is always off.
On/Off: On when P2DDR = 0 and P2PCR = 1; otherwise off.

8.4 Port 3

8.4.1 Overview

Port 3 is an 8-bit I/O port. Port 3 pins also function as data bus I/O pins. Port 3 functions change according to the operating mode. Port 3 has a built-in MOS input pull-up function that can be controlled by software.

Figure 8.9 shows the port 3 pin configuration.

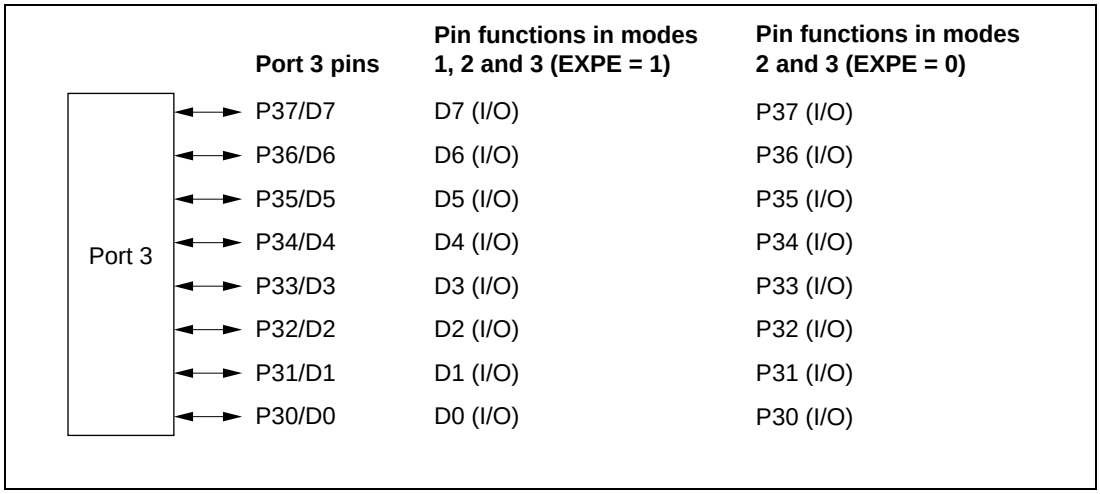


Figure 8.9 Port 3 Pin Functions

8.4.2 Register Configuration

Table 8.7 shows the port 3 register configuration.

Table 8.7 Port 3 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 3 data direction register	P3DDR	W	H'00	H'FFB4
Port 3 data register	P3DR	R/W	H'00	H'FFB6
Port 3 MOS pull-up control register	P3PCR	R/W	H'00	H'FFAE

Note: *Lower 16 bits of the address.

Port 3 Data Direction Register (P3DDR)

Bit	7	6	5	4	3	2	1	0
	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

P3DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 3. P3DDR cannot be read; if it is, an undefined value will be returned.

P3DDR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

- Modes 1, 2, and 3 (EXPE = 1)

  The input/output direction specified by P3DDR is ignored, and pins automatically function as data I/O pins.

  After a reset, and in hardware standby mode or software standby mode, the data I/O pins go to the high-impedance state.
- Modes 2 and 3 (EXPE = 0)

  The corresponding port 3 pins are output ports when P3DDR bits are set to 1, and input ports when cleared to 0.

Port 3 Data Register (P3DR)

Bit	7	6	5	4	3	2	1	0
	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3DR is an 8-bit readable/writable register that stores output data for the port 3 pins (P37 to P30). If a port 3 read is performed while P3DDR bits are set to 1, the P3DR values are read directly, regardless of the actual pin states. If a port 3 read is performed while P3DDR bits are cleared to 0, the pin states are read.

P3DR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

Port 3 MOS Pull-Up Control Register (P3PCR)

Bit	7	6	5	4	3	2	1	0
	P37PCR	P36PCR	P35PCR	P34PCR	P33PCR	P32PCR	P31PCR	P30PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3PCR is an 8-bit readable/writable register that controls the port 3 built-in MOS input pull-ups on a bit-by-bit basis.

In modes 2 and 3 (when EXPE = 0), the MOS input pull-up is turned on when a P3PCR bit is set to 1 while the corresponding P3DDR bit is cleared to 0 (input port setting).

P3PCR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

8.4.3 Pin Functions in Each Mode

Modes 1, 2, and 3 (EXPE = 1): In modes 1, 2, and 3 (when EXPE = 1), port 3 pins automatically function as data I/O pins. The port 3 pin functions are shown in figure 8.10.

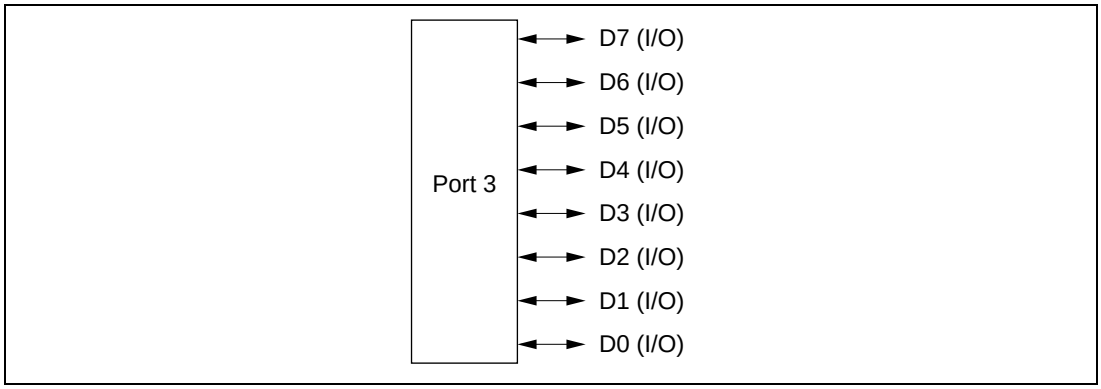


Figure 8.10 Port 3 Pin Functions (Modes 1, 2, and 3 (EXPE = 1))

Modes 2 and 3 (EXPE = 0): In modes 2 and 3 (when EXPE = 0), port 3 functions as an I/O port, and input or output can be specified on a bit-by-bit basis. When a bit in P3DDR is set to 1, the corresponding pin functions as an output port, and when cleared to 0, as an input port.

The port 3 pin functions are shown in figure 8.11.

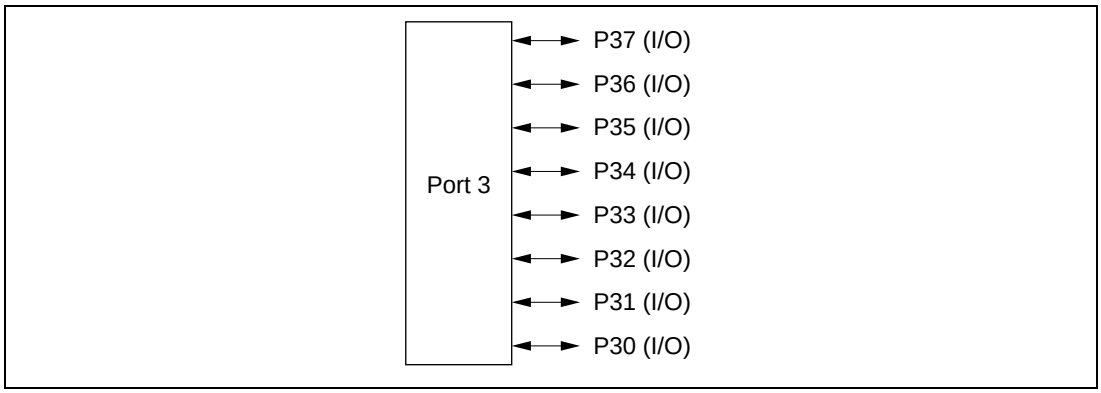


Figure 8.11 Port 3 Pin Functions (Modes 2 and 3 (EXPE = 0))

8.4.4 MOS Input Pull-Up Function

Port 3 has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 2 and 3 (when EXPE = 0), and can be specified as on or off on a bit-by-bit basis.

When a P3DDR bit is cleared to 0 in mode 2 or 3 (when EXPE = 0), setting the corresponding P3PCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset and in hardware standby mode. The prior state is retained in software standby mode.

Table 8.8 summarizes the MOS input pull-up states.

Table 8.8 MOS Input Pull-Up States (Port 3)

Mode	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
1, 2, 3 (EXPE = 1)	Off	Off	Off	Off
2, 3 (EXPE = 0)	Off	Off	On/Off	On/Off

Legend:

Off: MOS input pull-up is always off.

On/Off: On when P3DDR = 0 and P3PCR = 1; otherwise off.

8.5 Port 4

8.5.1 Overview

Port 4 is an 8-bit I/O port. Port 4 pins also function as the $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ2}}$ input pins, A/D converter external trigger input pin ($\overline{\text{ADTRG}}$), IIC0 I/O pin (SDA0) (option in H8S/2128 Series only), subclock input pin (EXCL), bus control signal I/O pins ($\overline{\text{AS}}$/$\overline{\text{IOS}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$), and system clock (ø) output pin. In the H8S/2128 Series, P47 is an NMOS push-pull output. SDA0 is an NMOS open-drain output, and has direct bus drive capability.

Figure 8.12 shows the port 4 pin configuration.

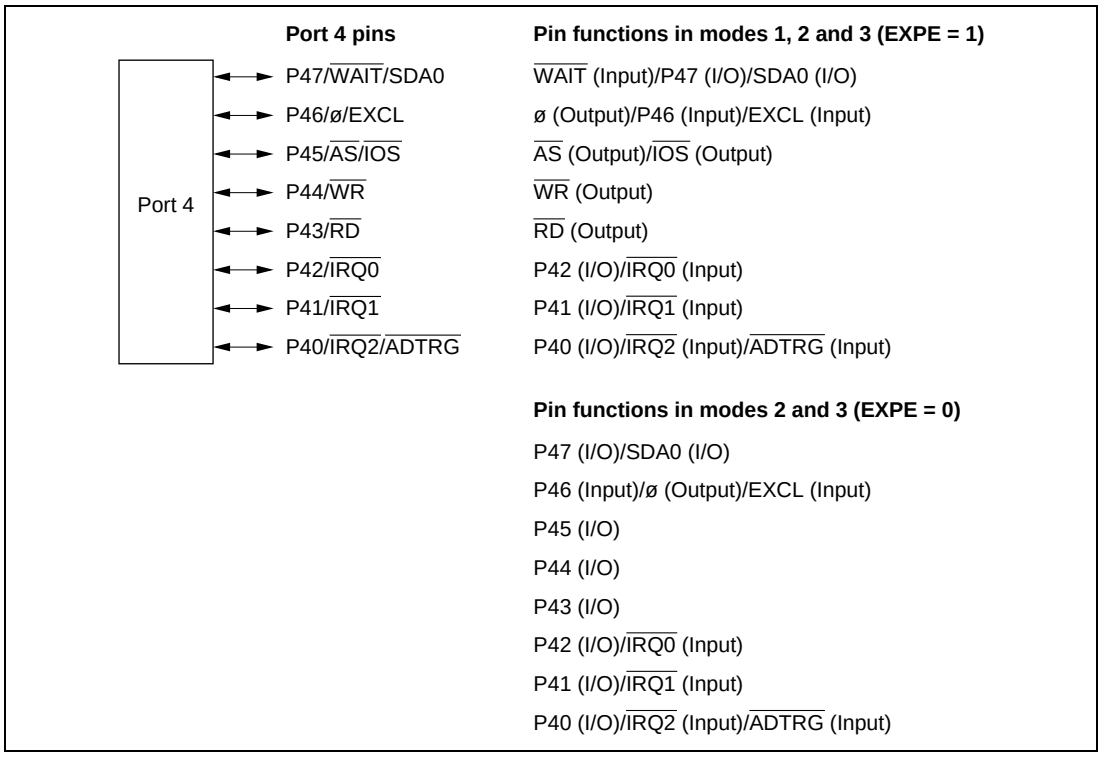


Figure 8.12 Port 4 Pin Functions

8.5.2 Register Configuration

Table 8.9 summarizes the port 4 registers.

Table 8.9 Port 4 Registers

Name	Abbreviation	R/W	Initial Value	Address[*1]
Port 4 data direction register	P4DDR	W	H'40/H'00*[2]	H'FFB5
Port 4 data register	P4DR	R/W	H'00	H'FFB7

Notes: 1. Lower 16 bits of the address.
2. Initial value depends on the mode.

Port 4 Data Direction Register (P4DDR)

Bit	7	6	5	4	3	2	1	0
	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR
Mode 1								
Initial value	0	1	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Modes 2 and 3								
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

P4DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 4. P4DDR cannot be read; if it is, an undefined value will be returned.

P4DDR is initialized to H'40 (mode 1) or H'00 (modes 2 and 3) by a reset and in hardware standby mode. It retains its prior state in software standby mode.

- Modes 1, 2, and 3 (EXPE = 1)

  Pin P47 functions as a bus control input ($\overline{\text{WAIT}}$), IIC0 I/O pin (SDA0), or I/O port, according to the wait mode setting. When P47 functions as an I/O port, it becomes an output port when P47DDR is set to 1, and an input port when P47DDR is cleared to 0.

  Pin P46 functions as the ø output pin when P46DDR is set to 1, and as the subclock input (EXCL) or an input port when P46DDR is cleared to 0.

  Pins P45 to P43 automatically become bus control outputs ($\overline{\text{AS}}$/$\overline{\text{IOS}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$), regardless of the input/output direction indicated by P45DDR to P43DDR.

  Pins P42 to P40 become output ports when P42DDR to P40DDR are set to 1, and input ports when P42DDR to P40DDR are cleared to 0.
- Modes 2 and 3 (EXPE = 0)

  When the corresponding P4DDR bits are set to 1, pin P46 functions as the ø output pin and pins P47 and P45 to P40 become output ports. When P4DDR bits are cleared to 0, the corresponding pins become input ports.

Port 4 Data Register (P4DR)

Bit	7	6	5	4	3	2	1	0
	P47DR	P46DR	P45DR	P44DR	P43DR	P42DR	P41DR	P40DR
Initial value	0	—*	0	0	0	0	0	0
Read/Write	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the state of pin P46.

P4DR is an 8-bit readable/writable register that stores output data for the port 4 pins (P47 to P40). With the exception of P46, if a port 4 read is performed while P4DDR bits are set to 1, the P4DR values are read directly, regardless of the actual pin states. If a port 4 read is performed while P4DDR bits are cleared to 0, the pin states are read.

P4DR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

8.5.3 Pin Functions

Port 4 pins also function as the $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ2}}$ input pins, A/D converter input pin ($\overline{\text{ADTRG}}$), IIC0 I/O pin (SDA0), subclock input pin (EXCL), bus control signal I/O pins ($\overline{\text{AS}}$/$\overline{\text{IOS}}$, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WAIT}}$), and system clock (ø) output pin. The pin functions differ between the mode 1, 2, and 3

(EXPE = 1) expanded modes and the mode 2 and 3 (EXPE = 0) single-chip modes. The port 4 pin functions are shown in table 8.10.

Table 8.10 Port 4 Pin Functions

Pin | **Selection Method and Pin Functions**

P47/$\overline{\text{WAIT}}$/SDA0

The pin function is switched as shown below according to the combination of operating mode, bit WMS1 in WSCR, bit ICE in ICCR of IIC0, and bit P47DDR.

Operating mode	Modes 1, 2, 3 (EXPE = 1)				Modes 2, 3 (EXPE = 0)		
WMS1	0			1	—		
ICE	0		1	—	0		1
P47DDR	0	1	—	—	0	1	—
Pin function	P47 input pin	P47 output pin	SDA0 I/O pin	$\overline{\text{WAIT}}$ input pin	P47 input pin	P47 output pin	SDA0 I/O pin

In the H8S/2128 Series, when this pin is set as the P47 output pin, it is an NMOS push-pull output. SDA0 is an NMOS open-drain output, and has direct bus drive capability.

P46/ø/EXCL

The pin function is switched as shown below according to the combination of bit EXCLE in LPWRCR and bit P46DDR.

P46DDR	0		1
EXCLE	0	1	0
Pin function	P46 input pin	EXCL input pin	ø output pin

When this pin is used as the EXCL input pin, P46DDR should be cleared to 0.

P45/$\overline{\text{AS}}$/$\overline{\text{IOS}}$

The pin function is switched as shown below according to the combination of operating mode, bits IOSE in SYSCR, and bit P45DDR.

Operating mode	Modes 1, 2, 3 (EXPE = 1)		Modes 2, 3 (EXPE = 0)	
P45DDR	—		0	1
IOSE	0	1	—	—
Pin function	$\overline{\text{AS}}$ output pin	$\overline{\text{IOS}}$ output pin	P45 input pin	P45 output pin

P44/$\overline{\text{WR}}$

The pin function is switched as shown below according to the combination of operating mode, and bit P44DDR.

Operating mode	Modes 1, 2, 3 (EXPE = 1)	Modes 2, 3 (EXPE = 0)	
P44DDR	—	0	1
Pin function	$\overline{\text{WR}}$ output pin	P44 input pin	P44 output pin

Table 8.10 Port 4 Pin Functions (cont)

Pin	Selection Method and Pin Functions
P43/$\overline{\text{RD}}$/$\overline{\text{IOR}}$	The pin function is switched as shown below according to the combination of operating mode, and bit P43DDR.

Operating mode	Modes 1, 2, 3 (EXPE = 1)	Modes 2, 3 (EXPE = 0)	
P43DDR	—	0	1
Pin function	$\overline{\text{RD}}$ output pin	P43 input pin	P43 output pin

P42/$\overline{\text{IRQ0}}$

P42DDR	0	1
Pin function	P42 input pin	P42 output pin
	$\overline{\text{IRQ0}}$ input pin	

When bit IRQ0E in IER is set to 1, this pin is used as the $\overline{\text{IRQ0}}$ input pin.

P41/$\overline{\text{IRQ1}}$

P41DDR	0	1
Pin function	P41 input pin	P41 output pin
	$\overline{\text{IRQ1}}$ input pin	

When bit IRQ1E in IER is set to 1, this pin is used as the $\overline{\text{IRQ1}}$ input pin.

P40/$\overline{\text{IRQ2}}$/$\overline{\text{ADTRG}}$

P40DDR	0	1
Pin function	P40 input pin	P40 output pin
	$\overline{\text{IRQ2}}$ input pin, $\overline{\text{ADTRG}}$ input pin	

When the IRQ2E bit in IER is set to 1, this pin is used as the $\overline{\text{IRQ2}}$ input pin.

When TRGS1 and TRGS0 bit in ADCR of the A/D converter are both set to 1, this pin is used as the $\overline{\text{ADTRG}}$ input pin.

8.6 Port 5

8.6.1 Overview

Port 5 is a 3-bit I/O port. Port 5 pins also function as SCI0 I/O pins (TxD0, RxD0, SCK0), and the IIC0 I/O pin (SCL0) (option in H8S/2128 Series only). In the H8S/2128 Series, P52 and SCK0 are NMOS push-pull outputs, and SCL0 is an NMOS open-drain output. Port 5 pin functions are the same in all operating modes.

Figure 8.13 shows the port 5 pin configuration.

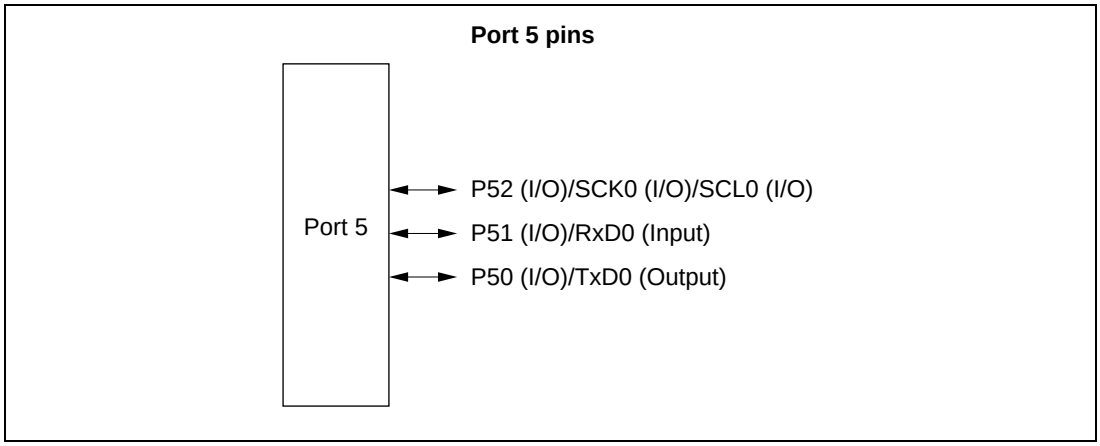


Figure 8.13 Port 5 Pin Functions

8.6.2 Register Configuration

Table 8.11 shows the port 5 register configuration.

Table 8.11 Port 5 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 5 data direction register	P5DDR	W	H'F8	H'FFB8
Port 5 data register	P5DR	R/W	H'F8	H'FFBA

Note: *Lower 16 bits of the address.

Port 5 Data Direction Register (P5DDR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	P52DDR	P51DDR	P50DDR
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	W	W	W

P5DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 5. P5DDR cannot be read; if it is, an undefined value will be returned. Bits 7 to 3 are reserved.

Setting a P5DDR bit to 1 makes the corresponding port 5 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P5DDR is initialized to H'F8 by a reset and in hardware standby mode. It retains its prior state in software standby mode. As SCI0 is initialized, the pin states are determined by the IIC0 ICCR, P5DDR, and P5DR specifications.

Port 5 Data Register (P5DR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	P52DR	P51DR	P50DR
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

P5DR is an 8-bit readable/writable register that stores output data for the port 5 pins (P52 to P50). If a port 5 read is performed while P5DDR bits are set to 1, the P5DR values are read directly, regardless of the actual pin states. If a port 5 read is performed while P5DDR bits are cleared to 0, the pin states are read.

Bits 7 to 3 are reserved; they cannot be modified and are always read as 1.

P5DR is initialized to H'F8 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

8.6.3 Pin Functions

Port 5 pins also function as SCI0 I/O pins (TxD0, RxD0, SCK0) and the IIC0 I/O pin (SCL0). The port 5 pin functions are shown in table 8.12.

Table 8.12 Port 5 Pin Functions

Pin — **Selection Method and Pin Functions**

P52/SCK0/SCL0

The pin function is switched as shown below according to the combination of bits CKE1 and CKE0 in SCR, bit C/$\overline{A}$ in SMR of SCI0, bit ICE in ICCR of IIC0, and bit P52DDR.

ICE	0					1
CKE1	0				1	0
C/$\overline{A}$	0			1	—	0
CKE0	0		1	—	—	0
P52DDR	0	1	—	—	—	—
Pin function	P52 input pin	P52 output pin	SCK0 output pin	SCK0 output pin	SCK0 input pin	SCL0 I/O pin

When this pin is used as the SCL0 I/O pin, bits CKE1 and CKE0 in SCR of SCI0 and bit C/$\overline{A}$ in SMR of SCI0 must all be cleared to 0.

SCL0 is an NMOS open-drain output, and has direct bus drive capability. In the H8S/2128 Series, when set as the P52 output pin or SCK0 output pin, this pin is an NMOS push-pull output.

P51/RxD0

The pin function is switched as shown below according to the combination of bit RE in SCR of SCI0 and bit P51DDR.

RE	0		1
P51DDR	0	1	—
Pin function	P51 input pin	P51 output pin	RxD input pin

P50/TxD0

The pin function is switched as shown below according to the combination of bit TE in SCR of SCI0 and bit P50DDR.

TE	0		1
P50DDR	0	1	—
Pin function	P50 input pin	P50 output pin	TxD0 output pin

8.7 Port 6

8.7.1 Overview

Port 6 is an 8-bit I/O port. Port 6 pins also function as the 16-bit free-running timer (FRT) I/O pins (FTOA, FTOB, FTIA to FTID, FTCI), timer 0 and 1 (TMR0, TMR1) I/O pins (TMCI0, TMRI0, TMO0, TMCI1, TMRI1, TMO1), timer X (TMRX) I/O pins (TMOX, TMIX) (H8S/2128 Series only), the timer Y (TMRY) input pin (TMIY), timer connection I/O pins (CSYNCI, HSYNCI, HSYNCO, HFBACKI, VSYNCI, VSYNCO, VFBACKI, CLAMPO) (H8S/2128 Series only), and expansion A/D converter input pins (CIN7 to CIN0). Port 6 pin functions are the same in all operating modes.

Figure 8.14 shows the port 6 pin configuration.

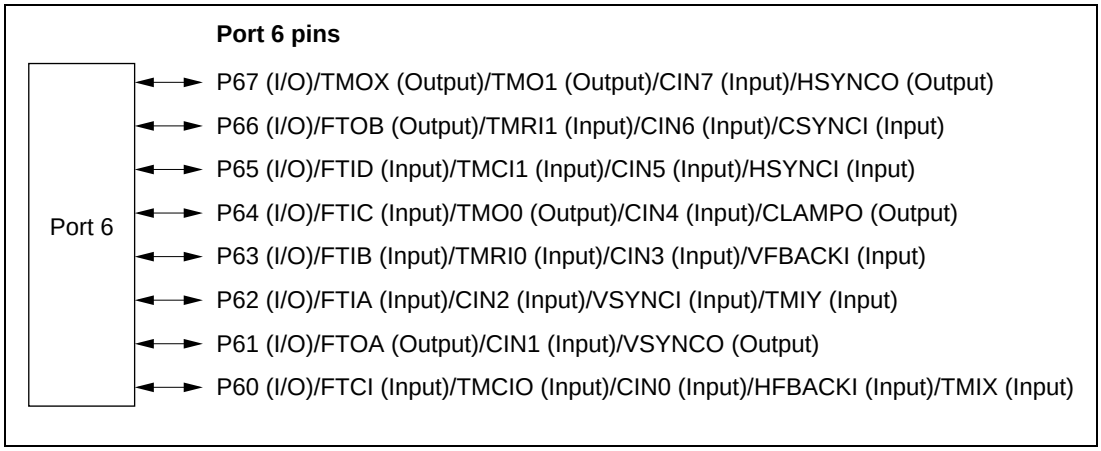


Figure 8.14 Port 6 Pin Functions

8.7.2 Register Configuration

Table 8.13 shows the port 6 register configuration.

Table 8.13 Port 6 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 6 data direction register	P6DDR	W	H'00	H'FFB9
Port 6 data register	P6DR	R/W	H'00	H'FFBB

Note: *Lower 16 bits of the address.

Port 6 Data Direction Register (P6DDR)

Bit	7	6	5	4	3	2	1	0
	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

P6DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 6. P6DDR cannot be read; if it is, an undefined value will be returned.

Setting a P6DDR bit to 1 makes the corresponding port 6 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P6DDR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

Port 6 Data Register (P6DR)

Bit	7	6	5	4	3	2	1	0
	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P6DR is an 8-bit readable/writable register that stores output data for the port 6 pins (P67 to P60). If a port 6 read is performed while P6DDR bits are set to 1, the P6DR values are read directly, regardless of the actual pin states. If a port 6 read is performed while P6DDR bits are cleared to 0, the pin states are read.

P6DR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

8.7.3 Pin Functions

Port 6 pins also function as the 16-bit free-running timer (FRT) I/O pins (FTOA, FTOB, FTIA to FTID, FTCI), timer 0 and 1 (TMR0, TMR1) I/O pins (TMCI0, TMRI0, TMO0, TMCI1, TMRI1, TMO1), timer X (TMRX) I/O pins (TMOX, TMIX), the timer Y (TMRY) input pin (TMIY), timer connection I/O pins (CSYNCI, HSYNCI, HSYNCO, HFBACKI, VSYNCI, VSYNCO, VFBACKI, CLAMPO), and expansion A/D converter input pins (CIN7 to CIN0). The port 6 pin functions are shown in table 8.14.

Table 8.14 Port 6 Pin Functions

Pin	Selection Method and Pin Functions
P67/TMO1/TMOX/ CIN7/HSYNCO	The pin function is switched as shown below according to the combination of bits OS3 to OS0 in TCSR of TMR1 and TMRX, bit HOE in TCONRO of the timer connection function, and bit P67DDR.

HOE	0				1
TMRX: OS3 to 0	All 0			Not all 0	—
TMR1: OS3 to 0	All 0		Not all 0	—	—
P67DDR	0	1	—	—	—
Pin function	P67 input pin	P67 output pin	TMO1 output pin	TMOX output pin	HSYNCO output pin
	CIN7 input pin				

It can always be used as the CIN7 input pin.

P66/FTOB/TMRI1/ CIN6/CSYNCI — The pin function is switched as shown below according to the combination of bit OEB in TOCR of the FRT and bit P66DDR.

OEB	0		1
P66DDR	0	1	—
Pin function	P66 input pin	P66 output pin	FTOB output pin
	TMRI1 input pin, CSYNCI input pin, CIN6 input pin		

This pin is used as the TMRI1 input pin when bits CCLR1 and CCLR0 are both set to 1 in TCR of TMR1.

It can always be used as the CSYNCI or CIN6 input pin.

Table 8.14 Port 6 Pin Functions (cont)

Pin	Selection Method and Pin Functions

P65/FTID/TMCI1/CIN5/HSYNCI

P65DDR	0	1
Pin function	P65 input pin	P65 output pin
	FTID input pin, TMCI1 input pin, HSYNCI input pin, CIN5 input pin	

This pin is used as the TMCI1 input pin when an external clock is selected with bits CKS2 to CKS0 in TCR of TMR1.

It can always be used as the FTID, HSYNCI or CIN5 input pin.

P64/FTIC/TMO0/CIN4/CLAMPO

The pin function is switched as shown below according to the combination of bits OS3 to OS0 in TCSR of TMR0, bit CLOE in TCONRO of the timer connection function, and bit P64DDR.

CLOE	0			1
OS3 to 0	All 0		Not all 0	—
P64DDR	0	1	—	—
Pin function	P64 input pin	P64 output pin	TMO0 output pin	CLAMPO output pin
	FTIC input pin, CIN4 input pin			

This pin can always be used as the FTIC or CIN4 input pin.

P63/FTIB/TMRI0/CIN3/VFBACKI

P63DDR	0	1
Pin function	P63 input pin	P63 output pin
	FTIB input pin, TMRI0 input pin, VFBACKI input pin, CIN3 input pin	

This pin is used as the TMRI0 input pin when bits CCLR1 and CCLR0 are both set to 1 in TCR of TMR0.

It can always be used as the FTIB, VFBACKI or CIN3 input pin.

P62/FTIA/CIN2/VSYNCI/TMIY

P62DDR	0	1
Pin function	P62 input pin	P62 output pin
	FTIA input pin, VSYNCI input pin, TMIY input pin, CIN2 input pin	

This pin can always be used as the FTIA, TMIY, VSYNCI or CIN2 input pin.

Table 8.14 Port 6 Pin Functions (cont)

Pin	Selection Method and Pin Functions
P61/FTOA/CIN1/ VSYNCO	The pin function is switched as shown below according to the combination of bit OEA in TOCR of the FRT, bit VOE in TCONRO of the timer connection function, and bit P61DDR.

VOE	0			1
OEA	0		1	0
P61DDR	0	1	—	—
Pin function	P61 input pin	P61 output pin	FTOA0 output pin	VSYNCO output pin
	CIN1 input pin			

When this pin is used as the VSYNCO pin, the OEA bit in TOCR of the FRT must be cleared. This pin can always be used as the CIN1 pin.

P60/FTCI/TMCI0/ CIN0/HFBACKI/ TMIX

P60DDR	0	1
Pin function	P60 I/O pin	P60 output pin
	FTCI input pin, TMCI0 input pin, HFBACKI input pin, CIN0 input pin, TMIX input pin	

This pin is used as the FTCI input pin when an external clock is selected with bits CKS1 and CKS0 in TCR of the FRT.

It is used as the TMCI0 input pin when an external clock is selected with bits CKS2 to CKS0 in TCR of TMR0.

It can always be used as the TMIX, HFBACKI, CIN0 input pin.

8.8 Port 7

8.8.1 Overview

Port 7 is an 8-bit input port. Port 7 pins also function as the A/D converter analog input pins (AN0 to AN7). Port 7 functions are the same in all operating modes.

Figure 8.15 shows the port 7 pin configuration.

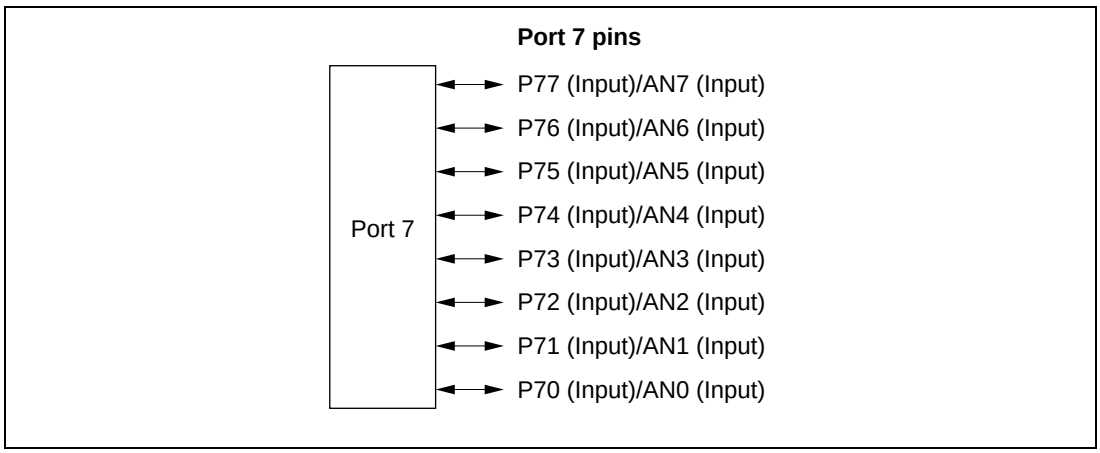


Figure 8.15 Port 7 Pin Functions

8.8.2 Register Configuration

Table 8.16 shows the port 7 register configuration. Port 7 is an input-only port, and does not have a data direction register or data register.

Table 8.16 Port 7 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 7 input data register	P7PIN	R	Undefined	H'FFBE

Note: *Lower 16 bits of the address.

Port 7 Input Data Register (P7PIN)

Bit	7	6	5	4	3	2	1	0
	P77PIN	P76PIN	P75PIN	P74PIN	P73PIN	P72PIN	P71PIN	P70PIN
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Note: * Determined by the state of pins P77 to P70.

When a P7PIN read is performed, the pin states are always read.

8.8.3 Pin Functions

Port 7 pins also function as the A/D converter analog input pins (AN0 to AN7).

Section 9 8-Bit PWM Timers [H8S/2128 Series]

9.1 Overview

The H8/2128 Series has an on-chip pulse width modulation (PWM) timer module with sixteen outputs. Sixteen output waveforms are generated from a common time base, enabling PWM output with a high carrier frequency to be produced using pulse division. The PWM timer module has sixteen 8-bit PWM data registers (PWDRs), and an output pulse with a duty cycle of 0 to 100% can be obtained as specified by PWDR and the port data register (P1DR or P2DR).

9.1.1 Features

The PWM timer module has the following features.

- Operable at a maximum carrier frequency of 1.25 MHz using pulse division (at 20 MHz operation)
- Duty cycles from 0 to 100% with 1/256 resolution (100% duty realized by port output)
- Direct or inverted PWM output, and PWM output enable/disable control

9.1.2 Block Diagram

Figure 9.1 shows a block diagram of the PWM timer module.

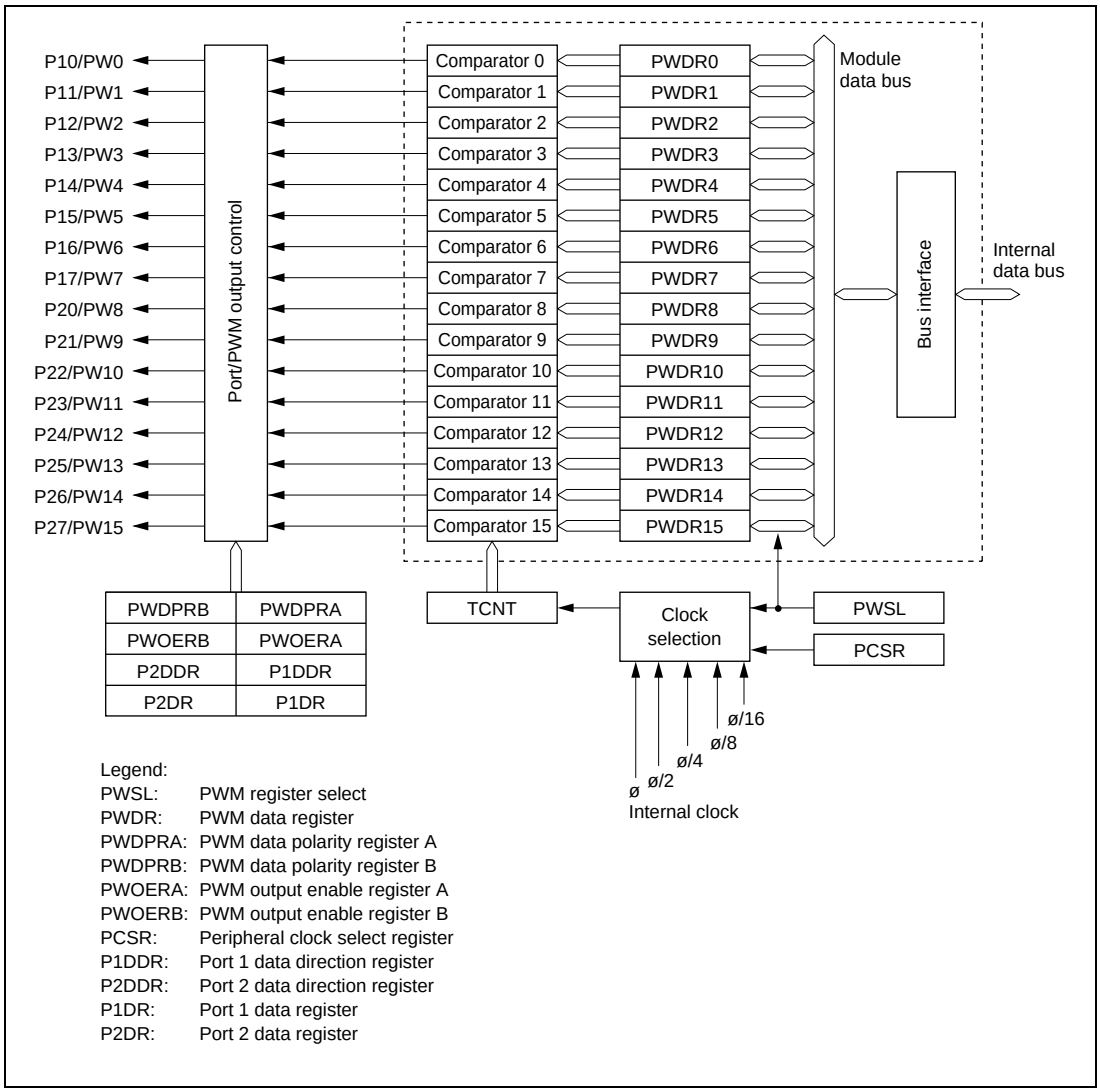


Figure 9.1 Block Diagram of PWM Timer Module

9.1.3 Pin Configuration

Table 9.1 shows the PWM output pin.

Table 9.1 Pin Configuration

Name	Abbreviation	I/O	Function
PWM output pin 0 to 15	PW0 to PW15	Output	PWM timer pulse output 0 to 15

9.1.4 Register Configuration

Table 9.2 lists the registers of the PWM timer module.

Table 9.2 PWM Timer Module Registers

Name	Abbreviation	R/W	Initial Value	Address*
PWM register select	PWSL	R/W	H'20	H'FFD6
PWM data registers 0 to 15	PWDR0 to PWDR15	R/W	H'00	H'FFD7
PWM data polarity register A	PWDPRA	R/W	H'00	H'FFD5
PWM data polarity register B	PWDPRB	R/W	H'00	H'FFD4
PWM output enable register A	PWOERA	R/W	H'00	H'FFD3
PWM output enable register B	PWOERB	R/W	H'00	H'FFD2
Port 1 data direction register	P1DDR	W	H'00	H'FFB0
Port 2 data direction register	P2DDR	W	H'00	H'FFB1
Port 1 data register	P1DR	R/W	H'00	H'FFB2
Port 2 data register	P2DR	R/W	H'00	H'FFB3
Peripheral clock select register	PCSR	R/W	H'00	H'FF82
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

Note: * Lower 16 bits of the address.

9.2 Register Descriptions

9.2.1 PWM Register Select (PWSL)

Bit	7	6	5	4	3	2	1	0
	PWCKE	PWCKS	—	—	RS3	RS2	RS1	RS0
Initial value	0	0	1	0	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

PWSL is an 8-bit readable/writable register used to select the PWM timer input clock and the PWM data register.

PWSL is initialized to H'20 by a reset, and in the standby modes, watch mode, subactive mode, subsleep mode, and module stop mode.

Bits 7 and 6—PWM Clock Enable, PWM Clock Select (PWCKE, PWCKS): These bits, together with bits PWCKA and PWCKB in PCSR, select the internal clock input to TCNT in the PWM timer.

PWSL		PCSR			
Bit 7	**Bit 6**	**Bit 2**	**Bit 1**		
PWCKE	**PWCKS**	**PWCKB**	**PWCKA**	**Description**	
0	—	—	—	Clock input is disabled	(Initial value)
1	0	—	—	ø (system clock) is selected	
	1	0	0	ø/2 is selected	
			1	ø/4 is selected	
		1	0	ø/8 is selected	
			1	ø/16 is selected	

The PWM resolution, PWM conversion period, and carrier frequency depend on the selected internal clock, and can be found from the following equations.

Resolution (minimum pulse width) = 1/internal clock frequency
PWM conversion period = resolution × 256
Carrier frequency = 16/PWM conversion period

Thus, with a 20 MHz system clock (ø), the resolution, PWM conversion period, and carrier frequency are as shown below.

Table 9.3 Resolution, PWM Conversion Period, and Carrier Frequency when ø = 20 MHz

Internal Clock Frequency	Resolution	PWM Conversion Period	Carrier Frequency
ø	50 ns	12.8 µs	1250 kHz
ø/2	100 ns	25.6 µs	625 kHz
ø/4	200 ns	51.2 µs	312.5 kHz
ø/8	400 ns	102.4 µs	156.3 kHz
ø/16	800 ns	204.8 µs	78.1 kHz

Bit 5—Reserved: This bit is always read as 1 and cannot be modified.

Bit 4—Reserved: This bit is always read as 0 and cannot be modified.

Bits 3 to 0—Register Select (RS3 to RS0): These bits select the PWM data register.

Bit 3	Bit 2	Bit 1	Bit 0	
RS3	RS2	RS1	RS0	Register Selection
0	0	0	0	PWDR0 selected
			1	PWDR1 selected
		1	0	PWDR2 selected
			1	PWDR3 selected
	1	0	0	PWDR4 selected
			1	PWDR5 selected
		1	0	PWDR6 selected
			1	PWDR7 selected
1	0	0	0	PWDR8 selected
			1	PWDR9 selected
		1	0	PWDR10 selected
			1	PWDR11 selected
	1	0	0	PWDR12 selected
			1	PWDR13 selected
		1	0	PWDR14 selected
			1	PWDR15 selected

9.2.2 PWM Data Registers (PWDR0 to PWDR15)

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Each PWDR is an 8-bit readable/writable register that specifies the duty cycle of the basic pulse to be output, and the number of additional pulses. The value set in PWDR corresponds to a 0 or 1 ratio in the conversion period. The upper 4 bits specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16. The lower 4 bits specify how many extra pulses are to be added within the conversion period comprising 16 basic pulses. Thus, a specification of 0/256 to 255/256 is possible for 0/1 ratios within the conversion period. For 256/256 (100%) output, port output should be used.

PWDR is initialized to H'00 by a reset, and in the standby modes, watch mode, subactive mode, subsleep mode, and module stop mode.

9.2.3 PWM Data Polarity Registers A and B (PWDPRA and PWDPRB)

PWDPRA

Bit	7	6	5	4	3	2	1	0
	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWDPRB

Bit	7	6	5	4	3	2	1	0
	OS15	OS14	OS13	OS12	OS11	OS10	OS9	OS8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Each PWDPR is an 8-bit readable/writable register that controls the polarity of the PWM output. Bits OS0 to OS15 correspond to outputs PW0 to PW15.

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

OS	Description
0	PWM direct output (PWDR value corresponds to high width of output) (Initial value)
1	PWM inverted output (PWDR value corresponds to low width of output)

9.2.4 PWM Output Enable Registers A and B (PWOERA and PWOERB)

PWOERA

Bit	7	6	5	4	3	2	1	0
	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWOERB

Bit	7	6	5	4	3	2	1	0
	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Each PWOER is an 8-bit readable/writable register that switches between PWM output and port output. Bits OE15 to OE0 correspond to outputs PW15 to PW0. To set a pin in the output state, a setting in the port direction register is also necessary. Bits P17DDR to P10DDR correspond to outputs PW7 to PW0, and bits P27DDR to P20DDR correspond to outputs PW15 to PW8.

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

DDR	OE	Description
0	0	Port input (Initial value)
	1	Port input
1	0	Port output or PWM 256/256 output
	1	PWM output (0 to 255/256 output)

9.2.5 Peripheral Clock Select Register (PCSR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	PWCKB	PWCKA	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	—

PCSR is an 8-bit readable/writable register that selects the PWM timer input clock.

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

Bits 7 to 3—Reserved: These bits cannot be modified and are always read as 0.

Bits 2 and 1—PWM Clock Select (PWCKB, PWCKA): Together with bits PWCKE and PWCKS in PWSL, these bits select the internal clock input to TCNT in the PWM timer. For details, see section 9.2.1, PWM Register Select (PWSL).

Bit 0—Reserved: Do not set this bit to 1.

9.2.6 Port 1 Data Direction Register (P1DDR)

Bit	7	6	5	4	3	2	1	0
	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register that specifies the input/output direction and PWM output for each pin of port 1 on a bit-by-bit basis.

Port 1 pins are multiplexed with pins PW0 to PW7. The bit corresponding to a pin to be used for PWM output should be set to 1.

For details on P1DDR, see section 8.2, Port 1.

9.2.7 Port 2 Data Direction Register (P2DDR)

Bit	7	6	5	4	3	2	1	0
	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

P2DDR is an 8-bit write-only register that specifies the input/output direction and PWM output for each pin of port J on a bit-by-bit basis.

Port 2 pins are multiplexed with pins PW8 to PW15. The bit corresponding to a pin to be used for PWM output should be set to 1.

For details on P2DDR, see section 8.3, Port 2.

9.2.8 Port 1 Data Register (P1DR)

Bit	7	6	5	4	3	2	1	0
	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1DR is an 8-bit readable/writable register used to fix PWM output at 1 (when OS = 0) or 0 (when OS = 1).

For details on P1DR, see section 8.2, Port 1.

9.2.9 Port 2 Data Register (P2DR)

Bit	7	6	5	4	3	2	1	0
	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2DR is an 8-bit readable/writable register used to fix PWM output at 1 (when OS = 0) or 0 (when OS = 1).

For details on P2DR, see section 8.3, Port 2.

9.2.10 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	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	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers, and is used to perform module stop mode control.

When the MSTP11 bit is set to 1, 8-bit PWM timer operation is halted and a transition is made to module stop mode. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 3—Module Stop (MSTP11): Specifies PWM module stop mode.

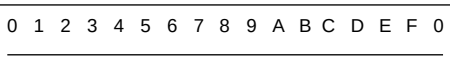
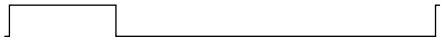
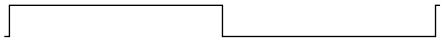
MSTPCRH Bit 3		
MSTP11	**Description**	
0	PWM module stop mode is cleared	
1	PWM module stop mode is set	(Initial value)

9.3 Operation

9.3.1 Correspondence between PWM Data Register Contents and Output Waveform

The upper 4 bits of PWDR specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16, as shown in table 9.4.

Table 9.4 Duty Cycle of Basic Pulse

Upper 6 Bits	Basic Pulse Waveform (Internal)
000000	
000001	
000010	
000011	
000100	
000101	
000110	
000111	
⋮	
111000	
111001	
111010	
111011	
111100	
111101	
111110	
111111	

The lower 4 bits of PWDR specify the position of pulses added to the 16 basic pulses, as shown in table 9.5. An additional pulse consists of a high period (when OS = 0) with a width equal to the resolution, added before the rising edge of a basic pulse. When the upper 4 bits of PWDR are 0000, there is no rising edge of the basic pulse, but the timing for adding pulses is the same.

Table 9.5 Position of Pulses Added to Basic Pulses

	Basic Pulse No.															
Lower 4 Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0000																
0001																Yes
0010								Yes								Yes
0011								Yes				Yes				Yes
0100				Yes				Yes				Yes				Yes
0101				Yes				Yes				Yes		Yes		Yes
0110				Yes		Yes		Yes				Yes		Yes		Yes
0111				Yes		Yes		Yes		Yes		Yes		Yes		Yes
1000		Yes		Yes		Yes		Yes		Yes		Yes		Yes		Yes
1001		Yes		Yes		Yes		Yes		Yes		Yes		Yes	Yes	Yes
1010		Yes		Yes		Yes	Yes	Yes		Yes		Yes		Yes	Yes	Yes
1011		Yes		Yes		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes
1100		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes
1101		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
1110		Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
1111		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

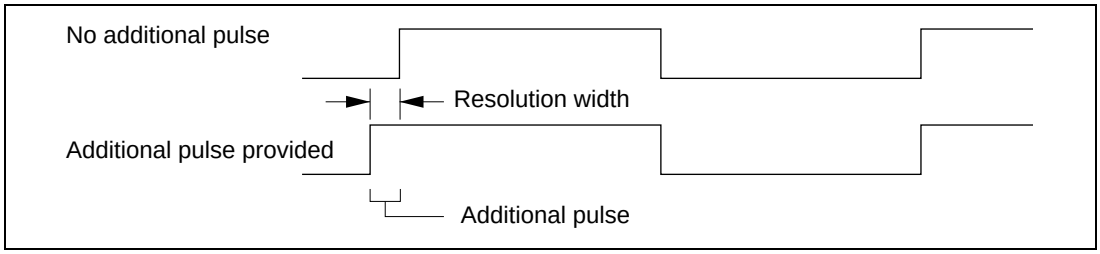


Figure 9.2 Example of Additional Pulse Timing (When Upper 4 Bits of PWDR = 1000)

Section 10 14-Bit PWM D/A

10.1 Overview

The H8S/2128 Series and H8S/2124 Series have an on-chip 14-bit pulse-width modulator (PWM) with two output channels.

Each channel can be connected to an external low-pass filter to operate as a 14-bit D/A converter.

Both channels share the same counter (DACNT) and control register (DACR).

10.1.1 Features

The features of the 14-bit PWM D/A are listed below.

- The pulse is subdivided into multiple base cycles to reduce ripple.
- Two resolution settings and two base cycle settings are available

  The resolution can be set equal to one or two system clock cycles. The base cycle can be set equal to $T \times 64$ or $T \times 256$, where T is the resolution.
- Four operating rates

  The two resolution settings and two base cycle settings combine to give a selection of four operating rates.

10.1.2 Block Diagram

Figure 10.1 shows a block diagram of the PWM D/A module.

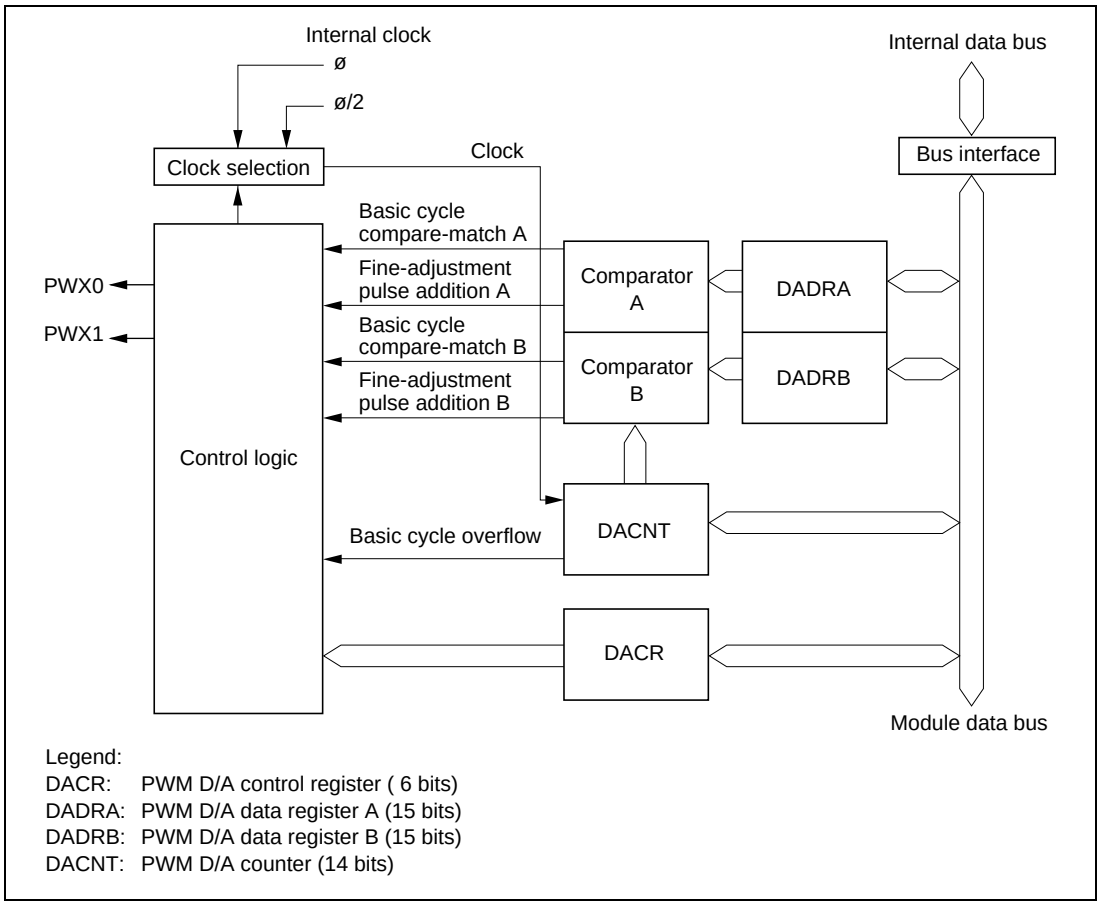


Figure 10.1 PWM D/A Block Diagram

10.1.3 Pin Configuration

Table 10.1 lists the pins used by the PWM D/A module.

Table 10.1 Input and Output Pins

Channel	Name	Abbr.	I/O	Function
A	PWM output pin 0	PWX0	Output	PWM output, channel A
B	PWM output pin 1	PWX1	Output	PWM output, channel B

10.1.4 Register Configuration

Table 10.2 lists the registers of the PWM D/A module.

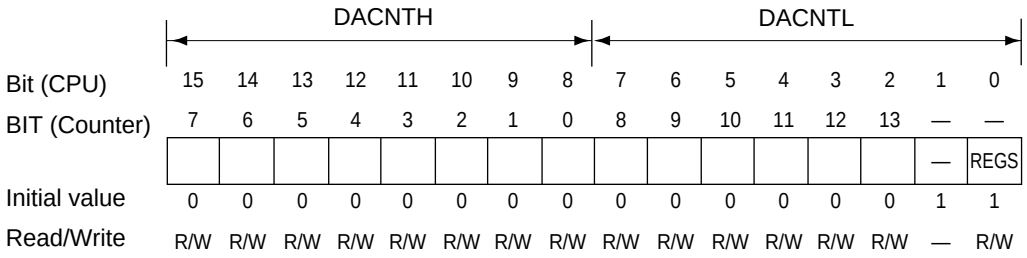
Table 10.2 Register Configuration

Name	Abbreviation	R/W	Initial value	Address*[1]
PWM D/A control register	DACR	R/W	H'30	H'FFA0*[2]
PWM D/A data register A high	DADRAH	R/W	H'FF	H'FFA0*[2]
PWM D/A data register A low	DADRAL	R/W	H'FF	H'FFA1*[2]
PWM D/A data register B high	DADRBH	R/W	H'FF	H'FFA6*[2]
PWM D/A data register B low	DADRBL	R/W	H'FF	H'FFA7*[2]
PWM D/A counter high	DACNTH	R/W	H'00	H'FFA6*[2]
PWM D/A counter low	DACNTL	R/W	H'03	H'FFA7*[2]
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

Notes: 1. Lower 16 bits of the address.
2. The same addresses are shared by DADRAH and DACR, and by DADRB and DACNT. Switching is performed by the REGS bit in DACNT or DADRB.

10.2 Register Descriptions

10.2.1 PWM D/A Counter (DACNT)



DACNT is a 14-bit readable/writable up-counter that increments on an input clock pulse. The input clock is selected by the clock select bit (CKS) in DACR. The CPU can read and write the DACNT value, but since DACNT is a 16-bit register, data transfers between it and the CPU are performed using a temporary register (TEMP). See section 10.3, Bus Master Interface, for details.

DACNT functions as the time base for both PWM D/A channels. When a channel operates with 14-bit precision, it uses all DACNT bits. When a channel operates with 12-bit precision, it uses the lower 12 (counter) bits and ignores the upper two (counter) bits.

DACNT is initialized to H'0003 by a reset, in the standby modes, watch mode, subactive mode, subsleep mode, and module stop mode, and by the PWME bit.

Bit 1 of DACNTL (CPU) is not used, and is always read as 1.

DACNTL Bit 0—Register Select (REGS): DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed. The REGS bit can be accessed regardless of whether DADRB or DACNT is selected.

Bit 0

REGS	Description	
0	DADRA and DADRB can be accessed	
1	DACR and DACNT can be accessed	(Initial value)

10.2.2 D/A Data Registers A and B (DADRA and DADRB)

	DADRH								DADRL							
Bit (CPU)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit (Data)	13	12	11	10	9	8	7	6	5	4	3	2	1	0	—	—
DADRA	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—
Initial value	1	1	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	R/W	—
DADRB	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS
Initial value	1	1	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	R/W	R/W

There are two 16-bit readable/writable D/A data registers: DADRA and DADRB. DADRA corresponds to PWM D/A channel A, and DADRB to PWM D/A channel B. The CPU can read and write the PWM D/A data register values, but since DADRA and DADRB are 16-bit registers, data transfers between them and the CPU are performed using a temporary register (TEMP). See section 10.3, Bus Master Interface, for details.

The least significant (CPU) bit of DADRA is not used and is always read as 1.

DADR is initialized to H'FFFF by a reset, and in the standby modes, watch mode, subactive mode, subsleep mode, and module stop mode.

Bits 15 to 3—PWM D/A Data 13 to 0 (DA13 to DA0): The digital value to be converted to an analog value is set in the upper 14 bits of the PWM D/A data register.

In each base cycle, the DACNT value is continually compared with these upper 14 bits to determine the duty cycle of the output waveform, and to decide whether to output a fine-adjustment pulse equal in width to the resolution. To enable this operation, the data register must be set within a range that depends on the carrier frequency select bit (CFS). If the DADR value is outside this range, the PWM output is held constant.

A channel can be operated with 12-bit precision by keeping the two lowest data bits (DA0 and DA1) cleared to 0 and writing the data to be converted in the upper 12 bits. The two lowest data bits correspond to the two highest counter (DACNT) bits.

Bit 1—Carrier Frequency Select (CFS)

Bit 1		
CFS	**Description**	
0	Base cycle = resolution (T) × 64 DADR range = H'0401 to H'FFFD	
1	Base cycle = resolution (T) × 256 DADR range = H'0103 to H'FFFFF	(Initial value)

DADRA Bit 0—Reserved: This bit cannot be modified and is always read as 1.

DADRB Bit 0—Register Select (REGS): DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed. The REGS bit can be accessed regardless of whether DADRB or DACNT is selected.

Bit 0		
REGS	**Description**	
0	DADRA and DADRB can be accessed	
1	DACR and DACNT can be accessed	(Initial value)

10.2.3 PWM D/A Control Register (DACR)

Bit	7	6	5	4	3	2	1	0
	TEST	PWME	—	—	OEB	OEA	OS	CKS
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

DACR is an 8-bit readable/writable register that selects test mode, enables the PWM outputs, and selects the output phase and operating speed.

DACR is initialized to H'30 by a reset, and in the standby modes, watch mode, subactive mode, subsleep mode, and module stop mode.

Bit 7—Test Mode (TEST): Selects test mode, which is used in testing the chip. Normally this bit should be cleared to 0.

Bit 7

TEST	Description	
0	PWM (D/A) in user state: normal operation	(Initial value)
1	PWM (D/A) in test state: correct conversion results unobtainable	

Bit 6—PWM Enable (PWME): Starts or stops the PWM D/A counter (DACNT).

Bit 6

PWME	Description	
0	DACNT operates as a 14-bit up-counter	(Initial value)
1	DACNT halts at H'0003	

Bits 5 and 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Output Enable B (OEB): Enables or disables output on PWM D/A channel B.

Bit 3

OEB	Description	
0	PWM (D/A) channel B output (at the PWX1 pin) is disabled	(Initial value)
1	PWM (D/A) channel B output (at the PWX1 pin) is enabled	

Bit 2—Output Enable A (OEA): Enables or disables output on PWM D/A channel A.

Bit 2		
OEA	**Description**	
0	PWM (D/A) channel A output (at the PWX0 pin) is disabled	(Initial value)
1	PWM (D/A) channel A output (at the PWX0 pin) is enabled	

Bit 1—Output Select (OS): Selects the phase of the PWM D/A output.

Bit 1		
OS	**Description**	
0	Direct PWM output	(Initial value)
1	Inverted PWM output	

Bit 0—Clock Select (CKS): Selects the PWM D/A resolution. If the system clock (ø) frequency is 10 MHz, resolutions of 100 ns and 200 ns can be selected.

Bit 0		
CKS	**Description**	
0	Operates at resolution (T) = system clock cycle time (t_{cyc})	(Initial value)
1	Operates at resolution (T) = system clock cycle time (t_{cyc}) × 2	

10.2.4 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	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	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers, and is used to perform module stop mode control.

When the MSTP11 bit is set to 1, 14-bit PWM timer operation is halted and a transition is made to module stop mode. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 3—Module Stop (MSTP11): Specifies PWMX module stop mode.

MSTPCRH Bit 3

MSTP11	Description	
0	PWMX module stop mode is cleared	
1	PWMX module stop mode is set	(Initial value)

10.3 Bus Master Interface

DACNT, DADRA, and DADRB are 16-bit registers. The data bus linking the bus master and the on-chip supporting modules, however, is only 8 bits wide. When the bus master accesses these registers, it therefore uses an 8-bit temporary register (TEMP).

These registers are written and read as follows (taking the example of the CPU interface).

- Write

  When the upper byte is written, the upper-byte write data is stored in TEMP. Next, when the lower byte is written, the lower-byte write data and TEMP value are combined, and the combined 16-bit value is written in the register.
- Read

  When the upper byte is read, the upper-byte value is transferred to the CPU and the lower-byte value is transferred to TEMP. Next, when the lower byte is read, the lower-byte value in TEMP is transferred to the CPU.

These registers should always be accessed 16 bits at a time using an MOV instruction (by word access or two consecutive byte accesses), and the upper byte should always be accessed before the lower byte. Correct data will not be transferred if only the upper byte or only the lower byte is accessed. Also note that a bit manipulation instruction cannot be used to access these registers.

Figure 10.2 shows the data flow for access to DACNT. The other registers are accessed similarly.

Example 1: Write to DACNT

```
MOV.W R0, @DACNT     ; Write R0 contents to DACNT
```

Example 2: Read DADRA

```
MOV.W @DADRA, R0     ; Copy contents of DADRA to R0
```

Table 10.3 Read and Write Access Methods for 16-Bit Registers

	Read		Write	
Register Name	**Word**	**Byte**	**Word**	**Byte**
DADRA and DADRB	Yes	Yes	Yes	×
DACNT	Yes	×	Yes	×

Notes: Yes: Permitted type of access. Word access includes successive byte accesses to the upper byte (first) and lower byte (second).
×: This type of access may give incorrect results.

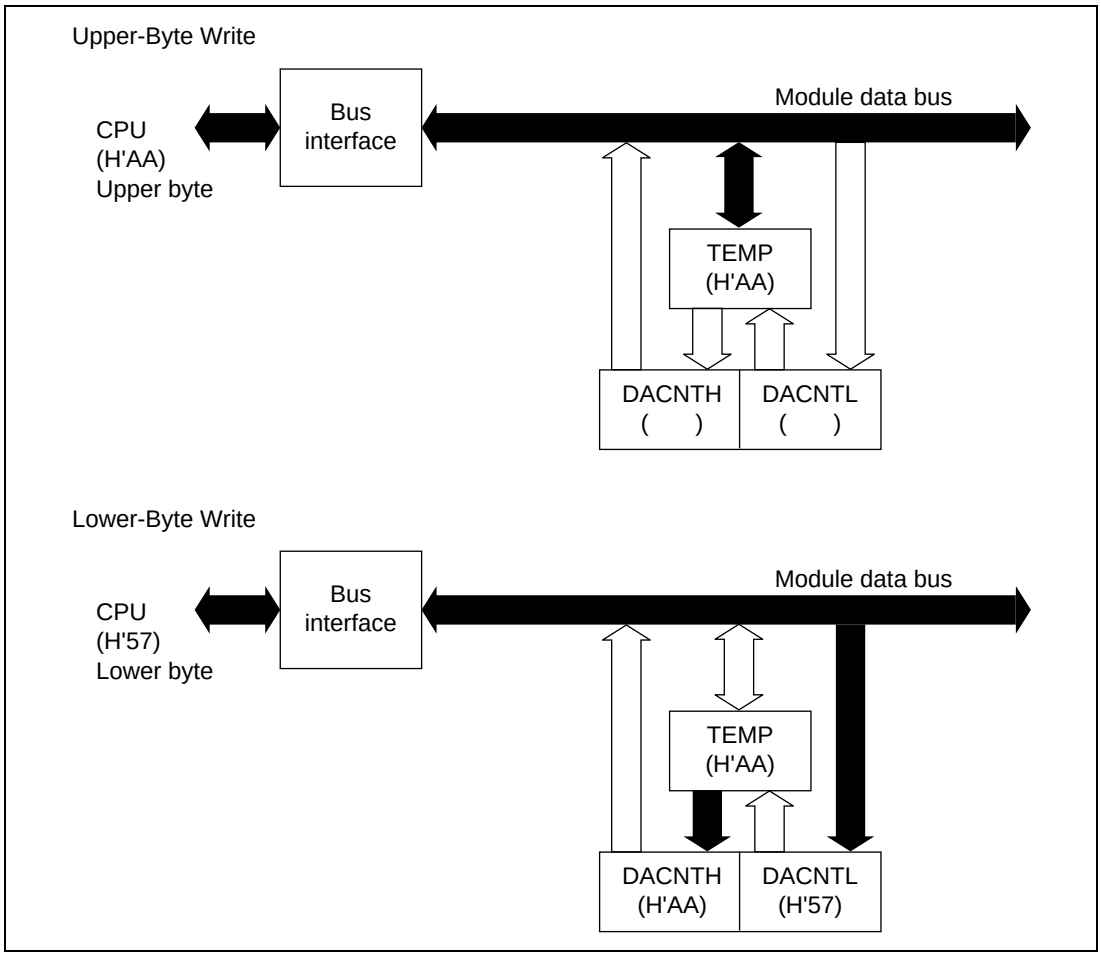


Figure 10.2 (a) Access to DACNT (CPU Writes H'AA57 to DACNT)

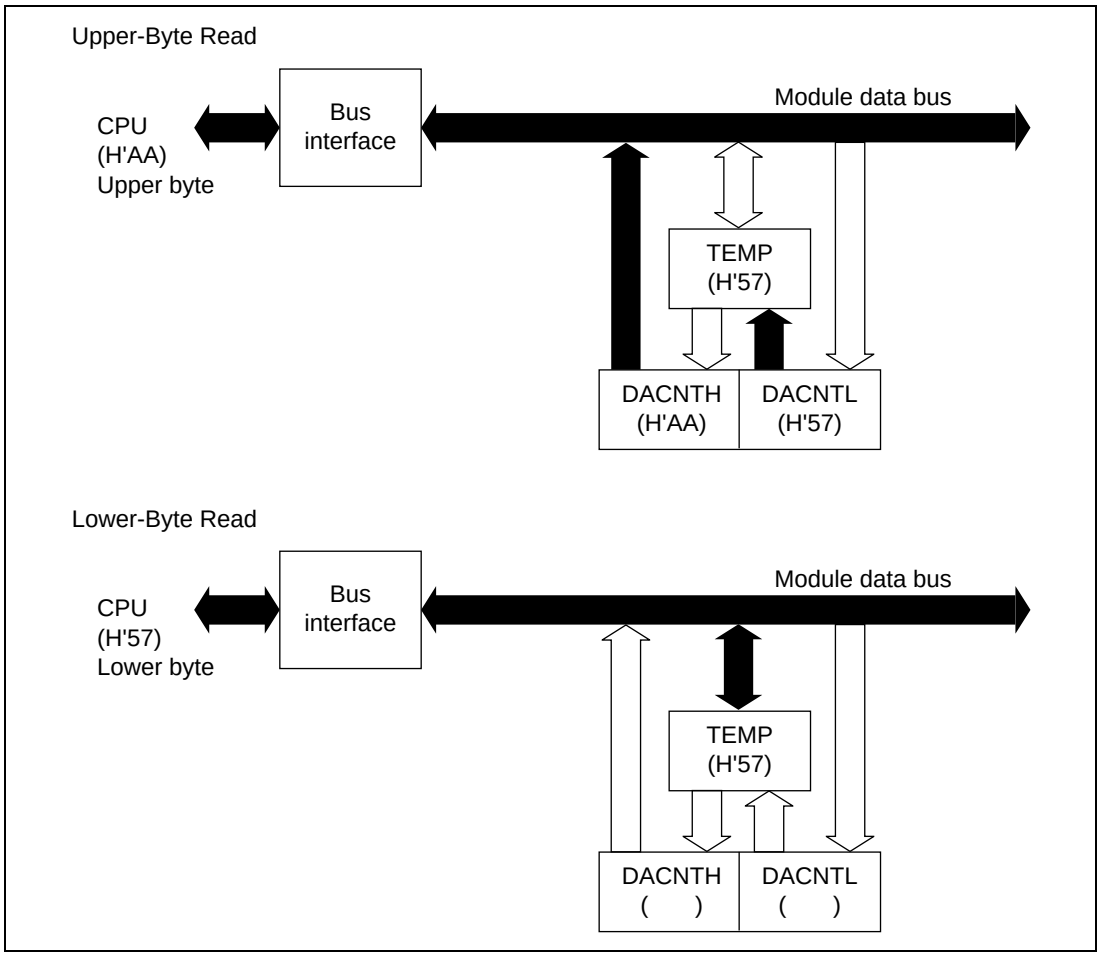


Figure 10.2 (b) Access to DACNT (CPU Reads H'AA57 from DACNT)

10.4 Operation

A PWM waveform like the one shown in figure 10.3 is output from the PWMX pin. When OS = 0, the value in DADR corresponds to the total width (T_L) of the low (0) pulses output in one conversion cycle (256 pulses when CFS = 0, 64 pulses when CFS = 1). When OS = 1, the output waveform is inverted and the DADR value corresponds to the total width (T_H) of the high (1) output pulses. Figure 10.4 shows the types of waveform output available.

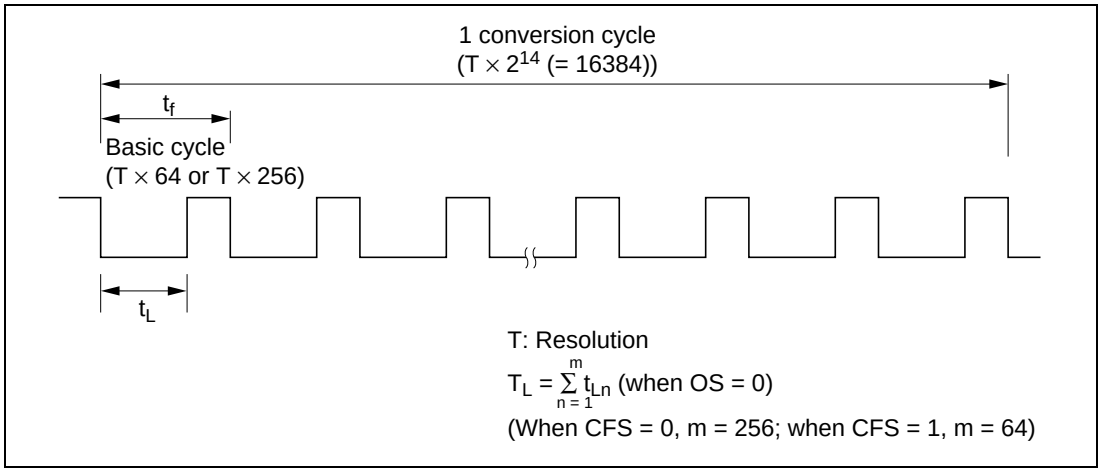


Figure 10.3 PWM D/A Operation

Table 10.4 summarizes the relationships of the CKS, CFS, and OS bit settings to the resolution, base cycle, and conversion cycle. The PWM output remains flat unless DADR contains at least a certain minimum value. Table 10.4 indicates the range of DADR settings that give an output waveform like the one in figure 10.3, and lists the conversion cycle length when low-order DADR bits are kept cleared to 0, reducing the conversion precision to 12 bits or 10 bits.

Table 10.4 Settings and Operation (Examples when ø = 10 MHz)

CKS	Resolution T (µs)	CFS	Base Cycle (µs)	Conversion Cycle (µs)	T_L (if OS = 0) T_H (if OS = 1)	Fixed DADR Bits: Precision (Bits)	Bit Data 3	2	1	0	Conversion Cycle* (µs)
0	0.1	0	6.4	1638.4	1. Always low (or high) (DADR = H'0001 to H'03FD)	14					1638.4
					2. (Data value) × T (DADR = H'0401 to H'FFFD)	12			0	0	409.6
						10	0	0	0	0	102.4
		1	25.6	1638.4	1. Always low (or high) (DADR = H'0003 to H'00FF)	14					1638.4
					2. (Data value) × T (DADR = H'0103 to H'FFFF)	12			0	0	409.6
						10	0	0	0	0	102.4
1	0.2	0	12.8	3276.8	1. Always low (or high) (DADR = H'0001 to H'03FD)	14					3276.8
					2. (Data value) × T (DADR = H'0401 to H'FFFD)	12			0	0	819.2
						10	0	0	0	0	204.8
		1	51.2	3276.8	1. Always low (or high) (DADR = H'0003 to H'00FF)	14					3276.8
					2. (Data value) × T (DADR = H'0103 to H'FFFF)	12			0	0	819.2
						10	0	0	0	0	204.8

Note: * This column indicates the conversion cycle when specific DADR bits are fixed.

1. OS = 0 (DADR corresponds to T_L)
   a. CFS = 0 [base cycle = resolution (T) × 64]

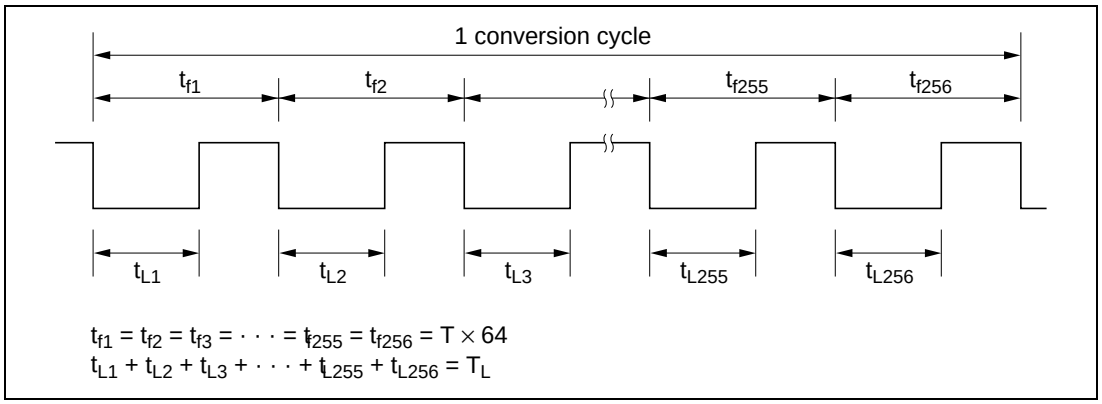


Figure 10.4 (1) Output Waveform

   b. CFS = 1 [base cycle = resolution (T) × 256]

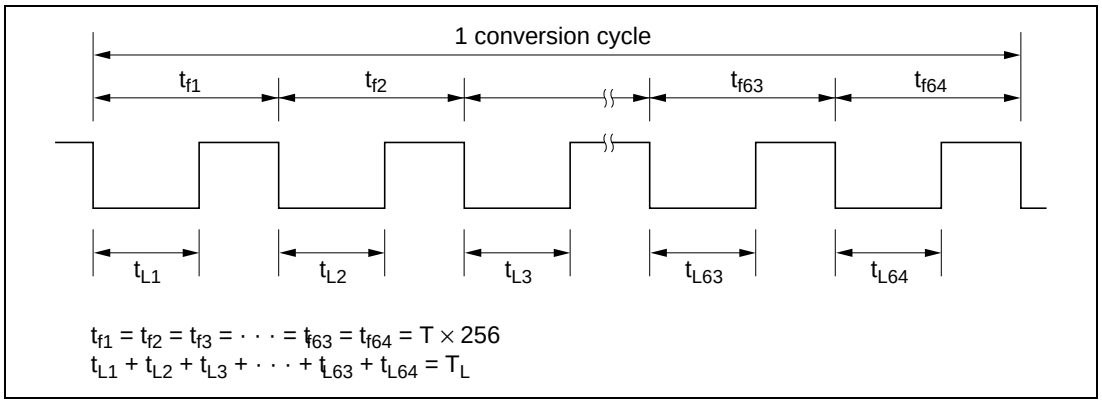


Figure 10.4 (2) Output Waveform

2. OS = 1 (DADR corresponds to T_H)
   a. CFS = 0 [base cycle = resolution (T) × 64]

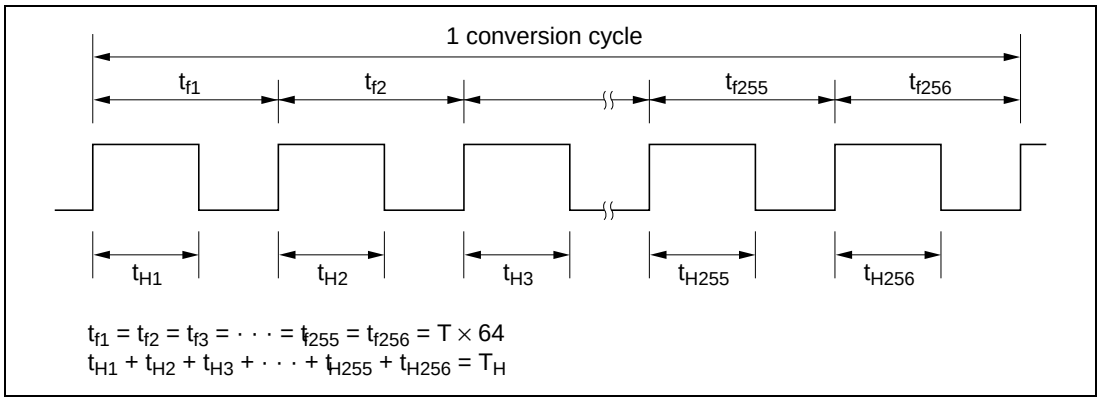


Figure 10.4 (3) Output Waveform

   b. CFS = 1 [base cycle = resolution (T) × 256]

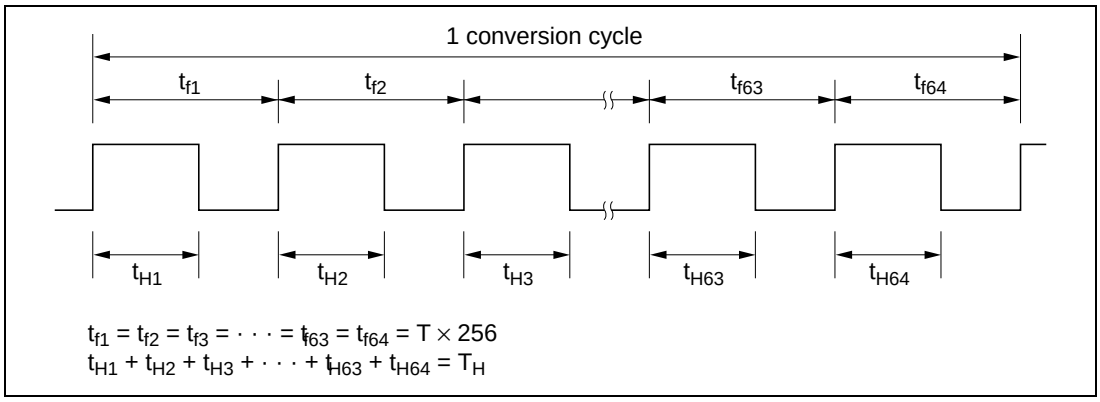


Figure 10.4 (4) Output Waveform

Section 11 16-Bit Free-Running Timer

11.1 Overview

The H8S/2128 Series and H8S/2124 Series have a single-channel on-chip 16-bit free-running timer (FRT) module that uses a 16-bit free-running counter as a time base. Applications of the FRT module include rectangular-wave output (up to two independent waveforms), input pulse width measurement, and measurement of external clock periods.

11.1.1 Features

The features of the free-running timer module are listed below.

- Selection of four clock sources
  - The free-running counter can be driven by an internal clock source (ø/2, ø/8, or ø/32), or an external clock input (enabling use as an external event counter).
- Two independent comparators
  - Each comparator can generate an independent waveform.
- Four input capture channels
  - The current count can be captured on the rising or falling edge (selectable) of an input signal.
  - The four input capture registers can be used separately, or in a buffer mode.
- Counter can be cleared under program control
  - The free-running counters can be cleared on compare-match A.
- Seven independent interrupts
  - Two compare-match interrupts, four input capture interrupts, and one overflow interrupt can be requested independently.
- Special functions provided by automatic addition function
  - The contents of OCRAR and OCRAF can be added to the contents of OCRA automatically, enabling a periodic waveform to be generated without software intervention.
  - The contents of ICRD can be added automatically to the contents of OCRDM × 2, enabling input capture operations in this interval to be restricted.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the free-running timer.

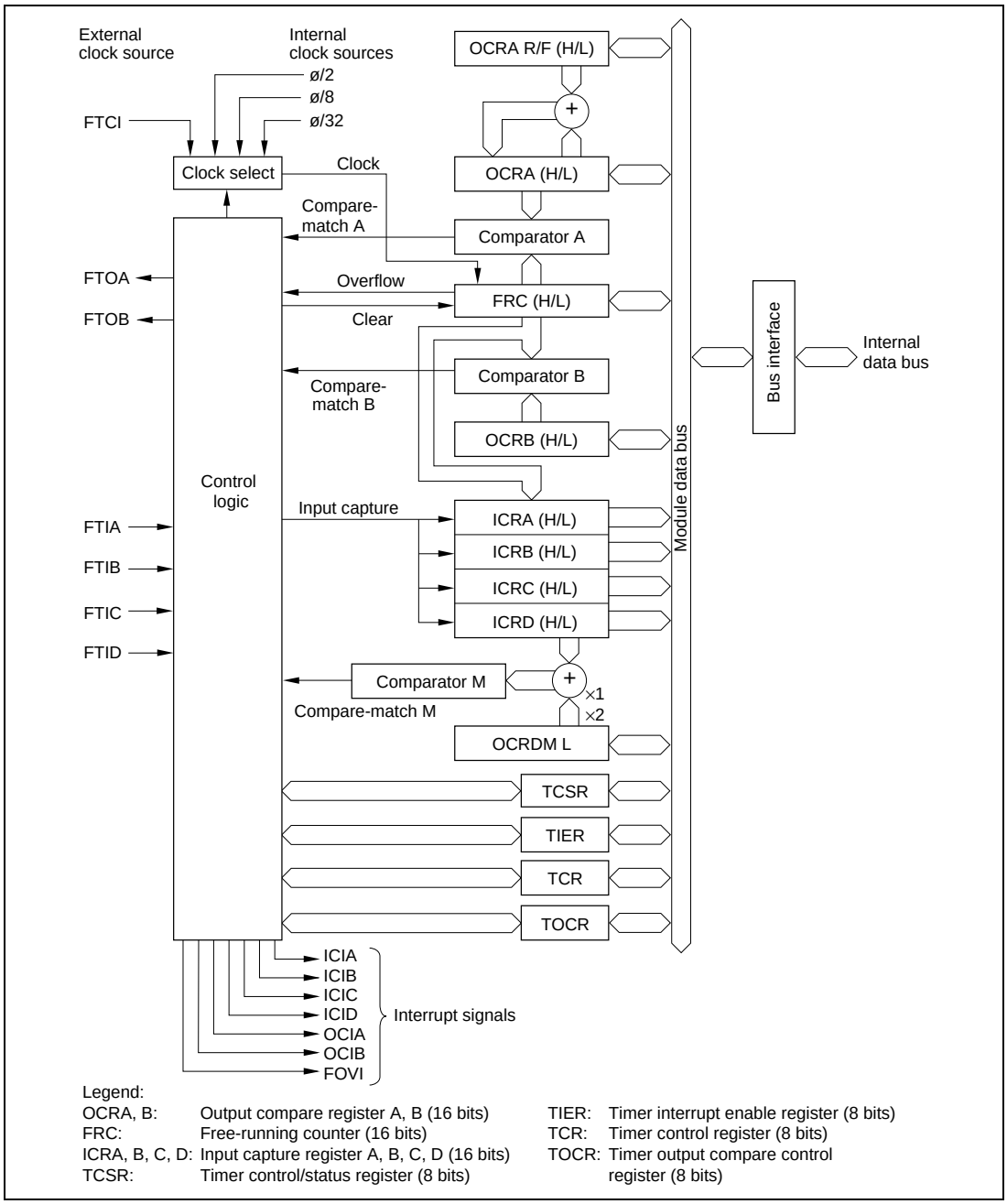


Figure 11.1 Block Diagram of 16-Bit Free-Running Timer

11.1.3 Input and Output Pins

Table 11.1 lists the input and output pins of the free-running timer module.

Table 11.1 Input and Output Pins of Free-Running Timer Module

Name	Abbreviation	I/O	Function
Counter clock input	FTCI	Input	FRC counter clock input
Output compare A	FTOA	Output	Output compare A output
Output compare B	FTOB	Output	Output compare B output
Input capture A	FTIA	Input	Input capture A input
Input capture B	FTIB	Input	Input capture B input
Input capture C	FTIC	Input	Input capture C input
Input capture D	FTID	Input	Input capture D input

11.1.4 Register Configuration

Table 11.2 lists the registers of the free-running timer module.

Table 11.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address*[1]
Timer interrupt enable register	TIER	R/W	H'01	H'FF90
Timer control/status register	TCSR	R/(W)*[2]	H'00	H'FF91
Free-running counter	FRC	R/W	H'0000	H'FF92
Output compare register A	OCRA	R/W	H'FFFF	H'FF94*[3]
Output compare register B	OCRB	R/W	H'FFFF	H'FF94*[3]
Timer control register	TCR	R/W	H'00	H'FF96
Timer output compare control register	TOCR	R/W	H'00	H'FF97
Input capture register A	ICRA	R	H'0000	H'FF98*[4]
Input capture register B	ICRB	R	H'0000	H'FF9A*[4]
Input capture register C	ICRC	R	H'0000	H'FF9C*[4]
Input capture register D	ICRD	R	H'0000	H'FF9E
Output compare register AR	OCRAR	R/W	H'FFFF	H'FF98*[4]
Output compare register AF	OCRAF	R/W	H'FFFF	H'FF9A*[4]
Output compare register DM	OCRDM	R/W	H'0000	H'FF9C*[4]
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

Notes: 1. Lower 16 bits of the address.
2. Bits 7 to 1 are read-only; only 0 can be written to clear the flags.
Bit 0 is readable/writable.
3. OCRA and OCRB share the same address. Access is controlled by the OCRS bit in TOCR.
4. ICRA, ICRB, and ICRC share the same addresses with OCRAR, OCRAF, and OCRDM. Access is controlled by the ICRS bit in TOCR.

11.2 Register Descriptions

11.2.1 Free-Running Counter (FRC)

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

FRC is a 16-bit readable/writable up-counter that increments on an internal pulse generated from a clock source. The clock source is selected by bits CKS1 and CKS0 in TCR.

FRC can also be cleared by compare-match A.

When FRC overflows from H'FFFF to H'0000, the overflow flag (OVF) in TCSR is set to 1.

FRC is initialized to H'0000 by a reset and in hardware standby mode.

11.2.2 Output Compare Registers A and B (OCRA, OCRB)

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

OCRA and OCRB are 16-bit readable/writable registers, the contents of which are continually compared with the value in the FRC. When a match is detected, the corresponding output compare flags (OCFA or OCFB) is set in TCSR.

In addition, if the output enable bit (OEA or OEB) in TOCR is set to 1, when OCR and FRC values match, the logic level selected by the output level bit (OLVLA or OLVLB) in TOCR is output at the output compare pin (FTOA or FTOB). Following a reset, the FTOA and FTOB output levels are 0 until the first compare-match.

OCR is initialized to H'FFFF by a reset and in hardware standby mode.

11.2.3 Input Capture Registers A to D (ICRA to ICRD)

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

There are four input capture registers, A to D, each of which is a 16-bit read-only register.

When the rising or falling edge of the signal at an input capture input pin (FTIA to FTID) is detected, the current FRC value is copied to the corresponding input capture register (ICRA to ICRD). At the same time, the corresponding input capture flag (ICFA to ICFD) in TCSR is set to 1. The input capture edge is selected by the input edge select bits (IEDGA to IEDGD) in TCR.

ICRC and ICRD can be used as ICRA and ICRB buffer registers, respectively, and made to perform buffer operations, by means of buffer enable bits A and B (BUFEA, BUFEB) in TCR.

Figure 11.2 shows the connections when ICRC is specified as the ICRA buffer register (BUFEA = 1). When ICRC is used as the ICRA buffer, both rising and falling edges can be specified as transitions of the external input signal by setting IEDGA _ IEDGC. When IEDGA = IEDGC, either the rising or falling edge is designated. See table 11.3.

Note: The FRC contents are transferred to the input capture register regardless of the value of the input capture flag (ICF).

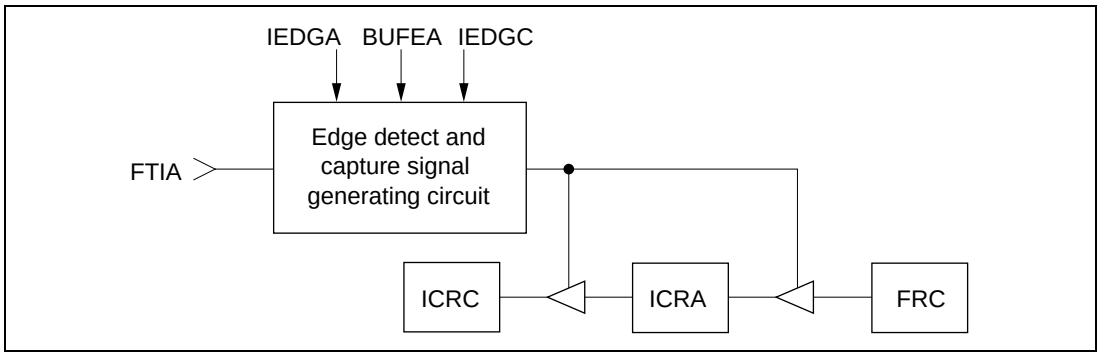


Figure 11.2 Input Capture Buffering (Example)

Table 11.3 Buffered Input Capture Edge Selection (Example)

IEDGA	IEDGC	Description
0	0	Captured on falling edge of input capture A (FTIA) (Initial value)
	1	Captured on both rising and falling edges of input capture A (FTIA)
1	0	
	1	Captured on rising edge of input capture A (FTIA)

To ensure input capture, the width of the input capture pulse should be at least 1.5 system clock periods (1.5ø). When triggering is enabled on both edges, the input capture pulse width should be at least 2.5 system clock periods.

ICR is initialized to H'0000 by a reset and in hardware standby mode.

11.2.4 Output Compare Registers AR and AF (OCRAR, OCRAF)

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

OCRAR and OCRAF are 16-bit readable/writable registers.

When the OCRAMS bit in TOCR is set to 1, the operation of OCRA is changed to include the use of OCRAR and OCRAF. The contents of OCRAR and OCRAF are automatically added alternately to OCRA, and the result is written to OCRA. The write operation is performed on the occurrence of compare-match A. In the first compare-match A after the OCRAMS bit is set to 1, OCRAF is added.

The operation due to compare-match A varies according to whether the compare-match follows addition of OCRAR or OCRAF. The value of the OLVLA bit in TOCR is ignored, and 1 is output on a compare-match A following addition of OCRAF, while 0 is output on a compare-match A following addition of OCRAR.

When the OCRA automatically addition function is used, do not set internal clock ø/2 as the FRC counter input clock together with an OCRAR (or OCRAF) value of H'0001 or less.

OCRAR and OCRAF are initialized to H'FFFF by a reset and in hardware standby mode.

11.2.5 Output Compare Register DM (OCRDM)

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

OCRDM is a 16-bit readable/writable register in which the upper 8 bits are fixed at H'00.

When the ICRDMS bit in TOCR is set to 1 and the contents of OCRDM are other than H'0000, the operation of ICRD is changed to include the use of OCRDM. The point at which input capture D occurs is taken as the start of a mask interval. Next, twice the contents of OCRDM is added to the contents of ICRD, and the result is compared with the FRC value. The point at which the values match is taken as the end of the mask interval. New input capture D events are disabled during the mask interval.

A mask interval is not generated when the ICRDMS bit is set to 1 and the contents of OCRDM are H'0000.

OCRDM is initialized to H'0000 by a reset and in hardware standby mode.

11.2.6 Timer Interrupt Enable Register (TIER)

Bit	7	6	5	4	3	2	1	0
	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

TIER is an 8-bit readable/writable register that enables and disables interrupts.

TIER is initialized to H'01 by a reset and in hardware standby mode.

Bit 7—Input Capture Interrupt A Enable (ICIAE): Selects whether to request input capture interrupt A (ICIA) when input capture flag A (ICFA) in TCSR is set to 1.

Bit 7

ICIAE	Description	
0	Input capture interrupt request A (ICIA) is disabled	(Initial value)
1	Input capture interrupt request A (ICIA) is enabled	

Bit 6—Input Capture Interrupt B Enable (ICIBE): Selects whether to request input capture interrupt B (ICIB) when input capture flag B (ICFB) in TCSR is set to 1.

Bit 6		
ICIBE	**Description**	
0	Input capture interrupt request B (ICIB) is disabled	(Initial value)
1	Input capture interrupt request B (ICIB) is enabled	

Bit 5—Input Capture Interrupt C Enable (ICICE): Selects whether to request input capture interrupt C (ICIC) when input capture flag C (ICFC) in TCSR is set to 1.

Bit 5		
ICICE	**Description**	
0	Input capture interrupt request C (ICIC) is disabled	(Initial value)
1	Input capture interrupt request C (ICIC) is enabled	

Bit 4—Input Capture Interrupt D Enable (ICIDE): Selects whether to request input capture interrupt D (ICID) when input capture flag D (ICFD) in TCSR is set to 1.

Bit 4		
ICIDE	**Description**	
0	Input capture interrupt request D (ICID) is disabled	(Initial value)
1	Input capture interrupt request D (ICID) is enabled	

Bit 3—Output Compare Interrupt A Enable (OCIAE): Selects whether to request output compare interrupt A (OCIA) when output compare flag A (OCFA) in TCSR is set to 1.

Bit 3		
OCIAE	**Description**	
0	Output compare interrupt request A (OCIA) is disabled	(Initial value)
1	Output compare interrupt request A (OCIA) is enabled	

Bit 2—Output Compare Interrupt B Enable (OCIBE): Selects whether to request output compare interrupt B (OCIB) when output compare flag B (OCFB) in TCSR is set to 1.

Bit 2		
OCIBE	**Description**	
0	Output compare interrupt request B (OCIB) is disabled	(Initial value)
1	Output compare interrupt request B (OCIB) is enabled	

Bit 1—Timer Overflow Interrupt Enable (OVIE): Selects whether to request a free-running timer overflow interrupt (FOVI) when the timer overflow flag (OVF) in TCSR is set to 1.

Bit 1		
OVIE	**Description**	
0	Timer overflow interrupt request (FOVI) is disabled	(Initial value)
1	Timer overflow interrupt request (FOVI) is enabled	

Bit 0—Reserved: This bit cannot be modified and is always read as 1.

11.2.7 Timer Control/Status Register (TCSR)

Bit	7	6	5	4	3	2	1	0
	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

Note: * Only 0 can be written in bits 7 to 1 to clear these flags.

TCSR is an 8-bit register used for counter clear selection and control of interrupt request signals.

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

Timing is described in section 11.3, Operation.

Bit 7—Input Capture Flag A (ICFA): This status flag indicates that the FRC value has been transferred to ICRA by means of an input capture signal. When BUFEA = 1, ICFA indicates that the old ICRA value has been moved into ICRC and the new FRC value has been transferred to ICRA.

ICFA must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 7		
ICFA	**Description**	
0	[Clearing condition] Read ICFA when ICFA = 1, then write 0 in ICFA	(Initial value)
1	[Setting condition] When an input capture signal causes the FRC value to be transferred to ICRA	

Bit 6—Input Capture Flag B (ICFB): This status flag indicates that the FRC value has been transferred to ICRB by means of an input capture signal. When BUFEB = 1, ICFB indicates that the old ICRB value has been moved into ICRD and the new FRC value has been transferred to ICRB.

ICFB must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 6

ICFB	Description	
0	[Clearing condition]	(Initial value)
	Read ICFB when ICFB = 1, then write 0 in ICFB	
1	[Setting condition]	
	When an input capture signal causes the FRC value to be transferred to ICRB	

Bit 5—Input Capture Flag C (ICFC): This status flag indicates that the FRC value has been transferred to ICRC by means of an input capture signal. When BUFEA = 1, on occurrence of the signal transition in FTIC (input capture signal) specified by the IEDGC bit, ICFC is set but data is not transferred to ICRC. Therefore, in buffer operation, ICFC can be used as an external interrupt signal (by setting the ICICE bit to 1).

ICFC must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 5

ICFC	Description	
0	[Clearing condition]	(Initial value)
	Read ICFC when ICFC = 1, then write 0 in ICFC	
1	[Setting condition]	
	When an input capture signal is received	

Bit 4—Input Capture Flag D (ICFD): This status flag indicates that the FRC value has been transferred to ICRD by means of an input capture signal. When BUFEB = 1, on occurrence of the signal transition in FTID (input capture signal) specified by the IEDGD bit, ICFD is set but data is not transferred to ICRD. Therefore, in buffer operation, ICFD can be used as an external interrupt by setting the ICIDE bit to 1.

ICFD must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 4

ICFD	Description	
0	[Clearing condition]	(Initial value)
	Read ICFD when ICFD = 1, then write 0 in ICFD	
1	[Setting condition]	
	When an input capture signal is received	

Bit 3—Output Compare Flag A (OCFA): This status flag indicates that the FRC value matches the OCRA value. This flag must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 3

OCFA	Description	
0	[Clearing condition]	(Initial value)
	Read OCFA when OCFA = 1, then write 0 in OCFA	
1	[Setting condition]	
	When FRC = OCRA	

Bit 2—Output Compare Flag B (OCFB): This status flag indicates that the FRC value matches the OCRB value. This flag must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 2

OCFB	Description	
0	[Clearing condition]	(Initial value)
	Read OCFB when OCFB = 1, then write 0 in OCFB	
1	[Setting condition]	
	When FRC = OCRB	

Bit 1—Timer Overflow Flag (OVF): This status flag indicates that the FRC has overflowed (changed from H'FFFF to H'0000). This flag must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 1

OVF	Description	
0	[Clearing condition]	(Initial value)
	Read OVF when OVF = 1, then write 0 in OVF	
1	[Setting condition]	
	When FRC changes from H'FFFF to H'0000	

Bit 0—Counter Clear A (CCLRA): This bit selects whether the FRC is to be cleared at compare-match A (when the FRC and OCRA values match).

Bit 0

CCLRA	Description	
0	FRC clearing is disabled	(Initial value)
1	FRC is cleared at compare-match A	

11.2.8 Timer Control Register (TCR)

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCR is an 8-bit readable/writable register that selects the rising or falling edge of the input capture signals, enables the input capture buffer mode, and selects the FRC clock source.

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

Bit 7—Input Edge Select A (IEDGA): Selects the rising or falling edge of the input capture A signal (FTIA).

Bit 7

IEDGA	Description	
0	Capture on the falling edge of FTIA	(Initial value)
1	Capture on the rising edge of FTIA	

Bit 6—Input Edge Select B (IEDGB): Selects the rising or falling edge of the input capture B signal (FTIB).

Bit 6

IEDGB	Description	
0	Capture on the falling edge of FTIB	(Initial value)
1	Capture on the rising edge of FTIB	

Bit 5—Input Edge Select C (IEDGC): Selects the rising or falling edge of the input capture C signal (FTIC).

Bit 5

IEDGC	Description	
0	Capture on the falling edge of FTIC	(Initial value)
1	Capture on the rising edge of FTIC	

Bit 4—Input Edge Select D (IEDGD): Selects the rising or falling edge of the input capture D signal (FTID).

Bit 4		
IEDGD	**Description**	
0	Capture on the falling edge of FTID	(Initial value)
1	Capture on the rising edge of FTID	

Bit 3—Buffer Enable A (BUFEA): Selects whether ICRC is to be used as a buffer register for ICRA.

Bit 3		
BUFEA	**Description**	
0	ICRC is not used as a buffer register for input capture A	(Initial value)
1	ICRC is used as a buffer register for input capture A	

Bit 2—Buffer Enable B (BUFEB): Selects whether ICRD is to be used as a buffer register for ICRB.

Bit 2		
BUFEB	**Description**	
0	ICRD is not used as a buffer register for input capture B	(Initial value)
1	ICRD is used as a buffer register for input capture B	

Bits 1 and 0—Clock Select (CKS1, CKS0): Select external clock input or one of three internal clock sources for the FRC. External clock pulses are counted on the rising edge of signals input to the external clock input pin (FTCI).

Bit 1	Bit 0		
CKS1	**CKS0**	**Description**	
0	0	ø/2 internal clock source	(Initial value)
	1	ø/8 internal clock source	
1	0	ø/32 internal clock source	
	1	External clock source (rising edge)	

11.2.9 Timer Output Compare Control Register (TOCR)

Bit	7	6	5	4	3	2	1	0
	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TOCR is an 8-bit readable/writable register that enables output from the output compare pins, selects the output levels, switches access between output compare registers A and B, controls the ICRD and OCRA operating mode, and switches access to input capture registers A, B, and C.

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

Bit 7—Input Capture D Mode Select (ICRDMS): Specifies whether ICRD is used in the normal operating mode or in the operating mode using OCRDM.

Bit 7

ICRDMS	Description	
0	The normal operating mode is specified for ICRD	(Initial value)
1	The operating mode using OCRDM is specified for ICRD	

Bit 6—Output Compare A Mode Select (OCRAMS): Specifies whether OCRA is used in the normal operating mode or in the operating mode using OCRAR and OCRAF.

Bit 6

OCRAMS	Description	
0	The normal operating mode is specified for OCRA	(Initial value)
1	The operating mode using OCRAR and OCRAF is specified for OCRA	

Bit 5—Input Capture Register Select (ICRS): The same addresses are shared by ICRA and OCRAR, by ICRB and OCRAF, and by ICRC and OCRDM. The ICRS bit determines which registers are selected when the shared addresses are read or written to. The operation of ICRA, ICRB, and ICRC is not affected.

Bit 5

ICRS	Description	
0	The ICRA, ICRB, and ICRC registers are selected	(Initial value)
1	The OCRAR, OCRAF, and OCRDM registers are selected	

Bit 4—Output Compare Register Select (OCRS): OCRA and OCRB share the same address. When this address is accessed, the OCRS bit selects which register is accessed. This bit does not affect the operation of OCRA or OCRB.

Bit 4		
OCRS	**Description**	
0	The OCRA register is selected	(Initial value)
1	The OCRB register is selected	

Bit 3—Output Enable A (OEA): Enables or disables output of the output compare A signal (FTOA).

Bit 3		
OEA	**Description**	
0	Output compare A output is disabled	(Initial value)
1	Output compare A output is enabled	

Bit 2—Output Enable B (OEB): Enables or disables output of the output compare B signal (FTOB).

Bit 2		
OEB	**Description**	
0	Output compare B output is disabled	(Initial value)
1	Output compare B output is enabled	

Bit 1—Output Level A (OLVLA): Selects the logic level to be output at the FTOA pin in response to compare-match A (signal indicating a match between the FRC and OCRA values). When the OCRAMS bit is 1, this bit is ignored.

Bit 1		
OLVLA	**Description**	
0	0 output at compare-match A	(Initial value)
1	1 output at compare-match A	

Bit 0—Output Level B (OLVLB): Selects the logic level to be output at the FTOB pin in response to compare-match B (signal indicating a match between the FRC and OCRB values).

Bit 0		
OLVLB	**Description**	
0	0 output at compare-match B	(Initial value)
1	1 output at compare-match B	

11.2.10 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	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	R/W	R/W

MSTPCR, comprising two 8-bit readable/writable registers, performs module stop mode control.

When the MSTP13 bit is set to 1, FRT operation is stopped at the end of the bus cycle, and module stop mode is entered. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 5—Module Stop (MSTP13): Specifies the FRT module stop mode.

Bit 5

MSTPCRH	Description	
0	FRT module stop mode is cleared	
1	FRT module stop mode is set	(Initial value)

11.3 Operation

11.3.1 FRC Increment Timing

FRC increments on a pulse generated once for each period of the selected (internal or external) clock source.

Internal Clock: Any of three internal clocks (ø/2, ø/8, or ø/32) created by division of the system clock (ø) can be selected by making the appropriate setting in bits CKS1 and CKS0 in TCR. Figure 11.3 shows the increment timing.

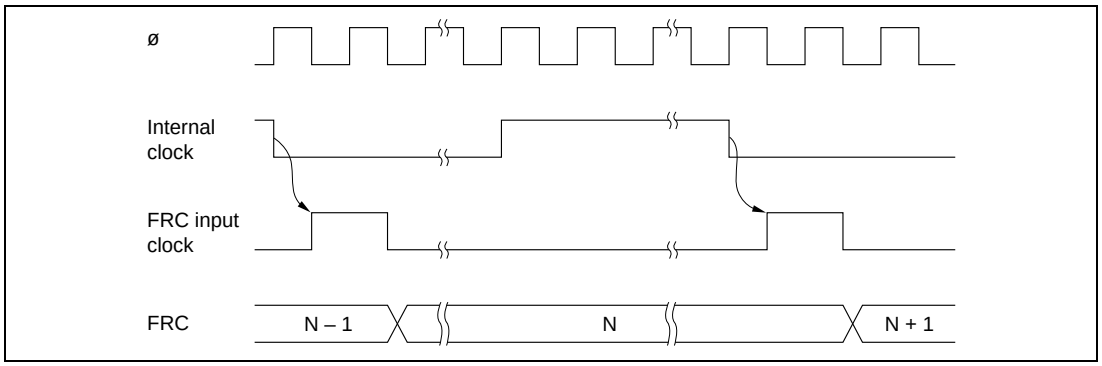


Figure 11.3 Increment Timing with Internal Clock Source

External Clock: If external clock input is selected by bits CKS1 and CKS0 in TCR, FRC increments on the rising edge of the external clock signal.

The pulse width of the external clock signal must be at least 1.5 system clock (ø) periods. The counter will not increment correctly if the pulse width is shorter than 1.5 system clock periods.

Figure 11.4 shows the increment timing.

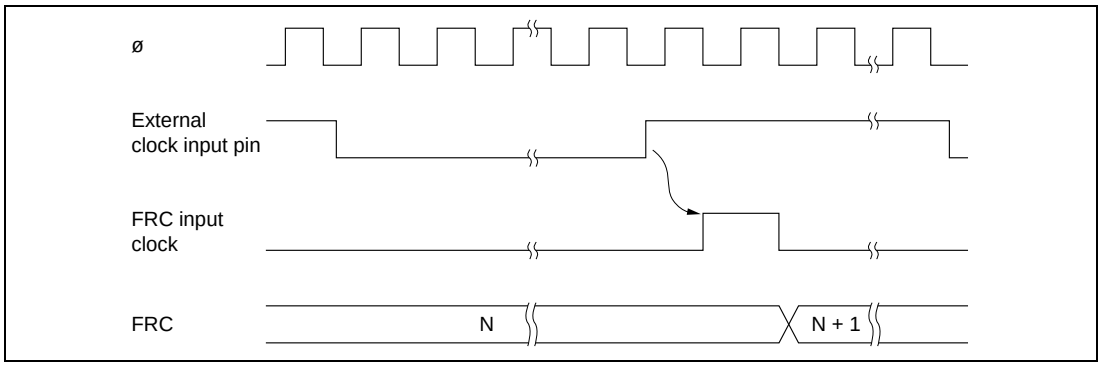


Figure 11.4 Increment Timing with External Clock Source

11.3.2 Output Compare Output Timing

When a compare-match occurs, the logic level selected by the output level bit (OLVLA or OLVLB) in TOCR is output at the output compare pin (FTOA or FTOB). Figure 11.5 shows the timing of this operation for compare-match A.

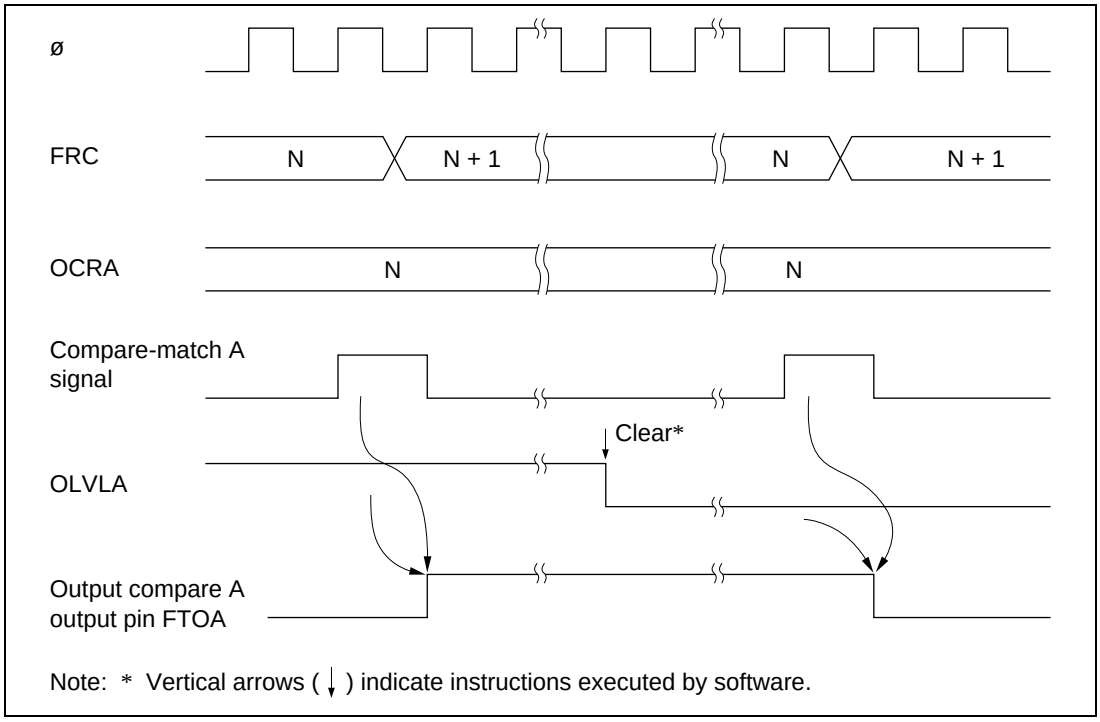


Figure 11.5 Timing of Output Compare A Output

11.3.3 FRC Clear Timing

FRC can be cleared when compare-match A occurs. Figure 11.6 shows the timing of this operation.

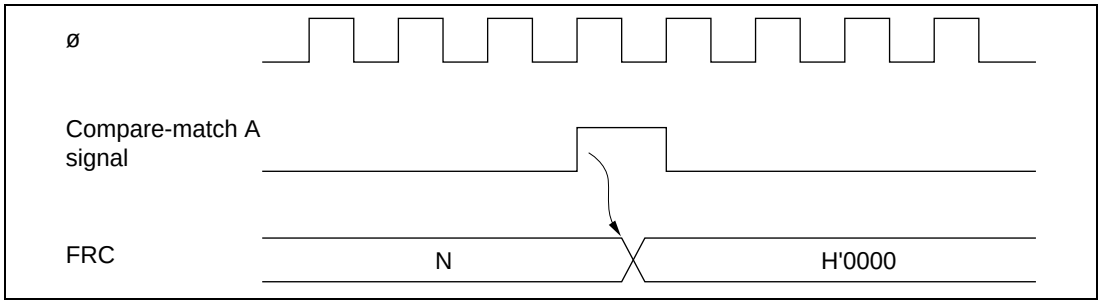


Figure 11.6 Clearing of FRC by Compare-Match A

11.3.4 Input Capture Input Timing

Input Capture Input Timing: An internal input capture signal is generated from the rising or falling edge of the signal at the input capture pin, as selected by the corresponding IEDGx (x = A to D) bit in TCR. Figure 11.7 shows the usual input capture timing when the rising edge is selected (IEDGx = 1).

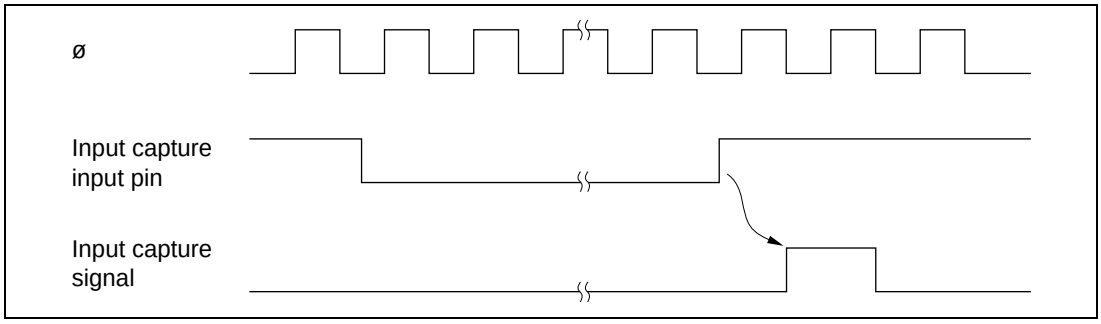


Figure 11.7 Input Capture Signal Timing (Usual Case)

If the upper byte of ICRA/B/C/D is being read when the corresponding input capture signal arrives, the internal input capture signal is delayed by one system clock (ø) period. Figure 11.8 shows the timing for this case.

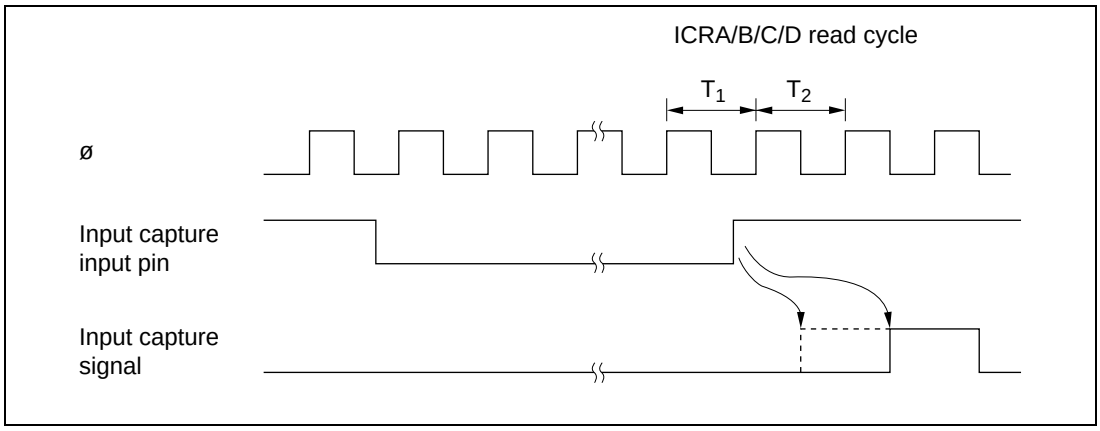


Figure 11.8 Input Capture Signal Timing (Input Capture Input when ICRA/B/C/D is Read)

Buffered Input Capture Input Timing: ICRC and ICRD can operate as buffers for ICRA and ICRB.

Figure 11.9 shows how input capture operates when ICRA and ICRC are used in buffer mode and IEDGA and IEDGC are set to different values (IEDGA = 0 and IEDGC = 1, or IEDG A = 1 and IEDGC = 0), so that input capture is performed on both the rising and falling edges of FTIA.

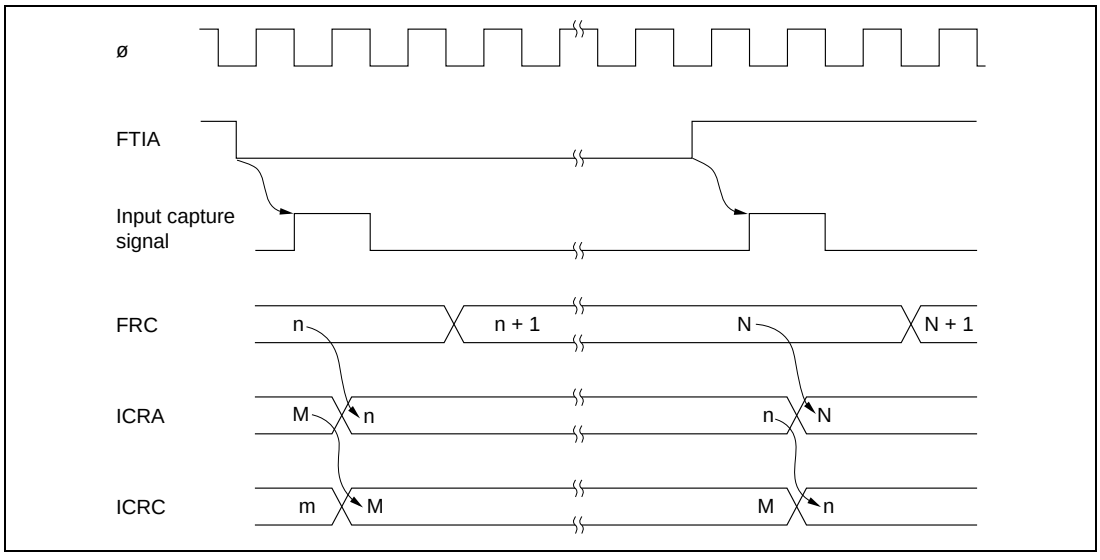


Figure 11.9 Buffered Input Capture Timing (Usual Case)

When ICRC or ICRD is used as a buffer register, its input capture flag is set by the selected transition of its input capture signal. For example, if ICRC is used to buffer ICRA, when the edge transition selected by the IEDGC bit occurs on the FTIC input capture line, ICFC will be set, and if the ICIEC bit is set, an interrupt will be requested. The FRC value will not be transferred to ICRC, however.

In buffered input capture, if the upper byte of either of the two registers to which data will be transferred (ICRA and ICRC, or ICRB and ICRD) is being read when the input signal arrives, input capture is delayed by one system clock (ø) period. Figure 11.10 shows the timing when BUFEA = 1.

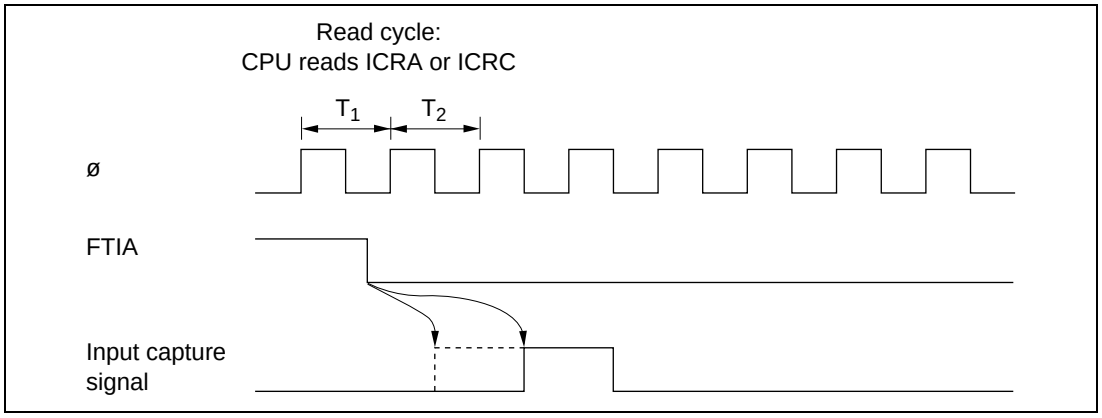


Figure 11.10 Buffered Input Capture Timing (Input Capture Input when ICRA or ICRC is Read)

11.3.5 Timing of Input Capture Flag (ICF) Setting

The input capture flag ICFx (x = A, B, C, D) is set to 1 by the internal input capture signal. The FRC value is simultaneously transferred to the corresponding input capture register (ICRx). Figure 11.11 shows the timing of this operation.

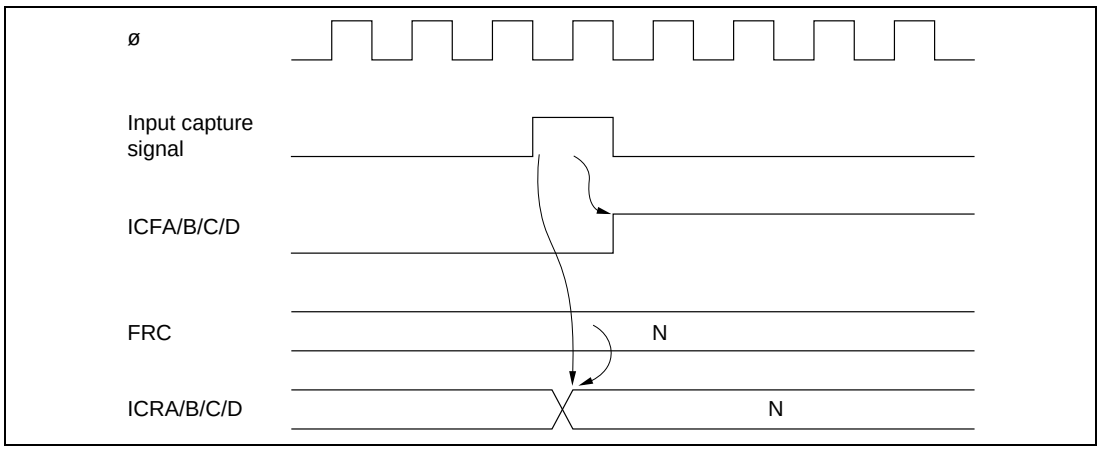


Figure 11.11 Setting of Input Capture Flag (ICFA/B/C/D)

11.3.6 Setting of Output Compare Flags A and B (OCFA, OCFB)

The output compare flags are set to 1 by an internal compare-match signal generated when the FRC value matches the OCRA or OCRB value. This compare-match signal is generated at the last state in which the two values match, just before FRC increments to a new value.

Accordingly, when the FRC and OCR values match, the compare-match signal is not generated until the next period of the clock source. Figure 11.12 shows the timing of the setting of OCFA and OCFB.

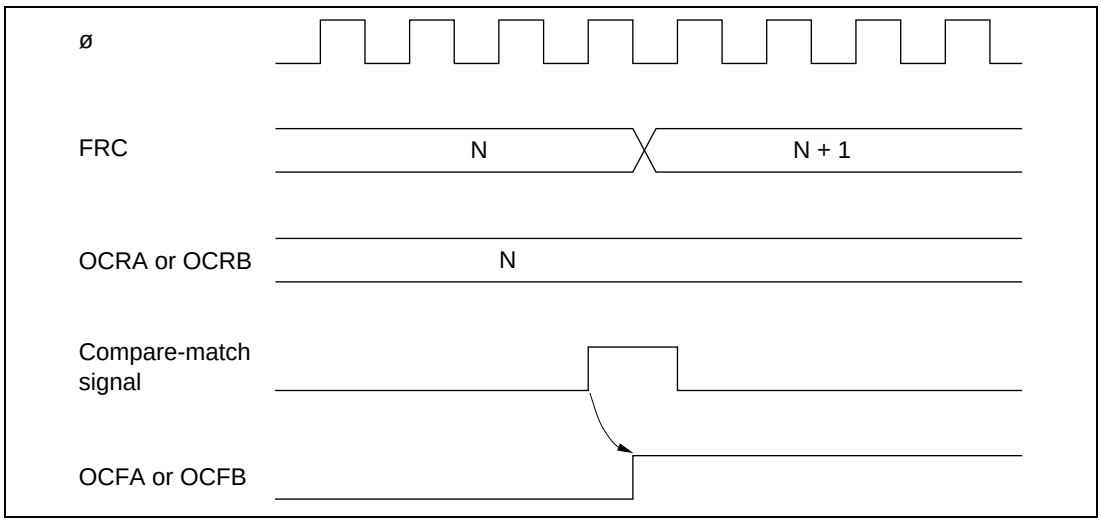


Figure 11.12 Setting of Output Compare Flag (OCFA, OCFB)

11.3.7 Setting of FRC Overflow Flag (OVF)

The FRC overflow flag (OVF) is set to 1 when FRC overflows (changes from H'FFFF to H'0000). Figure 11.13 shows the timing of this operation.

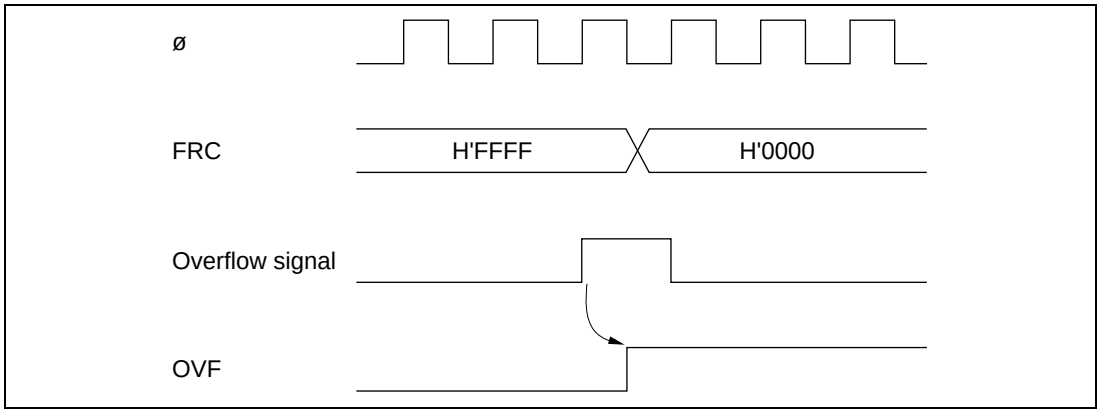


Figure 11.13 Setting of Overflow Flag (OVF)

11.3.8 Automatic Addition of OCRA and OCRAR/OCRAF

When the OCRAMS bit in TOCR is set to 1, the contents of OCRAR and OCRAF are automatically added to OCRA alternately, and when an OCRA compare-match occurs a write to OCRA is performed. The OCRA write timing is shown in figure 11.14.

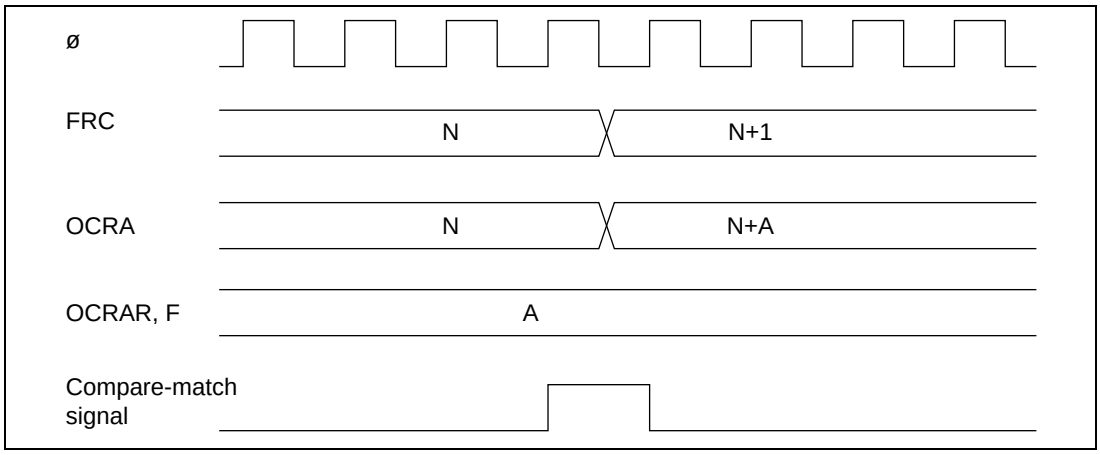


Figure 11.14 OCRA Automatic Addition Timing

11.3.9 ICRD and OCRDM Mask Signal Generation

When the ICRDMS bit in TOCR is set to 1 and the contents of OCRDM are other than H'0000, a signal that masks the ICRD input capture function is generated.

The mask signal is set by the input capture signal. The mask signal setting timing is shown in figure 11.15.

The mask signal is cleared by the sum of the ICRD contents and twice the OCRDM contents, and an FRC compare-match. The mask signal clearing timing is shown in figure 11.16.

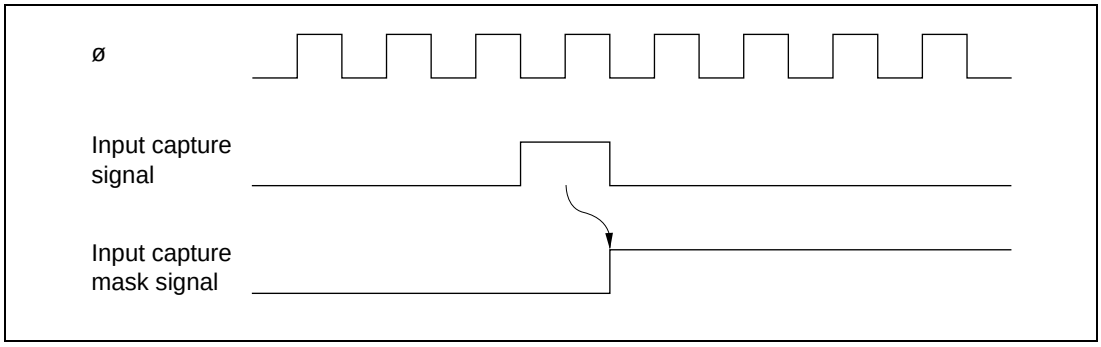


Figure 11.15 Input Capture Mask Signal Setting Timing

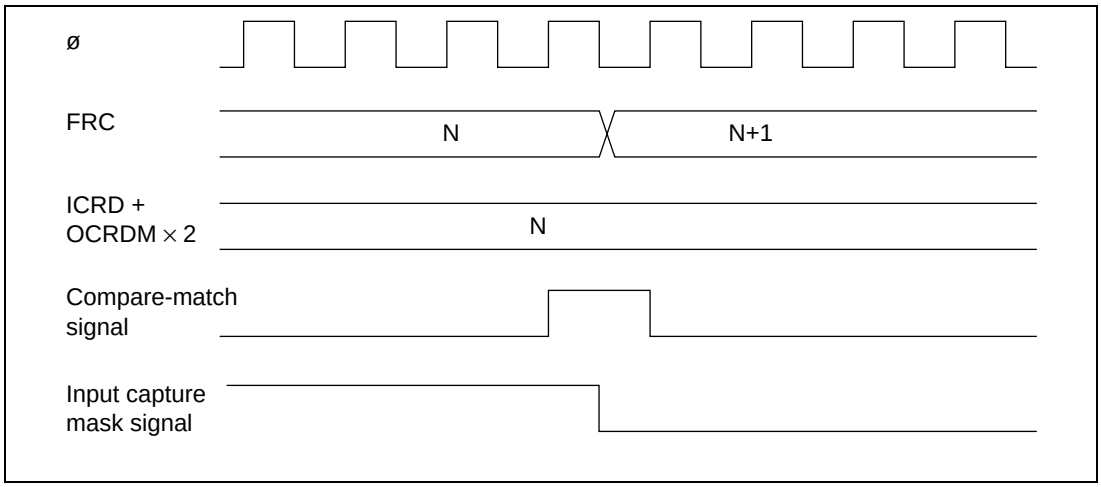


Figure 11.16 Input Capture Mask Signal Clearing Timing

11.4 Interrupts

The free-running timer can request seven interrupts (three types): input capture A to D (ICIA, ICIB, ICIC, ICID), output compare A and B (OCIA and OCIB), and overflow (FOVI). Each interrupt can be enabled or disabled by an enable bit in TIER. Independent signals are sent to the interrupt controller for each interrupt. Table 11.4 lists information about these interrupts.

Table 11.4 Free-Running Timer Interrupts

Interrupt	Description	DTC Activation	Priority
ICIA	Requested by ICFA	Possible	High
ICIB	Requested by ICFB	Possible	
ICIC	Requested by ICFC	Not possible	
ICID	Requested by ICFD	Not possible	
OCIA	Requested by OCFA	Possible	
OCIB	Requested by OCFB	Possible	
FOVI	Requested by OVF	Not possible	Low

11.5 Sample Application

In the example below, the free-running timer is used to generate pulse outputs with a 50% duty cycle and arbitrary phase relationship. The programming is as follows:

- The CCLRA bit in TCSR is set to 1.
- Each time a compare-match interrupt occurs, software inverts the corresponding output level bit in TOCR (OLVLA or OLVLB).

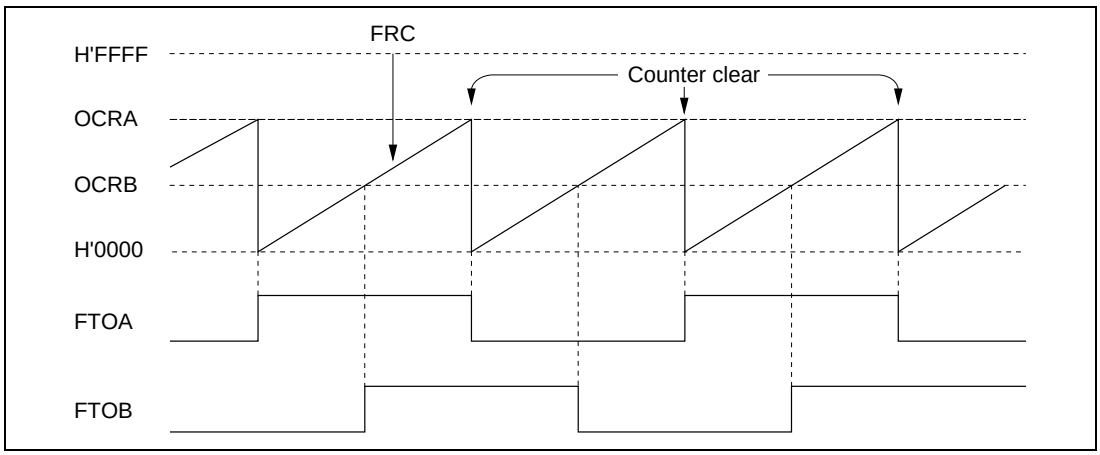


Figure 11.17 Pulse Output (Example)

11.6 Usage Notes

Application programmers should note that the following types of contention can occur in the free-running timer.

Contention between FRC Write and Clear: If an internal counter clear signal is generated during the state after an FRC write cycle, the clear signal takes priority and the write is not performed.

Figure 11.18 shows this type of contention.

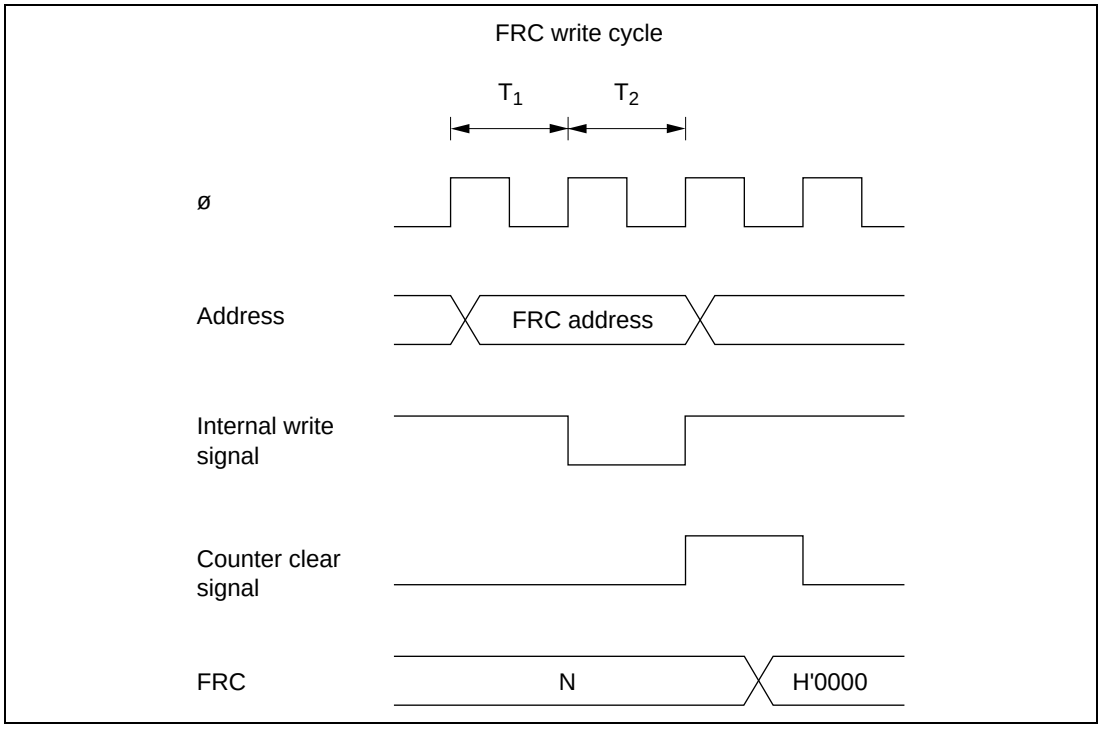


Figure 11.18 FRC Write-Clear Contention

Contention between FRC Write and Increment: If an FRC increment pulse is generated during the state after an FRC write cycle, the write takes priority and FRC is not incremented.

Figure 11.19 shows this type of contention.

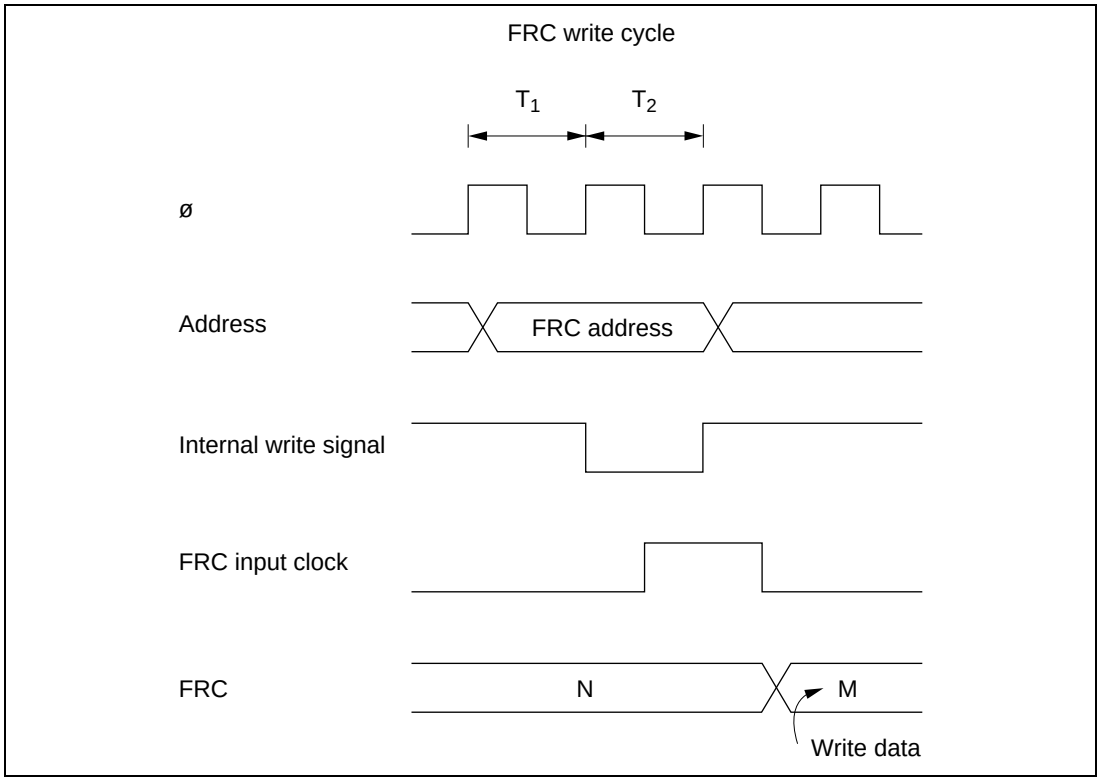


Figure 11.19 FRC Write-Increment Contention

Contention between OCR Write and Compare-Match: If a compare-match occurs during the state after an OCRA or OCRB write cycle, the write takes priority and the compare-match signal is inhibited.

Figure 11.20 shows this type of contention.

If automatic addition of OCRAR/OCRAF to OCRA is selected, and a compare-match occurs in the cycle following the OCRA, OCRAR, and OCRAF write cycle, the OCRA, OCRAR, and OCRAF write takes priority and the compare-match signal is inhibited. Consequently, the result of the automatic addition is not written to OCRA.

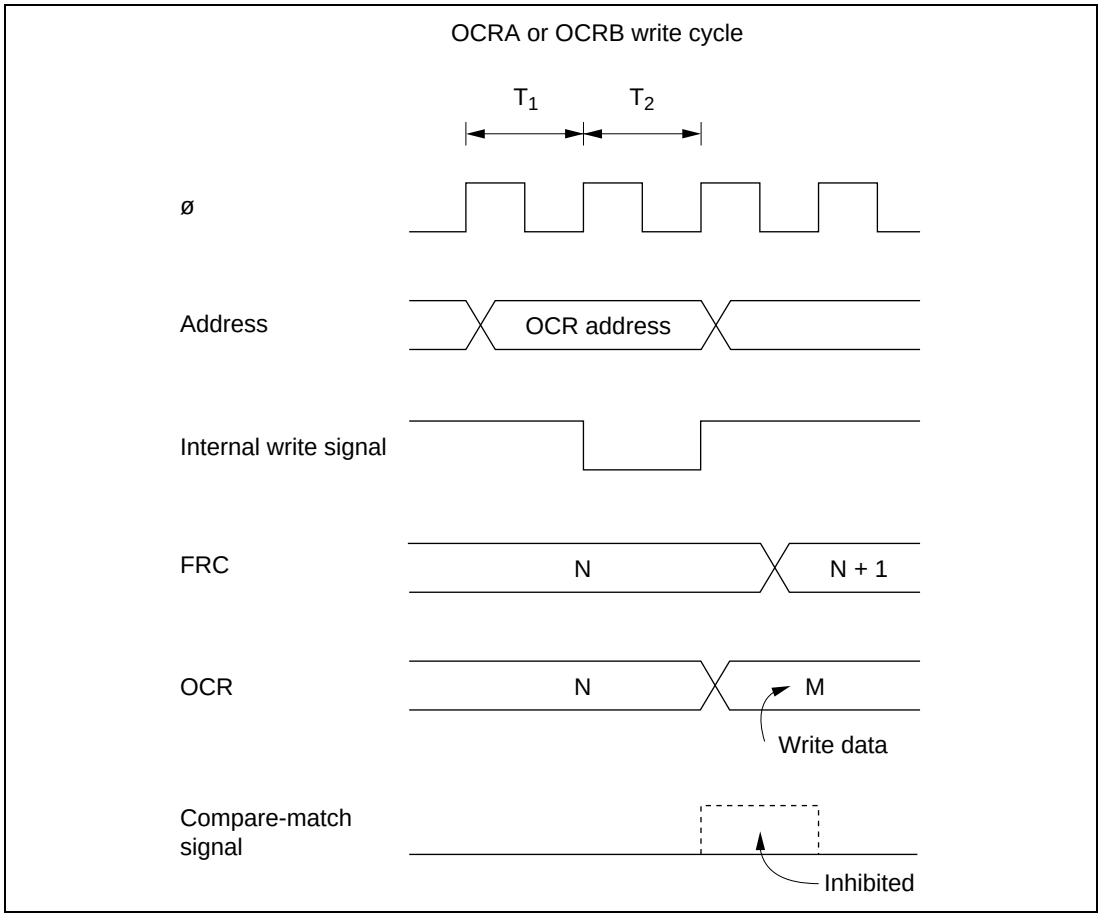


Figure 11.20 Contention between OCR Write and Compare-Match (When Automatic Addition Function Is Not Used)

Switching of Internal Clock and FRC Operation: When the internal clock is changed, the changeover may cause FRC to increment. This depends on the time at which the clock select bits (CKS1 and CKS0) are rewritten, as shown in table 11.5.

When an internal clock is used, the FRC clock is generated on detection of the falling edge of the internal clock scaled from the system clock (ø). If the clock is changed when the old source is high and the new source is low, as in case no. 3 in table 11.5, the changeover is regarded as a falling edge that triggers the FRC increment clock pulse.

Switching between an internal and external clock can also cause FRC to increment.

Table 11.5 Switching of Internal Clock and FRC Operation

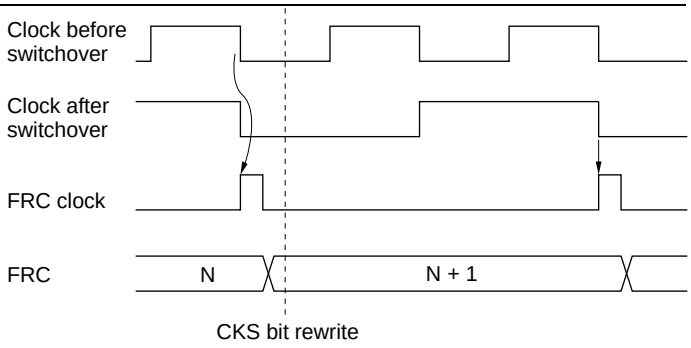
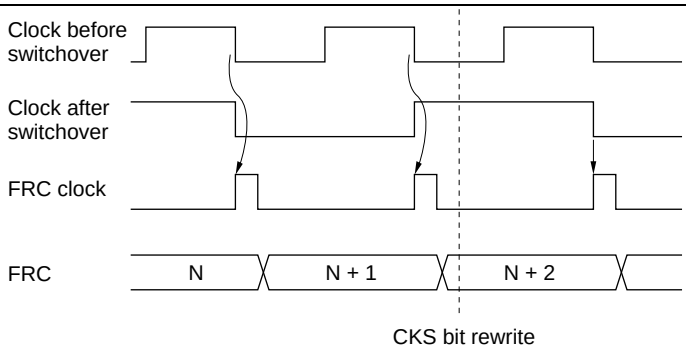
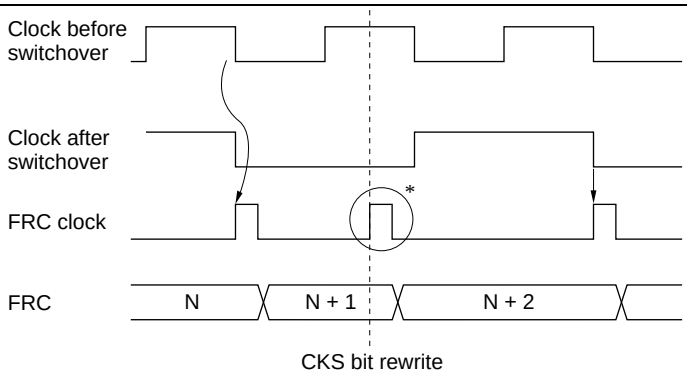
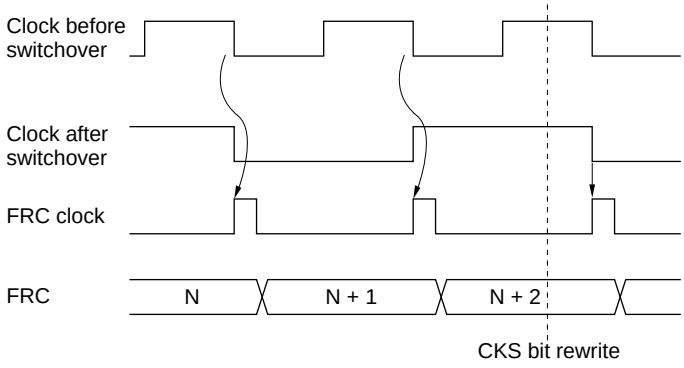
No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	FRC Operation
1	Switching from low to low	
2	Switching from low to high	

Table 11.5 Switching of Internal Clock and FRC Operation (cont)

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	FRC Operation
3	Switching from high to low	
4	Switching from high to high	

Note: *Generated on the assumption that the switchover is a falling edge; FRC is incremented.

Section 12 8-Bit Timers

12.1 Overview

The H8S/2128 Series and H8S/2124 Series include an 8-bit timer module with two channels (TMR0 and TMR1). Each channel has an 8-bit counter (TCNT) and two time constant registers (TCORA and TCORB) that are constantly compared with the TCNT value to detect compare-matches. The 8-bit timer module can be used as a multifunction timer in a variety of applications, such as generation of a rectangular-wave output with an arbitrary duty cycle.

The H8S/2128 Series also has two similar 8-bit timer channels (TMRX and TMRY), and the H8S/2124 Series has one (TMRY). These channels can be used in a connected configuration using the timer connection function. TMRX and TMRY have greater input/output and interrupt function related restrictions than TMR0 and TMR1.

12.1.1 Features

- Selection of clock sources
  - TMR0, TMR1: The counter input clock can be selected from six internal clocks and an external clock (enabling use as an external event counter).
  - TMRX, TMRY: The counter input clock can be selected from three internal clocks and an external clock (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 signal.
- Timer output controlled by two compare-match signals
  - The timer output signal in each channel is controlled by two independent compare-match signals, enabling the timer to be used for various applications, such as the generation of pulse output or PWM output with an arbitrary duty cycle.
    (Note: TMRY does not have a timer output pin.)
- Cascading of the two channels (TMR0, TMR1)
  - Operation as a 16-bit timer can be performed using channel 0 as the upper half and channel 1 as the lower half (16-bit count mode).
  - Channel 1 can be used to count channel 0 compare-match occurrences (compare-match count mode).
- Multiple interrupt sources for each channel
  - TMR0, TMR1, TMRY: Two compare-match interrupts and one overflow interrupt can be requested independently.
  - TMRX: One input capture source is available.

12.1.2 Block Diagram

Figure 12.1 shows a block diagram of the 8-bit timer module (TMR0 and TMR1).

TMRX and TMRY have a similar configuration, but cannot be cascaded. TMRX also has an input capture function. For details, see section 13, Timer Connection.

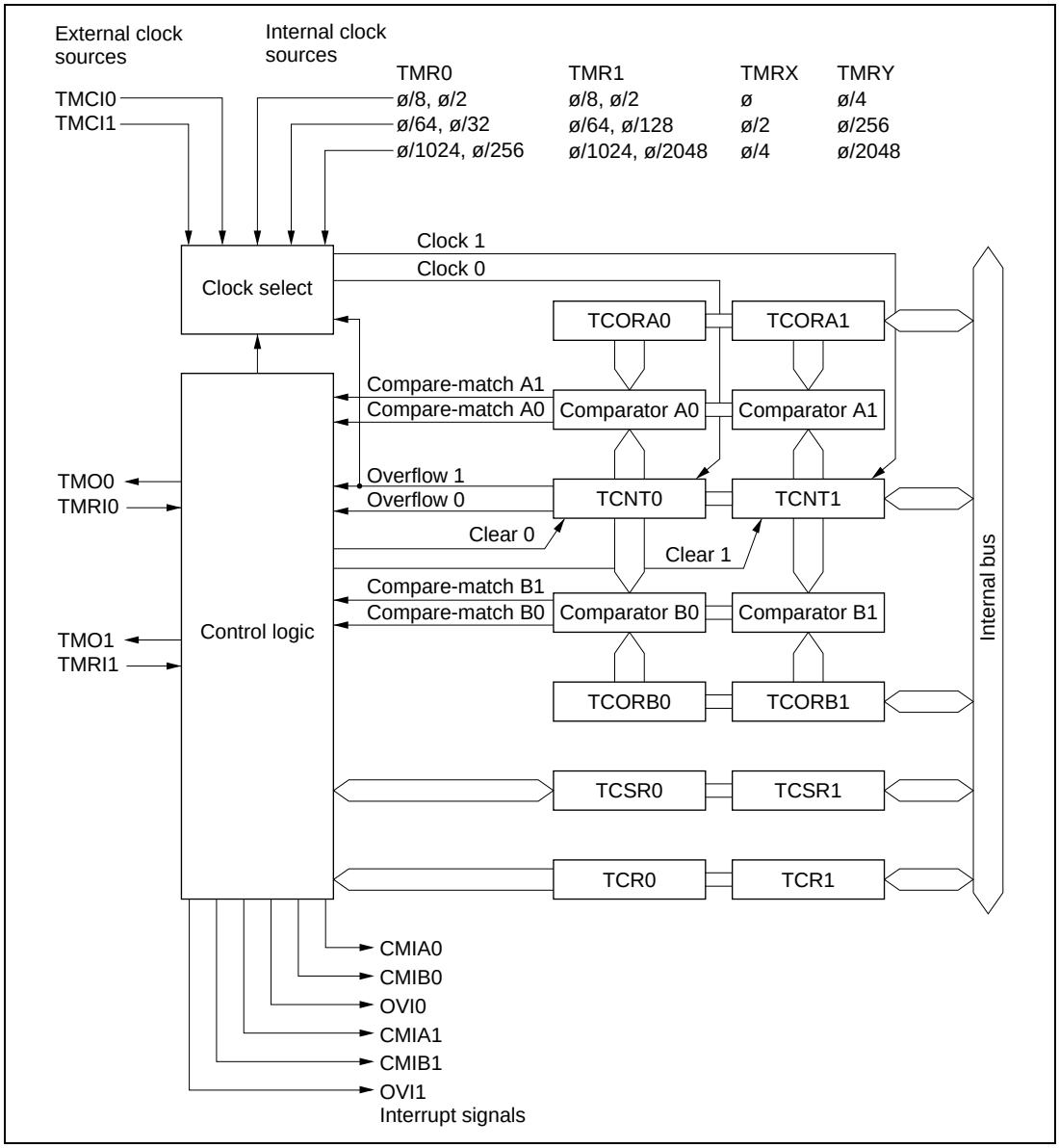


Figure 12.1 Block Diagram of 8-Bit Timer Module

12.1.3 Pin Configuration

Table 12.1 summarizes the input and output pins of the 8-bit timer module.

Table 12.1 8-Bit Timer Input and Output Pins

Channel	Name	Symbol*	I/O	Function
0	Timer output	TMO0	Output	Output controlled by compare-match
	Timer clock input	TMCI0	Input	External clock input for the counter
	Timer reset input	TMRI0	Input	External reset input for the counter
1	Timer output	TMO1	Output	Output controlled by compare-match
	Timer clock input	TMCI1	Input	External clock input for the counter
	Timer reset input	TMRI1	Input	External reset input for the counter
X	Timer output	TMOX	Output	Output controlled by compare-match
	Timer clock/ reset input	HFBACKI/TMIX (TMCIX/TMRIX)	Input	External clock/reset input for the counter
Y	Timer clock/reset input	VSYNCI/TMIY (TMCIY/TMRIY)	Input	External clock/reset input for the counter

Note: * The abbreviations TMO, TMCI, and TMRI are used in the text, omitting the channel number.

Channel X and Y I/O pins have the same internal configuration as channels 0 and 1, and therefore the same abbreviations are used.

12.1.4 Register Configuration

Table 12.2 summarizes the registers of the 8-bit timer module.

Table 12.2 8-Bit Timer Registers

Channel	Name	Abbreviation*[3]	R/W	Initial value	Address*[1]
0	Timer control register 0	TCR0	R/W	H'00	H'FFC8
	Timer control/status register 0	TCSR0	R/(W)*[2]	H'00	H'FFCA
	Time constant register A0	TCORA0	R/W	H'FF	H'FFCC
	Time constant register B0	TCORB0	R/W	H'FF	H'FFCE
	Time counter 0	TCNT0	R/W	H'00	H'FFD0
1	Timer control register 1	TCR1	R/W	H'00	H'FFC9
	Timer control/status register 1	TCSR1	R/(W)*[2]	H'10	H'FFCB
	Time constant register A1	TCORA1	R/W	H'FF	H'FFCD
	Time constant register B1	TCORB1	R/W	H'FF	H'FFCF
	Timer counter 1	TCNT1	R/W	H'00	H'FFD1
Common	Serial/timer control register	STCR	R/W	H'00	H'FFC3
	Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
		MSTPCRL	R/W	H'FF	H'FF87
	Timer connection register S	TCONRS	R/W	H'00	H'FFFE
X	Timer control register X	TCRX	R/W	H'00	H'FFF0
	Timer control/status register X	TCSRX	R/(W)*[2]	H'00	H'FFF1
	Time constant register AX	TCORAX	R/W	H'FF	H'FFF6
	Time constant register BX	TCORBX	R/W	H'FF	H'FFF7
	Timer counter X	TCNTX	R/W	H'00	H'FFF4
	Time constant register C	TCORC	R/W	H'FF	H'FFF5
	Input capture register R	TICRR	R	H'00	H'FFF2
	Input capture register F	TICRF	R	H'00	H'FFF3
Y	Timer control register Y	TCRY	R/W	H'00	H'FFF0
	Timer control/status register Y	TCSRY	R/(W)*[2]	H'00	H'FFF1
	Time constant register AY	TCORAY	R/W	H'FF	H'FFF2
	Time constant register BY	TCORBY	R/W	H'FF	H'FFF3
	Timer counter Y	TCNTY	R/W	H'00	H'FFF4
	Timer input select register	TISR	R/W	H'FE	H'FFF5

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written in bits 7 to 5, to clear these flags.
3. The abbreviations TCR, TCSR, TCORA, TCORB, and TCNT are used in the text, omitting the channel designation (0, 1, X, or Y).

Each pair of registers for channel 0 and channel 1 comprises a 16-bit register with the upper 8 bits for channel 0 and the lower 8 bits for channel 1, so they can be accessed together by word access. (Access is not divided into two 8-bit accesses.)

Certain of the channel X and channel Y registers are assigned to the same address. The TMRX/Y bit in TCONRS determines which register is accessed.

12.2 Register Descriptions

12.2.1 Timer Counter (TCNT)

	TCNT0								TCNT1							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
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	R/W	R/W

TCNTX,TCNTY

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Each TCNT is an 8-bit readable/writable up-counter.

TCNT0 and TCNT1 comprise a single 16-bit register, so they can be accessed together by word access.

TCNT increments on pulses generated from an internal or external clock source. This clock source is selected by clock select bits CKS2 to CKS0 in TCR.

TCNT can be cleared by an external reset input signal or compare-match signal. Counter clear bits CCLR1 and CCLR0 in TCR select the method of clearing.

When TCNT overflows from H'FF to H'00, the overflow flag (OVF) in TCSR is set to 1.

The timer counters are initialized to H'00 by a reset and in hardware standby mode.

12.2.2 Time Constant Register A (TCORA)

	TCORA0								TCORA1							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
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	R/W	R/W

TCORAX, TCORAY

Bit	7	6	5	4	3	2	1	0
Initial value	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

TCORA is an 8-bit readable/writable register.

TCORA0 and TCORA1 comprise a single 16-bit register, so they can be accessed together by word access.

TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag A (CMFA) in TCSR is set. Note, however, that comparison is disabled during the T2 state of a TCORA write cycle.

The timer output can be freely controlled by these compare-match signals and the settings of output select bits OS1 and OS0 in TCSR.

TCORA is initialized to H'FF by a reset and in hardware standby mode.

12.2.3 Time Constant Register B (TCORB)

	TCORB0								TCORB1							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
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	R/W	R/W

TCORBX, TCORBY

Bit	7	6	5	4	3	2	1	0
Initial value	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

TCORB is an 8-bit readable/writable register. TCORB0 and TCORB1 comprise a single 16-bit register, so they can be accessed together by word access.

TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag B (CMFB) in TCSR is set. Note, however, that comparison is disabled during the T2 state of a TCORB write cycle.

The timer output can be freely controlled by these compare-match signals and the settings of output select bits OS3 and OS2 in TCSR.

TCORB is initialized to H'FF by a reset and in hardware standby mode.

12.2.4 Timer Control Register (TCR)

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCR is an 8-bit readable/writable register that selects the clock source and the time at which TCNT is cleared, and enables interrupts.

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

For details of the timing, see section 12.3, Operation.

Bit 7—Compare-Match Interrupt Enable B (CMIEB): Selects whether the CMFB interrupt request (CMIB) is enabled or disabled when the CMFB flag in TCSR is set to 1.

Note that a CMIB interrupt is not requested by TMRX, regardless of the CMIEB value.

Bit 7

CMIEB	Description	
0	CMFB interrupt request (CMIB) is disabled	(Initial value)
1	CMFB interrupt request (CMIB) is enabled	

Bit 6—Compare-Match Interrupt Enable A (CMIEA): Selects whether the CMFA interrupt request (CMIA) is enabled or disabled when the CMFA flag in TCSR is set to 1.

Note that a CMIA interrupt is not requested by TMRX, regardless of the CMIEA value.

Bit 6

CMIEA	Description	
0	CMFA interrupt request (CMIA) is disabled	(Initial value)
1	CMFA interrupt request (CMIA) is enabled	

Bit 5—Timer Overflow Interrupt Enable (OVIE): Selects whether the OVF interrupt request (OVI) is enabled or disabled when the OVF flag in TCSR is set to 1.

Note that an OVI interrupt is not requested by TMRX, regardless of the OVIE value.

Bit 5		
OVIE	**Description**	
0	OVF interrupt request (OVI) is disabled	(Initial value)
1	OVF interrupt request (OVI) is enabled	

Bits 4 and 3—Counter Clear 1 and 0 (CCLR1, CCLR0): These bits select the method by which the timer counter is cleared: by compare-match A or B, or by an external reset input.

Bit 4	Bit 3		
CCLR1	**CCLR0**	**Description**	
0	0	Clearing is disabled	(Initial value)
	1	Cleared on compare-match A	
1	0	Cleared on compare-match B	
	1	Cleared on rising edge of external reset input	

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select whether the clock input to TCNT is an internal or external clock.

The input clock can be selected from either six or three clocks, all divided from the system clock (ø). The falling edge of the selected internal clock triggers the count.

When use of an external clock is selected, three types of count can be selected: at the rising edge, the falling edge, and both rising and falling edges.

Some functions differ between channel 0 and channel 1, because of the cascading function.

	TCR			STCR		
	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
Channel	CKS2	CKS1	CKS0	ICKS1	ICKS0	Description
0	0	0	0	—	—	Clock input disabled (Initial value)
	0	0	1	—	0	ø/8 internal clock source, counted on the falling edge
	0	0	1	—	1	ø/2 internal clock source, counted on the falling edge
	0	1	0	—	0	ø/64 internal clock source, counted on the falling edge
	0	1	0	—	1	ø/32 internal clock source, counted on the falling edge
	0	1	1	—	0	ø/1024 internal clock source, counted on the falling edge
	0	1	1	—	1	ø/256 internal clock source, counted on the falling edge
	1	0	0	—	—	Counted on TCNT1 overflow signal*
1	0	0	0	—	—	Clock input disabled (Initial value)
	0	0	1	0	—	ø/8 internal clock source, counted on the falling edge
	0	0	1	1	—	ø/2 internal clock source, counted on the falling edge
	0	1	0	0	—	ø/64 internal clock source, counted on the falling edge
	0	1	0	1	—	ø/128 internal clock source, counted on the falling edge
	0	1	1	0	—	ø/1024 internal clock source, counted on the falling edge
	0	1	1	1	—	ø/2048 internal clock source, counted on the falling edge
	1	0	0	—	—	Counted on TCNT0 compare-match A*

	TCR			STCR			
	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0		
Channel	CKS2	CKS1	CKS0	ICKS1	ICKS0	Description	
X	0	0	0	—	—	Clock input disabled	(Initial value)
	0	0	1	—	—	Counted on ø internal clock source	
	0	1	0	—	—	ø/2 internal clock source, counted on the falling edge	
	0	1	1	—	—	ø/4 internal clock source, counted on the falling edge	
	1	0	0	—	—	Clock input disabled	
Y	0	0	0	—	—	Clock input disabled	(Initial value)
	0	0	1	—	—	ø/4 internal clock source, counted on the falling edge	
	0	1	0	—	—	ø/256 internal clock source, counted on the falling edge	
	0	1	1	—	—	ø/2048 internal clock source, counted on the falling edge	
	1	0	0	—	—	Clock input disabled	
Common	1	0	1	—	—	External clock source, counted at rising edge	
	1	1	0	—	—	External clock source, counted at falling edge	
	1	1	1	—	—	External clock source, counted at both rising and falling edges	

Note: * If the count input of channel 0 is the TCNT1 overflow signal and that of channel 1 is the TCNT0 compare-match signal, no incrementing clock will be generated. Do not use this setting.

12.2.5 Timer Control/Status Register (TCSR)

TCSR0

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

TCSR1

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
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/W

TCSRX

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W

TCSRY

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written in bits 7 to 5, and in bit 4 in TCSRX, to clear these flags.

TCSR is an 8-bit register that indicates compare-match and overflow statuses (and input capture status in TMRX only), and controls compare-match output.

TCSR0, TCSRX, and TCSRY are initialized to H'00, and TCSR1 is initialized to H'10, by a reset and in hardware standby mode.

Bit 7—Compare-Match Flag B (CMFB): Status flag indicating whether the values of TCNT and TCORB match.

Bit 7

CMFB	Description	
0	[Clearing conditions]	(Initial value)
	• Read CMFB when CMFB = 1, then write 0 in CMFB	
	• When the DTC is activated by a CMIB interrupt	
1	[Setting condition]	
	When TCNT = TCORB	

Bit 6—Compare-match Flag A (CMFA): Status flag indicating whether the values of TCNT and TCORA match.

Bit 6

CMFA	Description	
0	[Clearing conditions]	(Initial value)
	• Read CMFA when CMFA = 1, then write 0 in CMFA	
	• When the DTC is activated by a CMIA interrupt	
1	[Setting condition]	
	When TCNT = TCORA	

Bit 5 —Timer Overflow Flag (OVF): Status flag indicating that TCNT has overflowed (changed from H'FF to H'00).

Bit 5

OVF	Description	
0	[Clearing condition]	(Initial value)
	Read OVF when OVF = 1, then write 0 in OVF	
1	[Setting condition]	
	When TCNT overflows from H'FF to H'00	

TCSR0

Bit 4—A/D Trigger Enable (ADTE): Enables or disables A/D converter start requests by compare-match A.

Bit 4		
ADTE	**Description**	
0	A/D converter start requests by compare-match A are disabled	(Initial value)
1	A/D converter start requests by compare-match A are enabled	

TCSR1

Bit 4—Reserved: This bit cannot be modified and is always read as 1.

TCSRX

Bit 4—Input Capture Flag (ICF): Status flag that indicates detection of a rising edge followed by a falling edge in the external reset signal after the ICST bit in TCONRI has been set to 1.

Bit 4		
ICF	**Description**	
0	[Clearing condition] Read ICF when ICF = 1, then write 0 in ICF	(Initial value)
1	[Setting condition] When a rising edge followed by a falling edge is detected in the external reset signal after the ICST bit in TCONRI has been set to 1	

TCSRY

Bit 4—Input Capture Interrupt Enable (ICIE): Selects enabling or disabling of the interrupt request by ICF (ICIX) when the ICF bit in TCSRX is set to 1.

Bit 4		
ICIE	**Description**	
0	Interrupt request by ICF (ICIX) is disabled	(Initial value)
1	Interrupt request by ICF (ICIX) is enabled	

Bits 3 to 0—Output Select 3 to 0 (OS3 to OS0): These bits specify how the timer output level is to be changed by a compare-match of TCOR and TCNT.

OS3 and OS2 select the effect of compare-match B on the output level, OS1 and OS0 select the effect of compare-match A on the output level, and both of them can be controlled independently.

Note, however, that priorities are set such that: trigger output > 1 output > 0 output. If compare-matches occur simultaneously, the output changes according to the compare-match with the higher 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 occurs.

Bit 3	Bit 2		
OS3	**OS2**	**Description**	
0	0	No change when compare-match B occurs	(Initial value)
	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	(Initial value)
	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)	

12.2.6 Serial/Timer Control Register (STCR)

Bit	7	6	5	4	3	2	1	0
	IICS	IICX1	IICX0	IICE	FLSHE	—	ICKS1	ICKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

STCR is an 8-bit readable/writable register that controls register access, the IIC operating mode (when the on-chip IIC option is included), and on-chip flash memory (in F-ZTAT versions), and also selects the TCNT input clock.

For details on functions not related to the 8-bit timers, see section 3.2.4, Serial/Timer Control Register (STCR), and the descriptions of the relevant modules. If a module controlled by STCR is not used, do not write 1 to the corresponding bit.

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

Bits 7 to 4—I^{2}C Control (IICS, IICX1, IICX0, IICE): These bits control the operation of the I^{2}C bus interface when the IIC option is included on-chip. See section 16, I^{2}C Bus Interface, for details.

Bit 3—Flash Memory Control Register Enable (FLSHE): Controls the operation of the flash memory in F-ZTAT versions. See section 19, ROM, for details.

Bit 2—Reserved: Do not write 1 to this bit.

Bits 1 and 0—Internal Clock Select 1 and 0 (ICKS1, ICKS0): These bits, together with bits CKS2 to CKS0 in TCR, select the clock to be input to TCNT. For details, see section 12.2.4, Timer Control Register.

12.2.7 System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
	CS2E	IOSE	INTM1	INTM0	XRST	NMIEG	HIE	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R	R/W	R	R/W	R/W	R/W

Only bit 1 is described here. For details on functions not related to the 8-bit timers, see sections 3.2.2 and 5.2.1, System Control Register (SYSCR), and the descriptions of the relevant modules.

Bit 1—Host Interface Enable (HIE): Controls CPU access to 8-bit timer (channel X and Y) data registers and control registers, and timer connection control registers.

Bit 1

HIE	Description
0	CPU access to 8-bit timer (channel X and Y) data registers and control (Initial value) registers, and timer connection control registers, is enabled
1	CPU access to 8-bit timer (channel X and Y) data registers and control registers, and timer connection control registers, is disabled

12.2.8 Timer Connection Register S (TCONRS)

Bit	7	6	5	4	3	2	1	0
	TMRX/Y	ISGENE	HOMOD1	HOMOD0	VOMOD1	VOMOD0	CLMOD1	CLMOD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCONRS is an 8-bit readable/writable register that controls access to the TMRX and TMRY registers and timer connection operation.

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

Bit 7—TMRX/TMRY Access Select (TMRX/Y): The TMRX and TMRY registers can only be accessed when the HIE bit in SYSCR is cleared to 0. In the H8S/2128 Series, some of the TMRX registers and the TMRY registers are assigned to the same memory space addresses (H'FFF0 to H'FFF5), and the TMRX/Y bit determines which registers are accessed. In the H8S/2124 Series, there is no control of TMRY register access by this bit.

Bit 7	**Accessible Registers**							
TMRX/Y	**H'FFF0**	**H'FFF1**	**H'FFF2**	**H'FFF3**	**H'FFF4**	**H'FFF5**	**H'FFF6**	**H'FFF7**
0 (Initial value)	TMRX	TMRX	TMRX	TMRX	TMRX	TMRX	TMRX	TMRX
	TCRX	TCSRX	TICRR	TICRF	TCNTX	TCORC	TCORAX	TCORBX
1	TMRY	TMRY	TMRY	TMRY	TMRY	TMRY		
	TCRY	TCSRY	TCORAY	TCORBY	TCNTY	TISR		

12.2.9 Input Capture Register (TICR) [TMRX Additional Function]

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	—	—	—

TICR is an 8-bit internal register to which the contents of TCNT are transferred on the falling edge of external reset input. The CPU cannot read or write to TICR directly.

The TICR function is used in timer connection. For details, see section 13, Timer Connection.

12.2.10 Time Constant Register C (TCORC) [TMRX Additional Function]

Bit	7	6	5	4	3	2	1	0
Initial value	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

TCORC is an 8-bit readable/writable register. The sum of the contents of TCORC and TICR is continually compared with the value in TCNT. When a match is detected, a compare-match C signal is generated. Note, however, that comparison is disabled during the T2 state of a TCORC write cycle and a TICR input capture cycle.

TCORC is initialized to H'FF by a reset and in hardware standby mode.

The TCORC function is used in timer connection. For details, see section 13, Timer Connection.

12.2.11 Input Capture Registers R and F (TICRR, TICRF) [TMRX Additional Functions]

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TICRR and TICRF are 8-bit read-only registers. When the ICST bit in TCONRI is set to 1, TICRR and TICRF capture the contents of TCNT successively on the rise and fall of the external reset input. When one capture operation ends, the ICST bit is cleared to 0.

TICRR and TICRF are each initialized to H'00 by a reset and in hardware standby mode.

The TICRR and TICRF functions are used in timer connection. For details, see section 13, Timer Connection.

12.2.12 Timer Input Select Register (TISR) [TMRY Additional Function]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	IS
Initial value	1	1	1	1	1	1	1	0
Read/Write	—	—	—	—	—	—	—	R/W

TISR is an 8-bit readable/writable register that selects the external clock/reset signal source for the counter.

TISR is initialized to H'FE by a reset and in hardware standby mode.

Bits 7 to 1—Reserved: Do not write 0 to these bits.

Bit 0—Input Select (IS): Selects the internal synchronization signal (IVG signal) or the timer clock/reset input pin (VSYNCI/TMIY (TMCIY/TMRIY)) as the external clock/reset signal source for the counter.

Bit 0

IS	Description	
0	IVG signal is selected (H8S/2128 Series) External clock/reset input is disabled (H8S/2124 Series)	(Initial value)
1	VSYNCI/TMIY (TMCIY/TMRIY) is selected	

12.2.13 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	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	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers, and is used to perform module stop mode control.

When the MSTP12 bit or MSTP8 bit is set to 1, 8-bit timer operation is halted on channels 0 and 1 or channels X and Y, respectively, and a transition is made to module stop mode. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 4—Module Stop (MSTP12): Specifies 8-bit timer (channel 0/1) module stop mode.

MSTPCRH Bit 4		
MSTP12	**Description**	
0	8-bit timer (channel 0/1) module stop mode is cleared	
1	8-bit timer (channel 0/1) module stop mode is set	(Initial value)

MSTPCRH Bit 0—Module Stop (MSTP8): Specifies 8-bit timer (channel X/Y) and timer connection module stop mode.

MSTPCRH Bit 0		
MSTP8	**Description**	
0	8-bit timer (channel X/Y) and timer connection module stop mode is cleared	
1	8-bit timer (channel X/Y) and timer connection module stop mode is set	(Initial value)

12.3 Operation

12.3.1 TCNT Incrementation Timing

TCNT is incremented by input clock pulses (either internal or external).

Internal Clock: An internal clock created by dividing the system clock (ø) can be selected by setting bits CKS2 to CKS0 in TCR. Figure 12.2 shows the count timing.

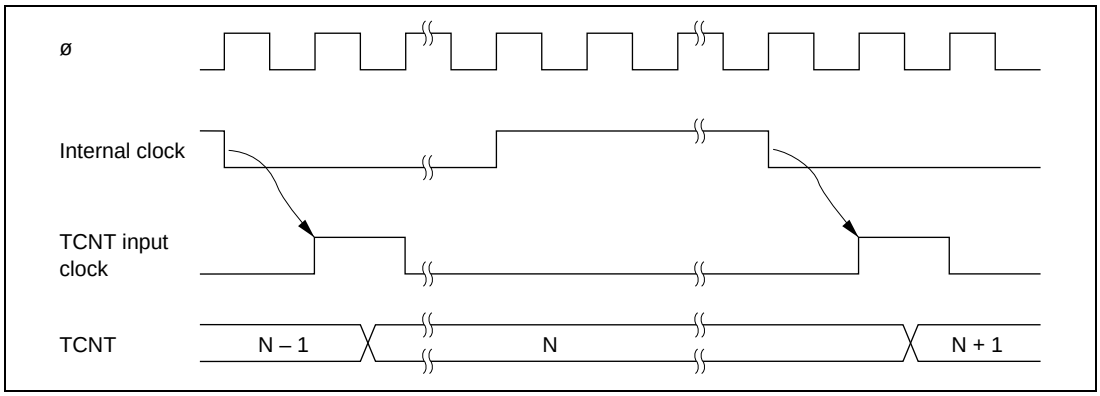


Figure 12.2 Count Timing for Internal Clock Input

External Clock: Three incrementation methods can be selected by setting bits CKS2 to CKS0 in 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 a single edge, and at least 2.5 states for incrementation at both edges. The counter will not increment correctly if the pulse width is less than these values.

Figure 12.3 shows the timing of incrementation at both edges of an external clock signal.

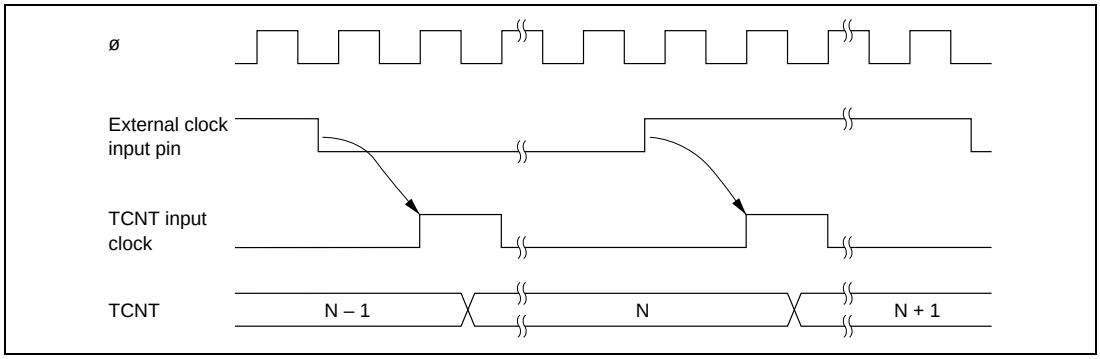


Figure 12.3 Count Timing for External Clock Input

12.3.2 Compare-Match Timing

Setting of Compare-Match Flags A and B (CMFA, CMFB): The CMFA and CMFB flags in TCSR are set to 1 by a compare-match signal generated when the TCOR and TCNT values match. The compare-match signal is generated at the last state in which the match is true, just before the timer counter is updated.

Therefore, when TCOR and TCNT match, the compare-match signal is not generated until the next incrementation clock input. Figure 12.4 shows this timing.

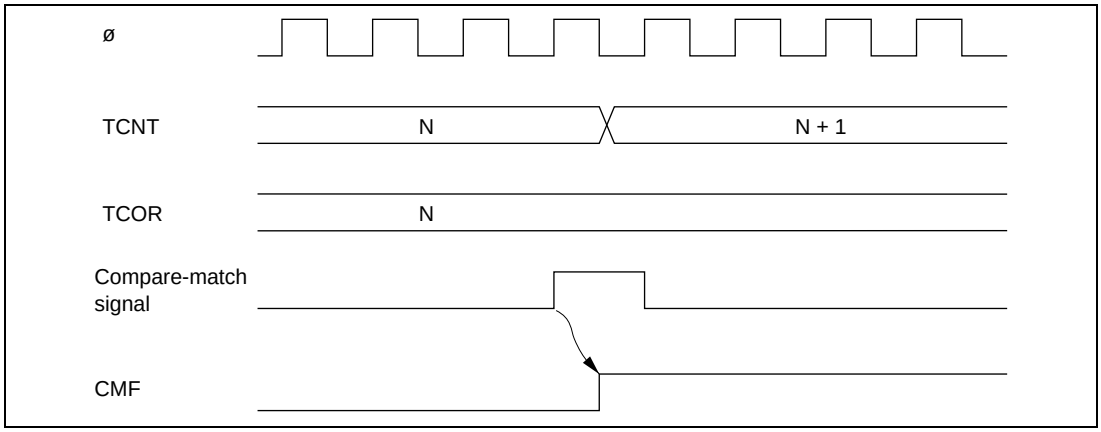


Figure 12.4 Timing of CMF Setting

Timer Output Timing: When compare-match A or B occurs, the timer output changes as specified by the output select bits (OS3 to OS0) in TCSR. Depending on these bits, the output can remain the same, be set to 0, be set to 1, or toggle.

Figure 12.5 shows the timing when the output is set to toggle at compare-match A.

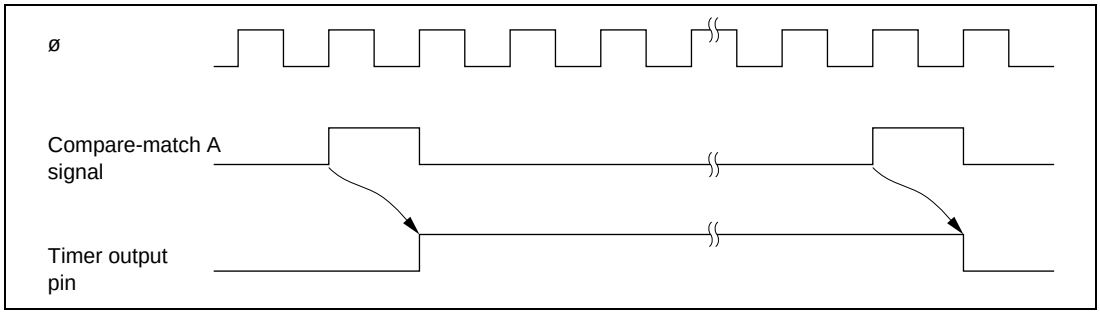


Figure 12.5 Timing of Timer Output

Timing of Compare-Match Clear: TCNT is cleared when compare-match A or B occurs, depending on the setting of the CCLR1 and CCLR0 bits in TCR. Figure 12.6 shows the timing of this operation.

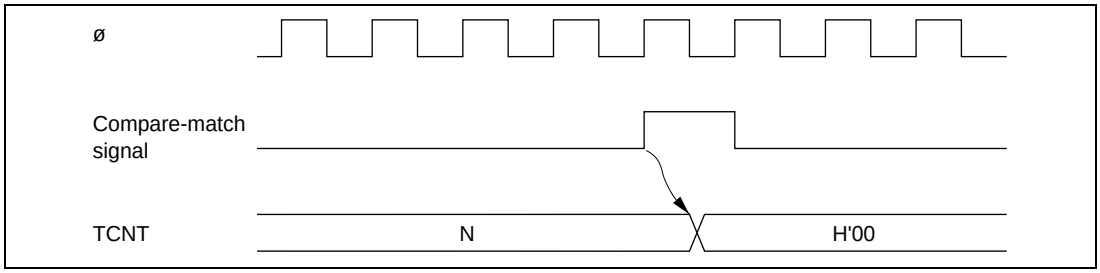


Figure 12.6 Timing of Compare-Match Clear

12.3.3 TCNT External Reset Timing

TCNT is cleared at the rising edge of an external reset input, depending on the settings of the CCLR1 and CCLR0 bits in TCR. The width of the clearing pulse must be at least 1.5 states. Figure 12.7 shows the timing of this operation.

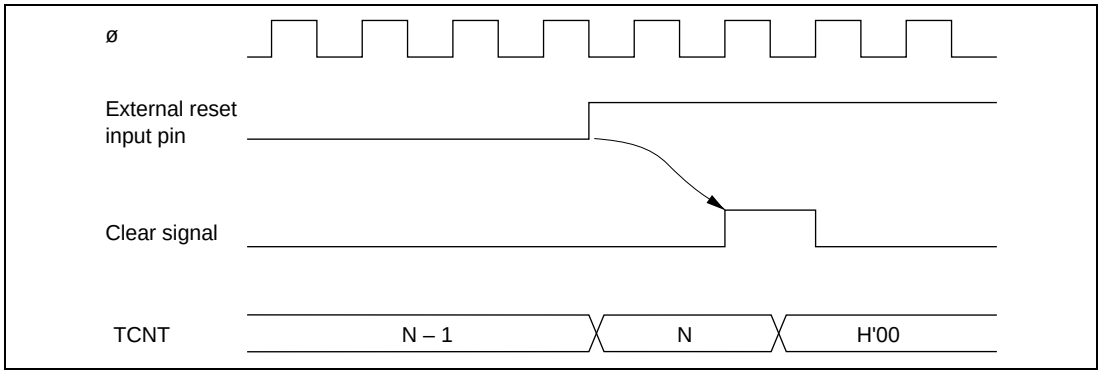


Figure 12.7 Timing of Clearing by External Reset Input

12.3.4 Timing of Overflow Flag (OVF) Setting

OVF in TCSR is set to 1 when the timer count overflows (changes from H'FF to H'00). Figure 12.8 shows the timing of this operation.

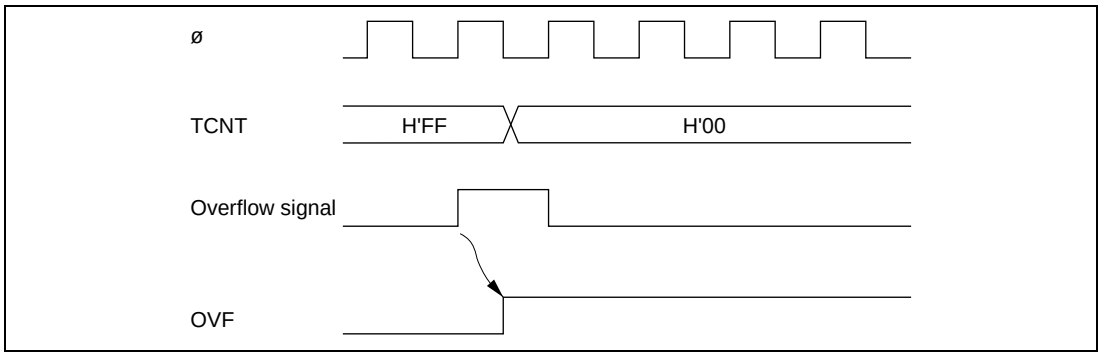


Figure 12.8 Timing of OVF Setting

12.3.5 Operation with Cascaded Connection

If bits CKS2 to CKS0 in either TCR0 or TCR1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, a single 16-bit timer can be used (16-bit timer mode) or compare-matches of 8-bit channel 0 can be counted by the timer of channel 1 (compare-match count mode). In this case, the timer operates as described below.

16-Bit Count Mode: When bits CKS2 to CKS0 in TCR0 are set to B'100, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

- Setting of compare-match flags
  - — The CMF flag in TCSR0 is set to 1 when a 16-bit compare-match occurs.
  - — The CMF flag in TCSR1 is set to 1 when a lower 8-bit compare-match occurs.
- Counter clear specification
  - — If the CCLR1 and CCLR0 bits in TCR0 have been set for counter clear at compare-match, the 16-bit counter (TCNT0 and TCNT1 together) is cleared when a 16-bit compare-match occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even if counter clear by the TMRI0 pin has also been set.
  - — The settings of the CCLR1 and CCLR0 bits in TCR1 are ignored. The lower 8 bits cannot be cleared independently.
- Pin output
  - — Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR0 is in accordance with the 16-bit compare-match conditions.
  - — Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR1 is in accordance with the lower 8-bit compare-match conditions.

Compare-Match Count Mode: When bits CKS2 to CKS0 in TCR1 are B'100, TCNT1 counts compare-match A’s for channel 0.

Channels 0 and 1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clearing are in accordance with the settings for each channel.

Usage Note: If the 16-bit count mode and compare-match count mode are set simultaneously, the input clock pulses for TCNT0 and TCNT1 are not generated and thus the counters will stop operating. Simultaneous setting of these two modes should therefore be avoided.

12.4 Interrupt Sources

The TMR0, TMR1, and TMRY 8-bit timers can generate three types of interrupt: compare-match A and B (CMIA and CMIB), and overflow (OVI). TMRX can generate only an ICIX interrupt. An interrupt is requested when the corresponding interrupt enable bit is set in TCR or TCSR. Independent signals are sent to the interrupt controller for each interrupt. It is also possible to activate the DTC by means of CMIA and CMIB interrupts from TMR0, TMR1 and TMRY.

An overview of 8-bit timer interrupt sources is given in tables 12.3 to 12.5.

Table 12.3 TMR0 and TMR1 8-Bit Timer Interrupt Sources

Interrupt source	Description	DTC Activation	Interrupt Priority
CMIA	Requested by CMFA	Possible	High
CMIB	Requested by CMFB	Possible	
OVI	Requested by OVF	Not possible	Low

Table 12.4 TMRX 8-Bit Timer Interrupt Source

Interrupt source	Description	DTC Activation
ICIX	Requested by ICF	Not possible

Table 12.5 TMRY 8-Bit Timer Interrupt Sources

Interrupt source	Description	DTC Activation	Interrupt Priority
CMIA	Requested by CMFA	Possible	High
CMIB	Requested by CMFB	Possible	
OVI	Requested by OVF	Not possible	Low

12.5 8-Bit Timer Application Example

In the example below, the 8-bit timer is used to generate a pulse output with a selected duty cycle, as shown in figure 12.9. The control bits are set as follows:

- In TCR, CCLR1 is cleared to 0 and CCLR0 is set to 1 so that the timer counter is cleared by a TCORA compare-match.
- In TCSR, bits OS3 to OS0 are set to B'0110, causing 1 output at a TCORA compare-match and 0 output at a TCORB compare-match.

With these settings, the 8-bit timer provides output of pulses at a rate determined by TCORA with a pulse width determined by TCORB. No software intervention is required.

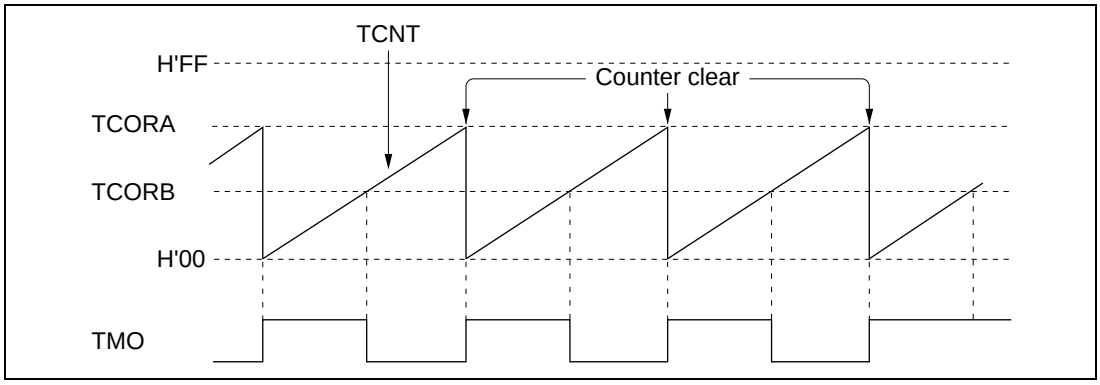


Figure 12.9 Pulse Output (Example)

12.6 Usage Notes

Application programmers should note that the following kinds of contention can occur in the 8-bit timer module.

12.6.1 Contention between TCNT Write and Clear

If a timer counter clock pulse is generated during the T2 state of a TCNT write cycle, the clear takes priority, so that the counter is cleared and the write is not performed. Figure 12.10 shows this operation.

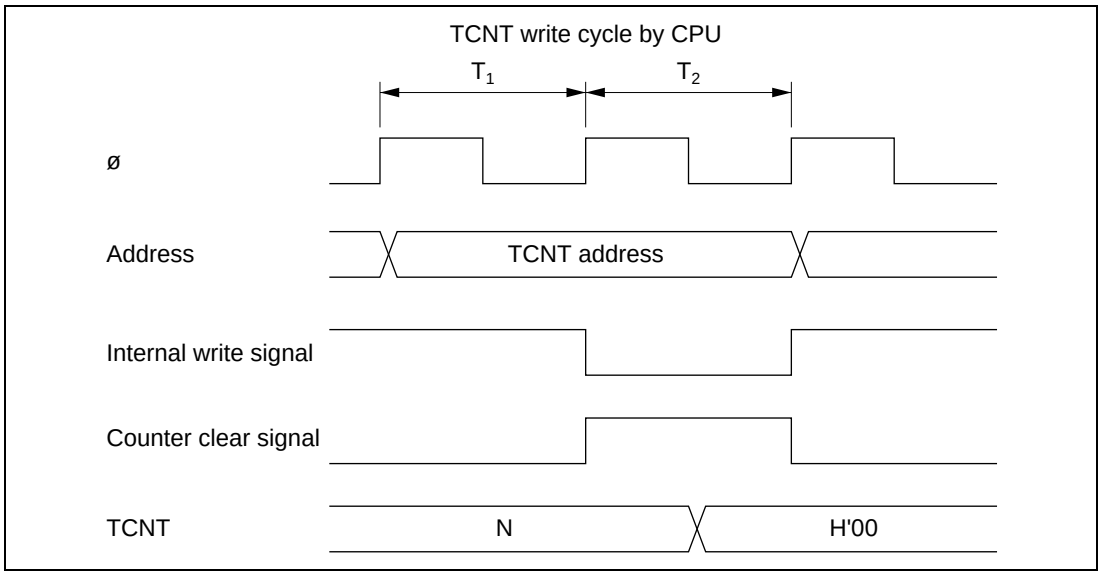


Figure 12.10 Contention between TCNT Write and Clear

12.6.2 Contention between TCNT Write and Increment

If a timer counter clock pulse is generated during the T2 state of a TCNT write cycle, the write takes priority and the counter is not incremented. Figure 12.11 shows this operation.

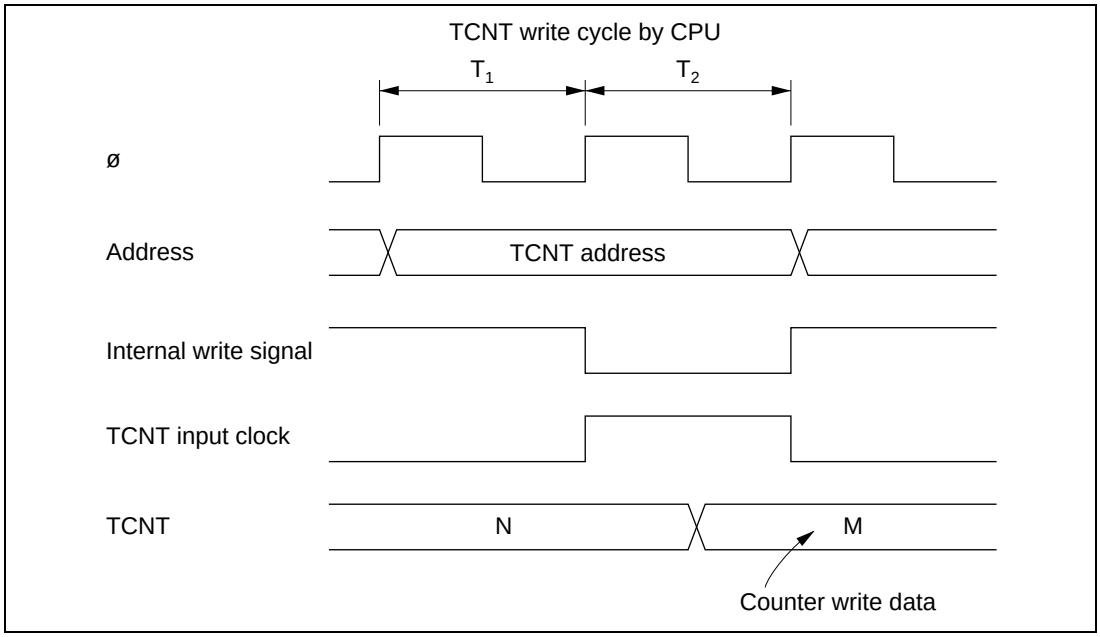


Figure 12.11 Contention between TCNT Write and Increment

12.6.3 Contention between TCOR Write and Compare-Match

During the T2 state of a TCOR write cycle, the TCOR write has priority even if a compare-match occurs and the compare-match signal is disabled. Figure 12.12 shows this operation.

With TMRX, an ICR input capture contends with a compare-match in the same way as with a write to TCORC. In this case, the input capture has priority and the compare-match signal is inhibited.

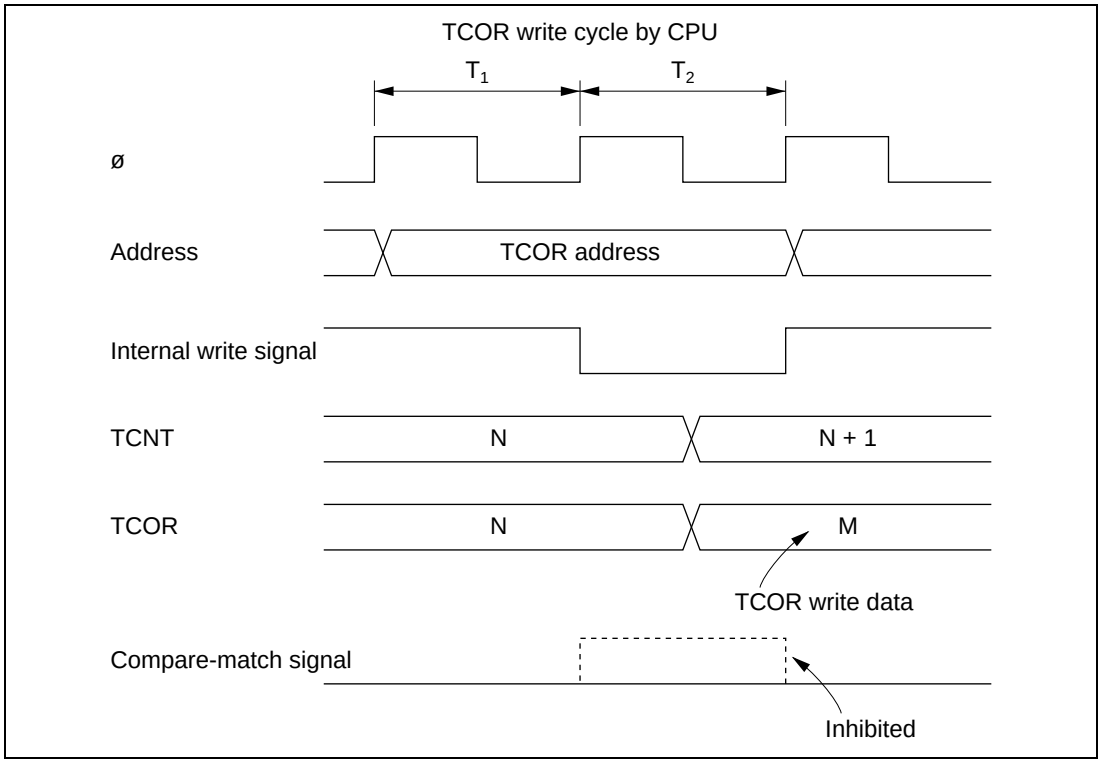


Figure 12.12 Contention between TCOR Write and Compare-Match

12.6.4 Contention between Compare-Matches A and B

If compare-matches A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output states set for compare-match A and compare-match B, as shown in table 12.6.

Table 12.6 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	
0 output	
No change	Low

12.6.5 Switching of Internal Clocks and TCNT Operation

TCNT may increment erroneously when the internal clock is switched over. Table 12.7 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation

When the TCNT clock is generated from an internal clock, the falling edge of the internal clock pulse is detected. If clock switching causes a change from high to low level, as shown in no. 3 in table 12.7, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge. This increments TCNT.

Erroneous incrementation can also happen when switching between internal and external clocks.

Table 12.7 Switching of Internal Clock and TCNT Operation

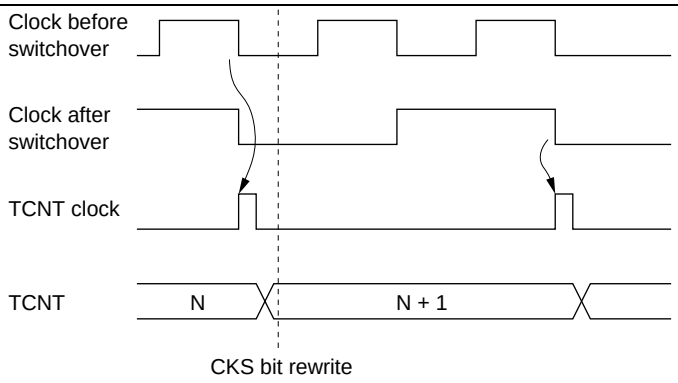
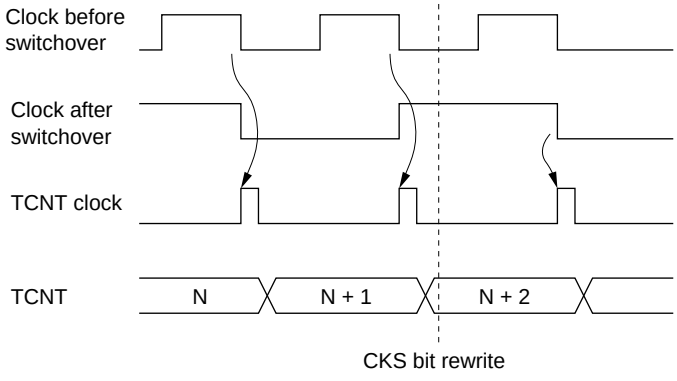
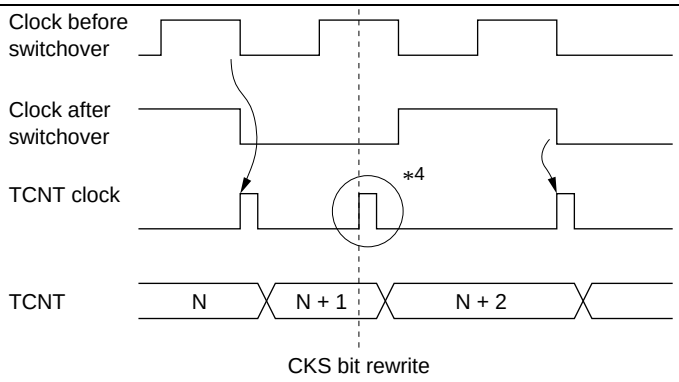
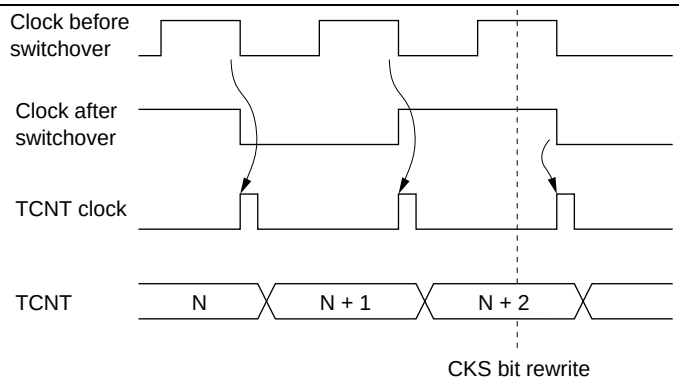
No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Switching from low to low*[1]	
2	Switching from low to high*[2]	

Table 12.7 Switching of Internal Clock and TCNT Operation (cont)

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
3	Switching from high to low*3	
4	Switching from high to high	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

Section 13 Timer Connection [H8S/2128 Series]

Provided in the H8S/2128 Series; not provided in the H8S/2124 Series.

13.1 Overview

H8S/2128 Series allows interconnection between a combination of input signals, the input/output of the single free-running timer (FRT) channel, and the three 8-bit timer channels (TMR1, TMRX, and TMRY). This capability can be used to implement complex functions such as PWM decoding and clamp waveform output. All the timers are initially set for independent operation.

13.1.1 Features

The features of the timer connection facility are as follows.

- Five input pins and four output pins, all of which can be designated for phase inversion. Positive logic is assumed for all signals used within the timer connection facility.
- An edge-detection circuit is connected to the input pins, simplifying signal input detection.
- TMRX can be used for PWM input signal decoding.
- TMRX can be used for clamp waveform generation.
- An external clock signal divided by TMR1 can be used as the FRT capture input signal.
- An internal synchronization signal can be generated using the FRT and TMRY.
- A signal generated/modified using an input signal and timer connection can be selected and output.

13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the timer connection facility.

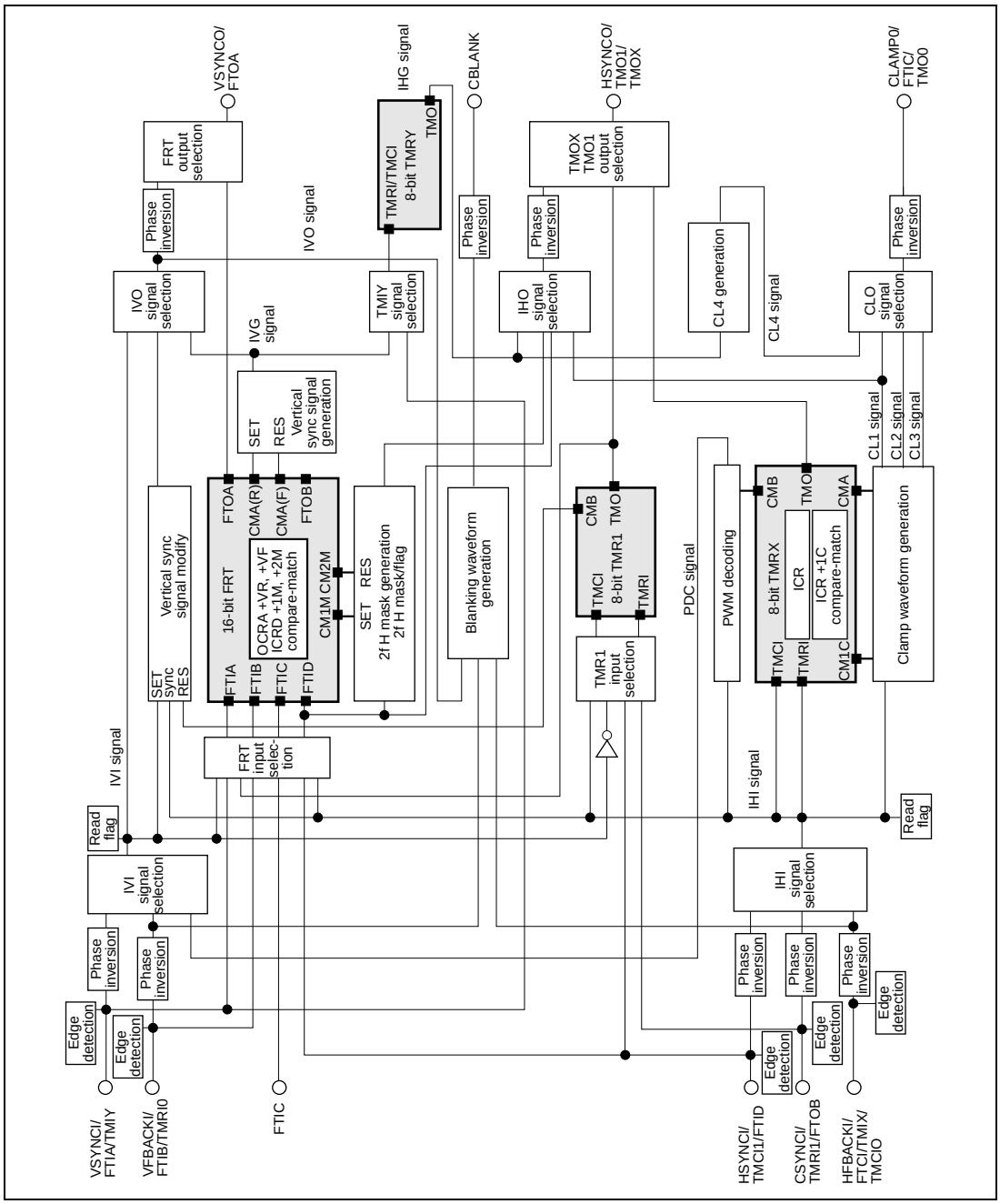


Figure 13.1 Block Diagram of Timer Connection Facility

13.1.3 Input and Output Pins

Table 13.1 lists the timer connection input and output pins.

Table 13.1 Timer Connection Input and Output Pins

Name	Abbreviation	Input/ Output	Function
Vertical synchronization signal input pin	VSYNCI	Input	Vertical synchronization signal input pin or FTIA input pin/TMIY input pin
Horizontal synchronization signal input pin	HSYNCI	Input	Horizontal synchronization signal input pin or FTID input pin/TMCI1 input pin
Composite synchronization signal input pin	CSYNCI	Input	Composite synchronization signal input pin or TMRI1 input pin/FTOB output pin
Spare vertical synchronization signal input pin	VFBACKI	Input	Spare vertical synchronization signal input pin or FTIB input pin/TMRI0 input pin
Spare horizontal synchronization signal input pin	HFBACKI	Input	Spare horizontal synchronization signal input pin or FTCI input pin/TMCI0 input pin/TMIX input pin
Vertical synchronization signal output pin	VSYNCO	Output	Vertical synchronization signal output pin or FTOA output pin
Horizontal synchronization signal output pin	HSYNCO	Output	Horizontal synchronization signal output pin or TMO1 output pin/TMOX output pin
Clamp waveform output pin	CLAMPO	Output	Clamp waveform output pin or TMO0 output pin/FTIC input pin
Blanking waveform output pin	CBLANK	Output	Blanking waveform output pin

13.1.4 Register Configuration

Table 13.2 lists the timer connection registers. Timer connection registers can only be accessed when the HIE bit in SYSCR is 0.

Table 13.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address*[1]
Timer connection register I	TCONRI	R/W	H'00	H'FFFC
Timer connection register O	TCONRO	R/W	H'00	H'FFFD
Timer connection register S	TCONRS	R/W	H'00	H'FFFE
Edge sense register	SEDGR	R/(W)*[2]	H'00*[3]	H'FFFF
Module stop control register	MSTPRH	R/W	H'3F	H'FF86
	MSTPRL	R/W	H'FF	H'FF87

Notes: 1. Lower 16 bits of the address.
2. Bits 7 to 2: Only 0 can be written to clear the flags.
3. Bits 1 and 0: Undefined (reflect the pin states).

13.2 Register Descriptions

13.2.1 Timer Connection Register I (TCONRI)

Bit	7	6	5	4	3	2	1	0
	SIMOD1	SIMOD0	SCONE	ICST	HFINV	VFINV	HIINV	VIINV
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCONRI is an 8-bit readable/writable register that controls connection between timers, the signal source for synchronization signal input, phase inversion, etc.

TCONR1 is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 and 6—Input Synchronization Mode Select 1 and 0 (SIMOD1, SIMOD0): These bits select the signal source of the IHI and IVI signals.

Bit 7	Bit 6	Description		
SIMOD1	**SIMOD0**	**Mode**	**IHI Signal**	**IVI Signal**
0	0	No signal (Initial value)	HFBACKI input	VFBACKI input
	1	S-on-G mode	CSYNCI input	PDC input
1	0	Composite mode	HSYNCI input	PDC input
	1	Separate mode	HSYNCI input	VSYNCI input

Bit 5—Synchronization Signal Connection Enable (SCONE): Selects the signal source of the FRT FTI input and the TMR1 TMCI1/TMRI1 input.

Bit 5	Description						
SCONE	**Mode**	**FTIA**	**FTIB**	**FTIC**	**FTID**	**TMCI1**	**TMRI1**
0	Normal connection (Initial value)	FTIA input	FTIB input	FTIC input	FTID input	TMCI1 input	TMRI1 input
1	Synchronization signal connection mode	IVI signal	TMO1 signal	VFBACKI input	IHI signal	IHI signal	IVI inverse signal

Bit 4—Input Capture Start Bit (ICST): The TMRX external reset input (TMRIX) is connected to the IHI signal. TMRX has input capture registers (TICR, TICRR, and TICRF). TICRR and TICRF can measure the width of a short pulse by means of a single capture operation under the control of the ICST bit. When a rising edge followed by a falling edge is detected on TMRIX after the ICST bit is set to 1, the contents of TCNT at those points are captured into TICRR and TICRF, respectively, and the ICST bit is cleared to 0.

Bit 4	
ICST	**Description**
0	The TICRR and TICRF input capture functions are stopped (Initial value) [Clearing condition] When a rising edge followed by a falling edge is detected on TMRIX
1	The TICRR and TICRF input capture functions are operating (Waiting for detection of a rising edge followed by a falling edge on TMRIX) [Setting condition] When 1 is written in ICST after reading ICST = 0

Bits 3 to 0—Input Synchronization Signal Inversion (HFINV, VFINV, HIINV, VIINV): These bits select inversion of the input phase of the spare horizontal synchronization signal (HFBACKI), the spare vertical synchronization signal (VFBACKI), the horizontal synchronization signal and composite synchronization signal (HSYNCI, CSYNCI), and the vertical synchronization signal (VSYNCI).

Bit 3

HFINV	Description	
0	The HFBACKI pin state is used directly as the HFBACKI input	(Initial value)
1	The HFBACKI pin state is inverted before use as the HFBACKI input	

Bit 2

VFINV	Description	
0	The VFBACKI pin state is used directly as the VFBACKI input	(Initial value)
1	The VFBACKI pin state is inverted before use as the VFBACKI input	

Bit 1

HIINV	Description	
0	The HSYNCI and CSYNCI pin states are used directly as the HSYNCI and CSYNCI inputs	(Initial value)
1	The HSYNCI and CSYNCI pin states are inverted before use as the HSYNCI and CSYNCI inputs	

Bit 0

VIINV	Description	
0	The VSYNCI pin state is used directly as the VSYNCI input	(Initial value)
1	The VSYNCI pin state is inverted before use as the VSYNCI input	

13.2.2 Timer Connection Register O (TCONRO)

Bit	7	6	5	4	3	2	1	0
	HOE	VOE	CLOE	CBOE	HOINV	VOINV	CLOINV	CBOINV
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCONRO is an 8-bit readable/writable register that controls output signal output, phase inversion, etc.

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

Bits 7 and 4—Output Enable (HOE, VOE, CLOE, CBOE): These bits control enabling/disabling of horizontal synchronization signal (HSYNCO), vertical synchronization signal (VSYNCO), clamp waveform (CLAMPO), and blanking waveform (CBLANK) output. When output is disabled, the state of the relevant pin is determined by the port DR and DDR, FRT, TMR, and PWM settings.

Output enabling/disabling control does not affect the port, FRT, or TMR input functions, but some FRT and TMR input signal sources are determined by the SCONE bit in TCONRI.

Bit 7

HOE	Description	
0	The P67/TMO1/TMOX/CIN7/HSYNCO pin functions as the P67/TMO1/TMOX/CIN7 pin	(Initial value)
1	The P67/TMO1/TMOX/CIN7/HSYNCO pin functions as the HSYNCO pin	

Bit 6

VOE	Description	
0	The P61/FTOA/CIN1/VSYNCO pin functions as the P61/FTOA/CIN1 pin	(Initial value)
1	The P61/FTOA/CIN1/VSYNCO pin functions as the VSYNCO pin	

Bit 5

CLOE	Description	
0	The P64/FTIC/CIN4/CLAMPO pin functions as the P64/FTIC/CIN4 pin	(Initial value)
1	The P64/FTIC/CIN4/CLAMPO pin functions as the CLAMPO pin	

Bit 4

CBOE	Description	
0	The P27/A15/PW15/CBLANK pin functions as the P27/A15/PW15 pin	(Initial value)
1	In mode 1 (expanded mode with on-chip ROM disabled): The P27/A15/PW15/CBLANK pin functions as the A15 pin In modes 2 and 3 (modes with on-chip ROM enabled): The P27/A15/PW15/CBLANK pin functions as the CBLANK pin	

Bits 3 to 0—Output Synchronization Signal Inversion (HOINV, VOINV, CLOINV, CBOINV): These bits select inversion of the output phase of the horizontal synchronization signal (HSYNCO), the vertical synchronization signal (VSYNCO), the clamp waveform (CLAMPO), and the blank waveform (CBLANK).

Bit 3

HOINV	Description	
0	The IHO signal is used directly as the HSYNCO output	(Initial value)
1	The IHO signal is inverted before use as the HSYNCO output	

Bit 2

VOINV	Description	
0	The IVO signal is used directly as the VSYNCO output	(Initial value)
1	The IVO signal is inverted before use as the VSYNCO output	

Bit 1

CLOINV	Description	
0	The CLO signal (CL1, CL2, CL3, or CL4 signal) is used directly as the CLAMPO output	(Initial value)
1	The CLO signal (CL1, CL2, CL3, or CL4 signal) is inverted before use as the CLAMPO output	

Bit 0

CBOINV	Description	
0	The CBLANK signal is used directly as the CBLANK output	(Initial value)
1	The CBLANK signal is inverted before use as the CBLANK output	

13.2.3 Timer Connection Register S (TCONRS)

Bit	7	6	5	4	3	2	1	0
	TMRX/Y	ISGENE	HOMOD1	HOMOD0	VOMOD1	VOMOD0	CLMOD1	CLMOD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCONRS is an 8-bit readable/writable register that selects 8-bit timer TMRX/TMRY access and the synchronization signal output signal source and generation method.

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

Bit 7—TMRX/TMRY Access Select (TMRX/Y): The TMRX and TMRY registers can only be accessed when the HIE bit in SYSCR is cleared to 0. In the H8S/2128 Series, some of the TMRX registers and the TMRY registers are assigned to the same memory space addresses (H'FFF0 to H'FFF5), and the TMRX/Y bit determines which registers are accessed. In the H8S/2124 Series, there is no control of TMRY register access by this bit.

Bit 7		
TMRX/Y	**Description**	
0	The TMRX registers are accessed at addresses H'FFF0 to H'FFF5	(Initial value)
1	The TMRY registers are accessed at addresses H'FFF0 to H'FFF5	

Bit 6—Internal Synchronization Signal Select (ISGENE): Selects internal synchronization signals (IHG, IVG, and CL4 signals) as the signal sources for the IHO, IVO, and CLO signals.

Bits 5 and 4—Horizontal Synchronization Output Mode Select 1 and 0 (HOMOD1, HOMOD0): These bits select the signal source and generation method for the IHO signal.

Bit 6	Bit 5	Bit 4		
ISGENE	**VOMOD1**	**VOMOD0**	**Description**	
0	0	0	The IHI signal (without 2fH modification) is selected	(Initial value)
		1	The IHI signal (with 2fH modification) is selected	
	1	0	The CL1 signal is selected	
		1		
1	0	0	The IHG signal is selected	
		1		
	1	0		
		1		

Bits 3 and 2—Vertical Synchronization Output Mode Select 1 and 0 (VOMOD1, VOMOD0): These bits select the signal source and generation method for the IVO signal.

Bit 6	Bit 3	Bit 2		
ISGENE	**VOMOD1**	**VOMOD0**	**Description**	
0	0	0	The IVI signal (without fall modification or IHI synchronization) is selected	(Initial value)
		1	The IVI signal (without fall modification, with IHI synchronization) is selected	
	1	0	The IVI signal (with fall modification, without IHI synchronization) is selected	
		1	The IVI signal (with fall modification and IHI synchronization) is selected	
1	0	0	The IVG signal is selected	
		1		
	1	0		
		1		

Bits 1 and 0—Clamp Waveform Mode Select 1 and 0 (CLMOD1, CLMOD0): These bits select the signal source for the CLO signal (clamp waveform).

Bit 6	Bit 1	Bit 0		
ISGENE	**CLMOD1**	**CLMOD2**	**Description**	
0	0	0	The CL1 signal is selected	(Initial value)
		1	The CL2 signal is selected	
	1	0	The CL3 signal is selected	
		1		
1	0	0	The CL4 signal is selected	
		1		
	1	0		
		1		

13.2.4 Edge Sense Register (SEDGR)

Bit	7	6	5	4	3	2	1	0
	VEDG	HEDG	CEDG	HFEDG	VFEDG	PREQF	IHI	IVI
Initial value	0	0	0	0	0	0	—*2	—*2
Read/Write	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R

Notes: 1. Only 0 can be written, to clear the flags.
2. The initial value is undefined since it depends on the pin states.

SEDGR is an 8-bit readable/writable register used to detect a rising edge on the timer connection input pins and the occurrence of 2fH modification, and to determine the phase of the IVI and IHI signals.

The upper 6 bits of SEDGR are initialized to 0 by a reset and in hardware standby mode. The initial value of the lower 2 bits is undefined, since it depends on the pin states.

Bit 7—VSYNCI Edge (VEDG): Detects a rising edge on the VSYNCI pin.

Bit 7

VEDG	Description	
0	[Clearing condition] When 0 is written in VEDG after reading VEDG = 1	(Initial value)
1	[Setting condition] When a rising edge is detected on the VSYNCI pin	

Bit 6—HSYNCI Edge (HEDG): Detects a rising edge on the HSYNCI pin.

Bit 6

HEDG	Description	
0	[Clearing condition] When 0 is written in HEDG after reading HEDG = 1	(Initial value)
1	[Setting condition] When a rising edge is detected on the HSYNCI pin	

Bit 5—CSYNCI Edge (CEDG): Detects a rising edge on the CSYNCI pin.

Bit 5

CEDG	Description	
0	[Clearing condition] When 0 is written in CEDG after reading CEDG = 1	(Initial value)
1	[Setting condition] When a rising edge is detected on the CSYNCI pin	

Bit 4—HFBACKI Edge (HFEDG): Detects a rising edge on the HFBACKI pin.

Bit 4

HFEDG	Description	
0	[Clearing condition] When 0 is written in HFEDG after reading HFEDG = 1	(Initial value)
1	[Setting condition] When a rising edge is detected on the HFBACKI pin	

Bit 3—VFBACKI Edge (VFEDG): Detects a rising edge on the VFBACKI pin.

Bit 3

VFEDG	Description	
0	[Clearing condition] When 0 is written in VFEDG after reading VFEDG = 1	(Initial value)
1	[Setting condition] When a rising edge is detected on the VFBACKI pin	

Bit 2—Pre-Equalization Flag (PREQF): Detects the occurrence of an IHI signal 2fH modification condition. The generation of a falling/rising edge in the IHI signal during a mask interval is expressed as the occurrence of a 2fH modification condition. For details, see section 13.3.4, IHI Signal 2fH Modification.

Bit 2

PREQF	Description	
0	[Clearing condition] When 0 is written in PREQF after reading PREQF = 1	(Initial value)
1	[Setting condition] When an IHI signal 2fH modification condition is detected	

Bit 1—IHI Signal Level (IHI): Indicates the current level of the IHI signal. Signal source and phase inversion selection for the IHI signal depends on the contents of TCONRI. Read this bit to determine whether the input signal is positive or negative, then maintain the IHI signal at positive phase by modifying TCONRI.

Bit 1

IHI	Description
0	The IHI signal is low
1	The IHI signal is high

Bit 0—IVI Signal Level (IVI): Indicates the current level of the IVI signal. Signal source and phase inversion selection for the IVI signal depends on the contents of TCONRI. Read this bit to determine whether the input signal is positive or negative, then maintain the IVI signal at positive phase by modifying TCONRI.

Bit 0

IVI	Description
0	The IVI signal is low
1	The IVI signal is high

13.2.5 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	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	R/W	R/W

MSTPCR, comprising two 8-bit readable/writable registers, performs module stop mode control. When the MSTP13, MSTP12, and MSTP8 bits are set to 1, the 16-bit free-running timer, 8-bit timer channels 0 and 1 and channels X and Y, and timer connection, respectively, halt and enter module stop mode at the end of the bus cycle. See section 21.5, Module Stop Mode, for details.

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

MSTPCRH Bit 5—Module Stop (MSTP13): Specifies FRT module stop mode.

MSTPCRH Bit 5		
MSTP13	**Description**	
0	FRT module stop mode is cleared	
1	FRT module stop mode is set	(Initial value)

MSTPCRH Bit 4—Module Stop (MSTP12): Specifies 8-bit timer channel 0 and 1 module stop mode.

MSTPCRH Bit 4		
MSTP12	**Description**	
0	8-bit timer channel 0 and 1 module stop mode is cleared	
1	8-bit timer channel 0 and 1 module stop mode is set	(Initial value)

MSTPCRH Bit 0—Module Stop (MSTP8): Specifies 8-bit timer channel X and Y and timer connection module stop mode.

MSTPCRH Bit 0		
MSTP8	**Description**	
0	8-bit timer channel X and Y and timer connection module stop mode is cleared	
1	8-bit timer channel X and Y and timer connection module stop mode is set	(Initial value)

13.3 Operation

13.3.1 PWM Decoding (PDC Signal Generation)

The timer connection facility and TMRX can be used to decode a PWM signal in which 0 and 1 are represented by the pulse width. To do this, a signal in which a rising edge is generated at regular intervals must be selected as the IHI signal.

The timer counter (TCNT) in TMRX is set to count the internal clock pulses and to be cleared on the rising edge of the external reset signal (IHI signal). The value to be used as the threshold for deciding the pulse width is written in TCORB. The PWM decoder contains a delay latch which uses the IHI signal as data and compare-match signal B (CMB) as a clock, and the state of the IHI signal (the result of the pulse width decision) at the compare-match signal B timing after TCNT is reset by the rise of the IHI signal is output as the PDC signal. The pulse width setting using TICRR and TICRF of TMRX can be used to determine the pulse width decision threshold. Examples of TCR and TCORB settings are shown in tables 13.3 and 13.4, and the timing chart is shown in figure 13.2.

Table 13.3 Examples of TCR Settings

Bit(s)	Abbreviation	Contents	Description
7	CMIEB	0	Interrupts due to compare-match and overflow are disabled
6	CMIEA	0	
5	OVIE	0	
4 and 3	CCLR1, CCLR0	11	TCNT is cleared by the rising edge of the external reset signal (IHI signal)
2 to 0	CKS2 to CKS0	001	Incremented on internal clock: ø

Table 13.4 Examples of TCORB (Pulse Width Threshold) Settings

	ø:10 MHz	ø: 12 MHz	ø: 16 MHz	ø: 20 MHz
H'07	0.8 µs	0.67 µs	0.5 µs	0.4 µs
H'0F	1.6 µs	1.33 µs	1 µs	0.8 µs
H'1F	3.2 µs	2.67 µs	2 µs	1.6 µs
H'3F	6.4 µs	5.33 µs	4 µs	3.2 µs
H'7F	12.8 µs	10.67 µs	8 µs	6.4 µs

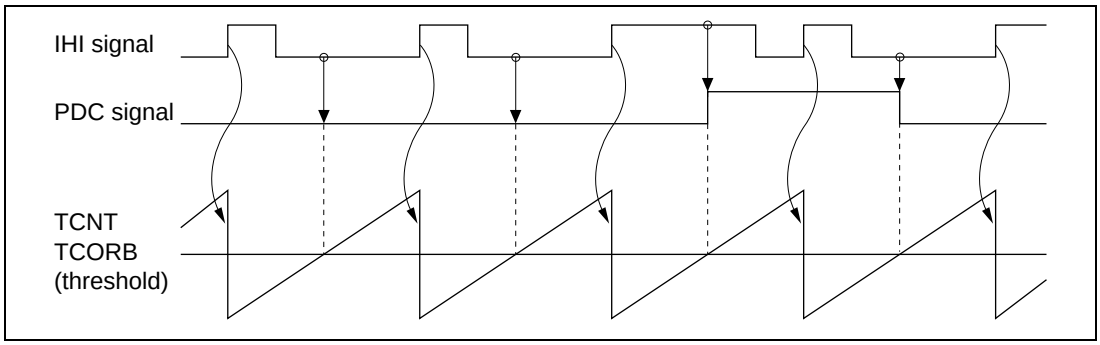


Figure 13.2 Timing Chart for PWM Decoding

13.3.2 Clamp Waveform Generation (CL1/CL2/CL3 Signal Generation)

The timer connection facility and TMRX can be used to generate signals with different duty cycles and rising/falling edges (clamp waveforms) in synchronization with the input signal (IHI signal). Three clamp waveforms can be generated: the CL1, CL2, and CL3 signals. In addition, the CL4 signal can be generated using TMRY.

The CL1 signal rises simultaneously with the rise of the IHI signal, and when the CL1 signal is high, the CL2 signal rises simultaneously with the fall of the IHI signal. The fall of both the CL1 and the CL2 signal can be specified by TCORA.

The rise of the CL3 signal can be specified as simultaneous with the sampling of the fall of the IHI signal using the system clock, and the fall of the CL3 signal can be specified by TCORC. The CL3 signal falls at the rise of the IHI signal.

TCNT in TMRX is set to count internal clock pulses and to be cleared on the rising edge of the external reset signal (IHI signal).

The value to be used as the CL1 signal pulse width is written in TCORA. Write a value of H'02 or more in TCORA when internal clock ø is selected as the TMRX counter clock, and a value or H'01 or more when ø/2 is selected. When internal clock ø is selected, the CL1 signal pulse width is (TCORA set value + 3 ± 0.5). When the CL2 signal is used, the setting must be made so that this pulse width is greater than the IHI signal pulse width.

The value to be used as the CL3 signal pulse width is written in TCORC. The TICR register in TMRX captures the value of TCNT at the inverse of the external reset signal edge (in this case, the falling edge of the IHI signal). The timing of the fall of the CL3 signal is determined by the sum of the contents of TICR and TCORC. Caution is required if the rising edge of the IHI signal precedes the fall timing set by the contents of TCORC, since the IHI signal will cause the CL3 signal to fall.

Examples of TMRX TCR settings are the same as those in table 13.3. The clamp waveform timing charts are shown in figures 13.3 and 13.4.

Since the rise of the CL1 and CL2 signals is synchronized with the edge of the IHI signal, and their fall is synchronized with the system clock, the pulse width variation is equivalent to the resolution of the system clock.

Both the rise and the fall of the CL3 signal are synchronized with the system clock and the pulse width is fixed, but there is a variation in the phase relationship with the IHI signal equivalent to the resolution of the system clock.

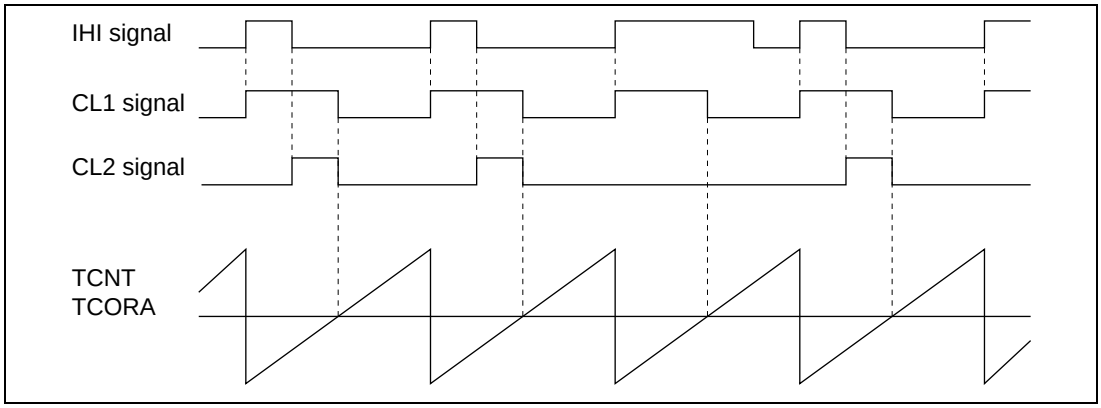


Figure 13.3 Timing Chart for Clamp Waveform Generation (CL1 and CL2 Signals)

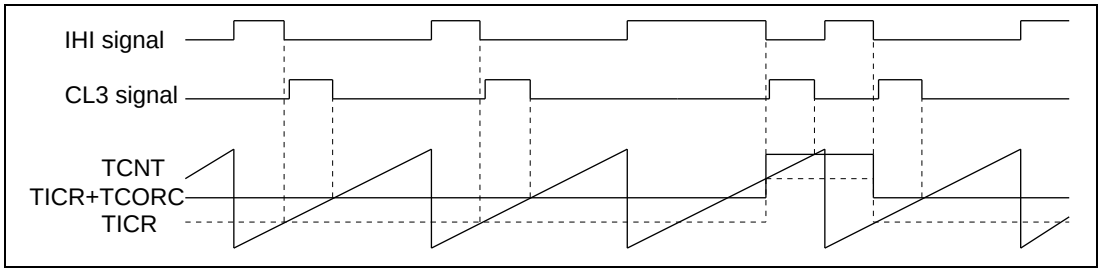


Figure 13.4 Timing Chart for Clamp Waveform Generation (CL3 Signal)

13.3.3 Measurement of 8-Bit Timer Divided Waveform Period

The timer connection facility, TMR1, and the free-running timer (FRT) can be used to measure the period of an IHI signal divided waveform. Since TMR1 can be cleared by a rising edge of the IVI signal, the rise and fall of the IHI signal divided waveform can be virtually synchronized with the IVI signal. This enables period measurement to be carried out efficiently.

To measure the period of an IHI signal divided waveform, TCNT in TMR1 is set to count the external clock (IHI signal) pulses and to be cleared on the rising edge of the external reset signal (IVI signal). The value to be used as the division factor is written in TCORA, and the TMO output method is specified by the OS bits in TCSR. Examples of TMR1 TCR and TCSR settings are shown in table 13.5, and the timing chart for measurement of the IVI signal and IHI signal divided waveform periods is shown in figure 13.5. The period of the IHI signal divided waveform is given by (ICRD(3) – ICRD(2)) × the resolution.

Table 13.5 Examples of TCR and TCSR Settings

Register	Bit(s)	Abbreviation	Contents	Description
TCR in TMR1	7	CMIEB	0	Interrupts due to compare-match and overflow are disabled
	6	CMIEA	0	
	5	OVIE	0	
	4 and 3	CCLR1, CCLR0	11	TCNT is cleared by the rising edge of the external reset signal (IVI signal)
	2 to 0	CKS2 to CKS0	101	TCNT is incremented on the rising edge of the external clock (IHI signal)
TCSR in TMR1	3 to 0	OS3 to OS0	0011	Not changed by compare-match B; output inverted by compare-match A (toggle output): division by 512
			1001	or when TCORB < TCORA, 1 output on compare-match B, and 0 output on compare-match A: division by 256
TCR in FRT	6	IEDGB	0/1	0: FRC value is transferred to ICRB on falling edge of input capture input B (IHI divided signal waveform) 1: FRC value is transferred to ICRB on rising edge of input capture input B (IHI divided signal waveform)
	1 and 0	CKS1, CKS0	01	FRC is incremented on internal clock: ø/8
TCSR in FRT	0	CCLRA	0	FRC clearing is disabled

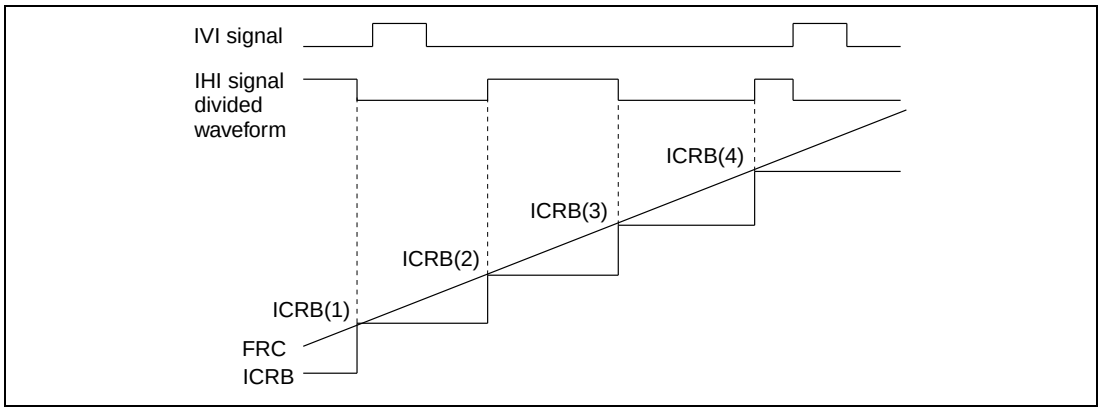


Figure 13.5 Timing Chart for Measurement of IVI Signal and IHI Signal Divided Waveform Periods

13.3.4 IHI Signal and 2fH Modification

By using the timer connection FRT, even if there is a part of the IHI signal with twice the frequency, this can be eliminated. In order for this function to operate properly, the duty cycle of the IHI signal must be approximately 30% or less, or approximately 70% or above.

The 8-bit OCRDM contents or twice the OCRDM contents can be added automatically to the data captured in ICRD in the FRT, and compare-matches generated at these points. The interval between the two compare-matches is called a mask interval. A value equivalent to approximately 1/3 the IHI signal period is written in OCRDM. ICRD is set so that capture is performed on the rise of the IHI signal.

Since the IHI signal supplied to the IHO signal selection circuit is normally set on the rise of the IHI signal and reset on the fall, its waveform is the same as that of the original IHI signal. When 2fH modification is selected, IHI signal edge detection is disabled during mask intervals. Capture is also disabled during these intervals.

Examples of FRT TCR settings are shown in table 13.6, and the 2fH modification timing chart is shown in figure 13.6.

Table 13.6 Examples of TCR, TCSR, TCOR, and OCRDM Settings

Register	Bit(s)	Abbreviation	Contents	Description
TCR in FRT	4	IEDGD	1	FRC value is transferred to ICRD on the rising edge of input capture input D (IHI signal)
	1 and 0	CKS1, CKS0	01	FRC is incremented on internal clock: ø/8
TCSR in FRT	0	CCLRA	0	FRC clearing is disabled
TCOR in FRT	7	ICRDMS	1	ICRD is set to the operating mode in which OCRDM is used
OCRDM in FRT	7 to 0	OCRDM7 to 0	H'01 to H'FF	Specifies the period during which ICRD operation is masked

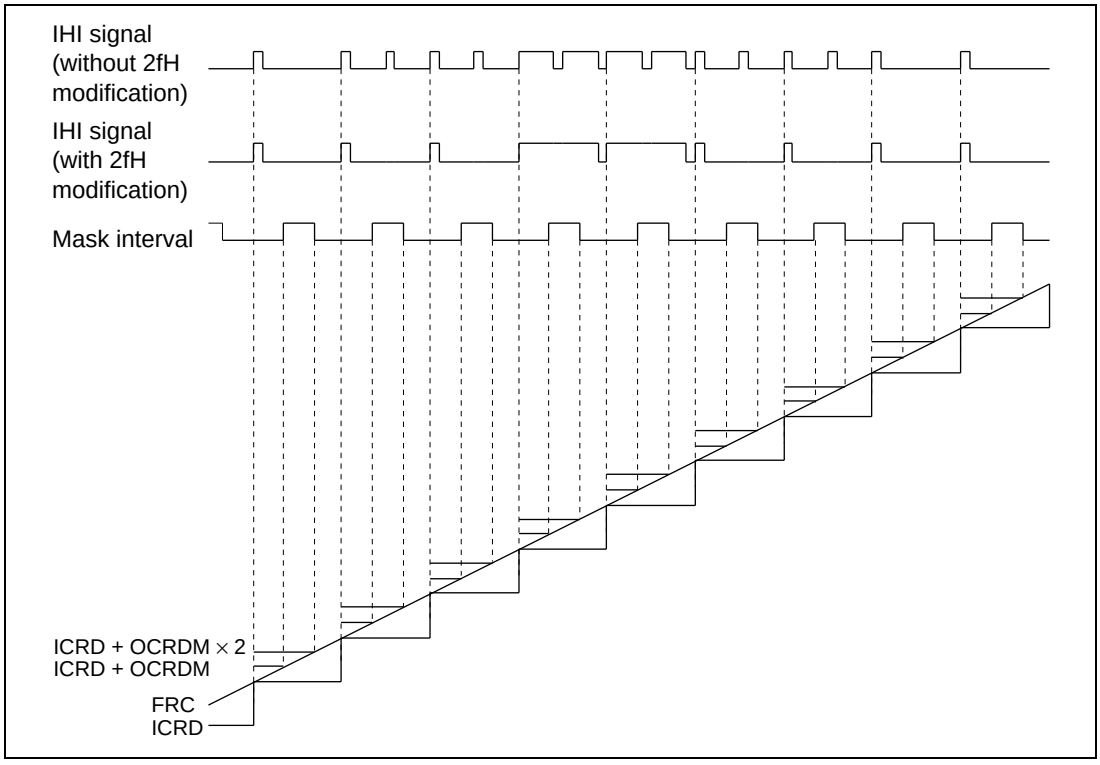


Figure 13.6 2fH Modification Timing Chart

13.3.5 IVI Signal Fall Modification and IHI Synchronization

By using the timer connection TMR1, the fall of the IVI signal can be shifted backward by the specified number of IHI signal waveforms. Also, the fall of the IVI signal can be synchronized with the rise of the IHI signal.

To perform 8-bit timer divided waveform period measurement, TCNT in TMR1 is set to count external clock (IHI signal) pulses, and to be cleared on the rising edge of the external reset signal (inverse of the IVI signal). The number of IHI signal pulses until the fall of the IVI signal is written in TCORB.

Since the IVI signal supplied to the IVO signal selection circuit is normally set on the rise of the IVI signal and reset on the fall, its waveform is the same as that of the original IVI signal. When fall modification is selected, a reset is performed on a TMR1 TCORB compare-match.

The fall of the waveform generated in this way can be synchronized with the rise of the IHI signal, regardless of whether or not fall modification is selected.

Examples of TMR1 TCORB, TCR, and TCSR settings are shown in table 13.7, and the fall modification/IHI synchronization timing chart is shown in figure 13.7.

Table 13.7 Examples of TCORB, TCR, and TCSR Settings

Register	Bit(s)	Abbreviation	Contents	Description
TCR in TMR1	7	CMIEB	0	Interrupts due to compare-match and overflow are disabled
	6	CMIEA	0	
	5	OVIE	0	
	4 and 3	CCLR1, CCLR0	11	TCNT is cleared by the rising edge of the external reset signal (inverse of the IVI signal)
	2 to 0	CKS2 to CKS0	101	TCNT is incremented on the rising edge of the external clock (IHI signal)
TCSR in TMR1	3 to 0	OS3 to OS0	0011	Not changed by compare-match B; output inverted by compare-match A (toggle output)
			1001	or when TCORB < TCORA, 1 output on compare-match B, 0 output on compare-match A
TOCRB in TMR1			H'03 (example)	Compare-match on the 4th (example) rise of the IHI signal after the rise of the inverse of the IVI signal

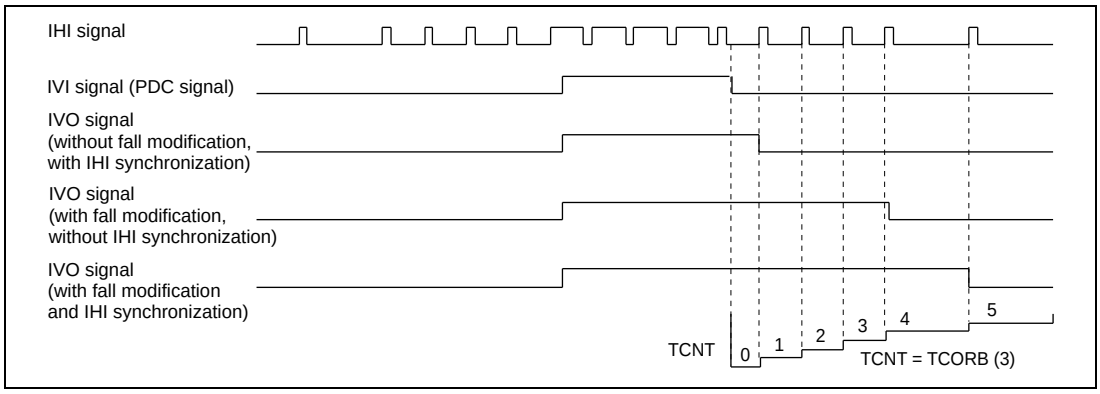


Figure 13.7 Fall Modification/IHI Synchronization Timing Chart

13.3.6 Internal Synchronization Signal Generation (IHG/IVG/CL4 Signal Generation)

By using the timer connection FRT and TMRY, it is possible to automatically generate internal signals (IHG and IVG signals) corresponding to the IHI and IVI signals. As the IHG signal is synchronized with the rise of the IVG signal, the IHG signal period must be made a divisor of the IVG signal period in order to keep it constant. In addition, the CL4 signal can be generated in synchronization with the IHG signal.

The contents of OCRA in the FRT are updated by the automatic addition of the contents of OCRAR or OCRAF, alternately, each time a compare-match occurs. A value corresponding to the 0 interval of the IVG signal is written in OCRAR, and a value corresponding to the 1 interval of the IVG signal is written in OCRAF. The IVG signal is set by a compare-match after an OCRAR addition, and reset by a compare-match after an OCRAF addition.

The IHG signal is the TMRY 8-bit timer output. TMRY is set to count internal clock pulses, and to be cleared on TCORA compare-match, to fix the period and set the timer output. TCORB is set so as to reset the timer output. The IVG signal is connected as the TMRY reset input (TMRI), and the rise of the IVG signal can be treated in the same way as a TCORA compare-match.

The CL4 signal is a waveform that rises within one system clock period after the fall of the IHG signal, and has a 1 interval of 6 system clock periods.

Examples of settings of TCORA, TCORB, TCR, and TCSR in TMRY, and OCRAR, OCRAF, and TCR in the FRT, are shown in table 13.8, and the IHG signal/IVG signal timing chart is shown in figure 13.8.

Table 13.8 Examples of OCRAR, OCRAF, TOCR, TCORA, TCORB, TCR, and TCSR Settings

Register	Bit(s)	Abbreviation	Contents	Description	
TCR in TMRY	7	CMIEB	0	Interrupts due to compare-match and overflow are disabled	
	6	CMIEA	0		
	5	OVIE	0		
	4 and 3	CCLR1, CCLR0	01	TCNT is cleared by compare-match A	
	2 to 0	CKS2 to CKS0	001	TCNT is incremented on internal clock: ø/4	
TCSR in TMRY	3 to 0	OS3 to OS0	0110	0 output on compare-match B 1 output on compare-match A	
TOCRA in TMRY			H'3F (example)	IHG signal period = ø × 256	
TOCRB in TMRY			H'03 (example)	IHG signal 1 interval = ø × 16	
TCR in FRT	1 and 0	CKS1, CKS0	01	FRC is incremented on internal clock: ø/8	
OCRAR in FRT			H'7FEF (example)	IVG signal 0 interval = ø × 262016	IVG signal period = ø × 262144 (1024 times IHG signal)
OCRAF in FRT			H'000F (example)	IVG signal 1 interval = ø × 128	
TOCR in FRT	6	OCRAMS	1	OCRA is set to the operating mode in which OCRAR and OCRAF are used	

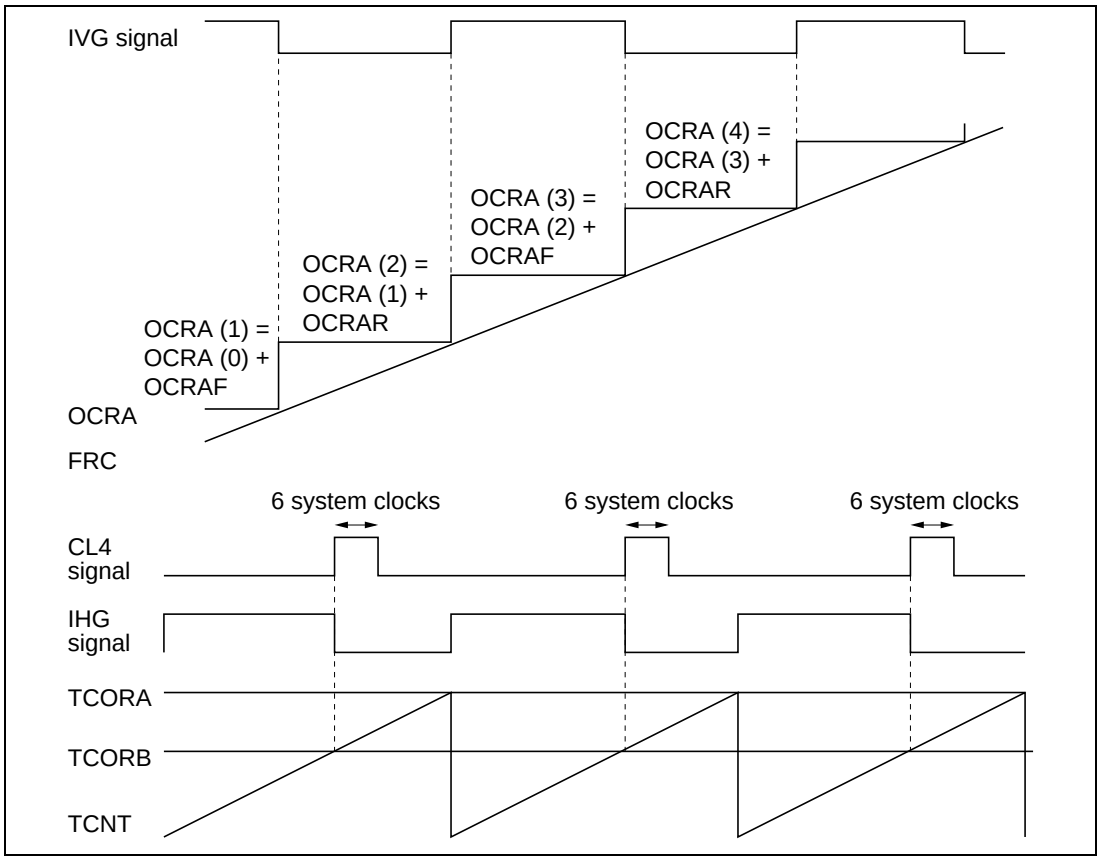


Figure 13.8 IVG Signal/IHG Signal/CL4 Signal Timing Chart

13.3.7 HSYNCO Output

With the HSYNCO output, the meaning of the signal source to be selected and use or non-use of modification varies according to the IHI signal source and the waveform required by external circuitry. The meaning of the HSYNCO output in each mode is shown in table 13.9.

Table 13.9 Meaning of HSYNCO Output in Each Mode

Mode	IHI Signal	IHO Signal	Meaning of IHO Signal
No signal	HFBACKI input	IHI signal (without 2fH modification)	HFBACKI input is output directly
		IHI signal (with 2fH modification)	Meaningless unless there is a double-frequency part in the HFBACKI input
		CL1 signal	HFBACKI input 1 interval is changed before output
		IHG signal	Internal synchronization signal is output
S-on-G mode	CSYNCI input	IHI signal (without 2fH modification)	CSYNCI input (composite synchronization signal) is output directly
		IHI signal (with 2fH modification)	Double-frequency part of CSYNCI input (composite synchronization signal) is eliminated before output
		CL1 signal	CSYNCI input (composite synchronization signal) horizontal synchronization signal part is separated before output
		IHG signal	Internal synchronization signal is output
Composite mode	HSYNCI input	IHI signal (without 2fH modification)	HSYNCI input (composite synchronization signal) is output directly
		IHI signal (with 2fH modification)	Double-frequency part of HSYNCI input (composite synchronization signal) is eliminated before output
		CL1 signal	HSYNCI input (composite synchronization signal) horizontal synchronization signal part is separated before output
		IHG signal	Internal synchronization signal is output
Separate mode	HSYNCI input	IHI signal (without 2fH modification)	HSYNCI input (horizontal synchronization signal) is output directly
		IHI signal (with 2fH modification)	Meaningless unless there is a double-frequency part in the HSYNCI input (horizontal synchronization signal)
		CL1 signal	HSYNCI input (horizontal synchronization signal) 1 interval is changed before output
		IHG signal	Internal synchronization signal is output

13.3.8 VSYNCO Output

With the VSYNCO output, the meaning of the signal source to be selected and use or non-use of modification varies according to the IVI signal source and the waveform required by external circuitry. The meaning of the VSYNCO output in each mode is shown in table 13.10.

Table 13.10 Meaning of VSYNCO Output in Each Mode

Mode	IVI Signal	IVO Signal	Meaning of IVO Signal
No signal	VFBACKI input	IVI signal (without fall modification or IHI synchronization)	VFBACKI input is output directly
		IVI signal (without fall modification, with IHI synchronization)	Meaningless unless VFBACKI input is synchronized with HFBACKI input
		IVI signal (with fall modification, without IHI synchronization)	VFBACKI input fall is modified before output
		IVI signal (with fall modification and IHI synchronization)	VFBACKI input fall is modified and signal is synchronized with HFBACKI input before output
		IVG signal	Internal synchronization signal is output
S-on-G mode or composite mode	PDC signal	IVI signal (without fall modification or IHI synchronization)	CSYNCI/HSYNCI input (composite synchronization signal) vertical synchronization signal part is separated before output
		IVI signal (without fall modification, with IHI synchronization)	CSYNCI/HSYNCI input (composite synchronization signal) vertical synchronization signal part is separated, and signal is synchronized with CSYNCI/HSYNCI input before output
		IVI signal (with fall modification, without IHI synchronization)	CSYNCI/HSYNCI input (composite synchronization signal) vertical synchronization signal part is separated, and fall is modified before output
		IVI signal (with fall modification and IHI synchronization)	CSYNCI/HSYNCI input (composite synchronization signal) vertical synchronization signal part is separated, fall is modified, and signal is synchronized with CSYNCI/HSYNCI input before output
		IVG signal	Internal synchronization signal is output

Table 13.10 Meaning of VSYNCO Output in Each Mode (cont)

Mode	IVI Signal	IVO Signal	Meaning of IVO Signal
Separate mode	VSYNCI input	IVI signal (without fall modification or IHI synchronization)	VSYNCI input (vertical synchronization signal) is output directly
		IVI signal (without fall modification, with IHI synchronization)	Meaningless unless VSYNCI input (vertical synchronization signal) is synchronized with HSYNCI input (horizontal synchronization signal)
		IVI signal (with fall modification, without IHI synchronization)	VSYNCI input (vertical synchronization signal) fall is modified before output
		IVI signal (with fall modification and IHI synchronization)	VSYNCI input (vertical synchronization signal) fall is modified and signal is synchronized with HSYNCI input (horizontal synchronization signal) before output
		IVG signal	Internal synchronization signal is output

13.3.9 CBLANK Output

Using the signals generated/selected with timer connection, it is possible to generate a waveform based on the composite synchronization signal (blanking waveform).

One kind of blanking waveform is generated by combining HFBACKI and VFBACKI inputs, with the phase polarity made positive by means of bits HFINV and VFINV in TCONRI, with the IVO signal.

The composition logic is shown in figure 13.9.

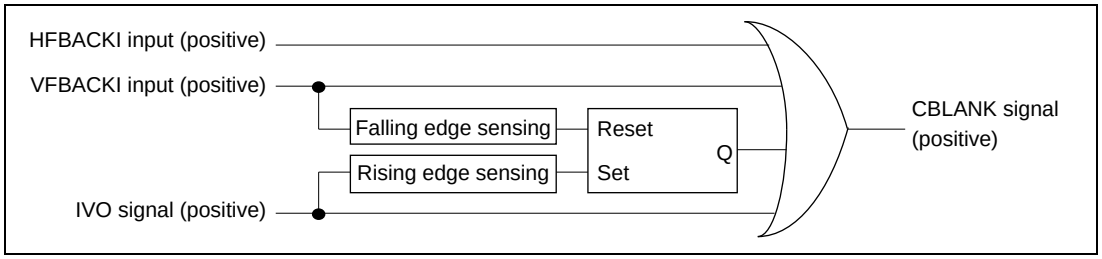


Figure 13.9 CBLANK Output Waveform Generation

Section 14 Watchdog Timer (WDT)

14.1 Overview

These series have an on-chip watchdog timer/watch timer with two channels (WDT0, WDT1). The WDT outputs an overflow signal if a system crash prevents the CPU from writing to the timer counter, allowing it to overflow. At the same time, the WDT can also generate an internal reset signal or internal NMI interrupt signal.

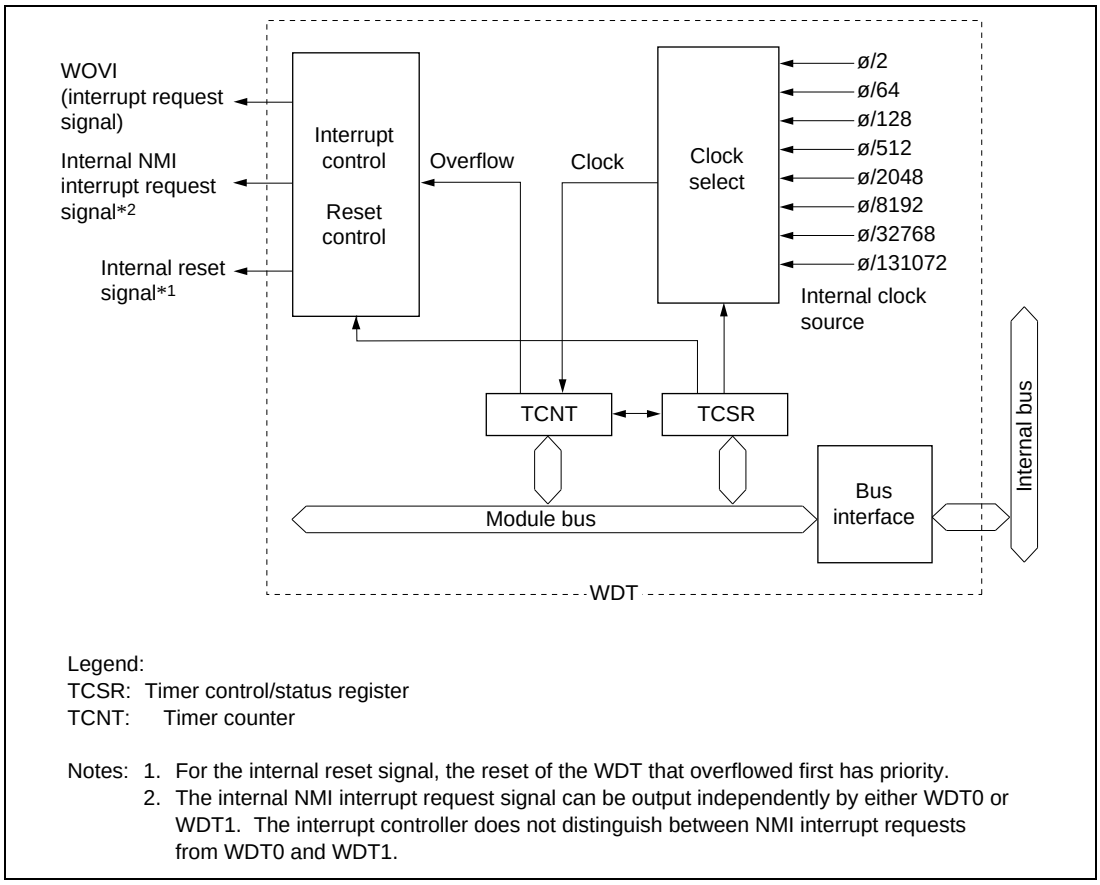
When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer mode, an interval timer interrupt is generated each time the counter overflows.

14.1.1 Features

- Switchable between watchdog timer mode and interval timer mode
  - WOVI interrupt generation in interval timer mode
- Internal reset or internal interrupt generated when the timer counter overflows
  - Choice of internal reset or NMI interrupt generation in watchdog timer mode
- Choice of 8 (WDT0) or 16 (WDT1) counter input clocks
  - Maximum WDT interval: system clock period × 131072 × 256
  - Subclock can be selected for the WDT1 input counter
    Maximum interval when the subclock is selected: subclock period × 256 × 256

14.1.2 Block Diagram

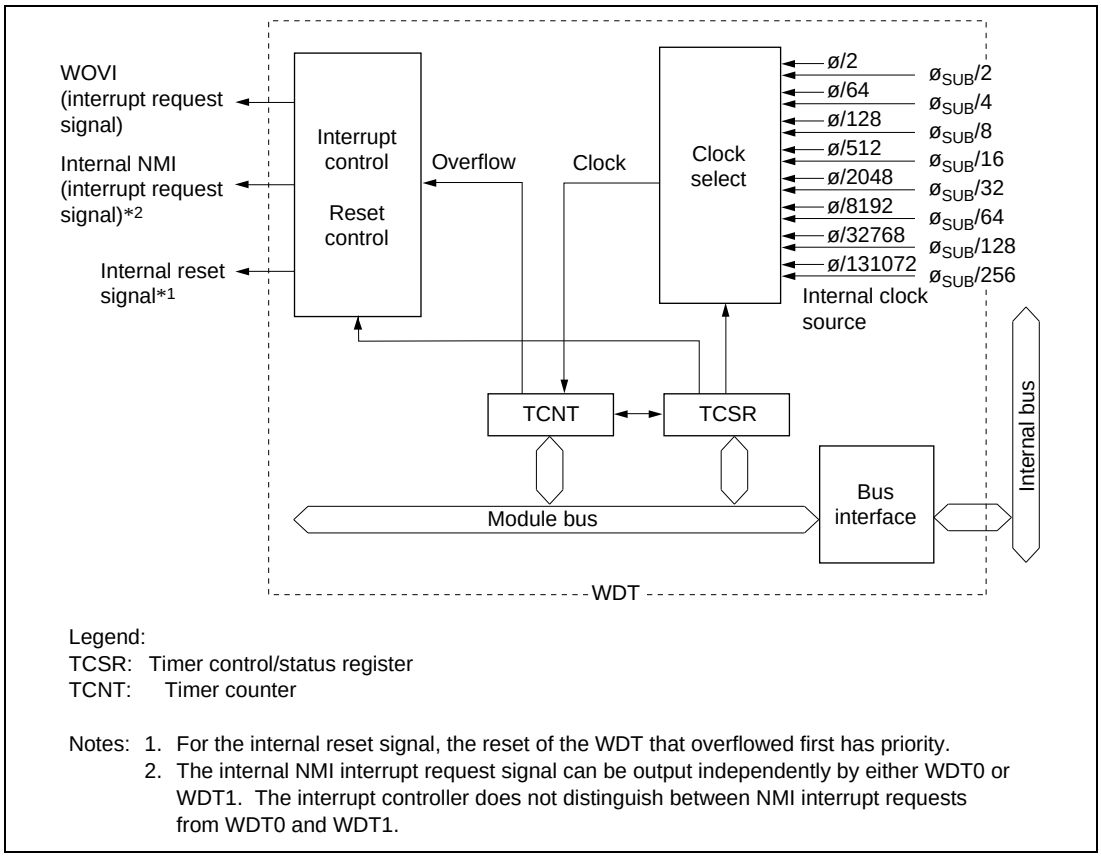
Figures 14.1 (a) and (b) show block diagrams of WDT0 and WDT1.



Legend:
TCSR: Timer control/status register
TCNT: Timer counter

Notes: 1. For the internal reset signal, the reset of the WDT that overflowed first has priority.
2. The internal NMI interrupt request signal can be output independently by either WDT0 or WDT1. The interrupt controller does not distinguish between NMI interrupt requests from WDT0 and WDT1.

Figure 14.1 (a) Block Diagram of WDT0



Legend:
TCSR: Timer control/status register
TCNT: Timer counter

Notes: 1. For the internal reset signal, the reset of the WDT that overflowed first has priority.
2. The internal NMI interrupt request signal can be output independently by either WDT0 or WDT1. The interrupt controller does not distinguish between NMI interrupt requests from WDT0 and WDT1.

Figure 14.1 (b) Block Diagram of WDT1

14.1.3 Pin Configuration

Table 14.1 describes the WDT input pin.

Table 14.1 WDT Pin

Name	Symbol	I/O	Function
External subclock input pin	EXCL	Input	WDT1 prescaler counter input clock

14.1.4 Register Configuration

The WDT has four registers, as summarized in table 14.2. These registers control clock selection, WDT mode switching, the reset signal, etc.

Table 14.2 WDT Registers

					Address*[1]	
Channel	**Name**	**Abbreviation**	**R/W**	**Initial Value**	**Write*[2]**	**Read**
0	Timer control/status register 0	TCSR0	R/(W)*[3]	H'00	H'FFA8	H'FFA8
	Timer counter 0	TCNT0	R/W	H'00	H'FFA8	H'FFA9
1	Timer control/status register 1	TCSR1	R/(W)*[3]	H'00	H'FFEA	H'FFEA
	Timer counter 1	TCNT1	R/W	H'00	H'FFEA	H'FFEB
Common	System control register	SYSCR	R/W	H'09	H'FFC4	H'FFC4

Notes: 1. Lower 16 bits of the address.
2. For details of write operations, see section 14.2.4, Notes on Register Access.
3. Only 0 can be written in bit 7, to clear the flag.

14.2 Register Descriptions

14.2.1 Timer Counter (TCNT)

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT is an 8-bit readable/writable* up-counter.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from the internal clock source selected by bits CKS2 to CKS0 in TCSR. When the TCNT value overflows (changes from H'FF to H'00), the OVF flag in TCSR is set to 1, and an internal reset, NMI interrupt, interval timer interrupt (WOVI), etc., can be generated, according to the mode selected by the WT/$\overline{\text{IT}}$ bit and RST/$\overline{\text{NMI}}$ bit.

TCNT is initialized to H'00 by a reset, in hardware standby mode, or when the TME bit is cleared to 0. It is not initialized in software standby mode.

Note: * The method of writing to TCNT is more complicated than for most other registers, to prevent accidental overwriting. For details see section 14.2.4, Notes on Register Access.

14.2.2 Timer Control/Status Register (TCSR)

- TCSR0

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	RSTS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

- TCSR1

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

TCSR is an 8-bit readable/writable* register. Its functions include selecting the clock source to be input to TCNT, and the timer mode.

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

Note: * The method of writing to TCSR is more complicated than for most other registers, to prevent accidental overwriting. For details see section 14.2.4, Notes on Register Access.

Bit 7—Overflow Flag (OVF): A status flag that indicates that TCNT has overflowed from H'FF to H'00.

Bit 7

OVF	Description	
0	[Clearing conditions]	
	• Write 0 in the TME bit	(Initial value)
	• Read TCSR when OVF = 1, then write 0 in OVF	
1	[Setting condition]	
	When TCNT overflows (changes from H'FF to H'00) (When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.)	

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether the WDT is used as a watchdog timer or interval timer. If used as an interval timer, the WDT generates an interval timer interrupt request (WOVI) when TCNT overflows. If used as a watchdog timer, the WDT generates a reset or NMI interrupt when TCNT overflows.

Bit 6

WT/$\overline{IT}$	Description	
0	Interval timer: Sends the CPU an interval timer interrupt request (WOVI) when TCNT overflows	(Initial value)
1	Watchdog timer: Generates a reset or NMI interrupt when TCNT overflows	

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	(Initial value)
1	TCNT counts	

TCSR0 Bit 4—Reset Select (RSTS): Reserved. This bit should not be set to 1.

TCSR1 Bit 4—Prescaler Select (PSS): Selects the input clock source for TCNT in WDT1. For details, see the description of the CKS2 to CKS0 bits below.

TCSR1 Bit 4		
PSS	**Description**	
0	TCNT counts ø-based prescaler (PSM) divided clock pulses	(Initial value)
1	TCNT counts øSUB-based prescaler (PSS) divided clock pulses	

Bit 3—Reset or NMI (RST/$\overline{\text{NMI}}$): Specifies whether an internal reset or NMI interrupt is requested on TCNT overflow in watchdog timer mode.

Bit 3		
RST/$\overline{\text{NMI}}$	**Description**	
0	An NMI interrupt is requested	(Initial value)
1	An internal reset is requested	

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select an internal clock source, obtained by dividing the system clock (ø), or subclock (øSUB) for input to TCNT.

- WDT0 input clock selection

Bit 2	Bit 1	Bit 0	Description	
CKS2	**CKS1**	**CKS0**	**Clock**	**Overflow Period* (when ø = 20 MHz)**
0	0	0	ø/2 (Initial value)	25.6 µs
		1	ø/64	819.2 µs
	1	0	ø/128	1.6 ms
		1	ø/512	6.6 ms
1	0	0	ø/2048	26.2 ms
		1	ø/8192	104.9 ms
	1	0	ø/32768	419.4 ms
		1	ø/131072	1.68 s

Note: * The overflow period is the time from when TCNT starts counting up from H'00 until overflow occurs.

- WDT1 input clock selection

Bit 4	Bit 2	Bit 1	Bit 0	Description	
PSS	**CKS2**	**CKS1**	**CKS0**	**Clock**	**Overflow Period* (when ø = 20 MHz and øSUB = 32.768 kHz)**
0	0	0	0	ø/2 (Initial value)	25.6 µs
			1	ø/64	819.2 µs
		1	0	ø/128	1.6 ms
			1	ø/512	6.6 ms
	1	0	0	ø/2048	26.2 ms
			1	ø/8192	104.9 ms
		1	0	ø/32768	419.4 ms
			1	ø/131072	1.68 s
1	0	0	0	øSUB/2	15.6 ms
			1	øSUB/4	31.3 ms
		1	0	øSUB/8	62.5 ms
			1	øSUB/16	125 ms
	1	0	0	øSUB/32	250 ms
			1	øSUB/64	500 ms
		1	0	øSUB/128	1 s
			1	øSUB/256	2 s

Note: * The overflow period is the time from when TCNT starts counting up from H'00 until overflow occurs.

14.2.3 System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
	CS2E	IOSE	INTM1	INTM0	XRST	NMIEG	HIE	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R	R/W	R	R/W	R/W	R/W

Only bit 3 is described here. For details on functions not related to the watchdog timer, see sections 3.2.2 and 5.2.1, System Control Register (SYSCR), and the descriptions of the relevant modules.

Bit 3—External Reset (XRST): Indicates the reset source. When the watchdog timer is used, a reset can be generated by watchdog timer overflow in addition to external reset input. XRST is a read-only bit. It is set to 1 by an external reset, and when the RST/$\overline{\text{NMI}}$ bit is 1, is cleared to 0 by an internal reset due to watchdog timer overflow.

Bit 3

XRST	Description	
0	Reset is generated by an internal reset due to watchdog timer overflow	
1	Reset is generated by external reset input	(Initial value)

14.2.4 Notes on Register Access

The watchdog timer's TCNT and TCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

Writing to TCNT and TCSR (Example of WDT0): These registers must be written to by a word transfer instruction. They cannot be written to with byte transfer instructions.

Figure 14.2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. For a write to TCNT, the upper byte of the written word must contain H'5A and the lower byte must contain the write data. For a write to TCSR, the upper byte of the written word must contain H'A5 and the lower byte must contain the write data. This transfers the write data from the lower byte to TCNT or TCSR.

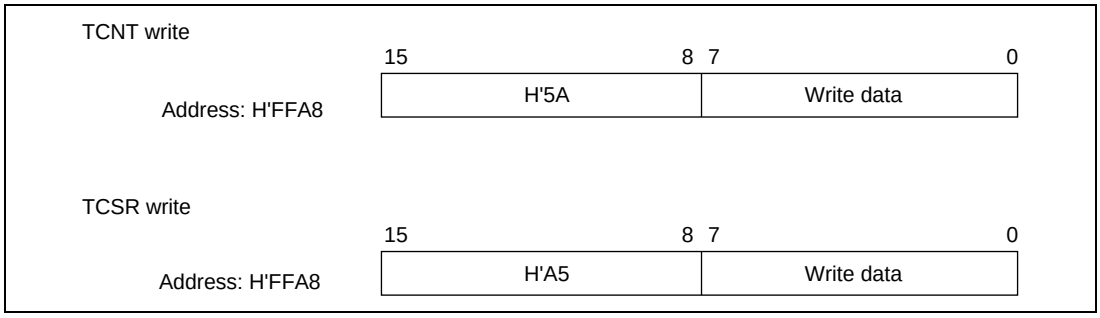


Figure 14.2 Format of Data Written to TCNT and TCSR (Example of WDT0)

Reading TCNT and TCSR (Example of WDT0): These registers are read in the same way as other registers. The read addresses are H'FFA8 for TCSR, and H'FFA9 for TCNT.

14.3 Operation

14.3.1 Watchdog Timer Operation

To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ and TME bits in TCSR to 1. Software must prevent TCNT overflows by rewriting the TCNT value (normally by writing H'00) before overflow occurs. This ensures that TCNT does not overflow while the system is operating normally. If TCNT overflows without being rewritten because of a system crash or other error, an internal reset or NMI interrupt request is generated.

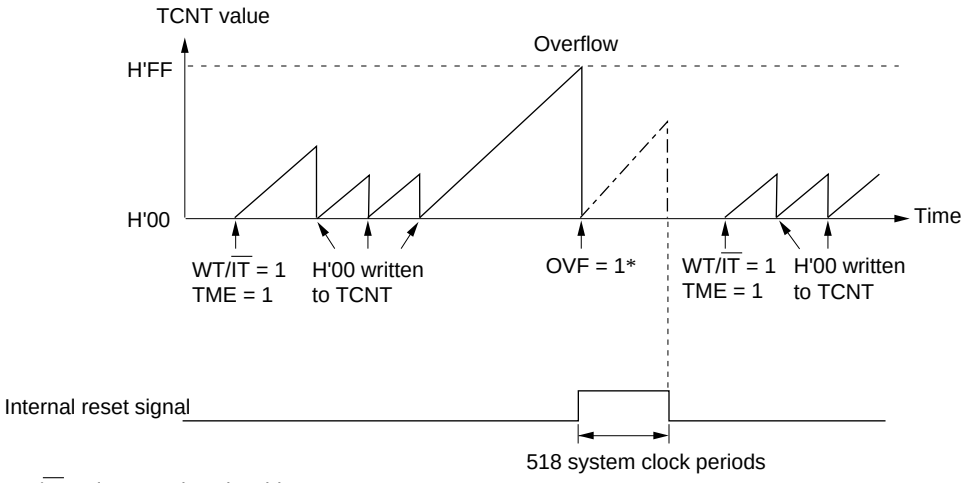
When the RST/$\overline{NMI}$ bit is set to 1, the chip is reset for 518 system clock periods (518 ø) by a counter overflow. This is illustrated in figure 14.3.

When the RST/$\overline{\text{NMI}}$ bit cleared to 0, an NMI interrupt request is generated by a counter overflow.

An internal reset request from the watchdog timer and reset input from the $\overline{\text{RES}}$ pin are handled via the same vector. The reset source can be identified from the value of the XRST bit in SYSCR.

If a reset caused by an input signal from the $\overline{\text{RES}}$ pin and a reset caused by WDT overflow occur simultaneously, the $\overline{\text{RES}}$ pin reset has priority, and the XRST bit in SYSCR is set to 1.

An NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin are handled via the same vector. Simultaneous handling of a watchdog timer NMI interrupt request and an NMI pin interrupt request must therefore be avoided.



WT/$\overline{\text{IT}}$: Timer mode select bit
TME: Timer enable bit
OVF: Overflow flag
Note: * Cleared to 0 by an internal reset when OVF is set to 1. XRST is cleared to 0.

Figure 14.3 Operation in Watchdog Timer Mode

14.3.2 Interval Timer Operation

To use the WDT as an interval timer, clear the WT/$\overline{IT}$ bit in TCSR to 0 and set the TME bit to 1. An interval timer interrupt (WOVI) is generated each time TCNT overflows, provided that the WDT is operating as an interval timer, as shown in figure 14.4. This function can be used to generate interrupt requests at regular intervals.

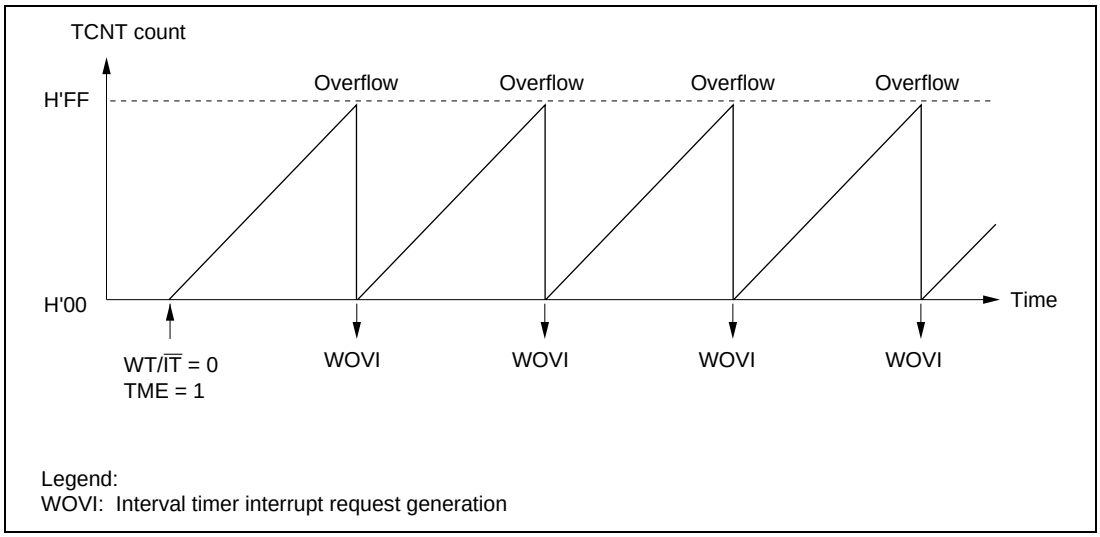


Figure 14.4 Operation in Interval Timer Mode

14.3.3 Timing of Setting of Overflow Flag (OVF)

The OVF bit in TCSR is set to 1 if TCNT overflows during interval timer operation. At the same time, an interval timer interrupt (WOVI) is requested. This timing is shown in figure 14.5.

If NMI request generation is selected in watchdog timer mode, when TCNT overflows the OVF bit in TCSR is set to 1 and at the same time an NMI interrupt is requested.

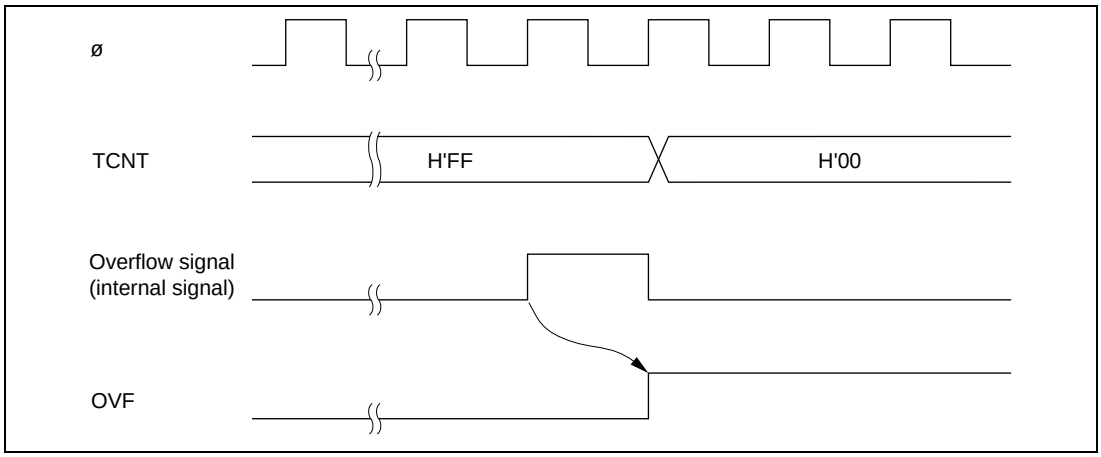


Figure 14.5 Timing of OVF Setting

14.4 Interrupts

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine. When NMI interrupt request generation is selected in watchdog timer mode, an overflow generates an NMI interrupt request.

14.5 Usage Notes

14.5.1 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 14.6 shows this operation.

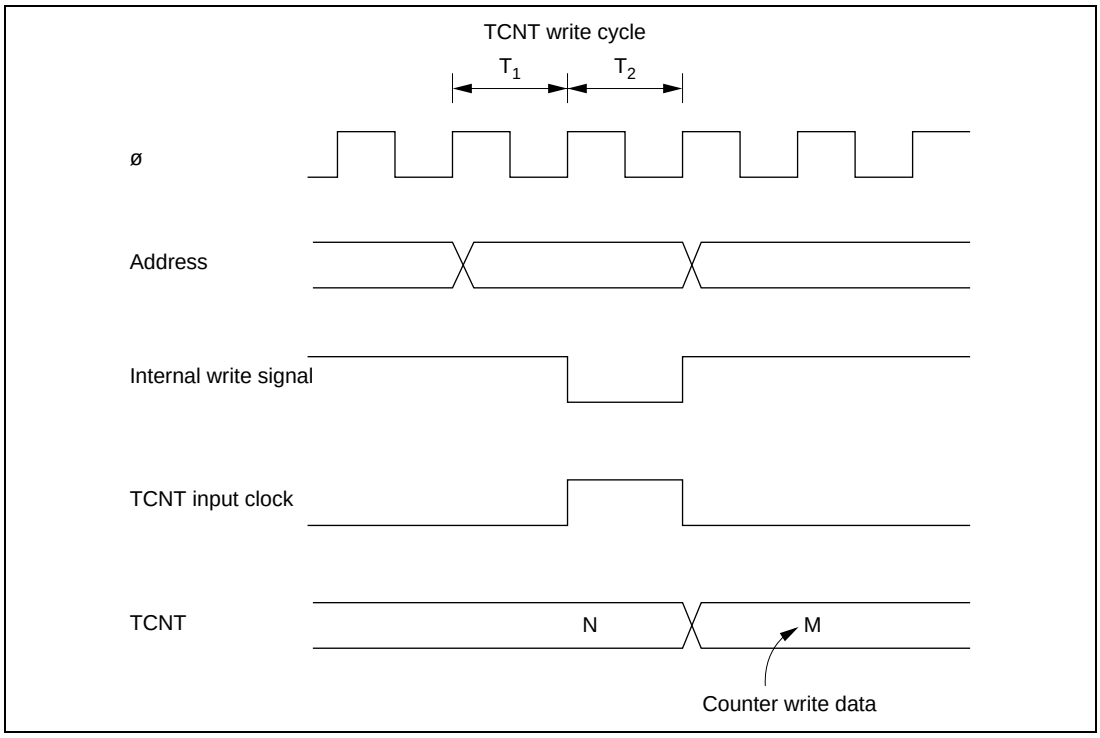


Figure 14.6 Contention between TCNT Write and Increment

14.5.2 Changing Value of CKS2 to CKS0

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits CKS2 to CKS0.

14.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, or vice versa, while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

14.5.4 Counter Value in Transitions between High-Speed Mode, Subactive Mode, and Watch Mode

If the mode is switched between high-speed mode and subactive mode or between high-speed mode and watch mode when WDT1 is used as a realtime clock counter, an error will occur in the counter value when the internal clock is switched.

When the mode is switched from high-speed mode to subactive mode or watch mode, the increment timing is delayed by approximately 2 or 3 clock cycles when the WDT1 control clock is switched from the main clock to the subclock.

Also, since the main clock oscillator is halted during subclock operation, when the mode is switched from watch mode or subactive mode to high-speed mode, the clock is not supplied until internal oscillation stabilizes. As a result, after oscillation is started, counter incrementing is halted during the oscillation stabilization time set by bits STS2 to STS0 in SBYCR, and there is a corresponding discrepancy in the counter value.

Caution is therefore required when using WDT1 as the realtime clock counter.

No error occurs in the counter value while WDT1 is operating in the same mode.

Section 15 Serial Communication Interface (SCI)

15.1 Overview

These series are equipped with a serial communication interface (SCI) with two independent channels. The SCI can handle both asynchronous and clocked synchronous serial communication. A function is also provided for serial communication between processors (multiprocessor communication function).

15.1.1 Features

SCI features are listed below.

- Choice of asynchronous or synchronous serial communication mode

  Asynchronous mode

  — Serial data communication is executed using an asynchronous system in which synchronization is achieved character by character

  Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA)

  — A multiprocessor communication function is provided that enables serial data communication with a number of processors

  — Choice of 12 serial data transfer formats

Data length:	7 or 8 bits
Stop bit length:	1 or 2 bits
Parity:	Even, odd, or none
Multiprocessor bit:	1 or 0

  — Receive error detection: Parity, overrun, and framing errors

  — Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error

  Synchronous mode

  — Serial data communication is synchronized with a clock

  Serial data communication can be carried out with other chips that have a synchronous communication function

  — One serial data transfer format

Data length:	8 bits

  — Receive error detection: Overrun errors detected

- Full-duplex communication capability
  - The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously
  - Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data
- LSB-first or MSB-first transfer can be selected
  - This selection can be made regardless of the communication mode (with the exception of 7-bit data transfer in asynchronous mode)*

Note: * LSB-first transfer is used in the examples in this section.

- Built-in baud rate generator allows any bit rate to be selected
- Choice of serial clock source: internal clock from baud rate generator or external clock from SCK pin
- Four interrupt sources
  - Four interrupt sources (transmit-data-empty, transmit-end, receive-data-full, and receive error) that can issue requests independently
  - The transmit-data-empty interrupt and receive-data-full interrupt can activate the data transfer controller (DTC) to execute data transfer

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the SCI.

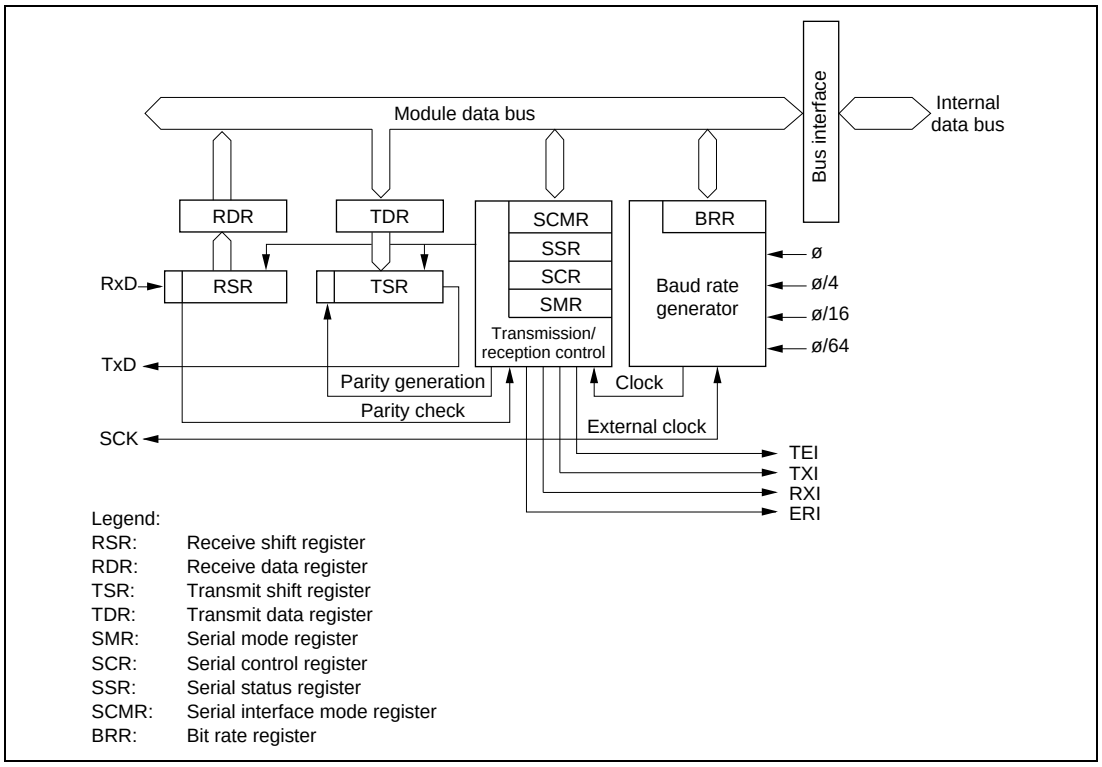


Figure 15.1 Block Diagram of SCI

15.1.3 Pin Configuration

Table 15.1 shows the serial pins used by the SCI.

Table 15.1 SCI Pins

Channel	Pin Name	Symbol*	I/O	Function
0	Serial clock pin 0	SCK0	I/O	SCI0 clock input/output
	Receive data pin 0	RxD0	Input	SCI0 receive data input
	Transmit data pin 0	TxD0	Output	SCI0 transmit data output
1	Serial clock pin 1	SCK1	I/O	SCI1 clock input/output
	Receive data pin 1	RxD1	Input	SCI1 receive data input
	Transmit data pin 1	TxD1	Output	SCI1 transmit data output

Note: * The abbreviations SCK, RxD, and TxD are used in the text, omitting the channel number.

15.1.4 Register Configuration

The SCI has the internal registers shown in table 15.2. These registers are used to specify asynchronous mode or synchronous mode, the data format, and the bit rate, and to control the transmitter/receiver.

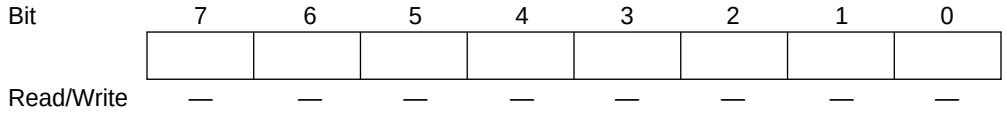
Table 15.2 SCI Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
0	Serial mode register 0	SMR0	R/W	H'00	H'FFD8*[3]
	Bit rate register 0	BRR0	R/W	H'FF	H'FFD9*[3]
	Serial control register 0	SCR0	R/W	H'00	H'FFDA
	Transmit data register 0	TDR0	R/W	H'FF	H'FFDB
	Serial status register 0	SSR0	R/(W)*[2]	H'84	H'FFDC
	Receive data register 0	RDR0	R	H'00	H'FFDD
	Serial interface mode register 0	SCMR0	R/W	H'F2	H'FFDE*[3]
1	Serial mode register 1	SMR1	R/W	H'00	H'FF83*[3]
	Bit rate register 1	BRR1	R/W	H'FF	H'FF89*[3]
	Serial control register 1	SCR1	R/W	H'00	H'FF8A
	Transmit data register 1	TDR1	R/W	H'FF	H'FF8B
	Serial status register 1	SSR1	R/(W)*[2]	H'84	H'FF8C
	Receive data register 1	RDD1	R	H'00	H'FF8D
	Serial interface mode register 1	SCMR1	R/W	H'F2	H'FF8E*[3]
Common	Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
		MSTPCRL	R/W	H'FF	H'FF87

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written, to clear flags.
3. Some serial communication interface registers are assigned to the same addresses as other registers. In this case, register selection is performed by the IICE bit in the serial timer control register (STCR).

15.2 Register Descriptions

15.2.1 Receive Shift Register (RSR)

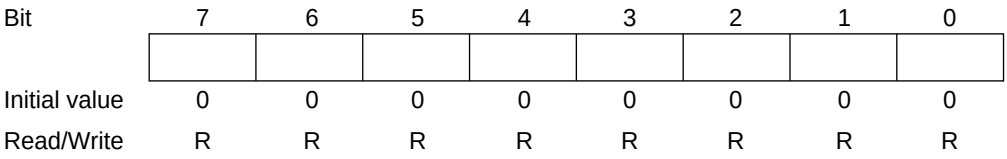


RSR is a register used to receive serial data.

The SCI sets serial data input from the RxD pin in RSR in the order received, starting with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to RDR automatically.

RSR cannot be directly read or written to by the CPU.

15.2.2 Receive Data Register (RDR)



RDR is a register that stores received serial data.

When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR where it is stored, and completes the receive operation. After this, RSR is receive-enabled.

Since RSR and RDR function as a double buffer in this way, continuous receive operations can 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, watch mode, subactive mode, subsleep mode, and module stop mode.

15.2.3 Transmit Shift Register (TSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

TSR is a register used to transmit serial data.

To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from TDR to TSR, and transmission started, automatically. However, data transfer from TDR to TSR is not performed if the TDRE bit in SSR is set to 1.

TSR cannot be directly read or written to by the CPU.

15.2.4 Transmit Data Register (TDR)

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TDR is an 8-bit register that stores data for serial transmission.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts serial transmission. Continuous serial transmission can be carried out by writing the next transmit data to TDR during serial transmission of the data in TSR.

TDR can be read or written to by the CPU at all times.

TDR is initialized to H'FF by a reset, and in standby mode, watch mode, subactive mode, subsleep mode, and module stop mode.

15.2.5 Serial Mode Register (SMR)

Bit	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SMR is an 8-bit register used to set the SCI's serial transfer format and select the baud rate generator clock source.

SMR can be read or written to by the CPU at all times.

SMR is initialized to H'00 by a reset, and in standby mode, watch mode, subactive mode, subsleep mode, and module stop mode.

Bit 7—Communication Mode (C/$\overline{A}$): Selects asynchronous mode or synchronous mode as the SCI operating mode.

Bit 7

C/$\overline{A}$	Description	
0	Asynchronous mode	(Initial value)
1	Synchronous mode	

Bit 6—Character Length (CHR): Selects 7 or 8 bits as the data length in asynchronous mode. In synchronous mode, a fixed data length of 8 bits is used regardless of the CHR setting.

Bit 6

CHR	Description	
0	8-bit data	(Initial value)
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and LSB-first/MSB-first selection is not available.

Bit 5—Parity Enable (PE): In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking in reception. In synchronous mode, or when a multiprocessor format is used, parity bit addition and checking is not performed, regardless of the PE bit setting.

Bit 5

PE	Description	
0	Parity bit addition and checking disabled	(Initial value)
1	Parity bit addition and checking enabled*	

Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

Bit 4—Parity Mode (O/$\overline{E}$): Selects either even or odd parity for use in parity addition and checking.

The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking, in asynchronous mode. The O/$\overline{E}$ bit setting is invalid in synchronous mode, when parity bit addition and checking is disabled in asynchronous mode, and when a multiprocessor format is used.

Bit 4

O/$\overline{E}$	Description	
0	Even parity*[1]	(Initial value)
1	Odd parity*[2]	

Notes: 1. When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is even.
2. When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.

Bit 3—Stop Bit Length (STOP): Selects 1 or 2 bits as the stop bit length in asynchronous mode. The STOP bit setting is only valid in asynchronous mode. If synchronous mode is set the STOP bit setting is invalid since stop bits are not added.

Bit 3

STOP	Description	
0	1 stop bit*[1]	(Initial value)
1	2 stop bits*[2]	

Notes: 1. In transmission, a single 1 bit (stop bit) is added to the end of a transmit character before it is sent.
2. In transmission, two 1 bits (stop bits) are added to the end of a transmit character before it is sent.

In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next transmit character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, the PE bit and O/$\overline{E}$ bit parity settings are invalid. The MP bit setting is only valid in asynchronous mode; it is invalid in synchronous mode.

For details of the multiprocessor communication function, see section 15.3.3, Multiprocessor Communication Function.

Bit 2

MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source for the baud rate generator. The clock source can be selected from ø, ø/4, ø/16, and ø/64, according to the setting of bits CKS1 and CKS0.

For the relation between the clock source, the bit rate register setting, and the baud rate, see section 15.2.8, Bit Rate Register.

Bit 1	Bit 0		
CKS1	**CKS0**	**Description**	
0	0	ø clock	(Initial value)
	1	ø/4 clock	
1	0	ø/16 clock	
	1	ø/64 clock	

15.2.6 Serial Control Register (SCR)

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR is a register that performs enabling or disabling of SCI transfer operations, serial clock output in asynchronous mode, and interrupt requests, and selection of the serial clock source.

SCR can be read or written to by the CPU at all times.

SCR is initialized to H'00 by a reset, and in standby mode, watch mode, subactive mode, subsleep mode, and module stop mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit-data-empty interrupt (TXI) request generation when serial transmit data is transferred from TDR to TSR and the TDRE flag in SSR is set to 1.

Bit 7		
TIE	**Description**	
0	Transmit-data-empty interrupt (TXI) request disabled*	(Initial value)
1	Transmit-data-empty interrupt (TXI) request enabled	

Note: * TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request generation when serial receive data is transferred from RSR to RDR and the RDRF flag in SSR is set to 1.

Bit 6

RIE	Description
0	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request disabled* (Initial value)
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

Note: * RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF, FER, PER, or ORER flag, then clearing the flag to 0, or clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of serial transmission by the SCI.

Bit 5

TE	Description
0	Transmission disabled*[1] (Initial value)
1	Transmission enabled*[2]

Notes: 1. The TDRE flag in SSR is fixed at 1.
2. In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0.
SMR setting must be performed to decide the transmission format before setting the TE bit to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of serial reception by the SCI.

Bit 4

RE	Description
0	Reception disabled*[1] (Initial value)
1	Reception enabled*[2]

Notes: 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.
2. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in synchronous mode.
SMR setting must be performed to decide the reception format before setting the RE bit to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE bit setting is only valid in asynchronous mode when receiving with the MP bit in SMR set to 1.

The MPIE bit setting is invalid in synchronous mode or when the MP bit is cleared to 0.

Bit 3

MPIE	Description	
0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received	(Initial value)
1	Multiprocessor interrupts enabled* Receive interrupt (RXI) requests, receive-error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.	

Note: * When receive data including MPB = 0 is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and ORER flags in SSR , is not performed. When receive data with MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting is enabled.

Bit 2—Transmit End Interrupt Enable (TEIE): Enables or disables transmit-end interrupt (TEI) request generation if there is no valid transmit data in TDR when the MSB is transmitted.

Bit 2

TEIE	Description	
0	Transmit-end interrupt (TEI) request disabled*	(Initial value)
1	Transmit-end interrupt (TEI) request enabled*	

Note: * TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the SCI clock source and enable or disable clock output from the SCK pin. The combination of the CKE1 and CKE0 bits determines whether the SCK pin functions as an I/O port, the serial clock output pin, or the serial clock input pin.

The setting of the CKE0 bit, however, is only valid for internal clock operation (CKE1 = 0) in asynchronous mode. The CKE0 bit setting is invalid in synchronous mode, and in the case of external clock operation (CKE1 = 1). The setting of bits CKE1 and CKE0 must be carried out before the SCI's operating mode is determined using SMR.

For details of clock source selection, see table 15.9 in section 15.3, Operation.

Bit 1	Bit 0		
CKE1	**CKE0**	**Description**	
0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port*[1]
		Synchronous mode	Internal clock/SCK pin functions as serial clock output*[1]
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*[2]
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*[3]
		Synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*[3]
		Synchronous mode	External clock/SCK pin functions as serial clock input

Notes: 1. Initial value
2. Outputs a clock of the same frequency as the bit rate.
3. Inputs a clock with a frequency 16 times the bit rate.

15.2.7 Serial Status Register (SSR)

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

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

SSR is an 8-bit register containing status flags that indicate the operating status of the SCI, and multiprocessor bits.

SSR can be read or written to by the CPU at all times. However, 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER. Also note that in order to clear these flags they must be read as 1 beforehand. The TEND flag and MPB flag are read-only flags and cannot be modified.

SSR is initialized to H'84 by a reset, and in standby mode, watch mode, subactive mode, subsleep mode, and module stop mode.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that data has been transferred from TDR to TSR and the next serial data can be written to TDR.

Bit 7 TDRE	Description
0	[Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Bit 6—Receive Data Register Full (RDRF): Indicates that the received data is stored in RDR.

Bit 6	
RDRF	**Description**
0	[Clearing conditions] (Initial value) • When 0 is written in RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and reads data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Note: RDR and the RDRF flag are not affected and retain their previous values when an error is detected during reception or when the RE bit in SCR is cleared to 0.
If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that an overrun error occurred during reception, causing abnormal termination.

Bit 5	
ORER	**Description**
0	[Clearing condition] (Initial value)*[1] When 0 is written in ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1*[2]

Notes: 1. The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
2. The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In synchronous mode, serial transmission cannot be continued, either.

Bit 4—Framing Error (FER): Indicates that a framing error occurred during reception in asynchronous mode, causing abnormal termination.

Bit 4

FER	Description
0	[Clearing condition] (Initial value)*[1] When 0 is written in FER after reading FER = 1
1	[Setting condition] When the SCI checks the stop bit at the end of the receive data when reception ends, and the stop bit is 0 *[2]

Notes: 1. The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

2. In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the FER flag is set to 1. In synchronous mode, serial transmission cannot be continued, either.

Bit 3—Parity Error (PER): Indicates that a parity error occurred during reception using parity addition in asynchronous mode, causing abnormal termination.

Bit 3

PER	Description
0	[Clearing condition] (Initial value)*[1] When 0 is written in PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR*[2]

Notes: 1. The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

2. If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In synchronous mode, serial transmission cannot be continued, either.

Bit 2—Transmit End (TEND): Indicates that there is no valid data in TDR when the last bit of the transmit character is sent, and transmission has been ended.

The TEND flag is read-only and cannot be modified.

Bit 2

TEND	Description
0	[Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Bit 1—Multiprocessor Bit (MPB): When reception is performed using a multiprocessor format in asynchronous mode, MPB stores the multiprocessor bit in the receive data.

MPB is a read-only bit, and cannot be modified.

Bit 1

MPB	Description
0	[Clearing condition] (Initial value)* When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Note: * Retains its previous state when the RE bit in SCR is cleared to 0 with multiprocessor format.

Bit 0—Multiprocessor Bit Transfer (MPBT): When transmission is performed using a multiprocessor format in asynchronous mode, MPBT stores the multiprocessor bit to be added to the transmit data.

The MPBT bit setting is invalid when a multiprocessor format is not used, when not transmitting, and in synchronous mode.

Bit 0

MPBT	Description
0	Data with a 0 multiprocessor bit is transmitted (Initial value)
1	Data with a 1 multiprocessor bit is transmitted

15.2.8 Bit Rate Register (BRR)

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BRR is an 8-bit register that sets the serial transfer bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SMR.

BRR can be read or written to by the CPU at all times.

BRR is initialized to H'FF by a reset, and in standby mode, watch mode, subactive mode, subsleep mode, and module stop mode.

As baud rate generator control is performed independently for each channel, different values can be set for each channel.

Table 15.3 shows sample BRR settings in asynchronous mode, and table 15.4 shows sample BRR settings in synchronous mode.

Table 15.3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	Operating Frequency ø (MHz)											
	ø = 2 MHz			ø = 2.097152 MHz			ø = 2.4576 MHz			ø = 3 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19	–2.34
9600	—	—	—	0	6	–2.48	0	7	0.00	0	9	–2.34
19200	—	—	—	—	—	—	0	3	0.00	0	4	–2.34
31250	0	1	0.00	—	—	—	—	—	—	0	2	0.00
38400	—	—	—	—	—	—	0	1	0.00	—	—	—

	Operating Frequency ø (MHz)											
	ø = 3.6864 MHz			ø = 4 MHz			ø = 4.9152 MHz			ø = 5 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	—	—	—	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4	0.00
38400	0	2	0.00	—	—	—	0	3	0.00	0	3	1.73

Table 15.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (cont)

	Operating Frequency ø (MHz)											
	ø = 6 MHz			ø = 6.144 MHz			ø = 7.3728 MHz			ø = 8 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	—	—	—	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	—	—	—

	Operating Frequency ø (MHz)											
	ø = 9.8304 MHz			ø = 10 MHz			ø = 12 MHz			ø = 12.288 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Table 15.3 BRR Settings for Various Bit Rates (Asynchronous Mode) (cont)

	Operating Frequency ø (MHz)											
	ø = 14 MHz			ø = 14.7456 MHz			ø = 16 MHz			ø = 17.2032 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	75	0.48
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16	1.20
38400	—	—	—	0	11	0.00	0	12	0.16	0	13	0.00

	Operating Frequency ø (MHz)								
	ø = 18 MHz			ø = 19.6608 MHz			ø = 20 MHz		
Bit Rate (bits/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

Table 15.4 BRR Settings for Various Bit Rates (Synchronous Mode)

Bit Rate (bits/s)	Operating Frequency ø (MHz)											
	ø = 2 MHz		ø = 4 MHz		ø = 8 MHz		ø = 10 MHz		ø = 16 MHz		ø = 20 MHz	
	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—								
250	2	124	2	249	3	124	—	—	3	249		
500	1	249	2	124	2	249	—	—	3	124	—	—
1 k	1	124	1	249	2	124	—	—	2	249	—	—
2.5 k	0	199	1	99	1	199	1	249	2	99	2	124
5 k	0	99	0	199	1	99	1	124	1	199	1	249
10 k	0	49	0	99	0	199	0	249	1	99	1	124
25 k	0	19	0	39	0	79	0	99	0	159	0	199
50 k	0	9	0	19	0	39	0	49	0	79	0	99
100 k	0	4	0	9	0	19	0	24	0	39	0	49
250 k	0	1	0	3	0	7	0	9	0	15	0	19
500 k	0	0*	0	1	0	3	0	4	0	7	0	9
1 M			0	0*	0	1			0	3	0	4
2.5 M							0	0*			0	1
5 M											0	0*

Note: As far as possible, the setting should be made so that the error is no more than 1%.

Legend:

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

*: Continuous transfer is not possible.

The BRR setting is found from the following equations.

Asynchronous mode:

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Synchronous mode:

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bits/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ø: Operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ø	0	0
1	ø/4	0	1
2	ø/16	1	0
3	ø/64	1	1

The bit rate error in asynchronous mode is found from the following equation:

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 15.5 shows the maximum bit rate for each frequency in asynchronous mode. Tables 15.6 and 15.7 show the maximum bit rates with external clock input.

Table 15.5 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ø (MHz)	Maximum Bit Rate (bits/s)	n	N
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
5	156250	0	0
6	187500	0	0
6.144	192000	0	0
7.3728	230400	0	0
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0

Table 15.6 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
5	1.2500	78125
6	1.5000	93750
6.144	1.5360	96000
7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500

Table 15.7 Maximum Bit Rate with External Clock Input (Synchronous Mode)

ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.3333	333333.3
4	0.6667	666666.7
6	1.0000	1000000.0
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3

15.2.9 Serial Interface Mode Register (SCMR)

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

SCMR is an 8-bit readable/writable register used to select SCI functions.

SCMR is initialized to H'F2 by a reset, and in standby mode, watch mode, subactive mode, subsleep mode, and module stop mode.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

Bit 3		
SDIR	**Description**	
0	TDR contents are transmitted LSB-first	(Initial value)
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

Bit 2—Data Invert (SINV): Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit(s): parity bit inversion requires inversion of the O/$\overline{E}$ bit in SMR.

Bit 2

SINV	Description	
0	TDR contents are transmitted without modification	(Initial value)
	Receive data is stored in RDR without modification	
1	TDR contents are inverted before being transmitted	
	Receive data is stored in RDR in inverted form	

Bit 1—Reserved: This bit cannot be modified and is always read as 1.

Bit 0—Serial Communication Interface Mode Select (SMIF): Reserved bit. 1 should not be written in this bit.

Bit 0

SMIF	Description	
0	Normal SCI mode	(Initial value)
1	Reserved mode	

15.2.10 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	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	R/W	R/W

MSTPCR, comprising two 8-bit readable/writable registers, performs module stop mode control. When bits MSTP7 and MSTP6 are set to 1, SCI0 and SCI1 operation, respectively, stops at the end of the bus cycle and a transition is made to module stop mode. For details, see section 21.5., Module Stop Mode.

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

Bit 7—Module Stop (MSTP7): Specifies the SCI0 module stop mode.

Bit 7		
MSTP7	**Description**	
0	SCI0 module stop mode is cleared	
1	SCI0 module stop mode is set	(Initial value)

Bit 6—Module Stop (MSTP6): Specifies the SCI1 module stop mode.

Bit 6		
MSTP6	**Description**	
0	SCI1 module stop mode is cleared	
1	SCI1 module stop mode is set	(Initial value)

15.3 Operation

15.3.1 Overview

The SCI can carry out serial communication in two modes: asynchronous mode in which synchronization is achieved character by character, and synchronous mode in which synchronization is achieved with clock pulses.

Selection of asynchronous or synchronous mode and the transmission format is made using SMR as shown in table 15.8. The SCI clock is determined by a combination of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 15.9.

Asynchronous Mode

- Data length: Choice of 7 or 8 bits
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bits (the combination of these parameters determines the transfer format and character length)
- Detection of framing, parity, and overrun errors, and breaks, during reception
- Choice of internal or external clock as SCI clock source
  - — When internal clock is selected:

    The SCI operates on the baud rate generator clock and a clock with the same frequency as the bit rate can be output
  - — When external clock is selected:

    A clock with a frequency of 16 times the bit rate must be input (the built-in baud rate generator is not used)

Synchronous Mode

- Transfer format: Fixed 8-bit data
- Detection of overrun errors during reception
- Choice of internal or external clock as SCI clock source
  - — When internal clock is selected:
    The SCI operates on the baud rate generator clock and a serial clock is output off-chip
  - — When external clock is selected:
    The built-in baud rate generator is not used, and the SCI operates on the input serial clock

Table 15.8 SMR Settings and Serial Transfer Format Selection

SMR Settings						SCI Transfer Format			
Bit 7	**Bit 6**	**Bit 2**	**Bit 5**	**Bit 3**		**Data**	**Multi-processor**	**Parity**	**Stop Bit**
C/$\overline{A}$	**CHR**	**MP**	**PE**	**STOP**	**Mode**	**Length**	**Bit**	**Bit**	**Length**
0	0	0	0	0	Asynchronous mode	8-bit data	No	No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	1		0	0		7-bit data		No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Yes	No	1 bit
			—	1					2 bits
	1		—	0		7-bit data			1 bit
			—	1					2 bits
1	—	—	—	—	Synchronous mode	8-bit data	No		None

Table 15.9 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Setting			SCI Transfer Clock	
Bit 7	Bit 1	Bit 0		Clock	
C/$\overline{A}$	CKE1	CKE0	Mode	Source	SCK Pin Function
0	0	0	Asynchronous mode	Internal	SCI does not use SCK pin
		1			Outputs clock with same frequency as bit rate
	1	0		External	Inputs clock with frequency of 16 times the bit rate
		1			
1	0	0	Synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

15.3.2 Operation in Asynchronous Mode

In asynchronous mode, characters are sent or received, each preceded by a start bit indicating the start of communication and followed by one or two stop bits indicating the end of communication. Serial communication is thus carried out with synchronization established on a character-by-character basis.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 15.2 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication.

One serial communication character consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally one or two stop bits (high level).

In asynchronous mode, the SCI performs synchronization at the falling edge of the start bit in reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16 times the length of one bit, so that the transfer data is latched at the center of each bit.

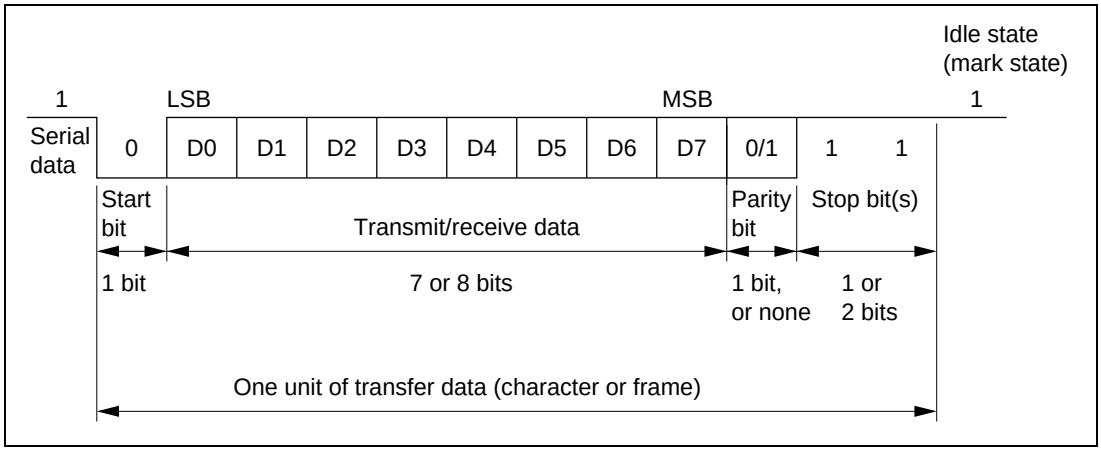


Figure 15.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

Data Transfer Format: Table 15.10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected by settings in SMR.

Table 15.10 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	S	8-bit data								STOP		
0	0	0	1	S	8-bit data								STOP	STOP	
0	1	0	0	S	8-bit data								P	STOP	
0	1	0	1	S	8-bit data								P	STOP	STOP
1	0	0	0	S	7-bit data							STOP			
1	0	0	1	S	7-bit data							STOP	STOP		
1	1	0	0	S	7-bit data							P	STOP		
1	1	0	1	S	7-bit data							P	STOP	STOP	
0	—	1	0	S	8-bit data								MPB	STOP	
0	—	1	1	S	8-bit data								MPB	STOP	STOP
1	—	1	0	S	7-bit data							MPB	STOP		
1	—	1	1	S	7-bit data							MPB	STOP	STOP	

Legend:
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Clock: Either an internal clock generated by the built-in baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 15.9.

When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is at the center of each transmit data bit, as shown in figure 15.3.

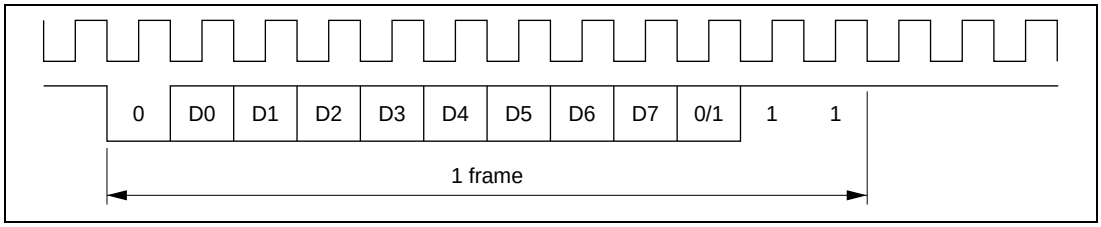


Figure 15.3 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

Data Transfer Operations

SCI Initialization (Asynchronous Mode): Before transmitting and receiving data, first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

When an external clock is used the clock should not be stopped during operation, including initialization, since operation is uncertain.

Figure 15.4 shows a sample SCI initialization flowchart.

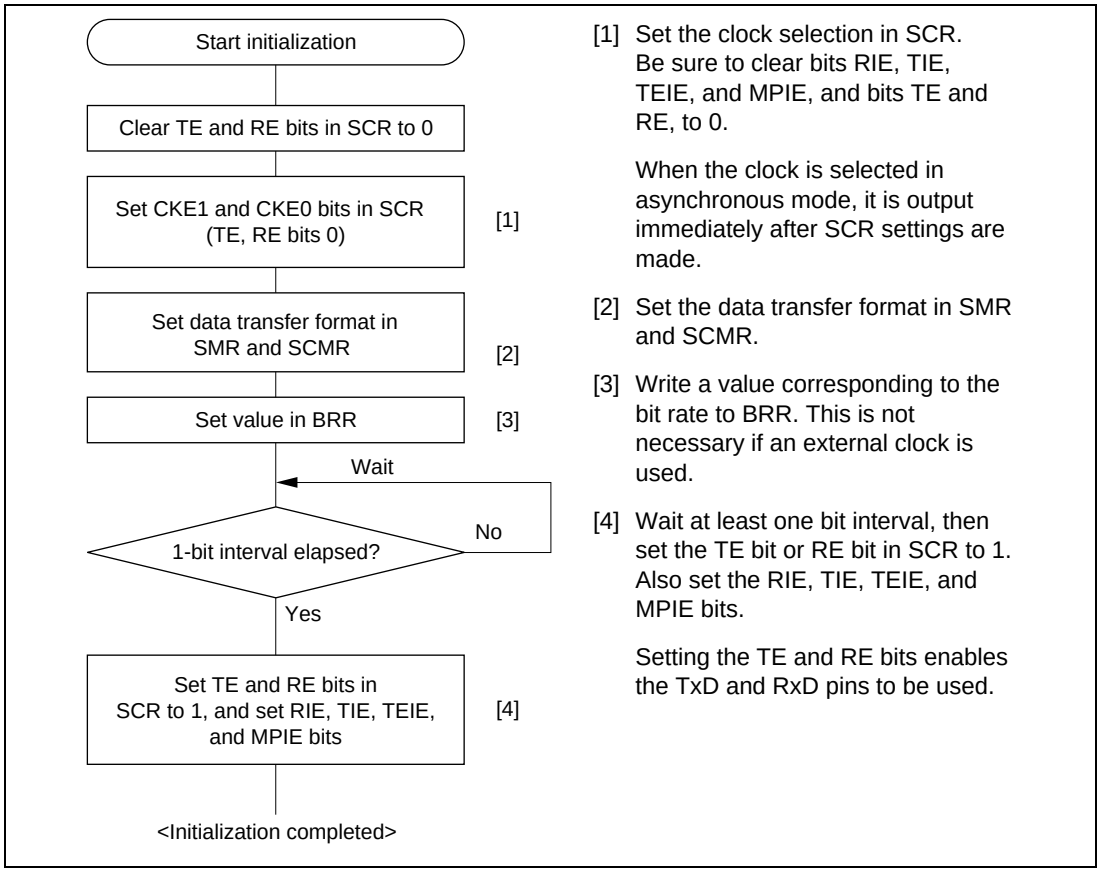
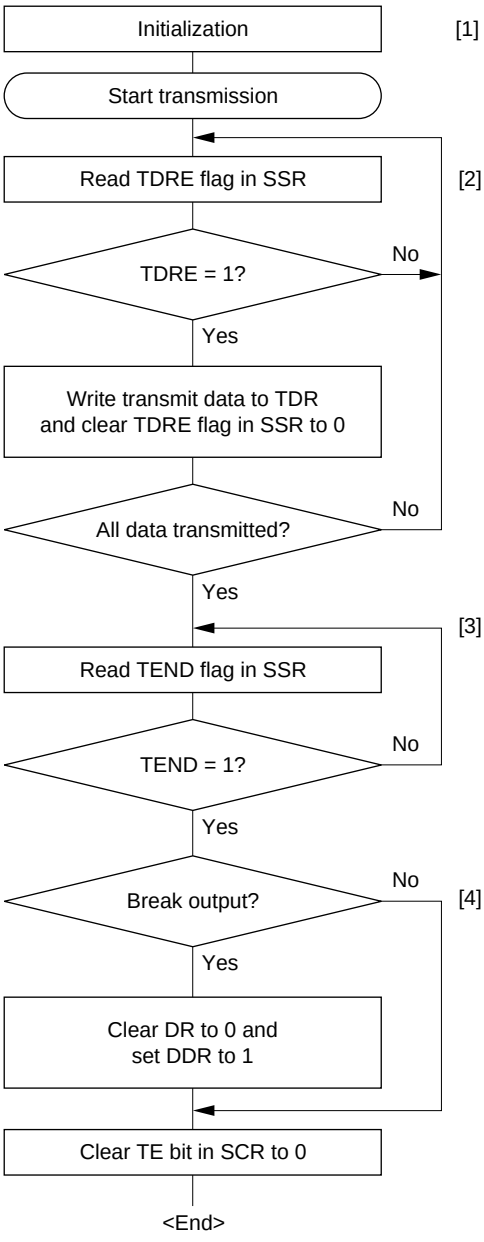


Figure 15.4 Sample SCI Initialization Flowchart

Serial Data Transmission (Asynchronous Mode): Figure 15.5 shows a sample flowchart for serial transmission.

The following procedure should be used for serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, one frame of 1s is output and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit-data-empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 15.5 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission.

   If the TIE bit is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.

   The serial transmit data is sent from the TxD pin in the following order.

   a. Start bit:

      One 0-bit is output.

   b. Transmit data:

      8-bit or 7-bit data is output in LSB-first order.

   c. Parity bit or multiprocessor bit:

      One parity bit (even or odd parity), or one multiprocessor bit is output.

      A format in which neither a parity bit nor a multiprocessor bit is output can also be selected.

   d. Stop bit(s):

      One or two 1-bits (stop bits) are output.

   e. Mark state:

      1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCI checks the TDRE flag at the timing for sending the stop bit.

   If the TDRE flag is cleared to 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 15.6 shows an example of the operation for transmission in asynchronous mode.

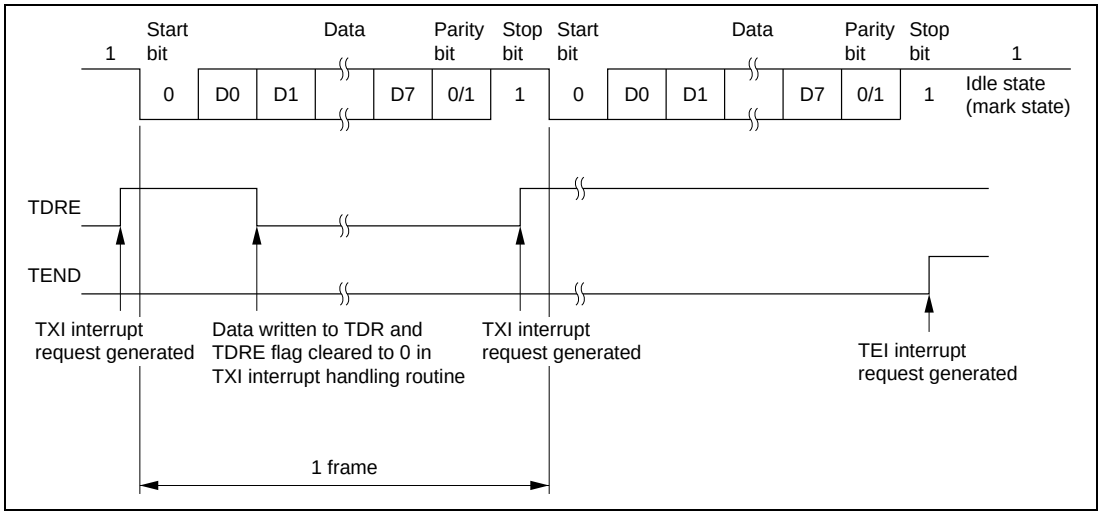
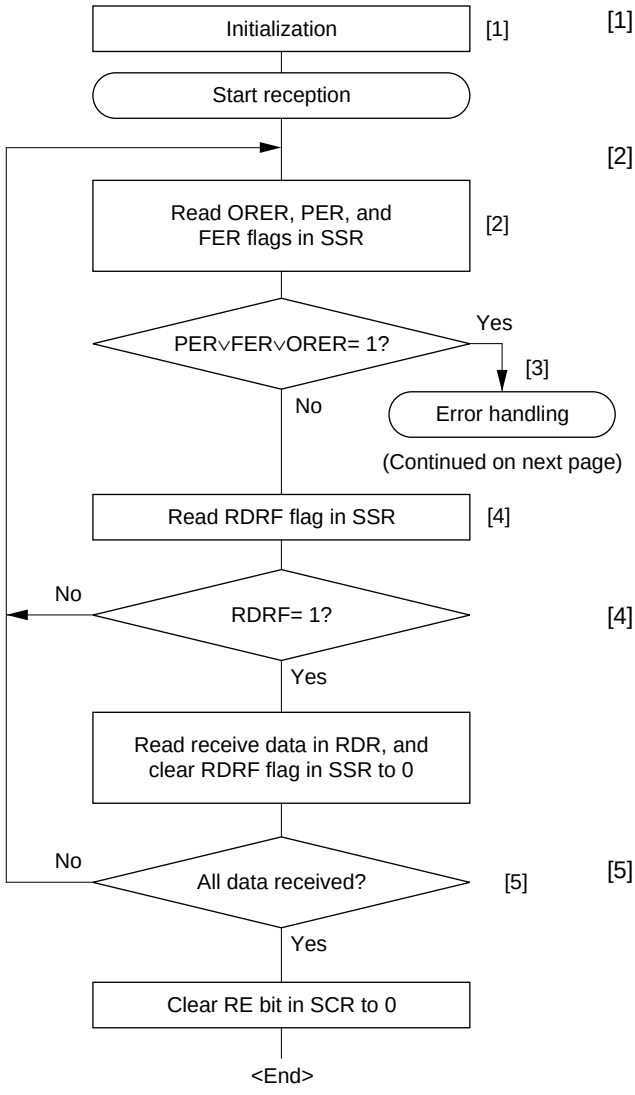


Figure 15.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

Serial Data Reception (Asynchronous Mode): Figure 15.7 shows a sample flowchart for serial reception.

The following procedure should be used for serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error handling and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error handling, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read :
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when the DTC is activated by an RXI interrupt and the RDR value is read.

Figure 15.7 Sample Serial Reception Data Flowchart

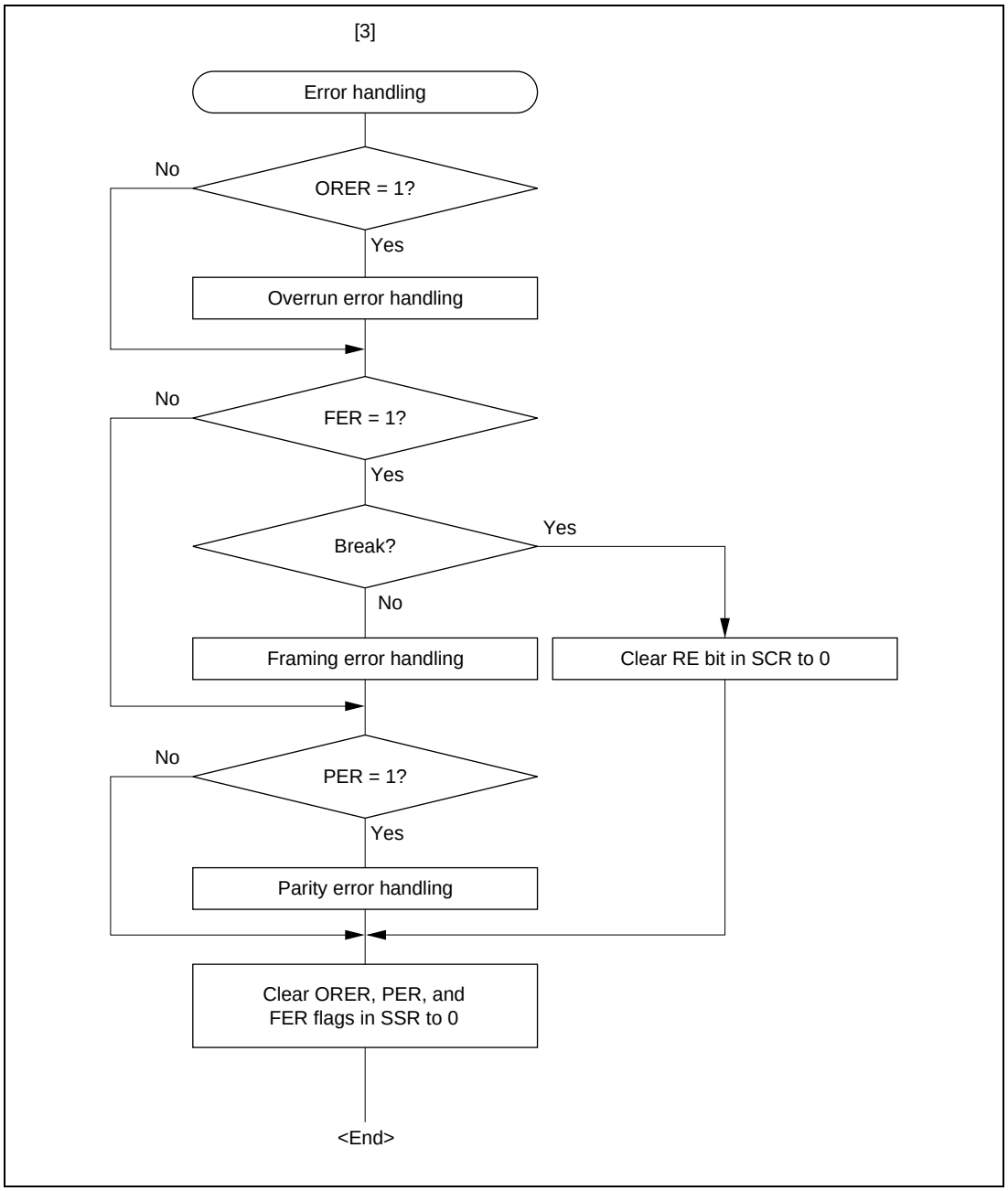


Figure 15.7 Sample Serial Reception Data Flowchart (cont)

In serial reception, the SCI operates as described below.

1. The SCI monitors the transmission line, and if a 0 stop bit is detected, performs internal synchronization and starts reception.
2. The received data is stored in RSR in LSB-to-MSB order.
3. The parity bit and stop bit are received.

   After receiving these bits, the SCI carries out the following checks.

   — Parity check:

   The SCI checks whether the number of 1 bits in the receive data agrees with the parity (even or odd) set in the O/$\overline{E}$ bit in SMR.

   — Stop bit check:

   The SCI checks whether the stop bit is 1.

   If there are two stop bits, only the first is checked.

   — Status check:

   The SCI checks whether the RDRF flag is 0, indicating that the receive data can be transferred from RSR to RDR.

   If all the above checks are passed, the RDRF flag is set to 1, and the receive data is stored in RDR.

   If a receive error* is detected in the error check, the operation is as shown in table 15.11.

Note: * Subsequent receive operations cannot be performed when a receive error has occurred. Also note that the RDRF flag is not set to 1 in reception, and so the error flags must be cleared to 0.

4. If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated.

   Also, if the RIE bit in SCR is set to 1 when the ORER, PER, or FER flag changes to 1, a receive-error interrupt (ERI) request is generated.

Table 15.11 Receive Errors and Conditions for Occurrence

Receive Error	Abbreviation	Occurrence Condition	Data Transfer
Overrun error	ORER	When the next data reception is completed while the RDRF flag in SSR is set to 1	Receive data is not transferred from RSR to RDR
Framing error	FER	When the stop bit is 0	Receive data is transferred from RSR to RDR
Parity error	PER	When the received data differs from the parity (even or odd) set in SMR	Receive data is transferred from RSR to RDR

Figure 15.8 shows an example of the operation for reception in asynchronous mode.

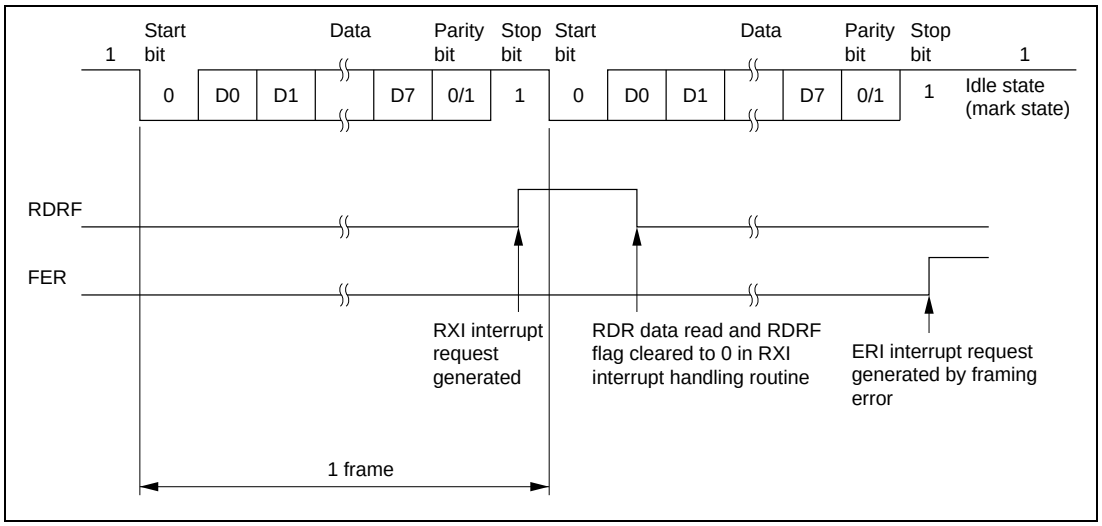


Figure 15.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

15.3.3 Multiprocessor Communication Function

The multiprocessor communication function performs serial communication using a multiprocessor format, in which a multiprocessor bit is added to the transfer data, in asynchronous mode. Use of this function enables data transfer to be performed among a number of processors sharing transmission lines.

When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code.

The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle.

The transmitting station first sends the ID of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added.

The receiving station skips the data until data with a 1 multiprocessor bit is sent.

When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip the data until data with a 1 multiprocessor bit is again received. In this way, data communication is carried out among a number of processors.

Figure 15.9 shows an example of inter-processor communication using a multiprocessor format.

Data Transfer Format: There are four data transfer formats.

When a multiprocessor format is specified, the parity bit specification is invalid.

For details, see table 15.10.

Clock: See the section on asynchronous mode.

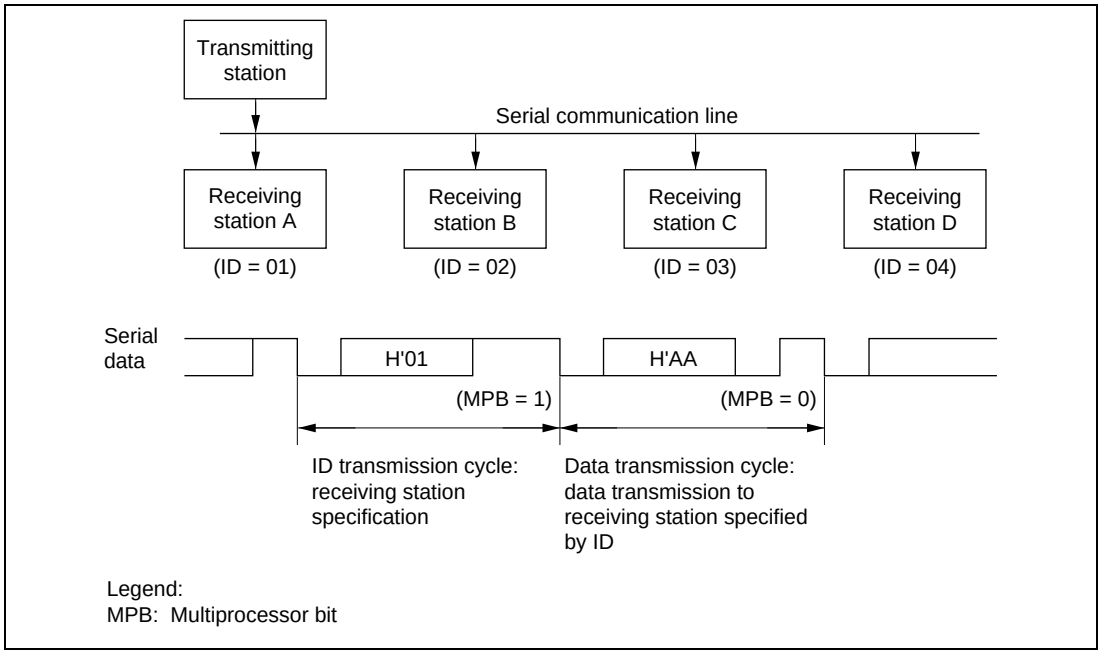
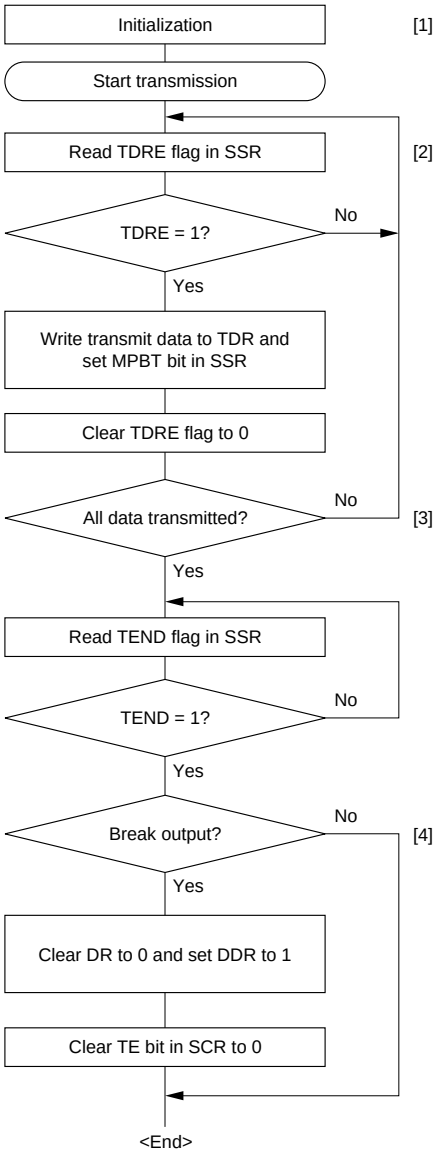


Figure 15.9 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

Data Transfer Operations

Multiprocessor Serial Data Transmission: Figure 15.10 shows a sample flowchart for multiprocessor serial data transmission.

The following procedure should be used for multiprocessor serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, one frame of 1s is output and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1.
Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit-data-empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the port DDR to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 15.10 Sample Multiprocessor Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission.

   If the TIE bit is set to 1 at this time, a transmit-data-empty interrupt (TXI) is generated.

   The serial transmit data is sent from the TxD pin in the following order.

   a. Start bit:

      One 0-bit is output.

   b. Transmit data:

      8-bit or 7-bit data is output in LSB-first order.

   c. Multiprocessor bit

      One multiprocessor bit (MPBT value) is output.

   d. Stop bit(s):

      One or two 1-bits (stop bits) are output.

   e. Mark state:

      1 is output continuously until the start bit that starts the next transmission is sent.
3. The SCI checks the TDRE flag at the timing for sending the stop bit.

   If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is set to 1 at this time, a transmit-end interrupt (TEI) request is generated.

Figure 15.11 shows an example of SCI operation for transmission using a multiprocessor format.

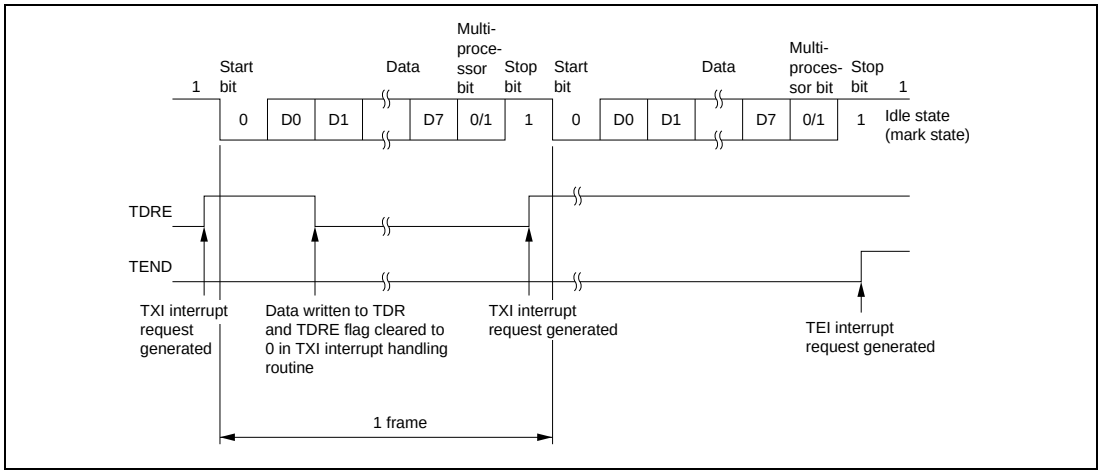


Figure 15.11 Example of SCI Operation in Transmission (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Multiprocessor Serial Data Reception: Figure 15.12 shows a sample flowchart for multiprocessor serial reception.

The following procedure should be used for multiprocessor serial data reception.

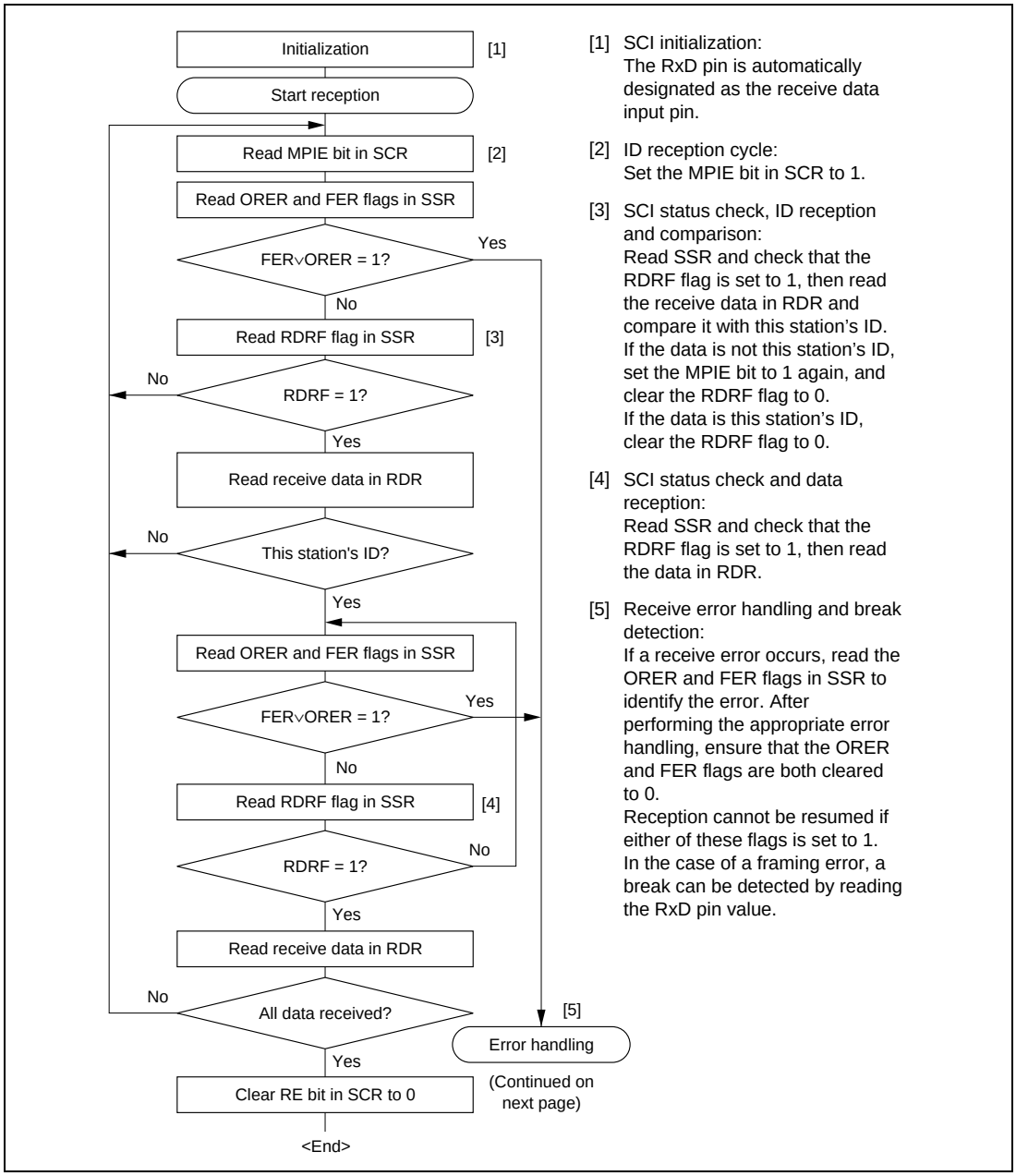


Figure 15.12 Sample Multiprocessor Serial Reception Flowchart

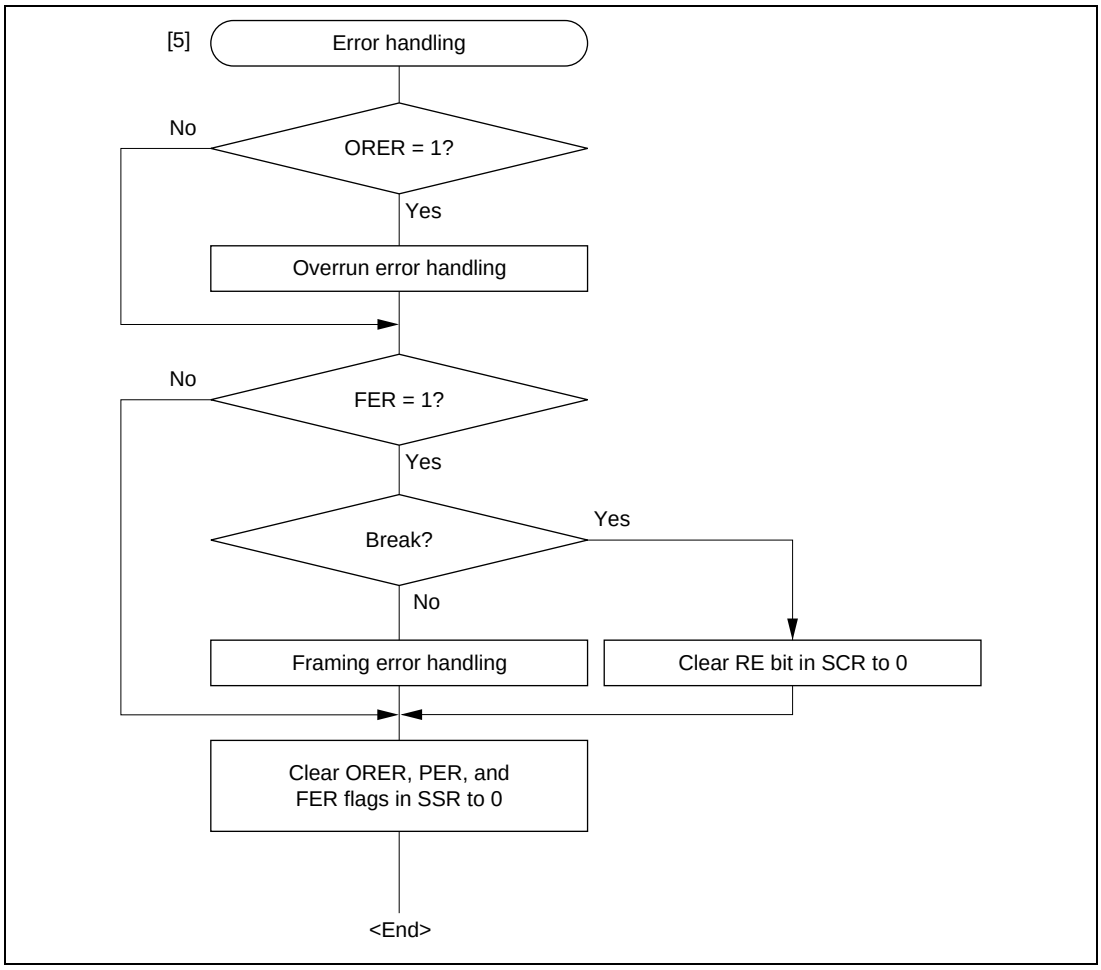


Figure 15.12 Sample Multiprocessor Serial Reception Flowchart (cont)

Figure 15.13 shows an example of SCI operation for multiprocessor format reception.

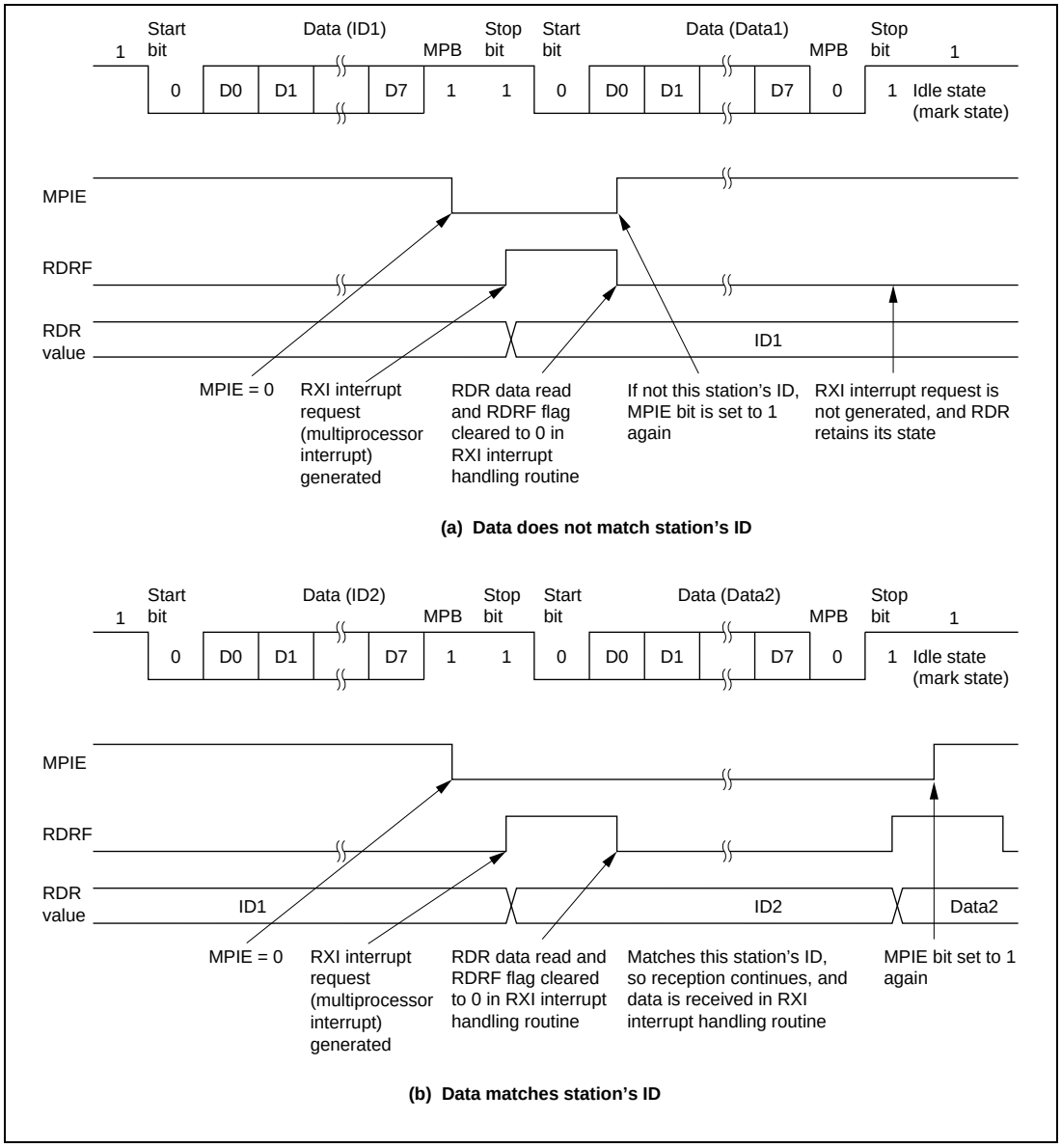


Figure 15.13 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

15.3.4 Operation in Synchronous Mode

In synchronous mode, data is transmitted or received in synchronization with clock pulses, making it suitable for high-speed serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 15.14 shows the general format for synchronous serial communication.

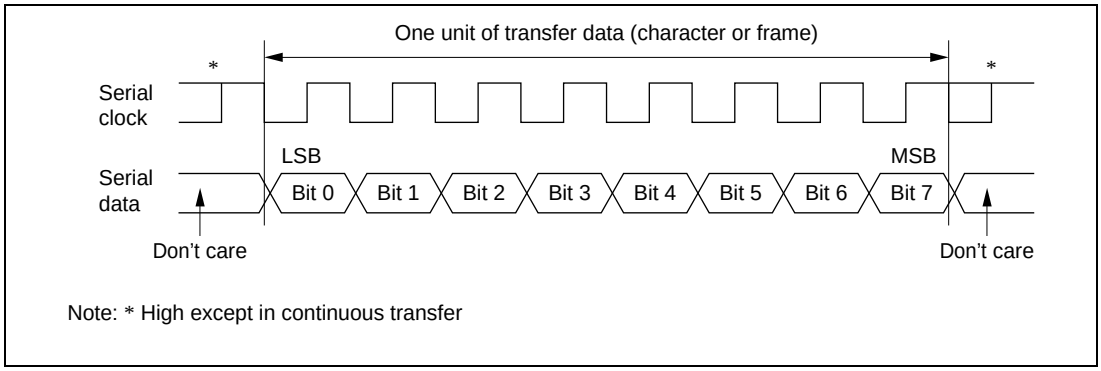


Figure 15.14 Data Format in Synchronous Communication

In synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock.

In synchronous serial communication, one character consists of data output starting with the LSB and ending with the MSB. After the MSB is output, the transmission line holds the MSB state.

In synchronous mode, the SCI receives data in synchronization with the rising edge of the serial clock.

Data Transfer Format: A fixed 8-bit data format is used.

No parity or multiprocessor bits are added.

Clock: Either an internal clock generated by the built-in baud rate generator or an external serial clock input at the SCK pin can be selected, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details on SCI clock source selection, see table 15.9.

When the SCI is operated on an internal clock, the serial clock is output from the SCK pin.

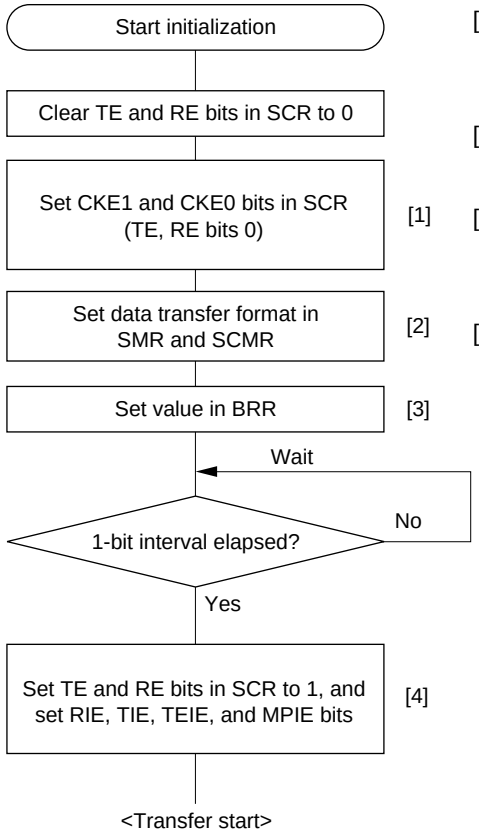
Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high. When only receive operations are performed, however, the serial clock is output until an overrun error occurs or the RE bit is cleared to 0. To perform receive operations in units of one character, select an external clock as the clock source.

Data Transfer Operations

SCI Initialization (Synchronous Mode): Before transmitting and receiving data, first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the settings of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

Figure 15.15 shows a sample SCI initialization flowchart.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, TE and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. This is not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: In simultaneous transmitting and receiving, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 15.15 Sample SCI Initialization Flowchart

Serial Data Transmission (Synchronous Mode): Figure 15.16 shows a sample flowchart for serial transmission.

The following procedure should be used for serial data transmission.

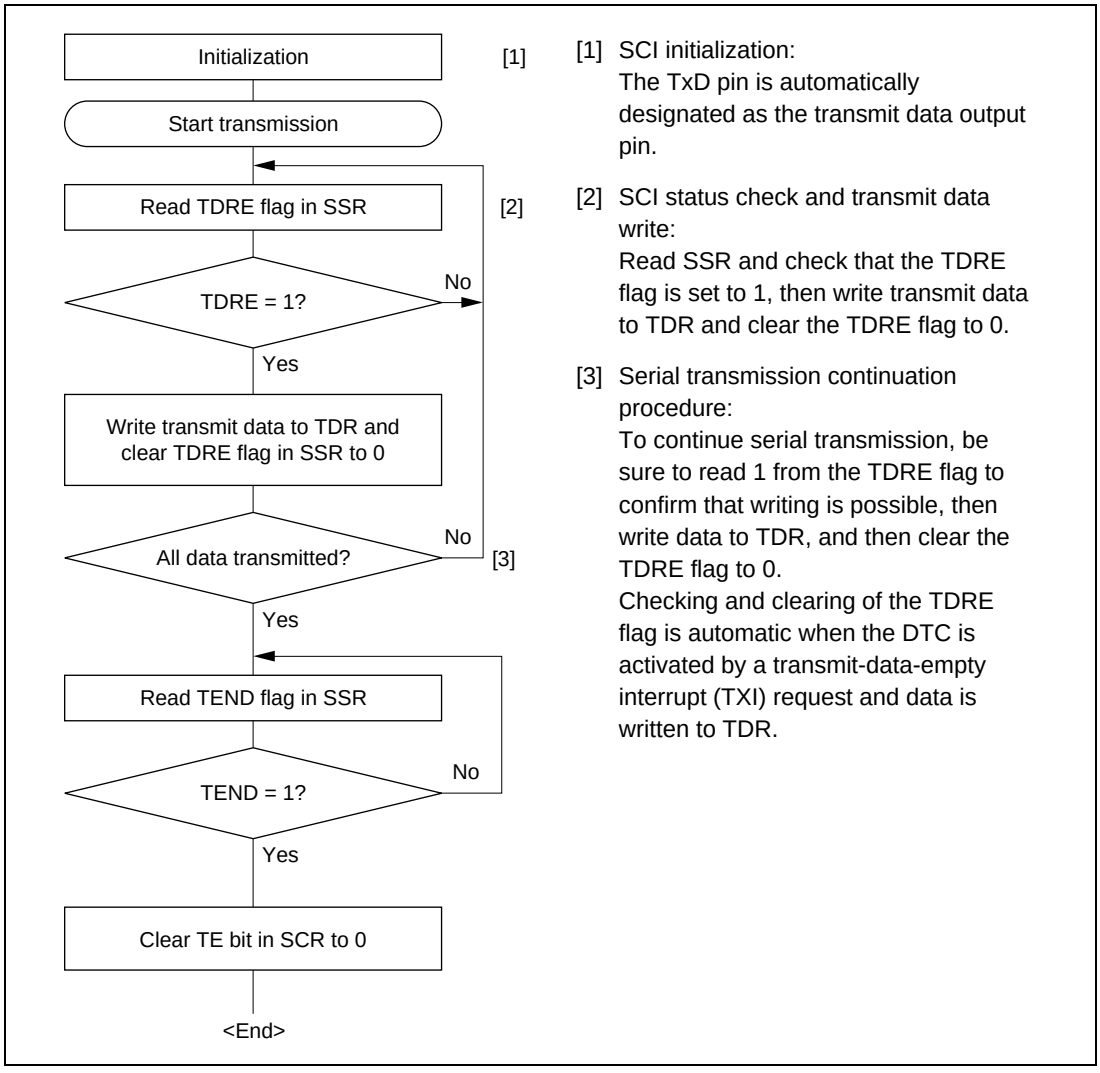


Figure 15.16 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit-data-empty interrupt (TXI) is generated.

   When clock output mode has been set, the SCI outputs 8 serial clock pulses. When use of an external clock has been specified, data is output synchronized with the input clock.

   The serial transmit data is sent from the TxD pin starting with the LSB (bit 0) and ending with the MSB (bit 7).
3. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).

   If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.

   If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the MSB (bit 7) is sent, and the TxD pin maintains its state.

   If the TEIE bit in SCR is set to 1 at this time, a transmit-end interrupt (TEI) request is generated.
4. After completion of serial transmission, the SCK pin is held in a constant state.

Figure 15.17 shows an example of SCI operation in transmission.

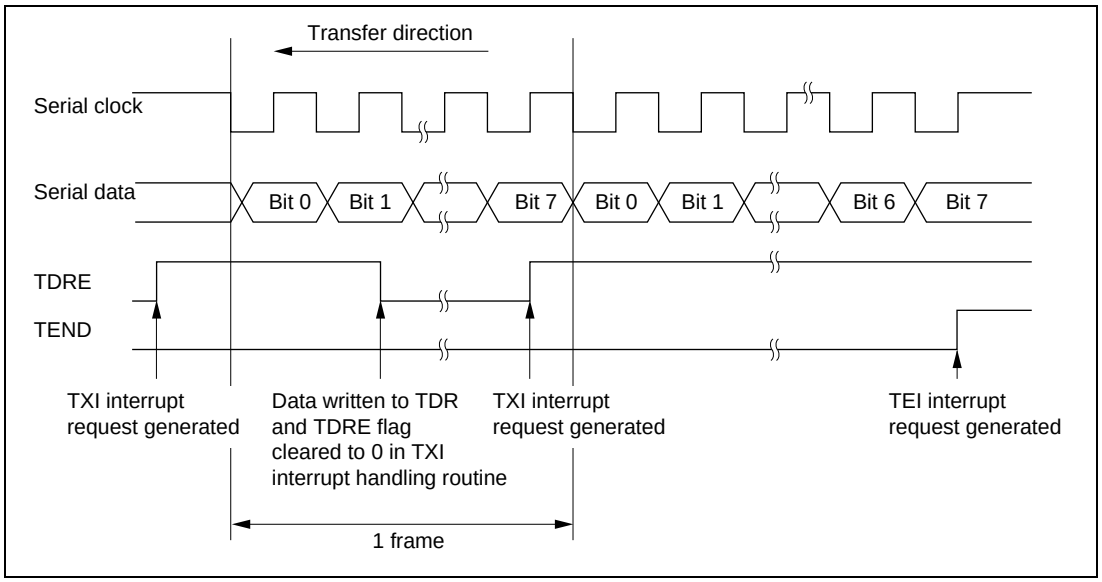


Figure 15.17 Example of SCI Operation in Transmission

Serial Data Reception (Synchronous Mode): Figure 15.18 shows a sample flowchart for serial reception.

The following procedure should be used for serial data reception.

When changing the operating mode from asynchronous to synchronous, be sure to check that the ORER, PER, and FER flags are all cleared to 0.

The RDRF flag will not be set if the FER or PER flag is set to 1, and neither transmit nor receive operations will be possible.

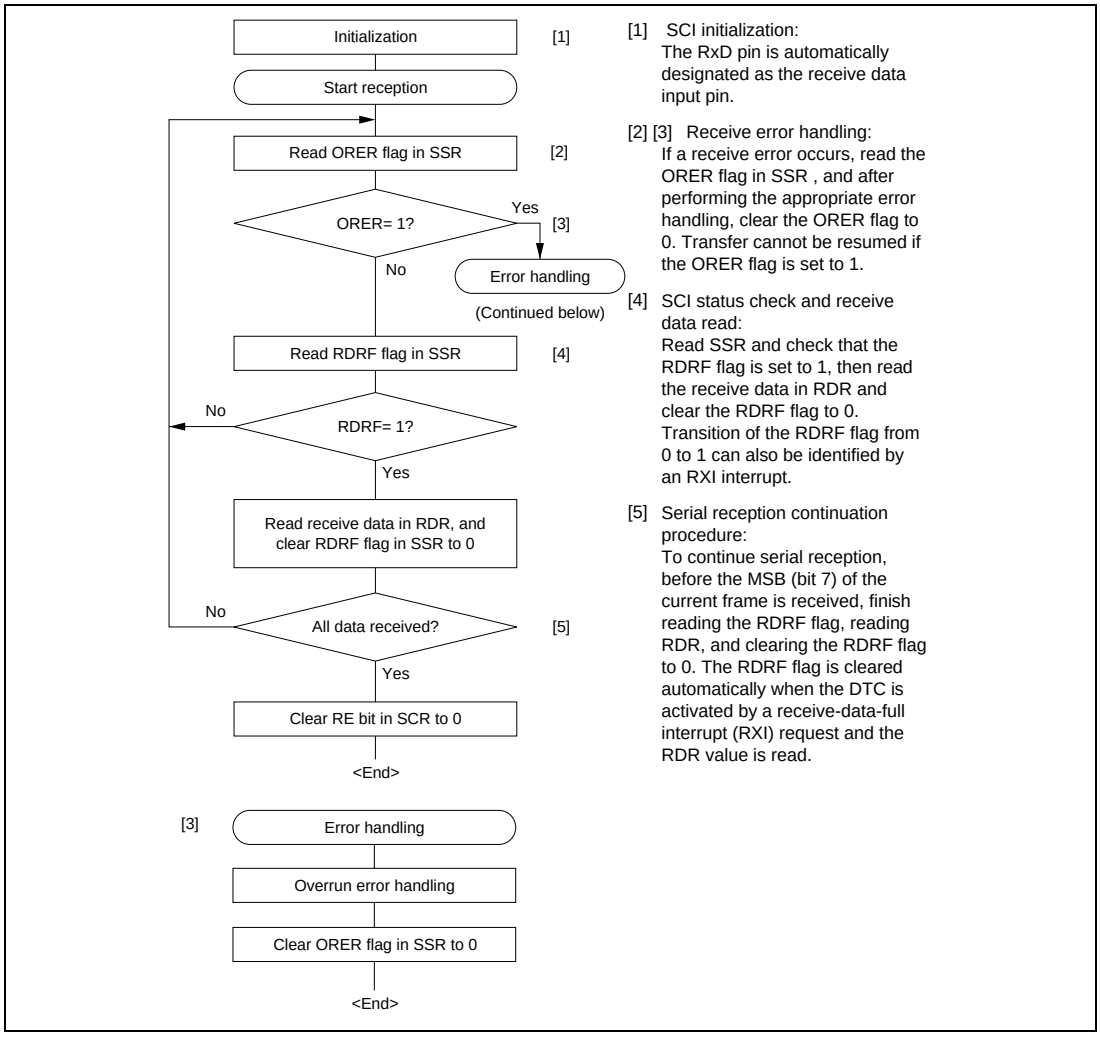


Figure 15.18 Sample Serial Reception Flowchart

In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization in synchronization with serial clock input or output.
2. The received data is stored in RSR in LSB-to-MSB order.

   After reception, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from RSR to RDR.

   If this check is passed, the RDRF flag is set to 1, and the receive data is stored in RDR. If a receive error is detected in the error check, the operation is as shown in table 15.11.

   Neither transmit nor receive operations can be performed subsequently when a receive error has been found in the error check.
3. If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated.

   Also, if the RIE bit in SCR is set to 1 when the ORER flag changes to 1, a receive-error interrupt (ERI) request is generated.

Figure 15.19 shows an example of SCI operation in reception.

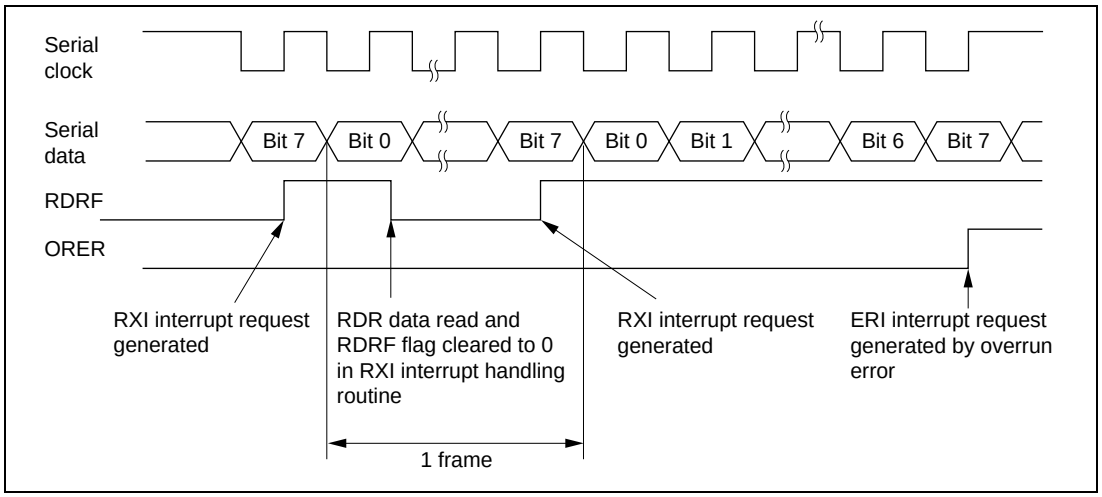
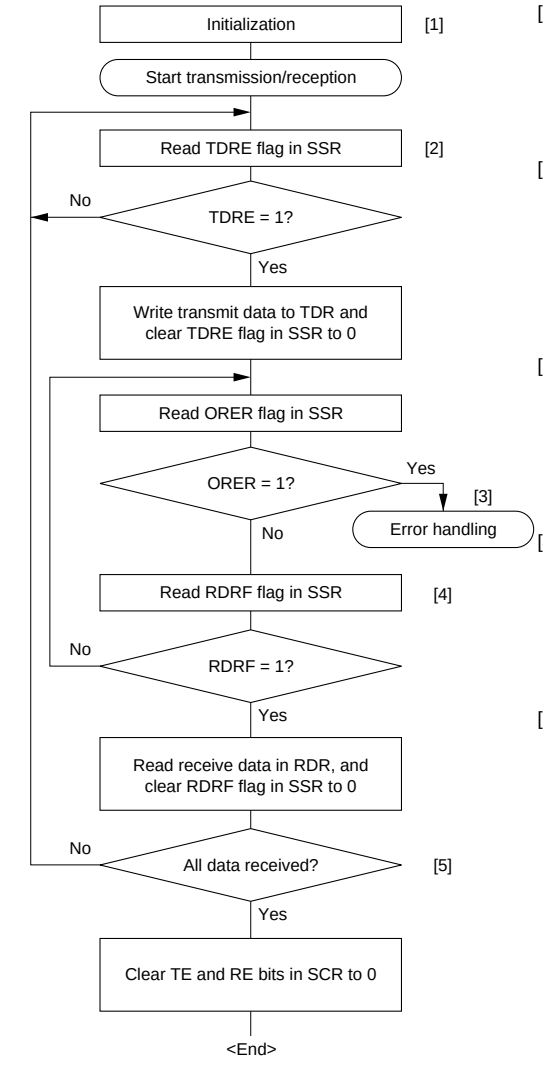


Figure 15.19 Example of SCI Operation in Reception

Simultaneous Serial Data Transmission and Reception (Synchronous Mode): Figure 15.20 shows a sample flowchart for simultaneous serial transmit and receive operations.

The following procedure should be used for simultaneous serial data transmit and receive operations.



Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error handling:
If a receive error occurs, read the ORER flag in SSR , and after performing the appropriate error handling, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/ reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit-data-empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DTC is activated by a receive-data-full interrupt (RXI) request and the RDR value is read.

Figure 15.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

15.4 SCI Interrupts

The SCI has four interrupt sources: the transmit-end interrupt (TEI) request, receive-error interrupt (ERI) request, receive-data-full interrupt (RXI) request, and transmit-data-empty interrupt (TXI) request. Table 15.12 shows the interrupt sources and their relative priorities. Individual interrupt sources can be enabled or disabled with the TIE, RIE, and TEIE bits in SCR. Each kind of interrupt request is sent to the interrupt controller independently.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer is 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 the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt can activate the DTC to perform data transfer. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC. The DTC cannot be activated by an ERI interrupt request.

Table 15.12 SCI Interrupt Sources

Channel	Interrupt Source	Description	DTC Activation	Priority*
0	ERI	Receive error (ORER, FER, or PER)	Not possible	High
	RXI	Receive data register full (RDRF)	Possible	
	TXI	Transmit data register empty (TDRE)	Possible	
	TEI	Transmit end (TEND)	Not possible	
1	ERI	Receive error (ORER, FER, or PER)	Not possible	
	RXI	Receive data register full (RDRF)	Possible	
	TXI	Transmit data register empty (TDRE)	Possible	
	TEI	Transmit end (TEND)	Not possible	Low

Note: * The table shows the initial state immediately after a reset. Relative channel priorities can be changed by the interrupt controller.

The TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1. The TEND flag is cleared at the same time as the TDRE flag. Consequently, if a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt will have priority for acceptance, and the TDRE flag and TEND flag may be cleared. Note that the TEI interrupt will not be accepted in this case.

15.5 Usage Notes

The following points should be noted when using the SCI.

Relation between Writes to TDR and the TDRE Flag: The TDRE flag in SSR is a status flag that indicates that transmit data has been transferred from TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to 1.

Data can be written to TDR regardless of the state of the TDRE flag. However, if new data is written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost since it has not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is set to 1 before writing transmit data to TDR.

Operation when Multiple Receive Errors Occur Simultaneously: If a number of receive errors occur at the same time, the state of the status flags in SSR is as shown in table 15.13. If there is an overrun error, data is not transferred from RSR to RDR, and the receive data is lost.

Table 15.13 State of SSR Status Flags and Transfer of Receive Data

	SSR Status Flags			Receive Data Transfer	
RDRF	**ORER**	**FER**	**PER**	**RSR to RDR**	**Receive Errors**
1	1	0	0	X	Overrun error
0	0	1	0	O	Framing error
0	0	0	1	O	Parity error
1	1	1	0	X	Overrun error + framing error
1	1	0	1	X	Overrun error + parity error
0	0	1	1	O	Framing error + parity error
1	1	1	1	X	Overrun error + framing error + parity error

Notes: O: Receive data is transferred from RSR to RDR.
X: Receive data is not transferred from RSR to RDR.

Break Detection and Processing: When a framing error (FER) is detected, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is set, and the parity error flag (PER) may also be set.

Note that, since the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

Sending a Break: The TxD pin has a dual function as an I/O port whose direction (input or output) is determined by DR and DDR. This feature can be used to send a break.

Between serial transmission initialization and setting of the TE bit to 1, the mark state is replaced by the value of DR (the pin does not function as the TxD pin until the TE bit is set to 1). Consequently, DDR and DR for the port corresponding to the TxD pin should first be set to 1.

To send a break during serial transmission, first clear DR to 0, then clear the TE bit to 0.

When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

Receive Error Flags and Transmit Operations (Synchronous Mode Only):
Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission.

Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

Receive Data Sampling Timing and Reception Margin in Asynchronous Mode:
In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times the transfer rate.

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the base clock. This is illustrated in figure 15.21.

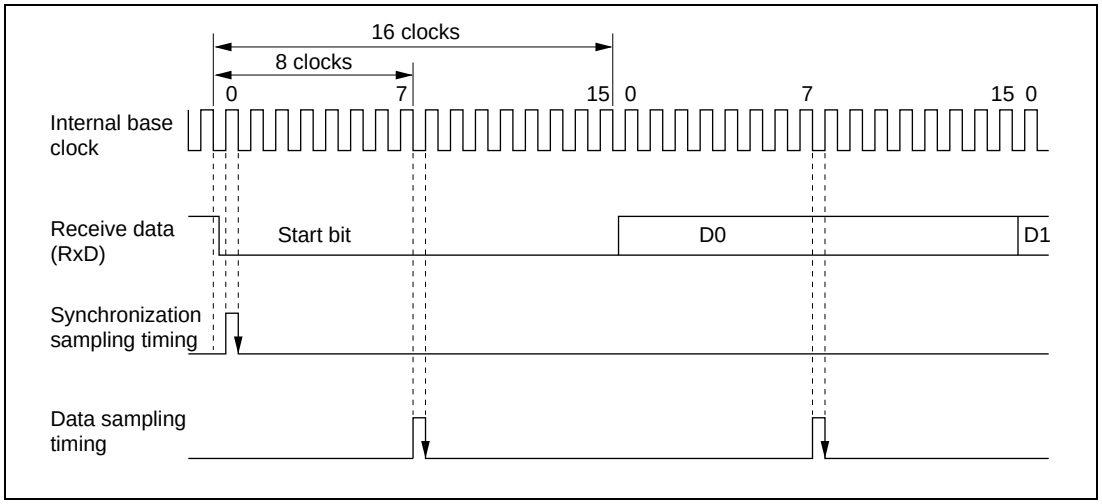


Figure 15.21 Receive Data Sampling Timing in Asynchronous Mode

Thus the receive margin in asynchronous mode is given by equation (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\% \quad \text{..... (1)}$$

Where M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in equation (1), a receive margin of 46.875% is given by equation (2) below.

When D = 0.5 and F = 0,

$$M = \left(0.5 - \frac{1}{2 \times 16}\right) \times 100\%$$
$$= 46.875\% \quad \text{..... (2)}$$

However, this is only a theoretical value, and a margin of 20% to 30% should be allowed in system design.

Restrictions on Use of DTC

- When an external clock source is used as the serial clock, the transmit clock should not be input until at least 5 ø clock cycles after TDR is updated by the DTC. Misoperation may occur if the transmit clock is input within 4 clock cycles after TDR is updated. (Figure 15.22)
- When RDR is read by the DTC, be sure to set the activation source to the relevant SCI receive-data-full interrupt (RXI).

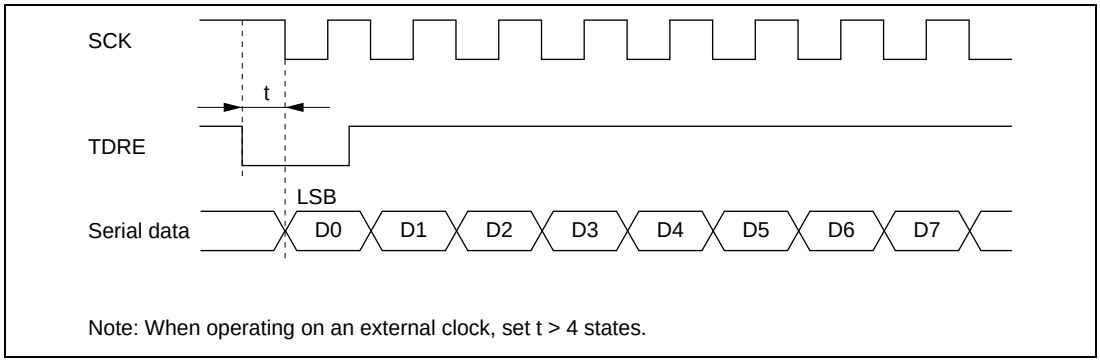


Figure 15.22 Example of Synchronous Transmission by DTC

Section 16 I^2C Bus Interface (IIC) [H8S/2128 Series Option]

A two-channel I^2C bus interface is available as an option in the H8S/2128 Series. The I^2C bus interface is not available for the H8S/2124 Series. Observe the following notes when using this option.

1. For mask-ROM versions, a W is added to the part number in products in which this optional function is used.

   Example: HD6432127RWF
2. The product number is identical for F-ZTAT versions. However, be sure to inform your Hitachi sales representative if you will be using this option.

16.1 Overview

A two-channel I^2C bus interface is available for the H8S/2128 Series as an option. The I^2C bus interface conforms to and provides a subset of the Philips I^2C bus (inter-IC bus) interface functions. The register configuration that controls the I^2C bus differs partly from the Philips configuration, however.

Each I^2C bus interface channel uses only one data line (SDA) and one clock line (SCL) to transfer data, saving board and connector space.

16.1.1 Features

- Selection of addressing format or non-addressing format
  - I^2C bus format: addressing format with acknowledge bit, for master/slave operation
  - Serial format: non-addressing format without acknowledge bit, for master operation only
- Conforms to Philips I^2C bus interface (I^2C bus format)
- Two ways of setting slave address (I^2C bus format)
- Start and stop conditions generated automatically in master mode (I^2C bus format)
- Selection of acknowledge output levels when receiving (I^2C bus format)
- Automatic loading of acknowledge bit when transmitting (I^2C bus format)
- Wait function in master mode (I^2C bus format)

  A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement. The wait can be cleared by clearing the interrupt flag.
- Wait function in slave mode (I^2C bus format)

  A wait request can be generated by driving the SCL pin low after data transfer, excluding acknowledgement. The wait request is cleared when the next transfer becomes possible.

- Three interrupt sources
  - Data transfer end (including transmission mode transition with I^2C bus format and address reception after loss of master arbitration)
  - Address match: when any slave address matches or the general call address is received in slave receive mode (I^2C bus format)
  - Stop condition detection
- Selection of 16 internal clocks (in master mode)
- Direct bus drive (with SCL and SDA pins)
  - Two pins—P52/SCL0 and P47/SDA0—(normally NMOS push-pull outputs) function as NMOS open-drain outputs when the bus drive function is selected.
  - Two pins—P24/SCL1 and P23/SDA1—(normally CMOS pins) function as NMOS-only outputs when the bus drive function is selected.
- Automatic switching from formatless mode to I^2C bus format (channel 0 only)
  - Formatless operation (no start/stop conditions, non-addressing mode) in slave mode
  - Operation using a common data pin (SDA) and independent clock pins (VSYNCI, SCL)
  - Automatic switching from formatless mode to I^2C bus format on the fall of the SCL pin

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the I²C bus interface.

Figure 16.2 shows an example of I/O pin connections to external circuits. Channel 0 I/O pins and channel 1 I/O pins differ in structure, and have different specifications for permissible applied voltages. For details, see section 22, Electrical Characteristics.

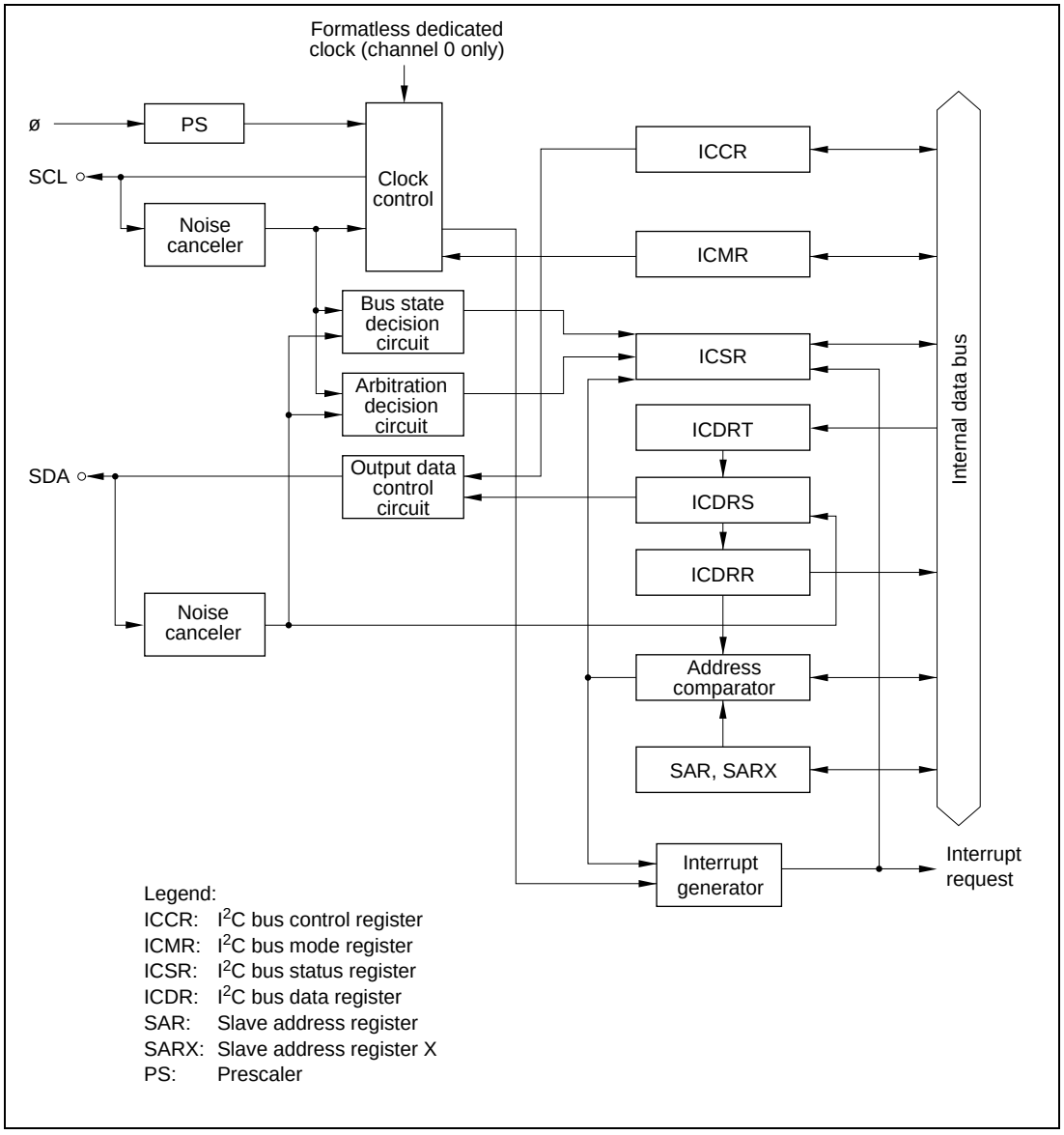


Figure 16.1 Block Diagram of I²C Bus Interface

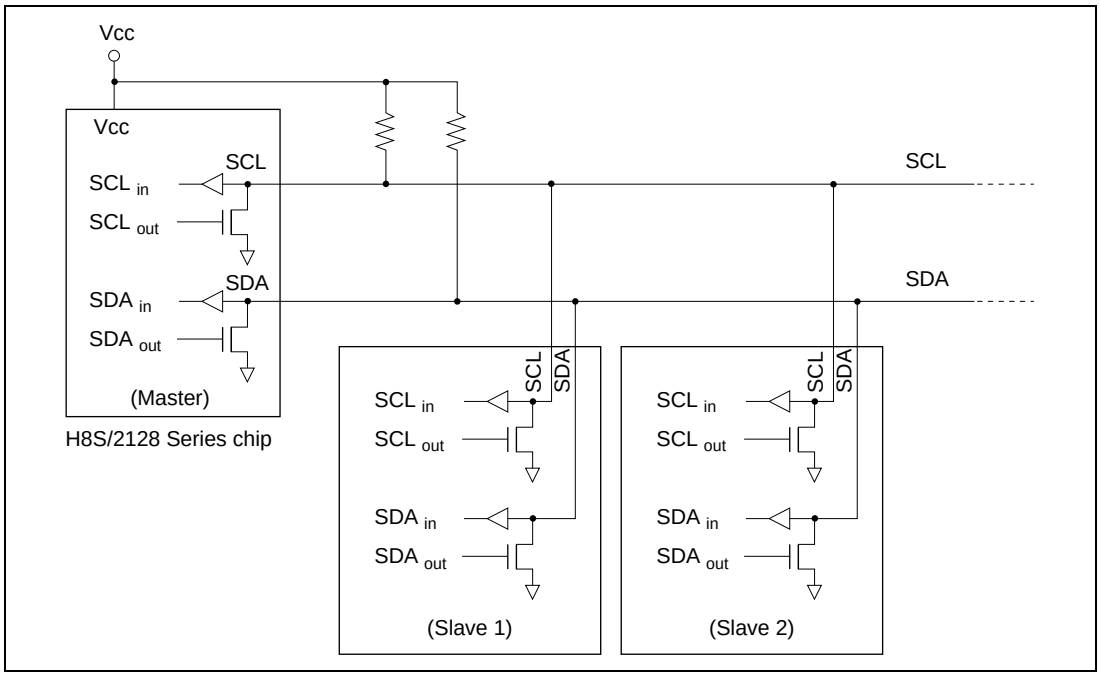


Figure 16.2 I²C Bus Interface Connections (Example: H8S/2128 Series Chip as Master)

16.1.3 Input/Output Pins

Table 16.1 summarizes the input/output pins used by the I²C bus interface.

Table 16.1 I²C Bus Interface Pins

Channel	Name	Abbreviation	I/O	Function
0	Serial clock	SCL0	I/O	IIC0 serial clock input/output
	Serial data	SDA0	I/O	IIC0 serial data input/output
	Formatless serial clock	VSYNCI	Input	IIC0 formatless serial clock input
1	Serial clock	SCL1	I/O	IIC1 serial clock input/output
	Serial data	SDA1	I/O	IIC1 serial data input/output

Note: In the text, the channel subscript is omitted, and only SCL and SDA are used.

16.1.4 Register Configuration

Table 16.2 summarizes the registers of the I²C bus interface.

Table 16.2 Register Configuration

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
0	I²C bus control register	ICCR0	R/W	H'01	H'FFD8
	I²C bus status register	ICSR0	R/W	H'00	H'FFD9
	I²C bus data register	ICDR0	R/W	—	H'FFDE*[2]
	I²C bus mode register	ICMR0	R/W	H'00	H'FFDF*[2]
	Slave address register	SAR0	R/W	H'00	H'FFDF*[2]
	Second slave address register	SARX0	R/W	H'01	H'FFDE*[2]
1	I²C bus control register	ICCR1	R/W	H'01	H'FF88
	I²C bus status register	ICSR1	R/W	H'00	H'FF89
	I²C bus data register	ICDR1	R/W	—	H'FF8E*[2]
	I²C bus mode register	ICMR1	R/W	H'00	H'FF8F*[2]
	Slave address register	SAR1	R/W	H'00	H'FF8F*[2]
	Second slave address register	SARX1	R/W	H'01	H'FF8E*[2]
Common	Serial/timer control register	STCR	R/W	H'00	H'FFC3
	DDC switch register	DDCSWR	R/W	H'0F	H'FEE6
	Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
		MSTPCRL	R/W	H'FF	H'FF87

Notes: 1. Lower 16 bits of the address.

2. The register that can be written or read depends on the ICE bit in the I²C bus control register. The slave address register can be accessed when ICE = 0, and the I²C bus mode register can be accessed when ICE = 1.

The I²C bus interface registers are assigned to the same addresses as other registers. Register selection is performed by means of the IICE bit in the serial/timer control register (STCR).

16.2 Register Descriptions

16.2.1 I²C Bus Data Register (ICDR)

Bit	7	6	5	4	3	2	1	0
	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ICDRR

Bit	7	6	5	4	3	2	1	0
	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R	R	R	R	R	R	R	R

- ICDRS

Bit	7	6	5	4	3	2	1	0
	ICDRS7	ICDRS6	ICDRR5	ICDRS4	ICDRS3	ICDRS2	ICDRS1	ICDRS0
Initial value	—	—	—	—	—	—	—	—
Read/Write	—	—	—	—	—	—	—	—

- ICDRT

Bit	7	6	5	4	3	2	1	0
	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0
Initial value	—	—	—	—	—	—	—	—
Read/Write	W	W	W	W	W	W	W	W

- TDRE, RDRF (internal flags)

Bit							—	—
							TDRE	RDRF
Initial value							0	0
Read/Write							—	—

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is divided internally into a shift register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). ICDRS cannot be read or written by the CPU, ICDRR is read-only, and ICDRT is write-only. Data transfers among the three registers are performed automatically in coordination with changes in the bus state, and affect the status of internal flags such as TDRE and RDRF.

If IIC is in transmit mode and the next data is in ICDRT (the TDRE flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRT to ICDRS. If IIC is in receive mode and no previous data remains in ICDRR (the RDRF flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRS to ICDRR.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

ICDR is assigned to the same address as SARX, and can be written and read only when the ICE bit is set to 1 in ICCR.

The value of ICDR is undefined after a reset.

The TDRE and RDRF flags are set and cleared under the conditions shown below. Setting the TDRE and RDRF flags affects the status of the interrupt flags.

TDRE	Description
0	The next transmit data is in ICDR (ICDRT), or transmission cannot be started (Initial value) [Clearing conditions] • When transmit data is written in ICDR (ICDRT) in transmit mode (TRS = 1) • When a stop condition is detected in the bus line state after a stop condition is issued with the I^2C bus format or serial format selected • When a stop condition is detected with the I^2C bus format selected • In receive mode (TRS = 0) (A 0 write to TRS during transfer is valid after reception of a frame containing an acknowledge bit)
1	The next transmit data can be written in ICDR (ICDRT) [Setting conditions] • In transmit mode (TRS = 1), when a start condition is detected in the bus line state after a start condition is issued in master mode with the I^2C bus format or serial format selected • When using formatless mode in transmit mode (TRS = 1) • When data is transferred from ICDRT to ICDRS (Data transfer from ICDRT to ICDRS when TRS = 1 and TDRE = 0, and ICDRS is empty) • When a switch is made from receive mode (TRS = 0) to transmit mode (TRS = 1) after detection of a start condition

RDRF	Description
0	The data in ICDR (ICDRR) is invalid (Initial value) [Clearing condition] When ICDR (ICDRR) receive data is read in receive mode
1	The ICDR (ICDRR) receive data can be read [Setting condition] When data is transferred from ICDRS to ICDRR (Data transfer from ICDRS to ICDRR in case of normal termination with TRS = 0 and RDRF = 0)

16.2.2 Slave Address Register (SAR)

Bit	7	6	5	4	3	2	1	0
	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SAR is an 8-bit readable/writable register that stores the slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SAR is assigned to the same address as ICMR, and can be written and read only when the ICE bit is cleared to 0 in ICCR.

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

Bits 7 to 1—Slave Address (SVA6 to SVA0): Set a unique address in bits SVA6 to SVA0, differing from the addresses of other slave devices connected to the I^2C bus.

Bit 0—Format Select (FS): Used together with the FSX bit in SARX and the SW bit in DDCSWR to select the communication format.

- I^2C bus format: addressing format with acknowledge bit
- Synchronous serial format: non-addressing format without acknowledge bit, for master mode only
- Formatless mode (channel 0 only): non-addressing format with or without acknowledge bit, slave mode only, start/stop conditions not detected

The FS bit also specifies whether or not SAR slave address recognition is performed in slave mode.

DDCSWR Bit 6	SAR Bit 0	SARX Bit 0	
SW	**FS**	**FSX**	**Operating Mode**
0	0	0	I^2C bus format • SAR and SARX slave addresses recognized
		1	I^2C bus format (Initial value) • SAR slave address recognized • SARX slave address ignored
	1	0	I^2C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	0	0	Formatless mode (start/stop conditions not detected)
		1	• Acknowledge bit used
	1	0	
		1	Formatless mode* (start/stop conditions not detected) • No acknowledge bit

Note: * Do not set this mode when automatic switching to the I^2C bus format is performed by means of the DDCSWR setting.

16.2.3 Second Slave Address Register (SARX)

Bit	7	6	5	4	3	2	1	0
	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SARX is an 8-bit readable/writable register that stores the second slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SARX match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SARX is assigned to the same address as ICDR, and can be written and read only when the ICE bit is cleared to 0 in ICCR.

SARX is initialized to H'01 by a reset and in hardware standby mode.

Bits 7 to 1—Second Slave Address (SVAX6 to SVAX0): Set a unique address in bits SVAX6 to SVAX0, differing from the addresses of other slave devices connected to the I^2C bus.

Bit 0—Format Select X (FSX): Used together with the FS bit in SAR and the SW bit in DDCSWR to select the communication format.

- I^2C bus format: addressing format with acknowledge bit
- Synchronous serial format: non-addressing format without acknowledge bit, for master mode only
- Formatless mode: non-addressing format with or without acknowledge bit, slave mode only, start/stop conditions not detected

The FSX bit also specifies whether or not SARX slave address recognition is performed in slave mode. For details, see the description of the FS bit in SAR.

16.2.4 I²C Bus Mode Register (ICMR)

Bit	7	6	5	4	3	2	1	0
	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the master mode transfer clock frequency and the transfer bit count. ICMR is assigned to the same address as SAR. ICMR can be written and read only when the ICE bit is set to 1 in ICCR.

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

Bit 7—MSB-First/LSB-First Select (MLS): Selects whether data is transferred MSB-first or LSB-first.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

Do not set this bit to 1 when the I²C bus format is used.

Bit 7

MLS	Description	
0	MSB-first	(Initial value)
1	LSB-first	

Bit 6—Wait Insertion Bit (WAIT): Selects whether to insert a wait between the transfer of data and the acknowledge bit, in master mode with the I^2C bus format. When WAIT is set to 1, after the fall of the clock for the final data bit, the IRIC flag is set to 1 in ICCR, and a wait state begins (with SCL at the low level). When the IRIC flag is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted.

The IRIC flag in ICCR is set to 1 on completion of the acknowledge bit transfer, regardless of the WAIT setting.

The setting of this bit is invalid in slave mode.

Bit 6

WAIT	Description	
0	Data and acknowledge bits transferred consecutively	(Initial value)
1	Wait inserted between data and acknowledge bits	

Bits 5 to 3—Serial Clock Select (CKS2 to CKS0): These bits, together with the IICX1 (channel 1) or IICX0 (channel 0) bit in the STCR register, select the serial clock frequency in master mode. They should be set according to the required transfer rate.

STCR Bit 5 or 6	Bit 5	Bit 4	Bit 3		Transfer Rate				
IICX	CKS2	CKS1	CKS0	Clock	ø = 5 MHz	ø = 8 MHz	ø = 10 MHz	ø = 16 MHz	ø = 20 MHz
0	0	0	0	ø/28	179 kHz	286 kHz	357 kHz	571 kHz*	714 kHz*
			1	ø/40	125 kHz	200 kHz	250 kHz	400 kHz	500 kHz*
		1	0	ø/48	104 kHz	167 kHz	208 kHz	333 kHz	417 kHz*
			1	ø/64	78.1 kHz	125 kHz	156 kHz	250 kHz	313 kHz
	1	0	0	ø/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
			1	ø/100	50.0 kHz	80.0 kHz	100 kHz	160 kHz	200 kHz
		1	0	ø/112	44.6 kHz	71.4 kHz	89.3 kHz	143 kHz	179 kHz
			1	ø/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	0	0	0	ø/56	89.3 kHz	143 kHz	179 kHz	286 kHz	357 kHz
			1	ø/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
		1	0	ø/96	52.1 kHz	83.3 kHz	104 kHz	167 kHz	208 kHz
			1	ø/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
	1	0	0	ø/160	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz	125 kHz
			1	ø/200	25.0 kHz	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz
		1	0	ø/224	22.3 kHz	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz
			1	ø/256	19.5 kHz	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz

Note: * Outside the I^2C bus interface specification range (normal mode: max. 100 kHz; high-speed mode: max. 400 kHz).

Bits 2 to 0—Bit Counter (BC2 to BC0): Bits BC2 to BC0 specify the number of bits to be transferred next. With the I^2C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the data is transferred with one addition acknowledge bit. Bits BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL line is low.

The bit counter is initialized to 000 by a reset and when a start condition is detected. The value returns to 000 at the end of a data transfer, including the acknowledge bit.

Bit 2	Bit 1	Bit 0	Bits/Frame		
BC2	**BC1**	**BC0**	**Synchronous Serial Format**	**I^2C Bus Format**	
0	0	0	8	9	(Initial value)
		1	1	2	
	1	0	2	3	
		1	3	4	
1	0	0	4	5	
		1	5	6	
	1	0	6	7	
		1	7	8	

16.2.5 I²C Bus Control Register (ICCR)

Bit	7	6	5	4	3	2	1	0
	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	W

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

ICCR is an 8-bit readable/writable register that enables or disables the I²C bus interface, enables or disables interrupts, selects master or slave mode and transmission or reception, enables or disables acknowledgement, confirms the I²C bus interface bus status, issues start/stop conditions, and performs interrupt flag confirmation.

ICCR is initialized to H'01 by a reset and in hardware standby mode.

Bit 7—I²C Bus Interface Enable (ICE): Selects whether or not the I²C bus interface is to be used. When ICE is set to 1, port pins function as SCL and SDA input/output pins and transfer operations are enabled. When ICE is cleared to 0, the I²C bus interface module is disabled.

The SAR and SARX registers can be accessed when ICE is 0. The ICMR and ICDR registers can be accessed when ICE is 1.

Bit 7

ICE	Description	
0	I²C bus interface module disabled, with SCL and SDA signal pins set to port function	(Initial value)
	SAR and SARX can be accessed	
1	I²C bus interface module enabled for transfer operations (pins SCL and SCA are driving the bus)	
	ICMR and ICDR can be accessed	

Bit 6—I²C Bus Interface Interrupt Enable (IEIC): Enables or disables interrupts from the I²C bus interface to the CPU.

Bit 6

IEIC	Description	
0	Interrupts disabled	(Initial value)
1	Interrupts enabled	

Bit 5—Master/Slave Select (MST)

Bit 4—Transmit/Receive Select (TRS)

MST selects whether the I^2C bus interface operates in master mode or slave mode.

TRS selects whether the I^2C bus interface operates in transmit mode or receive mode.

In master mode with the I^2C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. In slave receive mode with the addressing format (FS = 0 or FSX = 0), hardware automatically selects transmit or receive mode according to the R/W bit in the first frame after a start condition.

Modification of the TRS bit during transfer is deferred until transfer of the frame containing the acknowledge bit is completed, and the changeover is made after completion of the transfer.

MST and TRS select the operating mode as follows.

Bit 5	Bit 4		
MST	**TRS**	**Operating Mode**	
0	0	Slave receive mode	(Initial value)
	1	Slave transmit mode	
1	0	Master receive mode	
	1	Master transmit mode	

Bit 5	
MST	**Description**
0	Slave mode (Initial value) [Clearing conditions] 1. When 0 is written by software 2. When bus arbitration is lost after transmission is started in I^2C bus format master mode
1	Master mode [Setting conditions] 1. When 1 is written by software (in cases other than clearing condition 2) 2. When 1 is written in MST after reading MST = 0 (in case of clearing condition 2)

Bit 4

TRS	Description	
0	Receive mode	(Initial value)
	[Clearing conditions]	
	1. When 0 is written by software (in cases other than setting condition 3)	
	2. When 0 is written in TRS after reading TRS = 1 (in case of clearing condition 3)	
	3. When bus arbitration is lost after transmission is started in I^2C bus format master mode	
	4. When the SW bit in DDCSWR changes from 1 to 0	
1	Transmit mode	
	[Setting conditions]	
	1. When 1 is written by software (in cases other than clearing conditions 3 and 4)	
	2. When 1 is written in TRS after reading TRS = 0 (in case of clearing conditions 3 and 4)	
	3. When a 1 is received as the R/W bit of the first frame in I^2C bus format slave mode	

Bit 3—Acknowledge Bit Judgement Selection (ACKE): Specifies whether the value of the acknowledge bit returned from the receiving device when using the I^2C bus format is to be ignored and continuous transfer is performed, or transfer is to be aborted and error handling, etc., performed if the acknowledge bit is 1. When the ACKE bit is 0, the value of the received acknowledge bit is not indicated by the ACKB bit, which is always 0.

In the H8S/2128 Series, the DTC can be used to perform continuous transfer. The DTC is activated when the IRTR interrupt flag is set to 1 (IRTR is one of two interrupt flags, the other being IRIC). When the ACKE bit is 0, the TDRE, IRIC, and IRTR flags are set on completion of data transmission, regardless of the value of the acknowledge bit. When the ACKE bit is 1, the TDRE, IRIC, and IRTR flags are set on completion of data transmission when the acknowledge bit is 0, and the IRIC flag alone is set on completion of data transmission when the acknowledge bit is 1.

When the DTC is activated, the TDRE, IRIC, and IRTR flags are cleared to 0 after the specified number of data transfers have been executed. Consequently, interrupts are not generated during continuous data transfer, but if data transmission is completed with a 1 acknowledge bit when the ACKE bit is set to 1, the DTC is not activated and an interrupt is generated, if enabled.

Depending on the receiving device, the acknowledge bit may be significant, in indicating completion of processing of the received data, for instance, or may be fixed at 1 and have no significance.

Bit 3

ACKE	Description	
0	The value of the acknowledge bit is ignored, and continuous transfer is performed	(Initial value)
1	If the acknowledge bit is 1, continuous transfer is interrupted	

Bit 2—Bus Busy (BBSY): The BBSY flag can be read to check whether the I^2C bus (SCL, SDA) is busy or free. In master mode, this bit is also used to issue start and stop conditions.

A high-to-low transition of SDA while SCL is high is recognized as a start condition, setting BBSY to 1. A low-to-high transition of SDA while SCL is high is recognized as a stop condition, clearing BBSY to 0.

To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, use a MOV instruction to write 0 in BBSY and 0 in SCP. It is not possible to write to BBSY in slave mode; the I^2C bus interface must be set to master transmit mode before issuing a start condition. MST and TRS should both be set to 1 before writing 1 in BBSY and 0 in SCP.

Bit 2

BBSY	Description	
0	Bus is free	(Initial value)
	[Clearing condition]	
	When a stop condition is detected	
1	Bus is busy	
	[Setting condition]	
	When a start condition is detected	

Bit 1—I^2C Bus Interface Interrupt Request Flag (IRIC): Indicates that the I^2C bus interface has issued an interrupt request to the CPU. IRIC is set to 1 at the end of a data transfer, when a slave address or general call address is detected in slave receive mode, when bus arbitration is lost in master transmit mode, and when a stop condition is detected. IRIC is set at different times depending on the FS bit in SAR and the WAIT bit in ICMR. See section 16.3.6, IRIC Setting Timing and SCL Control. The conditions under which IRIC is set also differ depending on the setting of the ACKE bit in ICCR.

IRIC is cleared by reading IRIC after it has been set to 1, then writing 0 in IRIC.

When the DTC is used, IRIC is cleared automatically and transfer can be performed continuously without CPU intervention.

Bit 1

IRIC	Description
0	Waiting for transfer, or transfer in progress (Initial value) [Clearing conditions] 1. When 0 is written in IRIC after reading IRIC = 1 2. When ICDR is written or read by the DTC (When the TDRE or RDRF flag is cleared to 0) (This is not always a clearing condition; see the description of DTC operation for details)
1	Interrupt requested [Setting conditions] • I^2C bus format master mode 1. When a start condition is detected in the bus line state after a start condition is issued (when the TDRE flag is set to 1 because of first frame transmission) 2. When a wait is inserted between the data and acknowledge bit when WAIT = 1 3. At the end of data transfer (at the rise of the 9th transmit/receive clock pulse, and, when a wait is inserted, at the fall of the 8th transmit/receive clock pulse) 4. When a slave address is received after bus arbitration is lost (when the AL flag is set to 1) 5. When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) • I^2C bus format slave mode 1. When the slave address (SVA, SVAX) matches (when the AAS and AASX flags are set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) 2. When the general call address is detected (when FS = 0 and the ADZ flag is set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) 3. When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) 4. When a stop condition is detected (when the STOP or ESTP flag is set to 1) • Synchronous serial format, and formatless mode 1. At the end of data transfer (when the TDRE or RDRF flag is set to 1) 2. When a start condition is detected with serial format selected 3. When the SW bit is set to 1 in DDCSWR

When, with the I^2C bus format selected, IRIC is set to 1 and an interrupt is generated, other flags must be checked in order to identify the source that set IRIC to 1. Although each source has a corresponding flag, caution is needed at the end of a transfer.

When the TDRE or RDRF internal flag is set, the readable IRTR flag may or may not be set. The IRTR flag (the DTC start request flag) is not set at the end of a data transfer up to detection of a retransmission start condition or stop condition after a slave address (SVA) or general call address match in I^2C bus format slave mode.

Even when the IRIC flag and IRTR flag are set, the TDRE or RDRF internal flag may not be set. The IRIC and IRTR flags are not cleared at the end of the specified number of transfers in continuous transfer using the DTC. The TDRE or RDRF flag is cleared, however, since the specified number of ICDR reads or writes have been completed.

Table 16.3 shows the relationship between the flags and the transfer states.

Table 16.3 Flags and Transfer States

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	State
1/0	1/0	0	0	0	0	0	0	0	0	0	Idle state (flag clearing required)
1	1	0	0	0	0	0	0	0	0	0	Start condition issuance
1	1	1	0	0	1	0	0	0	0	0	Start condition established
1	1/0	1	0	0	0	0	0	0	0	0/1	Master mode wait
1	1/0	1	0	0	1	0	0	0	0	0/1	Master mode transmit/receive end
0	0	1	0	0	0	1/0	1	1/0	1/0	0	Arbitration lost
0	0	1	0	0	0	0	0	1	0	0	SAR match by first frame in slave mode
0	0	1	0	0	0	0	0	1	1	0	General call address match
0	0	1	0	0	0	1	0	0	0	0	SARX match
0	1/0	1	0	0	0	0	0	0	0	0/1	Slave mode transmit/receive end (except after SARX match)
0	1/0	1	0	0	1	1	0	0	0	0	Slave mode transmit/receive end (after SARX match)
0	1	1	0	0	0	1	0	0	0	1	
0	1/0	0	1/0	1/0	0	0	0	0	0	0/1	Stop condition detected

Bit 0—Start Condition/Stop Condition Prohibit (SCP): Controls the issuing of start and stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored.

Bit 0

SCP	Description	
0	Writing 0 issues a start or stop condition, in combination with the BBSY flag	
1	Reading always returns a value of 1	(Initial value)
	Writing is ignored	

16.2.6 I²C Bus Status Register (ICSR)

Bit	7	6	5	4	3	2	1	0
	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

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

ICSR is an 8-bit readable/writable register that performs flag confirmation and acknowledge confirmation and control.

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

Bit 7—Error Stop Condition Detection Flag (ESTP): Indicates that a stop condition has been detected during frame transfer in I²C bus format slave mode.

Bit 7

ESTP	Description	
0	No error stop condition	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in ESTP after reading ESTP = 1	
	2. When the IRIC flag is cleared to 0	
1	• In I²C bus format slave mode Error stop condition detected [Setting condition] When a stop condition is detected during frame transfer	
	• In other modes No meaning	

Bit 6—Normal Stop Condition Detection Flag (STOP): Indicates that a stop condition has been detected after completion of frame transfer in I^2C bus format slave mode.

Bit 6

STOP	Description	
0	No normal stop condition	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in STOP after reading STOP = 1	
	2. When the IRIC flag is cleared to 0	
1	• In I^2C bus format slave mode Normal stop condition detected [Setting condition] When a stop condition is detected after completion of frame transfer	
	• In other modes No meaning	

Bit 5—I^2C Bus Interface Continuous Transmission/Reception Interrupt Request Flag (IRTR): Indicates that the I^2C bus interface has issued an interrupt request to the CPU, and the source is completion of reception/transmission of one frame in continuous transmission/reception for which DTC activation is possible. When the IRTR flag is set to 1, the IRIC flag is also set to 1 at the same time.

IRTR flag setting is performed when the TDRE or RDRF flag is set to 1. IRTR is cleared by reading IRTR after it has been set to 1, then writing 0 in IRTR. IRTR is also cleared automatically when the IRIC flag is cleared to 0.

Bit 5

IRTR	Description	
0	Waiting for transfer, or transfer in progress	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in IRTR after reading IRTR = 1	
	2. When the IRIC flag is cleared to 0	
1	Continuous transfer state	
	[Setting condition]	
	• In I^2C bus interface slave mode When the TDRE or RDRF flag is set to 1 when AASX = 1	
	• In other modes When the TDRE or RDRF flag is set to 1	

Bit 4—Second Slave Address Recognition Flag (AASX): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVAX6 to SVAX0 in SARX.

AASX is cleared by reading AASX after it has been set to 1, then writing 0 in AASX. AASX is also cleared automatically when a start condition is detected.

Bit 4

AASX	Description	
0	Second slave address not recognized	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in AASX after reading AASX = 1	
	2. When a start condition is detected	
	3. In master mode	
1	Second slave address recognized	
	[Setting condition]	
	When the second slave address is detected in slave receive mode while FSX = 0	

Bit 3—Arbitration Lost (AL): This flag indicates that arbitration was lost in master mode. The I^2C bus interface monitors the bus. When two or more master devices attempt to seize the bus at nearly the same time, if the I^2C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master.

AL is cleared by reading AL after it has been set to 1, then writing 0 in AL. In addition, AL is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 3

AL	Description	
0	Bus arbitration won	(Initial value)
	[Clearing conditions]	
	1. When ICDR data is written (transmit mode) or read (receive mode)	
	2. When 0 is written in AL after reading AL = 1	
1	Arbitration lost	
	[Setting conditions]	
	1. If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode	
	2. If the internal SCL line is high at the fall of SCL in master transmit mode	

Bit 2—Slave Address Recognition Flag (AAS): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR, or if the general call address (H'00) is detected.

AAS is cleared by reading AAS after it has been set to 1, then writing 0 in AAS. In addition, AAS is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 2

AAS	Description	
0	Slave address or general call address not recognized	(Initial value)
	[Clearing conditions]	
	1. When ICDR data is written (transmit mode) or read (receive mode)	
	2. When 0 is written in AAS after reading AAS = 1	
	3. In master mode	
1	Slave address or general call address recognized	
	[Setting condition]	
	When the slave address or general call address is detected in slave receive mode while FS = 0	

Bit 1—General Call Address Recognition Flag (ADZ): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition is the general call address (H'00).

ADZ is cleared by reading ADZ after it has been set to 1, then writing 0 in ADZ. In addition, ADZ is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 1

ADZ	Description	
0	General call address not recognized	(Initial value)
	[Clearing conditions]	
	1. When ICDR data is written (transmit mode) or read (receive mode)	
	2. When 0 is written in ADZ after reading ADZ = 1	
	3. In master mode	
1	General call address recognized	
	[Setting condition]	
	When the general call address is detected in slave receive mode while FSX = 0 or FS = 0	

Bit 0—Acknowledge Bit (ACKB): Stores acknowledge data. In transmit mode, after the receiving device receives data, it returns acknowledge data, and this data is loaded into ACKB. In receive mode, after data has been received, the acknowledge data set in this bit is sent to the transmitting device.

When this bit is read, in transmission (when TRS = 1), the value loaded from the bus line (returned by the receiving device) is read. In reception (when TRS = 0), the value set by internal software is read.

Bit 0

ACKB	Description
0	Receive mode: 0 is output at acknowledge output timing (Initial value) Transmit mode: Indicates that the receiving device has acknowledged the data (signal is 0)
1	Receive mode: 1 is output at acknowledge output timing Transmit mode: Indicates that the receiving device has not acknowledged the data (signal is 1)

16.2.7 Serial/Timer Control Register (STCR)

Bit	7	6	5	4	3	2	1	0
	—	IICX1	IICX0	IICE	FLSHE	—	ICKS1	ICKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

STCR is an 8-bit readable/writable register that controls register access, the I^2C interface operating mode (when the on-chip IIC option is included), and on-chip flash memory (F-ZTAT versions), and selects the TCNT input clock source. For details of functions not related to the I^2C bus interface, see section 3.2.4, Serial/Timer Control Register (STCR), and the descriptions of the relevant modules. If a module controlled by STCR is not used, do not write 1 to the corresponding bit.

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

Bit 7—Reserved: This bit must not be set to 1.

Bits 6 and 5—I^2C Transfer Select 1 and 0 (IICX1, IICX0): These bits, together with bits CKS2 to CKS0 in ICMR, select the transfer rate in master mode. For details, see section 16.2.4, I^2C Bus Mode Register (ICMR).

Bit 4—I²C Master Enable (IICE): Controls CPU access to the I²C bus interface data and control registers (ICCR, ICSR, ICDR/SARX, ICMR/SAR).

Bit 4 IICE	Description	
0	CPU access to I²C bus interface data and control registers is disabled	(Initial value)
1	CPU access to I²C bus interface data and control registers is enabled	

Bit 3—Flash Memory Control Register Enable (FLSHE): Controls the operation of the flash memory in F-ZTAT versions. For details, see section 19, ROM.

Bit 2—Reserved: This bit must not be set to 1.

Bits 1 and 0—Internal Clock Source Select 1 and 0 (ICKS1, ICSK0): These bits, together with bits CKS2 to CKS0 in TCR, select the clock input to the timer counters (TCNT). For details, see section 12.2.4, Timer Control Register.

16.2.8 DDC Switch Register (DDCSWR)

Bit	7	6	5	4	3	2	1	0
	SWE	SW	IE	IF	CLR3	CLR2	CLR1	CLR0
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/(W)*1	W*2	W*2	W*2	W*2

Notes: 1. Only 0 can be written, to clear the flag.
2. Always read as 1.

DDCSWR is an 8-bit readable/writable register that controls the IIC channel 0 automatic format switching function and IIC internal latch clearance.

DDCSWR is initialized to H'0F by a reset and in hardware standby mode.

Bit 7—DDC Mode Switch Enable (SWE): Selects the function for automatically switching IIC channel 0 from formatless mode to the I²C bus format.

Bit 7 SWE	Description	
0	Automatic switching of IIC channel 0 from formatless mode to I²C bus format is disabled	(Initial value)
1	Automatic switching of IIC channel 0 from formatless mode to I²C bus format is enabled	

Bit 6—DDC Mode Switch (SW): Selects either formatless mode or the I^2C bus format for IIC channel 0.

Bit 6

SW	Description	
0	IIC channel 0 is used with the I^2C bus format	(Initial value)
	[Clearing conditions]	
	1. When 0 is written by software	
	2. When a falling edge is detected on the SCL pin when SWE = 1	
1	IIC channel 0 is used in formatless mode	
	[Setting condition]	
	When 1 is written in SW after reading SW = 0	

Bit 5—DDC Mode Switch Interrupt Enable Bit (IE): Enables or disables an interrupt request to the CPU when automatic format switching is executed for IIC channel 0.

Bit 5

IE	Description	
0	Interrupt when automatic format switching is executed is disabled	(Initial value)
1	Interrupt when automatic format switching is executed is enabled	

Bit 4—DDC Mode Switch Interrupt Flag (IF): Flag that indicates an interrupt request to the CPU when automatic format switching is executed for IIC channel 0.

Bit 4

IF	Description	
0	No interrupt is requested when automatic format switching is executed	(Initial value)
	[Clearing condition]	
	When 0 is written in IF after reading IF = 1	
1	An interrupt is requested when automatic format switching is executed	
	[Setting condition]	
	When a falling edge is detected on the SCL pin when SWE = 1	

Bits 3 to 0—IIC Clear 3 to 0 (CLR3 to CLR0): These bits control initialization of the internal state of IIC0 and IIC1.

These bits can only be written to; if read they will always return a value of 1.

When a write operation is performed on these bits, a clear signal is generated for the internal latch circuit of the corresponding module(s),and the internal state of the IIC module(s) is initialized.

The write data for these bits is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR.

When clearing is required again, all the bits must be written to in accordance with the setting.

Bit 3	Bit 2	Bit 1	Bit 0	
CLR3	**CLR2**	**CLR1**	**CLR0**	**Description**
0	0	—	—	Setting prohibited
	1	0	0	Setting prohibited
			1	IIC0 internal latch cleared
		1	0	IIC1 internal latch cleared
			1	IIC0 and IIC1 internal latches cleared
1	—	—	—	Invalid setting

16.2.9 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	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	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers, and is used to perform module stop mode control.

When the MSTP4 or MSTP3 bit is set to 1, operation of the corresponding IIC channel is halted at the end of the bus cycle, and a transition is made to module stop mode. For details, see section 21.5, Module Stop Mode.

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

MSTPCRL Bit 4—Module Stop (MSTP4): Specifies IIC channel 0 module stop mode.

MSTPCRL Bit 4		
MSTP4	**Description**	
0	IIC channel 0 module stop mode is cleared	
1	IIC channel 0 module stop mode is set	(Initial value)

MSTPCRL Bit 3—Module Stop (MSTP3): Specifies IIC channel 1 module stop mode.

MSTPCRL Bit 3		
MSTP3	**Description**	
0	IIC channel 1 module stop mode is cleared	
1	IIC channel 1 module stop mode is set	(Initial value)

16.3 Operation

16.3.1 I²C Bus Data Format

The I²C bus interface has serial and I²C bus formats.

The I²C bus formats are addressing formats with an acknowledge bit. These are shown in figures 16.3 (a) and (b). The first frame following a start condition always consists of 8 bits.

IIC channel 0 only is capable of formatless operation, as shown in figure 16.3 (c).

The serial format is a non-addressing format with no acknowledge bit. This is shown in figure 16.4.

Figure 16.5 shows the I²C bus timing.

The symbols used in figures 16.3 to 16.5 are explained in table 16.4.

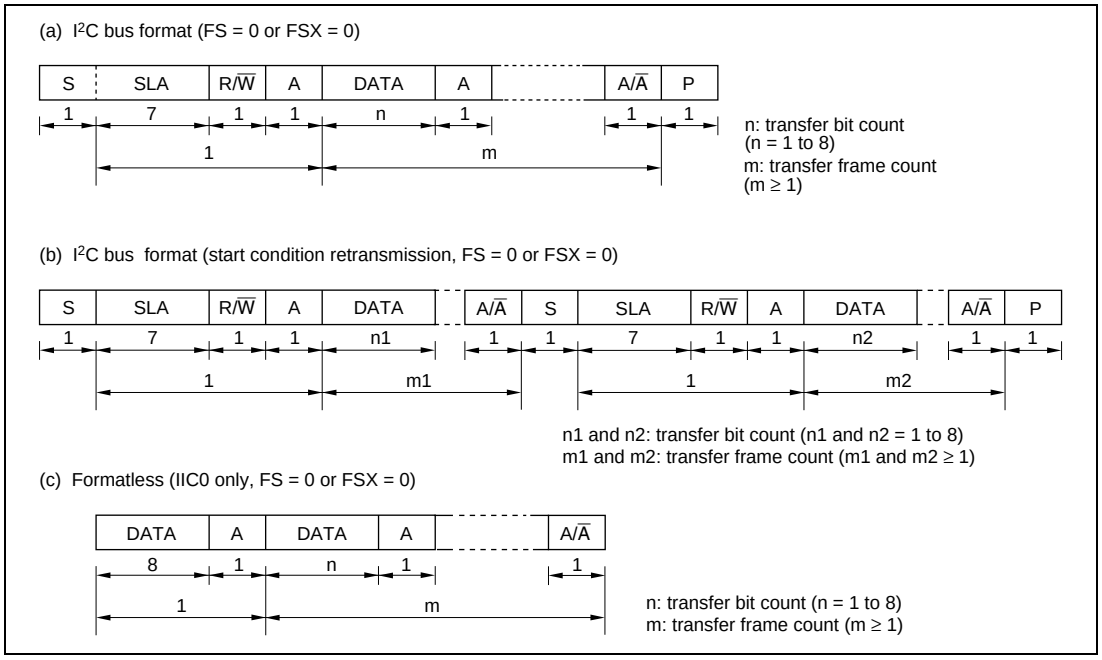


Figure 16.3 I²C Bus Data Formats (I²C Bus Formats)

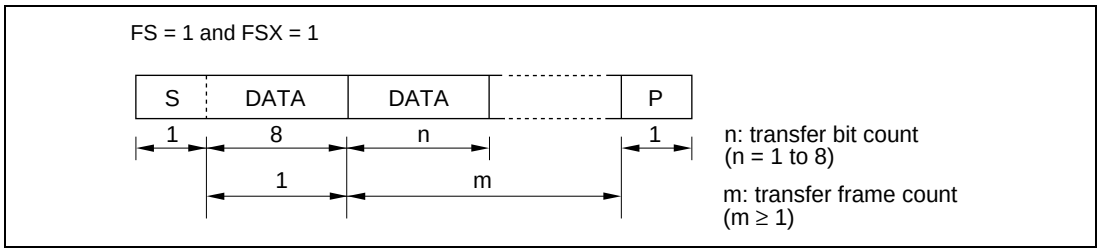


Figure 16.4 I²C Bus Data Format (Serial Format)

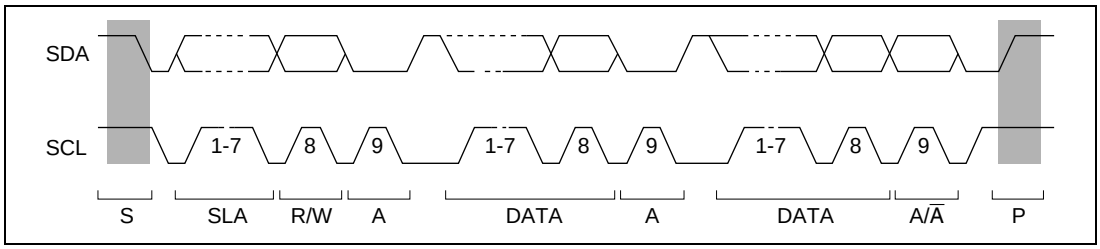


Figure 16.5 I²C Bus Timing

Table 16.4 I²C Bus Data Format Symbols

Legend	
S	Start condition. The master device drives SDA from high to low while SCL is high
SLA	Slave address, by which the master device selects a slave device
R/$\overline{W}$	Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0
A	Acknowledge. The receiving device (the slave in master transmit mode, or the master in master receive mode) drives SDA low to acknowledge a transfer
DATA	Transferred data. The bit length is set by bits BC2 to BC0 in ICMR. The MSB-first or LSB-first format is selected by bit MLS in ICMR
P	Stop condition. The master device drives SDA from low to high while SCL is high

16.3.2 Master Transmit Operation

In I^2C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The transmission procedure and operations are described below.

[1] Set the ICE bit in ICCR to 1. Set bits MLS, WAIT, and CKS2 to CKS0 in ICMR, and bit IICX in STCR, according to the operating mode.

[2] Read the BBSY flag in ICCR to confirm that the bus is free, then set bits MST and TRS to 1 in ICCR to select master transmit mode. Next, write 1 to BBSY and 0 to SCP. This changes SDA from high to low when SCL is high, and generates the start condition. The TDRE internal flag is then set to 1, and the IRIC and IRTR flags are also set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.

[3] With the I^2C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the first frame data following the start condition indicates the 7-bit slave address and transmit/receive direction. Write the data (slave address + R/W) to ICDR. The TDRE internal flag is then cleared to 0. The written address data is transferred to ICDRS, and the TDRE internal flag is set to 1 again. This is identified as indicating the end of the transfer, and so the IRIC flag is cleared to 0. The master device sequentially sends the transmit clock and the data written to ICDR using the timing shown in figure 16.6. The selected slave device (i.e. the slave device with the matching slave address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge signal.

[4] When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. If the TDRE internal flag has been set to 1, after one frame has been transmitted SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.

[5] To continue transfer, write the next data to be transmitted into ICDR. After the data has been transferred to ICDRS and the TDRE internal flag has been set to 1, clear the IRIC flag to 0. Transmission of the next frame is performed in synchronization with the internal clock.

Data can be transmitted sequentially by repeating steps [4] and [5]. To end transmission, clear the IRIC flag, write H'FF dummy data to ICDR after the last data has been transmitted (when ICDRT does not contain the next transmit data), and then write 0 to BBSY and SCP in ICCR when the IRIC flag is set again. This changes SDA from low to high when SCL is high, and generates the stop condition.

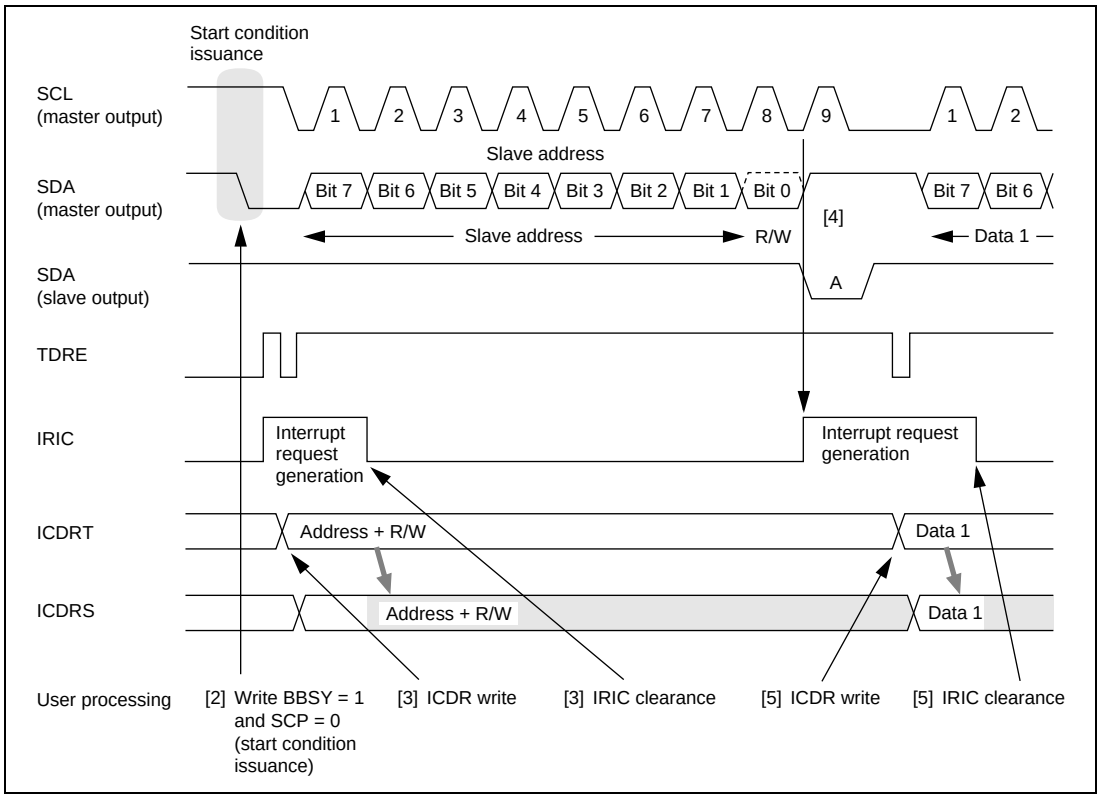


Figure 16.6 Example of Master Transmit Mode Operation Timing (MLS = WAIT = 0)

To transmit data continuously:

[6] Before the rise of the 9th transmit clock pulse for the data being transmitted, clear the IRIC flag to 0 and then write the next transmit data to ICDR.

[7] When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. At the same time, the next transmit data written into ICDR (ICDRT) is transferred to ICDRS, the TDRE internal flag is set to 1, and then the next frame is transmitted in synchronization with the internal clock.

Data can be transmitted continuously by repeating steps [6] and [7].

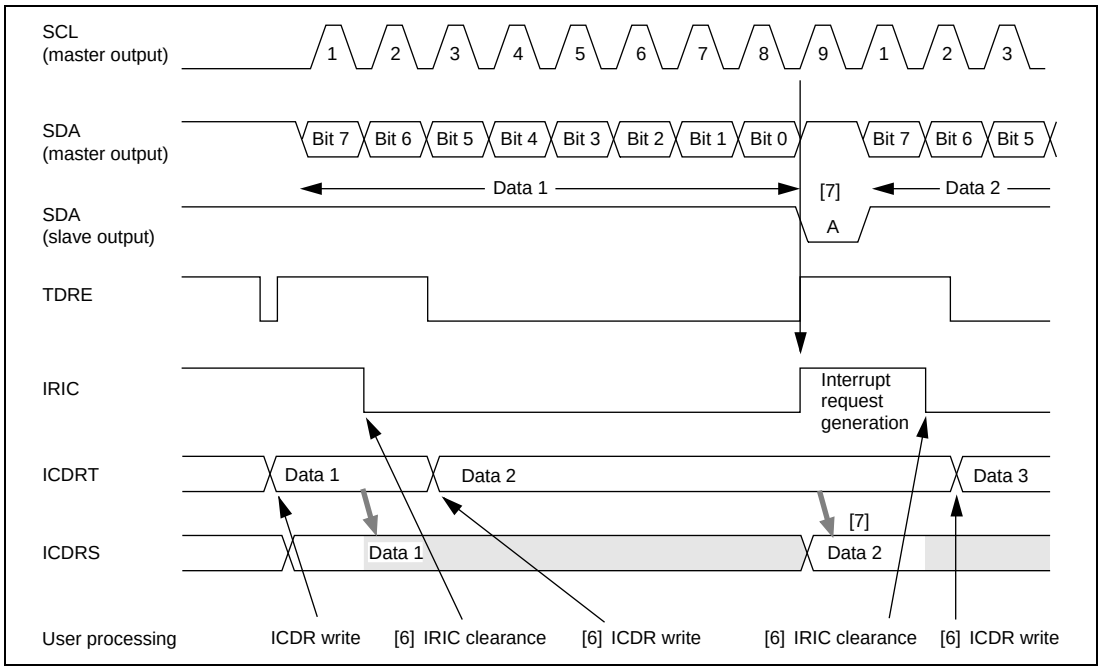


Figure 16.7 Example of Master Transmit Mode Continuous Transmit Operation Timing (MLS = WAIT = 0)

16.3.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data, and returns an acknowledge signal. The slave device transmits data. The reception procedure and operations in master receive mode are described below.

[1] Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode. Also clear the ACKB bit in ICSR to 0 (acknowledge data setting).

[2] When ICDR is read (dummy data read), reception is started, and the receive clock is output, and data received, in synchronization with the internal clock. In order to determine the end of reception, the IRIC flag in ICCR must be cleared beforehand.

[3] The master device drives SDA at the 9th receive clock pulse to return an acknowledge signal. When one frame of data has been received, the IRIC flag in ICCR is set to 1 at the rise of the 9th receive clock pulse. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the RDRF internal flag has been cleared to 0, it is set to 1, and the receive operation continues. If reception of the next frame ends before the ICDR read/IRIC flag clearing in [4] is performed, SCL is automatically fixed low in synchronization with the internal clock.

[4] Read ICDR and clear the IRIC flag in ICCR to 0. The RDRF flag is cleared to 0.

Data can be received continuously by repeating steps [3] and [4]. As the RDRF internal flag is cleared to 0 when reception is started after initially switching from master transmit mode to master receive mode, reception of the next frame of data is started automatically. To halt reception, the TRS bit must be set to 1 before the rise of the receive clock for the next frame.

To halt reception, set the TSR bit to 1, read ICDR, then write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

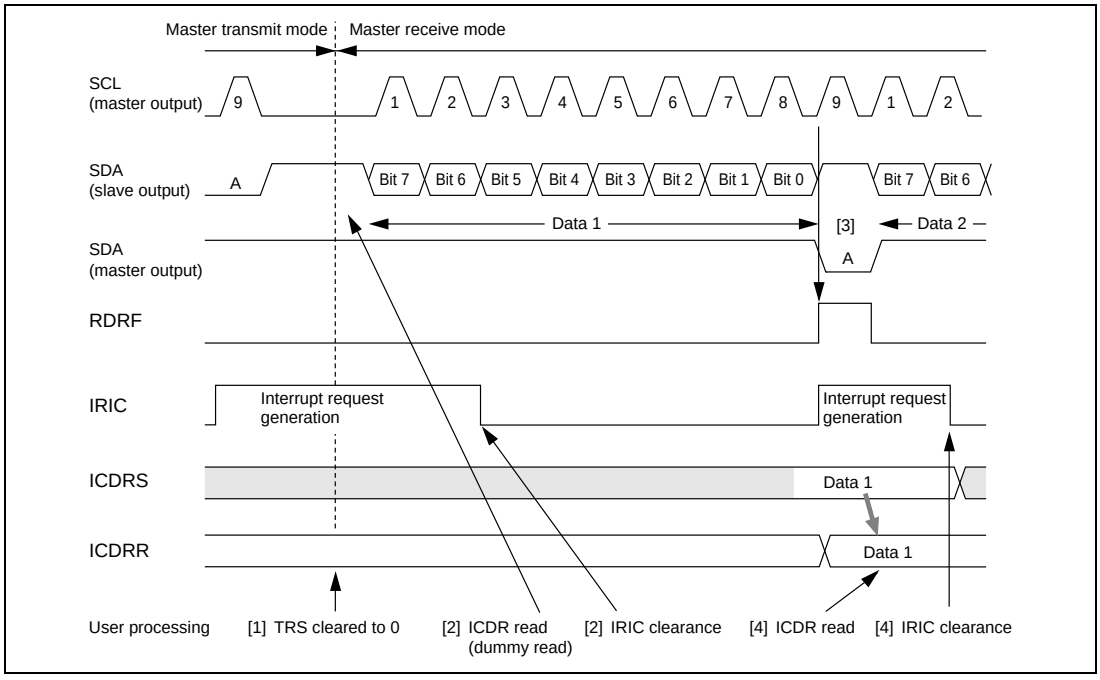


Figure 16.8 Example of Master Receive Mode Operation Timing (MLS = WAIT = ACKB = 0)

16.3.4 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The reception procedure and operations in slave receive mode are described below.

[1] Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

[2] When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1.

[3] When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit (R/W) is 0, the TRS bit in ICCR remains cleared to 0, and slave receive operation is performed.

[4] At the 9th clock pulse of the receive frame, the slave device drives SDA low and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the RDRF internal flag has been cleared to 0, it is set to 1, and the receive operation continues. If the RDRF internal flag has been set to 1, the slave device drives SCL low from the fall of the receive clock until data is read into ICDR.

[5] Read ICDR and clear the IRIC flag in ICCR to 0. The RDRF flag is cleared to 0.

Receive operations can be performed continuously by repeating steps [4] and [5]. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

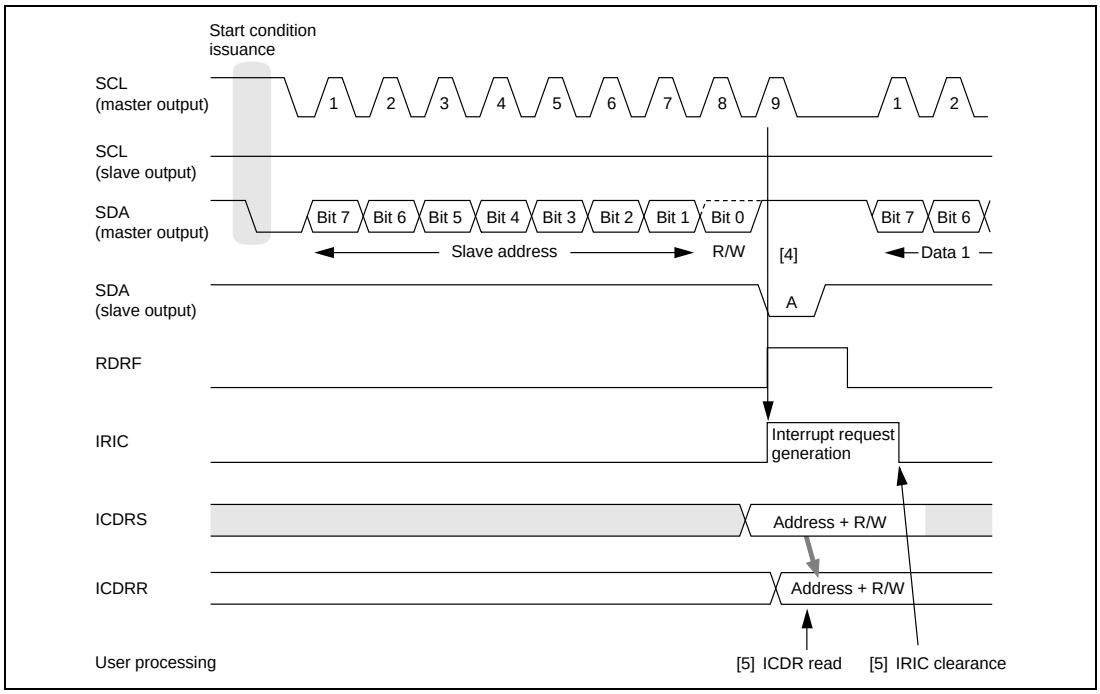


Figure 16.9 Example of Slave Receive Mode Operation Timing (1) (MLS = ACKB = 0)

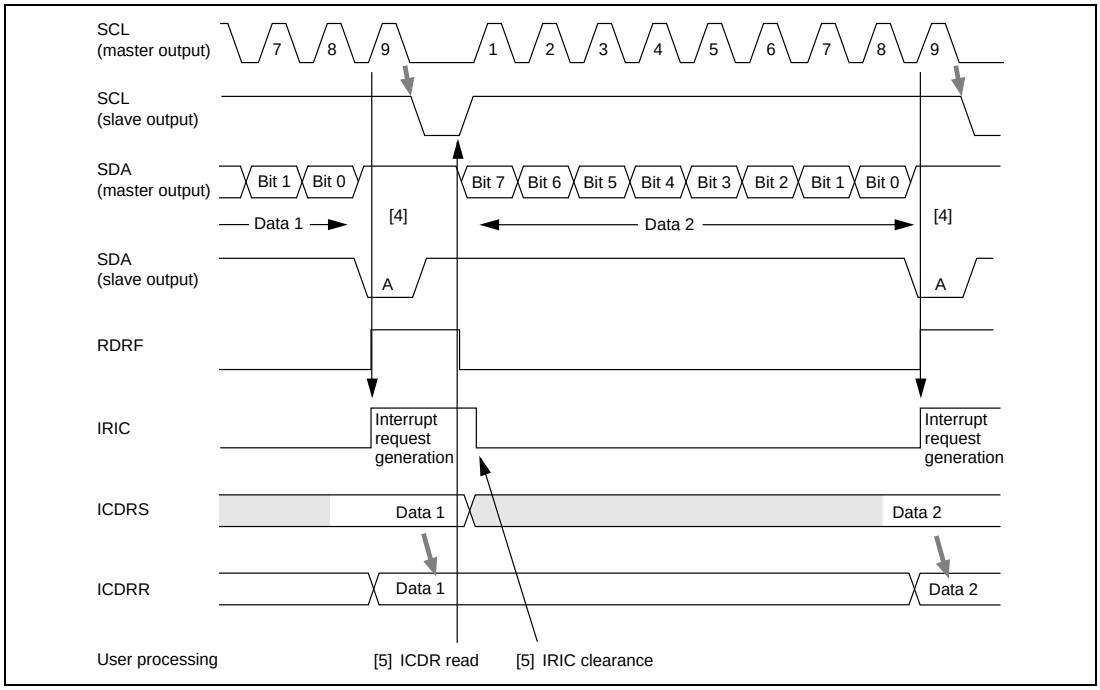


Figure 16.10 Example of Slave Receive Mode Operation Timing (2) (MLS = ACKB = 0)

16.3.5 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

[1] Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

[2] When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. .If the 8th data bit (R/W) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The TDRF flag is set to 1. The slave device drives SCL low from the fall of the transmit clock until ICDR data is written.

[3] After clearing the IRIC flag to 0, write data to ICDR. The TDRE internal flag is cleared to 0. The written data is transferred to ICDRS, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again. After clearing the IRIC flag to 0, write the next data to ICDR. The slave device sequentially sends the data written into ICDR in accordance with the clock output by the master device at the timing shown in figure 16.11.

[4] When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. If the TDRE internal flag has been set to 1, this slave device drives SCL low from the fall of the transmit clock until data is written to ICDR. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed normally. When the TDRE internal flag is 0, the data written into ICDR is transferred to ICDRS, transmission is started, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again.

[5] To continue transmission, clear the IRIC flag to 0, then write the next data to be transmitted into ICDR. The TDRE flag is cleared to 0.

Transmit operations can be performed continuously by repeating steps [4] and [5]. To end transmission, write H'FF to ICDR to release SDA on the slave side. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

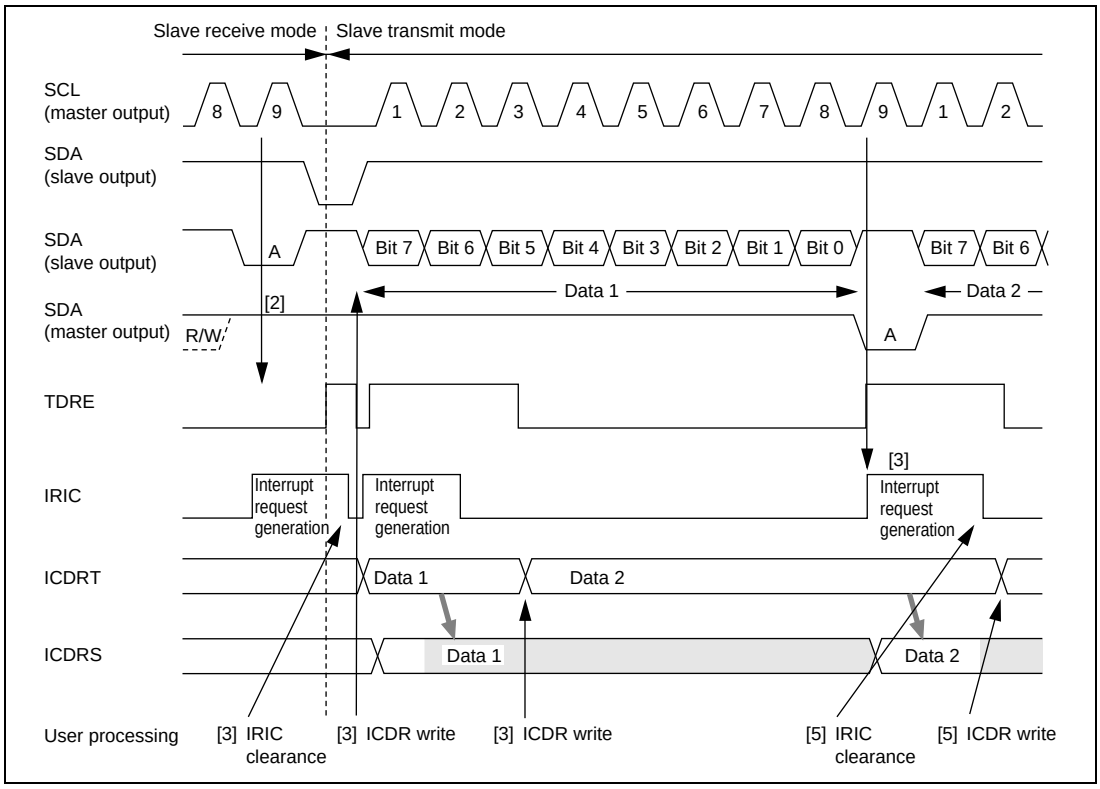


Figure 16.11 Example of Slave Transmit Mode Operation Timing (MLS = 0)

16.3.6 IRIC Setting Timing and SCL Control

The interrupt request flag (IRIC) is set at different times depending on the WAIT bit in ICMR, the FS bit in SAR, and the FSX bit in SARX. If the TDRE or RDRF internal flag is set to 1, SCL is automatically held low after one frame has been transferred; this timing is synchronized with the internal clock. Figure 16.12 shows the IRIC set timing and SCL control.

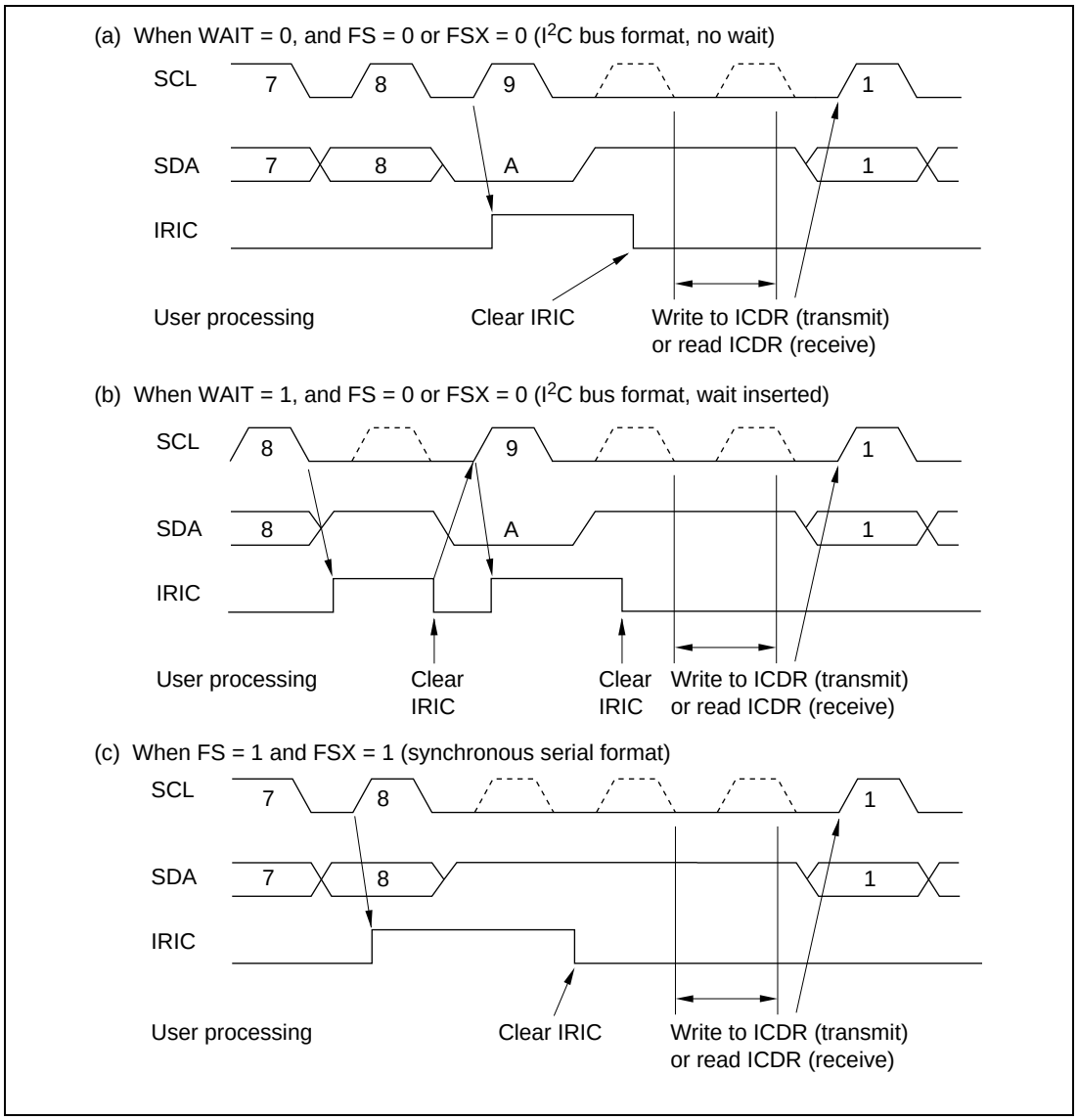


Figure 16.12 IRIC Setting Timing and SCL Control

16.3.7 Automatic Switching from Formatless Mode to I²C Bus Format

Setting the SW bit to 1 in DDCSWR enables formatless mode to be selected as the IIC0 operating mode. Switching from formatless mode to the I²C bus format (slave mode) is performed automatically when a falling edge is detected on the SCL pin.

The following four preconditions are necessary for this operation:

- A common data pin (SDA) for formatless and I²C bus format operation
- Separate clock pins for formatless operation (VSYNCI) and I²C bus format operation (SCL)
- A fixed 1 level for the SCL pin during format operation
- Settings of bits other than TRS in ICCR that allow I²C bus format operation

Automatic switching is performed from formatless mode to the I²C bus format when the SW bit in DDCSWR is automatically cleared to 0 on detection of a falling edge on the SCL pin. Switching from the I²C bus format to formatless mode is achieved by having software set the SW bit in DDCSWR to 1.

In formatless mode, bits (such as MSL and TRS) that control the I²C bus interface operating mode must not be modified. When switching from the I²C bus format to formatless mode, set the TRS bit to 1 or clear it to 0 according to the transmit data (transmission or reception) in formatless mode, then set the SW bit to 1. After automatic switching from formatless mode to the I²C bus format (slave mode), in order to wait for slave address reception, the TRS bit is automatically cleared to 0.

If a falling edge is detected on the SCL pin during formatless operation, the I²C bus interface format is switched at that point, without waiting for a stop condition.

16.3.8 Operation Using the DTC

The I^2C bus format provides for selection of the slave device and transfer direction by means of the slave address and the R/W bit, confirmation of reception with the acknowledge bit, indication of the last frame, and so on. Therefore, continuous data transfer using the DTC must be carried out in conjunction with CPU processing by means of interrupts.

Table 16.5 shows some examples of processing using the DTC. These examples assume that the number of transfer data bytes is known in slave mode.

Table 16.5 Examples of Operation Using the DTC

Item	Master Transmit Mode	Master Receive Mode	Slave Transmit Mode	Slave Receive Mode
Slave address + R/W bit transmission/ reception	Transmission by DTC (ICDR write)	Transmission by CPU (ICDR write)	Reception by CPU (ICDR read)	Reception by CPU (ICDR read)
Dummy data read	—	Processing by CPU (ICDR read)	—	—
Actual data transmission/ reception	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)
Dummy data (H'FF) write	—	—	Processing by DTC (ICDR write)	—
Last frame processing	Not necessary	Reception by CPU (ICDR read)	Not necessary	Reception by CPU (ICDR read)
Transfer request processing after last frame processing	1st time: Clearing by CPU 2nd time: End condition issuance by CPU	Not necessary	Automatic clearing on detection of end condition during transmission of dummy data (H'FF)	Not necessary
Setting of number of DTC transfer data frames	Transmission: Actual data count + 1 (+1 equivalent to slave address + R/W bits)	Reception: Actual data count	Transmission: Actual data count + 1 (+1 equivalent to dummy data (H'FF))	Reception: Actual data count

16.3.9 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 16.13 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

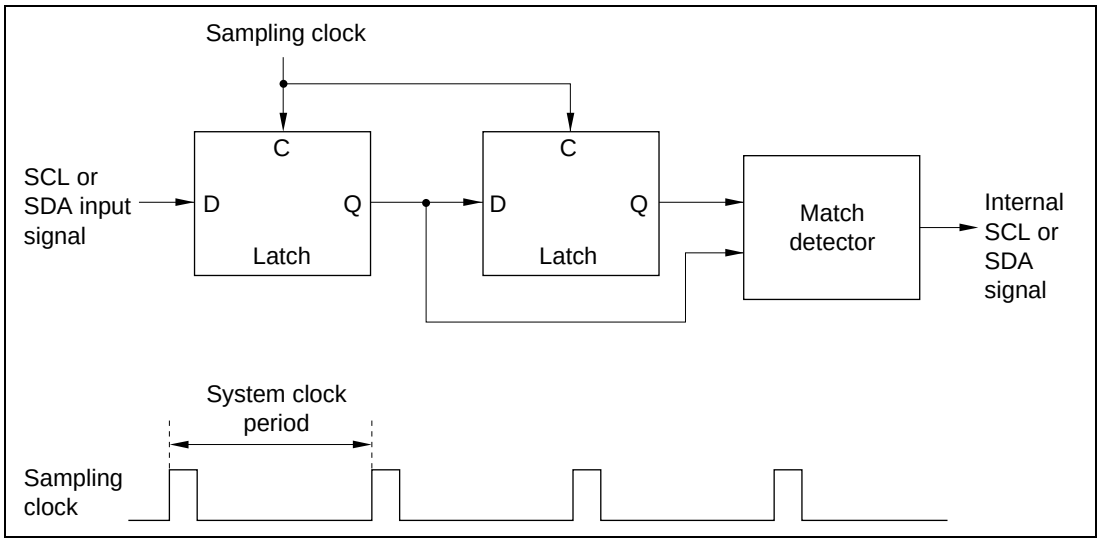


Figure 16.13 Block Diagram of Noise Canceler

16.3.10 Sample Flowcharts

Figures 16.14 to 16.17 show sample flowcharts for using the I^2C bus interface in each mode.

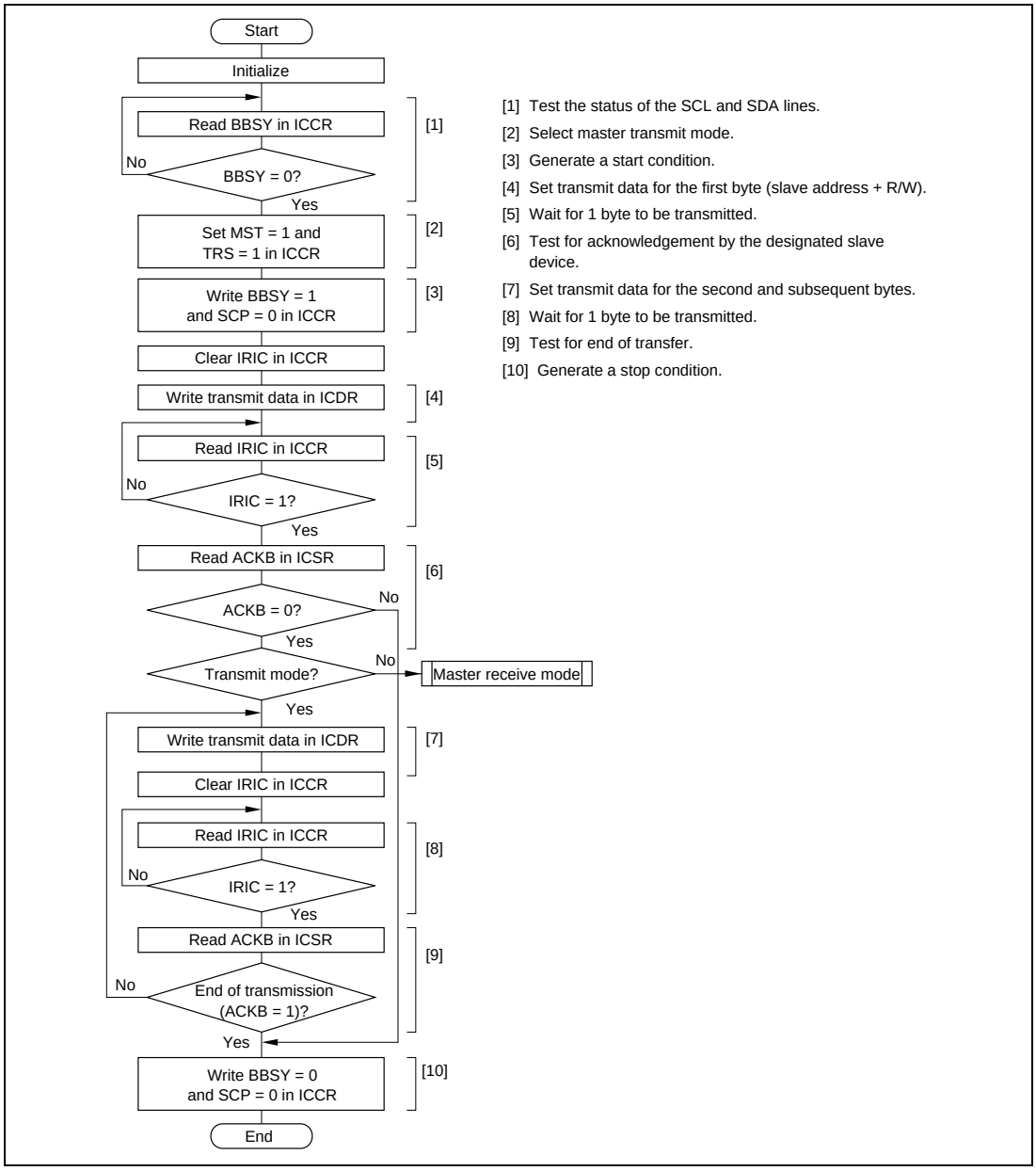


Figure 16.14 Flowchart for Master Transmit Mode (Example)

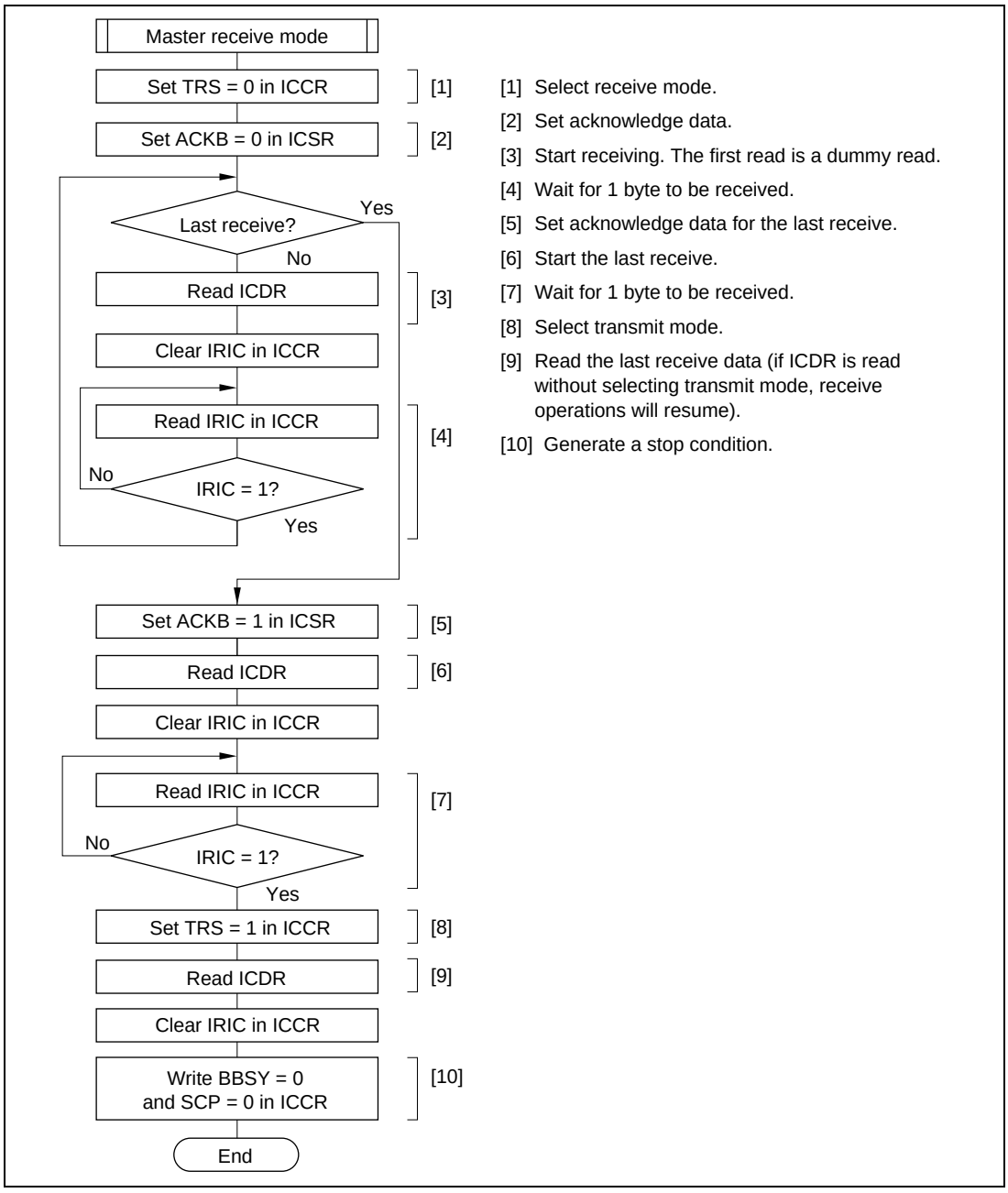


Figure 16.15 Flowchart for Master Receive Mode (Example)

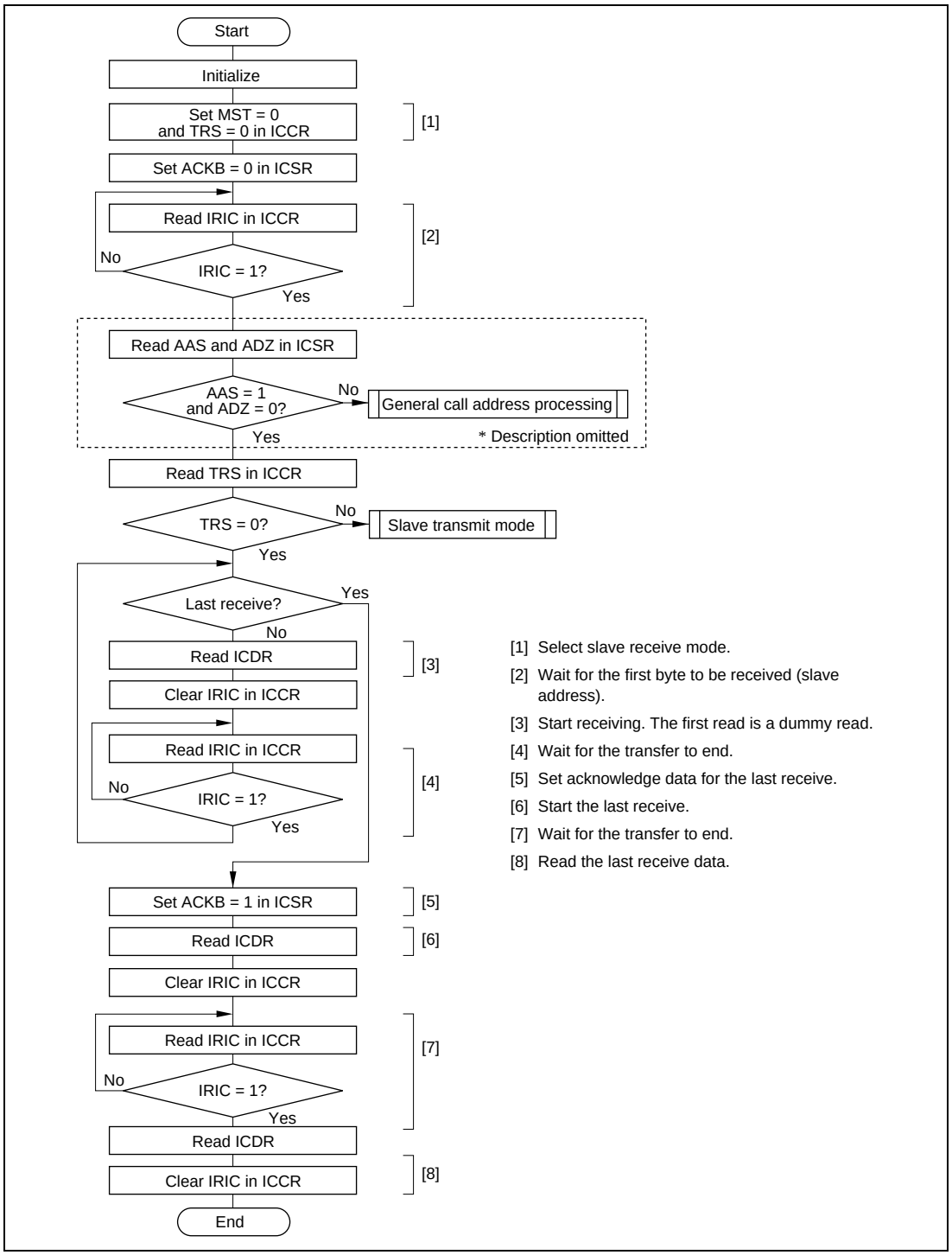


Figure 16.16 Flowchart for Slave Receive Mode (Example)

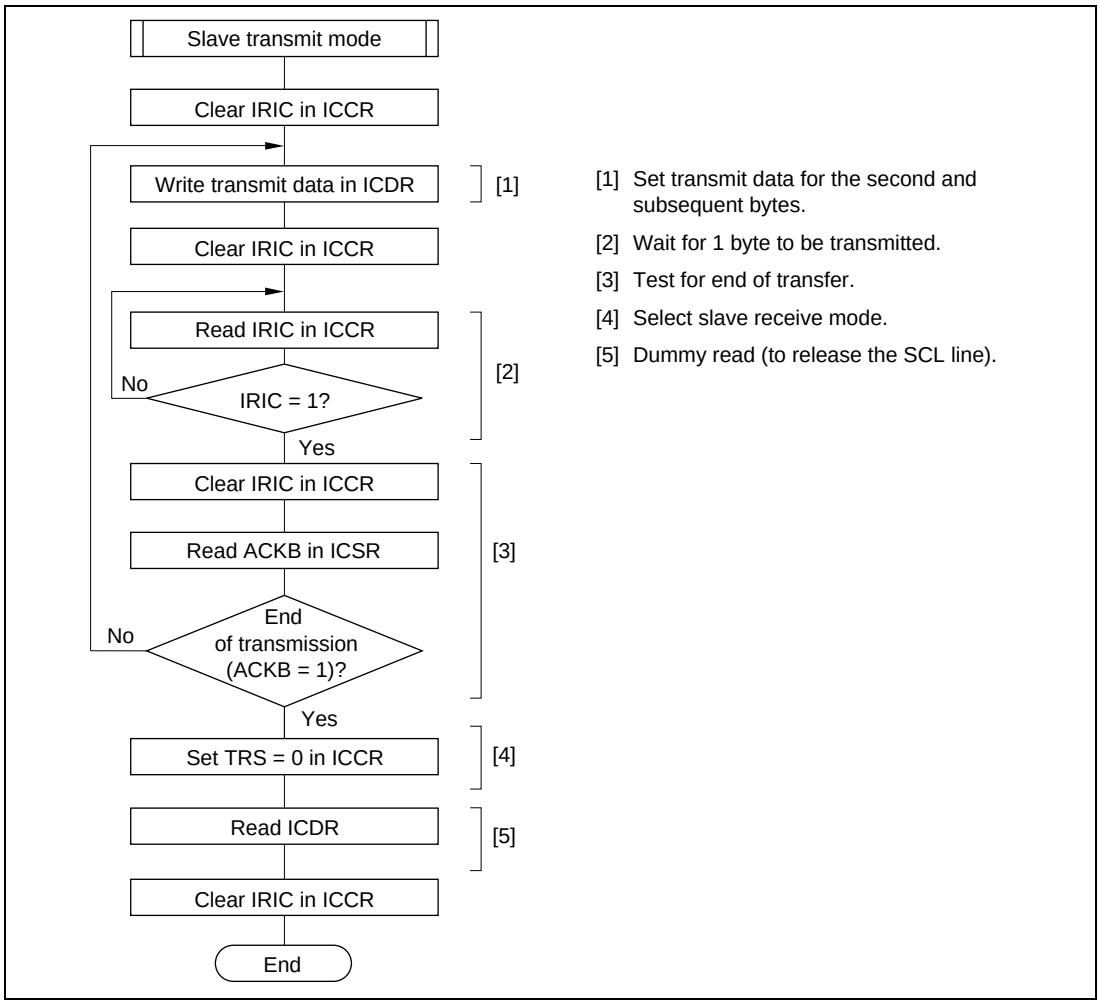


Figure 16.17 Flowchart for Slave Transmit Mode (Example)

16.3.11 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs during communication.

Initialization is executed in accordance with the setting of bits CLR3 to CLR0 in the DDCSWR register. For details see section 16.2.8, DDC Switch Register (DDCSWR).

(1) Scope of Initialization

The initialization executed by this function covers the following items:

- TDRE and RDRF internal flags
- Transmit/receive sequencer and internal operating clock counter
- Internal latches for retaining the output state of the SCL and SDA pins (wait, clock, data output, etc.)

The following items are not initialized:

- Actual register values (ICDR, SAR, SARX, ICMR, ICCR, ICSR, DDCSWR, STCR)
- Internal latches used to retain register read information for setting/clearing flags in the ICMR, ICCR, ICSR, and DDCSWR registers
- The value of the ICMR register bit counter (BC2 to BC0)
- Generated interrupt sources (interrupt sources transferred to the interrupt controller)

(2) Notes on Initialization

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures must be taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures must be taken as necessary.
- The write data for bits CLR3 to CLR0 is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR. Similarly, when clearing is required again, all the bits must be written to simultaneously in accordance with the setting.
- If a flag clearing setting is made during transmission/reception, the IIC module will stop transmitting/receiving at that point and the SCL and SDA pins will be released. When transmission/reception is started again, register initialization, etc., must be carried out as necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, but since the stop condition pin waveform is generated according to the state and release timing of the SCL and SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other bits and flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used when initializing the IIC state.

(1) Execute initialization of the internal state according to the setting of bits CLR3 to CLR0.
(2) Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the BBST bit to 0, and wait for two transfer rate clock cycles.
(3) Re-execute initialization of the internal state according to the setting of bits CLR3 to CLR0.
(4) Initialize (re-set) the IIC registers.

16.4 Usage Notes

- In master mode, if an instruction to generate a start condition is immediately followed by an instruction to generate a stop condition, neither condition will be output correctly. To output consecutive start and stop conditions, after issuing the instruction that generates the start condition, read the relevant ports, check that SCL and SDA are both low, then issue the instruction that generates the stop condition. Note that SCL may not yet have gone low when BBSY is cleared to 0.

- Either of the following two conditions will start the next transfer. Pay attention to these conditions when reading or writing to ICDR.
  - Write access to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICDRT to ICDRS)
  - Read access to ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRS to ICDRR)
- Table 16.6 shows the timing of SCL and SDA output in synchronization with the internal clock. Timings on the bus are determined by the rise and fall times of signals affected by the bus load capacitance, series resistance, and parallel resistance.

Table 16.6 I^2C Bus Timing (SCL and SDA Output)

Item	Symbol	Output Timing	Unit	Notes
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $256t_{cyc}$	ns	Figure 22.25 (reference)
SCL output high pulse width	t_{SCLHO}	$0.5t_{SCLO}$	ns	
SCL output low pulse width	t_{SCLLO}	$0.5t_{SCLO}$	ns	
SDA output bus free time	t_{BUFO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Start condition output hold time	t_{STAHO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Retransmission start condition output setup time	t_{STASO}	$1t_{SCLO}$	ns	
Stop condition output setup time	t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	t_{SDASO}	$1t_{SCLLO} – 3t_{cyc}$	ns	
Data output setup time (slave)		$1t_{SCLL} – (6t_{cyc}$ or $12t_{cyc}$*)		
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

Note: *$6t_{cyc}$ when IICX is 0, $12t_{cyc}$ when 1.

- SCL and SDA input is sampled in synchronization with the internal clock. The AC timing therefore depends on the system clock cycle t_{cyc}, as shown in table 22.9 in section 22, Electrical Characteristics. Note that the I^2C bus interface AC timing specifications will not be met with a system clock frequency of less than 5 MHz.
- The I^2C bus interface specification for the SCL rise time t_{sr} is under 1000 ns (300 ns for high-speed mode). In master mode, the I^2C bus interface monitors the SCL line and synchronizes one bit at a time during communication. If t_{Sr} (the time for SCL to go from low to V_{IH}) exceeds the time determined by the input clock of the I^2C bus interface, the high period of SCL is extended. The SCL rise time is determined by the pull-up resistance and load capacitance of the SCL line. To insure proper operation at the set transfer rate, adjust the pull-up resistance and load capacitance so that the SCL rise time does not exceed the values given in the table 16.7 below.

Table 16.7 Permissible SCL Rise Time (t_{Sr}) Values

			Time Indication					
IICX	t_{cyc} Indication		I^2C Bus Specification (Max.)	ø = 5 MHz	ø = 8 MHz	ø = 10 MHz	ø = 16 MHz	ø = 20 MHz
0	$7.5t_{cyc}$	Normal mode	1000 ns	1000 ns	937 ns	750 ns	468 ns	375 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns
1	$17.5t_{cyc}$	Normal mode	1000 ns	1000 ns	1000 ns	1000 ns	1000 ns	875 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns

- The I^2C bus interface specifications for the SCL and SDA rise and fall times are under 1000 ns and 300 ns. The I^2C bus interface SCL and SDA output timing is prescribed by t_{Scyc} and t_{cyc}, as shown in table 16.6. However, because of the rise and fall times, the I^2C bus interface specifications may not be satisfied at the maximum transfer rate. Table 16.8 shows output timing calculations for different operating frequencies, including the worst-case influence of rise and fall times.

  t_{BUFO} fails to meet the I^2C bus interface specifications at any frequency. The solution is either (a) to provide coding to secure the necessary interval (approximately 1 µs) between issuance of a stop condition and issuance of a start condition, or (b) to select devices whose input timing permits this output timing for use as slave devices connected to the I^2C bus.

  t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I^2C bus interface specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be investigated include (a) adjusting the rise and fall times by means of a pull-up resistor and capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecting devices whose input timing permits this output timing for use as slave devices connected to the I^2C bus.

Table 16.8 I^2C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf})

			Time Indication (at Maximum Transfer Rate) [ns]						
Item	**t_{cyc} Indication**		**t_{Sr}/t_{Sf} Influence (Max.)**	**I^2C Bus Specification (Min.)**	**ø = 5 MHz**	**ø = 8 MHz**	**ø = 10 MHz**	**ø = 16 MHz**	**ø = 20 MHz**
t_{SCLHO}	$0.5t_{SCLO}$ ($-t_{Sr}$)	Standard mode	–1000	4000	4000	4000	4000	4000	4000
		High-speed mode	–300	600	950	950	950	950	950
t_{SCLLO}	$0.5t_{SCLO}$ ($-t_{Sf}$)	Standard mode	–250	4700	4750	4750	4750	4750	4750
		High-speed mode	–250	1300	1000*[1]	1000*[1]	1000*[1]	1000*[1]	1000*[1]
t_{BUFO}	$0.5t_{SCLO} - 1t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	4700	3800*[1]	3875*[1]	3900*[1]	3938*[1]	3950*[1]
		High-speed mode	–300	1300	750*[1]	825*[1]	850*[1]	888*[1]	900*[1]
t_{STAHO}	$0.5t_{SCLO} - 1t_{cyc}$ ($-t_{Sf}$)	Standard mode	–250	4000	4550	4625	4650	4688	4700
		High-speed mode	–250	600	800	875	900	938	950
t_{STASO}	$1t_{SCLO}$ ($-t_{Sr}$)	Standard mode	–1000	4700	9000	9000	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200	2200	2200
t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	4000	4400	4250	4200	4125	4100
		High-speed mode	–300	600	1350	1200	1150	1075	1050
t_{SDASO} (master)	$1t_{SCLLO}$*[3] – $3t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	250	3100	3325	3400	3513	3550
		High-speed mode	–300	100	400	625	700	813	850
t_{SDASO} (slave)	$1t_{SCLL}$*[3] – $12t_{cyc}$*[2] ($-t_{Sr}$)	Standard mode	–1000	250	1300	2200	2500	2950	3100
		High-speed mode	–300	100	–1400*[1]	–500*[1]	–200*[1]	250	400

Table 16.8 I^2C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf}) (cont)

			Time Indication (at Maximum Transfer Rate) [ns]						
Item	**t_{cyc} Indication**		**t_{Sr}/t_{Sf} Influence (Max.)**	**I^2C Bus Specification (Min.)**	**ø = 5 MHz**	**ø = 8 MHz**	**ø = 10 MHz**	**ø = 16 MHz**	**ø = 20 MHz**
t_{SDAHO}	$3t_{cyc}$	Standard mode	0	0	600	375	300	188	150
		High-speed mode	0	0	600	375	300	188	150

Notes: 1. Does not meet the I^2C bus interface specification. Remedial action such as the following is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the rise and fall times by means of a pull-up resistor and capacitive load; (c) reduce the transfer rate; (d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the IICX bit and bits CKS0 to CKS2. Depending on the frequency it may not be possible to achieve the maximum transfer rate; therefore, whether or not the I^2C bus interface specifications are met must be determined in accordance with the actual setting conditions.

2. Value when the IICX bit is set to 1. When the IICX bit is cleared to 0, the value is ($t_{SCLL} - 6t_{cyc}$).
3. Calculated using the I^2C bus specification values (standard mode: 4700 ns min.; high-speed mode: 1300 ns min.).

- Note on ICDR Read at End of Master Reception

  To halt reception at the end of a receive operation in master receive mode, set the TRS bit to 1 and write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition. After this, receive data can be read by means of an ICDR read, but if data remains in the buffer the ICDRS receive data will not be transferred to ICDR, and so it will not be possible to read the second byte of data.

  If it is necessary to read the second byte of data, issue the stop condition in master receive mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm that the BBSY bit in the ICCR register is cleared to 0, the stop condition has been generated, and the bus has been released, then read the ICDR register with TRS cleared to 0.

  Note that if the receive data (ICDR data) is read in the interval between execution of the instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR) and the actual generation of the stop condition, the clock may not be output correctly in subsequent master transmission.

  Clearing of the MST bit after completion of master transmission/reception, or other modifications of IIC control bits to change the transmit/receive operating mode or settings, must be carried out during interval (a) in figure 16.18 (after confirming that the BBSY bit has been cleared to 0 in the ICCR register).

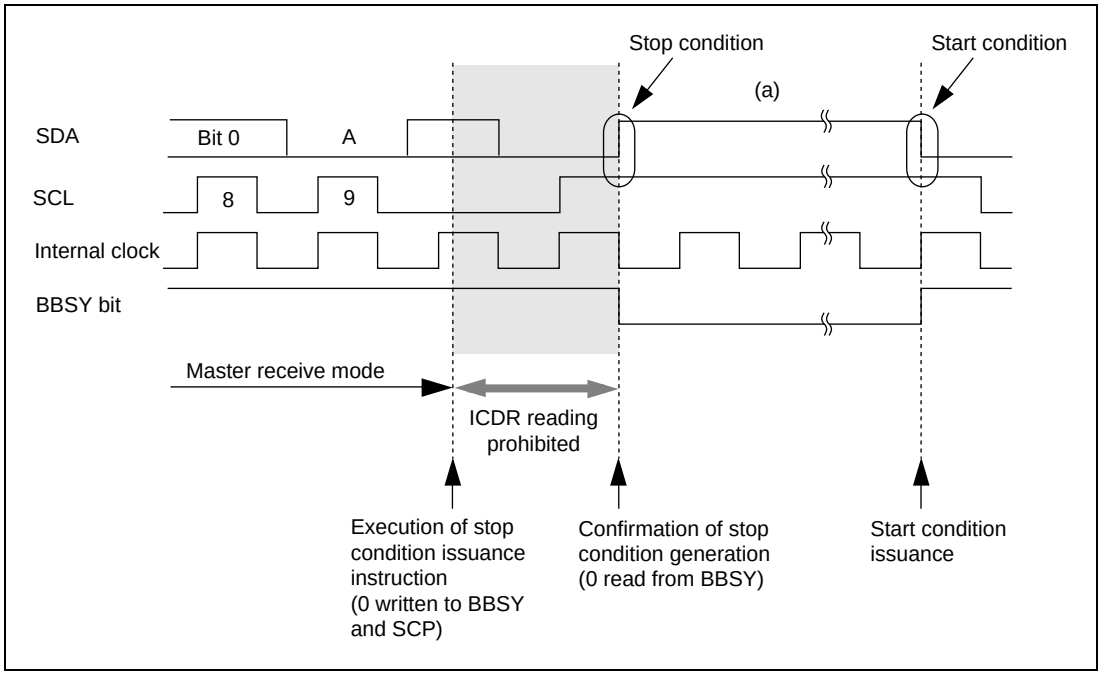


Figure 16.18 Points for Attention Concerning Reading of Master Receive Data

Section 17 A/D Converter

17.1 Overview

The H8S/2128 Series and H8S/2124 Series incorporate a 10-bit successive-approximations A/D converter that allows up to eight analog input channels to be selected.

In addition to the eight analog input channels, up to 8 channels of digital input can be selected for A/D conversion. Since the conversion precision falls to the equivalent of 6-bit resolution when digital input is selected, digital input is ideal for use by a comparator identifying multi-valued inputs, for example.

17.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Eight (analog) or 8 (digital) input channels
- Settable analog conversion voltage range
  - The analog conversion voltage range is set using the analog power supply voltage pin (AVcc) as the analog reference 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 (8-bit timer), or $\overline{\text{ADTRG}}$ pin
- A/D conversion end interrupt generation
  - An A/D conversion end interrupt (ADI) request can be generated at the end of A/D conversion

17.1.2 Block Diagram

Figure 17.1 shows a block diagram of the A/D converter.

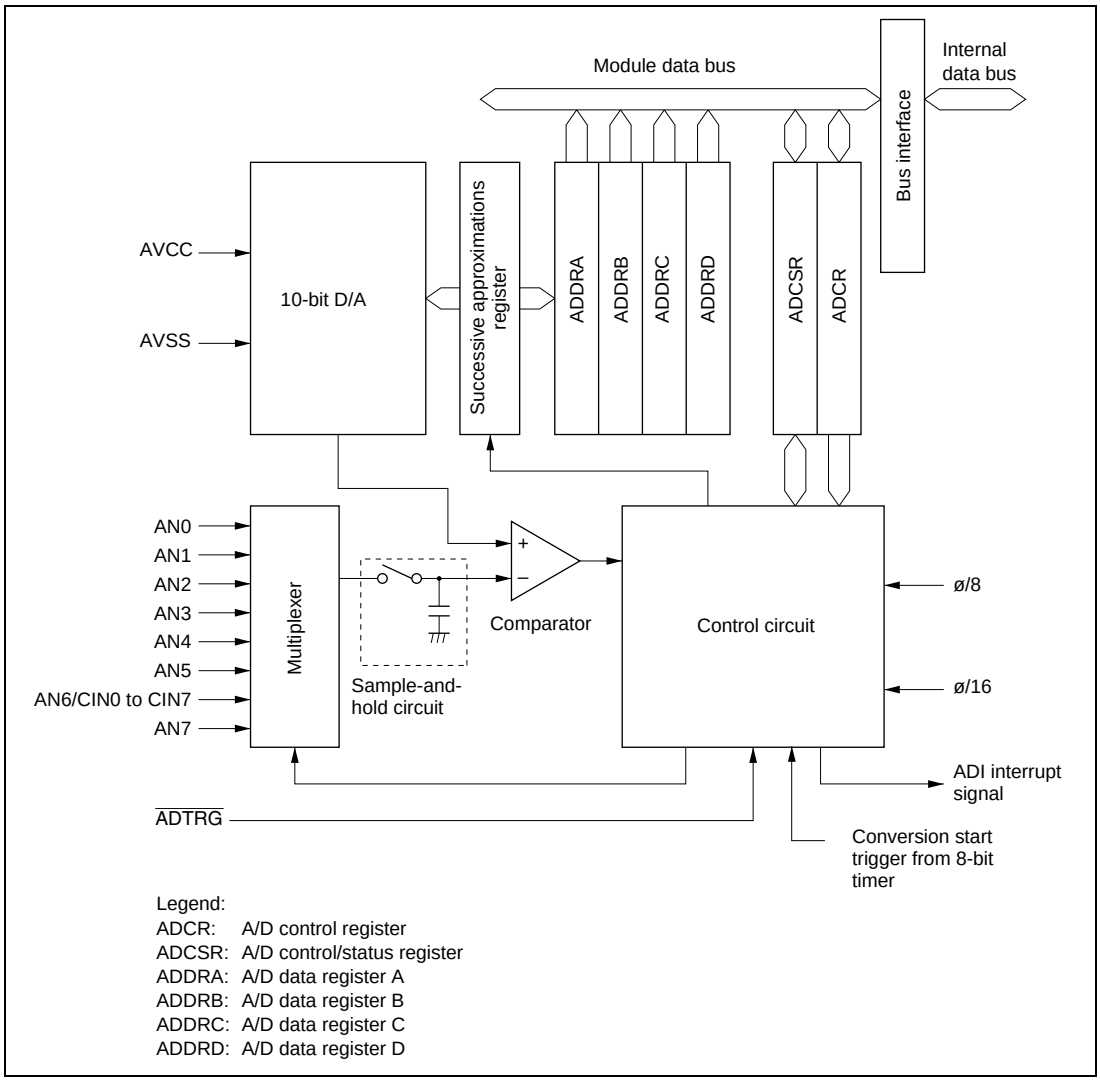


Figure 17.1 Block Diagram of A/D Converter

17.1.3 Pin Configuration

Table 17.1 summarizes the input pins used by the A/D converter.

The AVCC and AVSS pins are the power supply pins for the analog block in the A/D converter.

Table 17.1 A/D Converter Pins

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVCC	Input	Analog block power supply
Analog ground pin	AVSS	Input	Analog block ground and A/D conversion reference voltage
Analog input pin 0	AN0	Input	Analog input channel 0
Analog input pin 1	AN1	Input	Analog input channel 1
Analog input pin 2	AN2	Input	Analog input channel 2
Analog input pin 3	AN3	Input	Analog input channel 3
Analog input pin 4	AN4	Input	Analog input channel 4
Analog input pin 5	AN5	Input	Analog input channel 5
Analog input pin 6	AN6	Input	Analog input channel 6
Analog input pin 7	AN7	Input	Analog input channel 7
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion
Expansion A/D input pins 0 to 7	CIN0 to CIN7	Input	Expansion A/D conversion input (digital input pin) channels 0 to 7

17.1.4 Register Configuration

Table 17.2 summarizes the registers of the A/D converter.

Table 17.2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
A/D data register AH	ADDRAH	R	H'00	H'FFE0
A/D data register AL	ADDRAL	R	H'00	H'FFE1
A/D data register BH	ADDRBH	R	H'00	H'FFE2
A/D data register BL	ADDRBL	R	H'00	H'FFE3
A/D data register CH	ADDRCH	R	H'00	H'FFE4
A/D data register CL	ADDRCL	R	H'00	H'FFE5
A/D data register DH	ADDRDH	R	H'00	H'FFE6
A/D data register DL	ADDRDL	R	H'00	H'FFE7
A/D control/status register	ADCSR	R/(W)*[2]	H'00	H'FFE8
A/D control register	ADCR	R/W	H'3F	H'FFE9
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87
Keyboard comparator control register	KBCOMP	R/W	H'00	H'FEE4

Notes: 1. Lower 16 bits of the address.
2. Only 0 can be written in bit 7, to clear the flag.

17.2 Register Descriptions

17.2.1 A/D Data Registers A to D (ADDRA to ADDRD)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	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 the results of A/D conversion.

The 10-bit data resulting from A/D conversion is transferred to the ADDR register for the selected channel and stored there. The upper 8 bits of the converted data are transferred to the upper byte (bits 15 to 8) of ADDR, and the lower 2 bits are transferred to the lower byte (bits 7 and 6) and stored. Bits 5 to 0 are always read as 0.

The correspondence between the analog input channels and ADDR registers is shown in table 17.3.

The ADDR registers can always be read by the CPU. The upper byte can be read directly, but for the lower byte, data transfer is performed via a temporary register (TEMP). For details, see section 17.3, Interface to Bus Master.

The ADDR registers are initialized to H'0000 by a reset, and in standby mode, watch mode, subactive mode, subsleep mode, and module stop mode.

Table 17.3 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 or CIN0 to CIN7	ADDRC
AN3	AN7	ADDRD

17.2.2 A/D Control/Status Register (ADCSR)

Bit	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written in bit 7, to clear the flag.

ADCSR is an 8-bit readable/writable register that controls A/D conversion operations.

ADCSR is initialized to H'00 by a reset, and in standby mode, watch mode, subactive mode, subsleep mode, and module stop mode.

Bit 7—A/D End Flag (ADF): Status flag that indicates the end of A/D conversion.

Bit 7

ADF	Description	
0	[Clearing conditions]	(Initial value)
	• When 0 is written in 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) requests at the end of A/D conversion.

Bit 6

ADIE	Description	
0	A/D conversion end interrupt (ADI) request is disabled	(Initial value)
1	A/D conversion end interrupt (ADI) request is enabled	

Bit 5—A/D Start (ADST): Selects starting or stopping of A/D conversion. Holds a value of 1 during A/D conversion.

The ADST bit can be set to 1 by software, a timer conversion start trigger, or the A/D external trigger input pin ($\overline{\text{ADTRG}}$).

Bit 5

ADST	Description	
0	A/D conversion stopped	(Initial value)
1	Single mode: A/D conversion is started. Cleared to 0 automatically when conversion on the specified channel ends	
	Scan mode: A/D conversion is started. Conversion continues sequentially on the selected channels until ADST is cleared to 0 by software, a reset, or a transition to standby mode or module stop mode	

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode as the A/D conversion operating mode. See section 17.4, Operation, for single mode and scan mode operation. Only set the SCAN bit while conversion is stopped.

Bit 4

SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

Bit 3—Clock Select (CKS): Sets the A/D conversion time. Only change the conversion time while ADST = 0.

Bit 3

CKS	Description	
0	Conversion time = 266 states (max.)	(Initial value)
1	Conversion time = 134 states (max.)	

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): Together with the SCAN bit, these bits select the analog input channel(s).

One analog input channel can be switched to digital input.

Only set the input channel while conversion is stopped.

Group Selection	Channel Selection		Description	
CH2	**CH1**	**CH0**	**Single Mode**	**Scan Mode**
0	0	0	AN0 (Initial value)	AN0
		1	AN1	AN0, AN1
	1	0	AN2	AN0 to AN2
		1	AN3	AN0 to AN3
1	0	0	AN4	AN4
		1	AN5	AN4, AN5
	1	0	AN6 or CIN0 to CIN7	AN4, AN5, AN6 or CIN0 to CIN7
		1	AN7	AN4, AN5, AN6 or CIN0 to CIN7 AN7

17.2.3 A/D Control Register (ADCR)

Bit	7	6	5	4	3	2	1	0
	TRGS1	TRGS0	—	—	—	—	—	—
Initial value	0	0	1	1	1	1	1	1
Read/Write	R/W	R/W	—	—	—	—	—	—

ADCR is an 8-bit readable/writable register that enables or disables external triggering of A/D conversion operations.

ADCR is initialized to H'3F by a reset, and in standby mode, watch mode, subactive mode, subsleep mode, and module stop mode.

Bits 7 and 6—Timer Trigger Select 1 and 0 (TRGS1, TRGS0): These bits select enabling or disabling of the start of A/D conversion by a trigger signal. Only set bits TRGS1 and TRGS0 while conversion is stopped.

Bit 7	Bit 6		
TRGS1	**TRGS0**	**Description**	
0	0	Start of A/D conversion by external trigger is disabled	(Initial value)
	1	Start of A/D conversion by external trigger is disabled	
1	0	Start of A/D conversion by external trigger (8-bit timer) is enabled	
	1	Start of A/D conversion by external trigger pin is enabled	

Bits 5 to 0—Reserved: These bits cannot be modified and are always read as 1.

17.2.4 Keyboard Comparator Control Register (KBCOMP)

Bit	7	6	5	4	3	2	1	0
	IrE	IrCKS2	IrCKS1	IrCKS0	KBADE	KBCH2	KBCH1	KBCH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

KBCOMP is an 8-bit readable/writable register that selects the CIN input channels for A/D conversion.

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

Bits 7 to 4—Reserved

Bit 3—Keyboard A/D Enable: Selects either analog input pin (AN6) or digital input pin (CIN0 to CIN7) for A/D converter channel 6 input. If digital input pins are selected, input on A/D converter channel 7 will not be converted correctly.

Bits 2 to 0—Keyboard A/D Channel Select 2 to 0 (KBCH2 to KBCH0): These bits select the channels for A/D conversion from among the digital input pins. Only set the input channel while A/D conversion is stopped.

Bit 3	Bit 2	Bit 1	Bit 0	A/D Converter	A/D Converter
KBADE	KBCH2	KBCH1	KBCH0	Channel 6 Input	Channel 7 Input
0	—	—	—	AN6	AN7
1	0	0	0	CIN0	Undefined
			1	CIN1	Undefined
		1	0	CIN2	Undefined
			1	CIN3	Undefined
	1	0	0	CIN4	Undefined
			1	CIN5	Undefined
		1	0	CIN6	Undefined
			1	CIN7	Undefined

17.2.5 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	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	R/W	R/W

MSTPCR, comprising two 8-bit readable/writable registers, performs module stop mode control.

When the MSTP9 bit in MSTPCR is set to 1, A/D converter operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 1—Module Stop (MSTP9): Specifies the A/D converter module stop mode.

MSTPCRH Bit 1

MSTP9	Description	
0	A/D converter module stop mode is cleared	
1	A/D converter module stop mode is set	(Initial value)

17.3 Interface to Bus Master

ADDRA to ADDRD are 16-bit registers, but the data bus to the bus master is only 8 bits wide. Therefore, in accesses by the bus master, the upper byte is accessed directly, but the lower byte is accessed via a temporary register (TEMP).

A data read from ADDR is performed as follows. When the upper byte is read, the upper byte value is transferred to the CPU and the lower byte value is transferred to TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading ADDR, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 17.2 shows the data flow for ADDR access.

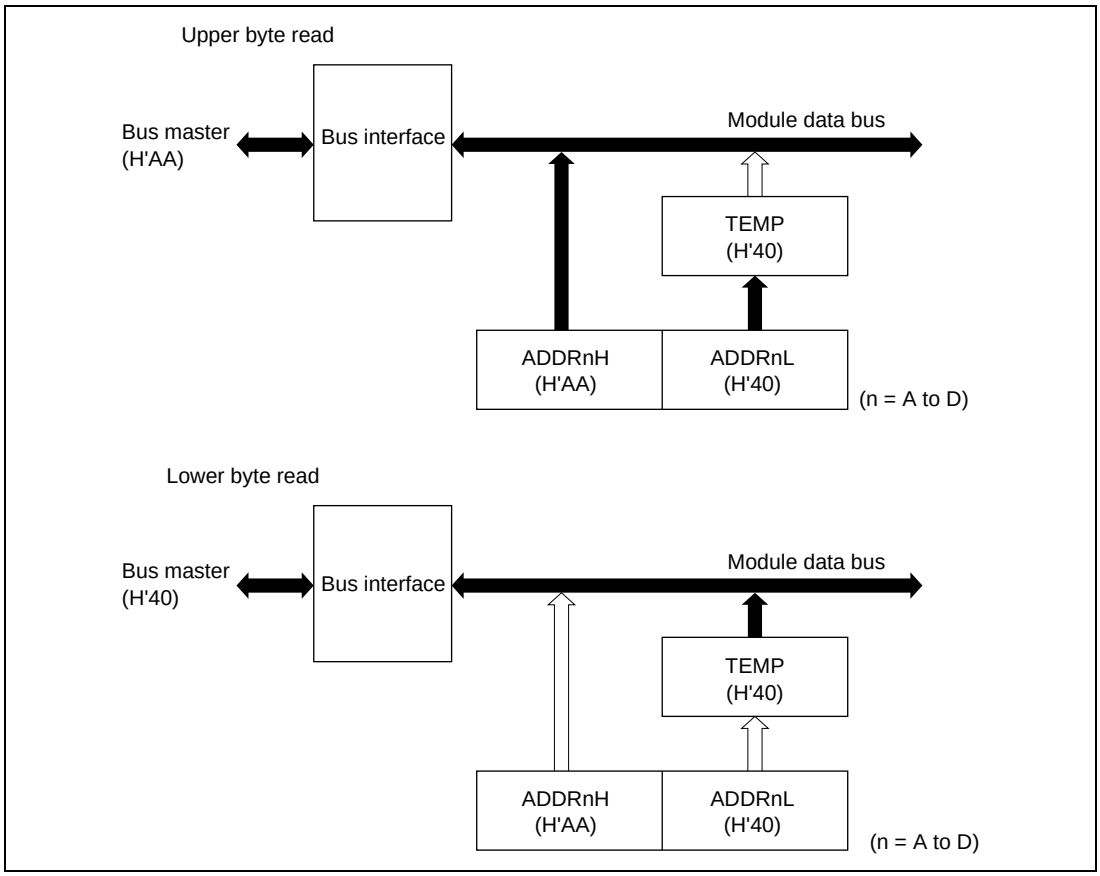


Figure 17.2 ADDR Access Operation (Reading H'AA40)

17.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

17.4.1 Single Mode (SCAN = 0)

Single mode is selected when A/D conversion is to be performed on a single channel only. A/D conversion is started when the ADST bit is set to 1 by software, or by external trigger input. The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends.

On completion of conversion, the ADF flag is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The ADF flag is cleared by writing 0 after reading ADCSR.

When the operating mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST 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 next. Figure 17.3 shows a timing diagram for this example.

1. Single mode is selected (SCAN = 0), input channel AN1 is selected (CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred to ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 to the ADF flag.
6. The routine reads and processes the conversion result (ADDRB).
7. Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is set to 1, A/D conversion starts again and steps 2 to 7 are repeated.

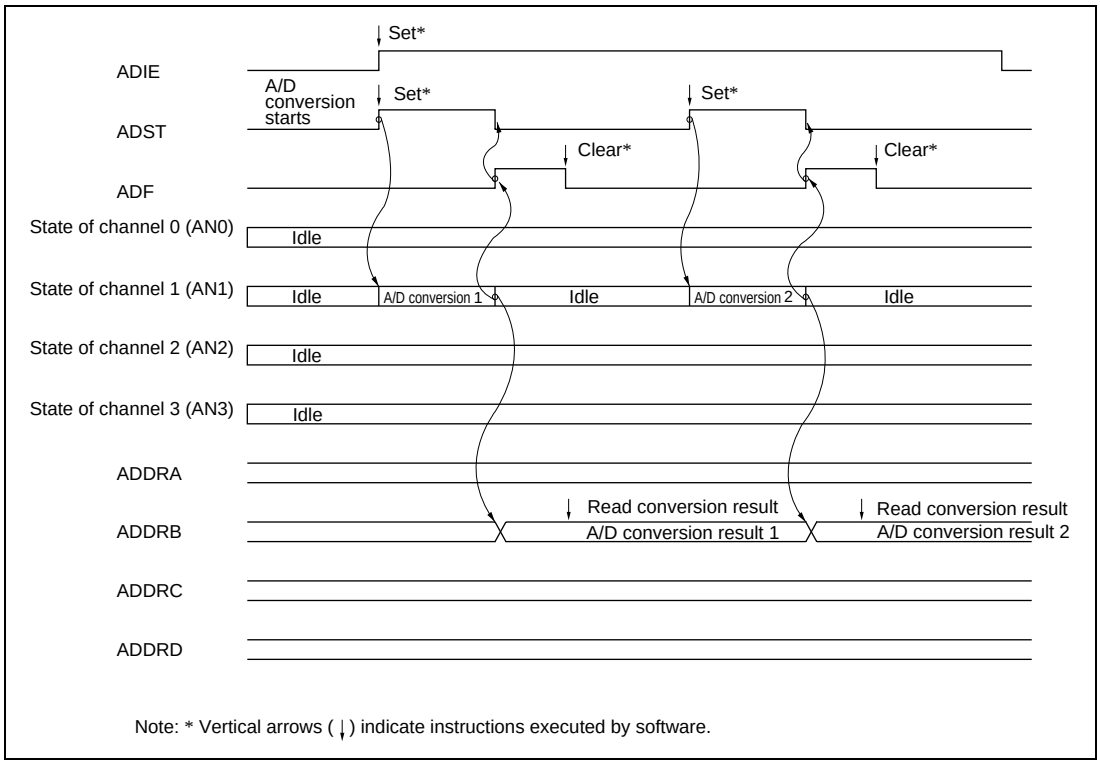


Figure 17.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

17.4.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by software, or by timer or external trigger input, A/D conversion starts on the first channel in the group (AN0 when CH2 = 0; AN4 when CH2 = 1). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1 or AN5) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the ADDR registers corresponding to the channels.

When the operating mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST 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 described next. Figure 17.4 shows a timing diagram for this example.

1. Scan mode is selected (SCAN = 1), scan group 0 is selected (CH2 = 0), analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1)
2. When A/D conversion of the first channel (AN0) is completed, the result is transferred to 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 ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1 at this 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 ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).

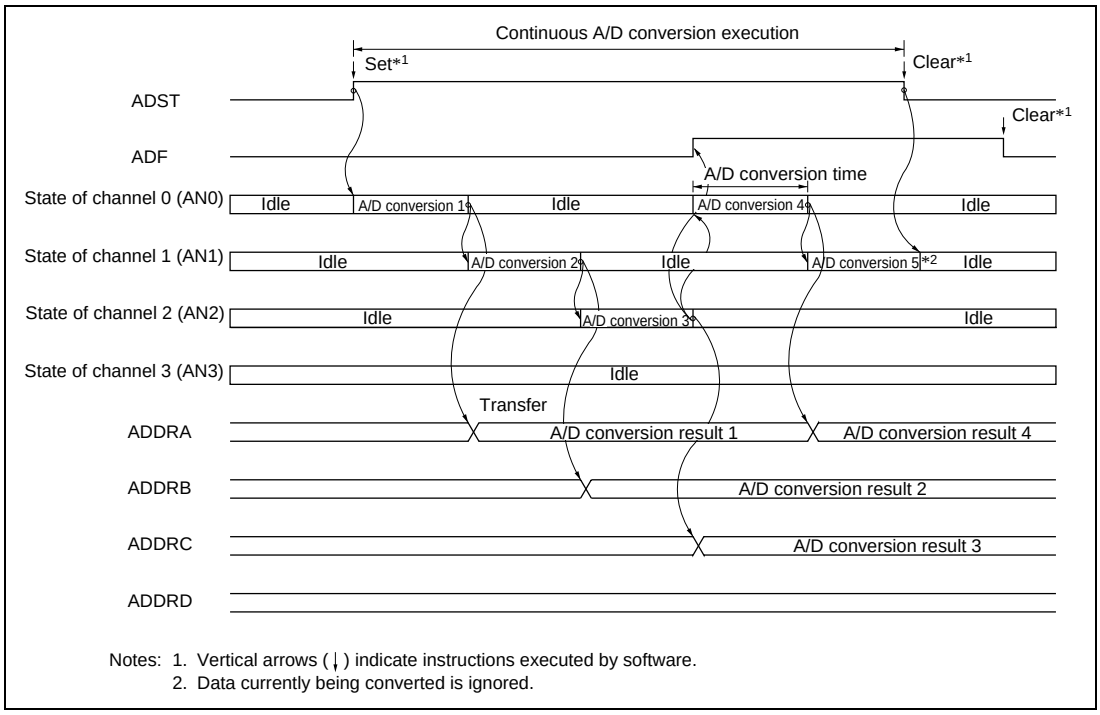


Figure 17.4 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

17.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit is set to 1, then starts conversion. Figure 17.5 shows the A/D conversion timing. Table 17.4 indicates the A/D conversion time.

As indicated in figure 17.5, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 17.4.

In scan mode, the values given in table 17.4 apply to the first conversion time. In the second and subsequent conversions the conversion time is fixed at 256 states when CKS = 0 or 128 states when CKS = 1.

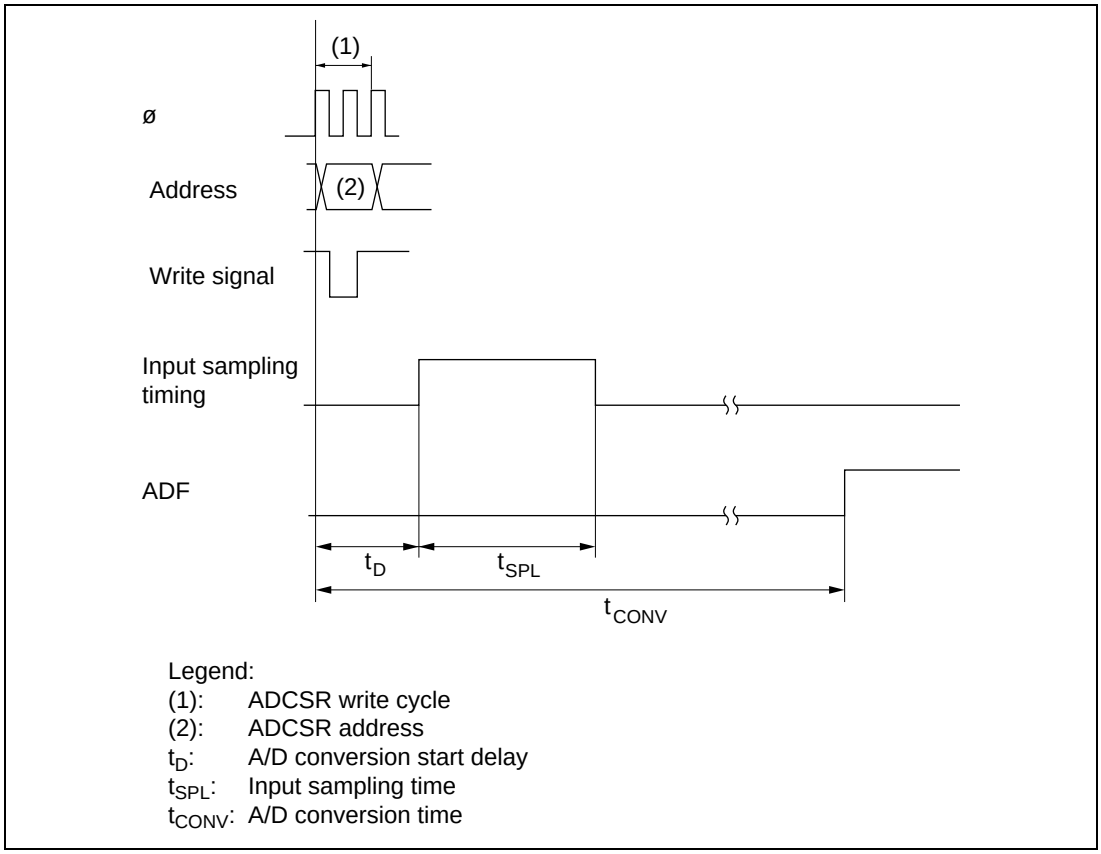


Figure 17.5 A/D Conversion Timing

Table 17.4 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1		
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay	t_D	10	—	17	6	—	9
Input sampling time	t_{SPL}	—	63	—	—	31	—
A/D conversion time	t_{CONV}	259	—	266	131	—	134

Note: Values in the table are the number of states.

17.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to 11 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the ADST bit is set to 1 by software. Figure 17.6 shows the timing.

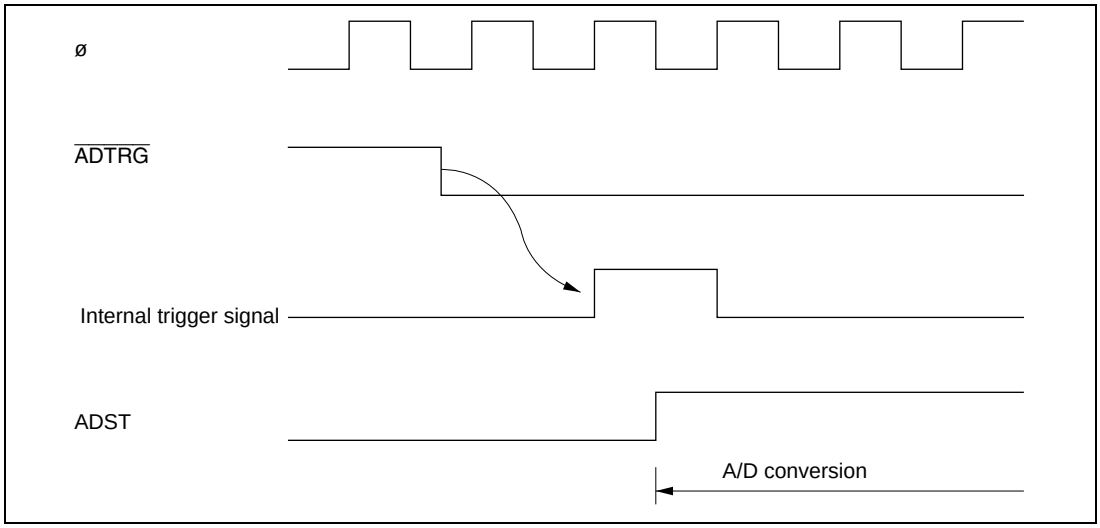


Figure 17.6 External Trigger Input Timing

17.5 Interrupts

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

17.6 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 the ANn analog input pins during A/D conversion should be in the range $AV_{SS} \leq ANn \leq AV_{CC}$ (n = 0 to 7).
2. Digital input voltage range
   The voltage applied to the CINn digital input pins should be in the range $AV_{SS} \leq CINn \leq AV_{CC}$ and $V_{SS} \leq CINn \leq V_{CC}$ (n = 0 to 7).
3. 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 $AV_{SS} = V_{SS}$. If the A/D converter is not used, the AVCC and AVSS pins must on no account be left open.

If conditions 1 to 3 above are not met, the reliability of the device may be adversely affected.

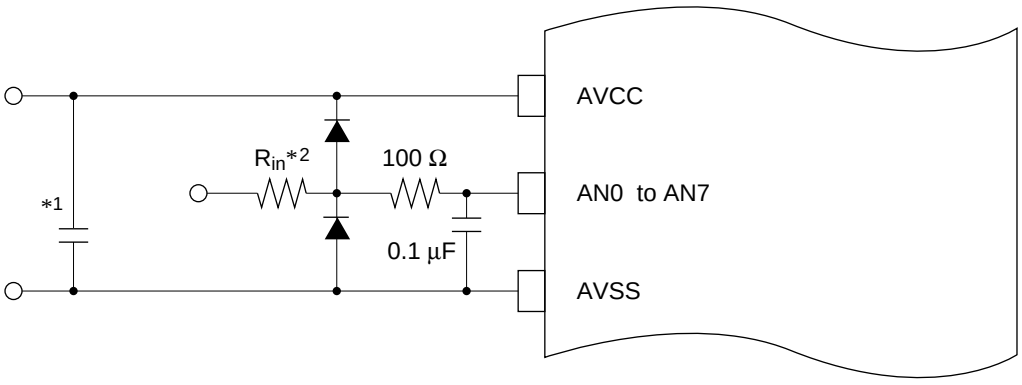
Notes on Board Design: In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN7), and analog power supply (AVCC) by the analog ground (AVSS). Also, the analog ground (AVSS) should be connected at one point to a stable digital ground (VSS) on the board.

Notes on Noise Countermeasures: A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN7) should be connected between AVCC and AVSS as shown in figure 17.7.

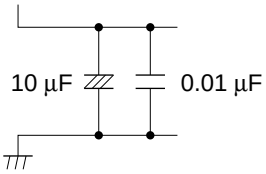
Also, the bypass capacitors connected to AVCC and the filter capacitor connected to AN0 to AN7 must be connected to AVSS.

If a filter capacitor is connected as shown in figure 17.7, the input currents at the analog input pins (AN0 to AN7) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.



Notes: Figures are reference values.

1.



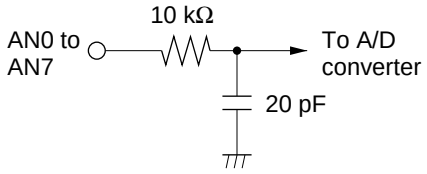
2. R_{in}: Input impedance

Figure 17.7 Example of Analog Input Protection Circuit

Table 17.5 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	10*	kΩ

Note: * When V_{cc} = 4.0 V to 5.5 V and ø ≤ 12 MHz



Note: Figures are reference values.

Figure 17.8 Analog Input Pin Equivalent Circuit

A/D Conversion Precision Definitions: H8S/2128 Series and H8S/2124 Series A/D conversion precision definitions 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 characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 17.10).
- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3FE) to B'111111111 (H'3FF) (see figure 17.10).
- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 17.9).
- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error.
- Absolute precision

  The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

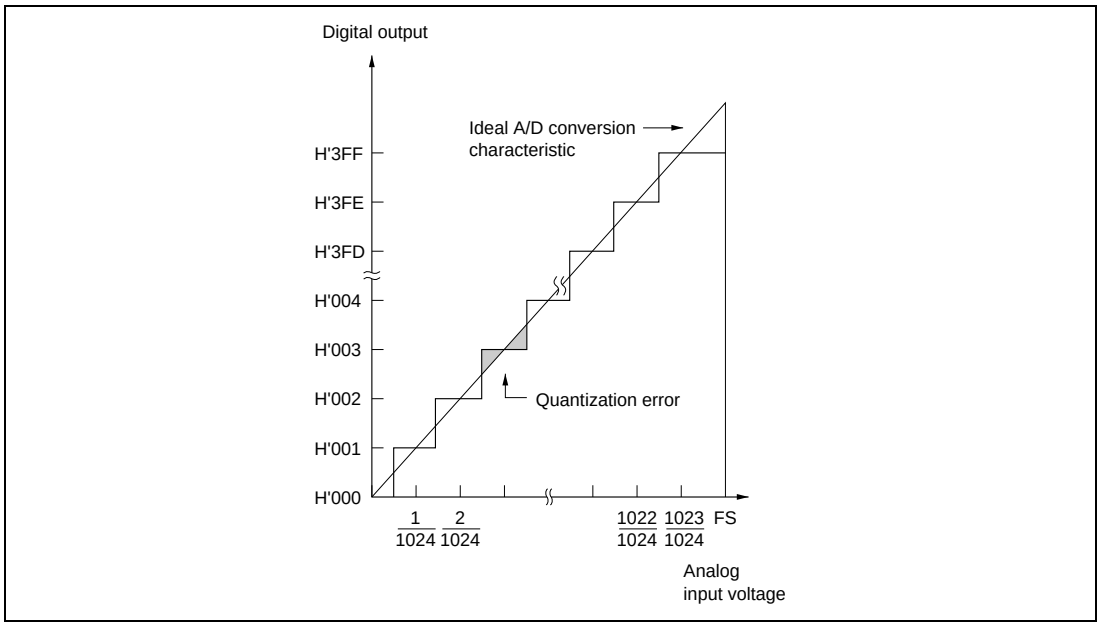


Figure 17.9 A/D Conversion Precision Definitions (1)

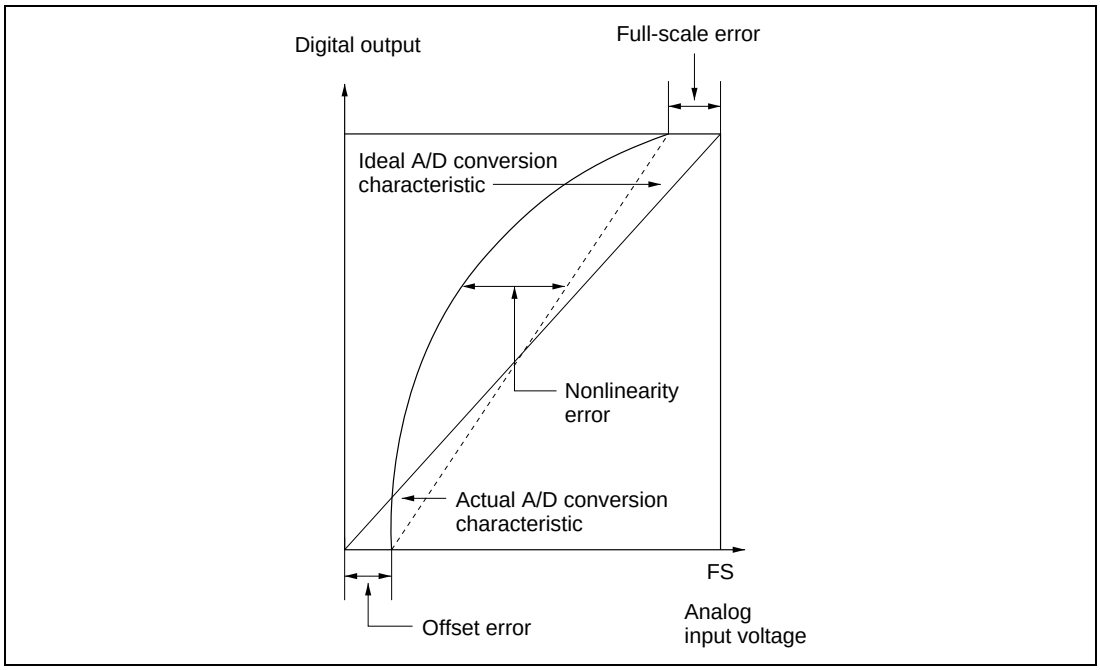


Figure 17.10 A/D Conversion Precision Definitions (2)

Permissible Signal Source Impedance: H8S/2128 Series and H8S/2124 Series analog input is designed so that conversion precision is guaranteed for an input signal for which the signal source impedance is 10 kΩ (Vcc = 4.0 to 5.5 V, when ø ≤ 12 MHz or CKS = 0) or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 10 kΩ (Vcc = 4.0 to 5.5 V, when ø ≤ 12 MHz or CKS = 0), charging may be insufficient and it may not be possible to guarantee the A/D conversion precision.

However, if a large capacitance is provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored.

But since a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µsec or greater).

When converting a high-speed analog signal, a low-impedance buffer should be inserted.

Influences on Absolute Precision: Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AV_{SS}.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, so acting as antennas.

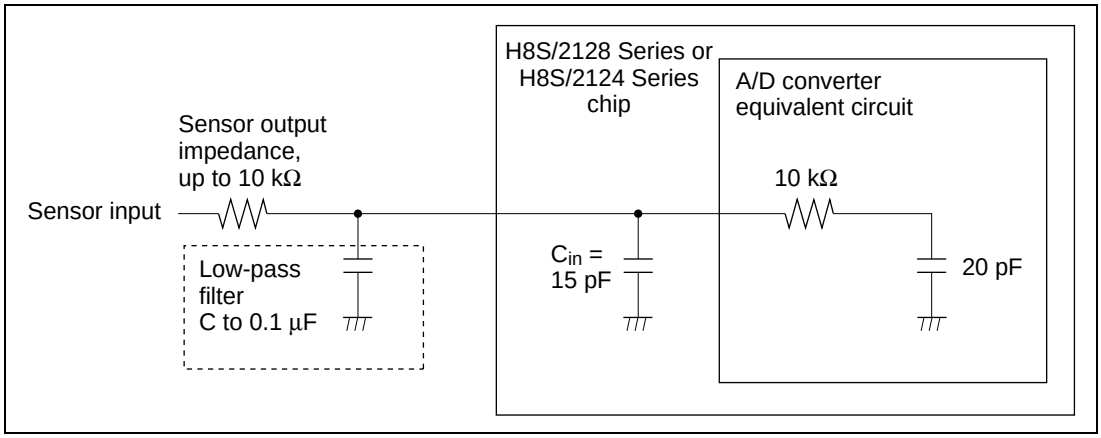


Figure 17.11 Example of Analog Input Circuit

Section 18 RAM

18.1 Overview

The H8S/2128, H8S/2124 and H8S/2123 have 4 kbytes of on-chip high-speed static RAM, and the H8S/2127, H8S/2126, H8S/2122 and H8S/2120 have 2 kbytes. The on-chip RAM is connected to the bus master by a 16-bit data bus, enabling both byte data and word data to be accessed in one state. This makes it possible to perform fast word data transfer.

The on-chip RAM can be enabled or disabled by means of the RAM enable bit (RAME) in the system control register (SYSCR).

18.1.1 Block Diagram

Figure 18.1 shows a block diagram of the on-chip RAM.

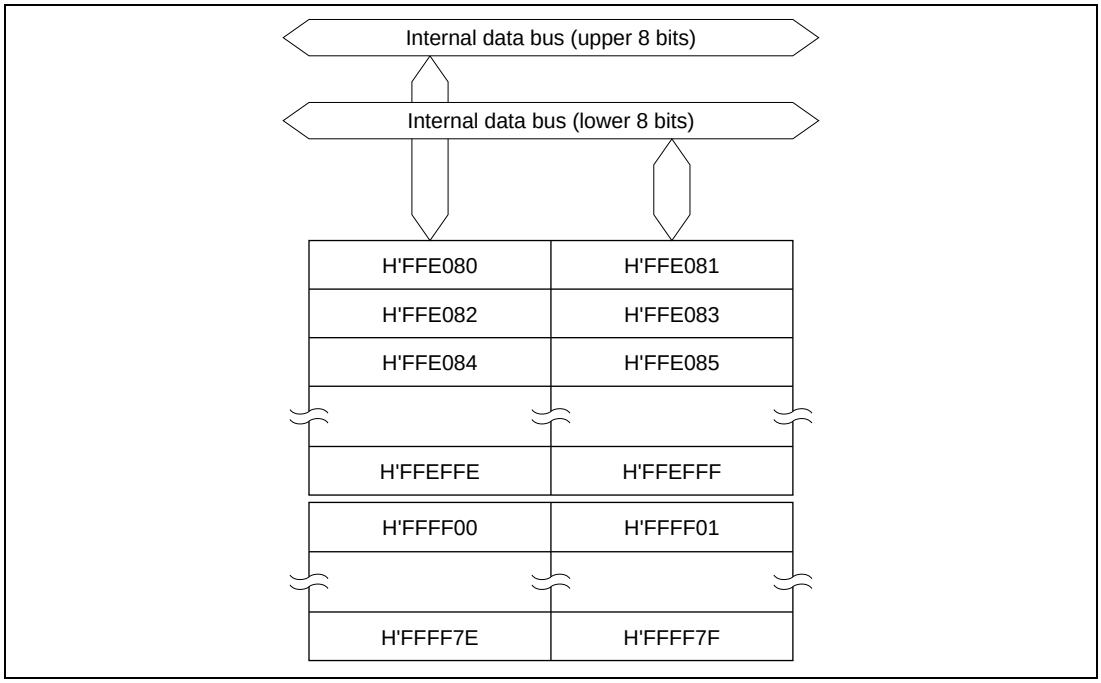


Figure 18.1 Block Diagram of RAM (H8S/2128, H8S/2124, H8S/2123)

18.1.2 Register Configuration

The on-chip RAM is controlled by SYSCR. Table 18.1 shows the register configuration.

Table 18.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address*
System control register	SYSCR	R/W	H'09	H'FFC4

Note: * Lower 16 bits of the address.

18.2 System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
	CS2E	IOSE	INTM1	INTM0	XRST	NMIEG	HIE	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R	R/W	R	R/W	R/W	R/W

The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. For details of other bits in SYSCR, see section 3.2.2, System Control Register (SYSCR).

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

Bit 0

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

18.3 Operation

18.3.1 Expanded Mode (Modes 1, 2, and 3 (EXPE = 1))

When the RAME bit is set to 1, accesses to H8S/2128, H8S/2124, and H8S/2123 addresses H'(FF)E080 to H'(FF)EFFF and H'(FF)FF00 to H'(FF)FF7F, and H8S/2127, H8S/2126, H8S/2122, and H8S/2120 addresses H'(FF)E880 to H'(FF)EFFF and H'(FF)FF00 to H'(FF)FF7F, are directed to the on-chip RAM. When the RAME bit is cleared to 0, accesses to addresses H'(FF)E080 to H'(FF)EFFF and H'(FF)FF00 to H'(FF)FF7F, are directed to the off-chip address space.

Since the on-chip RAM is connected to the bus master by a 16-bit data bus, it can be written to and read in byte or word units. Each type of access is performed in one state.

Even addresses use the upper 8 bits, and odd addresses use the lower 8 bits. Word data must start at an even address.

18.3.2 Single-Chip Mode (Modes 2 and 3 (EXPE = 0))

When the RAME bit is set to 1, accesses to H8S/2128, H8S/2124, and H8S/2123 addresses H'(FF)E080 to H'(FF)EFFF and H'(FF)FF00 to H'(FF)FF7F, and H8S/2127, H8S/2126, H8S/2122, and H8S/2120 addresses H'(FF)E880 to H'(FF)EFFF and H'(FF)FF00 to H'(FF)FF7F, are directed to the on-chip RAM. When the RAME bit is cleared to 0, the on-chip RAM is not accessed. A read will always return H'FF, and writes are invalid.

Since the on-chip RAM is connected to the bus master by a 16-bit data bus, it can be written to and read in byte or word units. Each type of access is performed in one state.

Even addresses use the upper 8 bits, and odd addresses use the lower 8 bits. Word data must start at an even address.

Section 19 ROM

19.1 Overview

The H8S/2128 and H8S/2124 have 128 kbytes of on-chip ROM (flash memory or mask ROM), the H8S/2123 has 96 kbytes, the H8S/2127 and H8S/2122 have 64 kbytes and the H8S/2126 and H8S/2120 have 32 kbytes. The ROM is connected to the bus master by a 16-bit data bus. The CPU accesses both byte and word data in one state, enabling faster instruction fetches and higher processing speed.

The mode pins (MD1 and MD0) and the EXPE bit in MDCR can be set to enable or disable the on-chip ROM.

The flash memory versions of the H8S/2128 can be erased and programmed on-board as well as with a general-purpose PROM programmer.

19.1.1 Block Diagram

Figure 19.1 shows a block diagram of the ROM.

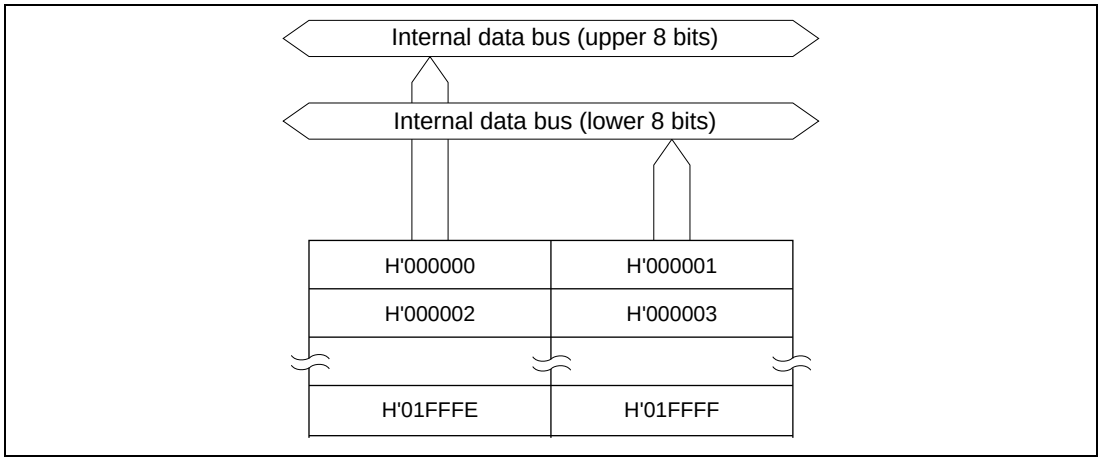


Figure 19.1 ROM Block Diagram (H8S/2128, H8S/2124)

19.1.2 Register Configuration

The H8S/2128 Series and H8S/2124 Series on-chip ROM is controlled by the operating mode and register MDCR. The register configuration is shown in table 19.1.

Table 19.1 ROM Register

Register Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R/W	Undefined Depends on the operating mode	H'FFC5

Note: * Lower 16 bits of the address.

19.2 Register Descriptions

19.2.1 Mode Control Register (MDCR)

Bit	7	6	5	4	3	2	1	0
	EXPE	—	—	—	—	—	MDS1	MDS0
Initial value	—*	0	0	0	0	0	—*	—*
Read/Write	R/W*	—	—	—	—	—	R	R

Note: * Determined by the MD1 and MD0 pins.

MDCR is an read-only 8-bit register used to set the H8S/2128 Series or H8S/2124 Series operating mode and monitor the current operating mode.

The EXPE bit is initialized in accordance with the mode pin states by a reset and in hardware standby mode.

Bit 7—Expanded Mode Enable (EXPE): Sets expanded mode. In mode 1, EXPE is fixed at 1 and cannot be modified. In modes 2 and 3, EXPE has an initial value of 0 and can be read or written.

Bit 7 EXPE	Description
0	Single-chip mode selected
1	Expanded mode selected

Bits 6 to 2—Reserved: These bits cannot be modified and are always read as 0.

Bits 1 and 0—Mode Select 1 and 0 (MDS1, MDS0): These bits indicate values that reflects the input levels of mode pins MD1 and MD0 (the current operating mode). Bits MDS1 and MDS0 correspond to pins MD1 and MD0, respectively. These are read-only bits, and cannot be modified. When MDCR is read, the input levels of mode pins MD1 and MD0 are latched in these bits.

19.3 Operation

The on-chip ROM is connected to the CPU by a 16-bit data bus, and both byte and word data is accessed in one state. Even addresses are connected to the upper 8 bits, and odd addresses to the lower 8 bits. Word data must start at an even address.

The mode pins (MD1 and MD0) and the EXPE bit in MDCR can be set to enable or disable the on-chip ROM, as shown in table 19.2.

In normal mode, the maximum amount of ROM that can be used is 56 kbytes.

Table 19.2 Operating Modes and ROM

Operating Mode			Mode Pins		MDCR	
MCU Operating Mode	**CPU Operating Mode**	**Description**	**MD1**	**MD0**	**EXPE**	**On-Chip ROM**
Mode 1	Normal	Expanded mode with on-chip ROM disabled	0	1	1	Disabled
Mode 2	Advanced	Single-chip mode	1	0	0	Enabled*
	Advanced	Expanded mode with on-chip ROM enabled			1	
Mode 3	Normal	Single-chip mode		1	0	Enabled (max. 56 kbytes)
	Normal	Expanded mode with on-chip ROM enabled			1	

Note: * 128 kbytes in the H8S/2128 and H8S/2124, 96 kbytes in the H8S/2123, 64 kbytes in the H8S/2127 and H8S/2122 and 32 kbytes in the H8S/2126 and H8S/2120.

19.4 Overview of Flash Memory

19.4.1 Features

The features of the flash memory are summarized below.

- Four flash memory operating modes
  - — Program mode
  - — Erase mode
  - — Program-verify mode
  - — Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 32 bytes at a time. Erasing is performed by block erase (in single-block units). When erasing multiple blocks, the individual blocks must be erased sequentially. Block erasing can be performed as required on 1-kbyte, 8-kbyte, 16-kbyte, 28-kbyte, and 32-kbyte.
- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 32-byte programming, 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

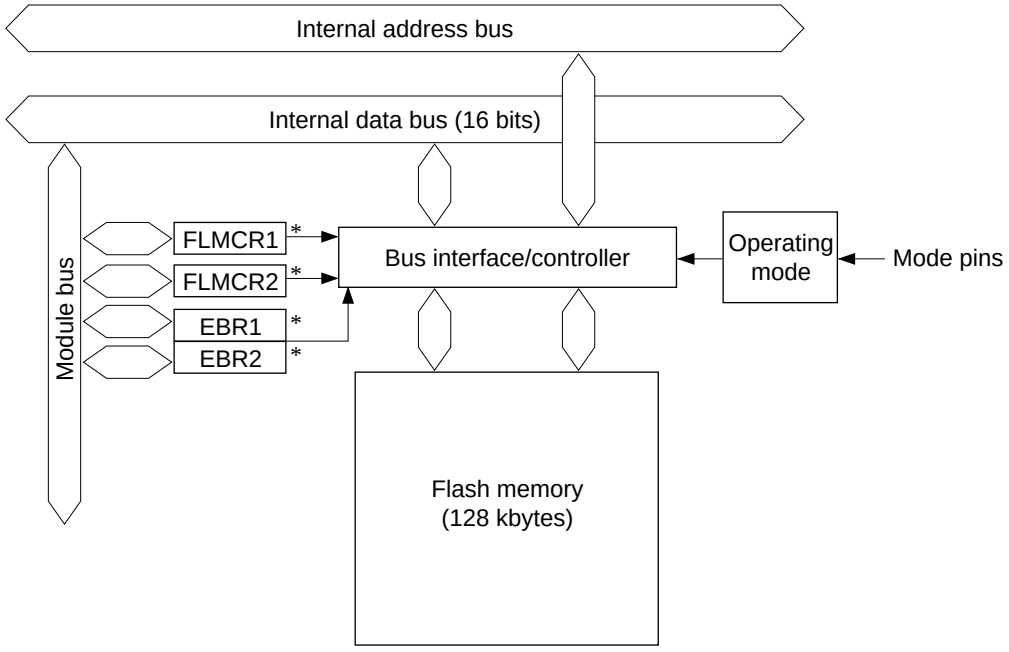
  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - — Boot mode
  - — User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the bit rate of the H8S/2128 Series chip can be automatically adjusted to match the transfer bit rate of the host.
- Protect modes

  There are three protect modes, hardware, software, and error protect, which allow protected status to be designated for flash memory program/erase/verify operations.
- Writer mode

  Flash memory can be programmed/erased in writer mode, using a PROM programmer, as well as in on-board programming mode.

19.4.2 Block Diagram



Legend:
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2

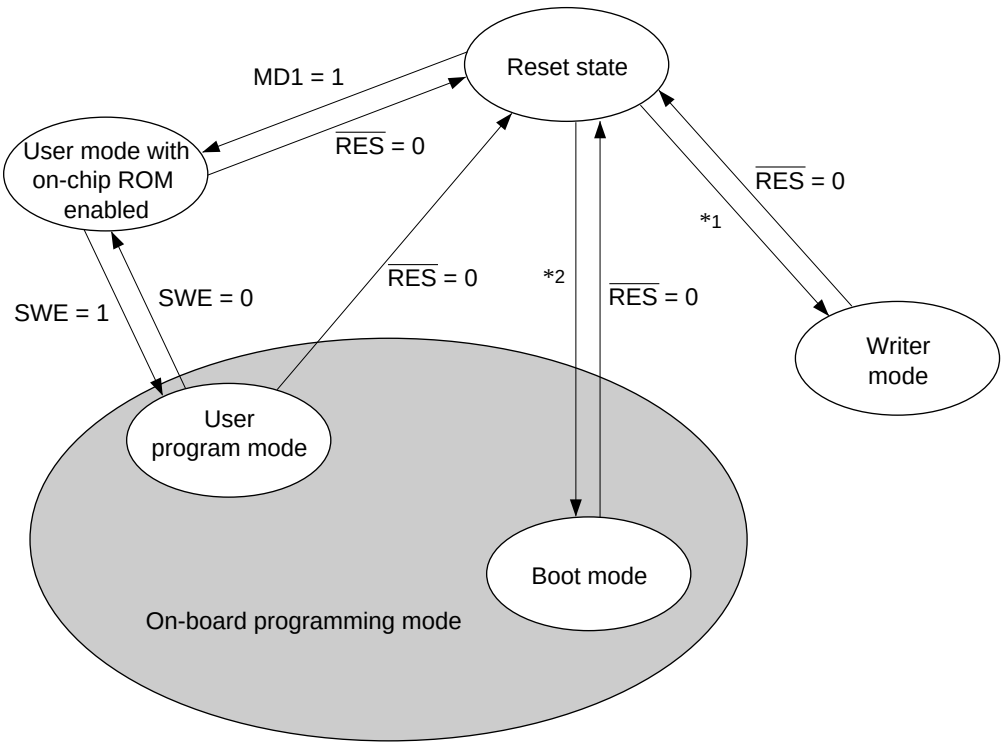
Note: * These registers are used only in the flash memory version. In the mask ROM version, a read at any of these addresses will return an undefined value, and writes are invalid.

Figure 19.2 Block Diagram of Flash Memory

19.4.3 Flash Memory Operating Modes

Mode Transitions: When the mode pins are set in the reset state and a reset-start is executed, the MCU enters one of the operating modes shown in figure 19.3. In user mode, flash memory can be read but not programmed or erased.

Flash memory can be programmed and erased in boot mode, user program mode, and Writer mode.



Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

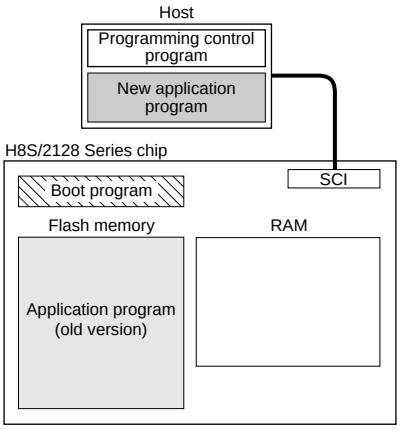
1. MD0 = MD1 = 0, P42 = 0, P41 = P40 = 1
2. MD1 = MD0 = 0, P42 = P41 = P40 = 1

Figure 19.3 Flash Memory Mode Transitions

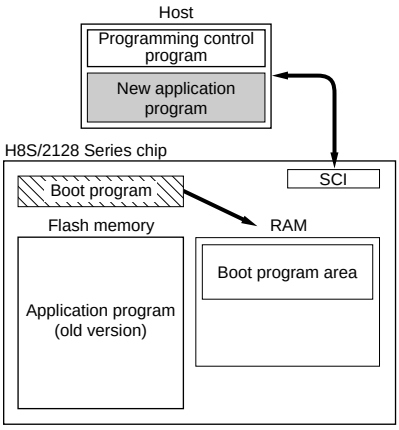
On-Board Programming Modes

- Boot mode

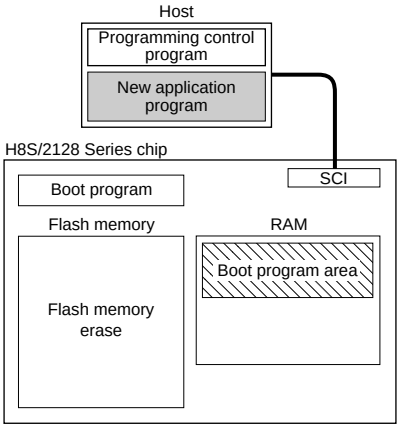
1. Initial state
   The flash memory is in the erased state when the device is shipped. The description here applies to the case where the old program version or data is being rewritten. The user should prepare the programming control program and new application program beforehand in the host.



2. SCI communication check
   When boot mode is entered, the boot program in the H8S/2128 Series chip (originally incorporated in the chip) is started, an SCI communication check is carried out, and the boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, entire flash memory erasure is performed, without regard to blocks.



4. Writing new application program
   The programming control program transferred from the host to RAM by SCI communication is executed, and the new application program in the host is written into the flash memory.

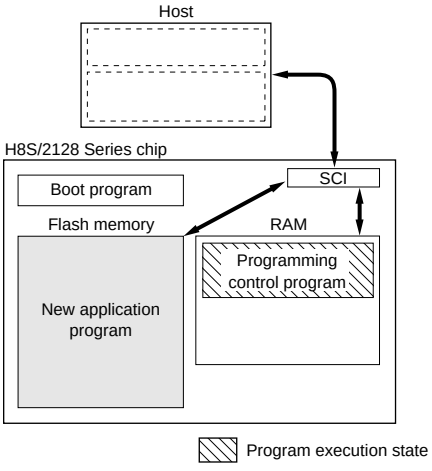
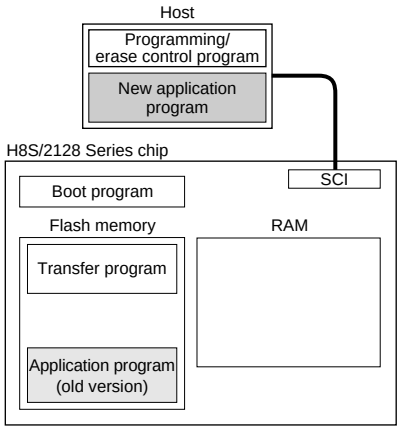


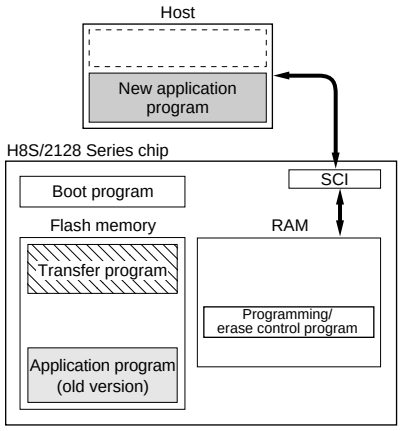
Figure 19.4 Boot Mode

- User program mode

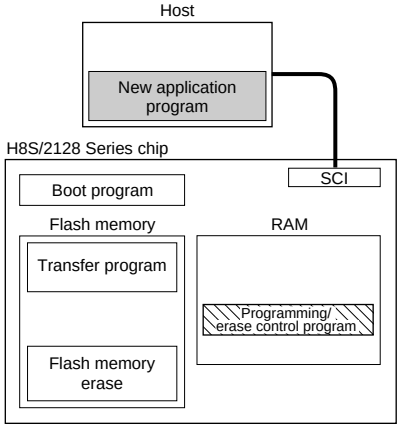
1. Initial state
   (1) The program that will transfer the programming/ erase control program to on-chip RAM should be written into the flash memory by the user beforehand.
   (2) The programming/erase control program should be prepared in the host or in the flash memory.



2. Programming/erase control program transfer executes the transfer program in the flash memory, and transfers the programming/erase control program to RAM.



3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.

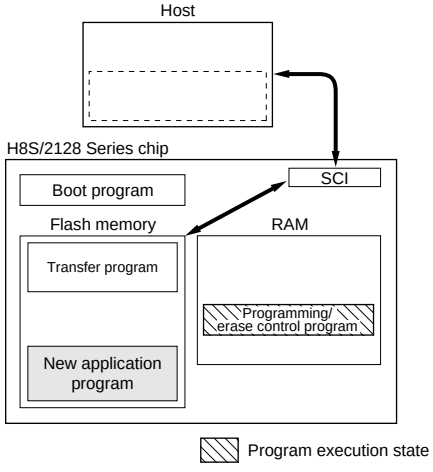


Figure 19.5 User Program Mode (Example)

Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Entire memory erase	Yes	Yes
Block erase	No	Yes
Programming control program*	Program/program-verify	Erase/erase-verify Program/program-verify

Note: * To be provided by the user, in accordance with the recommended algorithm.

Block Configuration: The flash memory is divided into two 32-kbyte blocks, two 8-kbyte blocks, one 16-kbyte block, one 28-kbyte block, and four 1-kbyte blocks.

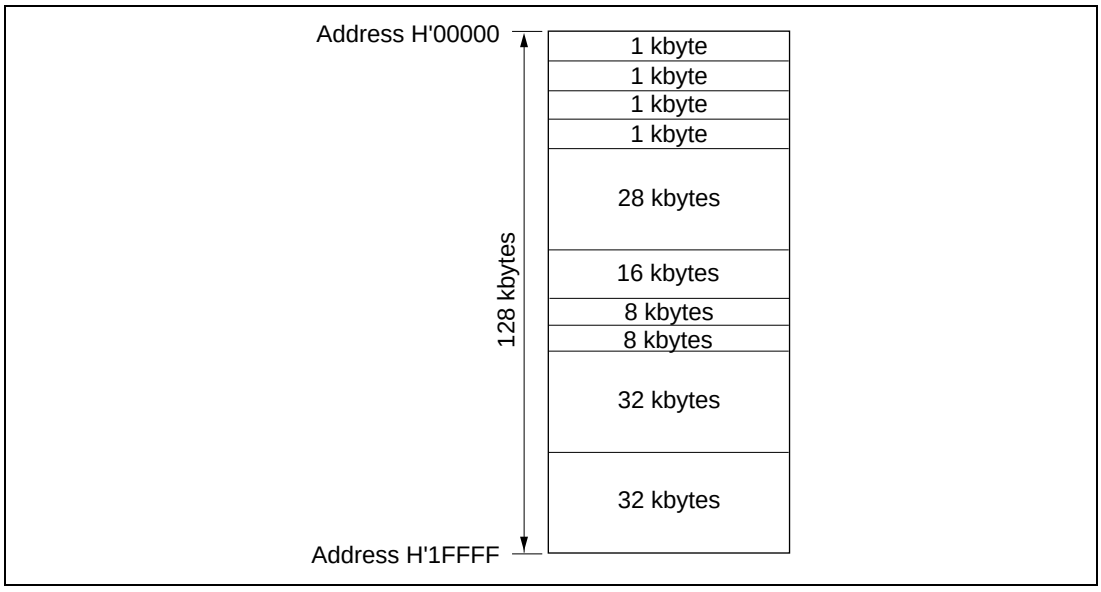


Figure 19.6 Flash Memory Block Configuration

19.4.4 Pin Configuration

The flash memory is controlled by means of the pins shown in table 19.3.

Table 19.3 Flash Memory Pins

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Mode 1	MD1	Input	Sets MCU operating mode
Mode 0	MD0	Input	Sets MCU operating mode
Port 42	P42	Input	Sets MCU operating mode when MD1 = MD0 = 0
Port 41	P41	Input	Sets MCU operating mode when MD1 = MD0 = 0
Port 40	P40	Input	Sets MCU operating mode when MD1 = MD0 = 0
Transmit data	TxD0	Output	Serial transmit data output
Receive data	RxD0	Input	Serial receive data input

19.4.5 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 19.4. In order for these registers to be accessed, the FLSHE bit must be set to 1 in STCR.

Table 19.4 Flash Memory Registers

Register Name	Abbreviation	R/W	Initial Value	Address*[1]
Flash memory control register 1	FLMCR1*[5]	R/W*[3]	H'80	H'FF80*[2]
Flash memory control register 2	FLMCR2*[5]	R/W*[3]	H'00*[4]	H'FF81*[2]
Erase block register 1	EBR1*[5]	R/W*[3]	H'00*[4]	H'FF82*[2]
Erase block register 2	EBR2*[5]	R/W*[3]	H'00*[4]	H'FF83*[2]
Serial/timer control register	STCR	R/W	H'00	H'FFC3

Notes: 1. Lower 16 bits of the address.
2. Flash memory registers share addresses with other registers. Register selection is performed by the FLSHE bit in the serial/timer control register (STCR).
3. In modes in which the on-chip flash memory is disabled, a read will return H'00, and writes are invalid.
4. When the SWE bit in FLMCR1 is not set, these registers are initialized to H'00.
5. FLMCR1, FLMCR2, EBR1, and EBR2 are 8-bit registers. Only byte accesses are valid for these registers, the access requiring 2 states. These registers are used only in the flash memory version. In the mask ROM version, a read at any of these addresses will return an undefined value, and writes are invalid.

19.5 Register Descriptions

19.5.1 Flash Memory Control Register 1 (FLMCR1)

Bit	7	6	5	4	3	2	1	0
	FWE	SWE	—	—	EV	PV	E	P
Initial value	1	0	0	0	0	0	0	0
Read/Write	R	R/W	—	—	R/W	R/W	R/W	R/W

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode is entered by setting SWE to 1 and setting the corresponding bit. Program mode is entered by setting SWE to 1, then setting the PSU bit in FLMCR2, and finally setting the P bit. Erase mode is entered by setting SWE to 1, then setting the ESU bit in FLMCR2, and finally setting the E bit. FLMCR1 is initialized to H'80 by a reset, and in hardware standby mode, software standby mode, subactive mode, subsleep mode, and watch mode. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writes to the EV and PV bits in FLMCR1 are enabled only when SWE = 1; writes to the E bit only when SWE = 1, and ESU = 1; and writes to the P bit only when SWE = 1, and PSU = 1.

Bit 7—Flash Write Enable Bit (FWE): Controls programming and erasing of on-chip flash memory. This bit cannot be modified and is always read as 1.

Bit 6—Software Write Enable Bit (SWE): Enables or disables flash memory programming. SWE should be set before setting bits ESU, PSU, EV, PV, E, P, and EB9 to EB0, and should not be cleared at the same time as these bits.

Bit 6

SWE	Description	
0	Writes disabled	(Initial value)
1	Writes enabled	

Bit 5 and 4—Reserved: These bits cannot be modified and are always read as 0.

Bit 3—Erase-Verify (EV): Selects erase-verify mode transition or clearing. Do not set the SWE, ESU, PSU, PV, E, or P bit at the same time.

Bit 3

EV	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When SWE = 1	

Bit 2—Program-Verify (PV): Selects program-verify mode transition or clearing. Do not set the SWE, ESU, PSU, EV, E, or P bit at the same time.

Bit 2

PV	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When SWE = 1	

Bit 1—Erase (E): Selects erase mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or P bit at the same time.

Bit 1

E	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When SWE = 1, and ESU = 1	

Bit 0—Program (P): Selects program mode transition or clearing. Do not set the SWE, PSU, ESU, EV, PV, or E bit at the same time.

Bit 0

P	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When SWE = 1, and PSU = 1	

19.5.2 Flash Memory Control Register 2 (FLMCR2)

Bit	7	6	5	4	3	2	1	0
	FLER	—	—	—	—	—	ESU	PSU
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	—	—	—	—	—	R/W	R/W

FLMCR2 is an 8-bit register that monitors the presence or absence of flash memory program/erase protection (error protection) and performs setup for flash memory program/erase mode. FLMCR2 is initialized to H'00 by a reset, and in hardware standby mode. The ESU and PSU bits are cleared to 0 in software standby mode, subactive mode, subsleep mode, and watch mode.

When on-chip flash memory is disabled, a read will return H'00 and writes are invalid.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7

FLER	Description	
0	Flash memory is operating normally	(Initial value)
	Flash memory program/erase protection (error protection) is disabled	
	[Clearing condition]	
	Reset, 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 19.8.3, Error Protection	

Bits 6 to 2—Reserved: These bits cannot be modified and are always read as 0.

Bit 1—Erase Setup (ESU): Prepares for a transition to erase mode. Set this bit to 1 before setting the E bit to 1 in FLMCR1. Do not set the SWE, PSU, EV, PV, E, or P bit at the same time.

Bit 1		
ESU	**Description**	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When SWE = 1	

Bit 0—Program Setup (PSU): Prepares for a transition to program mode. Set this bit to 1 before setting the P bit to 1 in FLMCR1. Do not set the SWE, ESU, EV, PV, E, or P bit at the same time.

Bit 0		
PSU	**Description**	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When SWE = 1	

19.5.3 Erase Block Registers 1 and 2 (EBR1, EBR2)

Bit	7	6	5	4	3	2	1	0
EBR1	—	—	—	—	—	—	EB9	EB8
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	—	R/W*	R/W*

Bit	7	6	5	4	3	2	1	0
EBR2	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * In normal mode, these bits cannot be modified and are always read as 0.

EBR1 and EBR2 are registers that specify the flash memory erase area block by block; bits 1 and 0 in EBR1 and bits 7 to 0 in EBR2 are readable/writable bits. EBR1 and EBR2 are each initialized to H'00 by a reset, in hardware standby mode, software standby mode, subactive mode, subsleep mode, and watch mode, when the SWE bit in FLMCR1 is not set. When a bit in EBR1 or EBR2 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Set only one bit in EBR1 or EBR2 (more than one bit cannot be set). When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 19.5.

Table 19.5 Flash Memory Erase Blocks

Block (Size) 128-kbyte Versions	Address
EB0 (1 kB)	H'(00)0000 to H'(00)03FF
EB1 (1 kB)	H'(00)0400 to H'(00)07FF
EB2 (1 kB)	H'(00)0800 to H'(00)0BFF
EB3 (1 kB)	H'(00)0C00 to H'(00)0FFF
EB4 (28 kB)	H'(00)1000 to H'(00)7FFF
EB5 (16 kB)	H'(00)8000 to H'(00)BFFF
EB6 (8 kB)	H'(00)C000 to H'(00)DFFF
EB7 (8 kB)	H'00E000 to H'00FFFF
EB8 (32 kB)	H'010000 to H'017FFF
EB9 (32 kB)	H'018000 to H'01FFFF

19.5.4 Serial/Timer Control Register (STCR)

Bit	7	6	5	4	3	2	1	0
	IICS	IICX1	IICX0	IICE	FLSHE	—	ICKS1	ICKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

STCR is an 8-bit readable/writable register that controls register access, the IIC operating mode (when the on-chip IIC option is included), and on-chip flash memory (in F-ZTAT versions), and also selects the TCNT input clock. For details on functions not related to on-chip flash memory, see section 3.2.4, Serial/Timer Control Register (STCR), and descriptions of individual modules. If a module controlled by STCR is not used, do not write 1 to the corresponding bit.

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

Bits 7 to 4—I^2C Control (IICS, IICX1, IICX0, IICE): These bits control the operation of the I^2C bus interface. For details, see section 16, I^2C Bus Interface.

Bit 3—Flash Memory Control Register Enable (FLSHE): Setting the FLSHE bit to 1 enables read/write access to the flash memory control registers. If FLSHE is cleared to 0, the flash memory control registers are deselected. In this case, the flash memory control register contents are retained.

Bit 3

FLSHE	Description	
0	Flash memory control registers deselected	(Initial value)
1	Flash memory control registers selected	

Bit 2—Reserved: Do not write 1 to this bit.

Bits 1 and 0—Internal Clock Source Select 1 and 0 (ICKS1, ICKS0): These bits control 8-bit timer operation. See section 12, 8-Bit Timers, for details.

19.6 On-Board Programming Modes

When pins are set to on-board programming mode, a transition is made to in which program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 19.6. For a diagram of the transitions to the various flash memory modes, see figure 19.3.

Only advanced mode setting is possible for boot mode.

In the case of user program mode, user program mode is established in advanced mode or normal mode, depending on the setting of the MD0 pin. In normal mode, only programming of a 56-kbyte area of flash memory is possible.

Table 19.6 Setting On-Board Programming Modes

Mode		Pin				
Mode Name	**CPU Operating Mode**	**MD1**	**MD0**	**P42**	**P41**	**P40**
Boot mode	Advanced mode	0	0	1*	1*	1*
User program mode	Advanced mode	1	0	—	—	—
	Normal mode		1	—	—	—

Note: * Can be used as I/O ports after boot mode is initiated.

19.6.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The channel 0 SCI to be used is set to asynchronous mode.

When a reset-start is executed after the H8S/2128 Series MCU's pins have been set to boot mode, the boot program built into the MCU is started and the programming control program prepared in the host is serially transmitted to the MCU via the SCI. In the MCU, the user program received via the SCI is written into the user program area in on-chip RAM. After the transfer is completed, control branches to the start address of the user program area and the user program execution state is entered (flash memory programming is performed).

The transferred user program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 19.7, and the boot program mode execution procedure in figure 19.8.

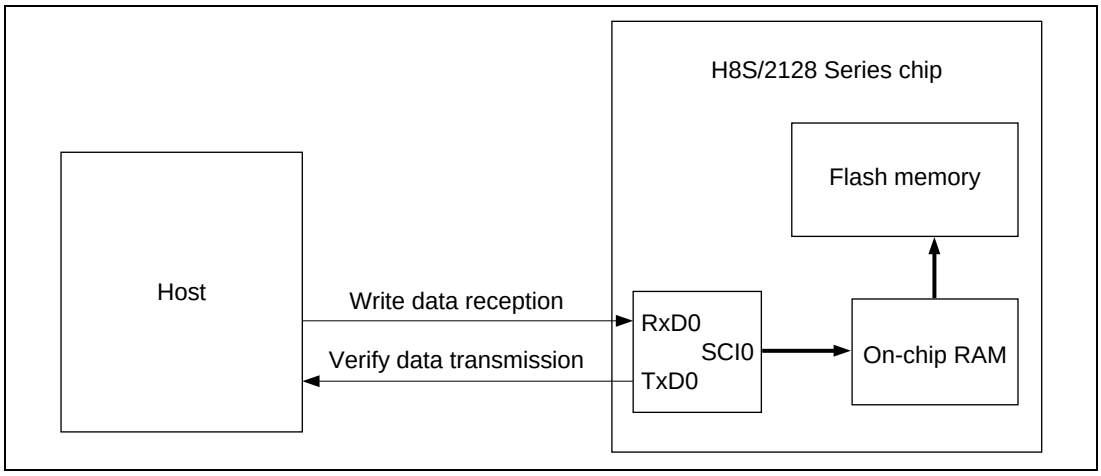
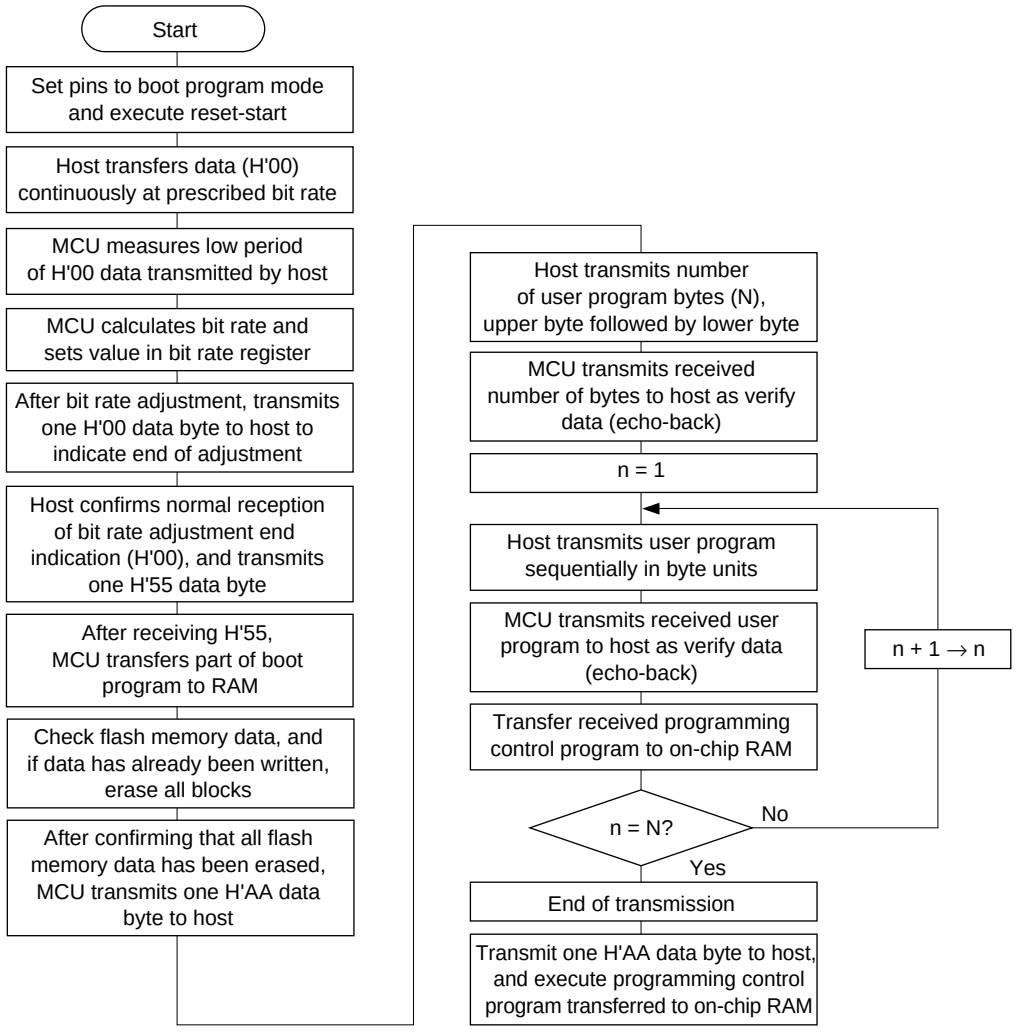


Figure 19.7 System Configuration in Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operations are halted.

Figure 19.8 Boot Mode Execution Procedure

Automatic SCI Bit Rate Adjustment

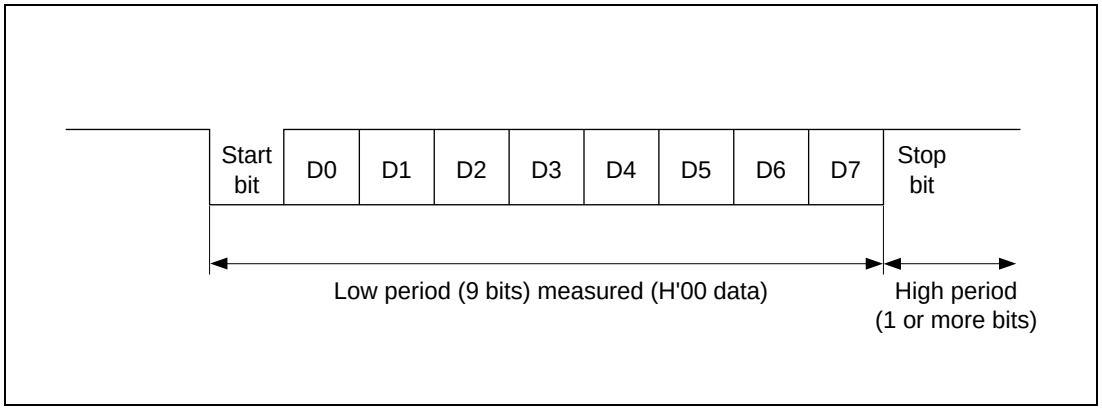


Figure 19.9 RxD0 Input Signal when Using Automatic SCI Bit Rate Adjustment

When boot mode is initiated, the H8S/2128 Series MCU measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The MCU calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the MCU. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the MCU's system clock frequency, there will be a discrepancy between the bit rates of the host and the MCU. To ensure correct SCI operation, the host's transfer bit rate should be set to (2400, 4800, or 9600) bps.

Table 19.7 shows typical host transfer bit rates and system clock frequencies for which automatic adjustment of the MCU's bit rate is possible. The boot program should be executed within this system clock range.

Table 19.7 System Clock Frequencies for which Automatic Adjustment of H8S/2128 Series Bit Rate is Possible

Host Bit Rate	System Clock Frequency for which Automatic Adjustment of H8S/2128 Series Bit Rate is Possible
9600 bps	8 MHz to 20 MHz
4800 bps	4 MHz to 20 MHz
2400 bps	2 MHz to 18 MHz

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the 128-byte area from H'(FF)FF00 to H'(FF)FF7F is reserved for use by the boot program, as shown in figure 19.10. The area to which the programming control program is transferred is the 3968-byte area from H'(FF)E080 to H'(FF)EFFF. The boot program area can be used when the programming control program transferred into RAM enters the execution state. A stack area should be set up as required.

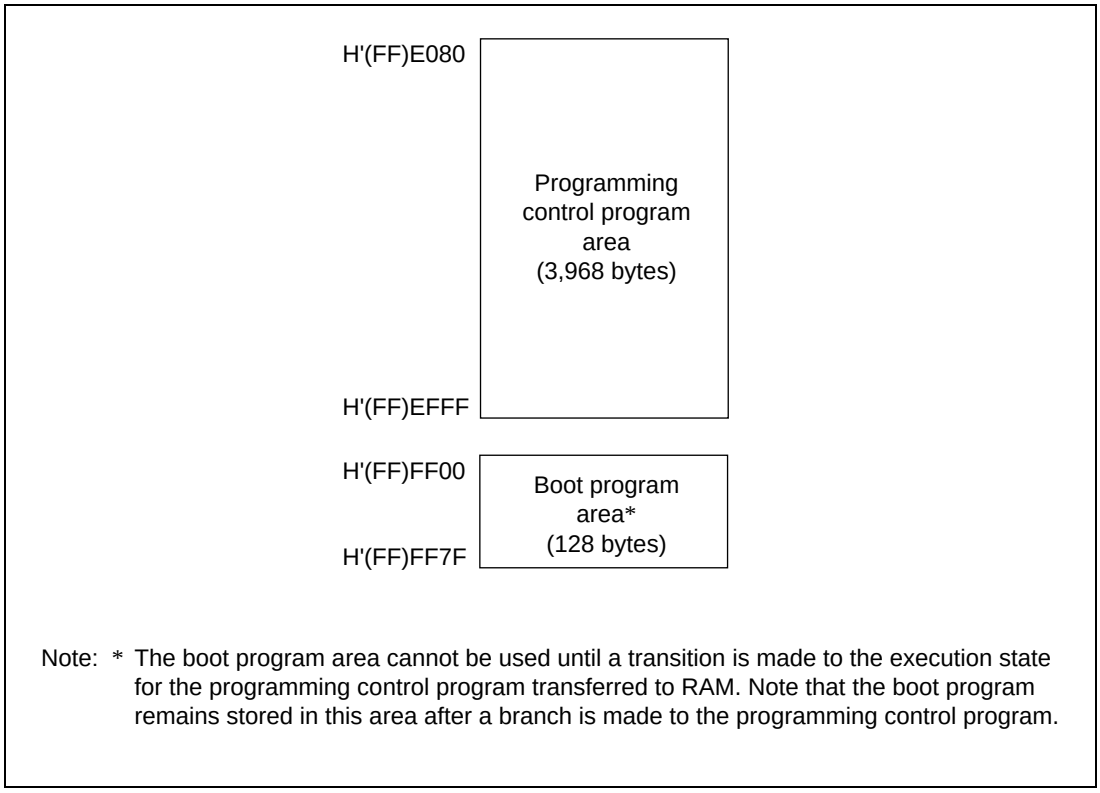


Figure 19.10 RAM Areas in Boot Mode

Notes on Use of Boot Mode:

- When the chip comes out of reset in boot mode, it measures the low period of the input at the SCI's RxD0 pin. The reset should end with RxD0 high. After the reset ends, it takes about 100 states for the chip to get ready to measure the low period of the RxD0 input.
- In boot mode, if any data has been programmed into the flash memory (if all data is not 1), all flash memory blocks are erased. Boot mode is for use when user program mode is unavailable, such as the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.
- Interrupts cannot be used while the flash memory is being programmed or erased.
- The RxD0 and TxD0 lines should be pulled up on the board.
- Before branching to the programming control program (RAM area address H'(FF)E080), the chip terminates transmit and receive operations by the on-chip SCI (channel 0) (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate remains set in BRR. The transmit data output pin, TxD0, goes to the high-level output state (P50DDR = 1, P50DR = 1).

  The contents of the CPU's internal general registers are undefined at this time, so these registers must be initialized immediately after branching to the programming control program. In particular, since the stack pointer (SP) is used implicitly in subroutine calls, etc., a stack area must be specified for use by the programming control program.

  The initial values of other on-chip registers are not changed.
- Boot mode can be entered by making the pin settings shown in table 19.6 and executing a reset-start.

  When the chip detects the boot mode setting at reset release*[1], P42, P41, and P40 can be used as I/O ports.

  Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the mode pin and executing reset release*[1]. Boot mode can also be cleared by a WDT overflow reset.

  The mode pin input levels must not be changed in boot mode.
- If the mode pin input levels are changed (for example, from low to high) during a reset, the state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{WR}$) will change according to the change in the microcomputer's operating mode*[2].

  Therefore, care must be taken to make pin settings to prevent these pins from becoming output signal pins during a reset, or to prevent collision with signals outside the microcomputer.

Notes: 1. Mode pin input must satisfy the mode programming setup time (t_{MDS} = 4 states) with respect to the reset release timing.

2. Ports with multiplexed address functions will output a low level as the address signal if a state in which the mode pin setting is for mode 1 is entered during a reset. In other modes, the port pins go to the high-impedance state. The bus control output signals will output a high level if a state in which the mode pin setting is for mode 1 is entered during a reset. In other modes, the port pins go to the high-impedance state.

19.6.2 User Program Mode

When set to user program mode, the chip can program and erase its flash memory by executing a user program/erase control program. Therefore, on-board reprogramming of the on-chip flash memory can be carried out by providing on-board supply of programming data, and storing a program/erase control program in part of the program area as necessary.

To select user program mode, select a mode that enables the on-chip flash memory (mode 2 or 3). In this mode, on-chip supporting modules other than flash memory operate as they normally would in mode 2 and 3.

The flash memory itself cannot be read while the SWE bit is set to 1 to perform programming or erasing, so the control program that performs programming and erasing should be run in on-chip RAM or external memory.

Figure 19.11 shows the procedure for executing the program/erase control program when transferred to on-chip RAM.

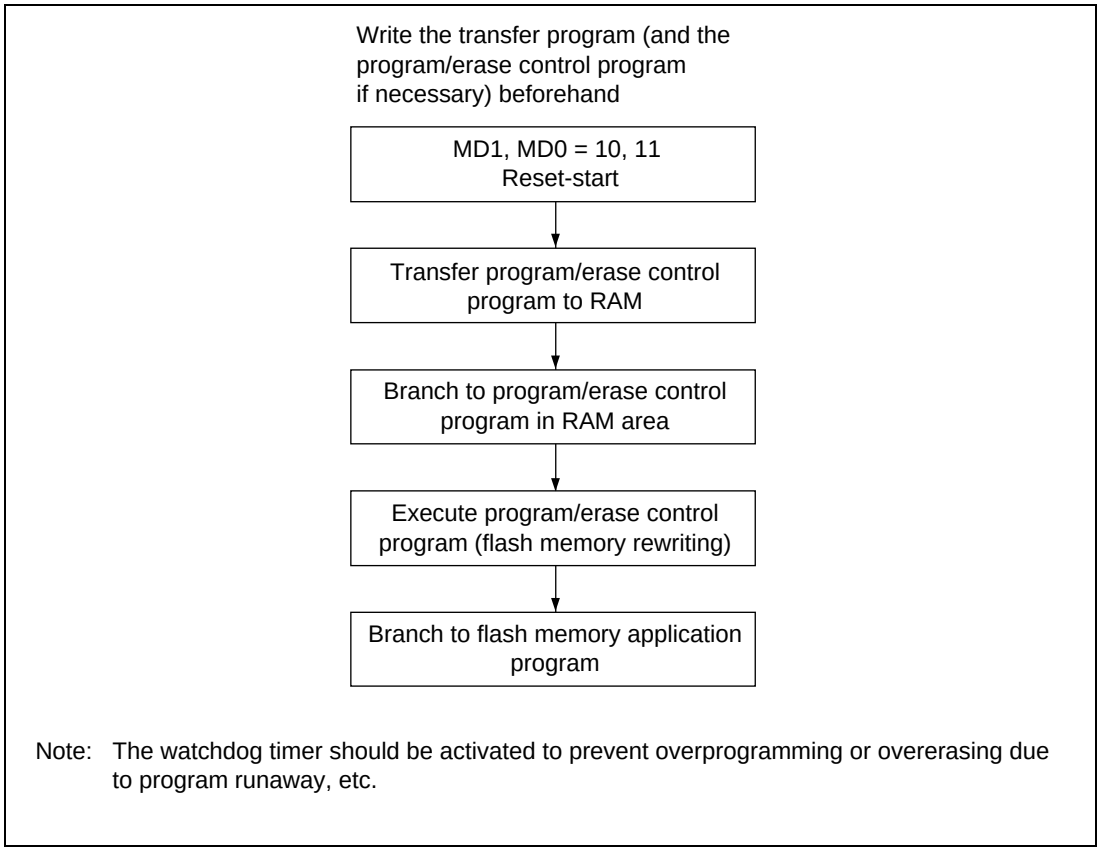


Figure 19.11 User Program Mode Execution Procedure

19.7 Programming/Erasing Flash Memory

In the on-board programming modes, flash memory programming and erasing is performed by software, using the CPU. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes can be made by setting the PSU and ESU bits in FLMCR2, and the P, E, PV, and EV bits in FLMCR1.

The flash memory cannot be read while being programmed or erased. Therefore, the program that controls flash memory programming/erasing (the programming control program) should be 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 P bits in FLMCR1, and the ESU and PSU bits in FLMCR2, is executed by a program in flash memory.
2. Perform programming in the erased state. Do not perform additional programming on previously programmed addresses.

19.7.1 Program Mode

Follow the procedure shown in the program/program-verify flowchart in figure 19.12 to write data or programs to flash memory. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 32 bytes at a time.

The wait times (x, y, z, α, β, γ, ε, η) after setting/clearing individual bits in flash memory control registers 1 and 2 (FLMCR1, FLMCR2) and the maximum number of writes (N) are shown in table 22.12 in section 22.5, Flash Memory Characteristics.

Following the elapse of (x) µs or more after the SWE bit is set to 1 in flash memory control register 1 (FLMCR1), 32-byte program data is stored in the program data area and reprogram data area, and the 32-byte data in the reprogram data area written consecutively to the write addresses. The lower 8 bits of the first address written to must be H'00, H'20, H'40, H'60, H'80, H'A0, H'C0, or H'E0. Thirty-two consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 32-byte data transfer must be performed even if writing fewer than 32 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer is set to prevent overprogramming in the event of program runaway, etc. Set a value greater than ($y + z + \alpha + \beta$) µs as the WDT overflow period. After this, preparation for program mode (program setup) is carried out by setting the PSU bit in FLMCR2, and after the elapse of (y) µs or more, the operating mode is switched to program mode by setting the P bit in FLMCR1. The time during which the P bit is set is the flash memory

programming time. Make a program setting so that the time for one programming operation is within the range of (z) µs.

19.7.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of a given programming time, the programming mode is exited (the P bit in FLMCR1 is cleared, then the PSU bit in FLMCR2 is cleared at least (α) µs later). The watchdog timer is cleared after the elapse of (β) µs or more, and the operating mode is switched to program-verify mode by setting the PV bit in FLMCR1. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 19.12) and transferred to the reprogram data area. After 32 bytes of data have been verified, exit program-verify mode, wait for at least
(η) µs, then clear the SWE bit in FLMCR1. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. However, ensure that the program/program-verify sequence is not repeated more than N times on the same bits.

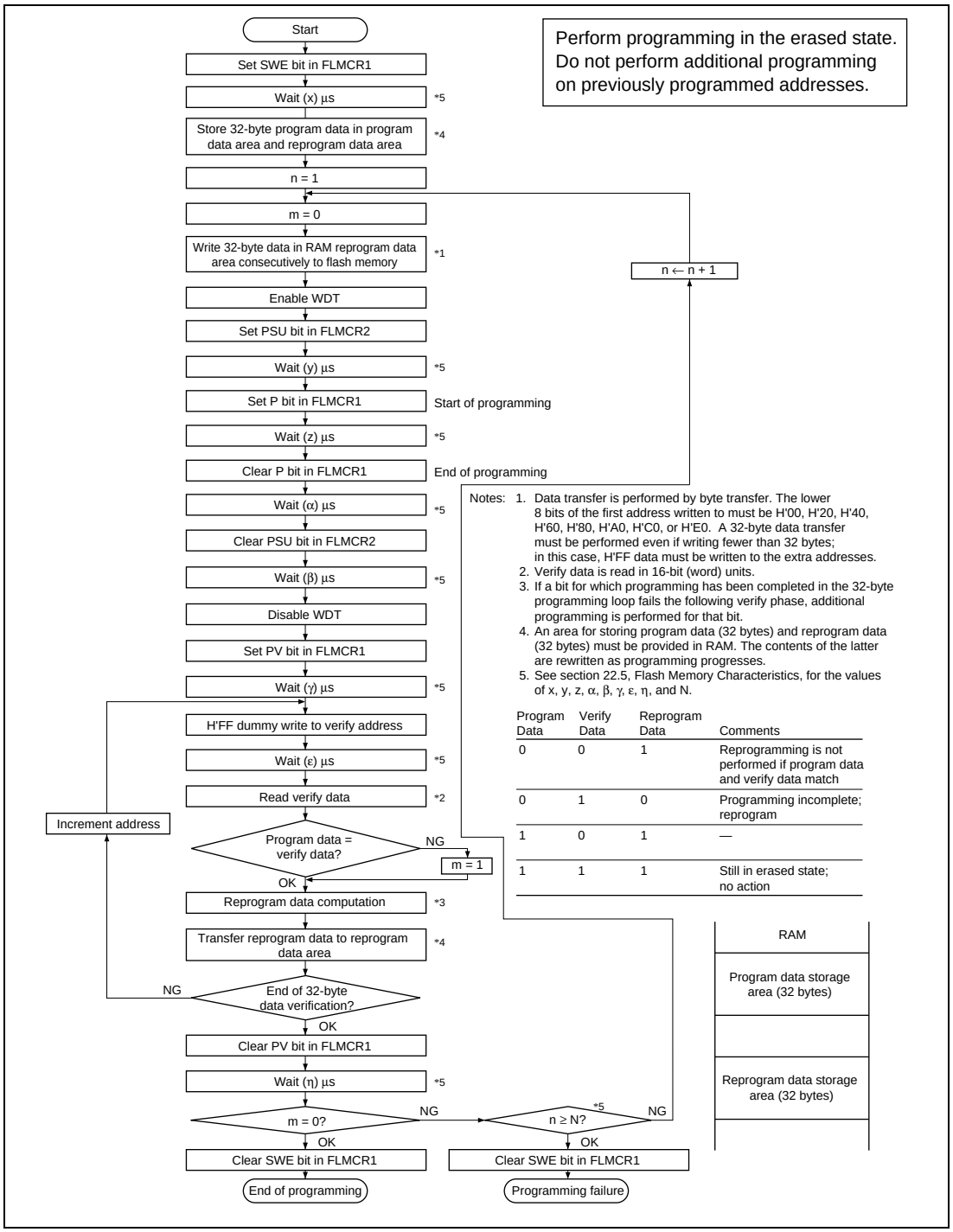


Figure 19.12 Program/Program-Verify Flowchart

19.7.3 Erase Mode

Flash memory erasing should be performed block by block following the procedure shown in the erase/erase-verify flowchart (single-block erase) shown in figure 19.13.

The wait times (x, y, z, α, β, γ, ε, η) after setting/clearing individual bits in flash memory control registers 1 and 2 (FLMCR1, FLMCR2) and the maximum number of erases (N) are shown in table 22.12 in section 22.5, Flash Memory Characteristics.

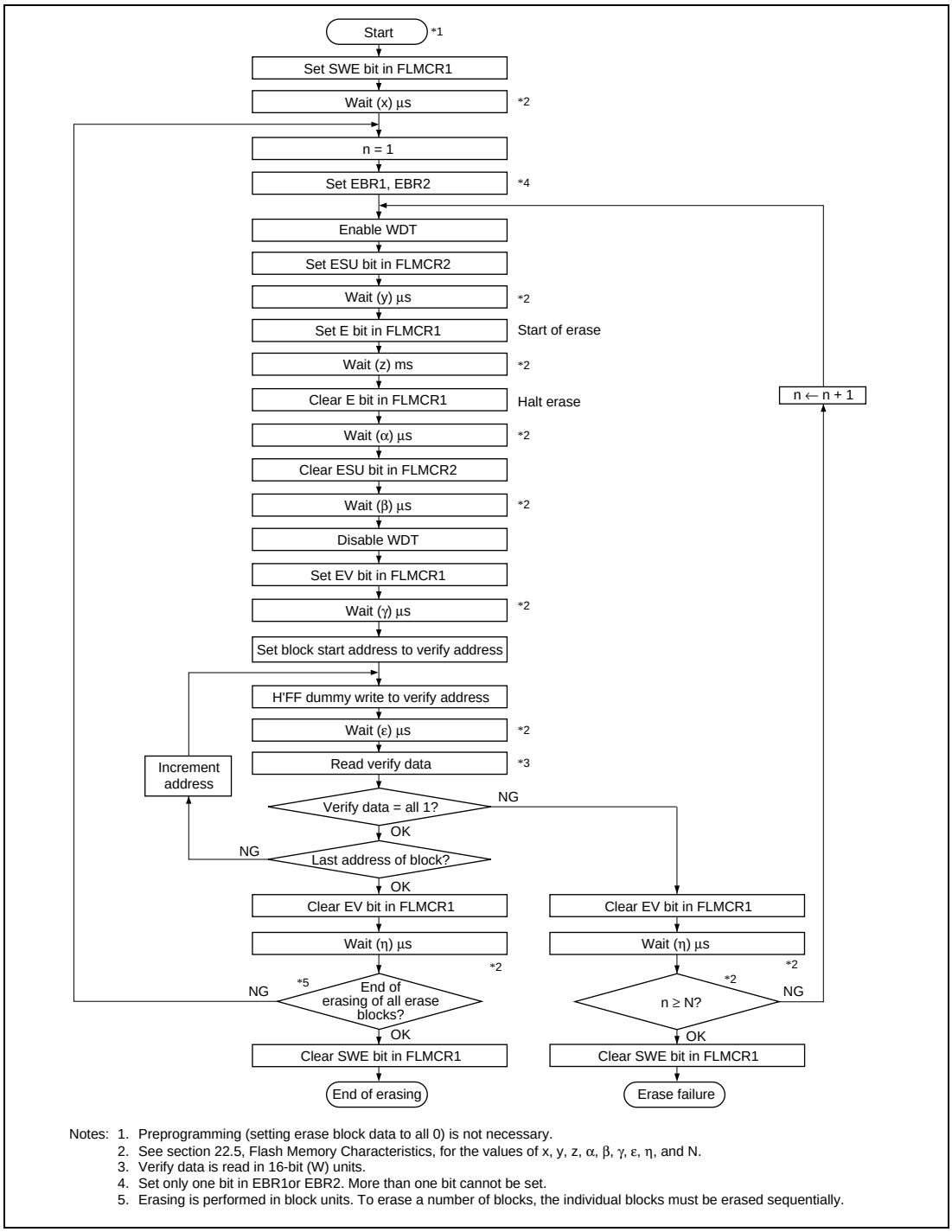
To perform data or program erasure, make a 1 bit setting for the flash memory area to be erased in erase block register 1 or 2 (EBR1 or EBR2) at least (x) µs after setting the SWE bit to 1 in flash memory control register 1 (FLMCR1). Next, the watchdog timer is set to prevent overerasing in the event of program runaway, etc. Set a value greater than ($y + z + \alpha + \beta$) ms as the WDT overflow period. After this, preparation for erase mode (erase setup) is carried out by setting the ESU bit in FLMCR2, and after the elapse of (y) µs or more, the operating mode is switched to erase mode by setting the E bit in FLMCR1. The time during which the E bit is set is the flash memory erase time. Ensure that the erase time does not exceed (z) ms.

Note: With flash memory erasing, preprogramming (setting all data in the memory to be erased to 0) is not necessary before starting the erase procedure.

19.7.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the erase time, erase mode is exited (the E bit in FLMCR1 is cleared, then the ESU bit in FLMCR2 is cleared at least (α) µs later), the watchdog timer is cleared after the elapse of (β) µs or more, and the operating mode is switched to erase-verify mode by setting the EV bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (γ) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (ε) µs after the dummy write before performing this read operation. If the read data has been erased (all 1), a dummy write is performed to the next address, and erase-verify is performed. If the read data has not been erased, set erase mode again, and repeat the erase/erase-verify sequence in the same way. However, ensure that the erase/erase-verify sequence is not repeated more than N times. When verification is completed, exit erase-verify mode, and wait for at least (η) µs. If erasure has been completed on all the erase blocks, clear the SWE bit in FLMCR1. If there are any unerased blocks, make a 1 bit setting in EBR1 or EBR2 for the flash memory area to be erased, and repeat the erase/erase-verify sequence in the same way.



Notes: 1. Preprogramming (setting erase block data to all 0) is not necessary.
2. See section 22.5, Flash Memory Characteristics, for the values of x, y, z, α, β, γ, ε, η, and N.
3. Verify data is read in 16-bit (W) 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 19.13 Erase/Erase-Verify Flowchart (Single-Block Erase)

19.8 Flash Memory Protection

There are three kinds of flash memory program/erase protection: hardware protection, software protection, and error protection.

19.8.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in flash memory control registers 1 and 2 (FLMCR1, FLMCR2) and erase block registers 1 and 2 (EBR1, EBR2). (See table 19.8.)

Table 19.8 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
Reset/standby protection	• In a reset (including a WDT overflow reset) and in hardware standby mode, software standby mode, subactive mode, subsleep mode, and watch mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

19.8.2 Software Protection

Software protection can be implemented by setting the SWE bit in FLMCR1 and erase block registers 1 and 2 (EBR1, EBR2). When software protection is in effect, setting the P or E bit in flash memory control register 1 (FLMCR1) does not cause a transition to program mode or erase mode. (See table 19.9.)

Table 19.9 Software Protection

		Functions	
Item	**Description**	**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.)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block registers 1 and 2 (EBR1, EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes

19.8.3 Error Protection

In error protection, an error is detected when MCU runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the MCU malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting 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 or instruction fetch)
- Immediately after exception handling (excluding a reset) during programming/erasing
- When a SLEEP instruction (transition to software standby, sleep, subactive, subsleep, or watch mode) is executed during programming/erasing
- When the bus is released during programming/erasing

Error protection is released only by a reset and in hardware standby mode.

Figure 19.14 shows the flash memory state transition diagram.

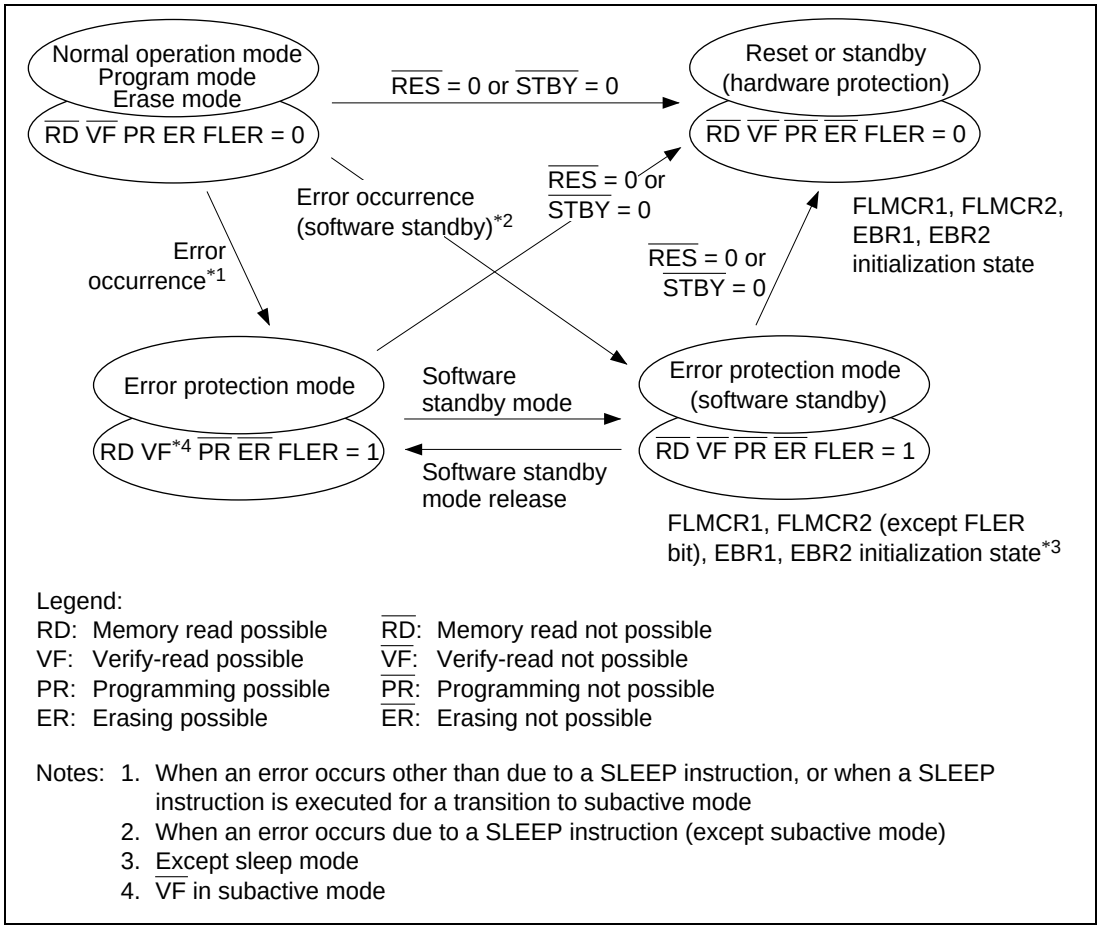


Figure 19.14 Flash Memory State Transitions

19.9 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI, should be disabled when flash memory is being programmed or erased (when the P or E bit is set in FLMCR1), and while the boot program is executing in boot mode*[1], to give priority to the program or erase operation. There are three reasons for this:

1. An interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. In the interrupt exception handling sequence during programming or erasing, the vector would not be read correctly*[2], possibly resulting in MCU runaway.
3. If an interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabling interrupts, as an exception to the general rule. However, this provision does not guarantee normal erasing and programming or MCU operation. All interrupt requests, including NMI, must therefore be disabled inside and outside the MCU when programming or erasing flash memory. Interrupts are also disabled in the error-protection state while the P or E bit remains set in FLMCR1.

Notes: 1. Interrupt requests must be disabled inside and outside the MCU until the programming control program has completed initial programming.

2. The vector may not be read correctly in this case for the following two reasons:
   - If flash memory is read while being programmed or erased (while the P or E bit is set in FLMCR1), correct read data will not be obtained (undetermined values will be returned).
   - If the interrupt entry in the vector table has not been programmed yet, interrupt exception handling will not be executed correctly.

19.10 Flash Memory Writer Mode

19.10.1 PROM Mode Setting

Programs and data can be written and erased in writer mode as well as in the on-board programming modes. In writer mode, the on-chip ROM can be freely programmed using a PROM programmer that supports Hitachi microcomputer device types with 128-kbyte on-chip flash memory. Flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

Table 19.10 shows writer mode pin settings.

Table 19.10 Writer Mode Pin Settings

Pin Names	Setting/External Circuit Connection
Mode pins: MD1, MD0	Low-level input to MD1, MD0
$\overline{\text{STBY}}$ pin	High-level input (Hardware standby mode not set)
$\overline{\text{RES}}$ pin	Power-on reset circuit
XTAL and EXTAL pins	Oscillation circuit
Other setting pins: P47, P42, P41, P40, P67	Low-level input to p42, p67, high-level input to P47, P41, P40

19.10.2 Socket Adapters and Memory Map

In writer mode, a socket adapter is mounted on the PROM programmer to match the package concerned. Ensure that the socket adapter is obtained from a writer manufacturer supporting the Hitachi microcomputer device type with 128-kbyte on-chip flash memory..

Figure 19.15 shows the memory map in writer mode. For pin names in writer mode, see section 1.3.2, Pin Functions in Each Operating Mode.

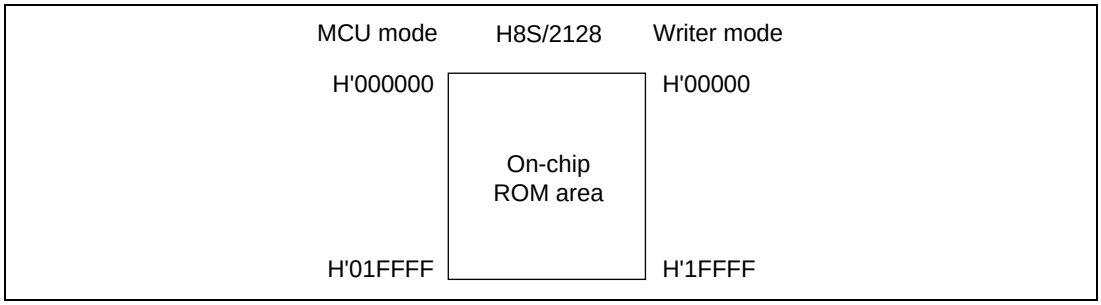


Figure 19.15 Memory Map in Writer Mode

19.10.3 Writer Mode Operation

Table 19.11 shows how the different operating modes are set when using writer mode, and table 19.12 lists the commands used in writer mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.
- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-erasing.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the FO6 signal. In status read mode, error information is output if an error occurs.

Table 19.11 Settings for Each Operating Mode in Writer Mode

	Pin Names				
Mode	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	**FO0 to FO7**	**FA0 to FA17**
Read	L	L	H	Data output	Ain
Output disable	L	H	H	Hi-z	X
Command write	L	H	L	Data input	Ain*[2]
Chip disable*[1]	H	X	X	Hi-z	X

Legend:
H: High level
L: Low level
Hi-z: High impedance
X: Don't care

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.

Table 19.12 Writer Mode Commands

	Number	1st Cycle			2nd Cycle		
Command Name	**of Cycles**	**Mode**	**Address**	**Data**	**Mode**	**Address**	**Data**
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	WA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

Legend:
RA: Read address
PA: Program address

Notes: 1. In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

19.10.4 Memory Read Mode

- After the end of an auto-program, auto-erase, or status read operation, the command wait state is entered. To read memory contents, a transition must be made to memory read mode by means of a command write before the read is executed.
- Command writes can be performed in memory read mode, just as in the command wait state.
- Once memory read mode has been entered, consecutive reads can be performed.
- After power-on, memory read mode is entered.

Table 19.13 AC Characteristics in Memory Read Mode
(Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

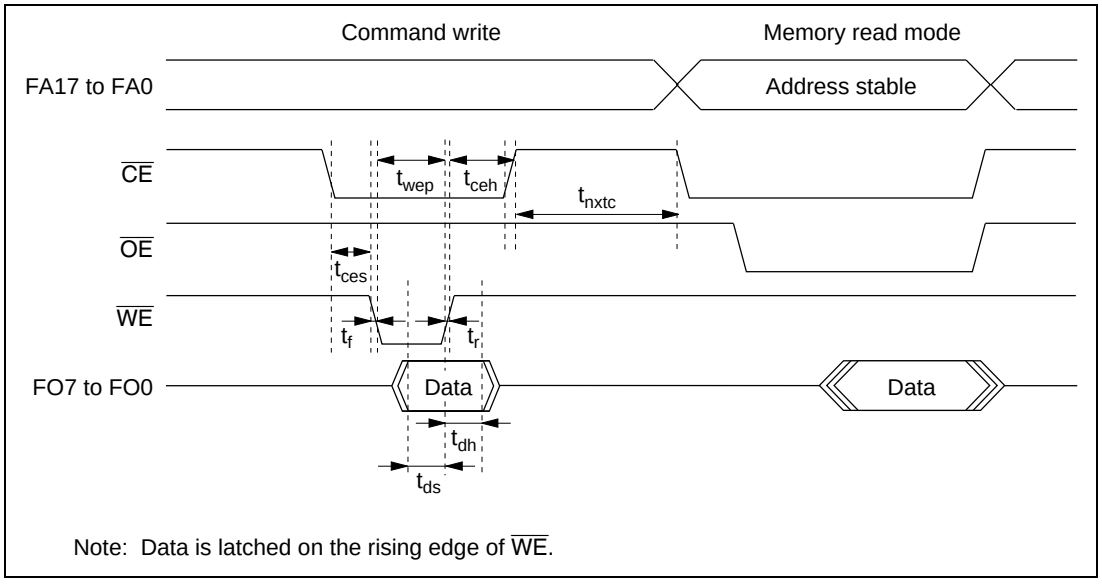


Figure 19.16 Memory Read Mode Timing Waveforms after Command Write

Table 19.14 AC Characteristics when Entering Another Mode from Memory Read Mode (Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

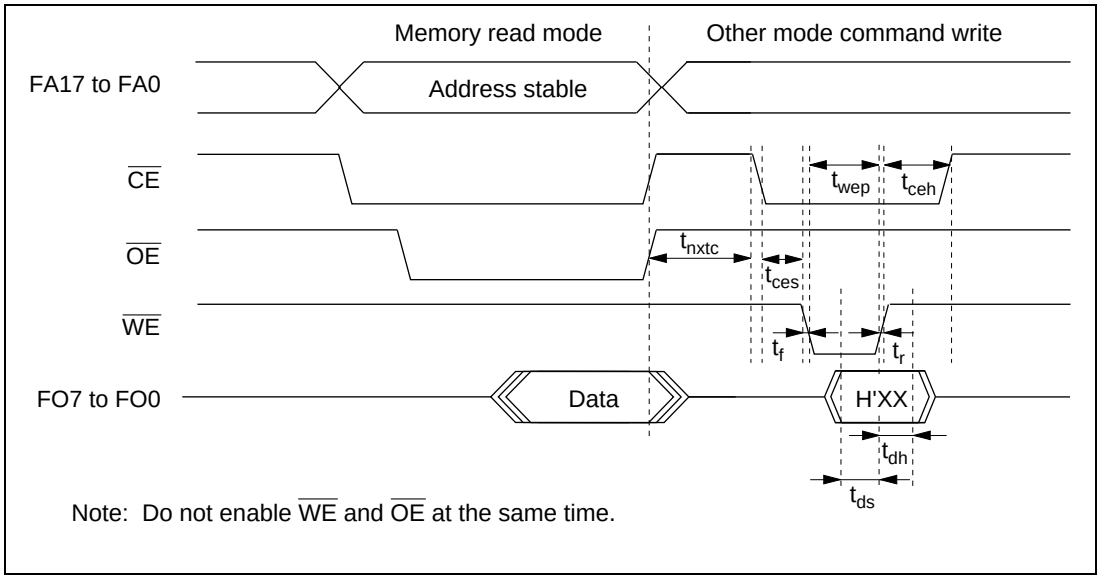


Figure 19.17 Timing Waveforms when Entering Another Mode from Memory Read Mode

Table 19.15 AC Characteristics in Memory Read Mode
(Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Access time	t_{acc}		20	µs	
$\overline{CE}$ output delay time	t_{ce}		150	ns	
$\overline{OE}$ output delay time	t_{oe}		150	ns	
Output disable delay time	t_{df}		100	ns	
Data output hold time	t_{oh}	5		ns	

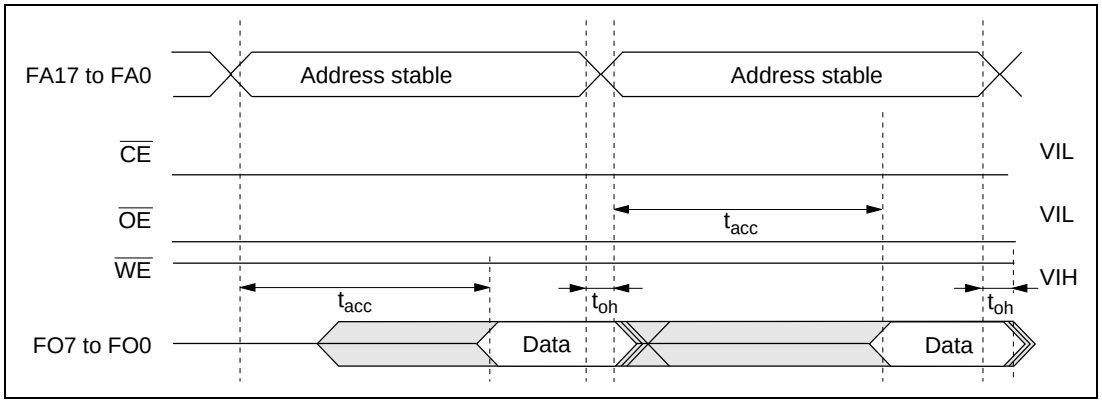


Figure 19.18 Timing Waveforms for $\overline{CE}$/$\overline{OE}$ Enable State Read

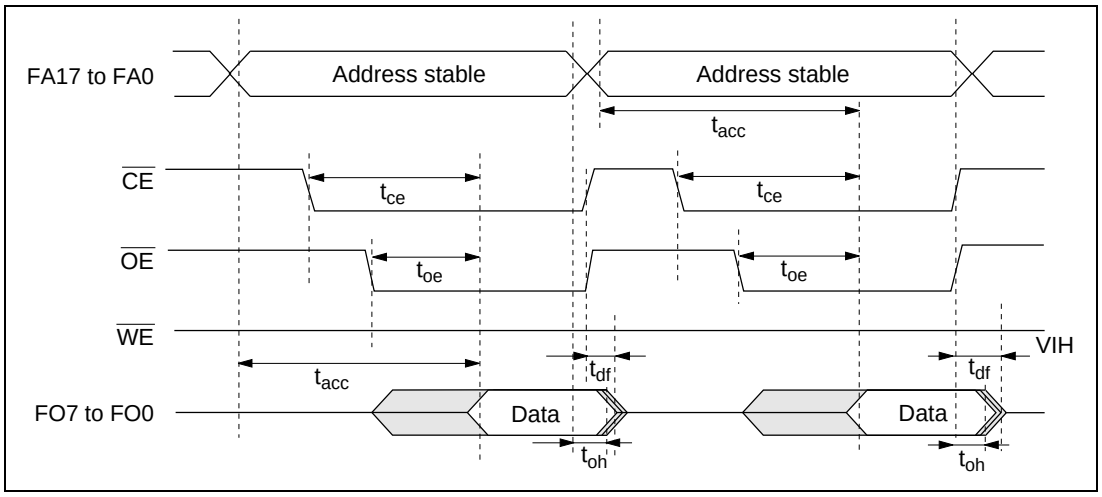


Figure 19.19 Timing Waveforms for $\overline{CE}$/$\overline{OE}$ Clocked Read

19.10.5 Auto-Program Mode

AC Characteristics

Table 19.16 AC Characteristics in Auto-Program Mode
(Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
Status polling start time	t_{wsts}	1		ms	
Status polling access time	t_{spa}		150	ns	
Address setup time	t_{as}	0		ns	
Address hold time	t_{ah}	60		ns	
Memory write time	t_{write}	1	3000	ms	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

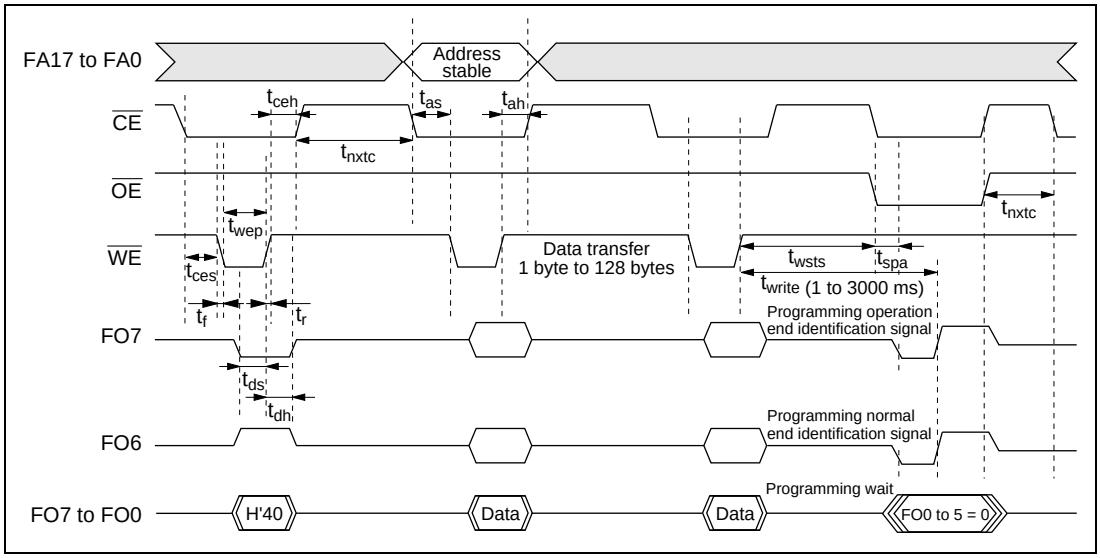


Figure 19.20 Auto-Program Mode Timing Waveforms

Notes on Use of Auto-Program Mode

- In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
- A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
- The lower 8 bits of the transfer address must be H'00 or H'80. If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
- Memory address transfer is performed in the second cycle (figure 19.20). Do not perform 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 FO6. Alternatively, status read mode can also be used for this purpose (FO7 status polling uses the auto-program operation end identification pin).
- The status polling FO6 and FO7 pin information is retained until the next command write. Until the next command write is performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

19.10.6 Auto-Erase Mode

AC Characteristics

Table 19.17 AC Characteristics in Auto-Erase Mode
(Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
Status polling start time	t_{ests}	1		ms	
Status polling access time	t_{spa}		150	ns	
Memory erase time	t_{erase}	100	40000	ms	
$\overline{WE}$ rise time	t_r		30	ns	
$\overline{WE}$ fall time	t_f		30	ns	

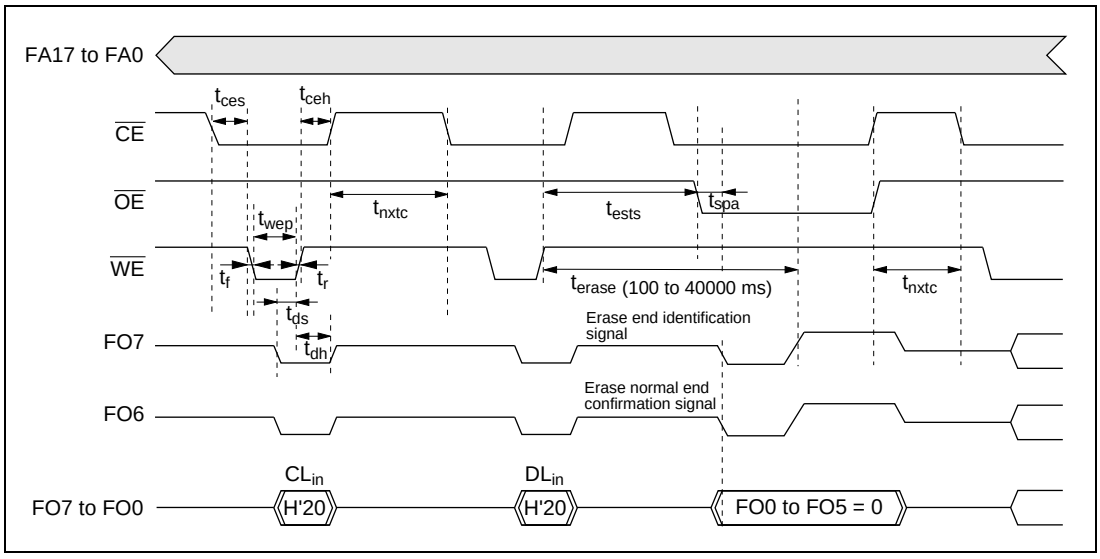


Figure 19.21 Auto-Erase Mode Timing Waveforms

Notes on Use of Auto-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 FO6. Alternatively, status read mode can also be used for this purpose (FO7 status polling uses the auto-erase operation end identification pin).
- The status polling FO6 and FO7 pin information is retained until the next command write. Until the next command write is performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

19.10.7 Status Read Mode

- Status read mode is used to identify what type of abnormal end has occurred. Use this mode 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 is performed.

Table 19.18 AC Characteristics in Status Read Mode
(Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit	Notes
Command write cycle	t_{nxtc}	20		µs	
$\overline{CE}$ hold time	t_{ceh}	0		ns	
$\overline{CE}$ setup time	t_{ces}	0		ns	
Data hold time	t_{dh}	50		ns	
Data setup time	t_{ds}	50		ns	
Write pulse width	t_{wep}	70		ns	
$\overline{OE}$ 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	

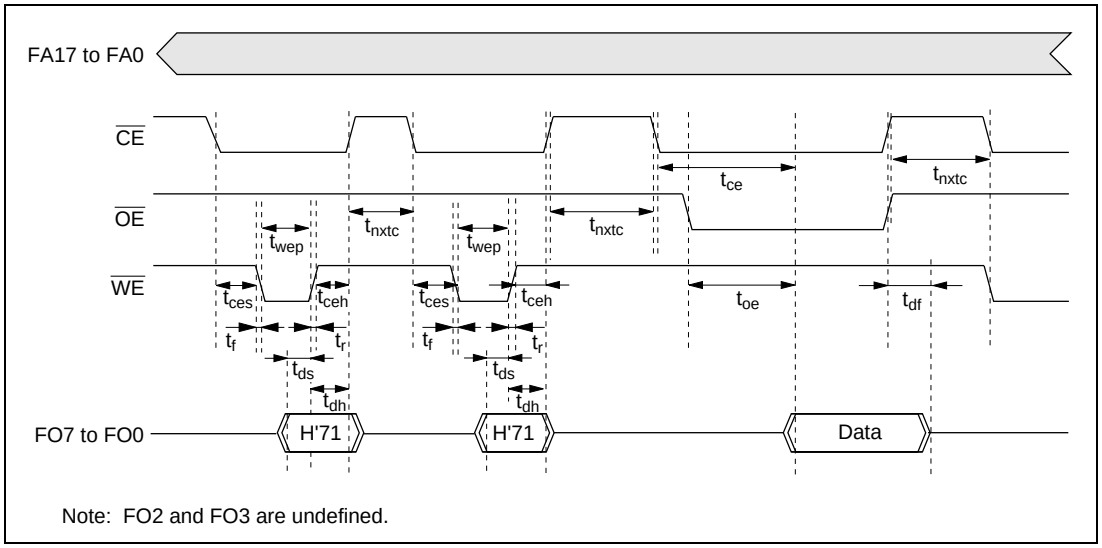


Figure 19.22 Status Read Mode Timing Waveforms

Table 19.19 Status Read Mode Return Commands

Pin Name	FO7	FO6	FO5	FO4	FO3	FO2	FO1	FO0
Attribute	Normal end identification	Command error	Programming error	Erase error	—	—	Programming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Programming error: 1 Otherwise: 0	Erase error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address error: 1 Otherwise: 0

Note: FO2 and FO3 are undefined.

19.10.8 Status Polling

- The FO7 status polling flag indicates the operating status in auto-program or auto-erase mode.
- The FO6 status polling flag indicates a normal or abnormal end in auto-program or auto-erase mode.

Table 19.20 Status Polling Output Truth Table

Pin Names	Internal Operation in Progress	Abnormal End	—	Normal End
FO7	0	1	0	1
FO6	0	0	1	1
FO0 to FO5	0	0	0	0

19.10.9 Writer Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the writer mode setup period. After the writer mode setup time, a transition is made to memory read mode.

Table 19.21 Command Wait State Transition Time Specifications

Item	Symbol	Min	Max	Unit	Notes
Standby release (oscillation stabilization time)	t_{osc1}	20	—	ms	
PROM mode setup time	t_{bmv}	10	—	ms	
V_{CC} hold time	t_{dwn}	0	—	ms	

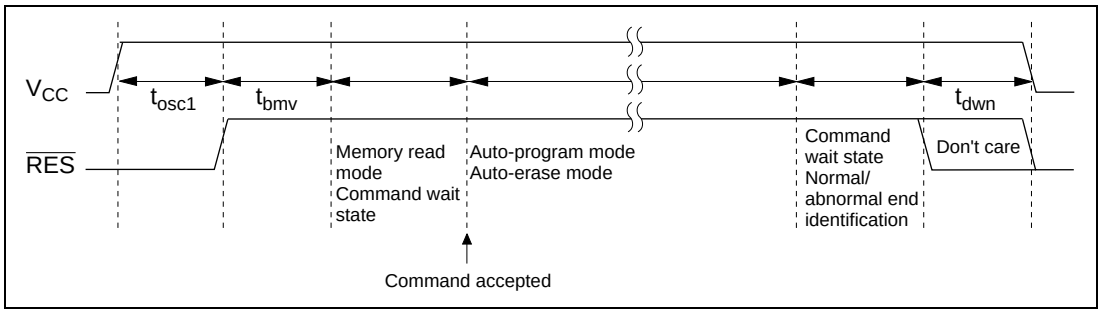


Figure 19.23 Oscillation Stabilization Time and Writer Mode Setup and Power Supply Fall Sequence

19.10.10 Notes On Memory Programming

- When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
- When performing programming using writer mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Hitachi. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.
2. Auto-programming should be performed once only on the same address block.

19.11 Flash Memory Programming and Erasing Precautions

Precautions concerning the use of on-board programming mode and writer mode are summarized below.

Use the specified voltages and timing for programming and erasing: Applied voltages in excess of the rating can permanently damage the device. Use a PROM programmer that supports Hitachi microcomputer device types with 128-kbyte on-chip flash memory.

Do not select the HN28F101 setting for the PROM programmer, and only use the specified socket adapter. Incorrect use will result in damaging the device.

Powering on and off: When applying or disconnecting V_{CC}, fix the $\overline{RES}$ pin low and place the flash memory in the hardware protection state.

The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery.

Use the recommended algorithm when programming and erasing flash memory: The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P or E bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

Do not set or clear the SWE bit during program execution in flash memory: Clear the SWE bit before executing a generated or reading data in flash memory. When the SWE bit is set, data in flash memory can be rewritten, but flash memory should only be accessed for verify operations (verification during programming/erasing).

Do not use interrupts while flash memory is being programmed or erased: All interrupt requests, including NMI, should be disabled when programming or erasing flash memory to give priority to program/erase operations.

Do not perform additional programming. Erase the memory before reprogramming: In on-board programming, perform only one programming operation on a 32-byte programming unit block. In PROM mode, too, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.

Before programming, check that the chip is correctly mounted in the PROM programmer: Overcurrent damage to the device can result if the index marks on the PROM programmer socket, socket adapter, and chip are not correctly aligned.

Do not touch the socket adapter or chip during programming: Touching either of these can cause contact faults and write errors.

Section 20 Clock Pulse Generator

20.1 Overview

The H8S/2128 Series and H8S/2124 Series have a built-in clock pulse generator (CPG) that generates the system clock (ø), the bus master clock, and internal clocks.

The clock pulse generator consists of an oscillator circuit, a duty adjustment circuit, clock selection circuit, medium-speed clock divider, bus master clock selection circuit, subclock input circuit, and waveform shaping circuit.

20.1.1 Block Diagram

Figure 20.1 shows a block diagram of the clock pulse generator.

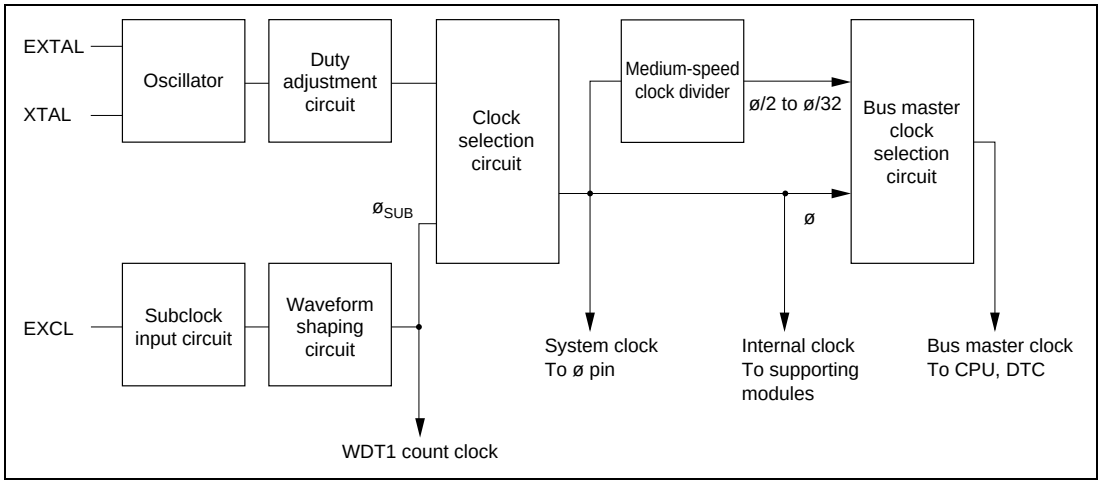


Figure 20.1 Block Diagram of Clock Pulse Generator

20.1.2 Register Configuration

The clock pulse generator is controlled by the standby control register (SBYCR) and low-power control register (LPWRCR). Table 20.1 shows the register configuration.

Table 20.1 CPG Registers

Name	Abbreviation	R/W	Initial Value	Address*
Standby control register	SBYCR	R/W	H'00	H'FF84
Low-power control register	LPWRCR	R/W	H'00	H'FF85

Note: * Lower 16 bits of the address.

20.2 Register Descriptions

20.2.1 Standby Control Register (SBYCR)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

Only bits 0 to 2 are described here. For a description of the other bits, see section 21.2.1, Standby Control Register (SBYCR).

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

Bits 2 to 0—System Clock Select 2 to 0 (SCK2 to SCK0): These bits select the bus master clock for high-speed mode and medium-speed mode.

When operating the device after a transition to subactive mode or watch mode bits SCK2 to SCK0 should all be cleared to 0.

Bit 2	Bit 1	Bit 0		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master is in high-speed mode	(Initial value)
		1	Medium-speed clock is ø/2	
	1	0	Medium-speed clock is ø/4	
		1	Medium-speed clock is ø/8	
1	0	0	Medium-speed clock is ø/16	
		1	Medium-speed clock is ø/32	
	1	—	—	

20.2.2 Low-Power Control Register (LPWRCR)

Bit	7	6	5	4	3	2	1	0
	DTON	LSON	NESEL	EXCLE	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

LPWRCR is an 8-bit readable/writable register that performs power-down mode control.

Only bit 4 is described here. For a description of the other bits, see section 21.2.2, Low-Power Control Register (LPWRCR).

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

Bit 4—Subclock Input Enable (EXCLE): Controls subclock input from the EXCL pin.

Bit 4		
EXCLE	**Description**	
0	Subclock input from EXCL pin is disabled	(Initial value)
1	Subclock input from EXCL pin is enabled	

20.3 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

20.3.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 20.2. Select the damping resistance R_d according to table 20.2. An AT-cut parallel-resonance crystal should be used.

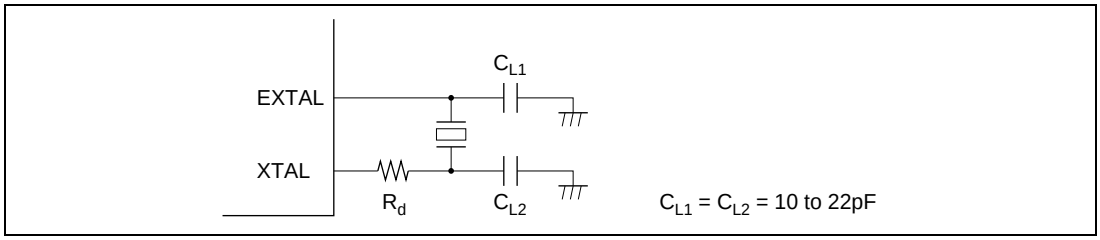


Figure 20.2 Connection of Crystal Resonator (Example)

Table 20.2 Damping Resistance Value

Frequency (MHz)	2	4	8	10	12	16	20
R_d (Ω)	1 k	500	200	0	0	0	0

Crystal resonator: Figure 20.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 20.3 and the same frequency as the system clock (ø).

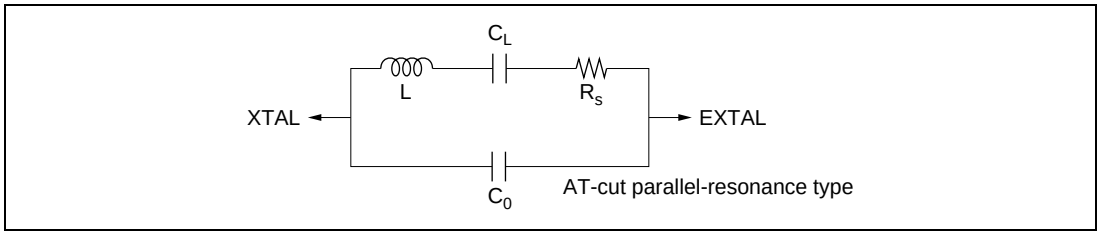


Figure 20.3 Crystal Resonator Equivalent Circuit

Table 20.3 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	10	12	16	20
R_S max (Ω)	500	120	80	70	60	50	40
C_0 max (pF)	7	7	7	7	7	7	7

Note on Board Design: When a crystal resonator is connected, the following points should be noted.

Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 20.4.

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins.

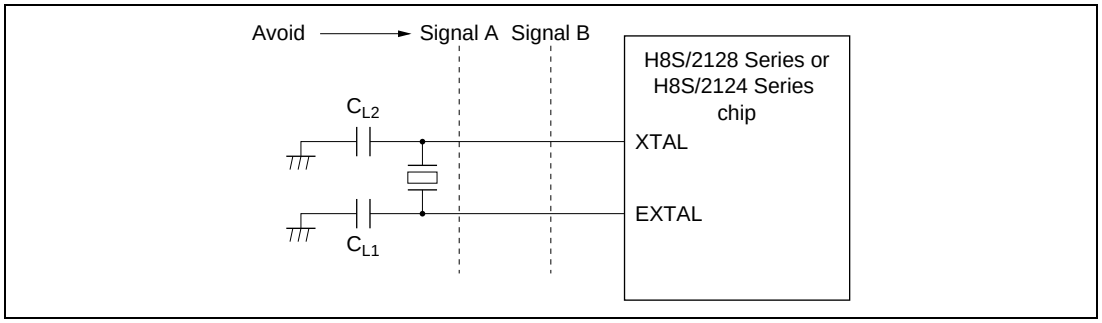


Figure 20.4 Example of Incorrect Board Design

20.3.2 External Clock Input

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 20.5. If the XTAL pin is left open, make sure that stray capacitance is no more than 10 pF.

In example (b), make sure that the external clock is held high in standby mode, subactive mode, subsleep mode, and wach mode.

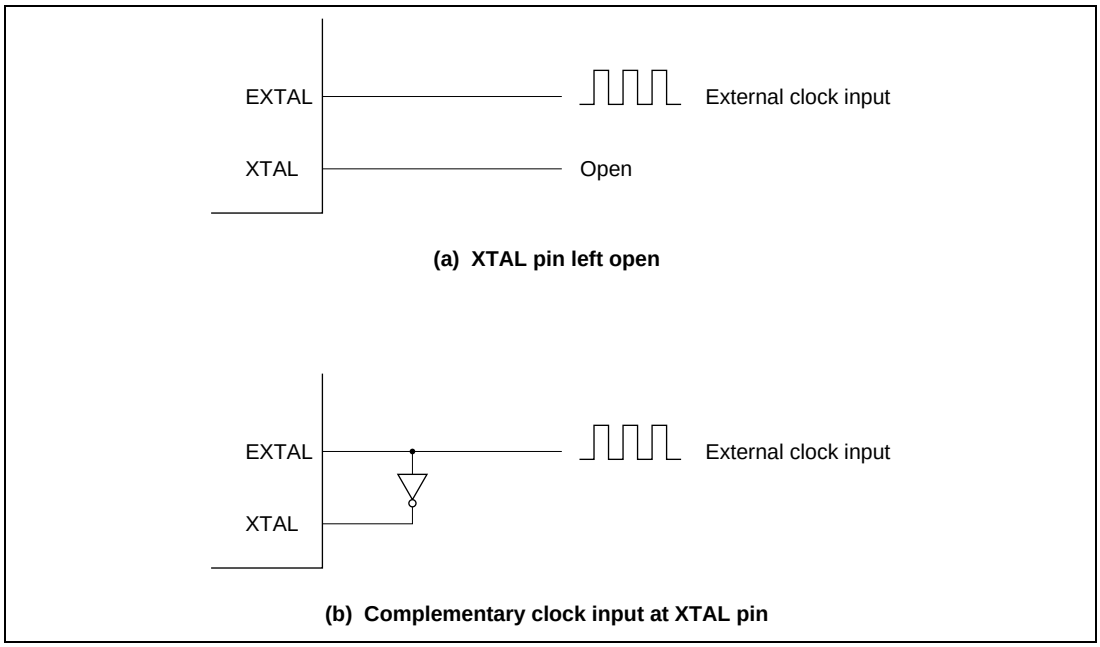


Figure 20.5 External Clock Input (Examples)

External Clock: The external clock signal should have the same frequency as the system clock (ø).

Table 20.4 and figure 20.6 show the input conditions for the external clock.

Table 20.4 External Clock Input Conditions

		V_{cc} = 2.7 to 5.5 V		V_{cc} = 5.0 V ±10%			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**
External clock input low pulse width	t_{EXL}	40	—	20	—	ns	Figure 20.6
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	t_{CL}	0.4	0.6	0.4	0.6	t_{cyc}	ø ≥ 5 MHz Figure 22.4
		80	—	80	—	ns	ø < 5 MHz
Clock high pulse width	t_{CH}	0.4	0.6	0.4	0.6	t_{cyc}	ø ≥ 5 MHz
		80	—	80	—	ns	ø < 5 MHz

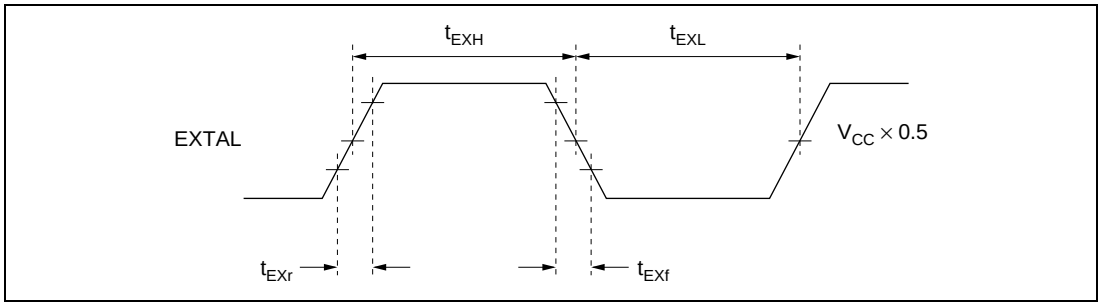


Figure 20.6 External Clock Input Timing

Table 20.5 shows the external clock output settling delay time, and figure 20.7 shows the external clock output settling delay timing. The oscillator and duty adjustment circuit have a function for adjusting the waveform of the external clock input at the EXTAL pin. When the prescribed clock signal is input at the EXTAL pin, internal clock signal output is fixed after the elapse of the external clock output settling delay time (t_{DEXT}). As the clock signal output is not fixed during the t_{DEXT} period, the reset signal should be driven low to maintain the reset state.

Table 20.5 External Clock Output Settling Delay Time

Conditions: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V, V_{SS} = AV_{SS} = 0 V

Item	Symbol	Min	Max	Unit	Notes
External clock output settling delay time	t_{DEXT}*	500	—	µs	Figure 20.7

Note: *t_{DEXT} includes $\overline{RES}$ pulse width (t_{RESW}).

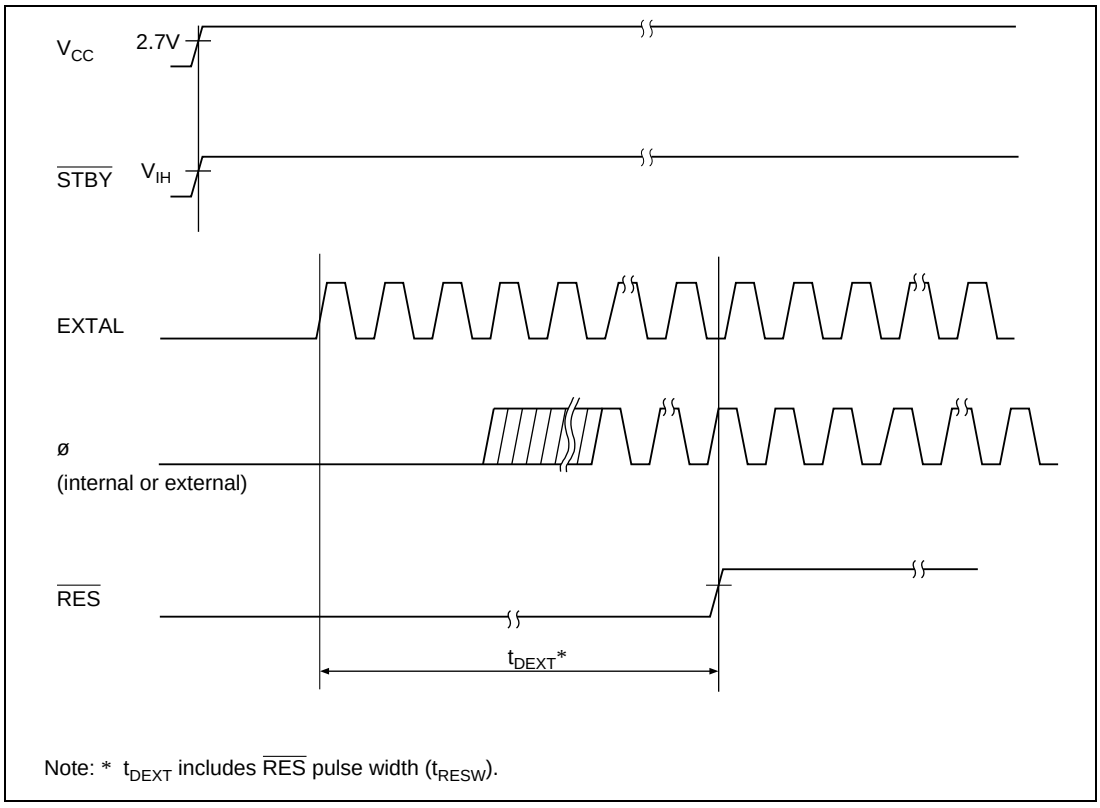


Figure 20.7 External Clock Output Settling Delay Timing

20.4 Duty Adjustment Circuit

When the oscillator frequency is 5 MHz or higher, the duty adjustment circuit adjusts the duty cycle of the clock signal from the oscillator to generate the system clock (ø).

20.5 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate ø/2, ø/4, ø/8, ø/16, and ø/32 clocks.

20.6 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the system clock (ø) or one of the medium-speed clocks (ø/2, ø/4, ø/8, ø/16, or ø/32) to be supplied to the bus master, according to the settings of bits SCK2 to SCK0 in SBYCR.

20.7 Subclock Input Circuit

The subclock input circuit controls the subclock input from the EXCL pin.

Inputting the Subclock: When a subclock is used, a 32.768 kHz external clock should be input from the EXCL pin. In this case, clear bit P46DDR to 0 in P4DDR and set bit EXCLE to 1 in LPWRCR.

The subclock input conditions are shown in table 20.6 and figure 20.8.

Table 20.6 Subclock Input Conditions

		V_{cc} = 2.7 to 5.5 V				
Item	**Symbol**	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
Subclock input low pulse width	t_{EXCLL}	—	15.26	—	µs	Figure 20.8
Subclock input high pulse width	t_{EXCLH}	—	15.26	—	µs	
Subclock input rise time	t_{EXCLr}	—	—	10	ns	
Subclock input fall time	t_{EXCLf}	—	—	10	ns	

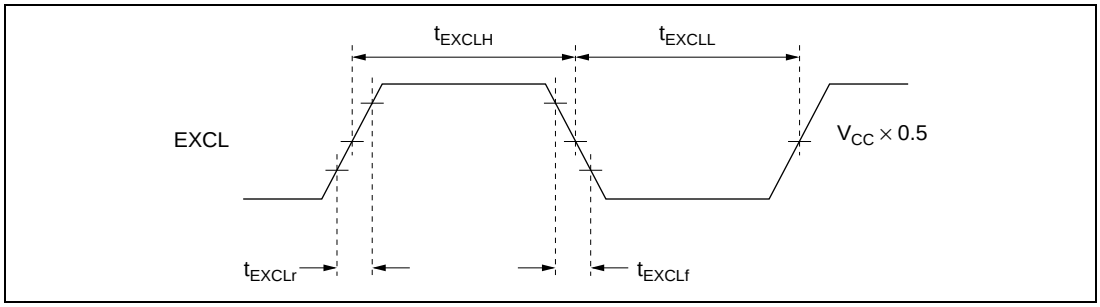


Figure 20.8 Subclock Input Timing

When Subclock is not Needed: Do not enable subclock input when the subclock is not needed.

20.8 Subclock Waveform Shaping Circuit

To eliminate noise in the subclock input from the EXCL pin, this circuit samples the clock using a clock obtained by dividing the ø clock. The sampling frequency is set with the NESEL bit in LPWRCR. For details, see section 21.2.2, Low-Power Control Register (LPWRCR). The clock is not sampled in subactive mode, subsleep mode, or watch mode.

Section 21 Power-Down State

21.1 Overview

In addition to the normal program execution state, the H8S/2128 Series and H8S/2124 Series have a power-down state in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip supporting modules, and so on.

The H8S/2128 Series and H8S/2124 Series operating modes are as follows:

1. High-speed mode
2. Medium-speed mode
3. Subactive mode
4. Sleep mode
5. Subsleep mode
6. Watch mode
7. Module stop mode
8. Software standby mode
9. Hardware standby mode

Of these, 2 to 9 are power-down modes. Sleep mode and subsleep mode are CPU modes, medium-speed mode is a CPU and bus master mode, subactive mode is a CPU, bus master, and on-chip supporting module mode, and module stop mode is an on-chip supporting module mode (including bus masters other than the CPU). Certain combinations of these modes can be set.

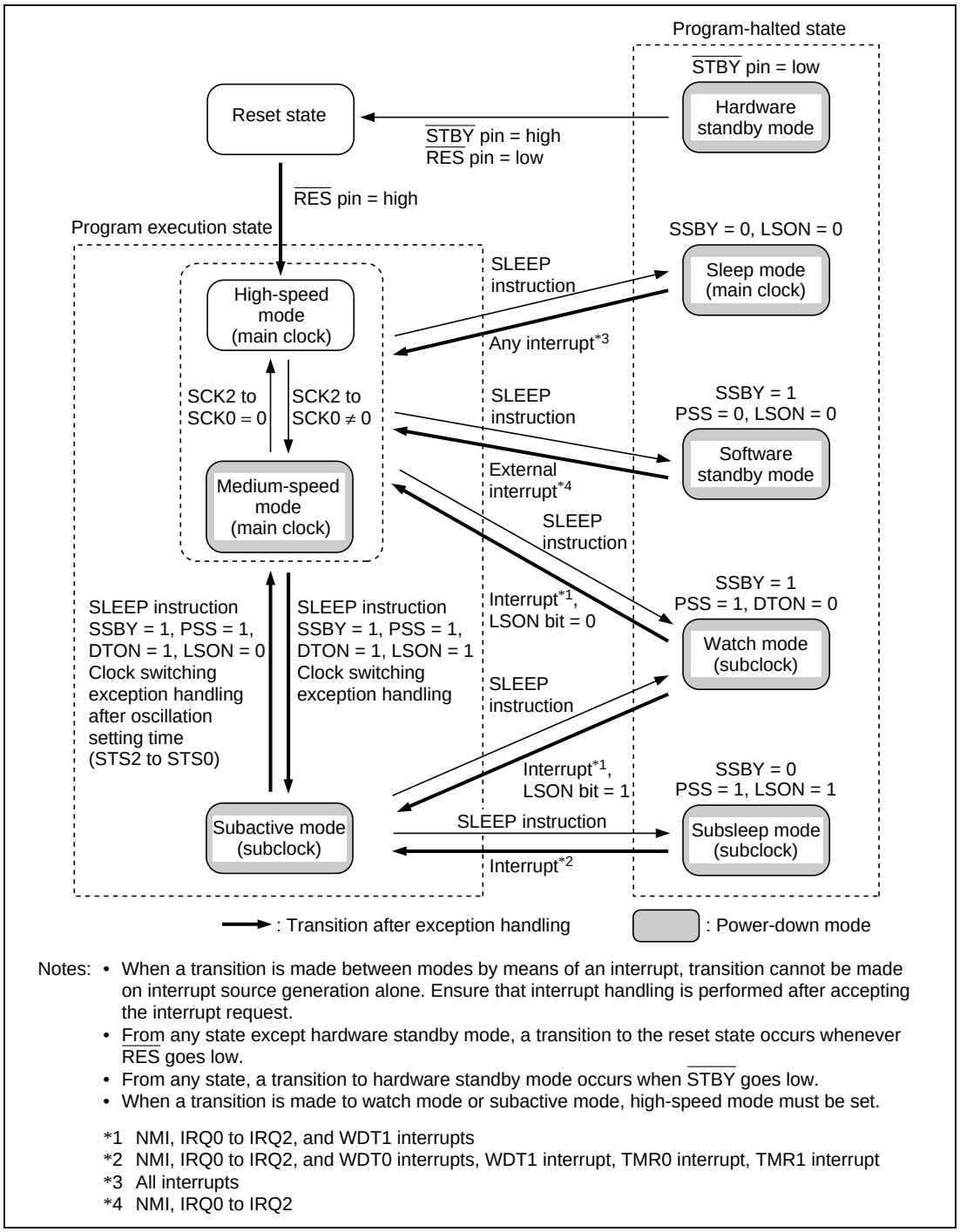
After a reset, the MCU is in high-speed mode and module stop mode (excluding the DTC).

Table 21.1 shows the internal chip states in each mode, and table 21.2 shows the conditions for transition to the various modes. Figure 21.1 shows a mode transition diagram.

Table 21.1 H8S/2128 Series and H8S/2124 Series Internal States in Each Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Subactive	Subsleep	Software Standby	Hardware Standby
System clock oscillator		Function-ing	Function-ing	Function-ing	Function-ing	Halted	Halted	Halted	Halted	Halted
Subclock input		Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Halted	Halted
CPU operation	Instruc-tions	Function-ing	Medium-speed	Halted	Function-ing	Halted	Subclock operation	Halted	Halted	Halted
	Registers			Retained		Retained		Retained	Retained	Undefined
External interrupts	NMI	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Halted
	IRQ0									
	IRQ1									
	IRQ2									
On-chip supporting module operation	DTC	Function-ing	Medium-speed	Function-ing	Function-ing/halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	WDT1	Function-ing	Function-ing	Function-ing	Function-ing	Subclock operation	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	WDT0					Halted (retained)				
	TMR0, 1				Function-ing/halted (retained)					
	FRT						Halted (retained)	Halted (retained)		
	TMRX, Y									
	Timer connec-tion									
	IIC0									
	IIC1									
	SCI0				Function-ing/halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	
	SCI1									
	PWM									
	PWMX									
	A/D									
	RAM	Function-ing	Function-ing	Function-ing (DTC)	Function-ing	Retained	Function-ing	Retained	Retained	Retained
	I/O	Function-ing	Function-ing	Function-ing	Function-ing	Retained	Function-ing	Retained	Retained	High impedance

Note: “Halted (retained)” means that internal register values are retained. The internal state is “operation suspended.”
“Halted (reset)” means that internal register values and internal states are initialized.
In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).



Notes:
- When a transition is made between modes by means of an interrupt, transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
- From any state except hardware standby mode, a transition to the reset state occurs whenever RES goes low.
- From any state, a transition to hardware standby mode occurs when STBY goes low.
- When a transition is made to watch mode or subactive mode, high-speed mode must be set.

*1 NMI, IRQ0 to IRQ2, and WDT1 interrupts
*2 NMI, IRQ0 to IRQ2, and WDT0 interrupts, WDT1 interrupt, TMR0 interrupt, TMR1 interrupt
*3 All interrupts
*4 NMI, IRQ0 to IRQ2

Figure 21.1 Mode Transitions

Table 21.2 Power-Down Mode Transition Conditions

State before Transition	Control Bit States at Time of Transition				State after Transition by SLEEP Instruction	State after Return by Interrupt
	SSBY	PSS	LSON	DTON		
High-speed/ medium-speed	0	*	0	*	Sleep	High-speed/ medium-speed
	0	*	1	*	—	—
	1	0	0	*	Software standby	High-speed/ medium-speed
	1	0	1	*	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Subactive
	1	1	0	1	—	—
	1	1	1	1	Subactive	—
Subactive	0	0	*	*	—	—
	0	1	0	*	—	—
	0	1	1	*	Subsleep	Subactive
	1	0	*	*	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Subactive
	1	1	0	1	High-speed	—
	1	1	1	1	—	—

*: Don’t care

—: Do not set.

21.1.1 Register Configuration

The power-down state is controlled by the SBYCR, LPWRCR, TCSR (WDT1), and MSTPCR registers. Table 21.3 summarizes these registers.

Table 21.3 Power-Down State Registers

Name	Abbreviation	R/W	Initial Value	Address*
Standby control register	SBYCR	R/W	H'00	H'FF84
Low-power control register	LPWRCR	R/W	H'00	H'FF85
Timer control/status register (WDT1)	TCSR	R/W	H'00	H'FFEA
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

Note: * Lower 16 bits of the address.

21.2 Register Descriptions

21.2.1 Standby Control Register (SBYCR)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

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

Bit 7—Software Standby (SSBY): Determines the operating mode, in combination with other control bits, when a power-down mode transition is made by executing a SLEEP instruction. The SSBY setting is not changed by a mode transition due to an interrupt, etc.

Bit 7		
SSBY	**Description**	
0	Transition to sleep mode after execution of SLEEP instruction in high-speed mode or medium-speed mode	(Initial value)
	Transition to subsleep mode after execution of SLEEP instruction in subactive mode	
1	Transition to software standby mode, subactive mode, or watch mode after execution of SLEEP instruction in high-speed mode or medium-speed mode	
	Transition to watch mode or high-speed mode after execution of SLEEP instruction in subactive mode	

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the time the MCU waits for the clock to stabilize when software standby mode, watch mode, or subactive mode is cleared and a transition is made to high-speed mode or medium-speed mode by means of a specific interrupt or instruction. With crystal oscillation, refer to table 21.4 and make a selection according to the operating frequency so that the standby time is at least 8 ms (the oscillation settling time). With an external clock, any selection can be made.

Bit 6	Bit 5	Bit 4		
STS2	**STS1**	**STS0**	**Description**	
0	0	0	Standby time = 8192 states	(Initial value)
		1	Standby time = 16384 states	
	1	0	Standby time = 32768 states	
		1	Standby time = 65536 states	
1	0	0	Standby time = 131072 states	
		1	Standby time = 262144 states	
	1	0	Reserved	
		1	Standby time = 16 states*	

Note: * This setting must not be used in the flash memory version.

Bit 3—Reserved: This bit cannot be modified and is always read as 0.

Bits 2 to 0—System Clock Select (SCK2 to SCK0): These bits select the clock for the bus master in high-speed mode and medium-speed mode. When operating the device after a transition to subactive mode or watch mode, bits SCK2 to SCK0 should all be cleared to 0.

Bit 2	Bit 1	Bit 0		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master is in high-speed mode	(Initial value)
		1	Medium-speed clock is ø/2	
	1	0	Medium-speed clock is ø/4	
		1	Medium-speed clock is ø/8	
1	0	0	Medium-speed clock is ø/16	
		1	Medium-speed clock is ø/32	
	1	—	—	

21.2.2 Low-Power Control Register (LPWRCR)

Bit	7	6	5	4	3	2	1	0
	DTON	LSON	NESEL	EXCLE	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

LPWRCR is an 8-bit readable/writable register that performs power-down mode control.

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

Bit 7—Direct-Transfer On Flag (DTON): Specifies whether a direct transition is made between high-speed mode, medium-speed mode, and subactive mode when making a power-down transition by executing a SLEEP instruction. The operating mode to which the transition is made after SLEEP instruction execution is determined by a combination of other control bits.

Bit 7

DTON	Description
0	When a SLEEP instruction is executed in high-speed mode or medium-speed mode, a transition is made to sleep mode, software standby mode, or watch mode*
	When a SLEEP instruction is executed in subactive mode, a transition is made to subsleep mode or watch mode (Initial value)
1	When a SLEEP instruction is executed in high-speed mode or medium-speed mode, a transition is made directly to subactive mode*, or a transition is made to sleep mode or software standby mode
	When a SLEEP instruction is executed in subactive mode, a transition is made directly to high-speed mode, or a transition is made to subsleep mode

Note: * When a transition is made to watch mode or subactive mode, high-speed mode must be set.

Bit 6—Low-Speed On Flag (LSON): Determines the operating mode in combination with other control bits when making a power-down transition by executing a SLEEP instruction. Also controls whether a transition is made to high-speed mode or to subactive mode when watch mode is cleared.

Bit 6

LSON	Description
0	When a SLEEP instruction is executed in high-speed mode or medium-speed mode, a transition is made to sleep mode, software standby mode, or watch mode*
	When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode, or directly to high-speed mode
	After watch mode is cleared, a transition is made to high-speed mode (Initial value)
1	When a SLEEP instruction is executed in high-speed mode a transition is made to watch mode or subactive mode*
	When a SLEEP instruction is executed in subactive mode, a transition is made to subsleep mode or watch mode
	After watch mode is cleared, a transition is made to subactive mode

Note: * When a transition is made to watch mode or subactive mode, high-speed mode must be set.

Bit 5—Noise Elimination Sampling Frequency Select (NESEL): Selects the frequency at which the subclock (øSUB) input from the EXCL pin is sampled with the clock (ø) generated by the system clock oscillator. When ø = 5 MHz or higher, clear this bit to 0.

Bit 5		
NESEL	**Description**	
0	Sampling at ø divided by 32	(Initial value)
1	Sampling at ø divided by 4	

Bit 4—Subclock Input Enable (EXCLE): Controls subclock input from the EXCL pin.

Bit 4		
EXCLE	**Description**	
0	Subclock input from EXCL pin is disabled	(Initial value)
1	Subclock input from EXCL pin is enabled	

Bits 3 to 0—Reserved: These bits cannot be modified and are always read as 0.

21.2.3 Timer Control/Status Register (TCSR)

TCSR1

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written in bit 7, to clear the flag.

TCSR1 is an 8-bit readable/writable register that performs selection of the WDT1 TCNT input clock, mode, etc.

Only bit 4 is described here. For details of the other bits, see section 14.2.2, Timer Control/Status Register (TCSR).

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

Bit 4—Prescaler Select (PSS): Selects the WDT1 TCNT input clock.

This bit also controls the operation in a power-down mode transition. The operating mode to which a transition is made after execution of a SLEEP instruction is determined in combination with other control bits.

For details, see the description of Clock Select 2 to 0 in section 14.2.2, Timer Control/Status Register (TCSR).

Bit 4

PSS	Description
0	TCNT counts ø-based prescaler (PSM) divided clock pulses When a SLEEP instruction is executed in high-speed mode or medium-speed mode, a transition is made to sleep mode or software standby mode (Initial value)
1	TCNT counts øSUB-based prescaler (PSM) divided clock pulses When a SLEEP instruction is executed in high-speed mode or medium-speed mode, a transition is made to sleep mode, watch mode*, or subactive mode* When a SLEEP instruction is executed in subactive mode, a transition is made to subsleep mode, watch mode, or high-speed mode

Note: * When a transition is made to watch mode or subactive mode, high-speed mode must be set.

21.2.4 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	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	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers that perform module stop mode control.

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

MSTRCRH and MSTPCRL Bits 7 to 0—Module Stop (MSTP 15 to MSTP 0): These bits specify module stop mode. See table 21.3 for the method of selecting on-chip supporting modules.

MSTPCRH, MSTPCRL Bits 7 to 0

MSTP15 to MSTP0	Description	
0	Module stop mode is cleared	(Initial value of MSTP15, MSTP14)
1	Module stop mode is set	(Initial value of MSTP13 to MSTP0)

21.3 Medium-Speed Mode

When the SCK2 to SCK0 bits in SBYCR are set to 1 in high-speed mode, the operating mode changes to medium-speed mode at the end of the bus cycle. In medium-speed mode, the CPU operates on the operating clock (ø/2, ø/4, ø/8, ø/16, or ø/32) specified by the SCK2 to SCK0 bits. The bus master other than the CPU (the DTC) also operates in medium-speed mode. On-chip supporting modules other than the bus masters always operate on the high-speed clock (ø).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if ø/4 is selected as the operating clock, on-chip memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR and the LSON bit in LPWRCR are cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, and the LSON bit in LPWRCR and the PSS bit in TCSR (WDT1) are both cleared to 0, a transition is made to software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is driven low, a transition is made to the reset state, and medium-speed mode is cleared. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 21.2 shows the timing for transition to and clearance of medium-speed mode.

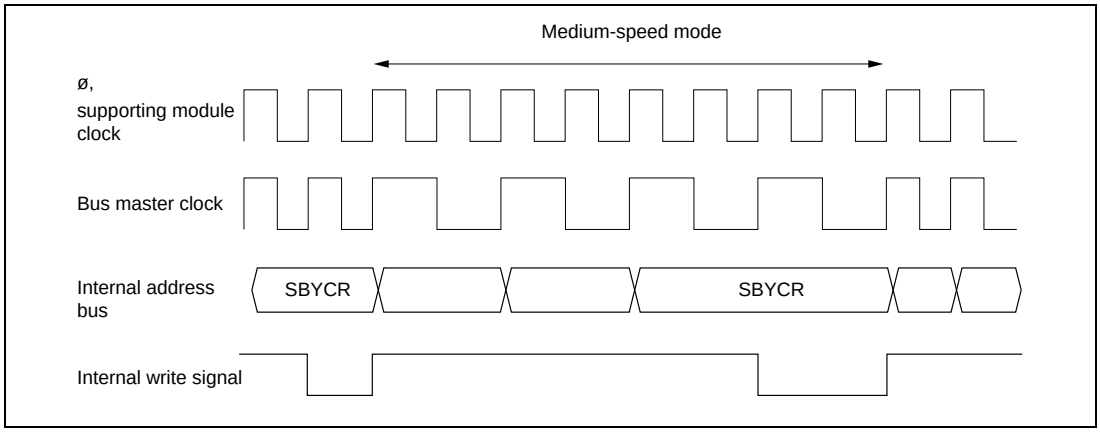


Figure 21.2 Medium-Speed Mode Transition and Clearance Timing

21.4 Sleep Mode

21.4.1 Sleep Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR and the LSON bit in LPWRCR are both cleared to 0, the CPU enters sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other supporting modules do not stop.

21.4.2 Clearing Sleep Mode

Sleep mode is cleared by any interrupt, or with the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

Clearing with an Interrupt: When an interrupt request signal is input, sleep mode is cleared and interrupt exception handling is started. Sleep mode will not be cleared if interrupts are disabled, or if interrupts other than NMI have been masked by the CPU.

Clearing with the $\overline{\text{RES}}$ Pin: When the $\overline{\text{RES}}$ pin is driven low, the reset state is entered. When the $\overline{\text{RES}}$ pin is driven high after the prescribed reset input period, the CPU begins reset exception handling.

Clearing with the $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

21.5 Module Stop Mode

21.5.1 Module Stop Mode

Module stop mode can be set for individual on-chip supporting modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

Table 21.4 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating again at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI, A/D converter, 8-bit PWM module, and 14-bit PWM module, are retained.

After reset release, all modules other than the DTC are in module stop mode.

When an on-chip supporting module is in module stop mode, read/write access to its registers is disabled.

Table 21.4 MSTP Bits and Corresponding On-Chip Supporting Modules

Register	Bit	Module
MSTPCRH	MSTP15	—
	MSTP14*	Data transfer controller (DTC)
	MSTP13	16-bit free-running timer (FRT)
	MSTP12	8-bit timers (TMR0, TMR1)
	MSTP11*	8-bit PWM timer (PWM), 14-bit PWM timer (PWMX)
	MSTP10*	—
	MSTP9	A/D converter
	MSTP8	8-bit timers (TMRX, TMRY), timer connection
MSTPCRL	MSTP7	Serial communication interface 0 (SCI0)
	MSTP6	Serial communication interface 1 (SCI1)
	MSTP5*	—
	MSTP4*	I^2C bus interface (IIC) channel 0 (option)
	MSTP3*	I^2C bus interface (IIC) channel 1 (option)
	MSTP2*	—
	MSTP1*	—
	MSTP0*	—

Note: Bits 10, 5, 2, 1, and 0 can be read or written to, but do not affect operation.
* Must be set to 1 in the H8S/2124 Series.

21.5.2 Usage Note

If there is conflict between DTC module stop mode setting and a DTC bus request, the bus request has priority and the MSTP bit will not be set to 1.

Write 1 to the MSTP bit again after the DTC bus cycle.

When using an H8S/2124 Series MCU, the MSTP bits for nonexistent modules must be set to 1.

21.6 Software Standby Mode

21.6.1 Software Standby Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, the LSON bit in LPWRCR is cleared to 0, and the PSS bit in TCSR (WDT1) is cleared to 0, software standby mode is entered. In this mode, the CPU, on-chip supporting modules, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip supporting modules other than the SCI, PWM, and PWMX, and of the I/O ports, are retained.

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

21.6.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pin $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, or $\overline{\text{IRQ2}}$), or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

Clearing with an Interrupt: When an NMI, IRQ0, IRQ1, or IRQ2 interrupt request signal is input, clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SYSCR, stable clocks are supplied to the entire chip, software standby mode is cleared, and interrupt exception handling is started.

Software standby mode cannot be cleared with an IRQ0, IRQ1, or IRQ2 interrupt if the corresponding enable bit has been cleared to 0 or has been masked by the CPU.

Clearing with the $\overline{\text{RES}}$ Pin: When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire chip. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin goes high, the CPU begins reset exception handling.

Clearing with the $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

21.6.3 Setting Oscillation Settling Time after Clearing Software Standby Mode

Bits STS2 to STS0 in SBYCR should be set as described below.

Using a Crystal Oscillator: Set bits STS2 to STS0 so that the standby time is at least 8 ms (the oscillation settling time).

Table 21.5 shows the standby times for different operating frequencies and settings of bits STS2 to STS0.

Table 21.5 Oscillation Settling Time Settings

STS2	STS1	STS0	Standby Time	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 MHz	Unit
0	0	0	8192 states	0.41	0.51	0.65	0.8	1.0	1.3	2.0	4.1	ms
		1	16384 states	0.82	1.0	1.3	1.6	2.0	2.7	4.1	[8.2]	
	1	0	32768 states	1.6	2.0	2.7	3.3	4.1	5.5	[8.2]	16.4	
		1	65536 states	3.3	4.1	5.5	6.6	[8.2]	[10.9]	16.4	32.8	
1	0	0	131072 states	6.6	[8.2]	[10.9]	[13.1]	16.4	21.8	32.8	65.5	
		1	262144 states	[13.1]	16.4	21.8	26.2	32.8	43.6	65.6	131.2	
	1	0	Reserved	—	—	—	—	—	—	—	—	µs
		1	16 states*	0.8	1.0	1.3	1.6	2.0	2.7	4.0	8.0	

[ ]: Recommended time setting *: Don't care

Note: * This setting must not be used in the flash memory version.

Using an External Clock: Any value can be set. Normally, use of the minimum time is recommended.

21.6.4 Software Standby Mode Application Example

Figure 21.3 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software standby mode is cleared at the rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.

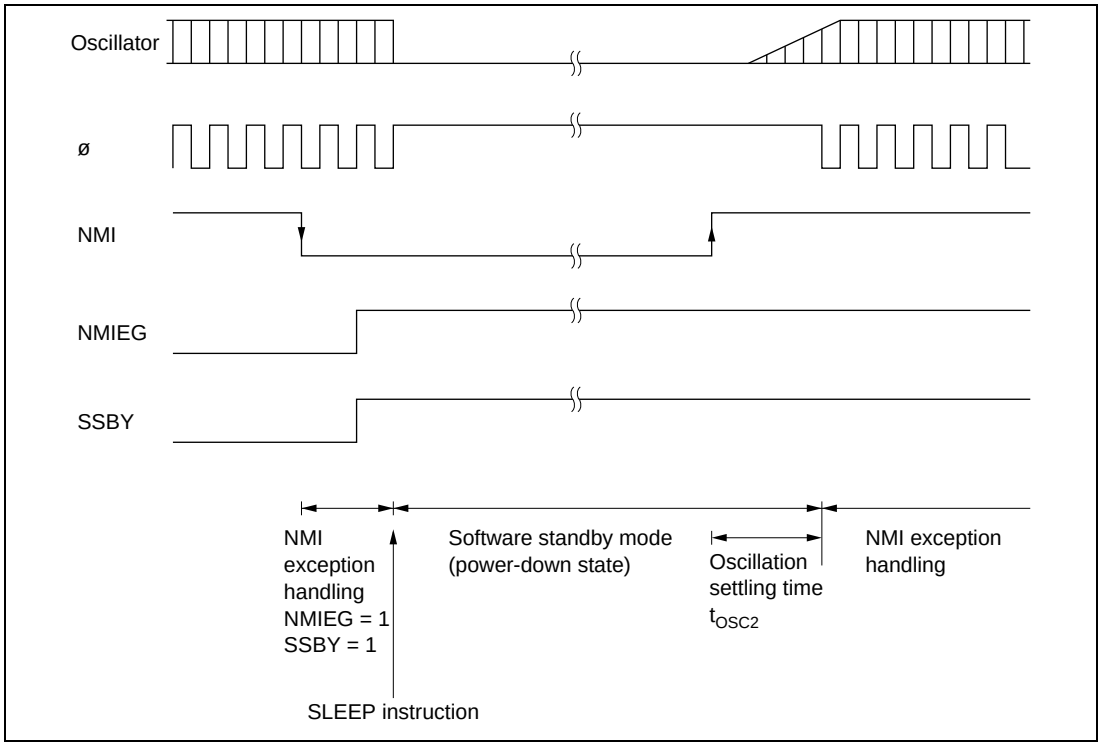


Figure 21.3 Software Standby Mode Application Example

21.6.5 Usage Note

In software standby mode, I/O port states are retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

Current dissipation increases while waiting for oscillation to settle.

21.7 Hardware Standby Mode

21.7.1 Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power dissipation. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD1 and MD0) while the chip is in hardware standby mode.

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillation settles (at least 8 ms—the oscillation settling time—when using a crystal oscillator). When the $\overline{\text{RES}}$ pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

21.7.2 Hardware Standby Mode Timing

Figure 21.4 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high, waiting for the oscillation settling time, then changing the $\overline{\text{RES}}$ pin from low to high.

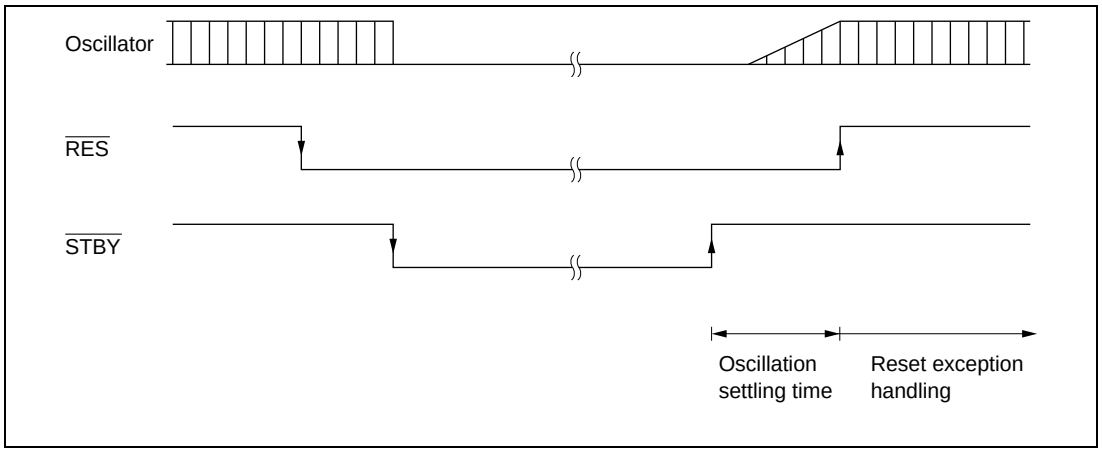


Figure 21.4 Hardware Standby Mode Timing

21.8 Watch Mode

21.8.1 Watch Mode

If a SLEEP instruction is executed in high-speed mode or subactive mode when the SSBY in SBYCR is set to 1, the DTON bit in LPWRCR is cleared to 0, and the PSS bit in TCSR (WDT1) is set to 1, the CPU makes a transition to watch mode.

In this mode, the CPU and all on-chip supporting modules except WDT1 stop. As long as the prescribed voltage is supplied, the contents of some of the CPU's internal registers and on-chip RAM are retained, and I/O ports retain their states prior to the transition.

21.8.2 Clearing Watch Mode

Watch mode is cleared by an interrupt (WOVI1 interrupt, NMI pin, or pin $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, or $\overline{\text{IRQ2}}$), or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

Clearing with an Interrupt: When an interrupt request signal is input, watch mode is cleared and a transition is made to high-speed mode or medium-speed mode if the LSON bit in LPWRCR is cleared to 0, or to subactive mode if the LSON bit is set to 1. When making a

transition to high-speed mode, after the elapse of the time set in bits STS2 to STS0 in SBYCR, stable clocks are supplied to the entire chip, and interrupt exception handling is started.

Watch mode cannot be cleared with an IRQ0, IRQ1, or IRQ2 interrupt if the corresponding enable bit has been cleared to 0, or with an on-chip supporting module interrupt if acceptance of the relevant interrupt has been disabled by the interrupt enable register or masked by the CPU.

See section 21.6.3, Setting Oscillation Settling Time after Clearing Software Standby Mode, for the oscillation settling time setting when making a transition from watch mode to high-speed mode.

Clearing with the $\overline{\text{RES}}$ Pin: See "Clearing with the $\overline{\text{RES}}$ Pin" in section 21.6.2, Clearing Software Standby Mode.

Clearing with the $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

21.9 Subsleep Mode

21.9.1 Subsleep Mode

If a SLEEP instruction is executed in subactive mode when the SSBY in SBYCR is cleared to 0, the LSON bit in LPWRCR is set to 1, and the PSS bit in TCSR (WDT1) is set to 1, the CPU makes a transition to subsleep mode.

In this mode, the CPU and all on-chip supporting modules except TMR0, TMR1, WDT0, and WDT1 stop. As long as the prescribed voltage is supplied, the contents of some of the CPU's internal registers and on-chip RAM are retained, and I/O ports retain their states prior to the transition.

21.9.2 Clearing Subsleep Mode

Subsleep mode is cleared by an interrupt (on-chip supporting module interrupt, NMI pin, or pin $\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$, or $\overline{\text{IRQ2}}$), or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

Clearing with an Interrupt: When an interrupt request signal is input, subsleep mode is cleared and interrupt exception handling is started. Subsleep mode cannot be cleared with an IRQ0 to IRQ2 interrupt if the corresponding enable bit has been cleared to 0, or with an on-chip supporting module interrupt if acceptance of the relevant interrupt has been disabled by the interrupt enable register or masked by the CPU.

Clearing with the $\overline{\text{RES}}$ Pin: See "Clearing with the $\overline{\text{RES}}$ Pin" in section 21.6.2, Clearing Software Standby Mode.

Clearing with the $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode

21.10 Subactive Mode

21.10.1 Subactive Mode

If a SLEEP instruction is executed in high-speed mode when the SSBY bit in SBYCR, the DTON bit in LPWRCR, and the PSS bit in TCSR (WDT1) are all set to 1, the CPU makes a transition to subactive mode. When an interrupt is generated in watch mode, if the LSON bit in LPWRCR is set to 1, a transition is made to subactive mode. When an interrupt is generated in subsleep mode, a transition is made to subactive mode.

In subactive mode, the CPU performs sequential program execution at low speed on the subclock. In this mode, all on-chip supporting modules except TMR0, TMR1, WDT0, and WDT1 stop.

When operating the device in subactive mode, bits SCK2 to SCK0 in SBYCR must all be cleared to 0.

21.10.2 Clearing Subactive Mode

Subsleep mode is cleared by a SLEEP instruction, or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

Clearing with a SLEEP Instruction: When a SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1, the DTON bit in LPWRCR is cleared to 0, and the PSS bit in TCSR (WDT1) is set to 1, subactive mode is cleared and a transition is made to watch mode. When a SLEEP instruction is executed while the SSBY bit in SBYCR is cleared to 0, the LSON bit in LPWRCR is set to 1, and the PSS bit in TCSR (WDT1) is set to 1, a transition is made to subsleep mode. When a SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1, the DTON bit is set to 1 and the LSON bit is cleared to 0 in LPWRCR, and the PSS bit in TCSR (WDT1) is set to 1, a transition is made directly to high-speed mode.

Fort details of direct transition, see section 21.11, Direct Transition.

Clearing with the $\overline{\text{RES}}$ Pin: See “Clearing with the $\overline{\text{RES}}$ Pin” in section 21.6.2, Clearing Software Standby Mode.

Clearing with the $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode

21.11 Direct Transition

21.11.1 Overview of Direct Transition

There are three operating modes in which the CPU executes programs: high-speed mode, medium-speed mode, and subactive mode. A transition between high-speed mode and subactive mode without halting the program is called a direct transition. A direct transition can be carried out by setting the DTON bit in LPWRCR to 1 and executing a SLEEP instruction. After the transition, direct transition interrupt exception handling is started.

Direct Transition from High-Speed Mode to Subactive Mode: If a SLEEP instruction is executed in high-speed mode while the SSBY bit in SBYCR, the LSON bit and DTON bit in LPWRCR, and the PSS bit in TSCR (WDT1) are all set to 1, a transition is made to subactive mode.

Direct Transition from Subactive Mode to High-Speed Mode: If a SLEEP instruction is executed in subactive mode while the SSBY bit in SBYCR is set to 1, the LSON bit is cleared to 0 and the DTON bit is set to 1 in LPWRCR, and the PSS bit in TSCR (WDT1) is set to 1, after the elapse of the time set in bits STS2 to STS0 in SBYCR, a transition is made to directly to high-speed mode.

Section 22 Electrical Characteristics

22.1 Absolute Maximum Ratings

Table 22.1 lists the absolute maximum ratings.

Table 22.1 Absolute Maximum Ratings **– Preliminary –**

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Input voltage (except ports 6, and 7)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (CIN input not selected for port 6)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (CIN input selected for port 6)	V_{in}	Lower voltage of –0.3 to V_{CC} +0.3 and AV_{CC} +0.3	V
Input voltage (port 7)	V_{in}	–0.3 to AV_{CC} + 0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75	°C
		Wide-range specifications: –40 to +85	°C
Operating temperature (Flash memory programming/ erasing)	T_{opr}	Regular specifications: 0 to +75	°C
		Wide-range specifications: 0 to +85	
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

22.2 DC Characteristics

Table 22.2 lists the DC characteristics. Table 22.3 lists the permissible output currents.

Table 22.2 DC Characteristics (1) **– Preliminary –**

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC}*[1] = 5.0 V ± 10%, V_{SS} = AV_{SS}*[1] = 0 V,
T_a = –20 to +75°C*[8] (regular specifications),
T_a = –40 to +85°C*[8] (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	P67 to P60*[2, 5], $\overline{IRQ2}$ to $\overline{IRQ0}$*[3] (1)	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, MD1, MD0 (2)	V_{IH}	V_{CC} – 0.7	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 7		2.0	—	AV_{CC} +0.3	V	
	Input pins except (1) and (2) above		2.0	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD1, MD0 (3)	V_{IL}	–0.3	—	0.5	V	
	NMI, EXTAL, input pins except (1) and (3) above		–0.3	—	0.8	V	
Output high voltage	All output pins (except P47, and P52*[4])	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1 mA
	P47, P52*[4]		2.5	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	Ports 1 to 3		—	—	1.0	V	I_{OL} = 10 mA
Input leakage current	$\overline{RES}$	$\Omega I_{in} \Omega$	—	—	10.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{STBY}$, NMI, MD1, MD0		—	—	1.0	µA	
	Port 7		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5 V

Table 22.2 DC Characteristics (1) (cont) **– Preliminary –**

Conditions: V_{CC} = 5.0 V ± 10%, AV_{CC}*[1] = 5.0 V ± 10%, V_{SS} = AV_{SS}*[1] = 0 V,
T_a = –20 to +75°C*[8] (regular specifications),
T_a = –40 to +85°C*[8] (wide-range specifications)

Item			Symbol	Min	Typ	Max	Unit	Test Conditions
Three-state leakage current (off state)	Ports 1 to 6		$\Omega I_{TSI}\Omega$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1 to 3		$-I_P$	50	—	300	µA	V_{in} = 0 V
Input capacitance	$\overline{RES}$	(4)	C_{in}	—	—	80	pF	V_{in} = 0 V f = 1 MHz
	NMI			—	—	50	pF	T_a = 25°C
	P52, P47, P24, P23			—	—	20	pF	
	Input pins except (4) above			—	—	15	pF	
Current dissipation*[6]	Normal operation		I_{CC}	—	70	90	mA	f = 20 MHz
	Sleep mode			—	55	75	mA	f = 20 MHz
	Standby mode*[7]			—	0.01	5.0	µA	$T_a \leq$ 50°C
				—	—	20.0	µA	50°C < T_a
Analog power supply current	During A/D conversion		AI_{CC}	—	1.5	3.0	mA	
	Idle			—	0.01	5.0	µA	AV_{CC} = 2.0 V to 5.5 V
Analog power supply voltage*[1]			AV_{CC}	4.5	—	5.5	V	Operating
				2.0	—	5.5	V	Idle/not used
RAM standby voltage			V_{RAM}	2.0	—	—	V	

Notes:
1. Do not leave the AVCC, and AVSS pins open even if the A/D converter is not used.
   Even if the A/D converter is not used, apply a value in the range 2.0 V to 5.5 V to AVCC by connection to the power supply (V_{CC}), or some other method.
2. P67 to P60 include supporting module inputs multiplexed on those pins.
3. $\overline{IRQ2}$ includes the $\overline{ADTRG}$ signal multiplexed on that pin.
4. In the H8S/2128 Series, P52/SCK0/SCL0 and P47/SDA0 are NMOS push-pull outputs.
   An external pull-up resistor is necessary to provide high-level output from SCL0 and SDA0 (ICE = 1).
   In the H8S/2128 Series, P52/SCK0 and P47 (ICE = 0) high levels are driven by NMOS.
5. The upper limit of the port 6 applied voltage is V_{CC} + 0.3 V when CIN input is not selected, and the lower of V_{CC} + 0.3 V and AV_{CC} + 0.3 V when CIN input is selected. When a pin is in output mode, the output voltage is equivalent to the applied voltage.
6. Current dissipation values are for V_{IH} min = V_{CC} – 0.5 V and V_{IL} max = 0.5 V with all output pins unloaded and the on-chip pull-up MOSs in the off state.
7. The values are for $V_{RAM} \leq V_{CC} <$ 4.5 V, V_{IH} min = $V_{CC} \times$ 0.9, and V_{IL} max = 0.3 V.
8. For flash memory program/erase operations, the applicable range is T_a = 0 to +75°C (regular specifications) or T_a = 0 to +85°C (wide-range specifications).

Table 22.2 DC Characteristics (2) **– Preliminary –**

Conditions: V_{CC} = 4.0 V to 5.5 V*[8], AV_{CC}*[1] = 4.0 V to 5.5 V, V_{SS} = AV_{SS}*[1] = 0 V,
T_a = –20 to +75°C*[8] (regular specifications),
T_a = –40 to +85°C*[8] (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	P67 to P60*[2, *5], $\overline{IRQ2}$ to $\overline{IRQ0}$*[3] (1)	V_T^-	1.0	—	—	V	V_{CC} = 4.5 V to 5.5 V
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
		V_T^-	0.8	—	—	V	V_{CC} < 4.5 V
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.3	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD1, MD0 (2)	V_{IH}	V_{CC} – 0.7	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 7		2.0	—	AV_{CC} +0.3	V	
	Input pins except (1) and (2) above		2.0	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD1, MD0 (3)	V_{IL}	–0.3	—	0.5	V	
	NMI, EXTAL, input pins except (1) and (3) above		–0.3	—	0.8	V	
Output high voltage	All output pins (except P47, and P52*[4])	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1 mA, V_{CC}= 4.5 V to 5.5 V
			3.0	—	—	V	I_{OH} = –1 mA, V_{CC} < 4.5 V
	P47, P52*[4]		2.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	Ports 1 to 3		—	—	1.0	V	I_{OL} = 10 mA
Input leakage current	$\overline{RES}$	$\Omega I_{in} \Omega$	—	—	10.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{STBY}$, NMI, MD1, MD0		—	—	1.0	µA	
	Port 7		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5 V

Table 22.2 DC Characteristics (2) (cont) **– Preliminary –**

Conditions: V_{CC} = 4.0 V to 5.5 V*[8], AV_{CC}*[1] = 4.0 V to 5.5 V, V_{SS} = AV_{SS}*[1] = 0 V, T_a = –20 to +75°C*[8] (regular specifications), T_a = –40 to +85°C*[8] (wide-range specifications)

Item			Symbol	Min	Typ	Max	Unit	Test Conditions
Three-state leakage current (off state)	Ports 1 to 6		$\Omega I_{TSI}\Omega$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1 to 3		$-I_P$	50	—	300	µA	V_{in} = 0 V, V_{CC} = 4.5 V to 5.5 V
				30	—	200	µA	V_{in} = 0 V, V_{CC} < 4.5 V
Input capacitance	$\overline{RES}$	(4)	C_{in}	—	—	80	pF	V_{in} = 0 V, f = 1 MHz,
	NMI			—	—	50	pF	T_a = 25°C
	P52, P47, P24, P23			—	—	20	pF	
	Input pins except (4) above			—	—	15	pF	
Current dissipation*[6]	Normal operation		I_{CC}	—	55	75	mA	f = 16 MHz
	Sleep mode			—	42	62	mA	f = 16 MHz
	Standby mode*[7]			—	0.01	5.0	µA	$T_a \leq 50°C$
				—	—	20.0	µA	50°C < T_a
Analog power supply current	During A/D conversion		AI_{CC}	—	1.5	3.0	mA	
	Idle			—	0.01	5.0	µA	AV_{CC} = 2.0 V to 5.5 V
Analog power supply voltage*[1]			AV_{CC}	4.0	—	5.5	V	Operating
				2.0	—	5.5	V	Idle/not used
RAM standby voltage			V_{RAM}	2.0	—	—	V	

Notes:
1. Do not leave the AVCC, and AVSS pins open even if the A/D converter is not used. Even if the A/D converter is not used, apply a value in the range 2.0 V to 5.5 V to AVCC by connection to the power supply (V_{CC}), or some other method.
2. P67 to P60 include supporting module inputs multiplexed on those pins.
3. $\overline{IRQ2}$ includes the $\overline{ADTRG}$ signal multiplexed on that pin.
4. In the H8S/2128 Series, P52/SCK0/SCL0 and P47/SDA0 are NMOS push-pull outputs. An external pull-up resistor is necessary to provide high-level output from SCL0 and SDA0 (ICE = 1). In the H8S/2128 Series, P52/SCK0 and P47 (ICE = 0) high levels are driven by NMOS.
5. The upper limit of the port 6 applied voltage is V_{CC} + 0.3 V when CIN input is not selected, and the lower of V_{CC} + 0.3 V and AV_{CC} + 0.3 V when CIN input is selected. When a pin is in output mode, the output voltage is equivalent to the applied voltage.
6. Current dissipation values are for V_{IH} min = V_{CC} – 0.5 V and V_{IL} max = 0.5 V with all output pins unloaded and the on-chip pull-up MOSs in the off state.
7. The values are for $V_{RAM} \leq V_{CC} < 4.0$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.
8. For flash memory program/erase operations, the applicable ranges are V_{CC} = 4.5 V to 5.5 V and T_a = 0 to +75°C (regular specifications) or T_a = 0 to +85°C (wide-range specifications).

Table 22.2 DC Characteristics (3) **– Preliminary –**

Conditions (Mask ROM version): V_{CC} = 2.7 V to 5.5 V, AV_{CC}*[1] = 2.7 V to 5.5 V, V_{SS} = AV_{SS}*[1] = 0 V, T_a = –20 to +75°C
(Flash memory version): V_{CC} = 3.0 V to 5.5 V, AV_{CC}*[1] = 3.0 V to 5.5 V, V_{SS} = AV_{SS}*[1] = 0 V, T_a = –20 to +75°C*[8]

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	P67 to P60*[2, 5], (1) $\overline{IRQ2}$ to $\overline{IRQ0}$*[3]	V_T^-	$V_{CC} \times 0.2$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, (2) NMI, MD1, MD0	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} +0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
	Port 7		$V_{CC} \times 0.7$	—	AV_{CC} +0.3	V	
	Input pins except (1) and (2) above		$V_{CC} \times 0.7$	—	V_{CC} +0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, (3) MD1, MD0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
	NMI, EXTAL, input pins except (1) and (3) above		–0.3	—	$V_{CC} \times 0.2$	V	V_{CC} < 4.0 V
					0.8	V	V_{CC} = 4.0 V to 5.5 V
Output high voltage	All output pins (except P47, and P52*[4])	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1 mA (V_{CC} < 4.0 V)
	P47, P52*[4]		1.0	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	Ports 1 to 3		—	—	1.0	V	I_{OL} = 5 mA (V_{CC} < 4.0 V), I_{OL} = 10 mA (4.0 V ≤ V_{CC} ≤ 5.5 V)
Input leakage current	$\overline{RES}$	ΩI$_{in}$Ω	—	—	10.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{STBY}$, NMI, MD1, MD0		—	—	1.0	µA	
	Port 7		—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5 V

Table 22.2 DC Characteristics (3) (cont) **– Preliminary –**

Conditions (Mask ROM version): V_{CC} = 2.7 V to 5.5 V, AV_{CC}*[1] = 2.7 V to 5.5 V, V_{SS} = AV_{SS}*[1] = 0 V, T_a = –20 to +75°C
(Flash memory version): V_{CC} = 3.0 V to 5.5 V, AV_{CC}*[1] = 3.0 V to 5.5 V, V_{SS} = AV_{SS}*[1] = 0 V, T_a = –20 to +75°C*[8]

Item			Symbol	Min	Typ	Max	Unit	Test Conditions
Three-state leakage current (off state)	Ports 1 to 6		$\Omega I_{TSI}\Omega$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1 to 3		$-I_P$	10	—	150	µA	V_{in} = 0 V, V_{CC} = 2.7 V to 3.6 V
Input capacitance	$\overline{RES}$	(4)	C_{in}	—	—	80	pF	V_{in} = 0 V, f = 1 MHz, Ta = 25°C
	NMI			—	—	50	pF	
	P52, P47, P24, P23			—	—	20	pF	
	Input pins except (4) above			—	—	15	pF	
Current dissipation*[6]	Normal operation		I_{CC}	—	40	52	mA	f = 10 MHz
	Sleep mode			—	30	42	mA	f = 10 MHz
	Standby mode*[7]			—	0.01	5.0	µA	$T_a \leq 50°C$
				—	—	20.0	µA	$50°C < T_a$
Analog power supply current	During A/D conversion		AI_{CC}	—	1.5	3.0	mA	
	Idle			—	0.01	5.0	µA	AV_{CC} = 2.0 V to 5.5 V
Analog power supply voltage*[1]			AV_{CC}	2.7	—	5.5	V	Operating
				2.0	—	5.5	V	Idle/not used
RAM standby voltage			V_{RAM}	2.0	—	—	V	

Notes:
1. Do not leave the AVCC, and AVSS pins open even if the A/D converter is not used. Even if the A/D converter is not used, apply a value in the range 2.0 V to 5.5 V to AVCC by connection to the power supply (V_{CC}), or some other method.
2. P67 to P60 include supporting module inputs multiplexed on those pins.
3. $\overline{IRQ2}$ includes the $\overline{ADTRG}$ signal multiplexed on that pin.
4. In the H8S/2128 Series, P52/SCK0/SCL0 and P47/SDA0 are NMOS push-pull outputs. An external pull-up resistor is necessary to provide high-level output from SCL0 and SDA0 (ICE = 1). In the H8S/2128 Series, P52/SCK0 and P47 (ICE = 0) high levels are driven by NMOS.
5. The upper limit of the port 6 applied voltage is V_{CC} + 0.3 V when CIN input is not selected, and the lower of V_{CC} + 0.3 V and AV_{CC} + 0.3 V when CIN input is selected. When a pin is in output mode, the output voltage is equivalent to the applied voltage.
6. Current dissipation values are for V_{IH} min = V_{CC} – 0.5 V and V_{IL} max = 0.5 V with all output pins unloaded and the on-chip pull-up MOSs in the off state.
7. The values are for $V_{RAM} \leq V_{CC} < 2.7$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.
8. For flash memory program/erase operations, the applicable range is V_{CC} = 3.0 V to 3.6 V and T_a = 0 to +75°C.

Table 22.3 Permissible Output Currents – Preliminary –

Conditions: V_{CC} = 4.0 to 5.5 V, V_{SS} = 0 V, Ta = –20 to +75°C (regular specifications), Ta = –40 to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	SCL1, SCL0, SDA1, SDA0	I_{OL}	—	—	20	mA
	Ports 1, 2, 3		—	—	10	mA
	Other output pins		—	—	2	mA
Permissible output low current (total)	Total of ports 1, 2, and 3	$\sum I_{OL}$	—	—	80	mA
	Total of all output pins, including the above		—	—	120	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	mA
Permissible output high current (total)	Total of all output pins	$\sum -I_{OH}$	—	—	40	mA

Notes: 1. To protect chip reliability, do not exceed the output current values in table 22.3.
2. When driving a Darlington pair or LED, always insert a current-limiting resistor in the output line, as show in figures 22.1 and 22.2.

Table 22.3 Permissible Output Currents (cont) – Preliminary –

Conditions: V_{CC} = 2.7 to 5.5 V, V_{SS} = 0 V, Ta = –20 to +75°C

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	SCL1, SCL0, SDA1, SDA0	I_{OL}	—	—	10	mA
	Ports 1, 2, 3		—	—	2	mA
	Other output pins		—	—	1	mA
Permissible output low current (total)	Total of ports 1, 2, and 3	$\sum I_{OL}$	—	—	40	mA
	Total of all output pins, including the above		—	—	60	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	mA
Permissible output high current (total)	Total of all output pins	$\sum -I_{OH}$	—	—	30	mA

Notes: 1. To protect chip reliability, do not exceed the output current values in table 22.3.
2. When driving a Darlington pair or LED, always insert a current-limiting resistor in the output line, as show in figures 22.1 and 22.2.

Table 22.4 Bus Drive Characteristics **– Preliminary –**

Conditions: V_{CC} = 2.7 to 5.5 V, V_{SS} = 0 V, Ta = –20 to +75°C

Applicable Pins: SCL1, SCL0, SDA1, SDA0 (bus drive function selected)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	V_T^-	$V_{CC} \times 0.3$	—	—	V	V_{CC} = 2.7 V to 5.5 V
	V_T^+	—	—	$V_{CC} \times 0.7$		V_{CC} = 2.7 V to 5.5 V
	$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—		V_{CC} = 2.7 V to 5.5 V
Input high voltage	V_{IH}	$V_{CC} \times 0.7$	—	V_{CC} + 0.5	V	V_{CC} = 2.7 V to 5.5 V
Input low voltage	V_{IL}	–0.5	—	$V_{CC} \times 0.3$		V_{CC} = 2.7 V to 5.5 V
Output low voltage	V_{OL}	—	—	0.8	V	I_{OL} = 16 mA, V_{CC} = 4.5 V to 5.5 V
		—	—	0.5		I_{OL} = 8 mA
		—	—	0.4		I_{OL} = 3 mA
Input capacitance	C_{in}	—	—	20	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
Three-state leakage current (off state)	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
SCL, SDA output fall time	t_{Of}	20 + 0.1Cb	—	250	ns	V_{CC} = 2.7 V to 5.5 V

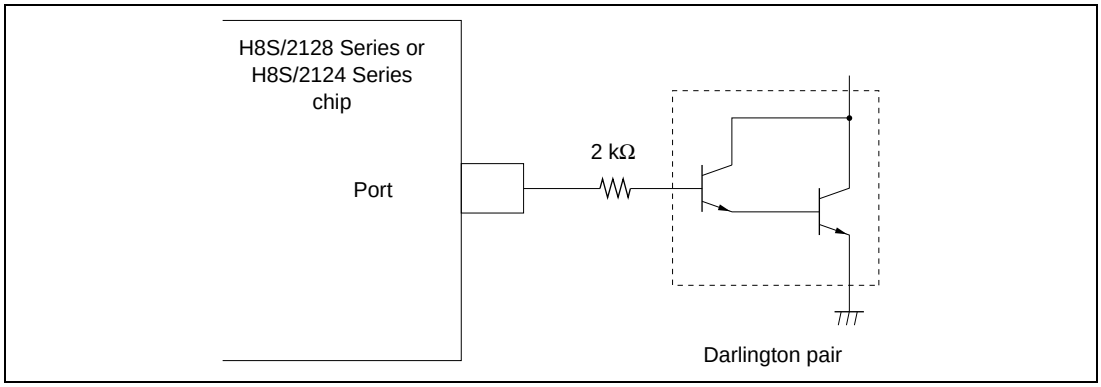


Figure 22.1 Darlington Pair Drive Circuit (Example)

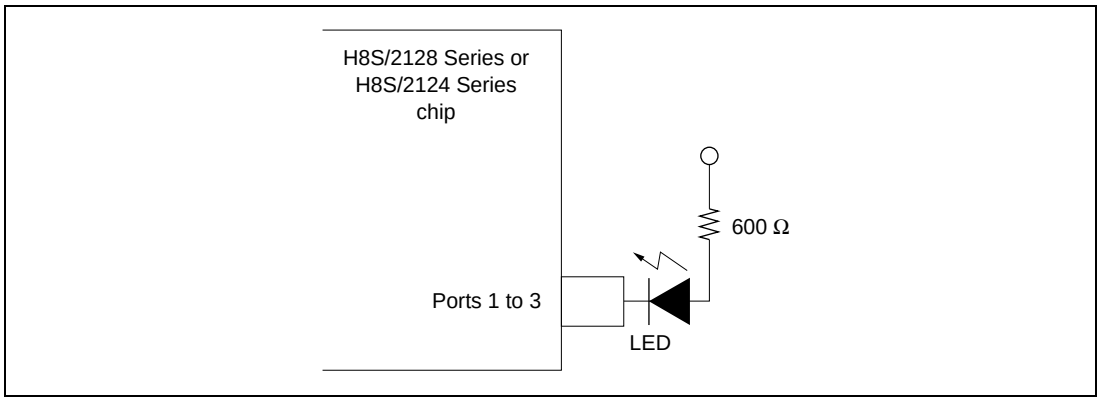


Figure 22.2 LED Drive Circuit (Example)

22.3 AC Characteristics

Figure 22.3 shows the test conditions for the AC characteristics.

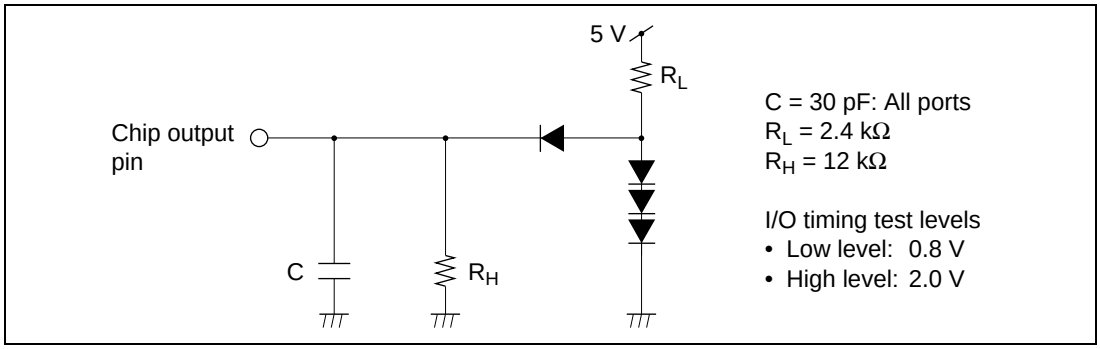


Figure 22.3 Output Load Circuit

22.3.1 Clock Timing

Table 22.5 shows the clock timing. The clock timing specified here covers clock (ø) output and clock pulse generator (crystal) and external clock input (EXTAL pin) oscillation settling times. For details of external clock input (EXTAL pin and EXCL pin) timing, see section 20, Clock Pulse Generator.

Table 22.5 Clock Timing **– Preliminary –**

Condition A: V_{CC} = 5.0 V ± 10%, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)
Condition B: V_{CC} = 4.0 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)
Condition C: V_{CC} = 2.7 V to 5.5 V*, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, T_a = –20 to +75°C

		Condition A		Condition B		Condition C			
		20 MHz		16 MHz		10 MHz			
Item	Symbol	Min	Max	Min	Max	Min	Max	Unit	Test Conditions
Clock cycle time	t_{cyc}	50	500	62.5	500	100	500	ns	Figure 22.4
Clock high pulse width	t_{CH}	17	—	20	—	30	—	ns	Figure 22.4
Clock low pulse width	t_{CL}	17	—	20	—	30	—	ns	
Clock rise time	t_{Cr}	—	8	—	10	—	20	ns	
Clock fall time	t_{Cf}	—	8	—	10	—	20	ns	
Oscillation settling time at reset (crystal)	t_{OSC1}	10	—	10	—	20	—	ms	Figure 22.5 Figure 22.6
Oscillation settling time in software standby (crystal)	t_{OSC2}	8	—	8	—	8	—	ms	
External clock output stabilization delay time	t_{DEXT}	500	—	500	—	500	—	µs	

Note: * For the low-voltage F-ZTAT version, V_{CC} = 3.0 V to 5.5 V.

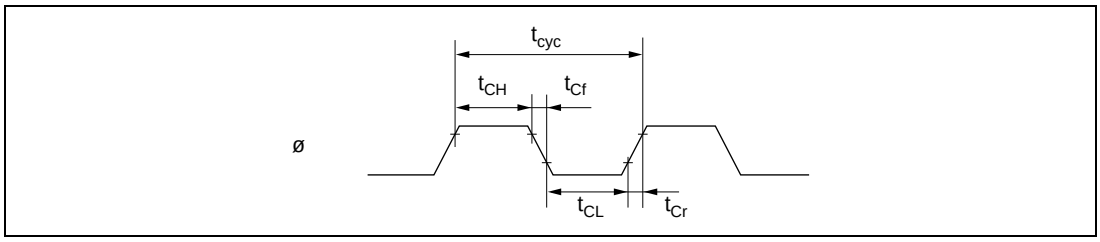


Figure 22.4 System Clock Timing

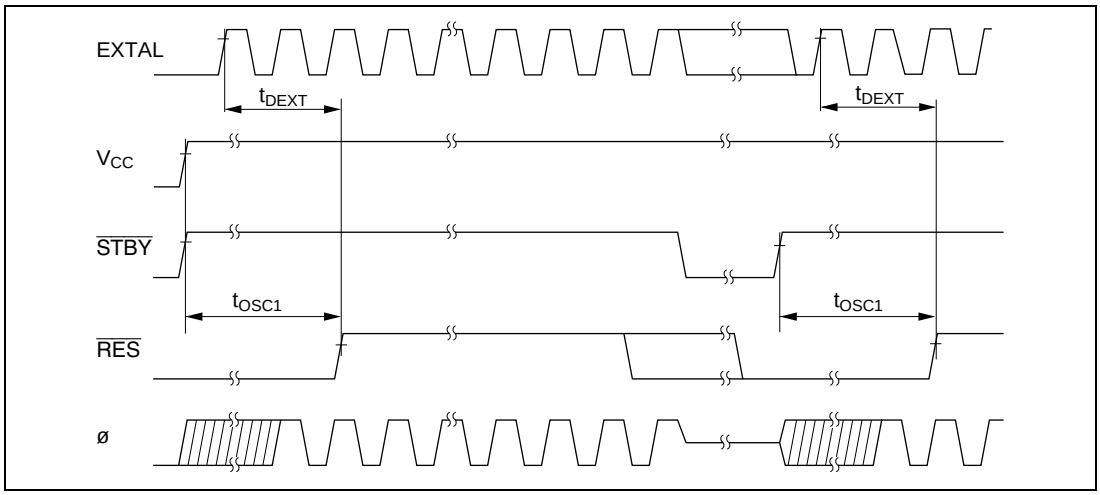


Figure 22.5 Oscillation Settling Timing

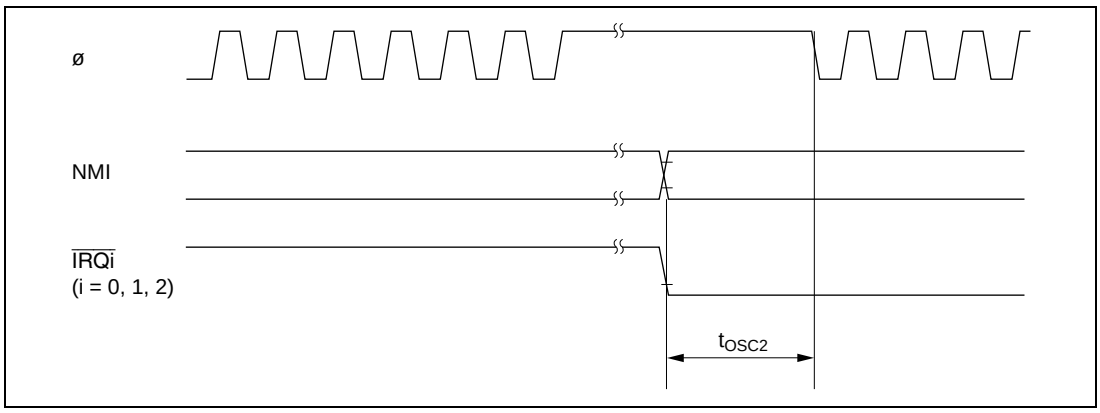


Figure 22.6 Oscillation Setting Timing (Exiting Software Standby Mode)

22.3.2 Control Signal Timing

Table 22.6 shows the control signal timing. The only external interrupts that can operate on the subclock (ø = 32.768 kHz) are NMI and IRQ0, 1, and IRQ2.

Table 22.6 Control Signal Timing – Preliminary –

Condition A: V_{CC} = 5.0 V ± 10%, V_{SS} = 0 V, ø = 32.768 kHz, 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)
Condition B: V_{CC} = 4.0 V to 5.5 V, V_{SS} = 0 V, ø = 32.768 kHz, 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)
Condition C: V_{CC} = 2.7 V to 5.5 V*, V_{SS} = 0 V, ø = 32.768 kHz, 2 MHz to maximum operating frequency, T_a = –20 to +75°C

		Condition A		Condition B		Condition C			
		20 MHz		16 MHz		10 MHz			
Item	Symbol	Min	Max	Min	Max	Min	Max	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	200	—	300	—	ns	Figure 22.7
$\overline{RES}$ pulse width	t_{RESW}	20	—	20	—	20	—	t_{cyc}	
NMI setup time (NMI)	t_{NMIS}	150	—	150	—	250	—	ns	Figure 22.8
NMI hold time (NMI)	t_{NMIH}	10	—	10	—	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	200	—	200	—	ns	
IRQ setup time ($\overline{IRQ2}$ to $\overline{IRQ0}$)	t_{IRQS}	150	—	150	—	250	—	ns	
IRQ hold time ($\overline{IRQ2}$ to $\overline{IRQ0}$)	t_{IRQH}	10	—	10	—	10	—	ns	
IRQ pulse width ($\overline{IRQ2}$ to $\overline{IRQ0}$) (exiting software standby mode)	t_{IRQW}	200	—	200	—	200	—	ns	

Note: * For the low-voltage F-ZTAT version, V_{CC} = 3.0 V to 5.5 V.

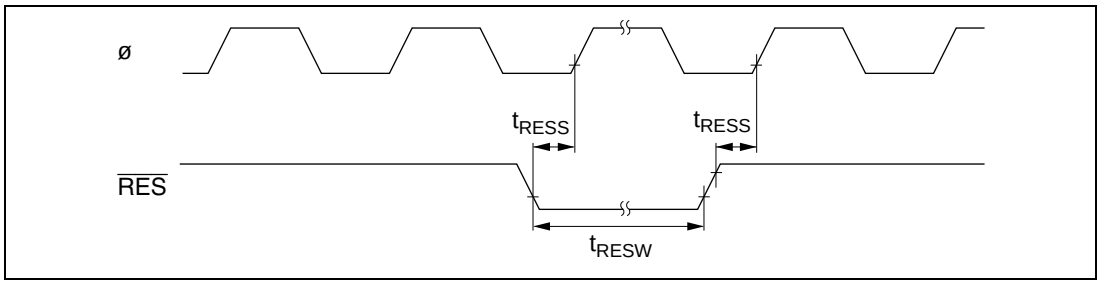


Figure 22.7 Reset Input Timing

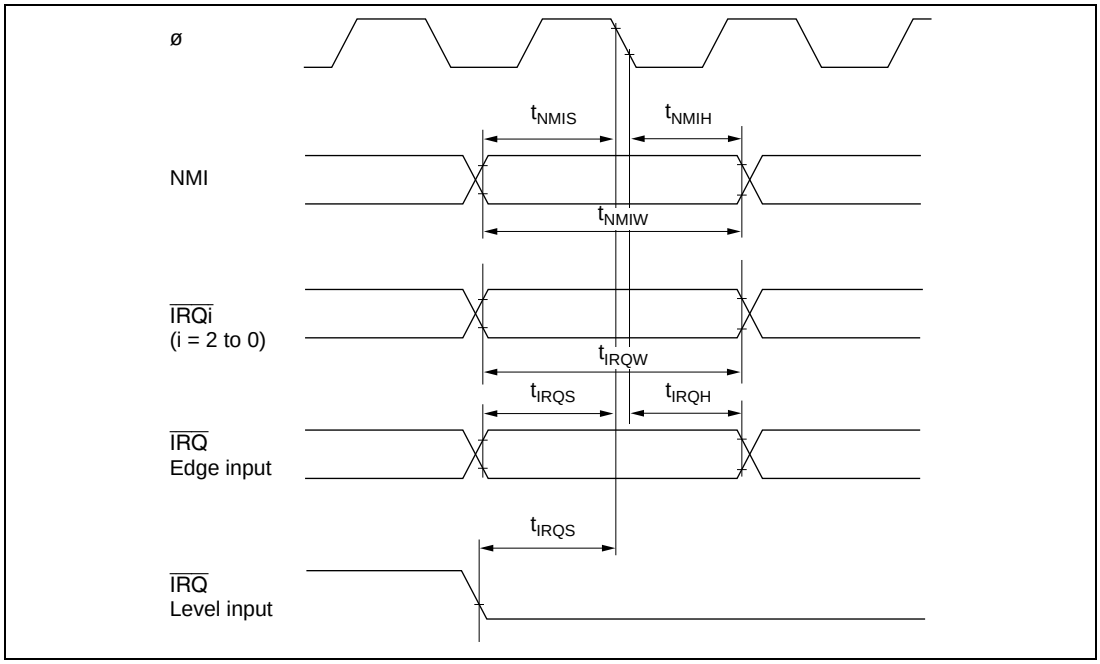


Figure 22.8 Interrupt Input Timing

22.3.3 Bus Timing

Table 22.7 shows the bus timing. Operation in external expansion mode is not guaranteed when operating on the subclock (ø = 32.768 kHz).

Table 22.7 Bus Timing **– Preliminary –**

Condition A: V_{CC} = 5.0 V ± 10%, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)
Condition B: V_{CC} = 4.0 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)
Condition C: V_{CC} = 2.7 V to 5.5 V*, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, T_a = –20 to +75°C

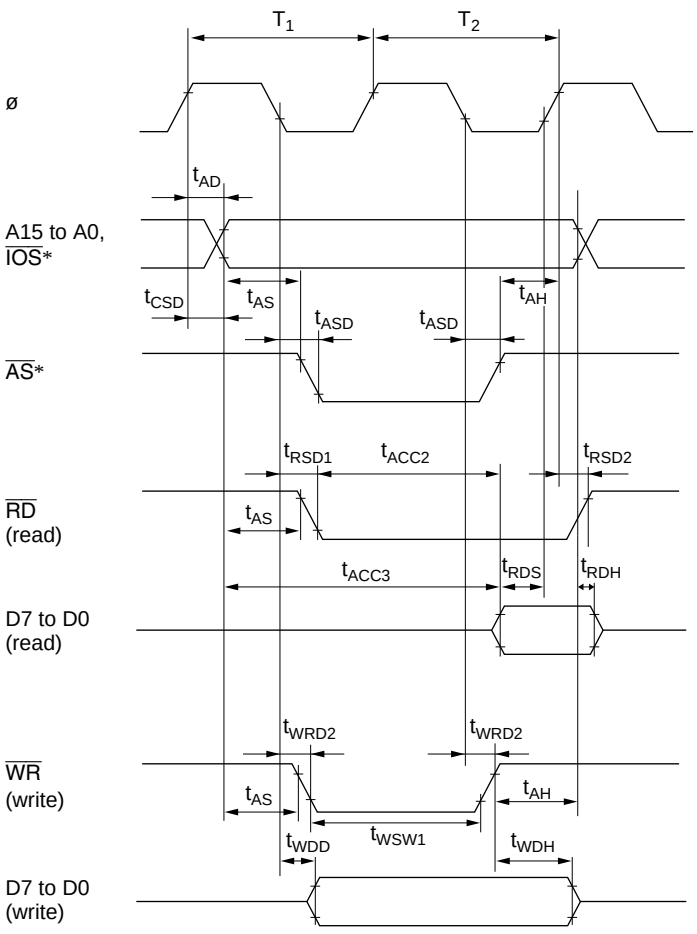
		Condition A		Condition B		Condition C			
		20 MHz		16 MHz		10 MHz			
Item	Symbol	Min	Max	Min	Max	Min	Max	Unit	Test Conditions
Address delay time	t_{AD}	—	20	—	30	—	40	ns	Figure 22.9 to figure 22.13
Address setup time	t_{AS}	0.5 × t_{cyc} – 15	—	0.5 × t_{cyc} – 20	—	0.5 × t_{cyc} – 30	—	ns	
Address hold time	t_{AH}	0.5 × t_{cyc} – 10	—	0.5 × t_{cyc} – 15	—	0.5 × t_{cyc} – 20	—	ns	
$\overline{CS}$ delay time ($\overline{IOS}$)	t_{CSD}	—	20	—	30	—	40	ns	
$\overline{AS}$ delay time	t_{ASD}	—	30	—	45	—	60	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	30	—	45	—	60	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	30	—	45	—	60	ns	
Read data setup time	t_{RDS}	15	—	20	—	35	—	ns	
Read data hold time	t_{RDH}	0	—	0	—	0	—	ns	
Read data access time 1	t_{ACC1}	—	1.0 × t_{cyc} – 30	—	1.0 × t_{cyc} – 40	—	1.0 × t_{cyc} – 60	ns	
Read data access time 2	t_{ACC2}	—	1.5 × t_{cyc} – 25	—	1.5 × t_{cyc} – 35	—	1.5 × t_{cyc} – 50	ns	

Table 22.7 Bus Timing (cont) **– Preliminary –**

Condition A: V_{CC} = 5.0 V ± 10%, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)
Condition B: V_{CC} = 4.0 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)
Condition C: V_{CC} = 2.7 V to 5.5 V*, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, T_a = –20 to +75°C

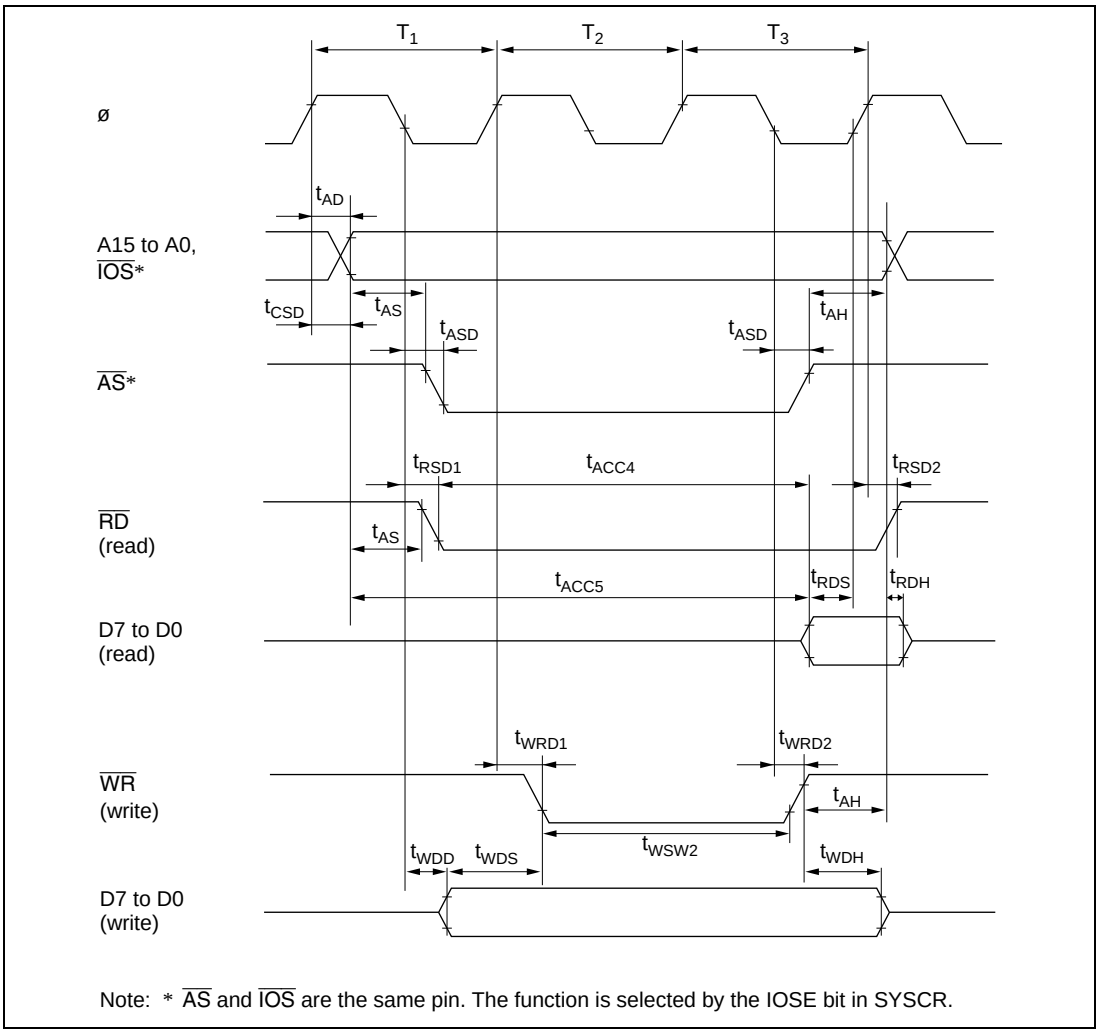
		Condition A		Condition B		Condition C			
		20 MHz		16 MHz		10 MHz			Test
Item	Symbol	Min	Max	Min	Max	Min	Max	Unit	Conditions
Read data access time 3	t_{ACC3}	—	2.0 × t_{cyc} – 30	—	2.0 × t_{cyc} – 40	—	2.0 × t_{cyc} – 60	ns	Figure 22.9 to figure 22.13
Read data access time 4	t_{ACC4}	—	2.5 × t_{cyc} – 25	—	2.5 × t_{cyc} – 35	—	2.5 × t_{cyc} – 50	ns	
Read data access time 5	t_{ACC5}	—	3.0 × t_{cyc} – 30	—	3.0 × t_{cyc} – 40	—	3.0 × t_{cyc} – 60	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	30	—	45	—	60	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	30	—	45	—	60	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	1.0 × t_{cyc} – 20	—	1.0 × t_{cyc} – 30	—	1.0× t_{cyc} – 40	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	1.5 × t_{cyc} – 20	—	1.5 × t_{cyc} – 30	—	1.5 × t_{cyc} – 40	—	ns	
Write data delay time	t_{WDD}	—	30	—	45	—	60	ns	
Write data setup time	t_{WDS}	0	—	0	—	0	—	ns	
Write data hold time	t_{WDH}	10	—	15	—	20	—	ns	
$\overline{WAIT}$ setup time	t_{WTS}	30	—	45	—	60	—	ns	
$\overline{WAIT}$ hold time	t_{WTH}	5	—	5	—	10	—	ns	

Note: * For the low-voltage F-ZTAT version, V_{CC} = 3.0 V to 5.5 V.



Note: * AS and IOS are the same pin. The function is selected by the IOSE bit in SYSCR.

Figure 22.9 Basic Bus Timing (Two-State Access)



Note: * AS and IOS are the same pin. The function is selected by the IOSE bit in SYSCR.

Figure 22.10 Basic Bus Timing (Three-State Access)

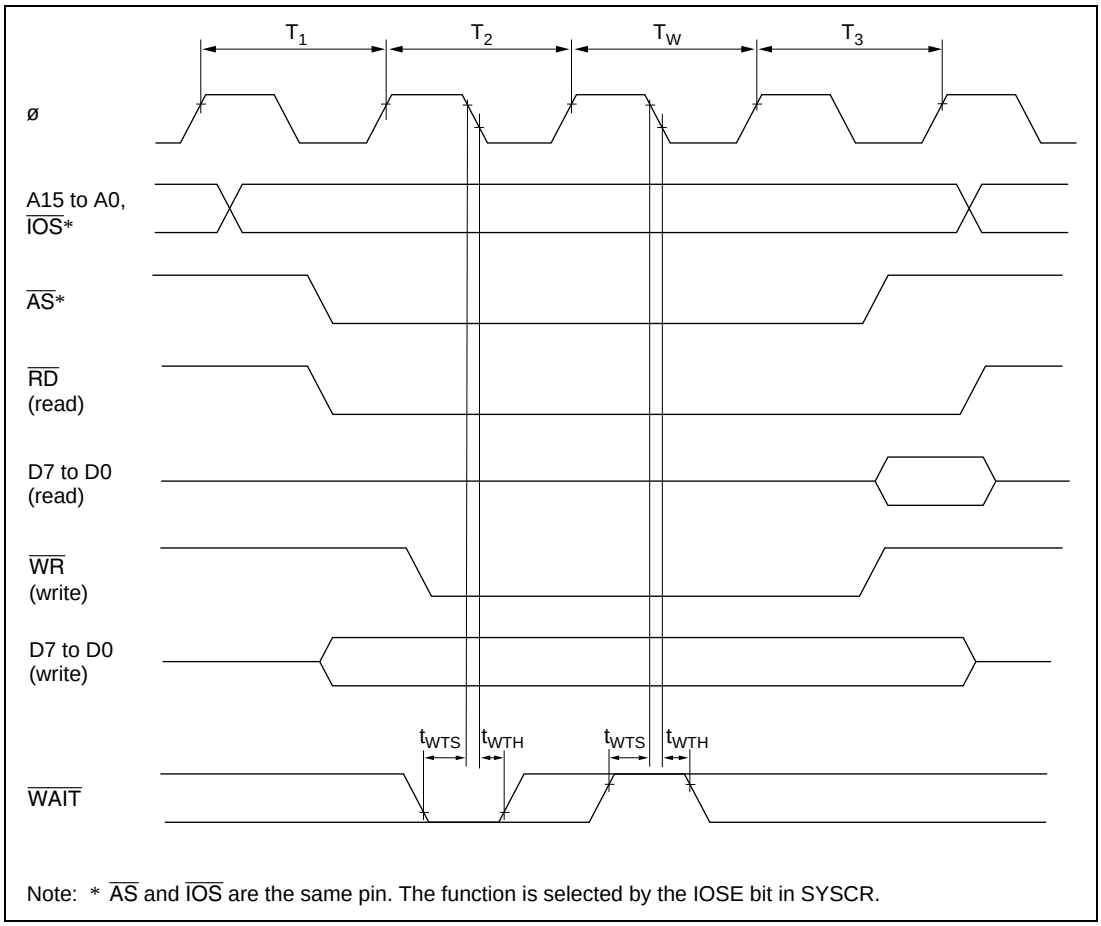


Figure 22.11 Basic Bus Timing (Three-State Access with One Wait State)

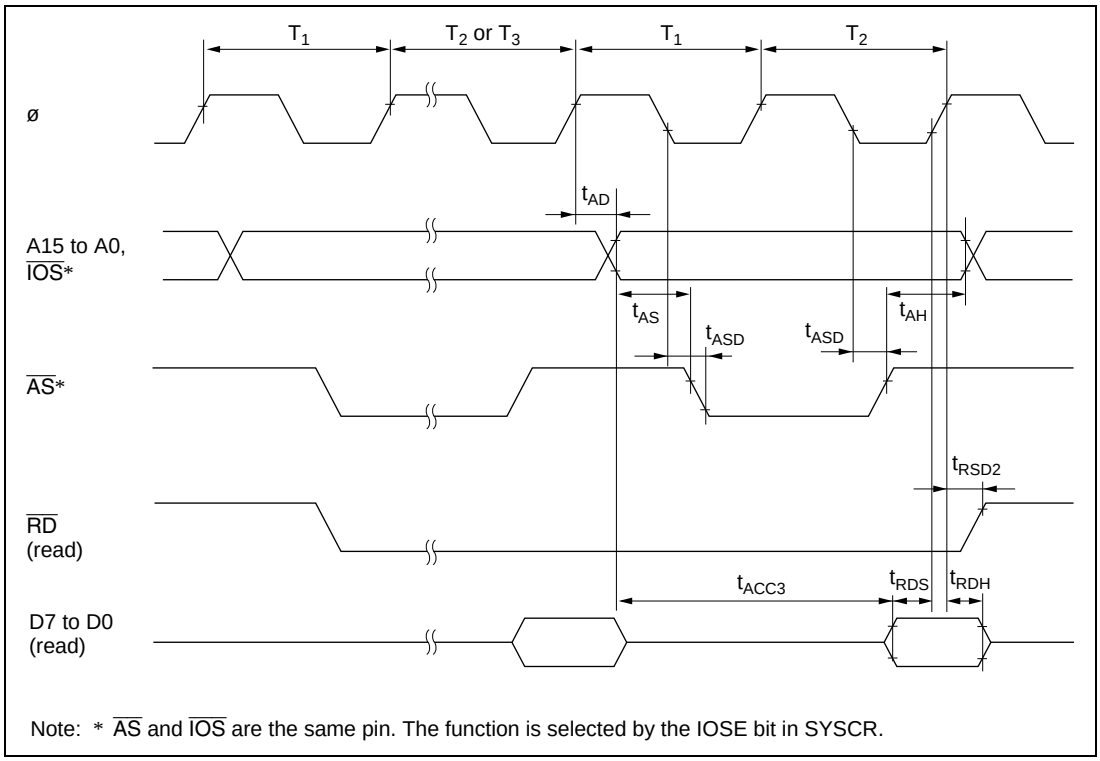


Figure 22.12 Burst ROM Access Timing (Two-State Access)

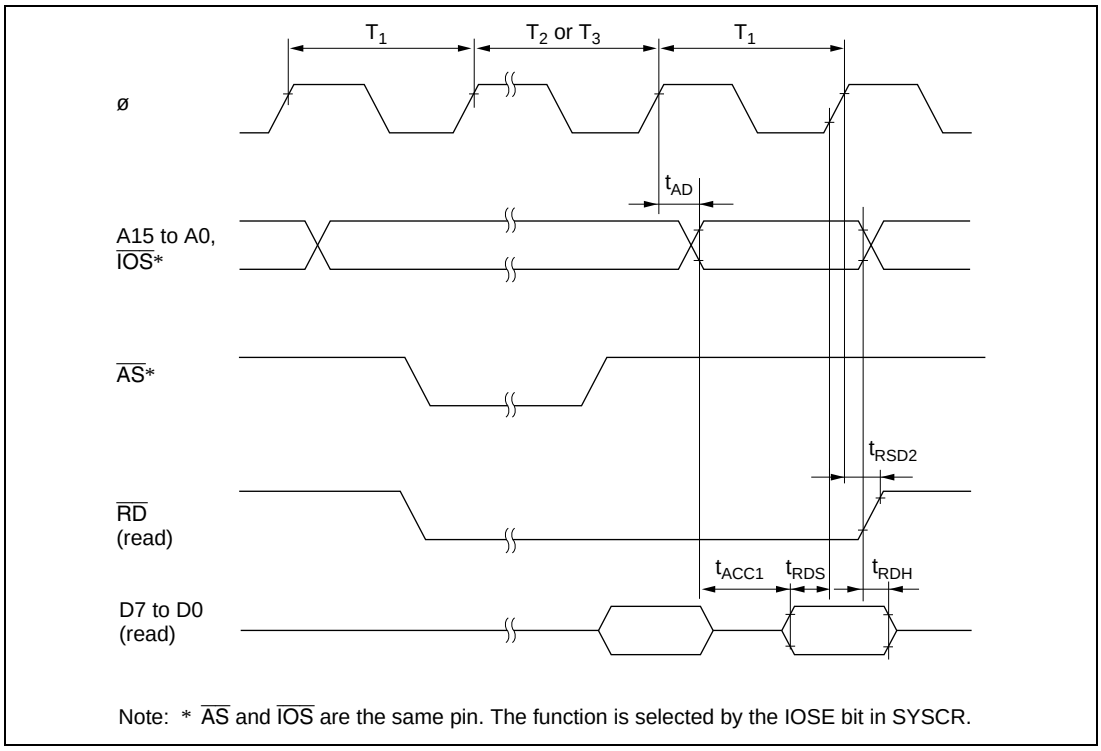


Figure 22.13 Burst ROM Access Timing (One-State Access)

22.3.4 Timing of On-Chip Supporting Modules

Tables 22.8 and 22.9 show the on-chip supporting module timing. The only on-chip supporting modules that can operate in subclock operation (ø = 32.768 kHz) are the I/O ports, external interrupts (NMI and IRQ0, 1, and IRQ2), the watchdog timer, and the 8-bit timer (channels 0 and 1).

Table 22.8 Timing of On-Chip Supporting Modules – Preliminary –

Condition A: V_{CC} = 5.0 V ± 10%, V_{SS} = 0 V, ø = 32.768 kHz*[1], 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)
Condition B: V_{CC} = 4.0 V to 5.5 V, V_{SS} = 0 V, ø = 32.768 kHz*[1], 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)
Condition C: V_{CC} = 2.7 V to 5.5 V*[2], V_{SS} = 0 V, ø = 32.768 kHz*[1], 2 MHz to maximum operating frequency, T_a = –20 to +75°C

				Condition A		Condition B		Condition C			
				20 MHz		16 MHz		10 MHz			
Item			Symbol	Min	Max	Min	Max	Min	Max	Unit	Test Conditions
I/O ports	Output data delay time		t_{PWD}	—	50	—	50	—	100	ns	Figure 22.14
	Input data setup time		t_{PRS}	30	—	30	—	50	—		
	Input data hold time		t_{PRH}	30	—	30	—	50	—		
FRT	Timer output delay time		t_{FTOD}	—	50	—	50	—	100	ns	Figure 22.15
	Timer input setup time		t_{FTIS}	30	—	30	—	50	—		
	Timer clock input setup time		t_{FTCS}	30	—	30	—	50	—		Figure 22.16
	Timer clock pulse width	Single edge	t_{FTCWH}	1.5	—	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{FTCWL}	2.5	—	2.5	—	2.5	—		

Table 22.8 Timing of On-Chip Supporting Modules (cont) **– Preliminary –**

Condition A: V_{CC} = 5.0 V ± 10%, V_{SS} = 0 V, ø = 32.768 kHz*[1], 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)
Condition B: V_{CC} = 4.0 V to 5.5 V, V_{SS} = 0 V, ø = 32.768 kHz*[1], 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)
Condition C: V_{CC} = 2.7 V to 5.5 V*[2], V_{SS} = 0 V, ø = 32.768 kHz*[1], 2 MHz to maximum operating frequency, T_a = –20 to +75°C

				Condition A		Condition B		Condition C			
				20 MHz		16 MHz		10 MHz			
Item			Symbol	Min	Max	Min	Max	Min	Max	Unit	Test Conditions
TMR	Timer output delay time		t_{TMOD}	—	50	—	50	—	100	ns	Figure 22.17
	Timer reset input setup time		t_{TMRS}	30	—	30	—	50	—		Figure 22.19
	Timer clock input setup time		t_{TMCS}	30	—	30	—	50	—		Figure 22.18
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	1.5	—	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—	2.5	—	2.5	—		
PWM, PWMX	Pulse output delay time		t_{PWOD}	—	50	—	50	—	100	ns	Figure 22.20
SCI	Input clock cycle	Asynchro-nous	t_{Scyc}	4	—	4	—	4	—	t_{cyc}	Figure 22.21
		Synchro-nous		6	—	6	—	6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	0.4	0.6	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	—	1.5	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5	—	1.5	—	1.5		

Table 22.8 Timing of On-Chip Supporting Modules (cont) **– Preliminary –**

Condition A: V_{CC} = 5.0 V ± 10%, V_{SS} = 0 V, ø = 32.768 kHz*[1], 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)
Condition B: V_{CC} = 4.0 V to 5.5 V, V_{SS} = 0 V, ø = 32.768 kHz*[1], 2 MHz to maximum operating frequency, T_a = –20 to +75°C (regular specifications), T_a = –40 to +85°C (wide-range specifications)
Condition C: V_{CC} = 2.7 V to 5.5 V*[2], V_{SS} = 0 V, ø = 32.768 kHz*[1], 2 MHz to maximum operating frequency, T_a = –20 to +75°C

			Condition A		Condition B		Condition C			
			20 MHz		16 MHz		10 MHz			
Item		Symbol	Min	Max	Min	Max	Min	Max	Unit	Test Conditions
SCI	Transmit data delay time (synchronous)	t_{TXD}	—	50	—	50	—	100	ns	Figure 22.22
	Receive data setup time (synchronous)	t_{RXS}	50	—	50	—	100	—	ns	
	Receive data hold time (synchronous)	t_{RXH}	50	—	50	—	100	—	ns	
A/D converter	Trigger input setup time	t_{TRGS}	30	—	30	—	50	—	ns	Figure 22.23

Notes: 1. Only supporting modules that can be used in subclock operation
2. For the low-voltage F-ZTAT version, V_{CC} = 3.0 V to 5.5 V

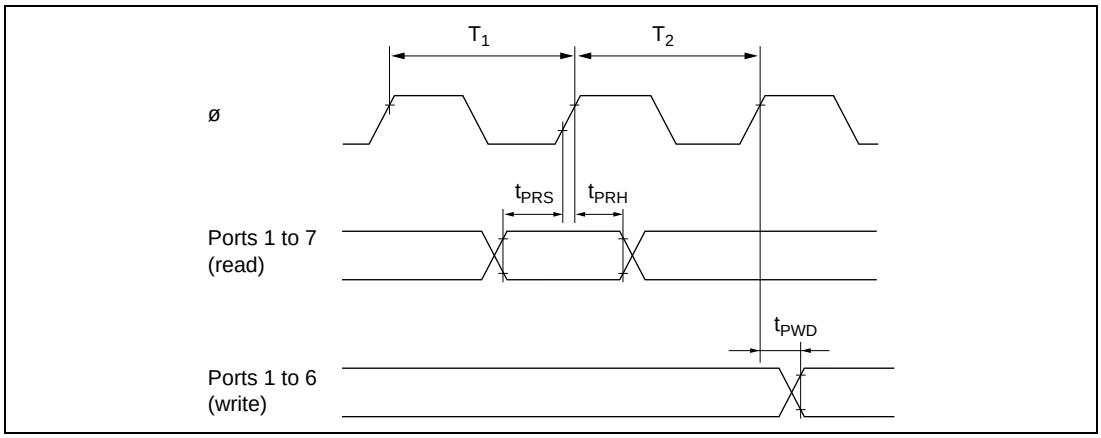


Figure 22.14 I/O Port Input/Output Timing

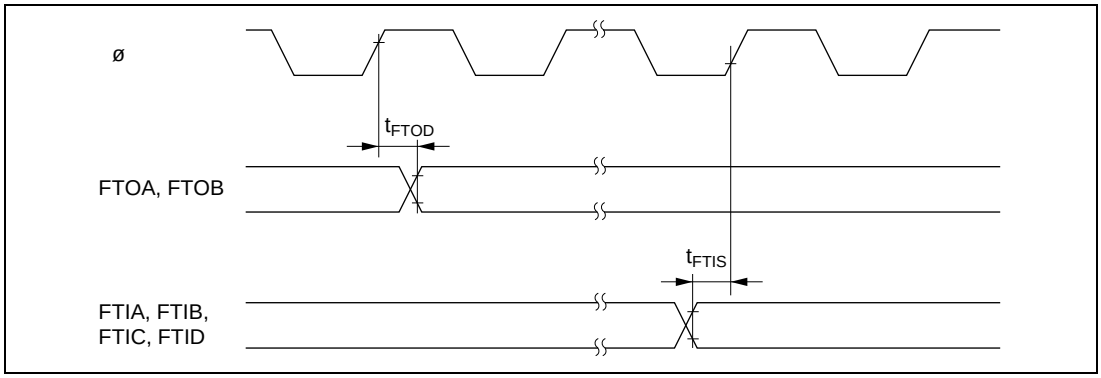


Figure 22.15 FRT Input/Output Timing

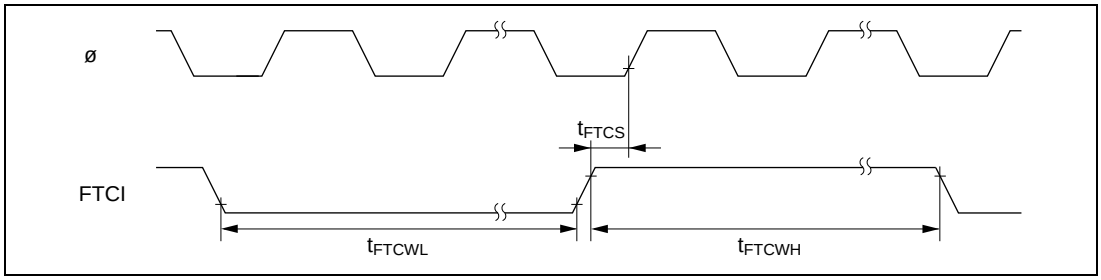


Figure 22.16 FRT Clock Input Timing

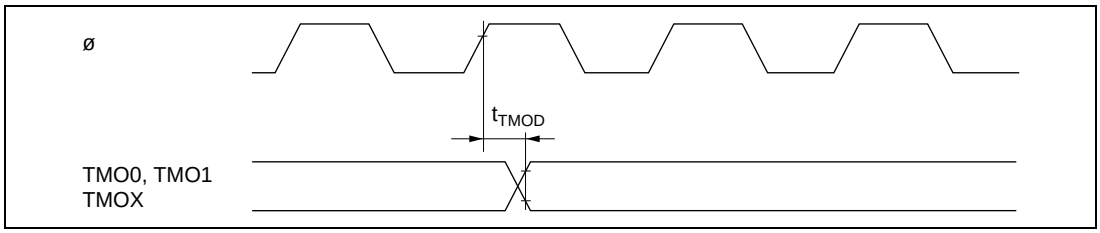


Figure 22.17 8-Bit Timer Output Timing

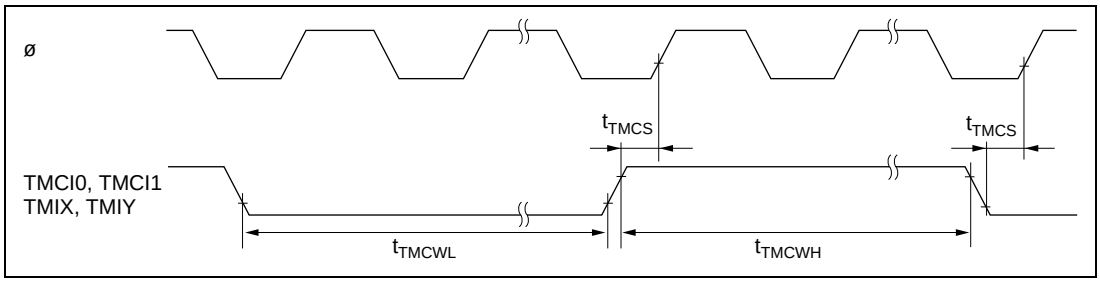


Figure 22.18 8-Bit Timer Clock Input Timing

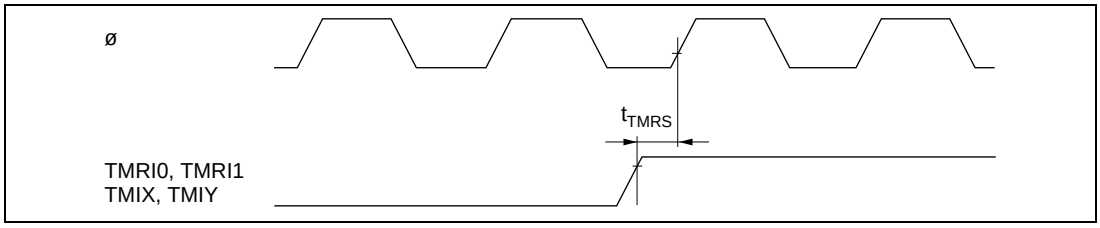


Figure 22.19 8-Bit Timer Reset Input Timing

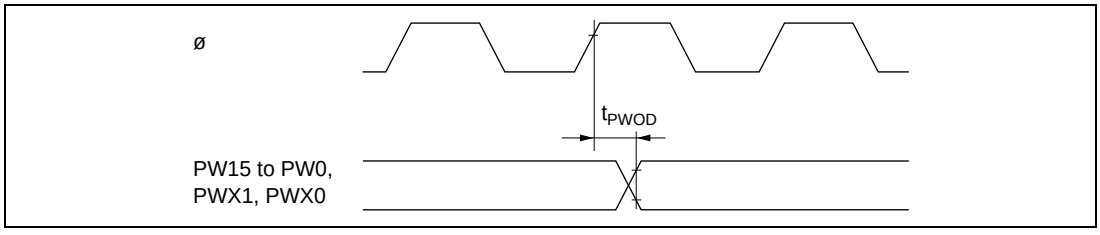


Figure 22.20 PWM, PWMX Output Timing

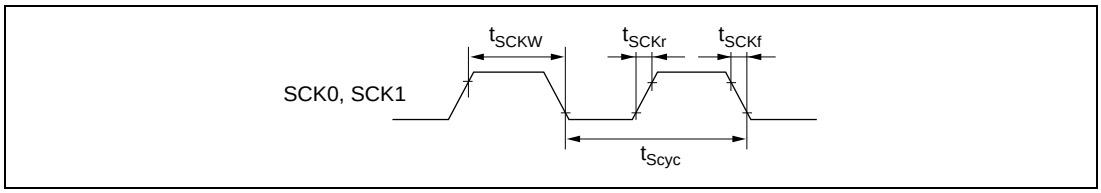


Figure 22.21 SCK Clock Input Timing

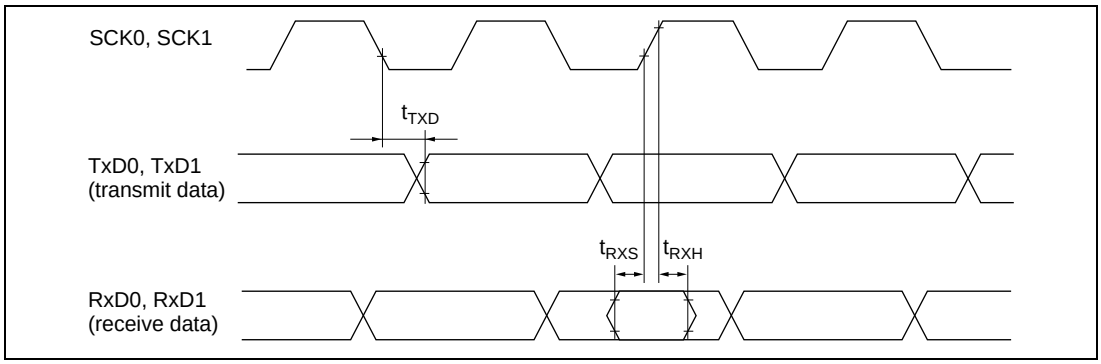


Figure 22.22 SCI Input/Output Timing (Synchronous Mode)

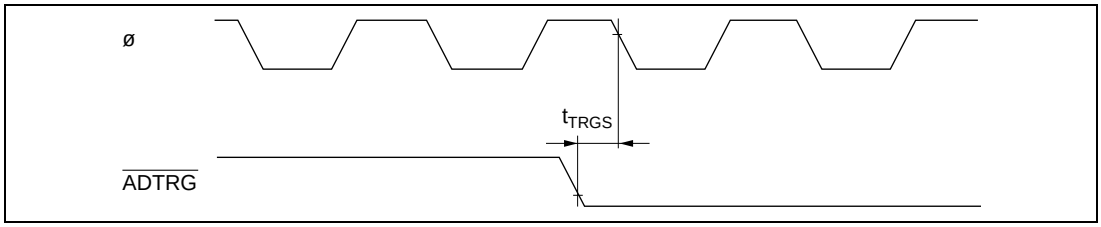


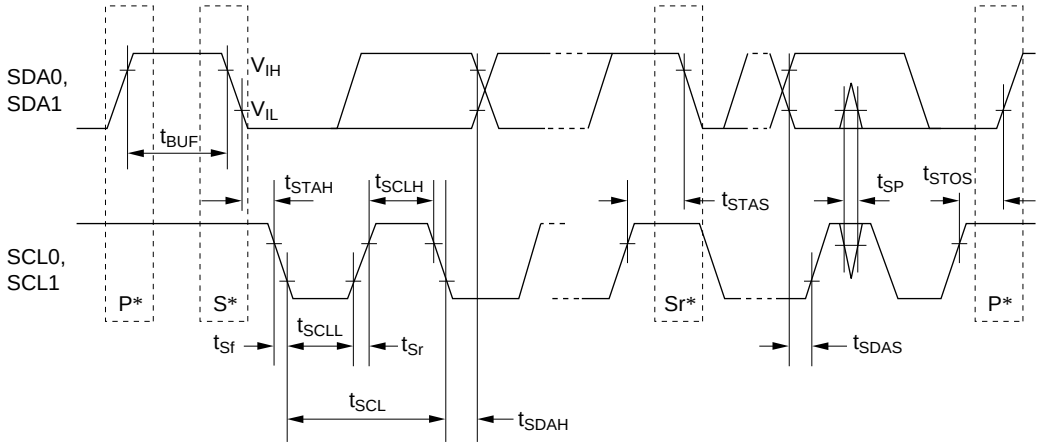
Figure 22.23 A/D Converter External Trigger Input Timing

Table 22.9 I^2C Bus Timing – Preliminary –

Conditions: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, ø = 5 MHz to maximum operating frequency, T_a = –20 to +75°C

Item	Symbol	Min	Typ	Max	Unit	Test Conditions	Notes
SCL clock cycle time	t_{SCL}	12	—	—	t_{cyc}		Figure 22.24
SCL clock high pulse width	t_{SCLH}	3	—	—	t_{cyc}		
SCL clock low pulse width	t_{SCLL}	5	—	—	t_{cyc}		
SCL, SDA input rise time	t_{Sr}	—	—	7.5*	t_{cyc}		
SCL, SDA input fall time	t_{Sf}	—	—	300	ns		
SCL, SDA input spike pulse elimination time	t_{SP}	—	—	1	t_{cyc}		
SDA input bus free time	t_{BUF}	5	—	—	t_{cyc}		
Start condition input hold time	t_{STAH}	3	—	—	t_{cyc}		
Retransmission start condition input setup time	t_{STAS}	3	—	—	t_{cyc}		
Stop condition input setup time	t_{STOS}	3	—	—	t_{cyc}		
Data input setup time	t_{SDAS}	0.5	—	—	t_{cyc}		
Data input hold time	t_{SDAH}	0	—	—	ns		
SCL, SDA capacitive load	C_b	—	—	400	pF		

Note: * $17.5t_{cyc}$ can be set according to the clock selected for use by the I^2C module. For details, see section 16.4, Usage Notes.



Note: * S, P, and Sr indicate the following conditions.

S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 22.24 I²C Bus Interface Input/Output Timing (Option)

22.4 A/D Conversion Characteristics

Tables 22.10 and 22.11 list the A/D conversion characteristics.

Table 22.10 A/D Conversion Characteristics

(AN7 to AN0 Input: 134/266-State Conversion) – Preliminary –

Condition A: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%
V_{SS} = AV_{SS} = 0 V, ø = 2 MHz to maximum operating frequency,
T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)

Condition B: V_{CC} = 4.0 V to 5.5 V, AV_{CC} = 4.0 V to 5.5 V
V_{SS} = AV_{SS} = 0 V, ø = 2 MHz to maximum operating frequency,
T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)

Condition C: V_{CC} = 2.7 V to 5.5 V*[5], AV_{CC} = 2.7 V to 5.5 V*[5]
V_{SS} = AV_{SS} = 0 V, ø = 2 MHz to maximum operating frequency,
T_a = –20 to +75°C

	Condition A			Condition B			Condition C			
	20 MHz			16 MHz			10 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	10	10	10	Bits
Conversion time	—	—	6.7	—	—	8.4	—	—	13.4	µs
Analog input capacitance	—	—	20	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	10*[3] 5*[4]	—	—	10*[3] 5*[4]	—	—	10*[1] 5*[2]	kΩ
Nonlinearity error	—	—	±3.0	—	—	±3.0	—	—	±7.0	LSB
Offset error	—	—	±3.5	—	—	±3.5	—	—	±7.5	LSB
Full-scale error	—	—	±3.5	—	—	±3.5	—	—	±7.5	LSB
Quantization error	—	—	±0.5	—	—	±0.5	—	—	±0.5	LSB
Absolute accuracy	—	—	±4.0	—	—	±4.0	—	—	±8.0	LSB

Notes:
1. When 4.0 V ≤ AV_{CC} ≤ 5.5 V
2. When 2.7 V ≤ AV_{CC} < 4.0 V
3. When conversion time ≥ 11. 17 µs (CKS = 1 and ø ≤ 12 MHz, or CKS = 0)
4. When conversion time < 11. 17 µs (CKS = 1 and ø > 12 MHz)
5. For the low-voltage F-ZTAT version, V_{CC} = 3.0 V to 5.5 V and AV_{CC} = 3.0 V to 5.5 V.

Table 22.11 A/D Conversion Characteristics

(CIN7 to CIN0 Input: 134/266-State Conversion) – Preliminary –

Condition A: V_{CC} = 5.0 V ± 10%, AV_{CC} = 5.0 V ± 10%
$V_{SS} = AV_{SS}$ = 0 V, ø = 2 MHz to maximum operating frequency,
T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)

Condition B: V_{CC} = 4.0 V to 5.5 V, AV_{CC} = 4.0 V to 5.5 V
$V_{SS} = AV_{SS}$ = 0 V, ø = 2 MHz to maximum operating frequency,
T_a = –20 to +75°C (regular specifications),
T_a = –40 to +85°C (wide-range specifications)

Condition C: V_{CC} = 2.7 V to 5.5 V*[5], AV_{CC} = 2.7 V to 5.5 V*[5]
$V_{SS} = AV_{SS}$ = 0 V, ø = 2 MHz to maximum operating frequency,
T_a = –20 to +75°C

	Condition A			Condition B			Condition C			
	20 MHz			16 MHz			10 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	10	10	10	Bits
Conversion time	—	—	6.7	—	—	8.4	—	—	13.4	µs
Analog input capacitance	—	—	20	—	—	20	—	—	20	pF
Permissible signal-source impedance	—	—	10*[3] 5*[4]	—	—	10*[3] 5*[4]	—	—	10*[1] 5*[2]	kΩ
Nonlinearity error	—	—	±5.0	—	—	±5.0	—	—	±11.0	LSB
Offset error	—	—	±5.5	—	—	±5.5	—	—	±11.5	LSB
Full-scale error	—	—	±5.5	—	—	±5.5	—	—	±11.5	LSB
Quantization error	—	—	±0.5	—	—	±0.5	—	—	±0.5	LSB
Absolute accuracy	—	—	±6.0	—	—	±6.0	—	—	±12.0	LSB

Notes:
1. When 4.0 V ≤ AV_{CC} ≤ 5.5 V
2. When 2.7 V ≤ AV_{CC} < 4.0 V
3. When conversion time ≥ 11. 17 µs (CKS = 1 and ø ≤ 12 MHz, or CKS = 0)
4. When conversion time < 11. 17 µs (CKS = 1 and ø > 12 MHz)
5. For the low-voltage F-ZTAT version, V_{CC} = 3.0 V to 5.5 V and AV_{CC} = 3.0 V to 5.5 V.

22.5 Flash Memory Characteristics

Table 22.12 shows the flash memory characteristics.

Table 22.12 Flash Memory Characteristics

Conditions (5 V version): V_{CC} = 5.0 V ± 10%, V_{SS} = 0 V, T_a = 0 to +75°C (regular specifications), T_a = 0 to +85°C (wide-range specifications)
Conditions for low-voltage version: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, T_a = 0 to +75°C (Programming/erasing operating temperature)

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Programming time*1,*2,*4		tP	—	10	200	ms/ 32 bytes	
Erase time*1,*3,*5		tE	—	100	1200	ms/ block	
Reprogramming count		N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE-bit setting*1	x	10	—	—	µs	
	Wait time after PSU-bit setting*1	y	50	—	—	µs	
	Wait time after P-bit setting*1, *4	z	150	—	200	µs	
	Wait time after P-bit clear*1	α	10	—	—	µs	
	Wait time after PSU-bit clear*1	β	10	—	—	µs	
	Wait time after PV-bit setting*1	γ	4	—	—	µs	
	Wait time after dummy write*1	ε	2	—	—	µs	
	Wait time after PV-bit clear*1	η	4	—	—	µs	
	Maximum programming count*1,*4,*5	N	—	—	1000	Times	tP = 200 µs

Table 22.12 Flash Memory Characteristics (cont)

Conditions (5 V version): V_{CC} = 5.0 V ± 10%, V_{SS} = 0 V, T_a = 0 to +75°C (regular specifications), T_a = 0 to +85°C (wide-range specifications)
Conditions for low-voltage version: V_{CC} = 3.0 V to 3.6 V, V_{SS} = 0 V, T_a = 0 to +75°C (Programming/erasing operating temperature)

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Erase	Wait time after SWE-bit setting*[1]	x	10	—	—	µs	
	Wait time after ESU-bit setting*[1]	y	200	—	—	µs	
	Wait time after E-bit setting*[1]*[6]	z	5	—	10	ms	
	Wait time after E-bit clear*[1]	α	10	—	—	µs	
	Wait time after ESU-bit clear*[1]	β	10	—	—	µs	
	Wait time after EV-bit setting*[1]	γ	20	—	—	µs	
	Wait time after dummy write*[1]	ε	2	—	—	µs	
	Wait time after EV-bit clear*[1]	η	5	—	—	µs	
	Maximum erase count*[1]*[6]*[7]	N	—	—	120	Times	tE = 10 ms

Notes: 1. Set the times according to the program/erase algorithms.
2. Programming time per 32 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR1) is set. It does not include the programming verification time.)
3. Block erase time (Shows the total period for which the E-bit in FLMCR1 is set. It does not include the erase verification time.)
4. Maximum programming time (tP (max) = wait time after P-bit setting (z) × maximum programming count (N))
5. Number of times when the wait time after P-bit setting (z) = 200 µs.
   The number of writes should be set according to the actual set value of z to allow programming within the maximum programming time (tP).
6. Maximum erase time (tE (max) = Wait time after E-bit setting (z) × maximum erase count (N))
7. Number of times when the wait time after E-bit setting (z) = 10 ms.
   The number of erases should be set according to the actual set value of z to allow erasing within the maximum erase time (tE).

22.6 Usage Note

The F-ZTAT and mask ROM versions have been confirmed as fully meeting the reference values for electrical characteristics shown in this manual. However, actual performance figures, operating margins, noise margins, and other properties may vary due to differences in the manufacturing process, on-chip ROM, layout patterns, etc.

When system evaluation testing is carried out using the F-ZTAT version, the same evaluation tests should also be conducted for the mask ROM version when changing over to that version.

Appendix A Instruction Set

A.1 Instruction

Operation Notation

Rd	General register (destination)*[1]
Rs	General register (source)*[1]
Rn	General register*[1]
ERn	General register (32-bit register)
MAC	Multiply-and-accumulate register (32-bit register)*[2]
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extend 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
⊕	Exclusive logical OR
→	Transfer from left-hand operand to right-hand operand, or transition from left-hand state to right-hand state
¬	NOT (logical complement)
( ) < >	Operand contents
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Notes: 1. General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).
2. MAC instructions cannot be used in the H8S/2128 Series and H8S/2124 Series.

Condition Code Notation

Symbol	Meaning
↕	Changes according operation result.
*	Indeterminate (value not guaranteed).
0	Always cleared to 0.
1	Always set to 1.
—	Not affected by operation result.

Table A.1 Instruction Set

1. Data Transfer Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
MOV	MOV.B #xx:8,Rd	B	2									#xx:8→Rd8	—	—	↕	↕	0	—	1	
	MOV.B Rs,Rd	B		2								Rs8→Rd8	—	—	↕	↕	0	—	1	
	MOV.B @ERs,Rd	B			2							@ERs→Rd8	—	—	↕	↕	0	—	2	
	MOV.B @(d:16,ERs),Rd	B				4						@(d:16,ERs)→Rd8	—	—	↕	↕	0	—	3	
	MOV.B @(d:32,ERs),Rd	B				8						@(d:32,ERs)→Rd8	—	—	↕	↕	0	—	5	
	MOV.B @ERs+,Rd	B					2					@ERs→Rd8,ERs32+1→ERs32	—	—	↕	↕	0	—	3	
	MOV.B @aa:8,Rd	B						2				@aa:8→Rd8	—	—	↕	↕	0	—	2	
	MOV.B @aa:16,Rd	B						4				@aa:16→Rd8	—	—	↕	↕	0	—	3	
	MOV.B @aa:32,Rd	B						6				@aa:32→Rd8	—	—	↕	↕	0	—	4	
	MOV.B Rs,@ERd	B			2							Rs8→@ERd	—	—	↕	↕	0	—	2	
	MOV.B Rs,@(d:16,ERd)	B				4						Rs8→@(d:16,ERd)	—	—	↕	↕	0	—	3	
	MOV.B Rs,@(d:32,ERd)	B				8						Rs8→@(d:32,ERd)	—	—	↕	↕	0	—	5	
	MOV.B Rs,@-ERd	B					2					ERd32-1→ERd32,Rs8→@ERd	—	—	↕	↕	0	—	3	
	MOV.B Rs,@aa:8	B						2				Rs8→@aa:8	—	—	↕	↕	0	—	2	
	MOV.B Rs,@aa:16	B						4				Rs8→@aa:16	—	—	↕	↕	0	—	3	
	MOV.B Rs,@aa:32	B						6				Rs8→@aa:32	—	—	↕	↕	0	—	4	
	MOV.W #xx:16,Rd	W	4									#xx:16→Rd16	—	—	↕	↕	0	—	2	
	MOV.W Rs,Rd	W		2								Rs16→Rd16	—	—	↕	↕	0	—	1	
	MOV.W @ERs,Rd	W			2							@ERs→Rd16	—	—	↕	↕	0	—	2	
	MOV.W @(d:16,ERs),Rd	W				4						@(d:16,ERs)→Rd16	—	—	↕	↕	0	—	3	
	MOV.W @(d:32,ERs),Rd	W				8						@(d:32,ERs)→Rd16	—	—	↕	↕	0	—	5	
	MOV.W @ERs+,Rd	W					2					@ERs→Rd16,ERs32+2→ERs32	—	—	↕	↕	0	—	3	
	MOV.W @aa:16,Rd	W						4				@aa:16→Rd16	—	—	↕	↕	0	—	3	
	MOV.W @aa:32,Rd	W						6				@aa:32→Rd16	—	—	↕	↕	0	—	4	
	MOV.W Rs,@ERd	W			2							Rs16→@ERd	—	—	↕	↕	0	—	2	
	MOV.W Rs,@(d:16,ERd)	W				4						Rs16→@(d:16,ERd)	—	—	↕	↕	0	—	3	
	MOV.W Rs,@(d:32,ERd)	W				8						Rs16→@(d:32,ERd)	—	—	↕	↕	0	—	5	
	MOV.W Rs,@-ERd	W					2					ERd32-2→ERd32,Rs16→@ERd	—	—	↕	↕	0	—	3	
	MOV.W Rs,@aa:16	W						4				Rs16→@aa:16	—	—	↕	↕	0	—	3	
	MOV.W Rs,@aa:32	W						6				Rs16→@aa:32	—	—	↕	↕	0	—	4	

Table A.1 Instruction Set (cont)

1. Data Transfer Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
MOV	MOV.L #xx:32,ERd	L	6									#xx:32→Rd32	—	—	↕	↕	0	—	3	
	MOV.L ERs,ERd	L		2								ERs32→ERd32	—	—	↕	↕	0	—	1	
	MOV.L @ERs,ERd	L			4							@ERs→ERd32	—	—	↕	↕	0	—	4	
	MOV.L @(d:16,ERs),ERd	L				6						@(d:16,ERs)→ERd32	—	—	↕	↕	0	—	5	
	MOV.L @(d:32,ERs),ERd	L				10						@(d:32,ERs)→ERd32	—	—	↕	↕	0	—	7	
	MOV.L @ERs+,ERd	L					4					@ERs→ERd32,ERs32+4→ERs32	—	—	↕	↕	0	—	5	
	MOV.L @aa:16,ERd	L						6				@aa:16→ERd32	—	—	↕	↕	0	—	5	
	MOV.L @aa:32,ERd	L						8				@aa:32→ERd32	—	—	↕	↕	0	—	6	
	MOV.L ERs,@ERd	L			4							ERs32→@ERd	—	—	↕	↕	0	—	4	
	MOV.L ERs,@(d:16,ERd)	L				6						ERs32→@(d:16,ERd)	—	—	↕	↕	0	—	5	
	MOV.L ERs,@(d:32,ERd)	L				10						ERs32→@(d:32,ERd)	—	—	↕	↕	0	—	7	
	MOV.L ERs,@-ERd	L					4					ERd32-4→ERd32,ERs32→@ERd	—	—	↕	↕	0	—	5	
	MOV.L ERs,@aa:16	L						6				ERs32→@aa:16	—	—	↕	↕	0	—	5	
	MOV.L ERs,@aa:32	L						8				ERs32→@aa:32	—	—	↕	↕	0	—	6	
POP	POP.W Rn	W									2	@SP→Rn16,SP+2→SP	—	—	↕	↕	0	—	3	
	POP.L ERn	L									4	@SP→ERn32,SP+4→SP	—	—	↕	↕	0	—	5	
PUSH	PUSH.W Rn	W									2	SP-2→SP,Rn16→@SP	—	—	↕	↕	0	—	3	
	PUSH.L ERn	L									4	SP-4→SP,ERn32→@SP	—	—	↕	↕	0	—	5	
LDM	LDM @SP+,(ERm-ERn)	L									4	(@SP→ERn32,SP+4→SP) Repeated for each restored register.	—	—	—	—	—	—	7/9/11 [1]	
STM	STM (ERm-ERn),@-SP	L									4	(SP-4→SP,ERn32→@SP) Repeated for each saved register.	—	—	—	—	—	—	7/9/11 [1]	
MOVFPE	MOVFPE @aa:16,Rd	Cannot be used with the H8S/2128 Series and H8S/2124 Series.																	[2]	
MOVTPE	MOVTPE Rs,@aa:16																		[2]	

Table A.1 Instruction Set (cont)

2. Arithmetic Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
ADD	ADD.B #xx:8,Rd	B	2									Rd8+#xx:8→Rd8	—	↕	↕	↕	↕	↕	1	
	ADD.B Rs,Rd	B		2								Rd8+Rs8→Rd8	—	↕	↕	↕	↕	↕	1	
	ADD.W #xx:16,Rd	W	4									Rd16+#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2	
	ADD.W Rs,Rd	W		2								Rd16+Rs16→Rd16	—	[3]	↕	↕	↕	↕	1	
	ADD.L #xx:32,ERd	L	6									ERd32+#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3	
	ADD.L ERs,ERd	L		2								ERd32+ERs32→ERd32	—	[4]	↕	↕	↕	↕	1	
ADDX	ADDX #xx:8,Rd	B	2									Rd8+#xx:8+C→Rd8	—	↕	↕	[5]	↕	↕	1	
	ADDX Rs,Rd	B		2								Rd8+Rs8+C→Rd8	—	↕	↕	[5]	↕	↕	1	
ADDS	ADDS #1,ERd	L		2								ERd32+1→ERd32	—	—	—	—	—	—	1	
	ADDS #2,ERd	L		2								ERd32+2→ERd32	—	—	—	—	—	—	1	
	ADDS #4,ERd	L		2								ERd32+4→ERd32	—	—	—	—	—	—	1	
INC	INC.B Rd	B		2								Rd8+1→Rd8	—	—	↕	↕	↕	—	1	
	INC.W #1,Rd	W		2								Rd16+1→Rd16	—	—	↕	↕	↕	—	1	
	INC.W #2,Rd	W		2								Rd16+2→Rd16	—	—	↕	↕	↕	—	1	
	INC.L #1,ERd	L		2								ERd32+1→ERd32	—	—	↕	↕	↕	—	1	
	INC.L #2,ERd	L		2								ERd32+2→ERd32	—	—	↕	↕	↕	—	1	
DAA	DAA Rd	B		2								Rd8 decimal adjust →Rd8	—	*	↕	↕	*	↕	1	
SUB	SUB.B Rs,Rd	B		2								Rd8-Rs8→Rd8	—	↕	↕	↕	↕	↕	1	
	SUB.W #xx:16,Rd	W	4									Rd16-#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2	
	SUB.W Rs,Rd	W		2								Rd16-Rs16→Rd16	—	[3]	↕	↕	↕	↕	1	
	SUB.L #xx:32,ERd	L	6									ERd32-#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3	
	SUB.L ERs,ERd	L		2								ERd32-ERs32→ERd32	—	[4]	↕	↕	↕	↕	1	
SUBX	SUBX #xx:8,Rd	B	2									Rd8-#xx:8-C→Rd8	—	↕	↕	[5]	↕	↕	1	
	SUBX Rs,Rd	B		2								Rd8-Rs8-C→Rd8	—	↕	↕	[5]	↕	↕	1	
SUBS	SUBS #1,ERd	L		2								ERd32-1→ERd32	—	—	—	—	—	—	1	
	SUBS #2,ERd	L		2								ERd32-2→ERd32	—	—	—	—	—	—	1	
	SUBS #4,ERd	L		2								ERd32-4→ERd32	—	—	—	—	—	—	1	
DEC	DEC.B Rd	B		2								Rd8-1→Rd8	—	—	↕	↕	↕	—	1	
	DEC.W #1,Rd	W		2								Rd16-1→Rd16	—	—	↕	↕	↕	—	1	
	DEC.W #2,Rd	W		2								Rd16-2→Rd16	—	—	↕	↕	↕	—	1	
	DEC.L #1,ERd	L		2								ERd32-1→ERd32	—	—	↕	↕	↕	—	1	
	DEC.L #2,ERd	L		2								ERd32-2→ERd32	—	—	↕	↕	↕	—	1	

Table A.1 Instruction Set (cont)

2. Arithmetic Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
DAS	DAS Rd	B		2								Rd8 decimal adjust →Rd8	—	*	↕	↕	*	—	1	
MULXU	MULXU.B Rs,Rd	B		2								Rd8×Rs8→Rd16 (unsigned multiplication)	—	—	—	—	—	—	12	
	MULXU.W Rs,ERd	W		2								Rd16×Rs16→ERd32 (unsigned multiplication)	—	—	—	—	—	—	20	
MULXS	MULXS.B Rs,Rd	B		4								Rd8×Rs8→Rd16 (signed multiplication)	—	—	↕	↕	—	—	13	
	MULXS.W Rs,ERd	W		4								Rd16×Rs16→ERd32 (signed multiplication)	—	—	↕	↕	—	—	21	
DIVXU	DIVXU.B Rs,Rd	B		2								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	[6]	[7]	—	—	12	
	DIVXU.W Rs,ERd	W		2								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	[6]	[7]	—	—	20	
DIVXS	DIVXS.B Rs,Rd	B		4								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	[8]	[7]	—	—	13	
	DIVXS.W Rs,ERd	W		4								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	[8]	[7]	—	—	21	
CMP	CMP.B #xx:8,Rd	B	2									Rd8-#xx:8	—	↕	↕	↕	↕	↕	1	
	CMP.B Rs,Rd	B		2								Rd8-Rs8	—	↕	↕	↕	↕	↕	1	
	CMP.W #xx:16,Rd	W	4									Rd16-#xx:16	—	[3]	↕	↕	↕	↕	2	
	CMP.W Rs,Rd	W		2								Rd16-Rs16	—	[3]	↕	↕	↕	↕	1	
	CMP.L #xx:32,ERd	L	6									ERd32-#xx:32	—	[4]	↕	↕	↕	↕	3	
	CMP.L ERs,ERd	L		2								ERd32-ERs32	—	[4]	↕	↕	↕	↕	1	
NEG	NEG.B Rd	B		2								0-Rd8→Rd8	—	↕	↕	↕	↕	↕	1	
	NEG.W Rd	W		2								0-Rd16→Rd16	—	↕	↕	↕	↕	↕	1	
	NEG.L ERd	L		2								0-ERd32→ERd32	—	↕	↕	↕	↕	↕	1	
EXTU	EXTU.W Rd	W		2								0 → (<bits 5 to 8> of Rd16)	—	—	0	↕	0	—	1	
	EXTU.L ERd	L		2								0 → (<bits 31 to 16> of ERd32)	—	—	0	↕	0	—	1	
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↕	↕	0	—	1	
	EXTS.L ERd	L		2								(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↕	↕	0	—	1	
TAS	TAS @ERd	B			4							@ERd-0 → CCR set, (1) → (<bit 7> of @ERd)	—	—	↕	↕	0	—	4	

Table A.1 Instruction Set (cont)

2. Arithmetic Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
MAC	MAC @ERn+,@ERm+	Cannot be used with the H8S/2128 Series and H8S/2124 Series.																	[2]	
CLRMAC	CLRMAC																			
LDMAC	LDMAC ERs,MACH																			
	LDMAC ERs,MACL																			
STMAC	STMAC MACH,ERd																			
	STMAC MACL,ERd																			

Table A.1 Instruction Set (cont)

3. Logic Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
AND	AND.B #xx:8,Rd	B	2									Rd8∧#xx:8→Rd8	—	—	↕	↕	0	—	1	
	AND.B Rs,Rd	B		2								Rd8∧Rs8→Rd8	—	—	↕	↕	0	—	1	
	AND.W #xx:16,Rd	W	4									Rd16∧#xx:16→Rd16	—	—	↕	↕	0	—	2	
	AND.W Rs,Rd	W		2								Rd16∧Rs16→Rd16	—	—	↕	↕	0	—	1	
	AND.L #xx:32,ERd	L	6									ERd32∧#xx:32→ERd32	—	—	↕	↕	0	—	3	
	AND.L ERs,ERd	L		4								ERd32∧ERs32→ERd32	—	—	↕	↕	0	—	2	
OR	OR.B #xx:8,Rd	B	2									Rd8∨#xx:8→Rd8	—	—	↕	↕	0	—	1	
	OR.B Rs,Rd	B		2								Rd8∨Rs8→Rd8	—	—	↕	↕	0	—	1	
	OR.W #xx:16,Rd	W	4									Rd16∨#xx:16→Rd16	—	—	↕	↕	0	—	2	
	OR.W Rs,Rd	W		2								Rd16∨Rs16→Rd16	—	—	↕	↕	0	—	1	
	OR.L #xx:32,ERd	L	6									ERd32∨#xx:32→ERd32	—	—	↕	↕	0	—	3	
	OR.L ERs,ERd	L		4								ERd32∨ERs32→ERd32	—	—	↕	↕	0	—	2	
XOR	XOR.B #xx:8,Rd	B	2									Rd8⊕#xx:8→Rd8	—	—	↕	↕	0	—	1	
	XOR.B Rs,Rd	B		2								Rd8⊕Rs8→Rd8	—	—	↕	↕	0	—	1	
	XOR.W #xx:16,Rd	W	4									Rd16⊕#xx:16→Rd16	—	—	↕	↕	0	—	2	
	XOR.W Rs,Rd	W		2								Rd16⊕Rs16→Rd16	—	—	↕	↕	0	—	1	
	XOR.L #xx:32,ERd	L	6									ERd32⊕#xx:32→ERd32	—	—	↕	↕	0	—	3	
	XOR.L ERs,ERd	L		4								ERd32⊕ERs32→ERd32	—	—	↕	↕	0	—	2	
NOT	NOT.B Rd	B		2								¬ Rd8→Rd8	—	—	↕	↕	0	—	1	
	NOT.W Rd	W		2								¬ Rd16→Rd16	—	—	↕	↕	0	—	1	
	NOT.L ERd	L		2								¬ ERd32→ERd32	—	—	↕	↕	0	—	1	

Table A.1 Instruction Set (cont)

4. Shift Instructions

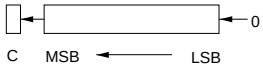
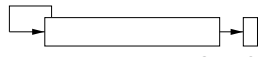
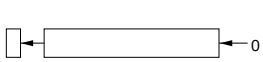
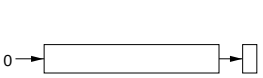
Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
SHAL	SHAL.B Rd	B		2									—	—	↕	↕	↕	↕	1	
	SHAL.B #2,Rd	B		2									—	—	↕	↕	↕	↕	1	
	SHAL.W Rd	W		2									—	—	↕	↕	↕	↕	1	
	SHAL.W #2,Rd	W		2									—	—	↕	↕	↕	↕	1	
	SHAL.L ERd	L		2									—	—	↕	↕	↕	↕	1	
	SHAL.L #2,ERd	L		2									—	—	↕	↕	↕	↕	1	
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0	↕	1	
	SHAR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	SHAR.W Rd	W		2									—	—	↕	↕	0	↕	1	
	SHAR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	SHAR.L ERd	L		2									—	—	↕	↕	0	↕	1	
	SHAR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0	↕	1	
	SHLL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	SHLL.W Rd	W		2									—	—	↕	↕	0	↕	1	
	SHLL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	SHLL.L ERd	L		2									—	—	↕	↕	0	↕	1	
	SHLL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	
SHLR	SHLR.B Rd	B		2									—	—	↕	↕	0	↕	1	
	SHLR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	SHLR.W Rd	W		2									—	—	↕	↕	0	↕	1	
	SHLR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	SHLR.L ERd	L		2									—	—	↕	↕	0	↕	1	
	SHLR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	
ROTXL	ROTXL.B Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTXL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTXL.W Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTXL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTXL.L ERd	L		2									—	—	↕	↕	0	↕	1	
	ROTXL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	

Table A.1 Instruction Set (cont)

4. Shift Instructions

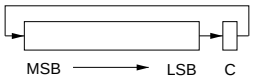
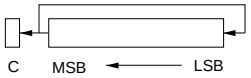
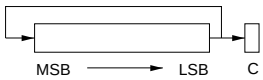
Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
ROTXR	ROTXR.B Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTXR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTXR.W Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTXR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTXR.L ERd	L		2									—	—	↕	↕	0	↕	1	
	ROTXR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	
ROTL	ROTL.B Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTL.W Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTL.L ERd	L		2									—	—	↕	↕	0	↕	1	
	ROTL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	
ROTR	ROTR.B Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1	
	ROTR.W Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1	
	ROTR.L ERd	L		2									—	—	↕	↕	0	↕	1	
	ROTR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1	

Table A.1 Instruction Set (cont)

5. Bit Manipulation Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
BSET	BSET #xx:3,Rd	B		2								(#xx:3 of Rd8)←1	—	—	—	—	—	—	1	
	BSET #xx:3,@ERd	B			4							(#xx:3 of @ERd)←1	—	—	—	—	—	—	4	
	BSET #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←1	—	—	—	—	—	—	4	
	BSET #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←1	—	—	—	—	—	—	5	
	BSET #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←1	—	—	—	—	—	—	6	
	BSET Rn,Rd	B		2								(Rn8 of Rd8)←1	—	—	—	—	—	—	1	
	BSET Rn,@ERd	B			4							(Rn8 of @ERd)←1	—	—	—	—	—	—	4	
	BSET Rn,@aa:8	B						4				(Rn8 of @aa:8)←1	—	—	—	—	—	—	4	
	BSET Rn,@aa:16	B						6				(Rn8 of @aa:16)←1	—	—	—	—	—	—	5	
	BSET Rn,@aa:32	B						8				(Rn8 of @aa:32)←1	—	—	—	—	—	—	6	
BCLR	BCLR #xx:3,Rd	B		2								(#xx:3 of Rd8)←0	—	—	—	—	—	—	1	
	BCLR #xx:3,@ERd	B			4							(#xx:3 of @ERd)←0	—	—	—	—	—	—	4	
	BCLR #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←0	—	—	—	—	—	—	4	
	BCLR #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←0	—	—	—	—	—	—	5	
	BCLR #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←0	—	—	—	—	—	—	6	
	BCLR Rn,Rd	B		2								(Rn8 of Rd8)←0	—	—	—	—	—	—	1	
	BCLR Rn,@ERd	B			4							(Rn8 of @ERd)←0	—	—	—	—	—	—	4	
	BCLR Rn,@aa:8	B						4				(Rn8 of @aa:8)←0	—	—	—	—	—	—	4	
	BCLR Rn,@aa:16	B						6				(Rn8 of @aa:16)←0	—	—	—	—	—	—	5	
	BCLR Rn,@aa:32	B						8				(Rn8 of @aa:32)←0	—	—	—	—	—	—	6	
BNOT	BNOT #xx:3,Rd	B		2								(#xx:3 of Rd8)← [¬ (#xx:3 of Rd8)]	—	—	—	—	—	—	1	
	BNOT #xx:3,@ERd	B			4							(#xx:3 of @ERd)← [¬ (#xx:3 of @ERd)]	—	—	—	—	—	—	4	
	BNOT #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)← [¬ (#xx:3 of @aa:8)]	—	—	—	—	—	—	4	
	BNOT #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)← [¬ (#xx:3 of @aa:16)]	—	—	—	—	—	—	5	
	BNOT #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)← [¬ (#xx:3 of @aa:32)]	—	—	—	—	—	—	6	
	BNOT Rn,Rd	B		2								(Rn8 of Rd8)← [¬ (Rn8 of Rd8)]	—	—	—	—	—	—	1	
	BNOT Rn,@ERd	B			4							(Rn8 of @ERd)← [¬ (Rn8 of @ERd)]	—	—	—	—	—	—	4	
	BNOT Rn,@aa:8	B						4				(Rn8 of @aa:8)← [¬ (Rn8 of @aa:8)]	—	—	—	—	—	—	4	
	BNOT Rn,@aa:16	B						6				(Rn8 of @aa:16)← [¬ (Rn8 of @aa:16)]	—	—	—	—	—	—	5	
	BNOT Rn,@aa:32	B						8				(Rn8 of @aa:32)← [¬ (Rn8 of @aa:32)]	—	—	—	—	—	—	6	

Table A.1 Instruction Set (cont)

5. Bit Manipulation Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
BTST	BTST #xx:3,Rd	B		2								¬ (#xx:3 of Rd8)→Z	—	—	—	↕	—	—	1	
	BTST #xx:3,@ERd	B			4							¬ (#xx:3 of @ERd)→Z	—	—	—	↕	—	—	3	
	BTST #xx:3,@aa:8	B						4				¬ (#xx:3 of @aa:8)→Z	—	—	—	↕	—	—	3	
	BTST #xx:3,@aa:16	B						6				¬ (#xx:3 of @aa:16)→Z	—	—	—	↕	—	—	4	
	BTST #xx:3,@aa:32	B						8				¬ (#xx:3 of @aa:32)→Z	—	—	—	↕	—	—	5	
	BTST Rn,Rd	B		2								¬ (Rn8 of Rd8)→Z	—	—	—	↕	—	—	1	
	BTST Rn,@ERd	B			4							¬ (Rn8 of @ERd)→Z	—	—	—	↕	—	—	3	
	BTST Rn,@aa:8	B						4				¬ (Rn8 of @aa:8)→Z	—	—	—	↕	—	—	3	
	BTST Rn,@aa:16	B						6				¬ (Rn8 of @aa:16)→Z	—	—	—	↕	—	—	4	
	BTST Rn,@aa:32	B						8				¬ (Rn8 of @aa:32)→Z	—	—	—	↕	—	—	5	
BLD	BLD #xx:3,Rd	B		2								(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BLD #xx:3,@ERd	B			4							(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BLD #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BLD #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BLD #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	
BILD	BILD #xx:3,Rd	B		2								¬ (#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BILD #xx:3,@ERd	B			4							¬ (#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BILD #xx:3,@aa:8	B						4				¬ (#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BILD #xx:3,@aa:16	B						6				¬ (#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BILD #xx:3,@aa:32	B						8				¬ (#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	
BST	BST #xx:3,Rd	B		2								C→(#xx:3 of Rd8)	—	—	—	—	—	—	1	
	BST #xx:3,@ERd	B			4							C→(#xx:3 of @ERd)	—	—	—	—	—	—	4	
	BST #xx:3,@aa:8	B						4				C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4	
	BST #xx:3,@aa:16	B						6				C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5	
	BST #xx:3,@aa:32	B						8				C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6	
BIST	BIST #xx:3,Rd	B		2								¬ C→(#xx:3 of Rd8)	—	—	—	—	—	—	1	
	BIST #xx:3,@ERd	B			4							¬ C→(#xx:3 of @ERd)	—	—	—	—	—	—	4	
	BIST #xx:3,@aa:8	B						4				¬ C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4	
	BIST #xx:3,@aa:16	B						6				¬ C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5	
	BIST #xx:3,@aa:32	B						8				¬ C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6	

Table A.1 Instruction Set (cont)

5. Bit Manipulation Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Normal	Advanced
BAND	BAND #xx:3,Rd	B		2								C∧ (#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BAND #xx:3,@ERd	B			4							C∧ (#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BAND #xx:3,@aa:8	B						4				C∧ (#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BAND #xx:3,@aa:16	B						6				C∧ (#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BAND #xx:3,@aa:32	B						8				C∧ (#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	
BIAND	BIAND #xx:3,Rd	B		2								C∧ [¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1	
	BIAND #xx:3,@ERd	B			4							C∧ [¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3	
	BIAND #xx:3,@aa:8	B						4				C∧ [¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3	
	BIAND #xx:3,@aa:16	B						6				C∧ [¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4	
	BIAND #xx:3,@aa:32	B						8				C∧ [¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5	
BOR	BOR #xx:3,Rd	B		2								C∨ (#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BOR #xx:3,@ERd	B			4							C∨ (#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BOR #xx:3,@aa:8	B						4				C∨ (#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BOR #xx:3,@aa:16	B						6				C∨ (#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BOR #xx:3,@aa:32	B						8				C∨ (#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	
BIOR	BIOR #xx:3,Rd	B		2								C∨ [¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1	
	BIOR #xx:3,@ERd	B			4							C∨ [¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3	
	BIOR #xx:3,@aa:8	B						4				C∨ [¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3	
	BIOR #xx:3,@aa:16	B						6				C∨ [¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4	
	BIOR #xx:3,@aa:32	B						8				C∨ [¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5	
BXOR	BXOR #xx:3,Rd	B		2								C⊕ (#xx:3 of Rd8)→C	—	—	—	—	—	↕	1	
	BXOR #xx:3,@ERd	B			4							C⊕ (#xx:3 of @ERd)→C	—	—	—	—	—	↕	3	
	BXOR #xx:3,@aa:8	B						4				C⊕ (#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3	
	BXOR #xx:3,@aa:16	B						6				C⊕ (#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4	
	BXOR #xx:3,@aa:32	B						8				C⊕ (#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5	
BIXOR	BIXOR #xx:3,Rd	B		2								C⊕ [¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1	
	BIXOR #xx:3,@ERd	B			4							C⊕ [¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3	
	BIXOR #xx:3,@aa:8	B						4				C⊕ [¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3	
	BIXOR #xx:3,@aa:16	B						6				C⊕ [¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4	
	BIXOR #xx:3,@aa:32	B						8				C⊕ [¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5	

Table A.1 Instruction Set (cont)

6. Branch Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Branch Condition	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|			I	H	N	Z	V	C	Normal	Advanced
Bcc	BRA d:8(BT d:8)	—							2			if condition is true then PC←PC+d else next;	Always	—	—	—	—	—	—	2	
	BRA d:16(BT d:16)	—							4					—	—	—	—	—	—	3	
	BRN d:8(BF d:8)	—							2				Never	—	—	—	—	—	—	2	
	BRN d:16(BF d:16)	—							4					—	—	—	—	—	—	3	
	BHI d:8	—							2				C∨Z=0	—	—	—	—	—	—	2	
	BHI d:16	—							4					—	—	—	—	—	—	3	
	BLS d:8	—							2				C∨Z=1	—	—	—	—	—	—	2	
	BLS d:16	—							4					—	—	—	—	—	—	3	
	BCC d:8(BHS d:8)	—							2				C=0	—	—	—	—	—	—	2	
	BCC d:16(BHS d:16)	—							4					—	—	—	—	—	—	3	
	BCS d:8(BLO d:8)	—							2				C=1	—	—	—	—	—	—	2	
	BCS d:16(BLO d:16)	—							4					—	—	—	—	—	—	3	
	BNE d:8	—							2				Z=0	—	—	—	—	—	—	2	
	BNE d:16	—							4					—	—	—	—	—	—	3	
	BEQ d:8	—							2				Z=1	—	—	—	—	—	—	2	
	BEQ d:16	—							4					—	—	—	—	—	—	3	
	BVC d:8	—							2				V=0	—	—	—	—	—	—	2	
	BVC d:16	—							4					—	—	—	—	—	—	3	
	BVS d:8	—							2				V=1	—	—	—	—	—	—	2	
	BVS d:16	—							4					—	—	—	—	—	—	3	
	BPL d:8	—							2				N=0	—	—	—	—	—	—	2	
	BPL d:16	—							4					—	—	—	—	—	—	3	
	BMI d:8	—							2				N=1	—	—	—	—	—	—	2	
	BMI d:16	—							4					—	—	—	—	—	—	3	
	BGE d:8	—							2				N⊕V=0	—	—	—	—	—	—	2	
	BGE d:16	—							4					—	—	—	—	—	—	3	
	BLT d:8	—							2				N⊕V=1	—	—	—	—	—	—	2	
	BLT d:16	—							4					—	—	—	—	—	—	3	
	BGT d:8	—							2				Z∨(N⊕V)=0	—	—	—	—	—	—	2	
	BGT d:16	—							4					—	—	—	—	—	—	3	
	BLE d:8	—							2				Z∨(N⊕V)=1	—	—	—	—	—	—	2	
	BLE d:16	—							4					—	—	—	—	—	—	3	

Table A.1 Instruction Set (cont)

6. Branch Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
JMP	JMP @ERn	—			2							PC←ERn	—	—	—	—	—	—	2	
	JMP @aa:24	—						4				PC←aa:24	—	—	—	—	—	—	3	
	JMP @@aa:8	—								2		PC←@aa:8	—	—	—	—	—	—	4	5
BSR	BSR d:8	—							2			PC→@-SP,PC←PC+d:8	—	—	—	—	—	—	3	4
	BSR d:16	—							4			PC→@-SP,PC←PC+d:16	—	—	—	—	—	—	4	5
JSR	JSR @ERn	—			2							PC→@-SP,PC←ERn	—	—	—	—	—	—	3	4
	JSR @aa:24	—						4				PC→@-SP,PC←aa:24	—	—	—	—	—	—	4	5
	JSR @@aa:8	—								2		PC→@-SP,PC←@aa:8	—	—	—	—	—	—	4	6
RTS	RTS	—									2	PC←@SP+	—	—	—	—	—	—	4	5

Table A.1 Instruction Set (cont)

7. System Control Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
TRAPA	TRAPA #xx:2	—										PC→@-SP,CCR→@-SP, EXR→@-SP,<vector>→PC	1	—	—	—	—	—	7 [9]	8 [9]
RTE	RTE	—										EXR←@SP+,CCR←@SP+, PC←@SP+	↕	↕	↕	↕	↕	↕	5 [9]	
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—	—	—	2	
LDC	LDC #xx:8,CCR	B	2									#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	LDC #xx:8,EXR	B	4									#xx:8→EXR	—	—	—	—	—	—	2	
	LDC Rs,CCR	B		2								Rs8→CCR	↕	↕	↕	↕	↕	↕	1	
	LDC Rs,EXR	B		2								Rs8→EXR	—	—	—	—	—	—	1	
	LDC @ERs,CCR	W			4							@ERs→CCR	↕	↕	↕	↕	↕	↕	3	
	LDC @ERs,EXR	W			4							@ERs→EXR	—	—	—	—	—	—	3	
	LDC @(d:16,ERs),CCR	W				6						@(d:16,ERs)→CCR	↕	↕	↕	↕	↕	↕	4	
	LDC @(d:16,ERs),EXR	W				6						@(d:16,ERs)→EXR	—	—	—	—	—	—	4	
	LDC @(d:32,ERs),CCR	W				10						@(d:32,ERs)→CCR	↕	↕	↕	↕	↕	↕	6	
	LDC @(d:32,ERs),EXR	W				10						@(d:32,ERs)→EXR	—	—	—	—	—	—	6	
	LDC @ERs+,CCR	W					4					@ERs→CCR,ERs32+2→ERs32	↕	↕	↕	↕	↕	↕	4	
	LDC @ERs+,EXR	W					4					@ERs→EXR,ERs32+2→ERs32	—	—	—	—	—	—	4	
	LDC @aa:16,CCR	W						6				@aa:16→CCR	↕	↕	↕	↕	↕	↕	4	
	LDC @aa:16,EXR	W						6				@aa:16→EXR	—	—	—	—	—	—	4	
	LDC @aa:32,CCR	W						8				@aa:32→CCR	↕	↕	↕	↕	↕	↕	5	
	LDC @aa:32,EXR	W						8				@aa:32→EXR	—	—	—	—	—	—	5	

Table A.1 Instruction Set (cont)

7. System Control Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
STC	STC CCR,Rd	B		2								CCR→Rd8	—	—	—	—	—	—	1	
	STC EXR,Rd	B		2								EXR→Rd8	—	—	—	—	—	—	1	
	STC CCR,@ERd	W			4							CCR→@ERd	—	—	—	—	—	—	3	
	STC EXR,@ERd	W			4							EXR→@ERd	—	—	—	—	—	—	3	
	STC CCR,@(d:16,ERd)	W				6						CCR→@(d:16,ERd)	—	—	—	—	—	—	4	
	STC EXR,@(d:16,ERd)	W				6						EXR→@(d:16,ERd)	—	—	—	—	—	—	4	
	STC CCR,@(d:32,ERd)	W				10						CCR→@(d:32,ERd)	—	—	—	—	—	—	6	
	STC EXR,@(d:32,ERd)	W				10						EXR→@(d:32,ERd)	—	—	—	—	—	—	6	
	STC CCR,@-ERd	W					4					ERd32-2→ERd32,CCR→@ERd	—	—	—	—	—	—	4	
	STC EXR,@-ERd	W					4					ERd32-2→ERd32,EXR→@ERd	—	—	—	—	—	—	4	
	STC CCR,@aa:16	W						6				CCR→@aa:16	—	—	—	—	—	—	4	
	STC EXR,@aa:16	W						6				EXR→@aa:16	—	—	—	—	—	—	4	
	STC CCR,@aa:32	W						8				CCR→@aa:32	—	—	—	—	—	—	5	
	STC EXR,@aa:32	W						8				EXR→@aa:32	—	—	—	—	—	—	5	
ANDC	ANDC #xx:8,CCR	B	2									CCR∧#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	ANDC #xx:8,EXR	B	4									EXR∧#xx:8→EXR	—	—	—	—	—	—	2	
ORC	ORC #xx:8,CCR	B	2									CCR∨#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	ORC #xx:8,EXR	B	4									EXR∨#xx:8→EXR	—	—	—	—	—	—	2	
XORC	XORC #xx:8,CCR	B	2									CCR⊕#xx:8→CCR	↕	↕	↕	↕	↕	↕	1	
	XORC #xx:8,EXR	B	4									EXR⊕#xx:8→EXR	—	—	—	—	—	—	2	
NOP	NOP	—									2	PC←PC+2	—	—	—	—	—	—	1	

Table A.1 Instruction Set (cont)

8. Block Transfer Instructions

Mnemonic		Size	Addressing Mode and Instruction Length (Bytes)									Operation	Condition Code						No. of States*1	
			#xx	Rn	@ERn	@(d,ERn)	@-ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
EEPMOV	EEPMOV.B	—									4	if R4L≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4L-1→R4L Until R4L=0 else next;	—	—	—	—	—	—	4+2n*2	
	EEPMOV.W	—									4	if R4≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4-1→R4 Until R4=0 else next;	—	—	—	—	—	—	4+2n*2	

Notes: 1. The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory.

2. n is the initial value set in R4L or R4.

[1] 7 states when the number of saved/restored registers is 2, 9 states when 3, and 11 states when 4.

[2] Cannot be used with the H8S/2128 Series and H8S/2124 Series.

[3] Set to 1 when there is a carry from or borrow to bit 11; otherwise cleared to 0.

[4] Set to 1 when there is a carry from or borrow to bit 27; otherwise cleared to 0.

[5] If the result is zero, the previous value of the flag is retained; otherwise the flag is cleared to 0.

[6] Set to 1 if the divisor is negative; otherwise cleared to 0.

[7] Set to 1 if the divisor is zero; otherwise cleared to 0.

[8] Set to 1 if the quotient is negative; otherwise cleared to 0.

[9] When EXR is valid, the number of states is increased by 1.

A.2 Instruction Codes

Table A.2 Instruction Codes

Instruction	Mnemonic	Size	Instruction Format									
			1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7th Byte	8th Byte	9th Byte	10th Byte
ADD	ADD.B #xx:8,Rd	B	8 rd	IMM								
	ADD.B Rs,Rd	B	0 8	rs rd								
	ADD.W #xx:16,Rd	W	7 9	1 rd	IMM							
	ADD.W Rs,Rd	W	0 9	rs rd								
	ADD.L #xx:32,ERd	L	7 A	1 0 erd	IMM							
	ADD.L ERs,ERd	L	0 A	1 ers 0 erd								
ADDS	ADDS #1,ERd	L	0 B	0 0 erd								
	ADDS #2,ERd	L	0 B	8 0 erd								
	ADDS #4,ERd	L	0 B	9 0 erd								
ADDX	ADDX #xx:8,Rd	B	9 rd	IMM								
	ADDX Rs,Rd	B	0 E	rs rd								
AND	AND.B #xx:8,Rd	B	E rd	IMM								
	AND.B Rs,Rd	B	1 6	rs rd								
	AND.W #xx:16,Rd	W	7 9	6 rd	IMM							
	AND.W Rs,Rd	W	6 6	rs rd								
	AND.L #xx:32,ERd	L	7 A	6 0 erd	IMM							
	AND.L ERs,ERd	L	0 1	F 0	6	6 0 ers 0 erd						
ANDC	ANDC #xx:8,CCR	B	0 6	IMM								
	ANDC #xx:8,EXR	B	0 1	4 1	0	6 IMM						
BAND	BAND #xx:3,Rd	B	7 6	0 IMM rd								
	BAND #xx:3,@ERd	B	7 C	0 erd 0	7	6 0 IMM 0						
	BAND #xx:3,@aa:8	B	7 E	abs	7	6 0 IMM 0						
	BAND #xx:3,@aa:16	B	6 A	1 0	abs		7 6	0 IMM 0				
	BAND #xx:3,@aa:32	B	6 A	3 0	abs				7 6	0 IMM 0		
Bcc	BRA d:8 (BT d:8)	—	4 0	disp								
	BRA d:16 (BT d:16)	—	5 8	0 0	disp							
	BRN d:8 (BF d:8)	—	4 1	disp								
	BRN d:16 (BF d:16)	—	5 8	1 0	disp							

Table A.2 Instruction Codes (cont)

Instruction	Mnemonic	Size	Instruction Format									
			1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7th Byte	8th Byte	9th Byte	10th Byte
Bcc	BHI d:8	—	4 2	disp								
	BHI d:16	—	5 8	2 0	disp							
	BLS d:8	—	4 3	disp								
	BLS d:16	—	5 8	3 0	disp							
	BCC d:8 (BHS d:8)	—	4 4	disp								
	BCC d:16 (BHS d:16)	—	5 8	4 0	disp							
	BCS d:8 (BLO d:8)	—	4 5	disp								
	BCS d:16 (BLO d:16)	—	5 8	5 0	disp							
	BNE d:8	—	4 6	disp								
	BNE d:16	—	5 8	6 0	disp							
	BEQ d:8	—	4 7	disp								
	BEQ d:16	—	5 8	7 0	disp							
	BVC d:8	—	4 8	disp								
	BVC d:16	—	5 8	8 0	disp							
	BVS d:8	—	4 9	disp								
	BVS d:16	—	5 8	9 0	disp							
	BPL d:8	—	4 A	disp								
	BPL d:16	—	5 8	A 0	disp							
	BMI d:8	—	4 B	disp								
	BMI d:16	—	5 8	B 0	disp							
	BGE d:8	—	4 C	disp								
	BGE d:16	—	5 8	C 0	disp							
	BLT d:8	—	4 D	disp								
	BLT d:16	—	5 8	D 0	disp							
	BGT d:8	—	4 E	disp								
	BGT d:16	—	5 8	E 0	disp							
	BLE d:8	—	4 F	disp								
	BLE d:16	—	5 8	F 0	disp							

Table A.2 Instruction Codes (cont)

Instruc-tion	Mnemonic	Size	Instruction Format																			
			1st Byte		2nd Byte		3rd Byte		4th Byte		5th Byte		6th Byte		7th Byte		8th Byte		9th Byte		10th Byte	
BCLR	BCLR #xx:3,Rd	B	7	2	0:IMM	rd																
	BCLR #xx:3,@ERd	B	7	D	0: erd	0	7	2	0:IMM	0												
	BCLR #xx:3,@aa:8	B	7	F	abs		7	2	0:IMM	0												
	BCLR #xx:3,@aa:16	B	6	A	1	8	abs				7	2	0:IMM	0								
	BCLR #xx:3,@aa:32	B	6	A	3	8	abs								7	2	0:IMM	0				
	BCLR Rn,Rd	B	6	2	rn	rd																
	BCLR Rn,@ERd	B	7	D	0: erd	0	6	2	rn	0												
	BCLR Rn,@aa:8	B	7	F	abs		6	2	rn	0												
	BCLR Rn,@aa:16	B	6	A	1	8	abs				6	2	rn	0								
	BCLR Rn,@aa:32	B	6	A	3	8	abs								6	2	rn	0				
BIAND	BIAND #xx:3,Rd	B	7	6	1:IMM	rd																
	BIAND #xx:3,@ERd	B	7	C	0: erd	0	7	6	1:IMM	0												
	BIAND #xx:3,@aa:8	B	7	E	abs		7	6	1:IMM	0												
	BIAND #xx:3,@aa:16	B	6	A	1	0	abs				7	6	1:IMM	0								
	BIAND #xx:3,@aa:32	B	6	A	3	0	abs								7	6	1:IMM	0				
BILD	BILD #xx:3,Rd	B	7	7	1:IMM	rd																
	BILD #xx:3,@ERd	B	7	C	0: erd	0	7	7	1:IMM	0												
	BILD #xx:3,@aa:8	B	7	E	abs		7	7	1:IMM	0												
	BILD #xx:3,@aa:16	B	6	A	1	0	abs				7	7	1:IMM	0								
	BILD #xx:3,@aa:32	B	6	A	3	0	abs								7	7	1:IMM	0				
BIOR	BIOR #xx:3,Rd	B	7	4	1:IMM	rd																
	BIOR #xx:3,@ERd	B	7	C	0: erd	0	7	4	1:IMM	0												
	BIOR #xx:3,@aa:8	B	7	E	abs		7	4	1:IMM	0												
	BIOR #xx:3,@aa:16	B	6	A	1	0	abs				7	4	1:IMM	0								
	BIOR #xx:3,@aa:32	B	6	A	3	0	abs								7	4	1:IMM	0				

Table A.2 Instruction Codes (cont)

Instruction	Mnemonic	Size	Instruction Format																					
			1st Byte		2nd Byte		3rd Byte		4th Byte		5th Byte		6th Byte		7th Byte		8th Byte		9th Byte		10th Byte	
BIST	BIST #xx:3,Rd	B	6	7	1:IMM:	rd																
	BIST #xx:3,@ERd	B	7	D	0: erd	0	6	7	1:IMM:	0												
	BIST #xx:3,@aa:8	B	7	F	abs		6	7	1:IMM:	0												
	BIST #xx:3,@aa:16	B	6	A	1	8	abs				6	7	1:IMM:	0								
	BIST #xx:3,@aa:32	B	6	A	3	8	abs								6	7	1:IMM:	0				
BIXOR	BIXOR #xx:3,Rd	B	7	5	1:IMM:	rd																
	BIXOR #xx:3,@ERd	B	7	C	0: erd	0	7	5	1:IMM:	0												
	BIXOR #xx:3,@aa:8	B	7	E	abs		7	5	1:IMM:	0												
	BIXOR #xx:3,@aa:16	B	6	A	1	0	abs				7	5	1:IMM:	0								
	BIXOR #xx:3,@aa:32	B	6	A	3	0	abs								7	5	1:IMM:	0				
BLD	BLD #xx:3,Rd	B	7	7	0:IMM:	rd																
	BLD #xx:3,@ERd	B	7	C	0: erd	0	7	7	0:IMM:	0												
	BLD #xx:3,@aa:8	B	7	E	abs		7	7	0:IMM:	0												
	BLD #xx:3,@aa:16	B	6	A	1	0	abs				7	7	0:IMM:	0								
	BLD #xx:3,@aa:32	B	6	A	3	0	abs								7	7	0:IMM:	0				
BNOT	BNOT #xx:3,Rd	B	7	1	0:IMM:	rd																
	BNOT #xx:3,@ERd	B	7	D	0: erd	0	7	1	0:IMM:	0												
	BNOT #xx:3,@aa:8	B	7	F	abs		7	1	0:IMM:	0												
	BNOT #xx:3,@aa:16	B	6	A	1	8	abs				7	1	0:IMM:	0								
	BNOT #xx:3,@aa:32	B	6	A	3	8	abs								7	1	0:IMM:	0				
	BNOT Rn,Rd	B	6	1	rn	rd																
	BNOT Rn,@ERd	B	7	D	0: erd	0	6	1	rn	0												
	BNOT Rn,@aa:8	B	7	F	abs		6	1	rn	0												
	BNOT Rn,@aa:16	B	6	A	1	8	abs				6	1	rn	0								
	BNOT Rn,@aa:32	B	6	A	3	8	abs								6	1	rn	0				

Table A.2 Instruction Codes (cont)

Instruction	Mnemonic	Size	Instruction Format									
			1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7th Byte	8th Byte	9th Byte	10th Byte
BOR	BOR #xx:3,Rd	B	7 4	0 IMM rd								
	BOR #xx:3,@ERd	B	7 C	0 erd 0	7 4	0 IMM 0						
	BOR #xx:3,@aa:8	B	7 E	abs	7 4	0 IMM 0						
	BOR #xx:3,@aa:16	B	6 A	1 0	abs		7 4	0 IMM 0				
	BOR #xx:3,@aa:32	B	6 A	3 0	abs				7 4	0 IMM 0		
BSET	BSET #xx:3,Rd	B	7 0	0 IMM rd								
	BSET #xx:3,@ERd	B	7 D	0 erd 0	7 0	0 IMM 0						
	BSET #xx:3,@aa:8	B	7 F	abs	7 0	0 IMM 0						
	BSET #xx:3,@aa:16	B	6 A	1 8	abs		7 0	0 IMM 0				
	BSET #xx:3,@aa:32	B	6 A	3 8	abs				7 0	0 IMM 0		
	BSET Rn,Rd	B	6 0	rn rd								
	BSET Rn,@ERd	B	7 D	0 erd 0	6 0	rn 0						
	BSET Rn,@aa:8	B	7 F	abs	6 0	rn 0						
	BSET Rn,@aa:16	B	6 A	1 8	abs		6 0	rn 0				
	BSET Rn,@aa:32	B	6 A	3 8	abs				6 0	rn 0		
BSR	BSR d:8	—	5 5	disp								
	BSR d:16	—	5 C	0 0	disp							
BST	BST #xx:3,Rd	B	6 7	0 IMM rd								
	BST #xx:3,@ERd	B	7 D	0 erd 0	6 7	0 IMM 0						
	BST #xx:3,@aa:8	B	7 F	abs	6 7	0 IMM 0						
	BST #xx:3,@aa:16	B	6 A	1 8	abs		6 7	0 IMM 0				
	BST #xx:3,@aa:32	B	6 A	3 8	abs				6 7	0 IMM 0		
BTST	BTST #xx:3,Rd	B	7 3	0 IMM rd								
	BTST #xx:3,@ERd	B	7 C	0 erd 0	7 3	0 IMM 0						
	BTST #xx:3,@aa:8	B	7 E	abs	7 3	0 IMM 0						
	BTST #xx:3,@aa:16	B	6 A	1 0	abs		7 3	0 IMM 0				
	BTST #xx:3,@aa:32	B	6 A	3 0	abs				7 3	0 IMM 0		
	BTST Rn,Rd	B	6 3	rn rd								
	BTST Rn,@ERd	B	7 C	0 erd 0	6 3	rn 0						

Table A.2 Instruction Codes (cont)

Instruction	Mnemonic	Size	Instruction Format																			
			1st Byte		2nd Byte		3rd Byte		4th Byte		5th Byte		6th Byte		7th Byte		8th Byte		9th Byte		10th Byte	
BTST	BTST Rn,@aa:8	B	7	E	abs		6	3	rn	0												
	BTST Rn,@aa:16	B	6	A	1	0	abs				6	3	m	0								
	BTST Rn,@aa:32	B	6	A	3	0	abs								6	3	rn	0				
BXOR	BXOR #xx:3,Rd	B	7	5	0	IMM	rd															
	BXOR #xx:3,@ERd	B	7	C	0	erd	0	7	5	0	IMM	0										
	BXOR #xx:3,@aa:8	B	7	E	abs		7	5	0	IMM	0											
	BXOR #xx:3,@aa:16	B	6	A	1	0	abs				7	5	0	IMM	0							
	BXOR #xx:3,@aa:32	B	6	A	3	0	abs								7	5	0	IMM	0			
CLRMAC	CLRMAC	—	Cannot be used with the H8S/2128 Series and H8S/2124 Series.																			
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											

Table A.2 Instruction Codes (cont)

Instruc-tion	Mnemonic	Size	Instruction Format											
			1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7th Byte	8th Byte	9th Byte	10th Byte		
EXTS	EXTS.W Rd	W	1 7	D rd										
	EXTS.L ERd	L	1 7	F 0:erd										
EXTU	EXTU.W Rd	W	1 7	5 rd										
	EXTU.L ERd	L	1 7	7 0:erd										
INC	INC.B Rd	B	0 A	0 rd										
	INC.W #1,Rd	W	0 B	5 rd										
	INC.W #2,Rd	W	0 B	D rd										
	INC.L #1,ERd	L	0 B	7 0:erd										
	INC.L #2,ERd	L	0 B	F 0:erd										
JMP	JMP @ERn	—	5 9	0:em 0										
	JMP @aa:24	—	5 A	abs										
	JMP @@aa:8	—	5 B	abs										
JSR	JSR @ERn	—	5 D	0:em 0										
	JSR @aa:24	—	5 E	abs										
	JSR @@aa:8	—	5 F	abs										
LDC	LDC #xx:8,CCR	B	0 7	IMM										
	LDC #xx:8,EXR	B	0 1	4 1	0 7	IMM								
	LDC Rs,CCR	B	0 3	0 rs										
	LDC Rs,EXR	B	0 3	1 rs										
	LDC @ERs,CCR	W	0 1	4 0	6 9	0:ers 0								
	LDC @ERs,EXR	W	0 1	4 1	6 9	0:ers 0								
	LDC @(d:16,ERs),CCR	W	0 1	4 0	6 F	0:ers 0	disp							
	LDC @(d:16,ERs),EXR	W	0 1	4 1	6 F	0:ers 0	disp							
	LDC @(d:32,ERs),CCR	W	0 1	4 0	7 8	0:ers 0	6	B	2	0	disp			
	LDC @(d:32,ERs),EXR	W	0 1	4 1	7 8	0:ers 0	6	B	2	0	disp			
	LDC @ERs+,CCR	W	0 1	4 0	6 D	0:ers 0								
	LDC @ERs+,EXR	W	0 1	4 1	6 D	0:ers 0								
	LDC @aa:16,CCR	W	0 1	4 0	6 B	0 0	abs							
	LDC @aa:16,EXR	W	0 1	4 1	6 B	0 0	abs							

Table A.2 Instruction Codes (cont)

Instruction	Mnemonic	Size	Instruction Format									
			1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7th Byte	8th Byte	9th Byte	10th Byte
LDC	LDC @aa:32,CCR	W	0 1	4 0	6	B	2 0	abs				
	LDC @aa:32,EXR	W	0 1	4 1	6	B	2 0	abs				
LDM	LDM.L @SP+, (ERn-ERn+1)	L	0 1	1 0	6	D	7 0:em+1					
	LDM.L @SP+, (ERn-ERn+2)	L	0 1	2 0	6	D	7 0:em+2					
	LDM.L @SP+, (ERn-ERn+3)	L	0 1	3 0	6	D	7 0:em+3					
LDMAC	LDMAC ERs,MACH	L	Cannot be used with the H8S/2128 Series and H8S/2124 Series.									
	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				

Table A.2 Instruction Codes (cont)

Instruc-tion	Mnemonic	Size	Instruction Format									
			1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7th Byte	8th Byte	9th Byte	10th Byte
MOV	MOV.W @ERs+,Rd	W	6 D	0 ers rd								
	MOV.W @aa:16,Rd	W	6 B	0 rd	abs							
	MOV.W @aa:32,Rd	W	6 B	2 rd		abs						
	MOV.W Rs,@ERd	W	6 9	1 erd rs								
	MOV.W Rs,@(d:16,ERd)	W	6 F	1 erd rs	disp							
	MOV.W Rs,@(d:32,ERd)	W	7 8	0 erd 0	6 B	A rs			disp			
	MOV.W Rs,@-ERd	W	6 D	1 erd rs								
	MOV.W Rs,@aa:16	W	6 B	8 rs	abs							
	MOV.W Rs,@aa:32	W	6 B	A rs		abs						
	MOV.L #xx:32,Rd	L	7 A	0 erd		IMM						
	MOV.L ERs,ERd	L	0 F	1 ers 0 erd								
	MOV.L @ERs,ERd	L	0 1	0	6 9	0 ers 0 erd						
	MOV.L @(d:16,ERs),ERd	L	0 1	0	6 F	0 ers 0 erd	disp					
	MOV.L @(d:32,ERs),ERd	L	0 1	0	7 8	0 ers 0	6 B	2 0 erd	disp			
	MOV.L @ERs+,ERd	L	0 1	0	6 D	0 ers 0 erd						
	MOV.L @aa:16 ,ERd	L	0 1	0	6 B	0 0 erd	abs					
	MOV.L @aa:32 ,ERd	L	0 1	0	6 B	2 0 erd		abs				
	MOV.L ERs,@ERd	L	0 1	0	6 9	1 erd 0 ers						
	MOV.L ERs,@(d:16,ERd)	L	0 1	0	6 F	1 erd 0 ers	disp					
	MOV.L ERs,@(d:32,ERd)*	L	0 1	0	7 8	0 erd 0	6 B	A 0 ers	disp			
	MOV.L ERs,@-ERd	L	0 1	0	6 D	1 erd 0 ers						
	MOV.L ERs,@aa:16	L	0 1	0	6 B	8 0 ers	abs					
	MOV.L ERs,@aa:32	L	0 1	0	6 B	A 0 ers		abs				
MOVFPE	MOVFPE @aa:16,Rd	B	Cannot be used with the H8S/2128 Series and H8S/2124 Series.									
MOVTPE	MOVTPE Rs,@aa:16	B										
MULXS	MULXS.B Rs,Rd	B	0 1	C 0	5 0	rs rd						
	MULXS.W Rs,ERd	W	0 1	C 0	5 2	rs 0 erd						
MULXU	MULXU.B Rs,Rd	B	5 0	rs rd								
	MULXU.W Rs,ERd	W	5 2	rs 0 erd								

Table A.2 Instruction Codes (cont)

Instruction	Mnemonic	Size	Instruction Format										
			1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7th Byte	8th Byte	9th Byte	10th Byte	
NEG	NEG.B Rd	B	1 7	8 rd									
	NEG.W Rd	W	1 7	9 rd									
	NEG.L ERd	L	1 7	B 0 erd									
NOP	NOP	—	0 0	0 0									
NOT	NOT.B Rd	B	1 7	0 rd									
	NOT.W Rd	W	1 7	1 rd									
	NOT.L ERd	L	1 7	3 0 erd									
OR	OR.B #xx:8,Rd	B	C rd	IMM									
	OR.B Rs,Rd	B	1 4	rs rd									
	OR.W #xx:16,Rd	W	7 9	4 rd	IMM								
	OR.W Rs,Rd	W	6 4	rs rd									
	OR.L #xx:32,ERd	L	7 A	4 0 erd	IMM								
	OR.L ERs,ERd	L	0 1	F 0	6 4	0 ers 0 erd							
ORC	ORC #xx:8,CCR	B	0 4	IMM									
	ORC #xx:8,EXR	B	0 1	4 1	0 4	IMM							
POP	POP.W Rn	W	6 D	7 rn									
	POP.L ERn	L	0 1	0 0	6 D	7 0 ern							
PUSH	PUSH.W Rn	W	6 D	F rn									
	PUSH.L ERn	L	0 1	0 0	6 D	F 0 ern							
ROTL	ROTL.B Rd	B	1 2	8 rd									
	ROTL.B #2, Rd	B	1 2	C rd									
	ROTL.W Rd	W	1 2	9 rd									
	ROTL.W #2, Rd	W	1 2	D rd									
	ROTL.L ERd	L	1 2	B 0 erd									
	ROTL.L #2, ERd	L	1 2	F 0 erd									

Table A.2 Instruction Codes (cont)

Instruc-tion	Mnemonic	Size	Instruction Format									
			1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7th Byte	8th Byte	9th Byte	10th Byte
ROTR	ROTR.B Rd	B	1 3	8 rd								
	ROTR.B #2, Rd	B	1 3	C rd								
	ROTR.W Rd	W	1 3	9 rd								
	ROTR.W #2, Rd	W	1 3	D rd								
	ROTR.L ERd	L	1 3	B 0 erd								
	ROTR.L #2, ERd	L	1 3	F 0 erd								
ROTXL	ROTXL.B Rd	B	1 2	0 rd								
	ROTXL.B #2, Rd	B	1 2	4 rd								
	ROTXL.W Rd	W	1 2	1 rd								
	ROTXL.W #2, Rd	W	1 2	5 rd								
	ROTXL.L ERd	L	1 2	3 0 erd								
	ROTXL.L #2, ERd	L	1 2	7 0 erd								
ROTXR	ROTXR.B Rd	B	1 3	0 rd								
	ROTXR.B #2, Rd	B	1 3	4 rd								
	ROTXR.W Rd	W	1 3	1 rd								
	ROTXR.W #2, Rd	W	1 3	5 rd								
	ROTXR.L ERd	L	1 3	3 0 erd								
	ROTXR.L #2, ERd	L	1 3	7 0 erd								
RTE	RTE	—	5 6	7 0								
RTS	RTS	—	5 4	7 0								
SHAL	SHAL.B Rd	B	1 0	8 rd								
	SHAL.B #2, Rd	B	1 0	C rd								
	SHAL.W Rd	W	1 0	9 rd								
	SHAL.W #2, Rd	W	1 0	D rd								
	SHAL.L ERd	L	1 0	B 0 erd								
	SHAL.L #2, ERd	L	1 0	F 0 erd								

Table A.2 Instruction Codes (cont)

Instruction	Mnemonic	Size	Instruction Format									
			1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7th Byte	8th Byte	9th Byte	10th Byte
SHAR	SHAR.B Rd	B	1 1	8 rd								
	SHAR.B #2, Rd	B	1 1	C rd								
	SHAR.W Rd	W	1 1	9 rd								
	SHAR.W #2, Rd	W	1 1	D rd								
	SHAR.L ERd	L	1 1	B 0 erd								
	SHAR.L #2, ERd	L	1 1	F 0 erd								
SHLL	SHLL.B Rd	B	1 0	0 rd								
	SHLL.B #2, Rd	B	1 0	4 rd								
	SHLL.W Rd	W	1 0	1 rd								
	SHLL.W #2, Rd	W	1 0	5 rd								
	SHLL.L ERd	L	1 0	3 0 erd								
	SHLL.L #2, ERd	L	1 0	7 0 erd								
SHLR	SHLR.B Rd	B	1 1	0 rd								
	SHLR.B #2, Rd	B	1 1	4 rd								
	SHLR.W Rd	W	1 1	1 rd								
	SHLR.W #2, Rd	W	1 1	5 rd								
	SHLR.L ERd	L	1 1	3 0 erd								
	SHLR.L #2, ERd	L	1 1	7 0 erd								
SLEEP	SLEEP	—	0 1	8 0								
STC	STC.B CCR,Rd	B	0 2	0 rd								
	STC.B EXR,Rd	B	0 2	1 rd								
	STC.W CCR,@ERd	W	0 1	4 0	6 9	1 erd 0						
	STC.W EXR,@ERd	W	0 1	4 1	6 9	1 erd 0						
	STC.W CCR,@(d:16,ERd)	W	0 1	4 0	6 F	1 erd 0	disp					
	STC.W EXR,@(d:16,ERd)	W	0 1	4 1	6 F	1 erd 0	disp					
	STC.W CCR,@(d:32,ERd)	W	0 1	4 0	7 8	0 erd 0	6 B	A 0	disp			
	STC.W EXR,@(d:32,ERd)	W	0 1	4 1	7 8	0 erd 0	6 B	A 0	disp			
	STC.W CCR,@-ERd	W	0 1	4 0	6 D	1 erd 0						
	STC.W EXR,@-ERd	W	0 1	4 1	6 D	1 erd 0						

Table A.2 Instruction Codes (cont)

Instruc-tion	Mnemonic	Size	Instruction Format									
			1st Byte	2nd Byte	3rd Byte	4th Byte	5th Byte	6th Byte	7th Byte	8th Byte	9th Byte	10th Byte
STC	STC.W CCR,@aa:16	W	0 1	4 0	6 B	8 0	abs					
	STC.W EXR,@aa:16	W	0 1	4 1	6 B	8 0	abs					
	STC.W CCR,@aa:32	W	0 1	4 0	6 B	A 0	abs					
	STC.W EXR,@aa:32	W	0 1	4 1	6 B	A 0	abs					
STM	STM.L (ERn-ERn+1), @-SP	L	0 1	1 0	6 D	F 0	ern					
	STM.L (ERn-ERn+2), @-SP	L	0 1	2 0	6 D	F 0	ern					
	STM.L (ERn-ERn+3), @-SP	L	0 1	3 0	6 D	F 0	ern					
STMAC	STMAC MACH,ERd	L	Cannot be used with the H8S/2128 Series and H8S/2124 Series.									
	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	8th byte	9th byte	10th byte
XORC	XORC #xx:8,CCR	B	0 5	IMM								
	XORC #xx:8,EXR	B	0 1	4 1	0 5	IMM						

Note: * Bit 7 of the 4th byte of the MOV.L ERs, @ (d:32, ERd) instruction can be either 0 or 1.

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, indicating an 8-bit or 16-bit register. rs, rd, and rn correspond to operand formats Rs, Rd, and Rn, respectively.)
ers, erd, ern, erm:	Register field (3 bits, indicating an address register or 32-bit register. ers, erd, ern, and erm correspond to operand formats ERs, ERd, ERn, and ERm, respectively.)

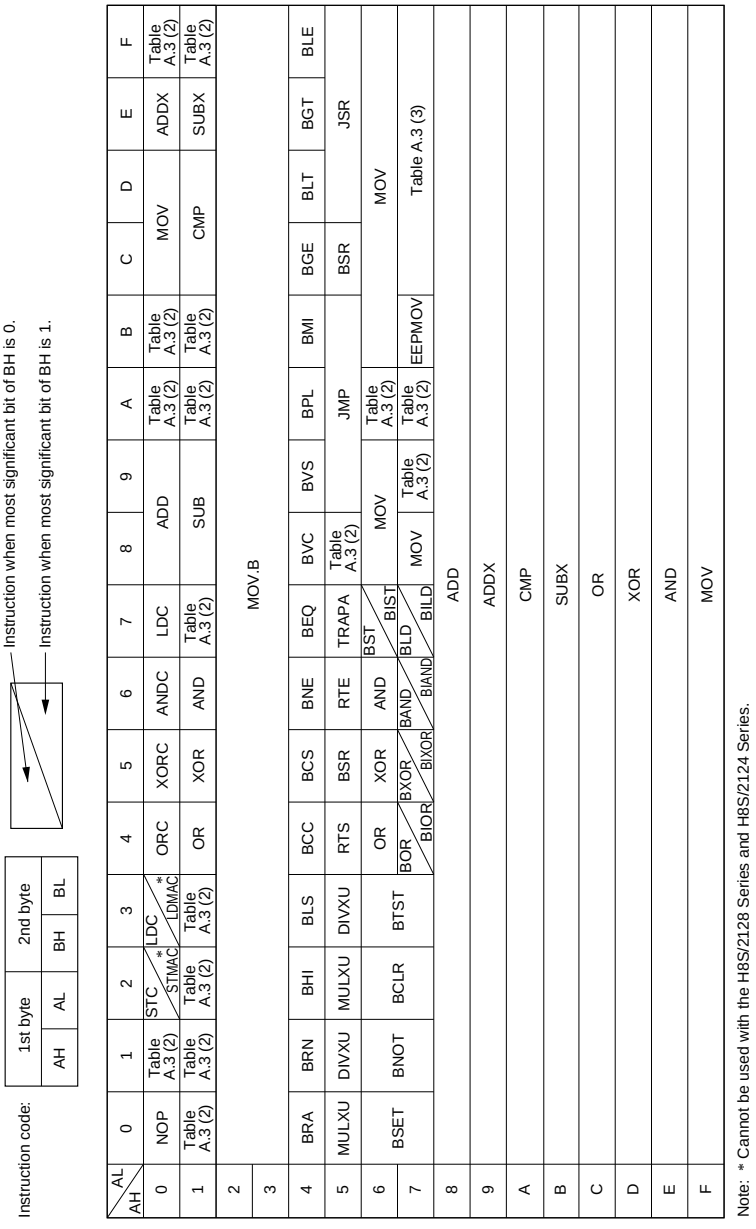
The correspondence between register fields and general registers is shown in the following table.

Address Registers 32-Bit Registers		16-Bit Register		8-Bit Register	
Register Field	General Register	Register Field	General Register	Register Field	General Register
000	ER0	0000	R0	0000	R0H
001	ER1	0001	R1	0001	R1H
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
111	ER7	0111	R7	0111	R7H
		1000	E0	1000	R0L
		1001	E1	1001	R1L
		•	•	•	•
		•	•	•	•
		•	•	•	•
		1111	E7	1111	R7L

A.3 Operation Code Map

Table A.3 shows the operation code map.

Table A.3 Operation Code Map (1)



Note: * Cannot be used with the H8S/2128 Series and H8S/2124 Series.

Table A.3 Operation Code Map (2)

Instruction code:

1st byte		2nd byte	
AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01	MOV	LDM / STM			LDC / STC		MAC*		SLEEP		CLRMAC*		Table A.3 (3)	Table A.3 (3)	TAS	Table A.3 (3)
0A	INC					INC		INC	ADD					INC		INC
0B	ADDS								ADDS							
0F	DAA								MOV							
10	SHLL				SHLL			SHLL	SHAL				SHAL			SHAL
11	SHLR				SHLR			SHLR	SHAR				SHAR			SHAR
12	ROTXL				ROTXL			ROTXL	ROTL				ROTL			ROTL
13	ROTXR				ROTXR			ROTXR	ROTR				ROTR			ROTR
17	NOT			NOT		EXTU		EXTU	NEG			NEG		EXTS		EXTS
1A	DEC								SUB							
1B	SUBS					DEC		DEC	SUBS					DEC		DEC
1F	DAS								CMP							
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
6A	MOV	Table A.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 with the H8S/2128 Series and H8S/2124 Series.

Table A.3 Operation Code Map (3)

Instruction code:

1st byte		2nd byte		3rd byte		4th byte	
AH	AL	BH	BL	CH	CL	DH	DL

Instruction when most significant bit of DH is 0.
Instruction when most significant bit of DH is 1.

AH AL BH BL CH \ CL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01C05	MULXS		MULXS													
01D05		DIVXS		DIVXS												
01F06					OR	XOR	AND									
7Cr06*1				BTST												
7Cr07*1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
7Dr06*1	BSET	BNOT	BCLR					BST / BIST								
7Dr07*1	BSET	BNOT	BCLR													
7Eaa6*2				BTST												
7Eaa7*2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
7Faa6*2	BSET	BNOT	BCLR					BST / BIST								
7Faa7*2	BSET	BNOT	BCLR													

Notes: 1. r is the register specification field.
2. aa is the absolute address specification.

Table A.3 Operation Code Map (4)

Instruction code:

1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte	
AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL

Instruction when most significant bit of FH is 0.
Instruction when most significant bit of FH is 1.

EL / AHALBHBLCHCLDHDLEH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6A10aaaa6*				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
6A10aaaa7*																
6A18aaaa6*	BSET	BNOT	BCLR					BST / BIST								
6A18aaaa7*																

Instruction code:

1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte	
AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL	GH	GL	HH	HL

Indicates case where MSB of HH is 0.
Indicates case where MSB of HH is 1.

GL / AHALBHBL ... FHFLGH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6A30aaaaaaaa6*				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
6A30aaaaaaaa7*																
6A38aaaaaaaa6*	BSET	BNOT	BCLR					BST / BIST								
6A38aaaaaaaa7*																

Note: * aa is the absolute address specification.

A.4 Number of States Required for Execution

The tables in this section can be used to calculate the number of states required for instruction execution by the H8S/2000 CPU. Table A.5 shows the number of instruction fetch, data read/write, and other cycles occurring in each instruction, and table A.4 shows the number of states required per cycle according to the bus size. The number of states required for execution of an instruction can be calculated from these two tables as follows:

$$\text{Number of states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples of Calculation of Number of States Required for Execution

Examples: Advanced mode, external address space designated for the program area and stack area, on-chip supporting modules accessed in two states with 8-bit bus width, external devices accessed in three states with one wait state and 8-bit bus width.

1. BSET #0,@FFFFC7:8
   From table A.5, I = L = 2 and J = K = M = N = 0
   From table A.4, $S_I = 8$ and $S_L = 2$
   Number of states = $2 \times 8 + 2 \times 2 = 20$
2. JSR @@30
   From table A.5, I = J = K = 2 and L = M = N = 0
   From table A.4, $S_I = S_J = S_K = 8$
   Number of states = $2 \times 8 + 2 \times 8 + 2 \times 8 = 48$

Table A.4 Number of States per Cycle

	Access Conditions						
		On-Chip		External Device			
		Supporting Module		8-Bit Bus		16-Bit Bus*	
Execution State (Cycle)	**On-Chip Memory**	**8-Bit Bus**	**16-Bit Bus**	**2-State Access**	**3-State Access**	**2-State Access**	**3-State Access**
Instruction fetch S_I	1	4	2	4	6 + 2m	2	3 + m
Branch address fetch S_J							
Stack operation S_K							
Byte data access S_L		2		2	3 + m		
Word data access S_M		4		4	6 + 2m		
Internal operation S_N	1	1	1	1	1	1	1

Legend:

m: Number of wait states inserted into external device access

Note: * Cannot be used in the H8S/2128 Series and H8S/2124 Series.

Table A.5 Number of Cycles per Instruction

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8,Rd	1					
	ADD.B Rs,Rd	1					
	ADD.W #xx:16,Rd	2					
	ADD.W Rs,Rd	1					
	ADD.L #xx:32,ERd	3					
	ADD.L ERs,ERd	1					
ADDS	ADDS #1/2/4,ERd	1					
ADDX	ADDX #xx:8,Rd	1					
	ADDX Rs,Rd	1					
AND	AND.B #xx:8,Rd	1					
	AND.B Rs,Rd	1					
	AND.W #xx:16,Rd	2					
	AND.W Rs,Rd	1					
	AND.L #xx:32,ERd	3					
	AND.L ERs,ERd	2					
ANDC	ANDC #xx:8,CCR	1					
	ANDC #xx:8,EXR	2					
BAND	BAND #xx:3,Rd	1					
	BAND #xx:3,@ERd	2			1		
	BAND #xx:3,@aa:8	2			1		
	BAND #xx:3,@aa:16	3			1		
	BAND #xx:3,@aa:32	4			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					
	BMI d:8	2					
	BGE d:8	2					
	BLT d:8	2					

Table A.5 Number of Cycles per Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
Bcc	BGT d:8	2					
	BLE d:8	2					
	BRA d:16 (BT d:16)	2					1
	BRN d:16 (BF d:16)	2					1
	BHI d:16	2					1
	BLS d:16	2					1
	BCC d:16 (BHS d:16)	2					1
	BCS d:16 (BLO d:16)	2					1
	BNE d:16	2					1
	BEQ d:16	2					1
	BVC d:16	2					1
	BVS d:16	2					1
	BPL d:16	2					1
	BMI d:16	2					1
	BGE d:16	2					1
	BLT d:16	2					1
	BGT d:16	2					1
	BLE d:16	2					1
BCLR	BCLR #xx:3,Rd	1					
	BCLR #xx:3,@ERd	2			2		
	BCLR #xx:3,@aa:8	2			2		
	BCLR #xx:3,@aa:16	3			2		
	BCLR #xx:3,@aa:32	4			2		
	BCLR Rn,Rd	1					
	BCLR Rn,@ERd	2			2		
	BCLR Rn,@aa:8	2			2		
	BCLR Rn,@aa:16	3			2		
	BCLR Rn,@aa:32	4			2		
BIAND	BIAND #xx:3,Rd	1					
	BIAND #xx:3,@ERd	2			1		
	BIAND #xx:3,@aa:8	2			1		
	BIAND #xx:3,@aa:16	3			1		
	BIAND #xx:3,@aa:32	4			1		

Table A.5 Number of Cycles per Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
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	BNOT #xx:3,Rd	1					
	BNOT #xx:3,@ERd	2			2		
	BNOT #xx:3,@aa:8	2			2		
	BNOT #xx:3,@aa:16	3			2		
	BNOT #xx:3,@aa:32	4			2		
	BNOT Rn,Rd	1					
	BNOT Rn,@ERd	2			2		
	BNOT Rn,@aa:8	2			2		
	BNOT Rn,@aa:16	3			2		
	BNOT Rn,@aa:32	4			2		

Table A.5 Number of Cycles per Instruction (cont)

Instruction	Mnemonic		Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
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	Normal	2		1			
		Advanced	2		2			
	BSR d:16	Normal	2		1			1
		Advanced	2		2			1
BST	BST #xx:3,Rd		1					
	BST #xx:3,@ERd		2			2		
	BST #xx:3,@aa:8		2			2		
	BST #xx:3,@aa:16		3			2		
	BST #xx:3,@aa:32		4			2		
BTST	BTST #xx:3,Rd		1					
	BTST #xx:3,@ERd		2			1		
	BTST #xx:3,@aa:8		2			1		
	BTST #xx:3,@aa:16		3			1		
	BTST #xx:3,@aa:32		4			1		
	BTST Rn,Rd		1					
	BTST Rn,@ERd		2			1		
	BTST Rn,@aa:8		2			1		
	BTST Rn,@aa:16		3			1		
	BTST Rn,@aa:32		4			1		

Table A.5 Number of Cycles per Instruction (cont)

Instruction	Mnemonic		Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
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 with the H8S/2128 Series and H8S/2124 Series.					
CMP	CMP.B #xx:8,Rd		1					
	CMP.B Rs,Rd		1					
	CMP.W #xx:16,Rd		2					
	CMP.W Rs,Rd		1					
	CMP.L #xx:32,ERd		3					
	CMP.L ERs,ERd		1					
DAA	DAA Rd		1					
DAS	DAS Rd		1					
DEC	DEC.B Rd		1					
	DEC.W #1/2,Rd		1					
	DEC.L #1/2,ERd		1					
DIVXS	DIVXS.B Rs,Rd		2					11
	DIVXS.W Rs,ERd		2					19
DIVXU	DIVXU.B Rs,Rd		1					11
	DIVXU.W Rs,ERd		1					19
EEPMOV	EEPMOV.B		2			2n+2 *[2]		
	EEPMOV.W		2			2n+2 *[2]		
EXTS	EXTS.W Rd		1					
	EXTS.L ERd		1					
EXTU	EXTU.W Rd		1					
	EXTU.L ERd		1					
INC	INC.B Rd		1					
	INC.W #1/2,Rd		1					
	INC.L #1/2,ERd		1					
JMP	JMP @ERn		2					
	JMP @aa:24		2					1
	JMP @@aa:8	Normal	2	1				1
		Advanced	2	2				1

Table A.5 Number of Cycles per Instruction (cont)

Instruction	Mnemonic		Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
JSR	JSR @ERn	Normal	2		1			
		Advanced	2		2			
	JSR @aa:24	Normal	2		1			1
		Advanced	2		2			1
	JSR @@aa:8	Normal	2	1	1			
		Advanced	2	2	2			
LDC	LDC #xx:8,CCR		1					
	LDC #xx:8,EXR		2					
	LDC Rs,CCR		1					
	LDC Rs,EXR		1					
	LDC @ERs,CCR		2				1	
	LDC @ERs,EXR		2				1	
	LDC @(d:16,ERs),CCR		3				1	
	LDC @(d:16,ERs),EXR		3				1	
	LDC @(d:32,ERs),CCR		5				1	
	LDC @(d:32,ERs),EXR		5				1	
	LDC @ERs+,CCR		2				1	1
	LDC @ERs+,EXR		2				1	1
	LDC @aa:16,CCR		3				1	
	LDC @aa:16,EXR		3				1	
	LDC @aa:32,CCR		4				1	
	LDC @aa:32,EXR		4				1	
LDM	LDM.L @SP+, (ERn-ERn+1)		2		4			1
	LDM.L @SP+, (ERn-ERn+2)		2		6			1
	LDM.L @SP+, (ERn-ERn+3)		2		8			1
LDMAC	LDMAC ERs, MACH		Cannot be used with the H8S/2128 Series and H8S/2124 Series.					
	LDMAC ERs, MACL							
MAC	MAC @ERn+, @ERm+							
MOV	MOV.B #xx:8,Rd		1					
	MOV.B Rs,Rd		1					
	MOV.B @ERs,Rd		1			1		
	MOV.B @(d:16,ERs),Rd		2			1		
	MOV.B @(d:32,ERs),Rd		4			1		
	MOV.B @ERs+,Rd		1			1		1
	MOV.B @aa:8,Rd		1			1		
	MOV.B @aa:16,Rd		2			1		

Table A.5 Number of Cycles per Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	MOV.B @aa:32,Rd	3			1		
	MOV.B Rs,@ERd	1			1		
	MOV.B Rs,@(d:16,ERd)	2			1		
	MOV.B Rs,@(d:32,ERd)	4			1		
	MOV.B Rs,@-ERd	1			1		1
	MOV.B Rs,@aa:8	1			1		
	MOV.B Rs,@aa:16	2			1		
	MOV.B Rs,@aa:32	3			1		
	MOV.W #xx:16,Rd	2					
	MOV.W Rs,Rd	1					
	MOV.W @ERs,Rd	1				1	
	MOV.W @(d:16,ERs),Rd	2				1	
	MOV.W @(d:32,ERs),Rd	4				1	
	MOV.W @ERs+,Rd	1				1	1
	MOV.W @aa:16,Rd	2				1	
	MOV.W @aa:32,Rd	3				1	
	MOV.W Rs,@ERd	1				1	
	MOV.W Rs,@(d:16,ERd)	2				1	
	MOV.W Rs,@(d:32,ERd)	4				1	
	MOV.W Rs,@-ERd	1				1	1
	MOV.W Rs,@aa:16	2				1	
	MOV.W Rs,@aa:32	3				1	
	MOV.L #xx:32,ERd	3					
	MOV.L ERs,ERd	1					
	MOV.L @ERs,ERd	2				2	
	MOV.L @(d:16,ERs),ERd	3				2	
	MOV.L @(d:32,ERs),ERd	5				2	
	MOV.L @ERs+,ERd	2				2	1
	MOV.L @aa:16,ERd	3				2	
	MOV.L @aa:32,ERd	4				2	
	MOV.L ERs,@ERd	2				2	
	MOV.L ERs,@(d:16,ERd)	3				2	
	MOV.L ERs,@(d:32,ERd)	5				2	
	MOV.L ERs,@-ERd	2				2	1
	MOV.L ERs,@aa:16	3				2	
	MOV.L ERs,@aa:32	4				2	

Table A.5 Number of Cycles per Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOVFPE	MOVFPE @:aa:16,Rd	Cannot be used with the H8S/2128 Series and H8S/2124 Series.					
MOVTPE	MOVTPE Rs,@:aa:16						
MULXS	MULXS.B Rs,Rd	2					11
	MULXS.W Rs,ERd	2					19
MULXU	MULXU.B Rs,Rd	1					11
	MULXU.W Rs,ERd	1					19
NEG	NEG.B Rd	1					
	NEG.W Rd	1					
	NEG.L ERd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
	NOT.W Rd	1					
	NOT.L ERd	1					
OR	OR.B #xx:8,Rd	1					
	OR.B Rs,Rd	1					
	OR.W #xx:16,Rd	2					
	OR.W Rs,Rd	1					
	OR.L #xx:32,ERd	3					
	OR.L ERs,ERd	2					
ORC	ORC #xx:8,CCR	1					
	ORC #xx:8,EXR	2					
POP	POP.W Rn	1				1	1
	POP.L ERn	2				2	1
PUSH	PUSH.W Rn	1				1	1
	PUSH.L ERn	2				2	1
ROTL	ROTL.B Rd	1					
	ROTL.B #2,Rd	1					
	ROTL.W Rd	1					
	ROTL.W #2,Rd	1					
	ROTL.L ERd	1					
	ROTL.L #2,ERd	1					

Table A.5 Number of Cycles per Instruction (cont)

Instruction	Mnemonic		Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
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	ROTXR.B Rd		1					
	ROTXR.B #2,Rd		1					
	ROTXR.W Rd		1					
	ROTXR.W #2,Rd		1					
	ROTXR.L ERd		1					
	ROTXR.L #2,ERd		1					
RTE	RTE		2		2/3 *[1]			1
RTS	RTS	Normal	2		1			1
		Advanced	2		2			1
SHAL	SHAL.B Rd		1					
	SHAL.B #2,Rd		1					
	SHAL.W Rd		1					
	SHAL.W #2,Rd		1					
	SHAL.L ERd		1					
	SHAL.L #2,ERd		1					
SHAR	SHAR.B Rd		1					
	SHAR.B #2,Rd		1					
	SHAR.W Rd		1					
	SHAR.W #2,Rd		1					
	SHAR.L ERd		1					
	SHAR.L #2,ERd		1					

Table A.5 Number of Cycles per Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
SHLL	SHLL.B Rd	1					
	SHLL.B #2,Rd	1					
	SHLL.W Rd	1					
	SHLL.W #2,Rd	1					
	SHLL.L ERd	1					
	SHLL.L #2,ERd	1					
SHLR	SHLR.B Rd	1					
	SHLR.B #2,Rd	1					
	SHLR.W Rd	1					
	SHLR.W #2,Rd	1					
	SHLR.L ERd	1					
	SHLR.L #2,ERd	1					
SLEEP	SLEEP	1					1
STC	STC.B CCR,Rd	1					
	STC.B EXR,Rd	1					
	STC.W CCR,@ERd	2				1	
	STC.W EXR,@ERd	2				1	
	STC.W CCR,@(d:16,ERd)	3				1	
	STC.W EXR,@(d:16,ERd)	3				1	
	STC.W CCR,@(d:32,ERd)	5				1	
	STC.W EXR,@(d:32,ERd)	5				1	
	STC.W CCR,@-ERd	2				1	1
	STC.W EXR,@-ERd	2				1	1
	STC.W CCR,@aa:16	3				1	
	STC.W EXR,@aa:16	3				1	
	STC.W CCR,@aa:32	4				1	
	STC.W EXR,@aa:32	4				1	
STM	STM.L (ERn-ERn+1),@-SP	2		4			1
	STM.L (ERn-ERn+2),@-SP	2		6			1
	STM.L (ERn-ERn+3),@-SP	2		8			1
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					

Table A.5 Number of Cycles per Instruction (cont)

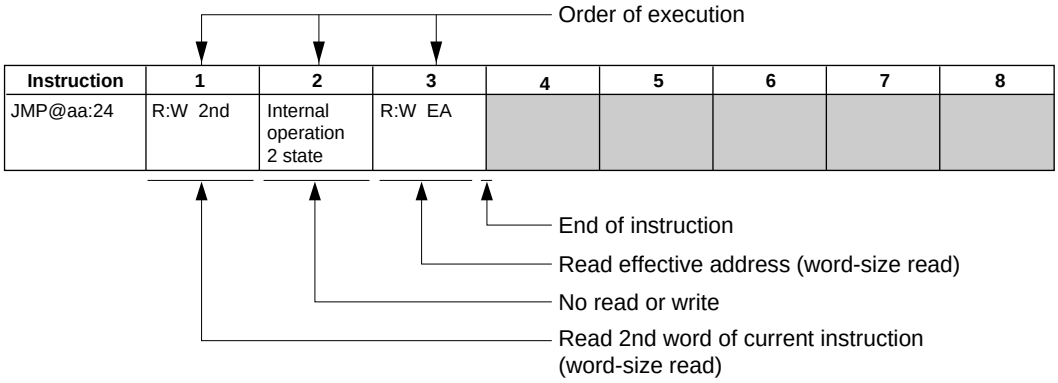
Instruction	Mnemonic		Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
SUBS	SUBS #1/2/4,ERd		1					
SUBX	SUBX #xx:8,Rd		1					
	SUBX Rs,Rd		1					
TAS	TAS @ERd		2			2		
TRAPA	TRAPA #x:2	Normal	2	1	2/3 *[1]			2
		Advanced	2	2	2/3 *[1]			2
XOR	XOR.B #xx:8,Rd		1					
	XOR.B Rs,Rd		1					
	XOR.W #xx:16,Rd		2					
	XOR.W Rs,Rd		1					
	XOR.L #xx:32,ERd		3					
	XOR.L ERs,ERd		2					
XORC	XORC #xx:8,CCR		1					
	XORC #xx:8,EXR		2					

Notes: 1. 2 when EXR is invalid, 3 when valid.
2. When n bytes of data are transferred.

A.5 Bus States During Instruction Execution

Table A.6 indicates the types of cycles that occur during instruction execution by the CPU. See table A.4 for the number of states per cycle.

How to Read the Table:



Legend

R:B	Byte-size read
R:W	Word-size read
W:B	Byte-size write
W:W	Word-size write
:M	Transfer of the bus is not performed immediately after this cycle
2nd	Address of 2nd word (3rd and 4th bytes)
3rd	Address of 3rd word (5th and 6th bytes)
4th	Address of 4th word (7th and 8th bytes)
5th	Address of 5th word (9th and 10th bytes)
NEXT	Start address of instruction following executing instruction
EA	Effective address
VEC	Vector address

Figure A.1 shows timing waveforms for the address bus and the $\overline{RD}$, $\overline{WR}$ signals during execution of the above instruction with an 8-bit bus, using three-state access with no wait states.

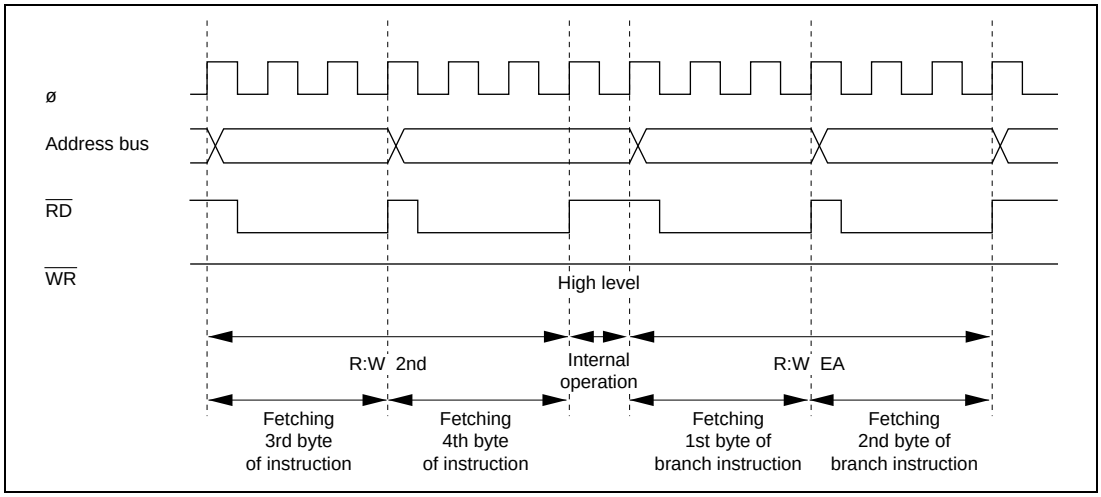


Figure A.1 Address Bus, $\overline{RD}$, $\overline{WR}$ Timing (8-Bit Bus, Three-State Access, No Wait States)

Table A.6 Instruction Execution Cycle

Instruction	1	2	3	4	5	6	7	8	9
ADD.B #xx:8,Rd	R:W NEXT								
ADD.B Rs,Rd	R:W NEXT								
ADD.W #xx:16,Rd	R:W 2nd	R:W NEXT							
ADD.W Rs,Rd	R:W NEXT								
ADD.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
ADD.L ERs,ERd	R:W NEXT								
ADDS #1/2/4,ERd	R:W NEXT								
ADDX #xx:8,Rd	R:W NEXT								
ADDX Rs,Rd	R:W NEXT								
AND.B #xx:8,Rd	R:W NEXT								
AND.B Rs,Rd	R:W NEXT								
AND.W #xx:16,Rd	R:W 2nd	R:W NEXT							
AND.W Rs,Rd	R:W NEXT								
AND.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
AND.L ERs,ERd	R:W 2nd	R:W NEXT							
ANDC #xx:8,CCR	R:W NEXT								
ANDC #xx:8,EXR	R:W 2nd	R:W NEXT							
BAND #xx:3,Rd	R:W NEXT								
BAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BAND #xx:3, @aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BAND #xx:3, @aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BRA d:8 (BT d:8)	R:W NEXT	R:W EA							
BRN d:8 (BF d:8)	R:W NEXT	R:W EA							
BHI d:8	R:W NEXT	R:W EA							
BLS d:8	R:W NEXT	R:W EA							
BCC d:8 (BHS d:8)	R:W NEXT	R:W EA							
BCS d:8 (BLO d:8)	R:W NEXT	R:W EA							
BNE d:8	R:W NEXT	R:W EA							
BEQ d:8	R:W NEXT	R:W EA							
BVC d:8	R:W NEXT	R:W EA							
BVS d:8	R:W NEXT	R:W EA							
BPL d:8	R:W NEXT	R:W EA							

Table A.6 Instruction Execution Cycle (cont)

Instruction	1	2	3	4	5	6	7	8	9
BMI d:8	R:W NEXT	R:W EA							
BGE d:8	R:W NEXT	R:W EA							
BLT d:8	R:W NEXT	R:W EA							
BGT d:8	R:W NEXT	R:W EA							
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						

Table A.6 Instruction Execution Cycle (cont)

Instruction	1	2	3	4	5	6	7	8	9
BLT d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BGT d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BLE d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BCLR #xx:3,Rd	R:W NEXT								
BCLR #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR#xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
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					

Table A.6 Instruction Execution Cycle (cont)

Instruction	1	2	3	4	5	6	7	8	9
BILD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BIOR #xx:3,Rd	R:W NEXT								
BIOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BIOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BIST #xx:3,Rd	R:W NEXT								
BIST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BIST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BIST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BIST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BIXOR #xx:3,Rd	R:W NEXT								
BIXOR #xx:3, @ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BIXOR #xx:3, @aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIXOR #xx:3, @aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIXOR #xx:3, @aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BLD #xx:3,Rd	R:W NEXT								
BLD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BLD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BLD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BLD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BNOT #xx:3,Rd	R:W NEXT								
BNOT #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					

Table A.6 Instruction Execution Cycle (cont)

Instruction	1	2	3	4	5	6	7	8	9
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 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			

Table A.6 Instruction Execution Cycle (cont)

Instruction		1	2	3	4	5	6	7	8	9
BSR d:8	Advanced	R:W NEXT	R:W EA	W:W:M Stack (H)	W:W Stack (L)					
BSR d:16	Advanced	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						
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/2128 Series and H8S/2124 Series								

Table A.6 Instruction Execution Cycle (cont)

Instruction		1	2	3	4	5	6	7	8	9
CMP.B #xx:8,Rd		R:W NEXT								
CMP.B Rs,Rd		R:W NEXT								
CMP.W #xx:16,Rd		R:W 2nd	R:W NEXT							
CMP.W Rs,Rd		R:W NEXT								
CMP.L #xx:32,ERd		R:W 2nd	R:W 3rd	R:W NEXT						
CMP.L ERs,ERd		R:W NEXT								
DAA Rd		R:W NEXT								
DAS Rd		R:W NEXT								
DEC.B Rd		R:W NEXT								
DEC.W #1/2,Rd		R:W NEXT								
DEC.L #1/2,ERd		R:W NEXT								
DIVXS.B Rs,Rd		R:W 2nd	R:W NEXT	Internal operation, 11 states						
DIVXS.W Rs,ERd		R:W 2nd	R:W NEXT	Internal operation, 19 states						
DIVXU.B Rs,Rd		R:W NEXT	Internal operation, 11 states							
DIVXU.W Rs,ERd		R:W NEXT	Internal operation, 19 states							
EEPMOV.B		R:W 2nd	R:B EAs*[1]	R:B EAd*[1]	R:B EAs*[2]	W:B EAd*[2]	R:W NEXT			
EEPMOV.W		R:W 2nd	R:B EAs*[1]	R:B EAd*[1]	R:B EAs*[2]	W:B EAd*[2]	R:W NEXT			
EXTS.W Rd		R:W NEXT			← Repeated n times *[2] →					
EXTS.L ERd		R:W NEXT								
EXTU.W Rd		R:W NEXT								
EXTU.L ERd		R:W NEXT								
INC.B Rd		R:W NEXT								
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	Advanced	R:W NEXT	R:W:M aa:8	R:W aa:8	Internal operation, 1 state	R:W EA				
JSR @ERn	Advanced	R:W NEXT	R:W EA	W:W:M Stack (H)	W:W Stack (L)					
JSR @aa:24	Advanced	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M Stack (H)	W:W Stack (L)				
JSR @@aa:8	Advanced	R:W NEXT	R:W:M aa:8	R:W aa:8	W:W:M Stack (H)	W:W Stack (L)	R:W EA			

Table A.6 Instruction Execution Cycle (cont)

Instruction	1	2	3	4	5	6	7	8	9
LDC #xx:8,CCR	R:W NEXT								
LDC #xx:8,EXR	R:W 2nd	R:W NEXT							
LDC Rs,CCR	R:W NEXT								
LDC Rs,EXR	R:W NEXT								
LDC @ERs,CCR	R:W 2nd	R:W NEXT	R:W EA						
LDC @ERs,EXR	R:W 2nd	R:W NEXT	R:W EA						
LDC@(d:16,ERs), CCR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC@(d:16,ERs), EXR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC@(d:32,ERs), CCR	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA			
LDC@(d:32,ERs), EXR	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA			
LDC @ERs+,CCR	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W EA					
LDC @ERs+,EXR	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W EA					
LDC @aa:16,CCR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC @aa:16,EXR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC @aa:32,CCR	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA				
LDC @aa:32,EXR	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA				
LDM.L @SP+, (ERn-ERn+1)	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M Stack (H) *[3]	R:W Stack (L) *[3]				
LDM.L @SP+, (ERn-ERn+2)	R:W 2nd	R:W:M 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:M 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/2128 Series and H8S/2124 Series								
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						

Table A.6 Instruction Execution Cycle (cont)

Instruction	1	2	3	4	5	6	7	8	9
MOV.B @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:B EA				
MOV.B @ERs+,Rd	R:W NEXT	Internal operation, 1 state	R:B EA						
MOV.B @aa:8,Rd	R:W NEXT	R:B EA							
MOV.B @aa:16,Rd	R:W 2nd	R:W NEXT	R:B EA						
MOV.B @aa:32,Rd	R:W 2nd	R:W 3rd	R:W NEXT	R:B EA					
MOV.B Rs,@ERd	R:W NEXT	W:B EA							
MOV.B Rs, @(d:16,ERd)	R:W 2nd	R:W NEXT	W:B EA						
MOV.B Rs, @(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:B EA				
MOV.B Rs,@-ERd	R:W NEXT	Internal operation, 1 state	W:B EA						
MOV.B Rs,@aa:8	R:W NEXT	W:B EA							
MOV.B Rs,@aa:16	R:W 2nd	R:W NEXT	W:B EA						
MOV.B Rs,@aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:B EA					
MOV.W #xx:16,Rd	R:W 2nd	R:W NEXT							
MOV.W Rs,Rd	R:W NEXT								
MOV.W @ERs,Rd	R:W NEXT	R:W EA							
MOV.W @(d:16,ERs),Rd	R:W 2nd	R:W NEXT	R:W EA						
MOV.W @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA				
MOV.W @ERs+,Rd	R:W NEXT	Internal operation, 1 state	R:W EA						
MOV.W @aa:16,Rd	R:W 2nd	R:W NEXT	R:W EA						
MOV.W @aa:32,Rd	R:W 2nd	R:W 3rd	R:W NEXT	R:B EA					
MOV.W Rs,@ERd	R:W NEXT	W:W EA							
MOV.W Rs, @(d:16,ERd)	R:W 2nd	R:W NEXT	W:W EA						
MOV.W Rs, @(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 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					

Table A.6 Instruction Execution Cycle (cont)

Instruction	1	2	3	4	5	6	7	8	9
MOV.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
MOV.L ERs,ERd	R:W NEXT								
MOV.L @ERs,ERd	R:W 2nd	R:W:M NEXT	R:W:M EA	R:W EA+2					
MOV.L @(d:16,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2				
MOV.L @(d:32,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	R:W:M EA	R:W EA+2		
MOV.L @ERs+, ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2				
MOV.L @aa:16, ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2				
MOV.L @aa:32, ERd	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	R:W:M EA	R:W EA+2			
MOV.L ERs,@ERd	R:W 2nd	R:W:M NEXT	W:W:M EA	W:W EA+2					
MOV.L ERs, @(d:16,ERd)	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2				
MOV.L ERs, @(d:32,ERd)	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	W:W:M EA	W:W EA+2		
MOV.L ERs,@-ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M EA	W:W EA+2				
MOV.L ERs, @aa:16	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2				
MOV.L ERs, @aa:32	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	W:W:M EA	W:W EA+2			
MOVFPE @aa:16,Rd	Cannot be used in the H8S/2128 Series and H8S/2124 Series								
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								

Table A.6 Instruction Execution Cycle (cont)

Instruction	1	2	3	4	5	6	7	8	9
NOT.L ERd	R:W NEXT								
OR.B #xx:8,Rd	R:W NEXT								
OR.B Rs,Rd	R:W NEXT								
OR.W #xx:16,Rd	R:W 2nd	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								

Table A.6 Instruction Execution Cycle (cont)

Instruction		1	2	3	4	5	6	7	8	9
ROTXL.L #2,ERd		R:W NEXT								
ROTXR.B Rd		R:W NEXT								
ROTXR.B #2,Rd		R:W NEXT								
ROTXR.W Rd		R:W NEXT								
ROTXR.W #2,Rd		R:W NEXT								
ROTXR.L ERd		R:W NEXT								
ROTXR.L #2,ERd		R:W NEXT								
RTE		R:W NEXT	R:W Stack (EXR)	R:W Stack (H)	R:W Stack (L)	Internal operation, 1 state	R:W *4			
RTS	Advanced	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								

Table A.6 Instruction Execution Cycle (cont)

Instruction	1	2	3	4	5	6	7	8	9
SLEEP	R:W NEXT	Internal operation :M							
STC CCR,Rd	R:W NEXT								
STC EXR,Rd	R:W NEXT								
STC CCR,@ERd	R:W 2nd	R:W NEXT	W:W EA						
STC EXR,@ERd	R:W 2nd	R:W NEXT	W:W EA						
STC CCR, @(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC EXR, @(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC CCR, @(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA			
STC EXR, @(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA			
STC CCR,@-ERd	R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA					
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/2128 Series and H8S/2124 Series								
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								

Table A.6 Instruction Execution Cycle (cont)

Instruction		1	2	3	4	5	6	7	8	9
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	Advanced	R:W NEXT	Internal operation, 1 state	W:W Stack (L)	W:W Stack (H)	W:W Stack (EXR)	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W *7
XOR.B #xx8,Rd		R:W NEXT								
XOR.B Rs,Rd		R:W NEXT								
XOR.W #xx:16,Rd		R:W 2nd	R:W NEXT							
XOR.W Rs,Rd		R:W NEXT								
XOR.L #xx:32,ERd		R:W 2nd	R:W 3rd	R:W NEXT						
XOR.L ERs,ERd		R:W 2nd	R:W NEXT							
XORC #xx:8,CCR		R:W NEXT								
XORC #xx:8,EXR		R:W 2nd	R:W NEXT							
Reset exception handling	Advanced	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W *5					
Interrupt exception handling	Advanced	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	Internal operation, 1 state	R:W *7

Notes:
1. EAs is the contents of ER5. EAd is the contents of ER6.
2. EAs is the contents of ER5. EAd is the contents of ER6. Both registers are incremented by 1 after execution of the instruction. n is the initial value of R4L or R4. If n = 0, these bus cycles are not executed.
3. Repeated two times to save or restore two registers, three times for three registers, or four times for four registers.
4. Start address after return.
5. 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 software standby mode the read operation is replaced by an internal operation.
7. Start address of the interrupt-handling routine.

Appendix B Internal I/O Registers

B.1 Addresses

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Bus Width
H'EC00 to H'EFFF	MRA	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz	DTC	16/32*
	SAR										
	MRB	CHNE	DISEL	—	—	—	—	—	—		
	DAR										
	CRA										
	CRB										
H'FEE4	KBCOMP	IrE	IrCKS2	IrCKS1	IrCKS0	KBADE	KBCH2	KBCH1	KBCH0	Expansion A/D	8
H'FEE6	DDCSWR	SWE	SW	IE	IF	CLR3	CLR2	CLR1	CLR0	IIC0	8
H'FEE8	ICRA	ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0	Interrupt controller	8
H'FEE9	ICRB	ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0		
H'FEEA	ICRC	ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0		
H'FEEB	ISR	—	—	—	—	—	IRQ2F	IRQ1F	IRQ0F		
H'FEEC	ISCRH	—	—	—	—	—	—	—	—		
H'FEED	ISCRL	—	—	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA		
H'FEEE	DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC	8
H'FEEF	DTCERB	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0		
H'FEF0	DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0		
H'FEF1	DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0		
H'FEF2	DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0		
H'FEF3	DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0		
H'FEF4	ABRKCR	CMF	—	—	—	—	—	—	BIE	Interrupt controller	8
H'FEF5	BARA	A23	A22	A21	A20	A19	A18	A17	A16		
H'FEF6	BARB	A15	A14	A13	A12	A11	A10	A9	A8		
H'FEF7	BARC	A7	A6	A5	A4	A3	A2	A1	—		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Bus Width
H'FF80	FLMCR1	FWE	SWE	—	—	EV	PV	E	P	FLASH	8
H'FF81	FLMCR2	FLER	—	—	—	—	—	ESU	PSU		
H'FF82	PCSR	—	—	—	—	—	PWCKB	PWCKA	—	PWM	8
	EBR1	—	—	—	—	—	—	EB9	EB8	FLASH	8
H'FF83	EBR2	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0		
H'FF84	SBYCR	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0	SYSTEM	8
H'FF85	LPWRCR	DTON	LSON	NESEL	EXCLE	—	—	—	—		
H'FF86	MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8		
H'FF87	MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0		
H'FF88	SMR1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI1	8
	ICCR1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC1	
H'FF89	BRR1									SCI1	8
	ICSR1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	IIC1	
H'FF8A	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI1	8
H'FF8B	TDR1										
H'FF8C	SSR1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
H'FF8D	RDR1										
H'FF8E	SCMR1	—	—	—	—	SDIR	SINV	—	SMIF		
	ICDR1	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	IIC1	8
	SARX1	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX		
H'FF8F	ICMR1	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0		
	SAR1	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS		
H'FF90	TIER	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	FRT	16
H'FF91	TCSR	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA		
H'FF92	FRCH										
H'FF93	FRCL										
H'FF94	OCRAH										
	OCRBH										
H'FF95	OCRAL										
	OCRBL										
H'FF96	TCR	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0		
H'FF97	TOCR	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB		
H'FF98	ICRAH										
	OCRARH										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Bus Width
H'FF99	ICRAL									FRT	16
	OCRARL										
H'FF9A	ICRBH										
	OCRAFH										
H'FF9B	ICRBL										
	OCRAFL										
H'FF9C	ICRCH										
	OCRDMH	0	0	0	0	0	0	0	0		
H'FF9D	ICRCL										
	OCRDML										
H'FF9E	ICRDH										
H'FF9F	ICRDL										
H'FFA0	DADRAH	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	PWMX	8
	DACR	TEST	PWME	—	—	OEB	OEA	OS	CKS		
H'FFA1	DADRAL	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—		
H'FFA6	DADRBH	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6		
	DACNTH										
H'FFA7	DADRBL	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS		
	DACNTL							—	REGS		
H'FFA8	TCSR0	OVF	WT/$\overline{IT}$	TME	RSTS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT0	16
	TCNT0 (write)										
H'FFA9	TCNT0 (read)										
H'FFAC	P1PCR	P17PCR	P16PCR	P15PCR	P14PCR	P13PCR	P12PCR	P11PCR	P10PCR	Ports	8
H'FFAD	P2PCR	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PCR		
H'FFAE	P3PCR	P37PCR	P36PCR	P35PCR	P34PCR	P33PCR	P32PCR	P31PCR	P30PCR		
H'FFB0	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR		
H'FFB1	P2DDR	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR		
H'FFB2	P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR		
H'FFB3	P2DR	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR		
H'FFB4	P3DDR	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR		
H'FFB5	P4DDR	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR		
H'FFB6	P3DR	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR		
H'FFB7	P4DR	P47DR	P46DR	P45DR	P44DR	P43DR	P42DR	P41DR	P40DR		
H'FFB8	P5DDR	—	—	—	—	—	P52DDR	P51DDR	P50DDR		
H'FFB9	P6DDR	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR		
H'FFBA	P5DR	—	—	—	—	—	P52DR	P51DR	P50DR		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Bus Width
H'FFBB	P6DR	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR	Ports	8
H'FFBE	P7PIN	P77PIN	P76PIN	P75PIN	P74PIN	P73PIN	P72PIN	P71PIN	P70PIN		
H'FFC2	IER	—	—	—	—	—	IRQ2E	IRQ1E	IRQ0E	Interrupt controller	8
H'FFC3	STCR	IICS	IICX1	IICX0	IICE	FLSHE	—	ICKS1	ICKS0	System	8
H'FFC4	SYSCR	CS2E	IOSE	INTM1	INTM0	XRST	NMIEG	HIE	RAME		
H'FFC5	MDCR	EXPE	—	—	—	—	—	MDS1	MDS0		
H'FFC6	BCR	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0	Bus controller	8
H'FFC7	WSCR	RAMS	RAM0	ABW	AST	WMS1	WMS0	WC1	WC0		
H'FFC8	TCR0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR0, TMR1	16
H'FFC9	TCR1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0		
H'FFCA	TCSR0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0		
H'FFCB	TCSR1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0		
H'FFCC	TCORA0										
H'FFCD	TCORA1										
H'FFCE	TCORB0										
H'FFCF	TCORB1										
H'FFD0	TCNT0										
H'FFD1	TCNT1										
H'FFD2	PWOERB	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8	PWM	8
H'FFD3	PWOERA	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0		
H'FFD4	PWDPRB	OS15	OS14	OS13	OS12	OS11	OS10	OS9	OS8		
H'FFD5	PWDPRA	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0		
H'FFD6	PWSL	PWCKE	PWCKS	—	—	RS3	RS2	RS1	RS0		
H'FFD7	PWDR0 to PWDR15										
H'FFD8	SMR0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI0	8
	ICCR0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC0	
H'FFD9	BRR0									SCI0	
	ICSR0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	IIC0	
H'FFDA	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI0	
H'FFDB	TDR0										
H'FFDC	SSR0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
H'FFDD	RDR0										
H'FFDE	SCMR0	—	—	—	—	SDIR	SINV	—	SMIF		
	ICDR0	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	IIC0	
	SARX0	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Bus Width
H'FFDF	ICMR0	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0	IIC0	8
	SAR0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS		
H'FFE0	ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D	8
H'FFE1	ADDRAL	AD1	AD0	—	—	—	—	—	—		
H'FFE2	ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FFE3	ADDRBL	AD1	AD0	—	—	—	—	—	—		
H'FFE4	ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FFE5	ADDRCL	AD1	AD0	—	—	—	—	—	—		
H'FFE6	ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FFE7	ADDRDL	AD1	AD0	—	—	—	—	—	—		
H'FFE8	ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0		
H'FFE9	ADCR	TRGS1	TRGS0	—	—	—	—	—	—		
H'FFEA	TCSR1	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT1	16
	TCNT1 (write)										
H'FFEB	TCNT1 (read)										
H'FFF0	TCRX	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMRX	8
	TCRY	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMRY	
H'FFF1	TCSRX	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0	TMRX	
	TCSRY	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0	TMRY	
H'FFF2	TICRR									TMRX	
	TCORAY									TMRY	
H'FFF3	TICRF									TMRX	
	TCORBY									TMRY	
H'FFF4	TCNTX									TMRX	
	TCNTY									TMRY	
H'FFF5	TCORC									TMRX	
	TISR	—	—	—	—	—	—	—	IS	TMRY	
H'FFF6	TCORAX									TMRX	
H'FFF7	TCORBX										
H'FFFC	TCONRI	SIMOD1	SIMOD0	SCONE	ICST	HFINV	VFINV	HIINV	VIINV	Timer connection	8
H'FFFD	TCONRO	HOE	VOE	CLOE	CBOE	HOINV	VOINV	CLOINV	CBOINV		
H'FFFE	TCONRS	TMRX/Y	ISGENE	HOMOD1	HOMOD0	VOMOD1	VOMOD0	CLMOD1	CLMOD0		
H'FFFF	SEDGR	VEDG	HEDG	CEDG	HFEDG	VFEDG	PREQF	IHI	IVI		

B.2 Register Selection Conditions

Lower Address	Register Name	H8S/2128 Series Register Selection Conditions	H8S/2124 Series Register Selection Conditions	Module Name
H'EC00 to H'EFFF	MRA	RAME = 1 in SYSCR	—	DTC
	SAR			
	MRB			
	DAR			
	CRA			
	CRB			
H'FEE4	KBCOMP	No conditions	No conditions	Expansion A/D
H'FEE6	DDCSWR	MSTP4 = 0	—	IIC0
H'FEE8	ICRA	No conditions	No conditions	Interrupt controller
H'FEE9	ICRB			
H'FEEA	ICRC			
H'FEEB	ISR			
H'FEEC	ISCRH			
H'FEED	ISCRL			
H'FEEE	DTCERA	No conditions	—	DTC
H'FEEF	DTCERB			
H'FEF0	DTCERC			
H'FEF1	DTCERD			
H'FEF2	DTCERE			
H'FEF3	DTVECR			
H'FEF4	ABRKCR	No conditions	No conditions	Interrupt controller
H'FEF5	BARA			
H'FEF6	BARB			
H'FEF7	BARC			
H'FF80	FLMCR1	FLSHE = 1 in STCR	FLSHE = 1 in STCR	Flash memory
H'FF81	FLMCR2			
H'FF82	PCSR	FLSHE = 0 in STCR	FLSHE = 0 in STCR	PWM
	EBR1	FLSHE = 1 in STCR	FLSHE = 1 in STCR	Flash memory
H'FF83	EBR2	FLSHE = 1 in STCR	FLSHE = 1 in STCR	Flash memory
H'FF84	SBYCR	FLSHE = 0 in STCR	FLSHE = 0 in STCR	System
H'FF85	LPWRCR			
H'FF86	MSTPCRH			
H'FF87	MSTPCRL			

Lower Address	Register Name	H8S/2128 Series Register Selection Conditions		H8S/2124 Series Register Selection Conditions	Module Name
H'FF88	SMR1	MSTP6=0, IICE=0 in STCR		MSTP6=0, IICE=0 in STCR	SCI1
	ICCR1	MSTP3=0, IICE=1 in STCR		—	IIC1
H'FF89	BRR1	MSTP6=0, IICE=0 in STCR		MSTP6=0, IICE=0 in STCR	SCI1
	ICSR1	MSTP3=0, IICE=1 in STCR		—	IIC1
H'FF8A	SCR1	MSTP6=0		MSTP6=0	SCI1
H'FF8B	TDR1				
H'FF8C	SSR1				
H'FF8D	RDR1				
H'FF8E	SCMR1	MSTP6=0, IICE=0 in STCR		MSTP6=0, IICE=0 in STCR	
	ICDR1	MSTP3=0, IICE=1 in STCR	ICE=1 in ICCR1	—	IIC1
	SARX1		ICE = 0 in ICCR1		
H'FF8F	ICMR1		ICE = 1 in ICCR1		
	SAR1		ICE = 0 in ICCR1		
H'FF90	TIER	MSTP13 = 0		MSTP13 = 0	FRT
H'FF91	TCSR				
H'FF92	FRCH				
H'FF93	FRCL				
H'FF94	OCRAH				
	OCRBH				
H'FF95	OCRAL				
	OCRBL				
H'FF96	TCR				
H'FF97	TOCR				
H'FF98	ICRAH				
	OCRARH				
H'FF99	ICRAL				
	OCRARL				
H'FF9A	ICRBH				
	OCRAFH				
H'FF9B	ICRBL				
	OCRAFL				
H'FF9C	ICRCH				
	OCRDMH				
H'FF9D	ICRCL				
	OCRDML				

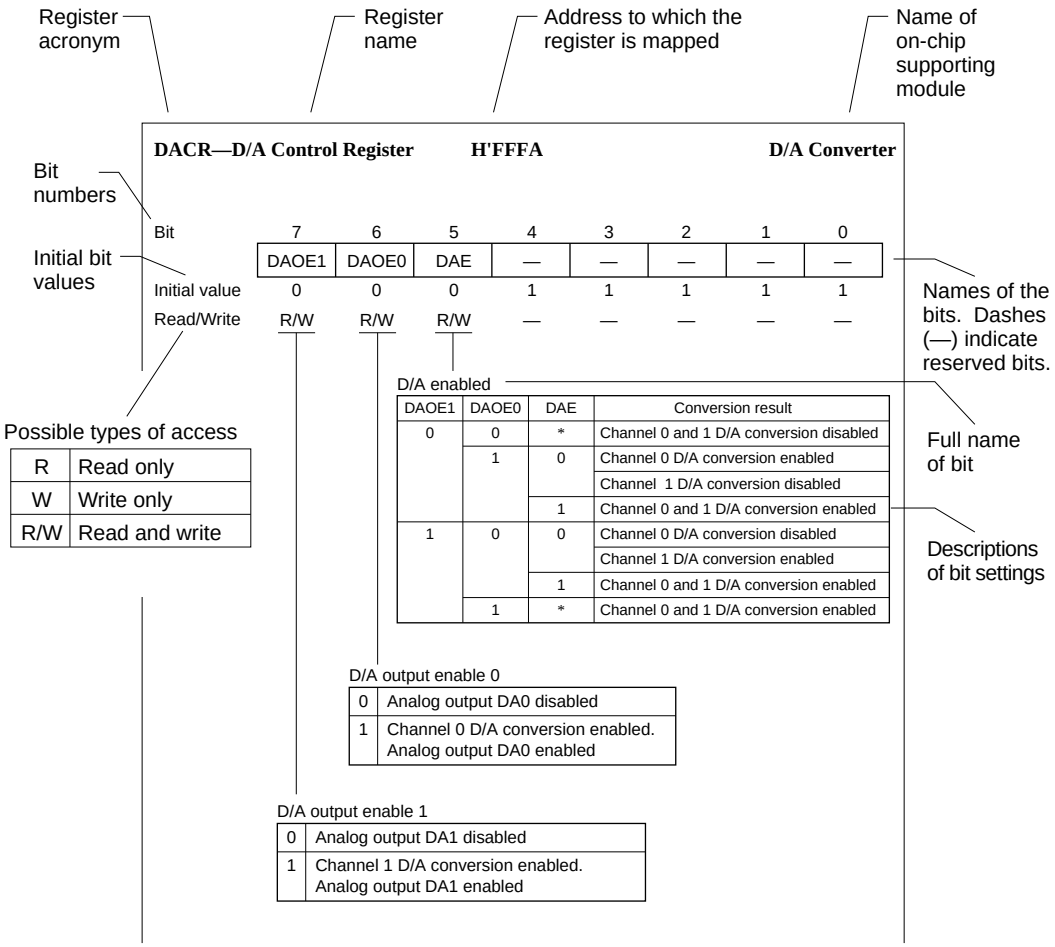
Lower Address	Register Name	H8S/2128 Series Register Selection Conditions		H8S/2124 Series Register Selection Conditions	Module Name
H'FF9E	ICRDH	MSTP13 = 0		MSTP13 = 0	FRT
H'FF9F	ICRDL				
H'FFA0	DADRAH	MSTP11 = 0, IICE = 1 in STCR	REGS = 0 in DACNT/ DADRB	—	PWMX
	DACR		REGS = 1 in DACNT/ DADRB		
H'FFA1	DADRAL	MSTP11 = 0, IICE = 1 in STCR	REGS = 0 in DACNT/ DADRB	—	PWMX
H'FFA6	DADRBH	MSTP11 = 0, IICE = 1 in STCR	REGS = 0 in DACNT/ DADRB		
	DACNTH		REGS = 1 in DACNT/ DADRB		
H'FFA7	DADRBL		REGS = 0 in DACNT/ DADRB		
	DACNTL		REGS = 1 in DACNT/ DADRB		
H'FFA8	TCSR0	No conditions		No conditions	WDT0
	TCNT0 (write)				
H'FFA9	TCNT0 (read)				
H'FFAC	P1PCR	No conditions		No conditions	Ports
H'FFAD	P2PCR				
H'FFAE	P3PCR				
H'FFB0	P1DDR				
H'FFB1	P2DDR				
H'FFB2	P1DR				
H'FFB3	P2DR				
H'FFB4	P3DDR				
H'FFB5	P4DDR				
H'FFB6	P3DR				
H'FFB7	P4DR				
H'FFB8	P5DDR				
H'FFB9	P6DDR				
H'FFBA	P5DR				
H'FFBB	P6DR				

Lower Address	Register Name	H8S/2128 Series Register Selection Conditions		H8S/2124 Series Register Selection Conditions	Module Name
H'FFBE	P7PIN	No conditions		No conditions	Ports
H'FFC2	IER	No conditions		No conditions	Interrupt controller
H'FFC3	STCR	No conditions		No conditions	System
H'FFC4	SYSCR				
H'FFC5	MDCR				
H'FFC6	BCR				Bus controller
H'FFC7	WSCR				
H'FFC8	TCR0	MSTP12 = 0		MSTP12 = 0	TMR0, TMR1
H'FFC9	TCR1				
H'FFCA	TCSR0				
H'FFCB	TCSR1				
H'FFCC	TCORA0				
H'FFCD	TCORA1				
H'FFCE	TCORB0				
H'FFCF	TCORB1				
H'FFD0	TCNT0				
H'FFD1	TCNT1				
H'FFD2	PWOERB	No conditions		—	PWM
H'FFD3	PWOERA				
H'FFD4	PWDPRB				
H'FFD5	PWDPRA				
H'FFD6	PWSL	MSTP11 = 0			
H'FFD7	PWDR0 to 15				
H'FFD8	SMR0	MSTP7 = 0, IICE = 0 in STCR		MSTP7 = 0, IICE = 0 in STCR	SCI0
	ICCR0	MSTP4 = 0, IICE = 1 in STCR		—	IIC0
H'FFD9	BRR0	MSTP7 = 0, IICE = 0 in STCR		MSTP7 = 0, IICE = 0 in STCR	SCI0
	ICSR0	MSTP4 = 0, IICE = 1 in STCR		—	IIC0
H'FFDA	SCR0	MSTP7 = 0		MSTP7 = 0	SCI0
H'FFDB	TDR0				
H'FFDC	SSR0				
H'FFDD	RDR0				
H'FFDE	SCMR0	MSTP7 = 0, IICE = 0 in STCR		MSTP7 = 0, IICE = 0 in STCR	
	ICDR0	MSTP4 = 0, IICE = 1 in STCR	ICE = 1 in ICCR0	—	IIC0
	SARX0		ICE = 0 in ICCR0		

Lower Address	Register Name	H8S/2128 Series Register Selection Conditions		H8S/2124 Series Register Selection Conditions	Module Name
H'FFDF	ICMR0	MSTP4 = 0, IICE = 1 in STCR	ICE = 1 in ICCR0	—	IIC0
	SAR0		ICE = 0 in ICCR0		
H'FFE0	ADDRAH	MSTP9 = 0		MSTP9 = 0	A/D
H'FFE1	ADDRAL				
H'FFE2	ADDRBH				
H'FFE3	ADDRBL				
H'FFE4	ADDRCH				
H'FFE5	ADDRCL				
H'FFE6	ADDRDH				
H'FFE7	ADDRDL				
H'FFE8	ADCSR				
H'FFE9	ADCR				
H'FFEA	TCSR1	No conditions		No conditions	WDT1
	TCNT1 (write)				
H'FFEB	TCNT1 (read)				
H'FFF0	TCRX	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	—	TMRX
	TCRY		TMRX/Y = 1 in TCONRS	MSTP8 = 0, HIE = 0 in SYSCR	TMRY
H'FFF1	TCSRX	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	—	TMRX
	TCSRY		TMRX/Y = 1 in TCONRS	MSTP8 = 0, HIE = 0 in SYSCR	TMRY
H'FFF2	TICRR	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	—	TMRX
	TCORAY		TMRX/Y = 1 in TCONRS	MSTP8 = 0, HIE = 0 in SYSCR	TMRY
H'FFF3	TICRF	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	—	TMRX
	TCORBY		TMRX/Y = 1 in TCONRS	MSTP8 = 0, HIE = 0 in SYSCR	TMRY
H'FFF4	TCNTX	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	—	TMRX
	TCNTY		TMRX/Y = 1 in TCONRS	MSTP8 = 0, HIE = 0 in SYSCR	TMRY
H'FFF5	TCORC	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	—	TMRX
	TISR		TMRX/Y = 1 in TCONRS	MSTP8 = 0, HIE = 0 in SYSCR	TMRY

Lower Address	Register Name	H8S/2128 Series Register Selection Conditions		H8S/2124 Series Register Selection Conditions	Module Name
H'FFF6	TCORAX	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	—	TMRX
H'FFF7	TCORBX				
H'FFFC	TCONRI	MSTP8 = 0, HIE = 0 in SYSCR		—	Timer connection
H'FFFD	TCONRO				
H'FFFE	TCONRS				
H'FFFF	SEDGR				

B.3 Functions



DAOE1	DAOE0	DAE	Conversion result
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 conversion 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

Bit	7	6	5	4	3	2	1	0
	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	—	—	—

DTC data transfer size

0	Byte-size transfer
1	Word-size transfer

DTC transfer mode select

0	Destination side is repeat area or block area
1	Source side is repeat area or block area

DTC mode

0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

Destination address mode

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)

MRB—DTC Mode Register B **H'EC00–H'EFFF** **DTC**

Bit	7	6	5	4	3	2	1	0
	CHNE	DISEL	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	—	—	—

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 enabled

DTC chain transfer enable

0	End of DTC data transfer
1	DTC chain transfer

SAR—DTC Source Address Register **H'EC00–H'EFFF** **DTC**

Bit	23	22	21	20	19	- - -	4	3	2	1	0
						- - -					
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	- - -	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	- - -	—	—	—	—	—

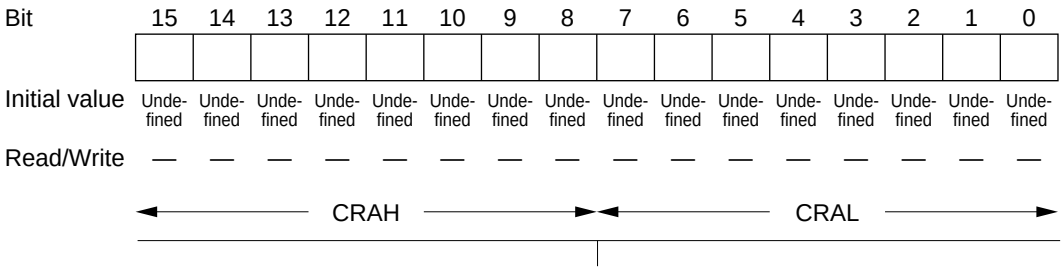
Specifies DTC transfer data source address

DAR—DTC Destination Address Register **H'EC00–H'EFFF** **DTC**

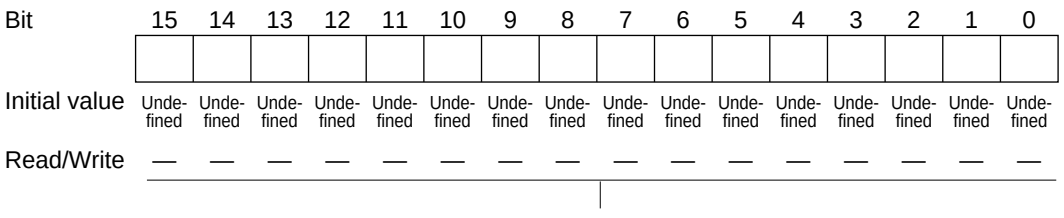
Bit	23	22	21	20	19	- - -	4	3	2	1	0
						- - -					
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	- - -	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	- - -	—	—	—	—	—

Specifies DTC transfer data destination address

CRA—DTC Transfer Count Register A **H'EC00–H'EFFF** **DTC**



CRB—DTC Transfer Count Register B **H'EC00–H'EFFF** **DTC**



KBCOMP—Keyboard Comparator Control Register H'FEE4 COMP

Bit	7	6	5	4	3	2	1	0
	IrE	IrCKS2	IrCKS1	IrCKS0	KBADE	KBCH2	KBCH1	KBCH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reserved bits

Keyboard comparator control

Bit 3 KBADE	Bit 3 KBCH2	Bit 3 KBCH1	Bit 3 KBCH0	A/D converter channel 6 input	A/D converter channel 7 input
0	—	—	—	AN6	AN7
1	0	0	0	CIN0	Undefined
			1	CIN1	
		1	0	CIN2	
			1	CIN3	
	1	0	0	CIN4	
			1	CIN5	
		1	0	CIN6	
			1	CIN7	

Bit	7	6	5	4	3	2	1	0
	SWE	SW	IE	IF	CLR3	CLR2	CLR1	CLR0
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/(W)*1	W*2	W*2	W*2	W*2

IIC clear bits

Bit 3	Bit 2	Bit 1	Bit 0	Description
CLR3	CLR2	CLR1	CLR0	
0	0	—	—	Setting prohibited
	1	0	0	Setting prohibited
			1	IIC0 internal latch cleared
		1	0	IIC1 internal latch cleared
			1	IIC0 and IIC1 internal latches cleared
1	—	—	—	Invalid setting

DDC mode switch interrupt flag

0	No interrupt is requested when automatic format switching is executed [Clearing condition] When 0 is written in IF after reading IF = 1
1	An interrupt is requested when automatic format switching is executed [Setting condition] When a falling edge is detected on the SCL pin when SWE = 1

DDC mode switch interrupt enable bit

0	Interrupt when automatic format switching is executed is disabled
1	Interrupt when automatic format switching is executed is enabled

DDC mode switch

0	IIC channel 0 is used with the I^2C bus format [Clearing conditions] • When 0 is written by software • When a falling edge is detected on the SCL pin when SWE = 1
1	IIC channel 0 is used in formatless mode [Setting condition] When 1 is written in SW after reading SW = 0

DDC Mode switch enable

0	Automatic switching of IIC channel 0 from formatless mode to I^2C bus format is disabled
1	Automatic switching of IIC channel 0 from formatless mode to I^2C bus format is enabled

Notes: 1. Only 0 can be written, to clear the flag.
2. Always read as 1.

ICRA—Interrupt Control Register A **H'FEE8** **Interrupt Controller**
ICRB—Interrupt Control Register B **H'FEE9** **Interrupt Controller**
ICRC—Interrupt Control Register C **H'FEEA** **Interrupt Controller**

Bit	7	6	5	4	3	2	1	0
	ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Interrupt control level

0	Corresponding interrupt source is control level 0 (non-priority)
1	Corresponding interrupt source is control level 1 (priority)

Correspondence between Interrupt Sources and ICR Settings

Register	Bits							
	7	6	5	4	3	2	1	0
ICRA	IRQ0	IRQ1	IRQ2	—	—	DTC	Watchdog timer 0	Watchdog timer 1
ICRB	A/D converter	Free-running timer	—	—	8-bit timer channel 0	8-bit timer channel 1	8-bit timer channels X, Y	—
ICRC	SCI channel 0	SCI channel 1	—	IIC channel 0 (option)	IIC channel 1 (option)	—	—	—

ISR—IRQ Status Register **H'FEEB** **Interrupt Controller**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	IRQ2F	IRQ1F	IRQ0F
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

IRQ2 to IRQ0 flags

0	[Clearing conditions] • When 0 is written in IRQnF after reading IRQnF = 1 • When interrupt exception handling is executed while low-level detection is set (IRQnSCB = IRQnSCA = 0) and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed while falling, rising, or both-edge detection is set (IRQnSCB = 1 or IRQnSCA = 1)
1	[Setting conditions] • When $\overline{\text{IRQn}}$ input goes low while low-level detection is set (IRQnSCB = IRQnSCA = 0) • When a falling edge occurs in $\overline{\text{IRQn}}$ input while falling edge detection is set (IRQnSCB = 0, IRQnSCA = 1) • When a rising edge occurs in $\overline{\text{IRQn}}$ input while rising edge detection is set (IRQnSCB = 1, IRQnSCA = 0) • When a falling or rising edge occurs in $\overline{\text{IRQn}}$ input while both-edge detection is set (IRQnSCB = IRQnSCA = 1)

(n = 2 to 0)

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

ISCRH—IRQ Sense Control Register H **H'FEEC** **Interrupt Controller**
ISCRL—IRQ Sense Control Register L **H'FEED** **Interrupt Controller**

ISCRH

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

Reserved

ISCRL

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ2 to IRQ0 sense control A and B

ISCRL bits 5–0		Description
IRQ2SCB–IRQ0SCB	IRQ2SCA–IRQ0SCA	
0	0	Interrupt request generated by low level of $\overline{\text{IRQ2}}$–$\overline{\text{IRQ0}}$ input
	1	Interrupt request generated by falling edge of $\overline{\text{IRQ2}}$–$\overline{\text{IRQ0}}$ input
1	0	Interrupt request generated by rising edge of $\overline{\text{IRQ2}}$–$\overline{\text{IRQ0}}$ input
	1	Interrupt request generated by rising and falling edges of $\overline{\text{IRQ2}}$–$\overline{\text{IRQ0}}$ input

DTCER—DTC Enable Register **H'FFEE to H'FFF2** **DTC**

Bit	7	6	5	4	3	2	1	0
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DTC activation enable

0	DTC activation by interrupt is disabled [Clearing conditions] • When data transfer ends while the DISEL bit is 1 • When the specified number of transfers are completed
1	DTC activation by interrupt is enabled [Maintenance condition] When the DISEL bit is 0 and the specified number of transfers have not been completed

DTVECR—DTC Vector Register **H'FEF3** **DTC**

Bit	7	6	5	4	3	2	1	0
	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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 have not been completed
1	DTC software activation is enabled [Maintenance conditions] • When data transfer ends while the DISEL bit is 1 • When the specified number of transfers are completed • During data transfer activated by software

Note: * A value of 1 can always be written to the SWDTE bit, but 0 can only be written after 1 is read.

ABRKCR—Address Break Control Register H'FEF4 Interrupt Controller

Bit	7	6	5	4	3	2	1	0
	CMF	—	—	—	—	—	—	BIE
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	—	—	—	—	—	—	R/W

Break interrupt enable

0	Address break disabled
1	Address break enabled

Condition match flag

0	[Clearing condition] When address break interrupt exception handling is executed
1	[Setting condition] When address set by BARA–BARC is prefetched while BIE = 1

BARA—Break Address Register A **H'FEF5** **Interrupt Controller**
BARB—Break Address Register B **H'FEF6** **Interrupt Controller**
BARC—Break Address Register C **H'FEF7** **Interrupt Controller**

Bit	7	6	5	4	3	2	1	0
BARA	A23	A22	A21	A20	A19	A18	A17	A16
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Specifies address (bits 23–16) at which address break is to be generated

Bit	7	6	5	4	3	2	1	0
BARB	A15	A14	A13	A12	A11	A10	A9	A8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Specifies address (bits 15–8) at which address break is to be generated

Bit	7	6	5	4	3	2	1	0
BARC	A7	A6	A5	A4	A3	A2	A1	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

Specifies address (bits 7–1) at which address break is to be generated

Bit	7	6	5	4	3	2	1	0
	FWE	SWE	—	—	EV	PV	E	P
Initial value	1	0	0	0	0	0	0	0
Read/Write	R	R/W	—	—	R/W	R/W	R/W	R/W

Program

0	Program mode cleared
1	Transition to program mode [Setting condition] When SWE = 1, and PSU = 1

Erase

0	Erase mode cleared
1	Transition to erase mode [Setting condition] When SWE = 1, and ESU = 1

Program-verify

0	Program-verify mode cleared
1	Transition to program-verify mode [Setting condition] When SWE = 1

Erase-verify

0	Erase-verify mode cleared
1	Transition to erase-verify mode [Setting condition] When SWE = 1

Software write enable

0	Writes disabled
1	Writes enabled

Reserved

Bit	7	6	5	4	3	2	1	0
	FLER	—	—	—	—	—	ESU	PSU
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	—	—	—	—	—	R/W	R/W

Program setup bit

0	Program setup cleared
1	Program setup [Setting condition] When SWE = 1

Erase setup bit

0	Erase setup cleared
1	Erase setup [Setting condition] When SWE = 1

Flash memory error

0	Flash memory is operating normally Flash memory program/erase protection (error protection) is disabled [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 19.8.3, Error Protection

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	PWCKB	PWCKA	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	—

PWM clock select

PWSL		PCSR		Description
Bit 7	Bit 6	Bit 2	Bit 1	
PWCKE	PWCKS	PWCKB	PWCKA	
0	—	—	—	Clock input disabled
1	0	—	—	ø (system clock) selected
	1	0	0	ø/2 selected
			1	ø/4 selected
		1	0	ø/8 selected
			1	ø/16 selected

EBR1—Erase Block Register 1 **H'FF82** **Flash Memory**
EBR2—Erase Block Register 2 **H'FF83** **Flash Memory**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	EB9	EB8
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	—	R/W*	R/W*

Bit	7	6	5	4	3	2	1	0
	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * In normal mode, a read will return 0, and writes are invalid.

Erase Blocks

Block (Size) 128-kbyte versions	Addresses
EB0 (1 kbyte)	H'(00)0000–H'(00)03FF
EB1 (1 kbyte)	H'(00)4000–H'(00)07FF
EB2 (1 kbyte)	H'(00)8000–H'(00)0BFF
EB3 (1 kbyte)	H'(00)C000–H'(00)0FFF
EB4 (28 kbytes)	H'(00)1000–H'(00)7FFF
EB5 (16 kbytes)	H'(00)8000–H'(00)BFFF
EB6 (8 kbytes)	H'(00)C000–H'(00)DFFF
EB7 (8 kbytes)	H'00E000–H'00FFFF
EB8 (32 kbytes)	H'010000–H'017FFF
EB9 (32 kbytes)	H'018000–H'01FFFF

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

System clock select 2 to 0

0	0	0	Bus master is in high-speed mode
		1	Medium-speed clock = ø/2
	1	0	Medium-speed clock = ø/4
		1	Medium-speed clock = ø/8
1	0	0	Medium-speed clock = ø/16
		1	Medium-speed clock = ø/32
	1	—	—

Standby timer select 2 to 0

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*

Note: * This setting must not be used in the flash memory version.

Software standby

0	Transition to sleep mode on execution of SLEEP instruction in high-speed mode or medium-speed mode Transition to subsleep mode on execution of SLEEP instruction in subactive mode
1	Transition to software standby mode, subactive mode, or watch mode on execution of SLEEP instruction in high-speed mode or medium-speed mode Transition to watch mode or high-speed mode on execution of SLEEP instruction in subactive mode

LPWRCR—Low-Power Control Register H'FF85 System

Bit	7	6	5	4	3	2	1	0
	DTON	LSON	NESEL	EXCLE	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Subclock input enable

0	Subclock input from EXCL pin disabled
1	Subclock input from EXCL pin enabled

Noise elimination sampling frequency select

0	Sampling at ø divided by 32
1	Sampling at ø divided by 4

Low-speed on flag

0	• Transition to sleep mode, software standby mode, or watch mode* on execution of SLEEP instruction in high-speed mode or medium-speed mode • Transition to watch mode, or direct transition to high-speed mode, on execution of SLEEP instruction in subactive mode • Transition to high-speed mode after watch mode is cleared
1	• Transition to watch mode or subactive mode* on execution of SLEEP instruction in high-speed mode • Transition to subsleep mode or watch mode on execution of SLEEP instruction in subactive mode • Transition to subactive mode after watch mode is cleared

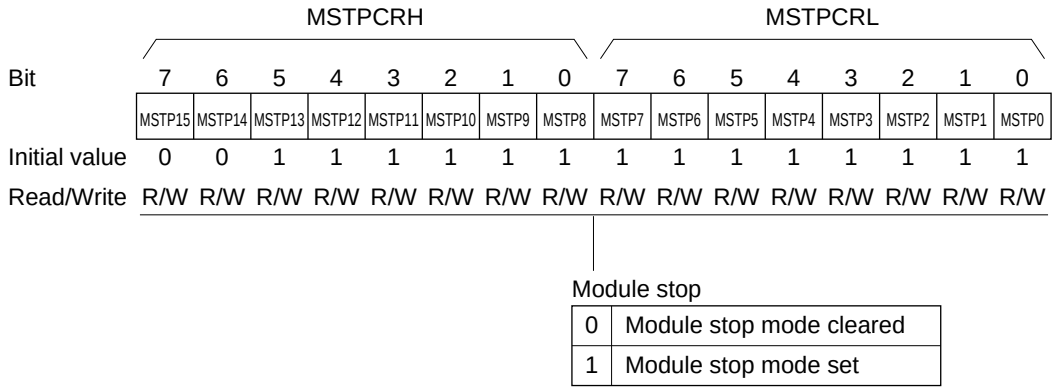
Note: * When a transition is made to watch mode or subactive mode, high-speed mode must be set.

Direct transfer on flag

0	• Transition to sleep mode, software standby mode, or watch mode* on execution of SLEEP instruction in high-speed mode or medium-speed mode • Transition to subsleep mode or watch mode on execution of SLEEP instruction in subactive mode
1	• Direct transition to subactive mode*, or transition to sleep mode or software standby mode, on execution of SLEEP instruction in high-speed mode or medium-speed mode • Direct transition to high-speed mode, or transition to subsleep mode, on execution of SLEEP instruction in subactive mode

Note: * When a transition is made to watch mode or subactive mode, high-speed mode must be set.

MSTPCRH—Module Stop Control Register H **H'FF86** **System**
MSTPCRL—Module Stop Control Register L **H'FF87** **System**



The correspondence between MSTPCR bits and on-chip supporting modules is shown below.

Register	Bit	Module
MSTPCRH	MSTP15	—
	MSTP14*	Data transfer controller (DTC)
	MSTP13	16-bit free-running timer (FRT)
	MSTP12	8-bit timers (TMR0, TMR1)
	MSTP11*	8-bit PWM timer (PWM), 14-bit PWM timer (PWMX)
	MSTP10*	—
	MSTP9	A/D converter
	MSTP8	8-bit timers (TMRX, TMRY), timer connection
MSTPCRL	MSTP7	Serial communication interface 0 (SCI0)
	MSTP6*	Serial communication interface 1 (SCI1)
	MSTP5*	—
	MSTP4*	I^2C bus interface (IIC) channel 0 (option)
	MSTP3*	I^2C bus interface (IIC) channel 1 (option)
	MSTP2*	—
	MSTP1*	—
	MSTP0*	—

Note: Bits 10, 5, 2, 1, and 0 can be read and written but do not affect operation.
* Must be set to 1 in the H8S/2124 Series.

SMR1—Serial Mode Register 1 **H'FF88** **SCI1**
SMR0—Serial Mode Register 0 **H'FFD8** **SCI0**

Bit	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select 1 and 0

0	0	ø clock
	1	ø/4 clock
1	0	ø/16 clock
	1	ø/64 clock

Multiprocessor mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop bit length

0	1 stop bit
1	2 stop bits

Parity mode

0	Even parity
1	Odd parity

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and the choice of LSB-first or MSB-first mode is not available.

Communication mode

0	Asynchronous mode
1	Synchronous mode

ICCR1—I²C Bus Control Register 1 **H'FF88** **IIC1**
ICCR0—I²C Bus Control Register 0 **H'FFD8** **IIC0**

Bit	7	6	5	4	3	2	1	0
	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	W

Start condition/stop condition prohibit

0	Writing issues a start or stop condition, in combination with the BBSY flag
1	Reading always returns a value of 1; writing is ignored

I²C bus interface interrupt request flag

0	Waiting for transfer, or transfer in progress
1	Interrupt requested

Note: For the clearing and setting conditions, see section 16.2.5, I²C Bus Control Register (ICCR).

Bus busy

0	Bus is free [Clearing condition] When a stop condition is detected
1	Bus is busy [Setting condition] When a start condition is detected

I²C bus interface interrupt enable

0	Interrupt requests disabled
1	Interrupt requests enabled

Acknowledge mode select

0	Acknowledge bit is ignored and transfer is performed continuously
1	When acknowledge bit is 1, continuous transfer is discontinued

I²C bus interface enable

0	Module is non-operational (SCL/SDA pin has port function) SAR and SARX can be accessed
1	Module is enabled for transfer operations (SC/SDA pin in bus drive state) ICMR and ICDR can be accessed

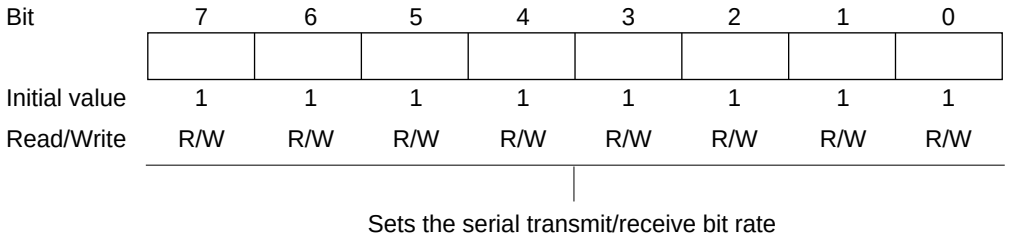
Master/slave select (MST), transmit/receive select (TRS)

0	0	Slave receive mode
	1	Slave transmit mode
1	0	Master receive mode
	1	Master transmit mode

Note: For details, see section 16.2.5, I²C Bus Control Register (ICCR).

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

BRR1—Bit Rate Register 1 **H'FF89** **SCI1**
BRR0—Bit Rate Register 0 **H'FFD9** **SCI0**



ICSR1—I²C Bus Status Register 1 **H'FF89** **IIC1**
ICSR0—I²C Bus Status Register 0 **H'FFD9** **IIC0**

Bit	7	6	5	4	3	2	1	0
	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/W

Acknowledge bit

0	Receive mode: 0 is output at acknowledge output timing Transmit mode: indicates that the receiving device has acknowledged the data (0 value)
1	Receive mode: 1 is output at acknowledge output timing Transmit mode: indicates that the receiving device has not acknowledged the data (1 value)

General call address recognition flag*2

0	General call address not recognized
1	General call address recognized

Slave address recognition flag*2

0	Slave address or general call address not recognized
1	Slave address or general call address recognized

Arbitration lost flag*2

0	Bus arbitration won
1	Bus arbitration lost

Second slave address recognition flag*2

0	Second slave address not recognized
1	Second slave address recognized

I²C bus interface continuous transmission/reception interrupt request flag*2

0	Waiting for transfer, or transfer in progress
1	Continuous transfer state

Normal stop condition detection flag*2

0	No normal stop condition
1	In I²C bus format slave mode: Normal stop condition detected In other modes: No meaning

Error stop condition detection flag*2

0	No error stop condition
1	In I²C bus format slave mode: Error stop condition detected In other modes: No meaning

Notes: 1. Only 0 can be written, to clear the flag.
2. For the clearing and setting conditions, see section 16.2.6, I²C Bus Status Register (ICSR).

SCR1—Serial Control Register 1 **H'FF8A** **SCI1**
SCR0—Serial Control Register 0 **H'FFDA** **SCI0**

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock enable 1 and 0

0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port
	1	Synchronous mode	Internal clock/SCK pin functions as serial clock output
	0	Asynchronous mode	Internal clock/SCK pin functions as clock output
	1	Synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input
	1	Synchronous mode	External clock/SCK pin functions as serial clock input
	0	Asynchronous mode	External clock/SCK pin functions as clock input
	1	Synchronous mode	External clock/SCK pin functions as serial clock input

Transmit end interrupt enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 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

Transmit interrupt enable

0	Transmit data empty interrupt (TXI) request disabled
1	Transmit data empty interrupt (TXI) request 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

Receive enable

0	Reception disabled
1	Reception enabled

Transmit enable

0	Transmission disabled
1	Transmission enabled

RDR1—Receive Data Register 1 **H'FF8D** **SCI1**
RDR0—Receive Data Register 0 **H'FFDD** **SCI0**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Stores serial receive data

TDR1—Transmit Data Register 1 **H'FF8B** **SCI1**
TDR0—Transmit Data Register 0 **H'FFDB** **SCI0**

Bit	7	6	5	4	3	2	1	0
Initial value	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

Stores serial transmit data

SSR1—Serial Status Register 1 **H'FF8C** **SCI1**
SSR0—Serial Status Register 0 **H'FFDC** **SCI0**

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor bit transfer

0	Data with a 0 multiprocessor bit is transmitted
1	Data with a 1 multiprocessor bit is transmitted

Multiprocessor bit

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit end

0	[Clearing conditions] • When 0 is written in 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 transmit character

Parity error

0	[Clearing condition] When 0 is written in PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR

Framing error

0	[Clearing condition] When 0 is written in FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data is 1 when reception ends, and the stop bit is 0

Overrun error

0	[Clearing condition] When 0 is written in 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 in 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 in 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 in TDR

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

SCMR1—Serial Interface Mode Register 1 **H'FF8E** **SCI1**
SCMR0—Serial Interface Mode Register 0 **H'FFDE** **SCI0**

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

Serial communication interface mode select

0	Normal SCI mode
1	Setting prohibited

Data invert

0	TDR contents are transmitted as they are TDR contents are stored in RDR as they are
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Data transfer direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

ICDR1—I²C Bus Data Register 1 **H'FF8E** **IIC1**
ICDR0—I²C Bus Data Register 0 **H'FFDE** **IIC0**

Bit	7	6	5	4	3	2	1	0
	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ICDRR

Bit	7	6	5	4	3	2	1	0
	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R	R	R	R	R	R	R	R

ICDRS

Bit	7	6	5	4	3	2	1	0
	ICDRS7	ICDRS6	ICDRS5	ICDRS4	ICDRS3	ICDRS2	ICDRS1	ICDRS0
Initial value	—	—	—	—	—	—	—	—
Read/Write	—	—	—	—	—	—	—	—

ICDRT

Bit	7	6	5	4	3	2	1	0
	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0
Initial value	—	—	—	—	—	—	—	—
Read/Write	W	W	W	W	W	W	W	W

TDRE, RDRF (internal flags)

Bit							—	—
							TDRE	RDRF
Initial value							0	0
Read/Write							—	—

Note: For details, see section 16.2.1, I²C Bus Data Register (ICDR).

SARX1—Second Slave Address Register 1 **H'FF8E** **IIC1**
SAR1—Slave Address Register 1 **H'FF8F** **IIC1**
SARX0—Second Slave Address Register 0 **H'FFDE** **IIC0**
SAR0—Slave Address Register 0 **H'FFDF** **IIC0**

SAR

Bit	7	6	5	4	3	2	1	0
	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Slave address — Format select

SARX

Bit	7	6	5	4	3	2	1	0
	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Second slave address — Format select

DDCSWR Bit 6	SAR Bit 0	SARX Bit 0	Operating Mode
SW	FS	FSX	
0	0	0	I²C bus format • SAR and SARX slave addresses recognized
		1	I²C bus format • SAR slave address recognized • SARX slave address ignored
	1	0	I²C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	0	0	Formatless mode (start/stop conditions not detected) • Acknowledge bit present
		1	
	1	0	
		1	Formatless mode* (start/stop conditions not detected) • No acknowledge bit

Note: * Do not select this mode when automatic switching to the I²C bus format is performed by means of a DDCSWR setting.

ICMR1—I²C Bus Mode Register 1 **H'FF8F** **IIC1**
ICMR0—I²C Bus Mode Register 0 **H'FFDF** **IIC0**

Bit	7	6	5	4	3	2	1	0
	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit counter

BC2	BC1	BC0	Synchronous serial format	I²C bus format
0	0	0	8	9
		1	1	2
	1	0	2	3
		1	3	4
1	0	0	4	5
		1	5	6
	1	0	6	7
		1	7	8

Transfer clock select

IICX	CKS2	CKS1	CKS0	Clock
0	0	0	0	ø/28
			1	ø/40
		1	0	ø/48
			1	ø/64
	1	0	0	ø/80
			1	ø/100
		1	0	ø/112
			1	ø/128
1	0	0	0	ø/56
			1	ø/80
		1	0	ø/96
			1	ø/128
	1	0	0	ø/160
			1	ø/200
		1	0	ø/224
			1	ø/256

Wait insertion bit

0	Data and acknowledge transferred consecutively
1	Wait inserted between data and acknowledge

MSB-first/LSB-first select*

0	MSB-first
1	LSB-first

Note: * Do not set this bit to 1 when using the I²C bus format.

TIER—Timer Interrupt Enable Register **H'FF90** **FRT**

Bit	7	6	5	4	3	2	1	0
	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

Timer overflow interrupt enable

0	OVF interrupt request (FOVI) is disabled
1	OVF interrupt request (FOVI) is enabled

Output compare interrupt B enable

0	OCFB interrupt request (OCIB) is disabled
1	OCFB interrupt request (OCIB) is enabled

Output compare interrupt A enable

0	OCFA interrupt request (OCIA) is disabled
1	OCFA interrupt request (OCIA) is enabled

Input capture interrupt D enable

0	ICFD interrupt request (ICID) is disabled
1	ICFD interrupt request (ICID) is enabled

Input capture interrupt C enable

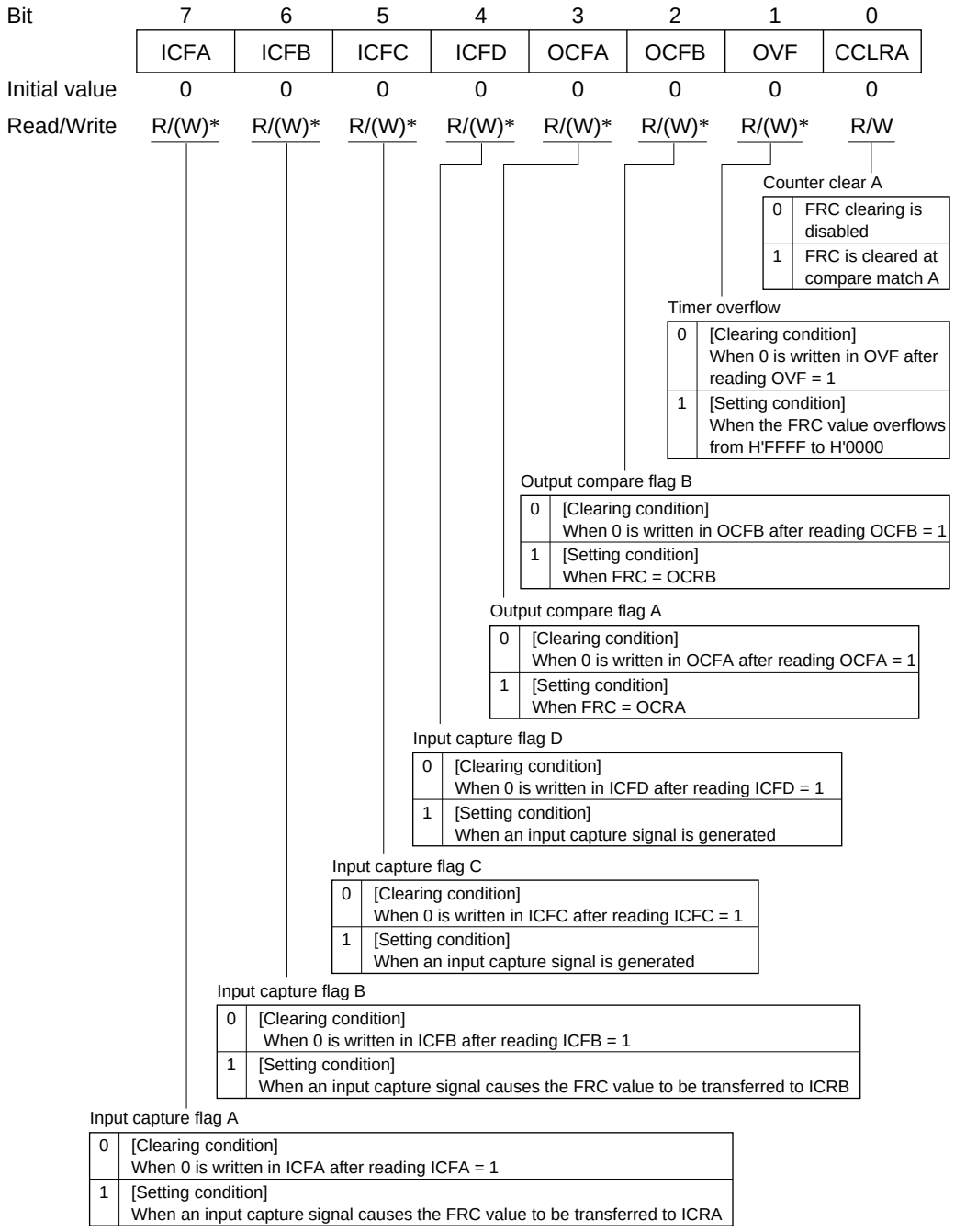
0	ICFC interrupt request (ICIC) is disabled
1	ICFC interrupt request (ICIC) is enabled

Input capture interrupt B enable

0	ICFB interrupt request (ICIB) is disabled
1	ICFB interrupt request (ICIB) is enabled

Input capture interrupt A enable

0	ICFA interrupt request (ICIA) is disabled
1	ICFA interrupt request (ICIA) is enabled



Note: * Only 0 can be written in bits 7 to 1, to clear the flags.

FRC—Free-Running Counter **H'FF92** **FRT**

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

Count value

OCRA/OCRB—Output Compare Register A/B **H'FF94** **FRT**

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

Constantly compared with FRC value; OCF is set when OCR = FRC

Bit	7	6	5	4	3	2	1	0
	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select

0	0	Internal clock: counting on ø/2
	1	Internal clock: counting on ø/8
1	0	Internal clock: counting on ø/32
	1	External clock: counting on rising edge

Buffer enable B

0	ICRD is not used as ICRB buffer register
1	ICRD is used as ICRB buffer register

Buffer enable A

0	ICRC is not used as ICRA buffer register
1	ICRC is used as ICRA buffer register

Input edge select D

0	Capture at falling edge of input capture input D
1	Capture at rising edge of input capture input D

Input edge select C

0	Capture at falling edge of input capture input C
1	Capture at rising edge of input capture input C

Input edge select B

0	Capture at falling edge of input capture input B
1	Capture at rising edge of input capture input B

Input edge select A

0	Capture at falling edge of input capture input A
1	Capture at rising edge of input capture input A

Bit	7	6	5	4	3	2	1	0
	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output level B

0	0 output at compare match B
1	1 output at compare match B

Output level A

0	0 output at compare match A
1	1 output at compare match A

Output enable B

0	Output compare B output disabled
1	Output compare B output enabled

Output enable A

0	Output compare A output disabled
1	Output compare A output enabled

Output compare register select

0	OCRA register selected
1	OCRB register selected

Input capture register select

0	ICRA, ICRB, and ICRC registers selected
1	OCRAR, OCRAF, and OCRDM registers selected

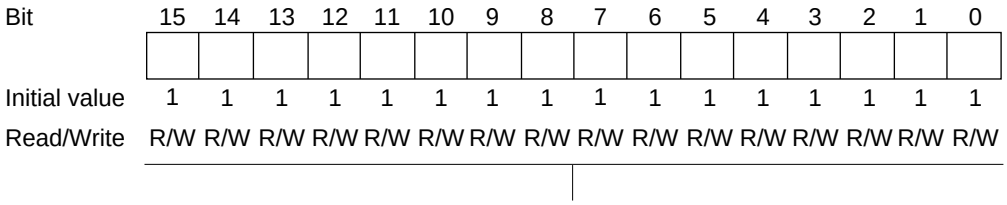
Output compare A mode select

0	OCRA set to normal operating mode
1	OCRA set to operating mode using OCRAR and OCRAF

Input capture D mode select

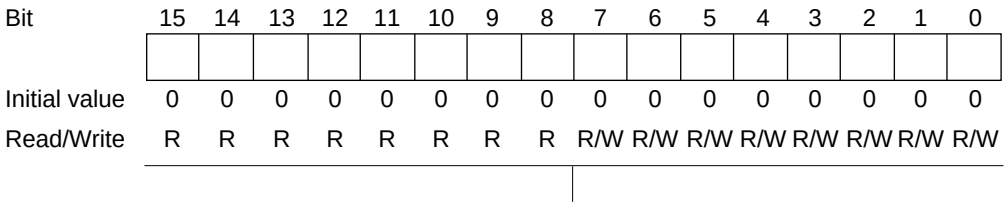
0	ICRD set to normal operating mode
1	ICRD set to operating mode using OCRDM

OCRAR—Output Compare Register AR **H'FF98** **FRT**
OCRAF—Output Compare Register AF **H'FF9A** **FRT**



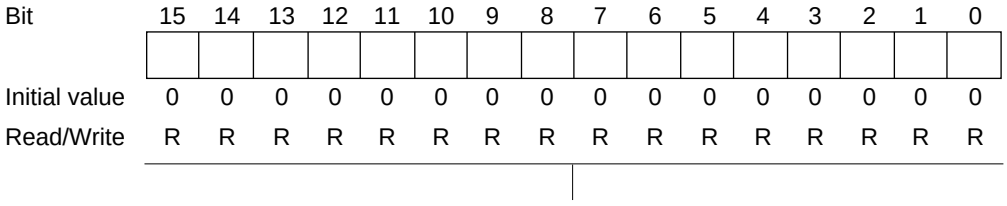
Used for OCRA operation when OCRAMS = 1 in TOCR
(For details, see section 11.2.4, Output Compare Registers AR and AF (OCRAR, OCRAF).)

OCRDM—Output Compare Register DM **H'FF9C** **FRT**



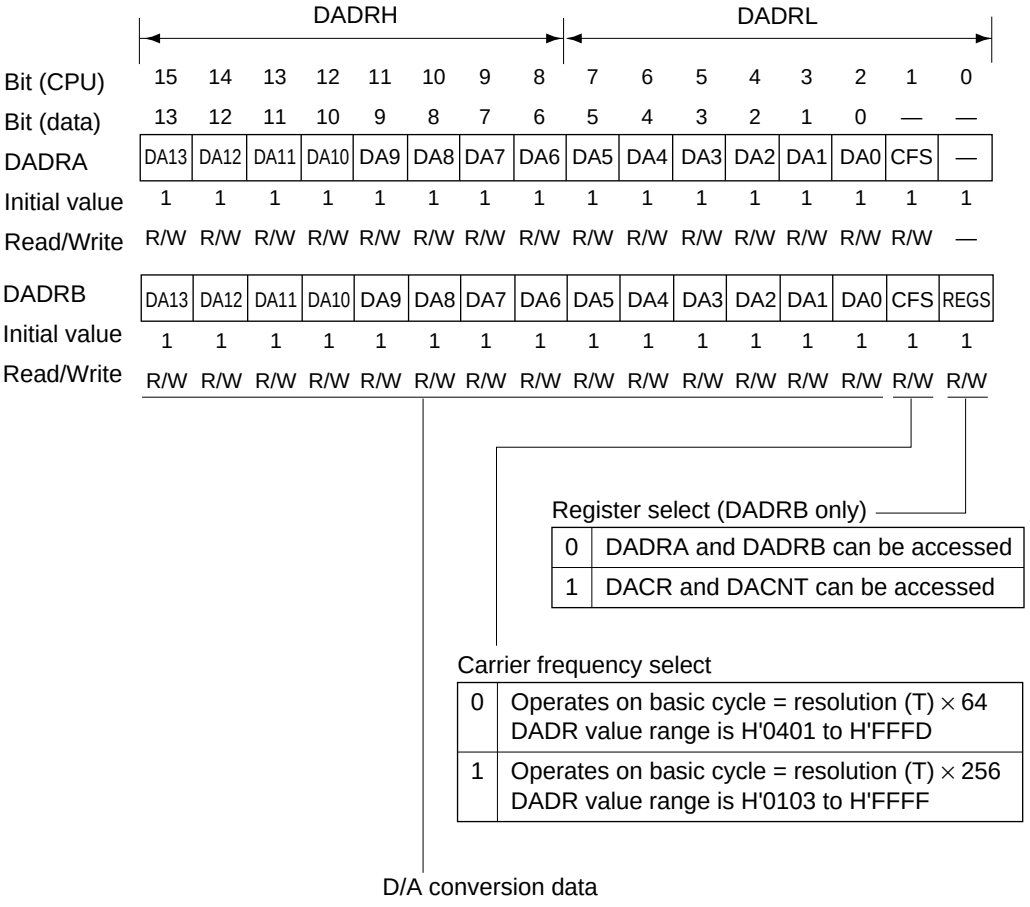
Used for ICRD operation when ICRDMS = 1 in TOCR
(For details, see section 11.2.5, Output Compare Register DM (OCRDM).)

ICRA—Input Capture Register A **H'FF98** **FRT**
ICRB—Input Capture Register B **H'FF9A** **FRT**
ICRC—Input Capture Register C **H'FF9C** **FRT**
ICRD—Input Capture Register D **H'FF9E** **FRT**



Stores FRC value when input capture signal is input
(ICRC and ICRD can be used for buffer operation.
For details, see section 11.2.3, Input Capture Registers A to D (ICRA to ICRD).)

DADRAH—PWM (D/A) Data Register AH	**H'FFA0**	**PWMX**
DADRAL—PWM (D/A) Data Register AL	**H'FFA1**	**PWMX**
DADRBH—PWM (D/A) Data Register BH	**H'FFA6**	**PWMX**
DADRBL—PWM (D/A) Data Register BL	**H'FFA7**	**PWMX**



DACR—PWM (D/A) Control Register **H'FFA0** **PWMX**

Bit	7	6	5	4	3	2	1	0
	TEST	PWME	—	—	OEB	OEA	OS	CKS
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

Clock select

0	Operates at resolution (T) = system clock cycle (t_{cyc})
1	Operates at resolution (T) = system clock cycle (t_{cyc}) × 2

Output select

0	PWM direct output
1	PWM inverted output

Output enable A

0	PWM (D/A) channel A output (PWX0 output pin) disabled
1	PWM (D/A) channel A output (PWX0 output pin) enabled

Output enable B

0	PWM (D/A) channel B output (PWX1 output pin) disabled
1	PWM (D/A) channel B output (PWX1 output pin) enabled

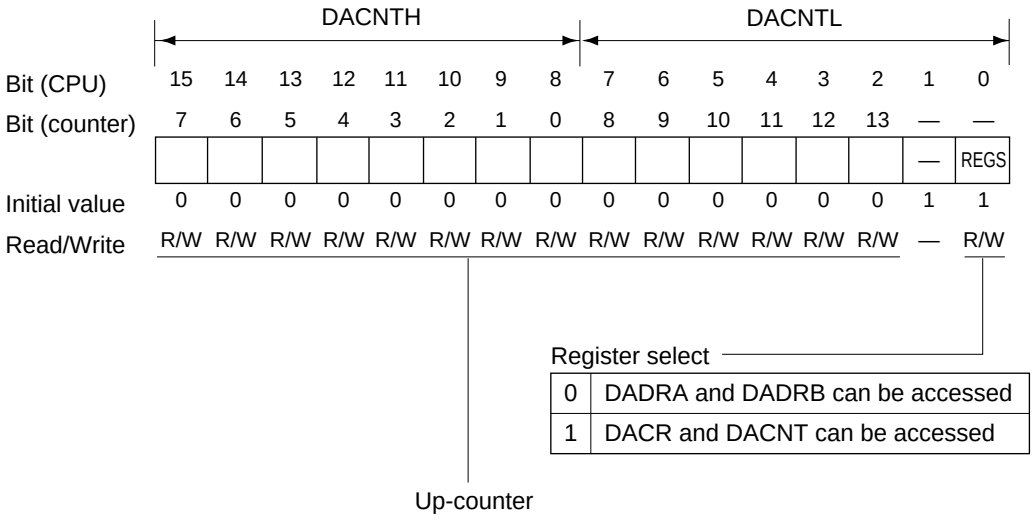
PWM enable

0	DACNT operates as 14-bit up-counter
1	Stop at DACNT = H'0003

Test mode

0	PWM (D/A) in user state, normal operation
1	PWM (D/A) in test state, correct conversion results unobtainable

DACNTH—PWM (D/A) Counter H **H'FFA6** **PWMX**
DACNTL—PWM (D/A) Counter L **H'FFA7** **PWMX**



Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	RSTS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select 2 to 0

CKS2	CKS1	CKS0	Clock
0	0	0	ø/2
		1	ø/64
	1	0	ø/128
		1	ø/512
1	0	0	ø/2048
		1	ø/8192
	1	0	ø/32768
		1	ø/131072

Reset or NMI

0	NMI interrupt requested
1	Internal reset requested

Reserved

Timer enable

0	TCNT is initialized to H'00 and halted
1	TCNT counts

Timer mode select

0	Interval timer mode: Interval timer interrupt request (WOVI) sent to CPU when TCNT overflows
1	Watchdog timer mode: Reset or NMI interrupt request sent to CPU when TCNT overflows

Overflow flag

0	[Clearing conditions] • When 0 is written in the TME bit • When 0 is written in OVF after reading TCSR when OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00 When internal reset request is selected in watchdog timer mode, OVF is cleared automatically by an internal reset after being set

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

TCNT0—Timer Counter 0 **H'FFA8 (W), H'FFA9 (R)** **WDT0**
TCNT1—Timer Counter 1 **H'FFEA (W), H'FFEB (R)** **WDT1**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Up-counter

P1PCR—Port 1 MOS Pull-Up Control Register **H'FFAC** **Port 1**

Bit	7	6	5	4	3	2	1	0
	P17PCR	P16PCR	P15PCR	P14PCR	P13PCR	P12PCR	P11PCR	P10PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Control of port 1 built-in MOS input pull-ups

P2PCR—Port 2 MOS Pull-Up Control Register **H'FFAD** **Port 2**

Bit	7	6	5	4	3	2	1	0
	P27PCR	P26PCR	P25PCR	P24PCR	P23PCR	P22PCR	P21PCR	P20PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Control of port 2 built-in MOS input pull-ups

P3PCR—Port 3 MOS Pull-Up Control Register **H'FFAE** **Port 3**

Bit	7	6	5	4	3	2	1	0
	P37PCR	P36PCR	P35PCR	P34PCR	P33PCR	P32PCR	P31PCR	P30PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Control of port 3 built-in MOS input pull-ups

P1DDR—Port 1 Data Direction Register **H'FFB0** **Port 1**

Bit	7	6	5	4	3	2	1	0
	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specification of input or output for port 1 pins

P2DDR—Port 2 Data Direction Register **H'FFB1** **Port 2**

Bit	7	6	5	4	3	2	1	0
	P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specification of input or output for port 2 pins

P1DR—Port 1 Data Register **H'FFB2** **Port 1**

Bit	7	6	5	4	3	2	1	0
	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 1 pins

P2DR—Port 2 Data Register **H'FFB3** **Port 2**

Bit	7	6	5	4	3	2	1	0
	P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 2 pins

P3DDR—Port 3 Data Direction Register **H'FFB4** **Port 3**

Bit	7	6	5	4	3	2	1	0
	P37DDR	P36DDR	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specification of input or output for port 3 pins

P4DDR—Port 4 Data Direction Register **H'FFB5** **Port 4**

Bit	7	6	5	4	3	2	1	0
	P47DDR	P46DDR	P45DDR	P44DDR	P43DDR	P42DDR	P41DDR	P40DDR
Mode 1								
Initial value	0	1	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W
Modes 2 and 3								
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specification of input or output for port 4 pins

P3DR—Port 3 Data Register **H'FFB6** **Port 3**

Bit	7	6	5	4	3	2	1	0
	P37DR	P36DR	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 3 pins

P4DR—Port 4 Data Register **H'FFB7** **Port 4**

Bit	7	6	5	4	3	2	1	0
	P47DR	P46DR	P45DR	P44DR	P43DR	P42DR	P41DR	P40DR
Initial value	0	—*	0	0	0	0	0	0
Read/Write	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 4 pins

Note: * Determined by state of pin P46.

P5DDR—Port 5 Data Direction Register **H'FFB8** **Port 5**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	P52DDR	P51DDR	P50DDR
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	W	W	W

Specification of input or output for port 5 pins

P6DDR—Port 6 Data Direction Register **H'FFB9** **Port 6**

Bit	7	6	5	4	3	2	1	0
	P67DDR	P66DDR	P65DDR	P64DDR	P63DDR	P62DDR	P61DDR	P60DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specification of input or output for port 6 pins

P5DR—Port 5 Data Register **H'FFBA** **Port 5**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	P52DR	P51DR	P50DR
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Stores output data for port 5 pins

P6DR—Port 6 Data Register **H'FFBB** **Port 6**

Bit	7	6	5	4	3	2	1	0
	P67DR	P66DR	P65DR	P64DR	P63DR	P62DR	P61DR	P60DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores output data for port 6 pins

P7PIN—Port 7 Input Data Register **H'FFBE** **Port 7**

Bit	7	6	5	4	3	2	1	0
	P77PIN	P76PIN	P75PIN	P74PIN	P73PIN	P72PIN	P71PIN	P70PIN
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Port 7 pin states

Note: * Determined by state of pins P77 to P70.

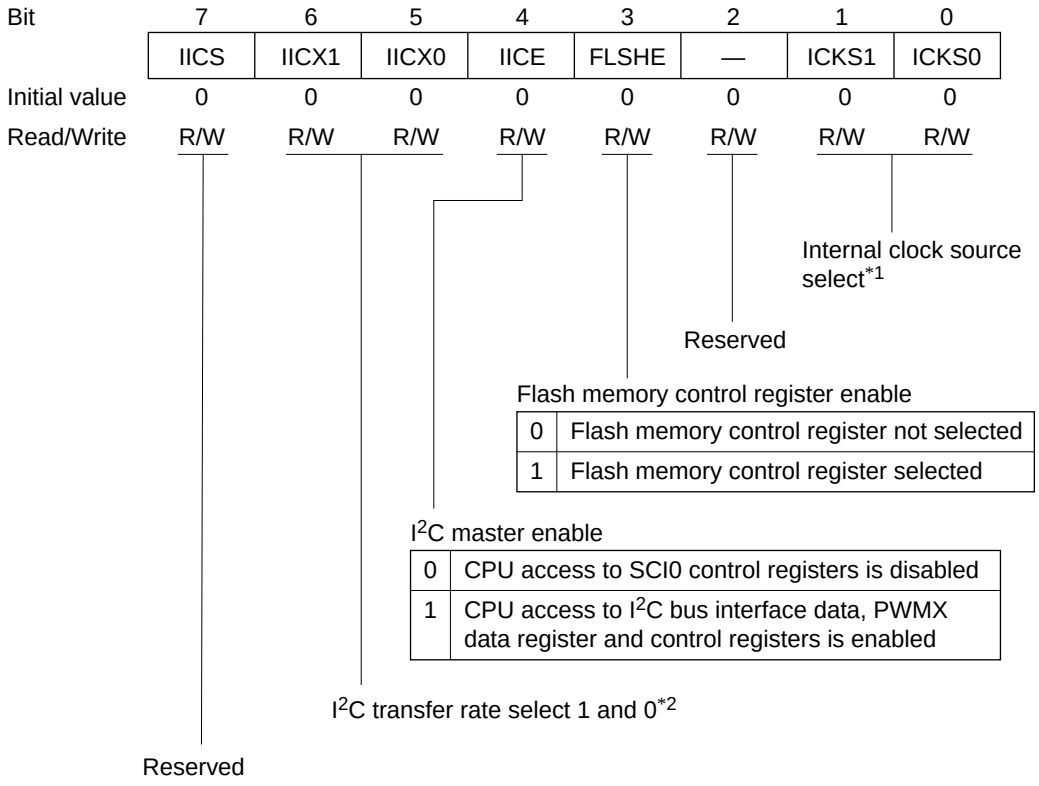
IER—IRQ Enable Register **H'FFC2** **Interrupt Controller**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	IRQ2E	IRQ1E	IRQ0E
Initial value	1	1	1	1	1	0	0	0
Read/Write	R	R	R	R	R	R/W	R/W	R/W

IRQ2 to IRQ0 enable

0	IRQn interrupt disabled
1	IRQn interrupt enabled

(n = 0 to 2)



Notes: 1. Used for 8-bit timer input clock selection. For details, see section 12.2.4, Timer Control Register (TCR).
2. Used for I^2C bus interface transfer clock selection. For details, see section 16.2.4, I^2C Bus Mode Register (ICMR).

Bit	7	6	5	4	3	2	1	0
	CS2E	IOSE	INTM1	INTM0	XRST	NMIEG	HIE	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R	R/W	R	R/W	R/W	R/W

RAM Enable

0	On-chip RAM is disabled
1	On-chip RAM is enabled

Host interface enable

0	Access to 8-bit timer (channel X and Y) data registers and control registers, and timer connection control registers is disabled
1	Access to 8-bit timer (channel X and Y) data registers and control registers, and timer connection control registers is enabled

NMI edge select

0	Falling edge
1	Rising edge

External reset

0	Reset generated by watchdog timer overflow
1	Reset generated by an external reset

Reserved

Interrupt control mode select

0	0	Interrupt control mode 0
	1	Interrupt control mode 1

IOS enable

0	$\overline{AS}$/$\overline{IOS}$ is address strobe pin (low output in external area access)
1	$\overline{AS}$/$\overline{IOS}$ is I/O strobe pin (low output when accessing specified address from H'(FF)F000 to H(FF)FE4F)

MDCR—Mode Control Register **H'FFC5** **System**

Bit	7	6	5	4	3	2	1	0
	EXPE	—	—	—	—	—	MDS1	MDS0
Initial value	—*	0	0	0	0	0	—*	—*
Read/Write	R/W*	—	—	—	—	—	R	R

Current mode pin operating mode

Expanded mode enable

0	Single-chip mode selected
1	Expanded mode selected

Note: * Determined by pins MD1 and MD0.

Bit	7	6	5	4	3	2	1	0
	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0
Initial value	1	1	0	1	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IOS select

IOS1	IOS0	Addresses for which $\overline{AS}$/$\overline{IOS}$ pin output goes low when IOSE = 1
0	0	Low in accesses to addresses H'(FF)F000 to H'(FF)F03F
	1	Low in accesses to addresses H'(FF)F000 to H'(FF)F0FF
1	0	Low in accesses to addresses H'(FF)F000 to H'(FF)F3FF
	1	Low in accesses to addresses H'(FF)F000 to H'(FF)FE4F

Burst cycle select 0

0	Max. 4 words in burst access
1	Max. 8 words in burst access

Burst cycle select 1

0	Burst cycle comprises 1 state
1	Burst cycle comprises 2 states

Burst ROM enable

0	Basic bus interface
1	Burst ROM interface

Reserved

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

Bit	7	6	5	4	3	2	1	0
	RAMS	RAM0	ABW	AST	WMS1	WMS0	WC1	WC0
Initial value	0	0	1	1	0	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reserved

Wait count 1 and 0

0	0	No programmable waits inserted
	1	1 programmable wait state inserted in external memory space access
1	0	2 programmable wait states inserted in external memory space access
	1	3 programmable wait states inserted in external memory space access

Wait mode select 1 and 0

0	0	Programmable wait mode
	1	Wait disabled mode
1	0	Pin wait mode
	1	Pin auto-wait mode

Access state control

0	External memory space designated as 2-state access space Wait state insertion in external memory space access is disabled
1	External memory space designated as 3-state access space Wait state insertion in external memory space access is enabled

Bus width control

0	External memory space designated as 16-bit access space
1	External memory space designated as 8-bit access space

TCR0—Timer Control Register 0 **H'FFC8** **TMR0**
TCR1—Timer Control Register 1 **H'FFC9** **TMR1**
TCRX—Timer Control Register X **H'FFF0** **TMRX**
TCRY—Timer Control Register Y **H'FFF0** **TMRY**

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Compare Match Interrupt Enable B

0	CMFB interrupt request (CMIB) is disabled
1	CMFB interrupt request (CMIB) is enabled

Compare match interrupt enable A

0	CMFA interrupt request (CMIA) is disabled
1	CMFA interrupt request (CMIA) is enabled

Timer overflow interrupt enable

0	OVF interrupt request (OVI) is disabled
1	OVF interrupt request (OVI) is enabled

Counter clear 1 and 0

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

Clock select 2 to 0

Channel	Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Description
0	0	0	0	Clock input disabled
			1*1	Internal clock: counting at falling edge of ø/8
		1	0	Internal clock: counting at falling edge of ø/2
			0*1	Internal clock: counting at falling edge of ø/64
			1	Internal clock: counting at falling edge of ø/32
		0	1*1	Internal clock: counting at falling edge of ø/1024
			1	Internal clock: counting at falling edge of ø/256
	1	0	0	Counting at TCNT1 overflow signal*2
1	0	0	0	Clock input disabled
			1*1	Internal clock: counting at falling edge of ø/8
		1	0	Internal clock: counting at falling edge of ø/2
			0*1	Internal clock: counting at falling edge of ø/64
			1	Internal clock: counting at falling edge of ø/128
		0	1*1	Internal clock: counting at falling edge of ø/1024
			1	Internal clock: counting at falling edge of ø/2048
	1	0	0	Count at TCNT0 compare match A*2
X	0	0	0	Clock input disabled
			1	Internal clock: counting on ø
		1	0	Internal clock: counting at falling edge of ø/2
			1	Internal clock: counting at falling edge of ø/4
	1	0	0	Clock input disabled
Y	0	0	0	Clock input disabled
			1	Internal clock: counting at falling edge of ø/4
		1	0	Internal clock: counting at falling edge of ø/256
			1	Internal clock: counting at falling edge of ø/2048
	1	0	0	Clock input disabled
All	1	0	1	External clock: counting at rising edge
		1	0	External clock: counting at falling edge
			1	External clock: counting at both rising and falling edges

Notes: 1. Selected by ICKS1 and ICKS0 in STCR. For details, see section 12.2.4, Timer Control Register (TCR).
2. If the clock input of channel 0 is the TCNT1 overflow signal and that of channel 1 is the TCNT0 compare match signal, no incrementing clock is generated. Do not use this setting.

TCSR0

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Output select 1 and 0

0	0	No change at compare match A
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Output inverted at compare match A (toggle output)

Output select 3 and 2

0	0	No change at compare match B
	1	0 output at compare match B
1	0	1 output at compare match B
	1	Output inverted at compare match B (toggle output)

A/D trigger enable

0	A/D converter start requests by compare match A are disabled
1	A/D converter start requests by compare match A are enabled

Timer overflow flag

0	[Clearing condition] When 0 is written in OVF after reading OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00

Compare match flag A

0	[Clearing conditions] • When 0 is written in CMFA after reading CMFA = 1 • When the DTC is activated by a CMIA interrupt
1	[Setting condition] When TCNT = TCORA

Compare match flag B

0	[Clearing conditions] • When 0 is written in CMFB after reading CMFB = 1 • When the DTC is activated by a CMIB interrupt
1	[Setting condition] When TCNT = TCORB

Note: * Only 0 can be written in bits 7 to 5, to clear the flags.

TCSR1—Timer Control/Status Register 1 **H'FFCB** **TMR1**

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
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/W

Output select 1 and 0

0	0	No change at compare match A
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Output inverted at compare match A (toggle output)

Output select 3 and 2

0	0	No change at compare match B
	1	0 output at compare match B
1	0	1 output at compare match B
	1	Output inverted at compare match B (toggle output)

Timer overflow flag

0	[Clearing condition] When 0 is written in OVF after reading OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00

Compare match flag A

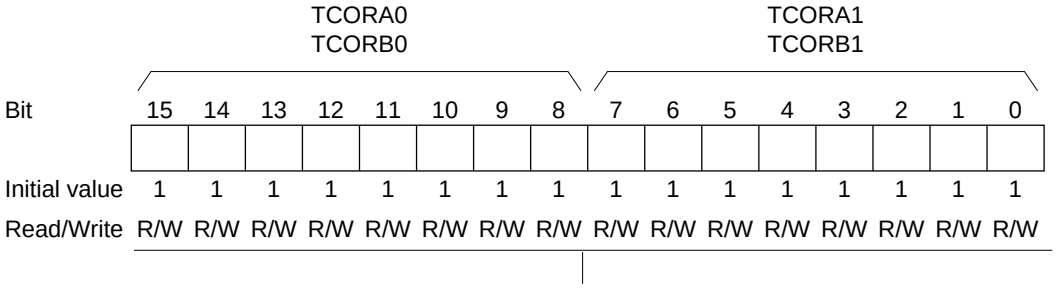
0	[Clearing conditions] • When 0 is written in CMFA after reading CMFA = 1 • When the DTC is activated by a CMIA interrupt
1	[Setting condition] When TCNT = TCORA

Compare match flag B

0	[Clearing conditions] • When 0 is written in CMFB after reading CMFB = 1 • When the DTC is activated by a CMIB interrupt
1	[Setting condition] When TCNT = TCORB

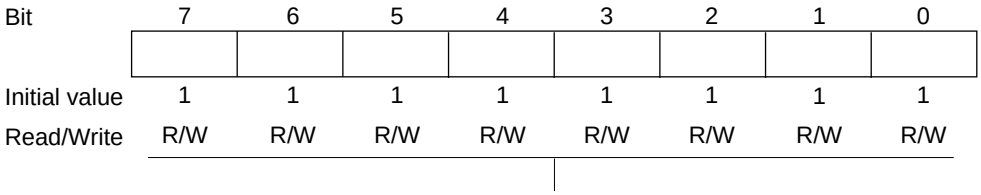
Note: * Only 0 can be written in bits 7 to 5, to clear the flags.

TCORA0—Time Constant Register A0	**H'FFCC**	**TMR0**
TCORA1—Time Constant Register A1	**H'FFCD**	**TMR1**
TCORB0—Time Constant Register B0	**H'FFCE**	**TMR0**
TCORB1—Time Constant Register B1	**H'FFCF**	**TMR1**
TCORAY—Time Constant Register AY	**H'FFF2**	**TMRY**
TCORBY—Time Constant Register BY	**H'FFF3**	**TMRY**
TCORC—Time Constant Register C	**H'FFF5**	**TMRX**
TCORAX—Time Constant Register AX	**H'FFF6**	**TMRX**
TCORBX—Time Constant Register BX	**H'FFF7**	**TMRX**



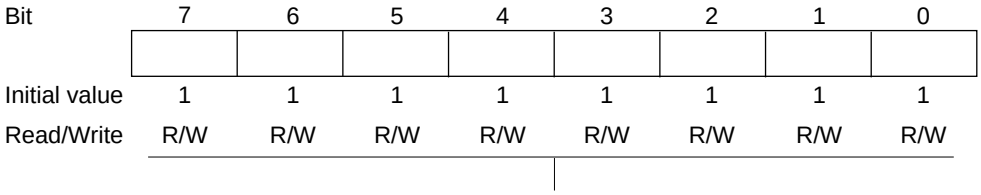
Compare match flag (CMF) is set when TCOR and TCNT values match

TCORAX, TCORAY
TCORBX, TCORBY



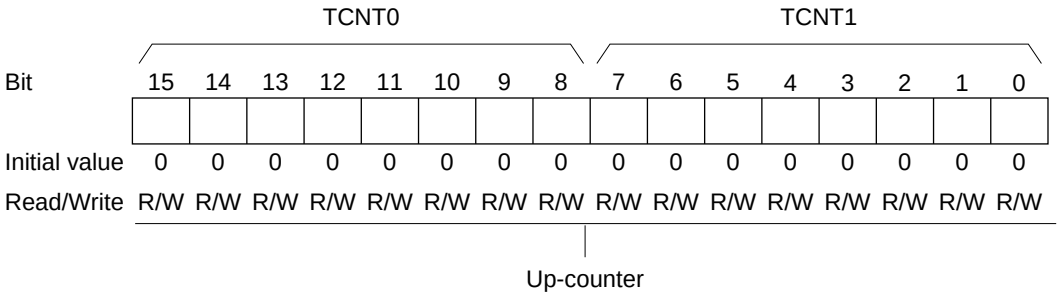
Compare match flag (CMF) is set when TCOR and TCNT values match

TCORC

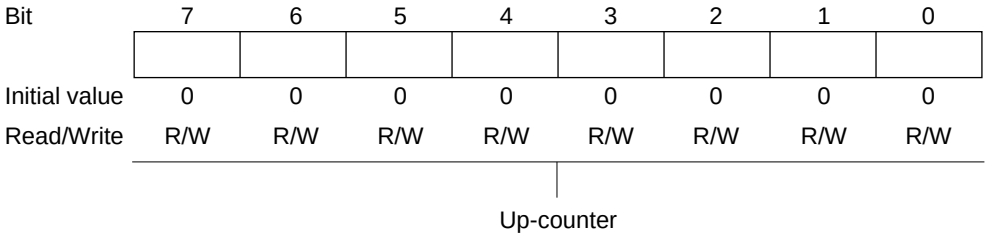


Compare match C signal is generated when sum of TCORC and TICR contents match TCNT value

TCNT0—Timer Counter 0	**H'FFD0**	**TMR0**
TCNT1—Timer Counter 1	**H'FFD1**	**TMR1**
TCNTX—Timer Counter X	**H'FFF4**	**TMRX**
TCNTY—Timer Counter Y	**H'FFF4**	**TMRY**



TCNTX, TCNTY



PWOERA—PWM Output Enable Register A **H'FFD3** **PWM**
PWOERB—PWM Output Enable Register B **H'FFD2** **PWM**

Bit	7	6	5	4	3	2	1	0
PWOERA	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
PWOERB	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Switching between PWM output and port output

DDR	OE	Description
0	0	Port input
	1	Port input
1	0	Port output or PWM 256/256 output
	1	PWM output (0 to 255/256 output)

PWDPRA—PWM Data Polarity Register A **H'FFD5** **PWM**
PWDPRB—PWM Data Polarity Register B **H'FFD4** **PWM**

Bit	7	6	5	4	3	2	1	0
PWDPRA	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
PWDPRB	OS15	OS14	OS13	OS12	OS11	OS10	OS9	OS8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM output polarity control

0	PWM direct output (PWDR value corresponds to high width of output)
1	PWM inverted output (PWDR value corresponds to low width of output)

Bit	7	6	5	4	3	2	1	0
	PWCKE	PWCKS	—	—	RS3	RS2	RS1	RS0
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

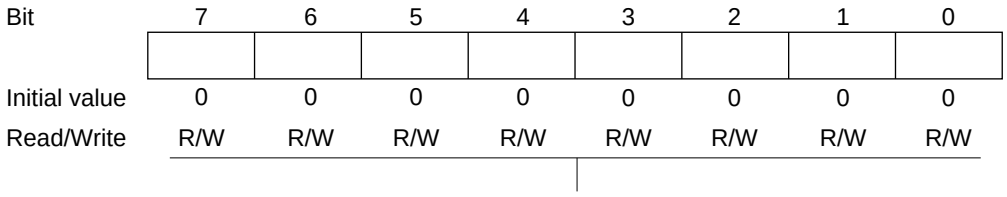
Register Select

0	0	0	0	PWDR0 selected
			1	PWDR1 selected
		1	0	PWDR2 selected
			1	PWDR3 selected
	1	0	0	PWDR4 selected
			1	PWDR5 selected
		1	0	PWDR6 selected
			1	PWDR7 selected
1	0	0	0	PWDR8 selected
			1	PWDR9 selected
		1	0	PWDR10 selected
			1	PWDR11 selected
	1	0	0	PWDR12 selected
			1	PWDR13 selected
		1	0	PWDR14 selected
			1	PWDR15 selected

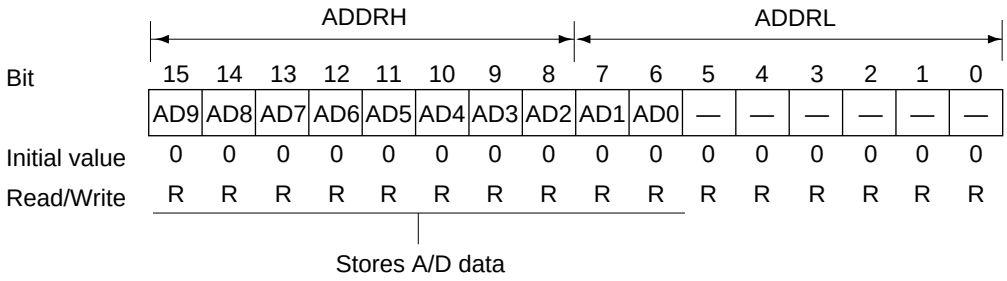
PWM clock enable, PWM clock select

PWSL		PCSR		Description
Bit 7	Bit 6	Bit 2	Bit 1	
PWCKE	PWCKS	PWCKB	PWCKA	
0	—	—	—	Clock input disabled
1	0	—	—	ø (system clock) selected
	1	0	0	ø/2 selected
			1	ø/4 selected
		1	0	ø/8 selected
			1	ø/16 selected

PWDR0 to PWDR15—PWM Data Registers H'FFD7 PWM



ADDRAH—A/D Data Register AH	**H'FFE0**	**A/D Converter**
ADDRAL—A/D Data Register AL	**H'FFE1**	**A/D Converter**
ADDRBH—A/D Data Register BH	**H'FFE2**	**A/D Converter**
ADDRBL—A/D Data Register BL	**H'FFE3**	**A/D Converter**
ADDRCH—A/D Data Register CH	**H'FFE4**	**A/D Converter**
ADDRCL—A/D Data Register CL	**H'FFE5**	**A/D Converter**
ADDRDH—A/D Data Register DH	**H'FFE6**	**A/D Converter**
ADDRDL—A/D Data Register DL	**H'FFE7**	**A/D Converter**



Correspondence between analog input channels and ADDR registers

Analog Input Channel		A/D Data Register
Group 0	Group 1	
AN0	AN4	ADDRA
AN1	AN5	ADDRB
AN2	AN6 or CIN0–CIN7	ADDRC
AN3	AN7	ADDRD

Bit	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel select

Group selection	Channel selection		Description	
CH2	CH1	CH0	Single mode	Scan mode
0	0	0	AN0	AN0
		1	AN1	AN0, AN1
	1	0	AN2	AN0, AN1, AN2
		1	AN3	AN0, AN1, AN2, AN3
1	0	0	AN4	AN4
		1	AN5	AN4, AN5
	1	0	AN6 or CIN0–7	AN4, AN5, AN6 or CIN0–7
		1	AN7	AN4, AN5, AN6 or CIN0–7, AN7

Clock 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 on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues consecutively on the selected channels until ADST is cleared to 0 by software, a reset, or a 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 ADF 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 for all specified channels

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

Bit	7	6	5	4	3	2	1	0
	TRGS1	TRGS0	—	—	—	—	—	—
Initial value	0	0	1	1	1	1	1	1
Read/Write	R/W	R/W	—	—	—	—	—	—

Timer trigger select

0	0	A/D conversion start by external trigger is disabled
	1	A/D conversion start by external trigger is disabled
1	0	A/D conversion start by external trigger (8-bit timer) is enabled
	1	A/D conversion start by external trigger pin is enabled

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	PSS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*1	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select 2 to 0

PSS	CSK2	CSK1	CSK0	Clock
0	0	0	0	ø/2
			1	ø/64
		1	0	ø/128
			1	ø/512
	1	0	0	ø/2048
			1	ø/8192
		1	0	ø/32768
			1	ø/131072
1	0	0	0	øSUB/2
			1	øSUB/4
		1	0	øSUB/8
			1	øSUB/16
	1	0	0	øSUB/32
			1	øSUB/64
		1	0	øSUB/128
			1	øSUB/256

Reset or NMI

0	NMI interrupt requested
1	Internal reset requested

Prescaler select*2

0	TCNT counts on a ø-based prescaler (PSM) scaled clock
1	TCNT counts on a øSUB-based prescaler (PSS) scaled clock

Timer enable

0	TCNT is initialized to H'00 and halted
1	TCNT counts

Timer mode select

0	Interval timer mode: Interval timer interrupt request (WOVI) sent to CPU when TCNT overflows
1	Watchdog timer mode: Reset or NMI interrupt request sent to CPU when TCNT overflows

Overflow flag

0	[Clearing conditions] • When 0 is written in the TME bit • When 0 is written in OVF after reading TCSR when OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00 When internal reset request is selected in watchdog timer mode, OVF is cleared automatically by an internal reset after being set

Notes: 1. Only 0 can be written, to clear the flag.
2. For operation control when a transition is made to power-down mode, see section 21.2.3, Timer Control/Status Register (TCSR).

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W

Output select 1 and 0

0	0	No change at compare match A
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Output inverted at compare match A (toggle output)

Output select 3 and 2

0	0	No change at compare match B
	1	0 output at compare match B
1	0	1 output at compare match B
	1	Output inverted at compare match B (toggle output)

Input capture flag

0	[Clearing condition] When 0 is written in ICF after reading ICF = 1
1	[Setting condition] When a rising edge followed by a falling edge is detected in the external reset signal after the ICST bit is set to 1 in TCONRI

Timer overflow flag

0	[Clearing condition] When 0 is written in OVF after reading OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00

Compare match flag A

0	[Clearing conditions] • When 0 is written in CMFA after reading CMFA = 1 • When the DTC is activated by a CMIA interrupt
1	[Setting condition] When TCNT = TCORA

Compare match flag B

0	[Clearing conditions] • When 0 is written in CMFB after reading CMFB = 1 • When the DTC is activated by a CMIB interrupt
1	[Setting condition] When TCNT = TCORB

Note: * Only 0 can be written in bits 7 to 4, to clear the flags.

TCSRY—Timer Control/Status Register Y **H'FFF1** **TMRY**

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Output select 1 and 0

0	0	No change at compare match A
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Output inverted at compare match A (toggle output)

Output select 3 and 2

0	0	No change at compare match B
	1	0 output at compare match B
1	0	1 output at compare match B
	1	Output inverted at compare match B (toggle output)

Input capture interrupt enable

0	ICF interrupt request (ICIX) is disabled
1	ICF interrupt request (ICIX) is enabled

Timer overflow flag

0	[Clearing condition] When 0 is written in OVF after reading OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00

Compare match flag A

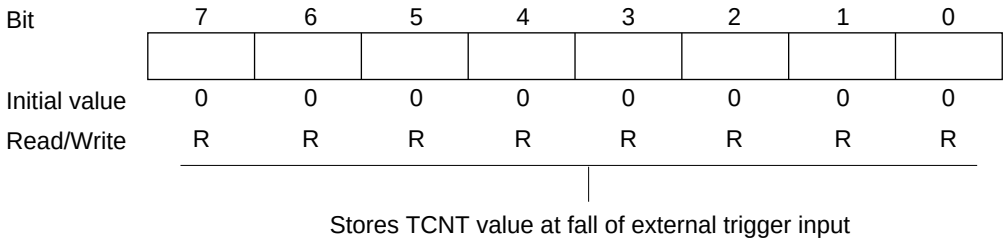
0	[Clearing conditions] • When 0 is written in CMFA after reading CMFA = 1 • When the DTC is activated by a CMIA interrupt
1	[Setting condition] When TCNT = TCORA

Compare match flag B

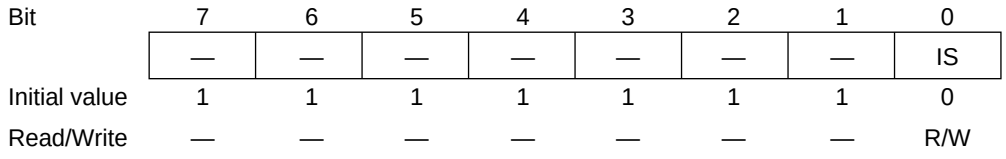
0	[Clearing conditions] • When 0 is written in CMFB after reading CMFB = 1 • When the DTC is activated by a CMIB interrupt
1	[Setting condition] When TCNT = TCORB

Note: * Only 0 can be written in bits 7 to 5, to clear the flags.

TICRR—Input Capture Register R **H'FFF2** **TMRX**
TICRF—Input Capture Register F **H'FFF3** **TMRX**



TISR—Timer Input Select Register **H'FFF5** **TMRY**



0	IVG signal is selected (H8S/2128 Series) External clock/reset input is disabled (H8S/2124 Series)
1	VSYNC1/TMIY (TMCIY/TMRIY) is selected

TCONRI—Timer Connection Register I **H'FFFC** **Timer Connection**

Bit	7	6	5	4	3	2	1	0
	SIMOD1	SIMOD0	SCONE	ICST	HFINV	VFINV	HIINV	VIINV
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Input synchronization signal inversion

0	The VSYNCI pin state is used directly as the VSYNCI input
1	The VSYNCI pin state is inverted before use as the VSYNCI input

Input synchronization signal inversion

0	The HSYNCI and CSYNCI pin states are used directly as the HSYNCI and CSYNCI inputs
1	The HSYNCI and CSYNCI pin states are inverted before use as the HSYNCI and CSYNCI inputs

Input synchronization signal inversion

0	The VFBACKI pin state is used directly as the VFBACKI input
1	The VFBACKI pin state is inverted before use as the VFBACKI input

Input synchronization signal inversion

0	The HFBACKI pin state is used directly as the HFBACKI input
1	The HFBACKI pin state is inverted before use as the HFBACKI input

Input capture start bit

0	The TICRR and TICRF input capture functions are stopped [Clearing condition] When a rising edge followed by a falling edge is detected on TMRIX
1	The TICRR and TICRF input capture functions are operating (Waiting for detection of a rising edge followed by a falling edge on TMRIX) [Setting condition] When 1 is written in ICST after reading ICST = 0

Synchronization signal connection enable

SCONE	Mode	FTIA	FTIB	FTIC	FTID	TMCI1	TMRI1
0	Normal connection	FTIA input	FTIB input	FTIC input	FTID input	TMCI1 input	TMRI1 input
1	Synchronization signal connec-tion mode	IVI signal	TMO1 signal	VFBACKI input	IHI signal	IHI signal	IVI inverse signal

Input synchronization mode select 1 and 0

SIMOD1	SIMOD0	Mode	IHI signal	IVI signal
0	0	No signal	HFBACKI input	VFBACKI input
	1	S-on-G mode	CSYNCI input	PDC input
1	0	Composite mode	HSYNCI input	PDC input
	1	Separate mode	HSYNCI input	VSYNCI input

TCONRO—Timer Connection Register O **H'FFFD** **Timer Connection**

Bit	7	6	5	4	3	2	1	0
	HOE	VOE	CLOE	CBOE	HOINV	VOINV	CLOINV	CBOINV
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output synchronization signal inversion

0	The CBLANK signal is used directly as the CBLANK output
1	The CBLANK signal is inverted before use as the CBLANK output

Output synchronization signal inversion

0	The CLO signal (CL1, CL2, CL3, or CL4 signal) is used directly as the CLAMPO output
1	The CLO signal (CL1, CL2, CL3, or CL4 signal) is inverted before use as the CLAMPO output

Output synchronization signal inversion

0	The IVO signal is used directly as the VSYNCO output
1	The IVO signal is inverted before use as the VSYNCO output

Output synchronization signal inversion

0	The IHO signal is used directly as the HSYNCO output
1	The IHO signal is inverted before use as the HSYNCO output

Output enable

0	The P27/A15/PW15/CBLANK pin functions as the P27/A15/PW15 pin
1	In mode 1 (expanded mode with on-chip ROM disabled): The P27/A15/PW15/CBLANK pin functions as the A15 pin In modes 2 and 3 (expanded modes with on-chip ROM enabled): The P27/A15/PW15/CBLANK pin functions as the CBLANK pin

Output enable

0	The P64/FTIC/CIN4/CLAMPO pin functions as the P64/FTIC/CIN4 pin
1	The P64/FTIC/CIN4/CLAMPO pin functions as the CLAMPO pin

Output enable

0	The P61/FTOA/CIN1/VSYNCO pin functions as the P61/FTOA/CIN1 pin
1	The P61/FTOA/CIN1/VSYNCO pin functions as the VSYNCO pin

Output enable

0	The P67/TMO1/TMOX/CIN7/HSYNCO pin functions as the P67/TMO1/TMOX/CIN7 pin
1	The P67/TMO1/TMOX/CIN7/HSYNCO pin functions as the HSYNCO pin

TCONRS—Timer Connection Register S **H'FFFE** **Timer Connection**

Bit	7	6	5	4	3	2	1	0
	TMRX/Y	ISGENE	HOMOD1	HOMOD0	VOMOD1	VOMOD0	CLMOD1	CLMOD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clamp waveform mode select 1 and 0

ISGENE	CLMOD1	CLMOD0	Description
0	0	0	The CL1 signal is selected
		1	The CL2 signal is selected
	1	0	The CL3 signal is selected
		1	
1	0	0	The CL4 signal is selected
		1	
	1	0	
		1	

Vertical synchronization output mode select 1 and 0

ISGENE	VOMOD1	VOMOD0	Description
0	0	0	The IVI signal (without fall modification or IHI synchronization) is selected
		1	The IVI signal (without fall modification, with IHI synchronization) is selected
	1	0	The IVI signal (with fall modification, without IHI synchronization) is selected
		1	The IVI signal (with fall modification and IHI synchronization) is selected
1	0	0	The IVG signal is selected
		1	
	1	0	
		1	

Horizontal synchronization output mode select 1 and 0

ISGENE	HOMOD1	HOMOD0	Description
0	0	0	The IHI signal (without 2fH modification) is selected
		1	The IHI signal (with 2fH modification) is selected
	1	0	The CLI signal is selected
		1	
1	0	0	The IHG signal is selected
		1	
	1	0	
		1	

Internal synchronization signal select

8-bit timer access select

0	The TMRX registers are accessed at addresses H'FFF0 to H'FFF5
1	The TMRY registers are accessed at addresses H'FFF0 to H'FFF5

SEDGR—Edge Sense Register **H'FFFF** **Timer Connection**

Bit	7	6	5	4	3	2	1	0
	VEDG	HEDG	CEDG	HFEDG	VFEDG	PREQF	IHI	IVI
Initial value	0	0	0	0	0	0	—*2	—*2
Read/Write	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R

IVI signal level

0	The IVI signal is low
1	The IVI signal is high

IHI signal level

0	The IHI signal is low
1	The IHI signal is high

Pre-equalization flag

0	[Clearing condition] When 0 is written in PREQF after reading PREQF = 1
1	[Setting condition] When an IHI signal 2fH modification condition is detected

VFBACKI edge

0	[Clearing condition] When 0 is written in VFEDG after reading VFEDG = 1
1	[Setting condition] When a rising edge is detected on the VFBACKI pin

HFBACKI edge

0	[Clearing condition] When 0 is written in HFEDG after reading HFEDG = 1
1	[Setting condition] When a rising edge is detected on the HFBACKI pin

CSYNCI edge

0	[Clearing condition] When 0 is written in CEDG after reading CEDG = 1
1	[Setting condition] When a rising edge is detected on the CSYNCI pin

HSYNCI edge

0	[Clearing condition] When 0 is written in HEDG after reading HEDG = 1
1	[Setting condition] When a rising edge is detected on the HSYNCI pin

VSYNCI edge

0	[Clearing condition] When 0 is written in VEDG after reading VEDG = 1
1	[Setting condition] When a rising edge is detected on the VSYNCI pin

Notes: 1. Only 0 can be written, to clear the flags.
2. The initial value is undefined since it depends on the pin states.

Appendix C I/O Port Block Diagrams

C.1 Port 1 Block Diagram

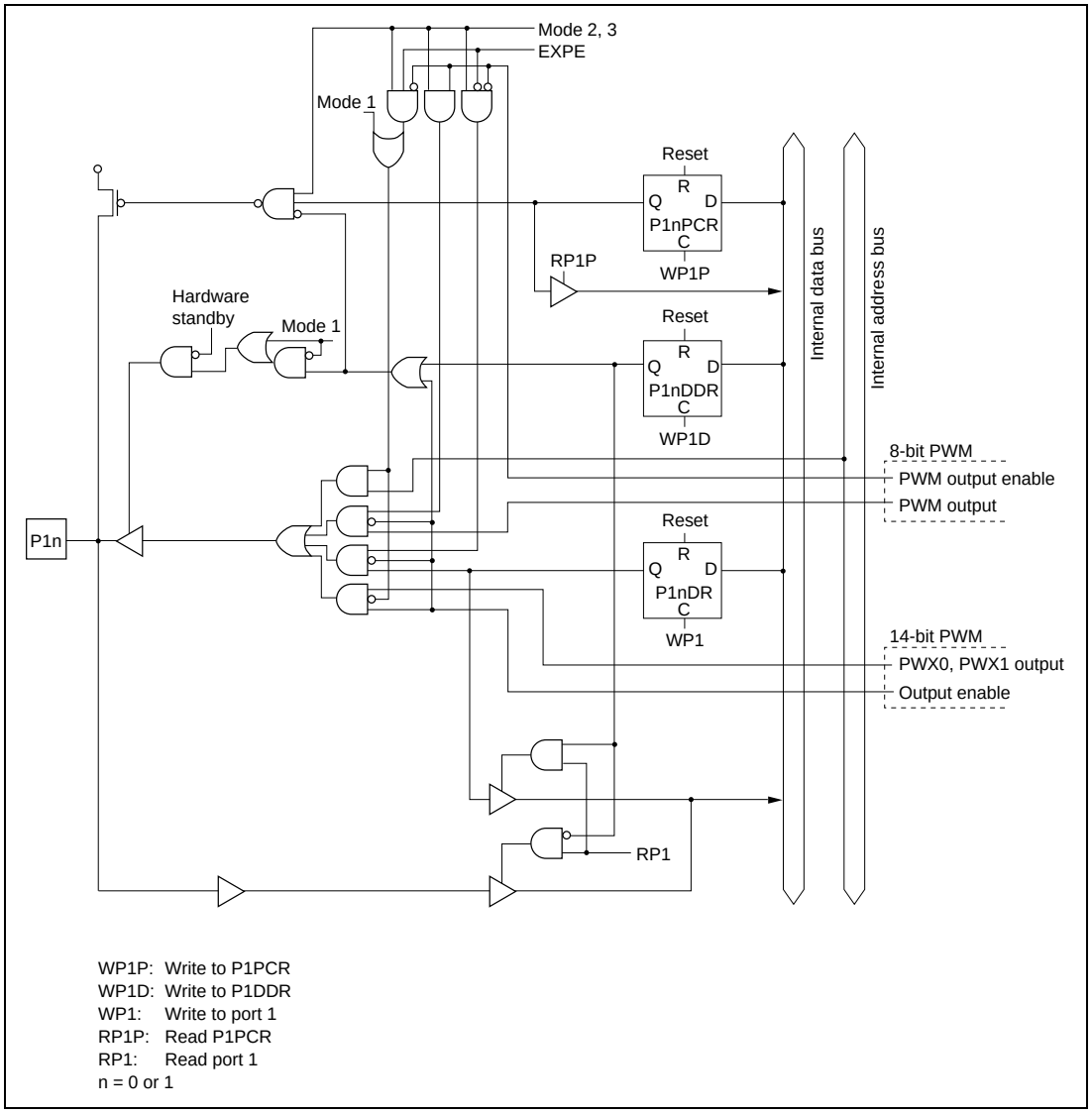


Figure C.1 Port 1 Block Diagram (Pins P10 and P11)

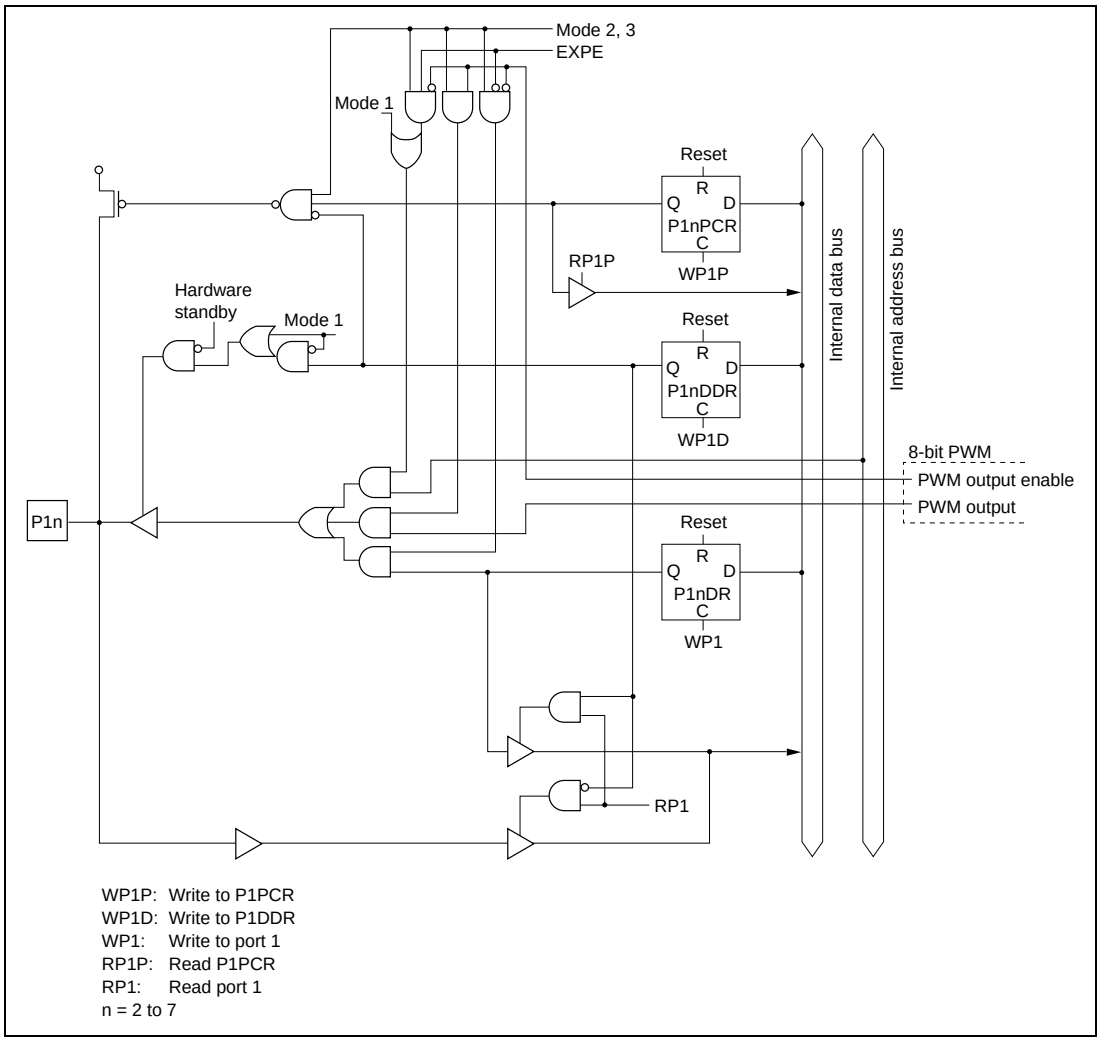


Figure C.2 Port 1 Block Diagram (Pins P12 to P17)

C.2 Port 2 Block Diagrams

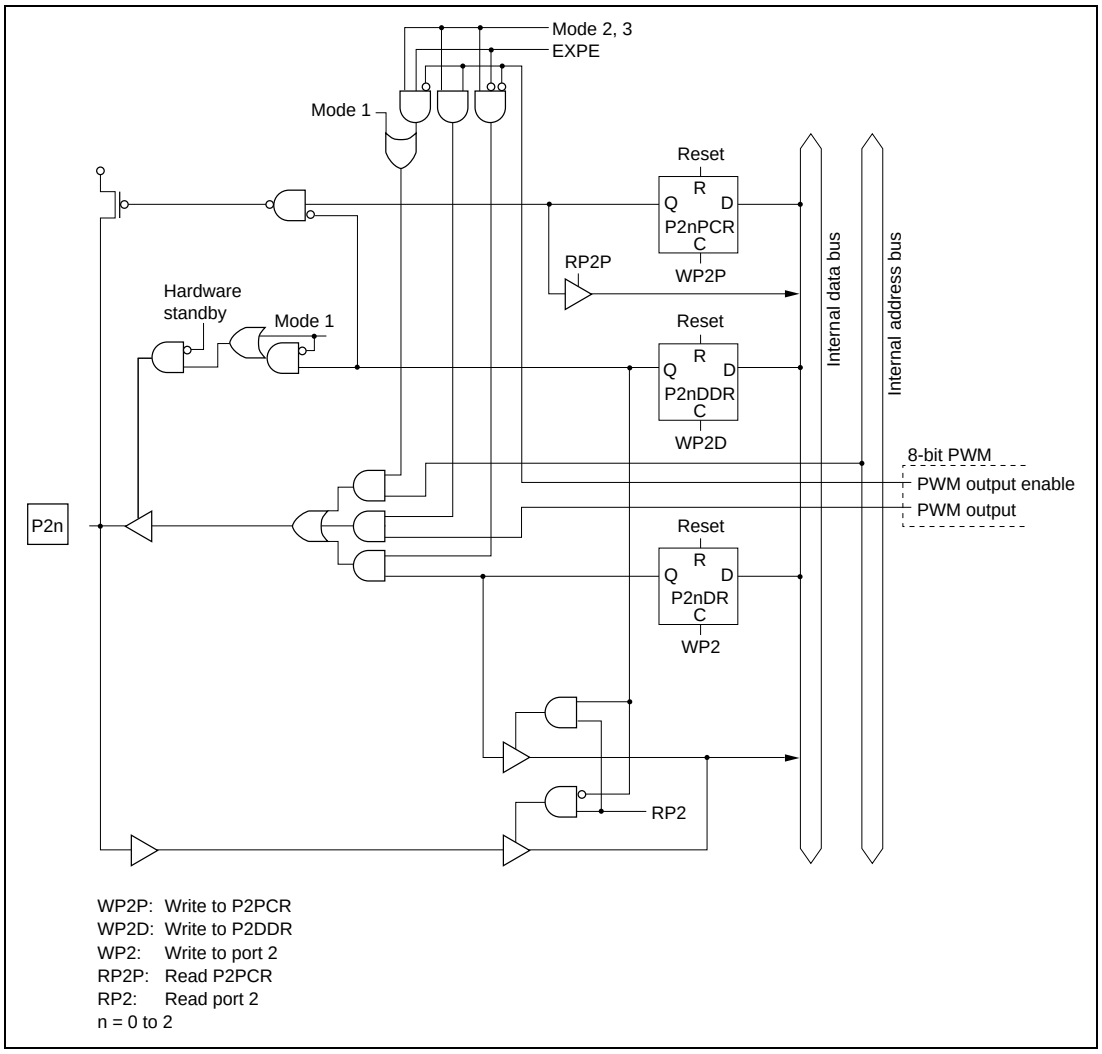


Figure C.3 Port 2 Block Diagram (Pins P20 to P22)

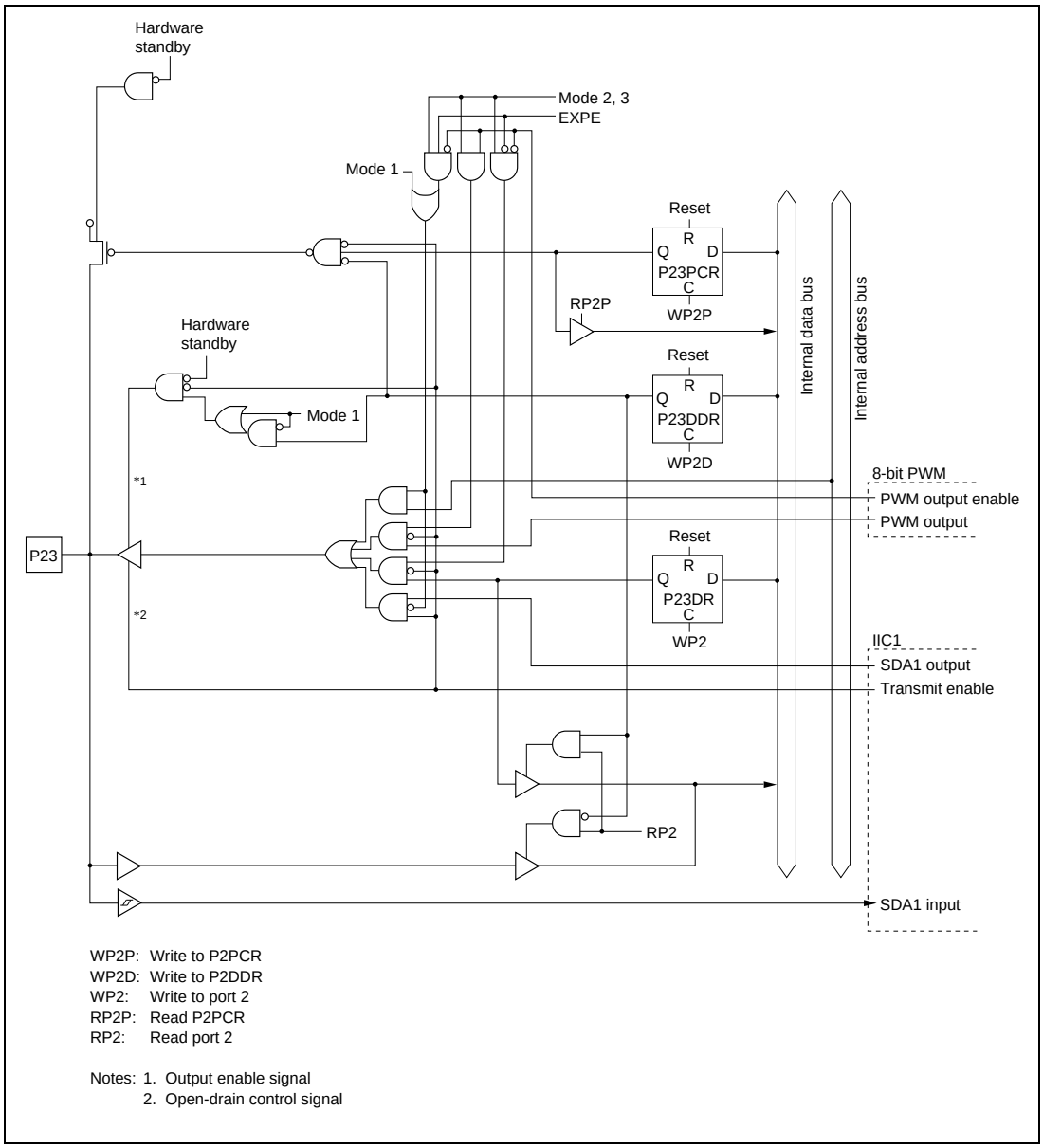


Figure C.4 Port 2 Block Diagram (Pin 23)

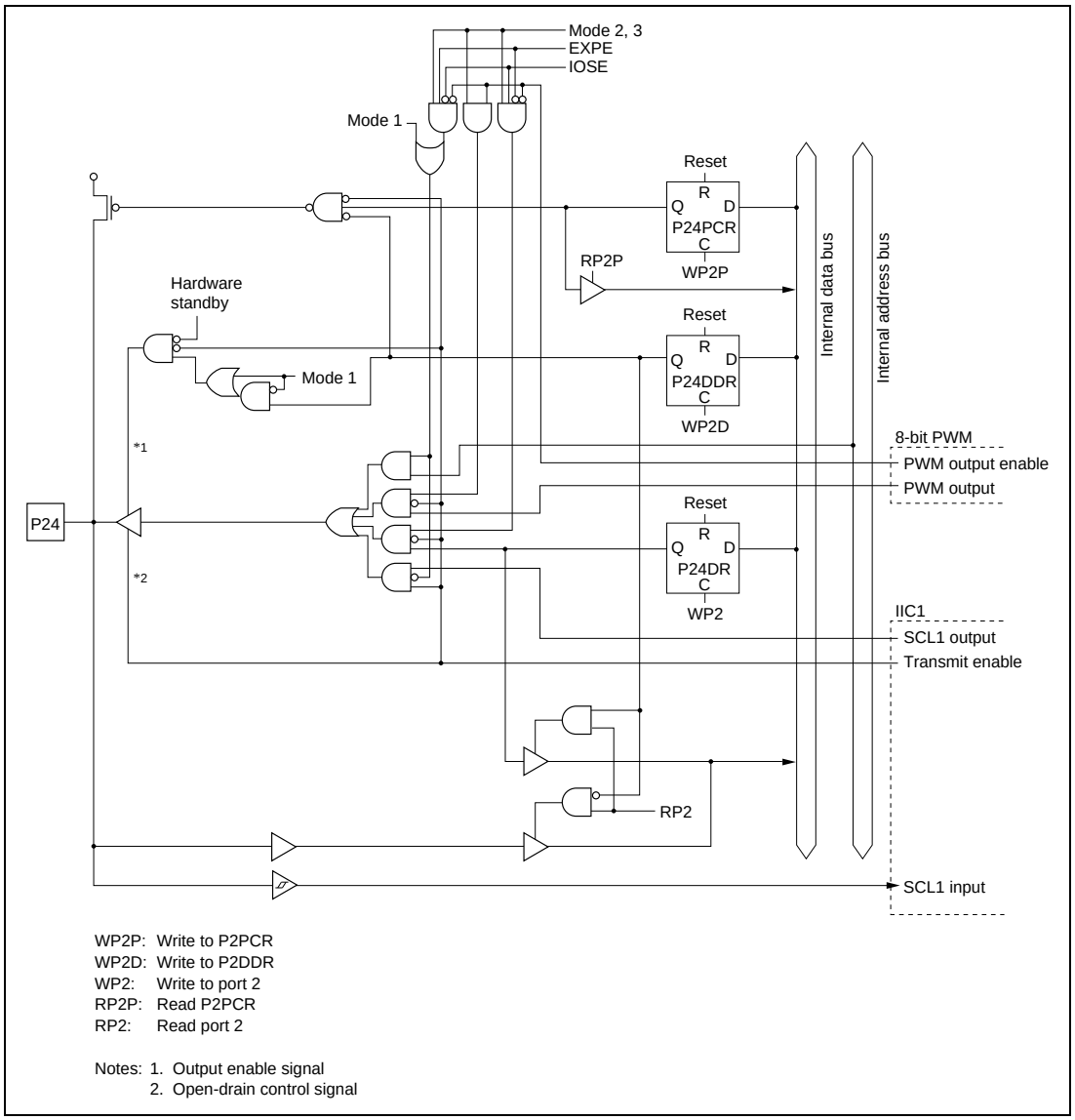


Figure C.5 Port 2 Block Diagram (Pin 24)

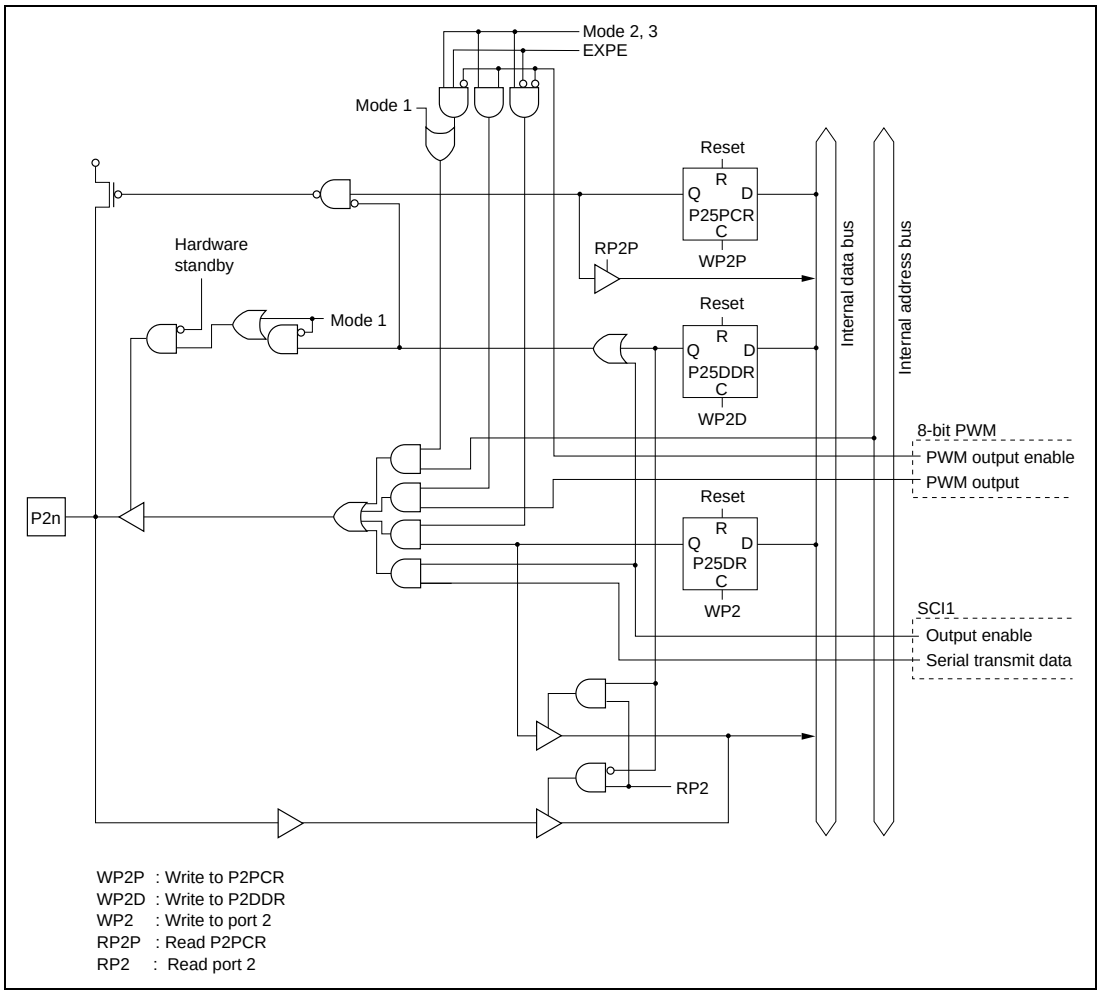


Figure C.6 Port 2 Block Diagram (Pin P25)

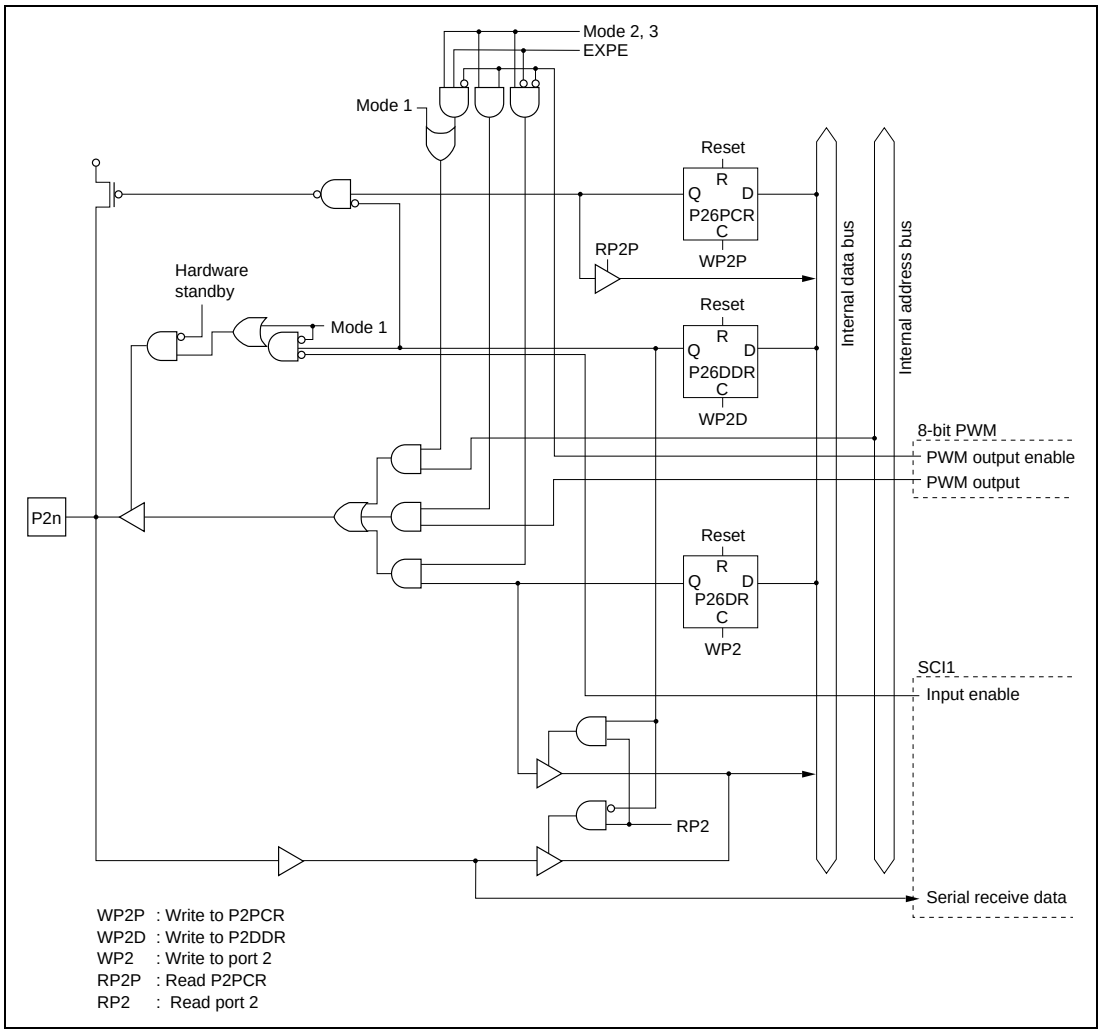


Figure C.7 Port 2 Block Diagram (Pin P26)

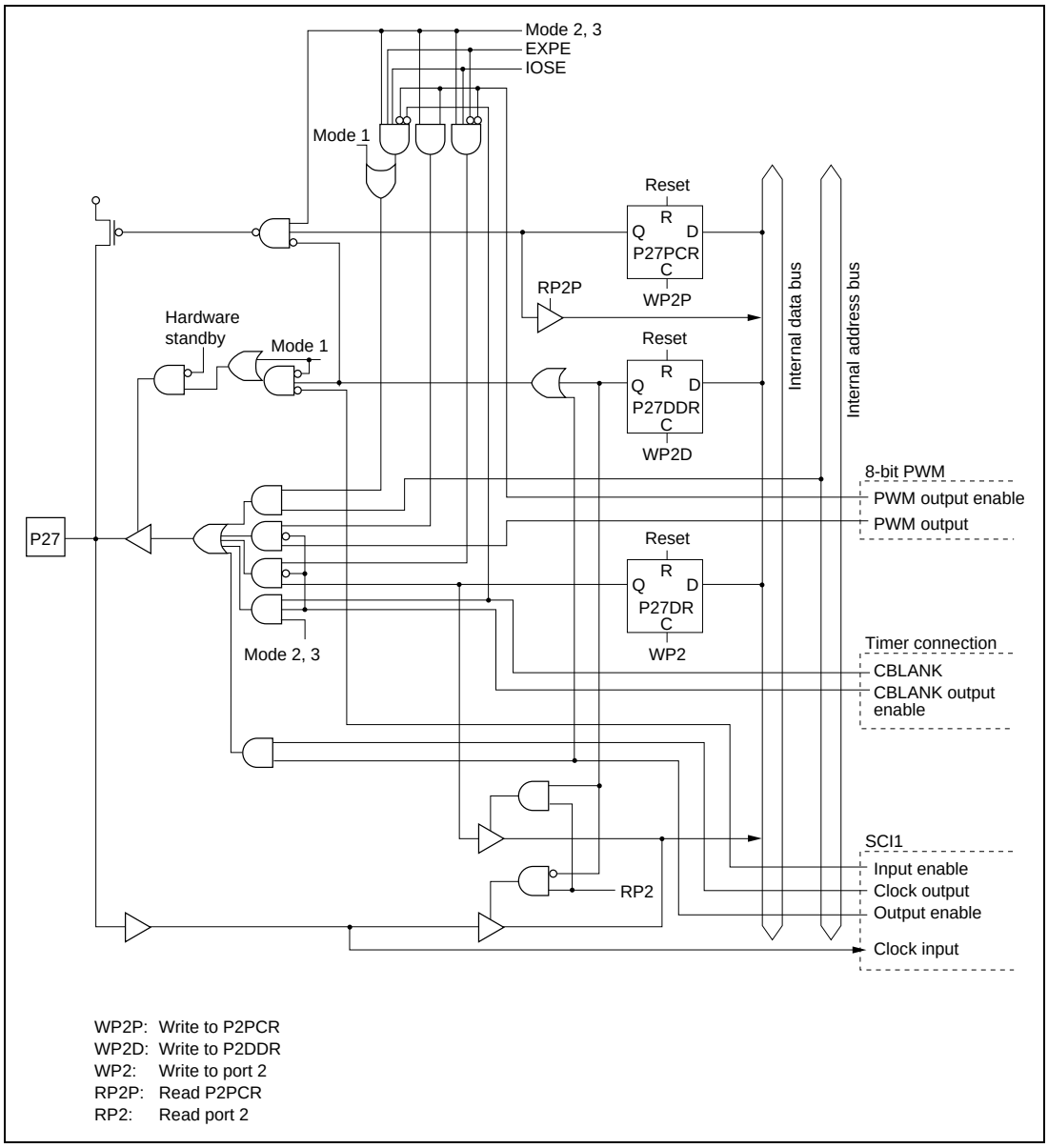


Figure C.8 Port 2 Block Diagram (Pin P27)

C.3 Port 3 Block Diagram

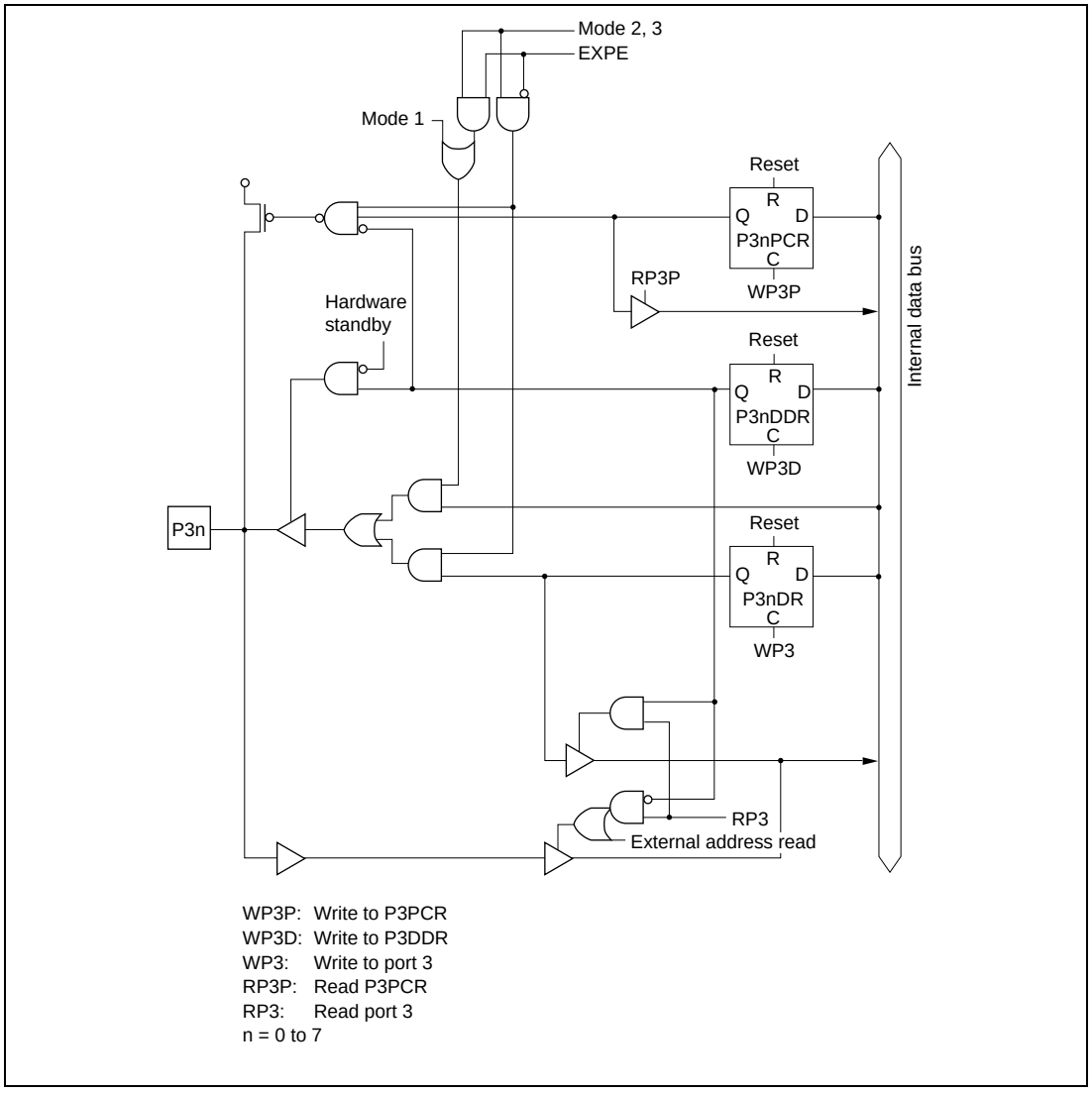


Figure C.9 Port 3 Block Diagram

C.4 Port 4 Block Diagrams

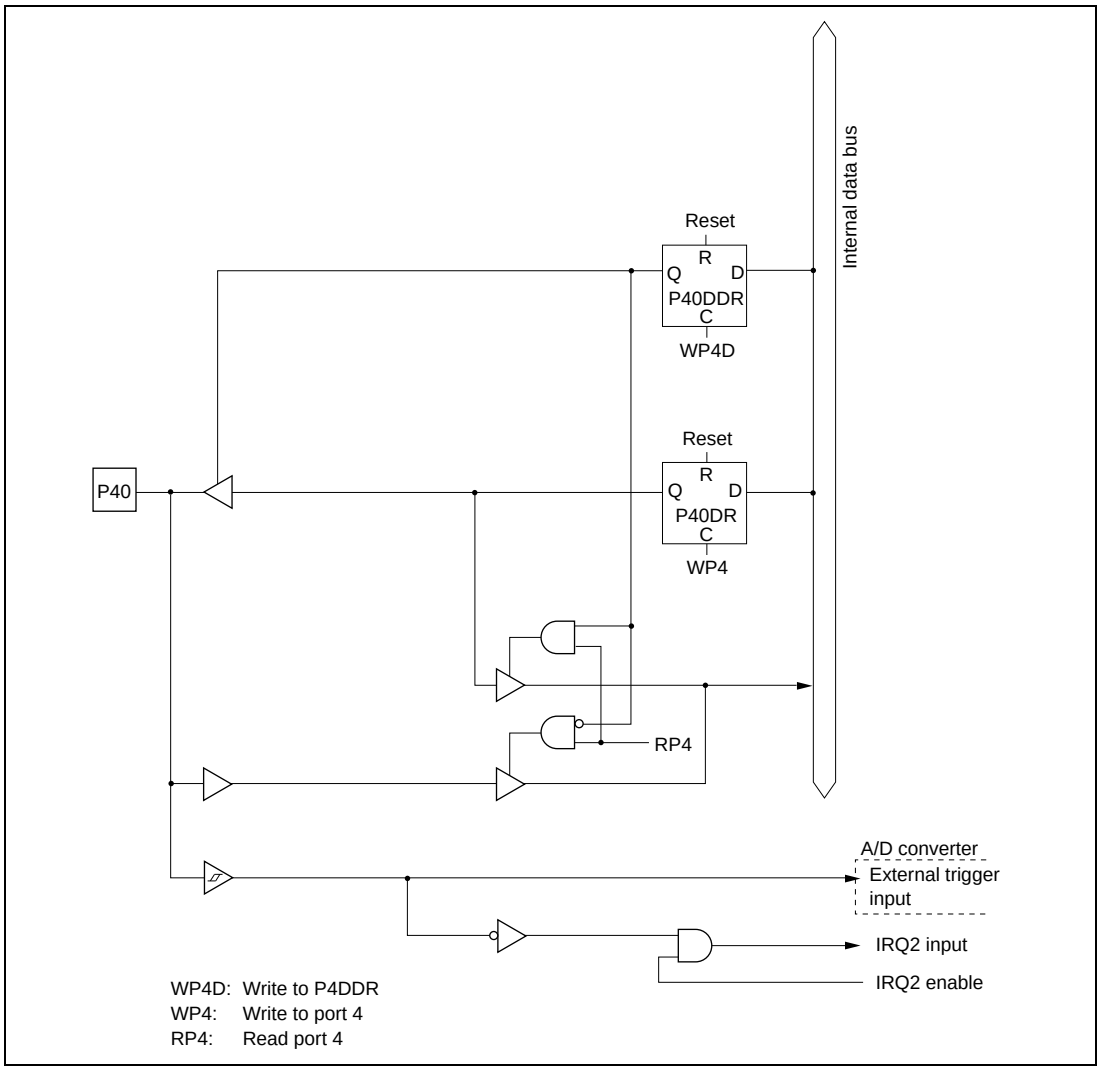


Figure C.10 Port 4 Block Diagram (Pin P40)

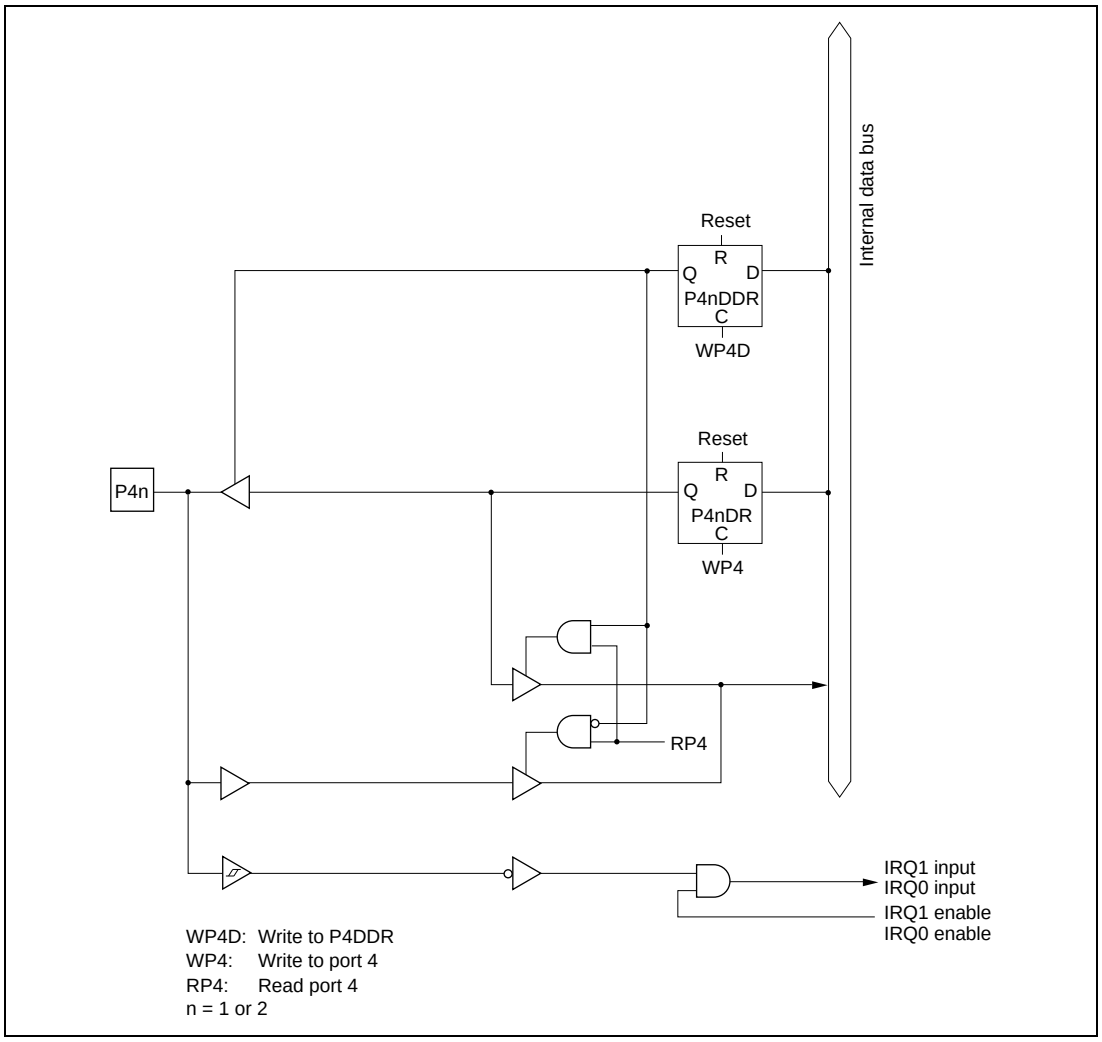


Figure C.11 Port 4 Block Diagram (Pins P41, P42)

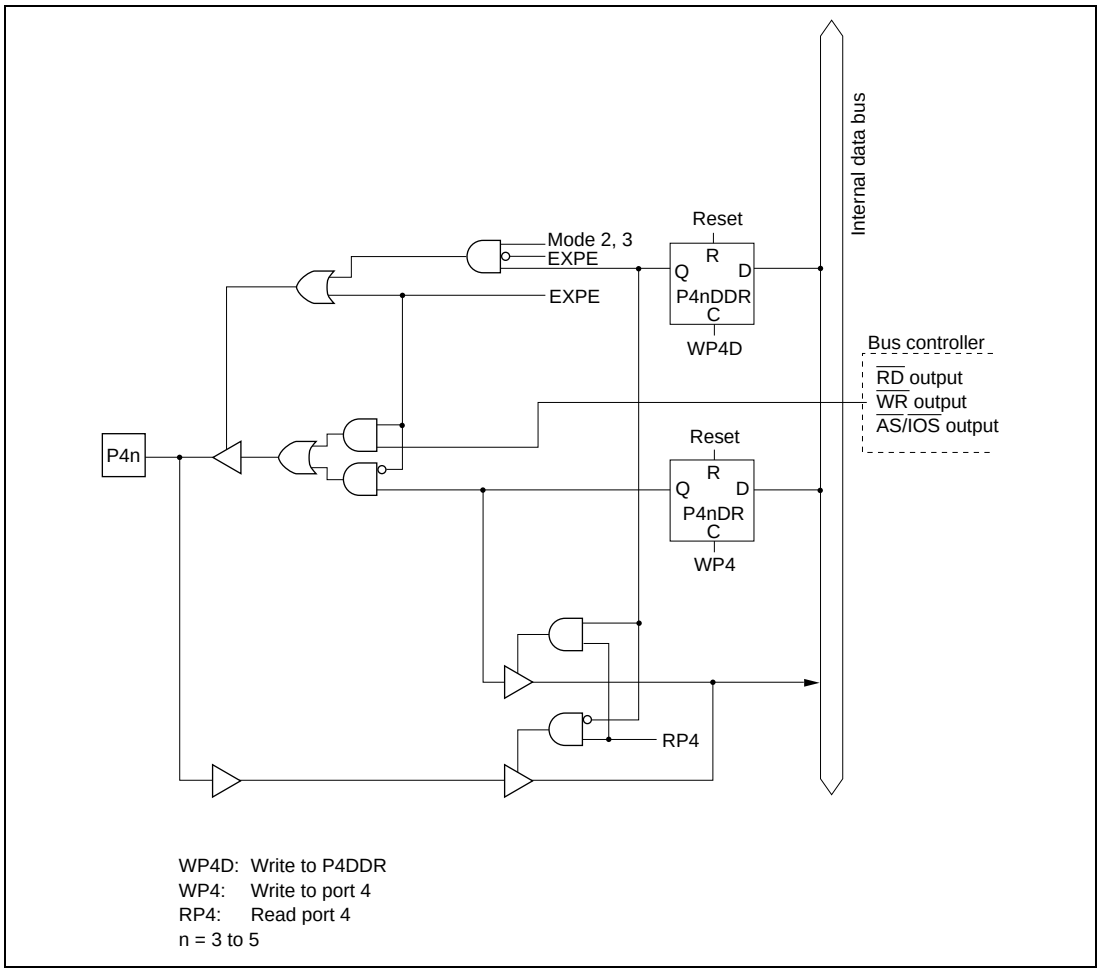


Figure C.12 Port 4 Block Diagram (Pins P43 to P45)

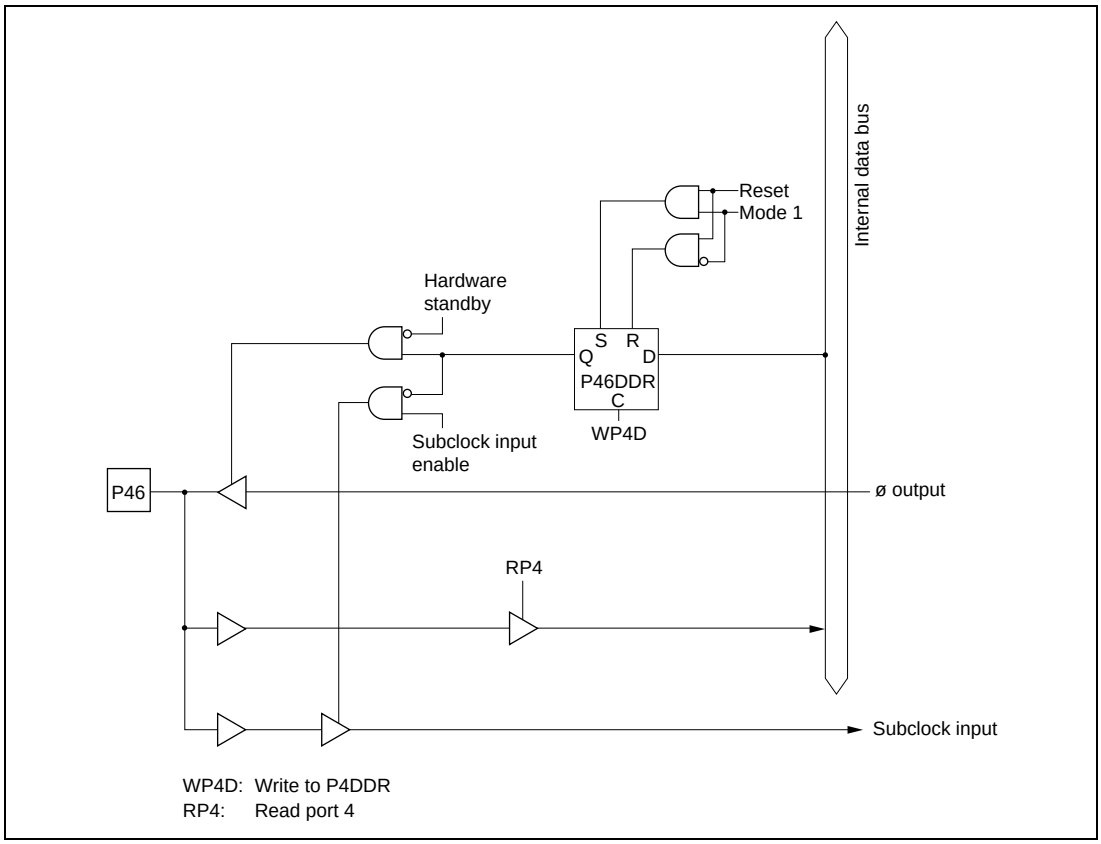


Figure C.13 Port 4 Block Diagram (Pin P46)

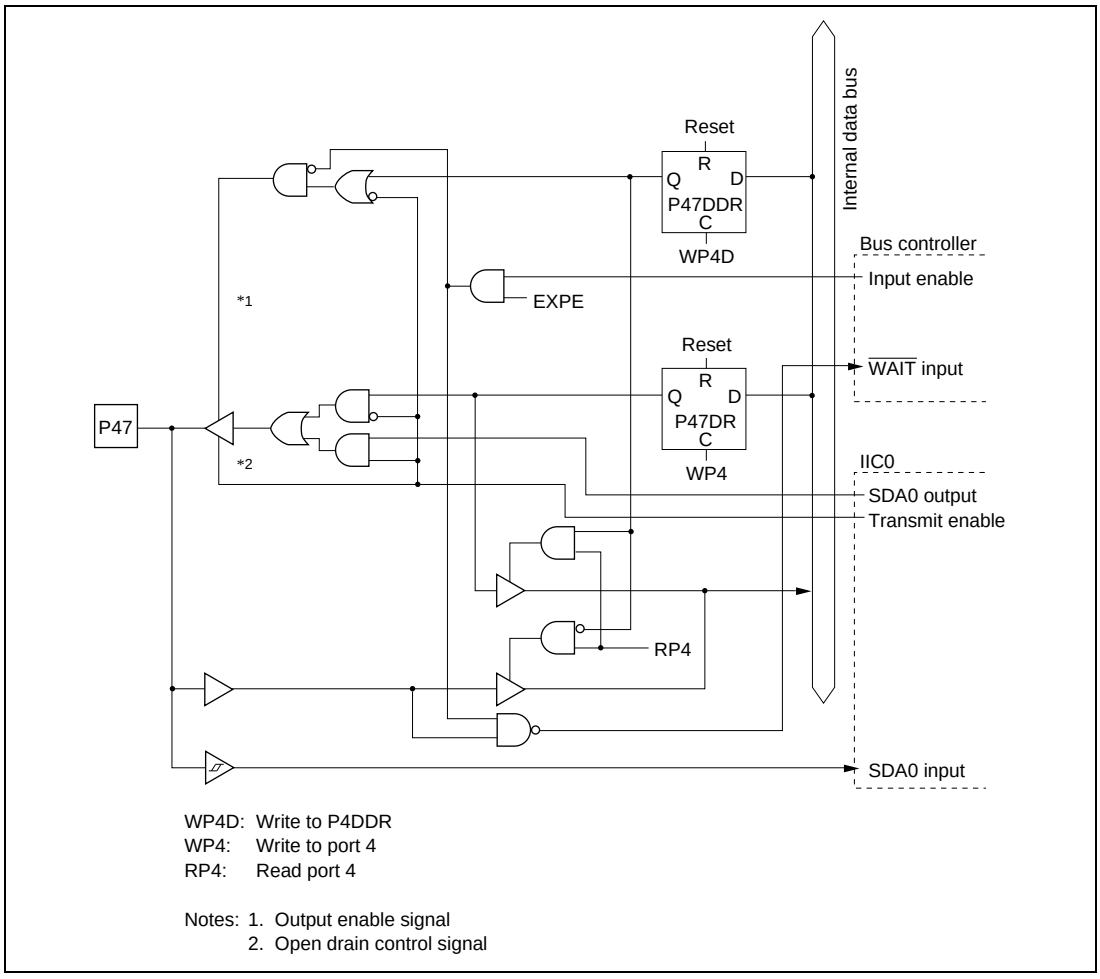


Figure C.14 Port 4 Block Diagram (Pin P47)

C.5 Port 5 Block Diagrams

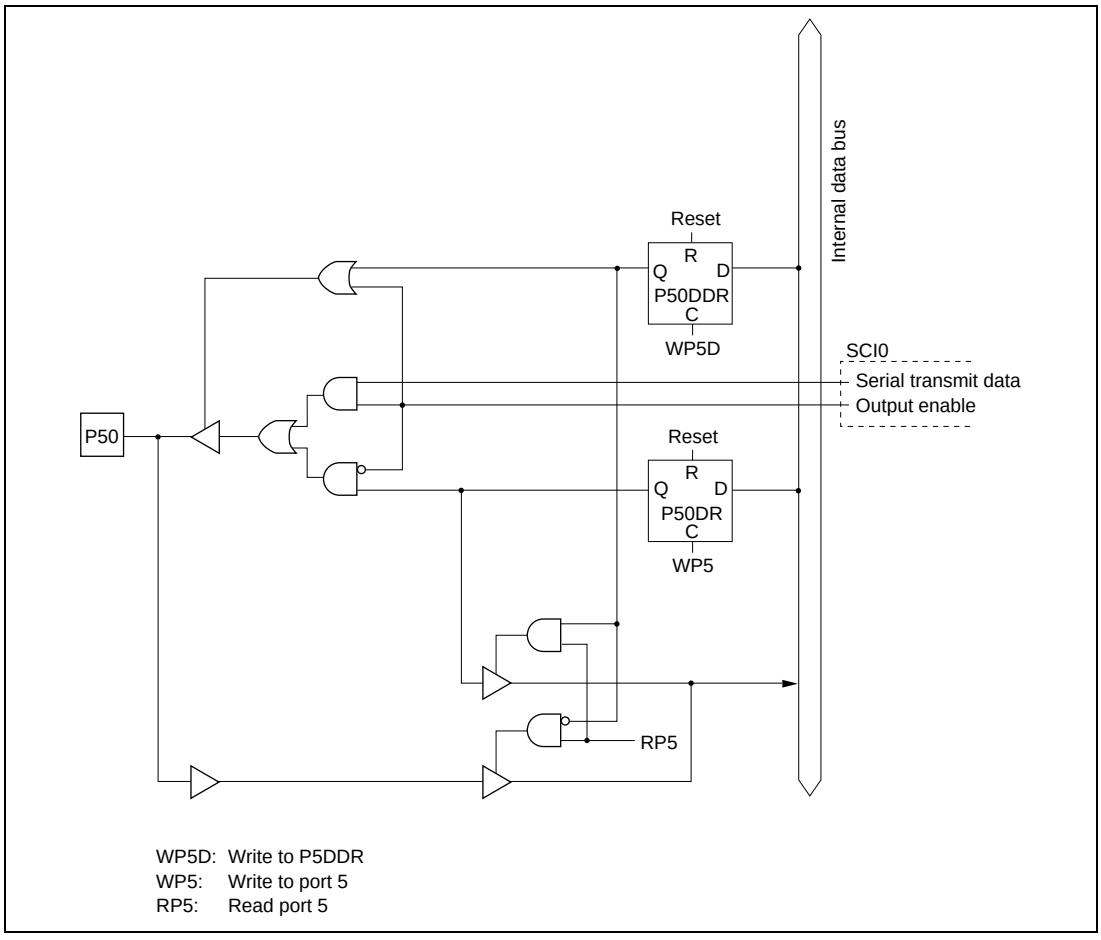


Figure C.15 Port 5 Block Diagram (Pin P50)

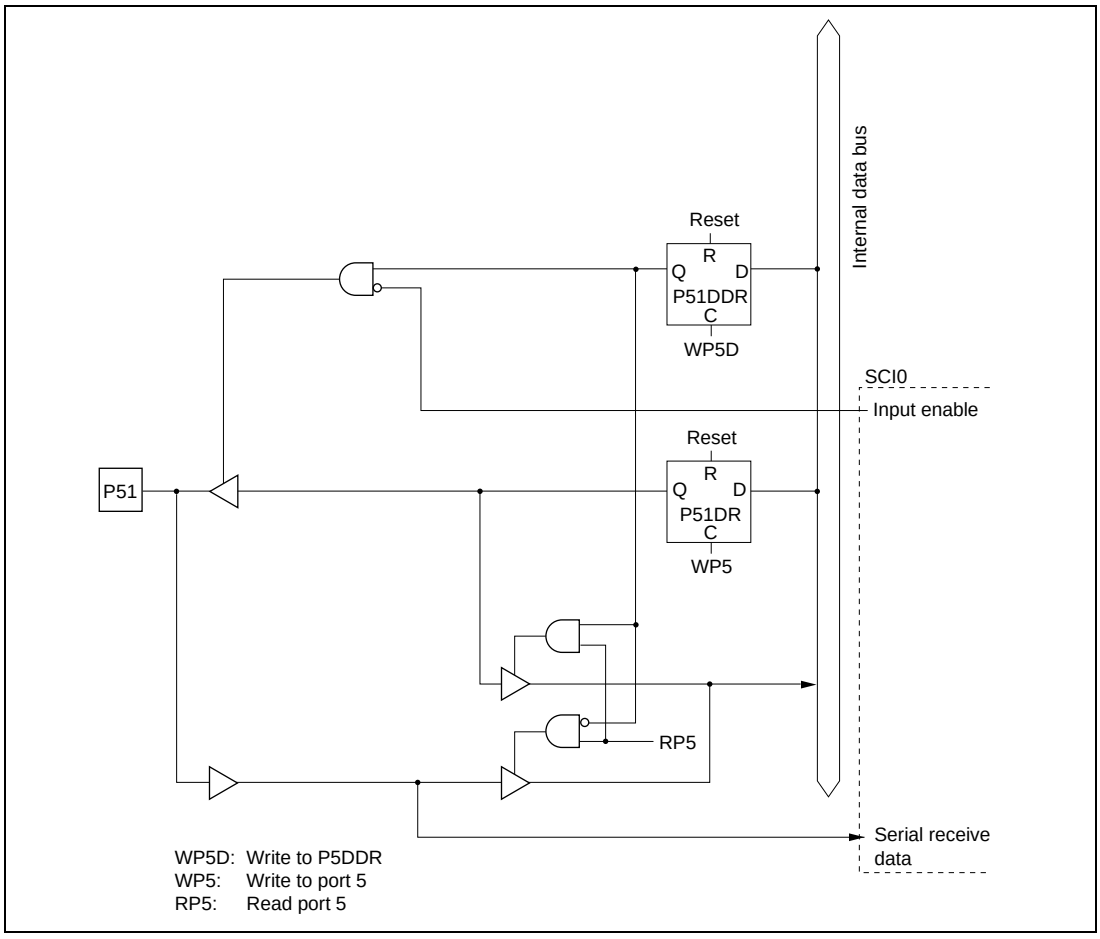


Figure C.16 Port 5 Block Diagram (Pin P51)

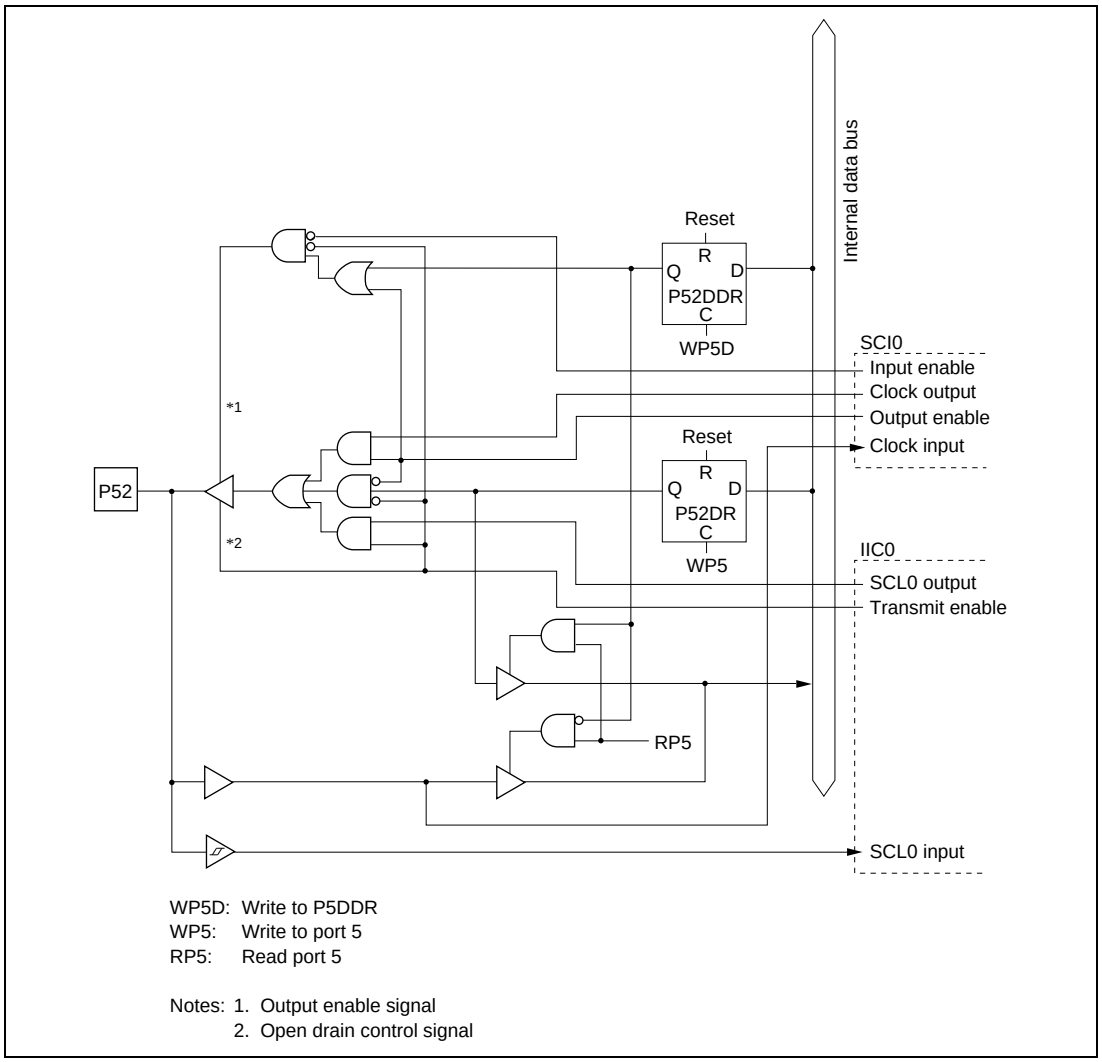


Figure C.17 Port 5 Block Diagram (Pin P52)

C.6 Port 6 Block Diagrams

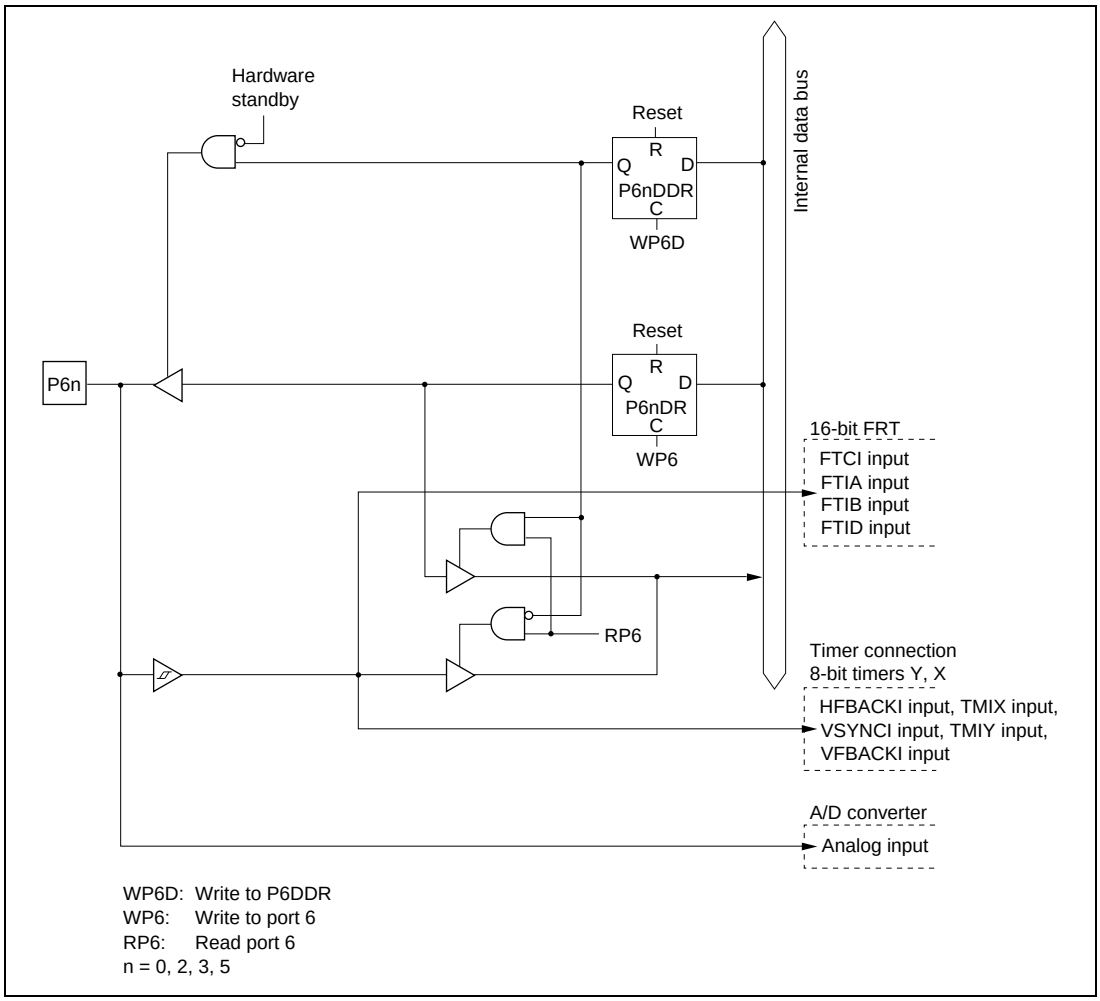


Figure C.18 Port 6 Block Diagram (Pins P60, P62, P63, P65)

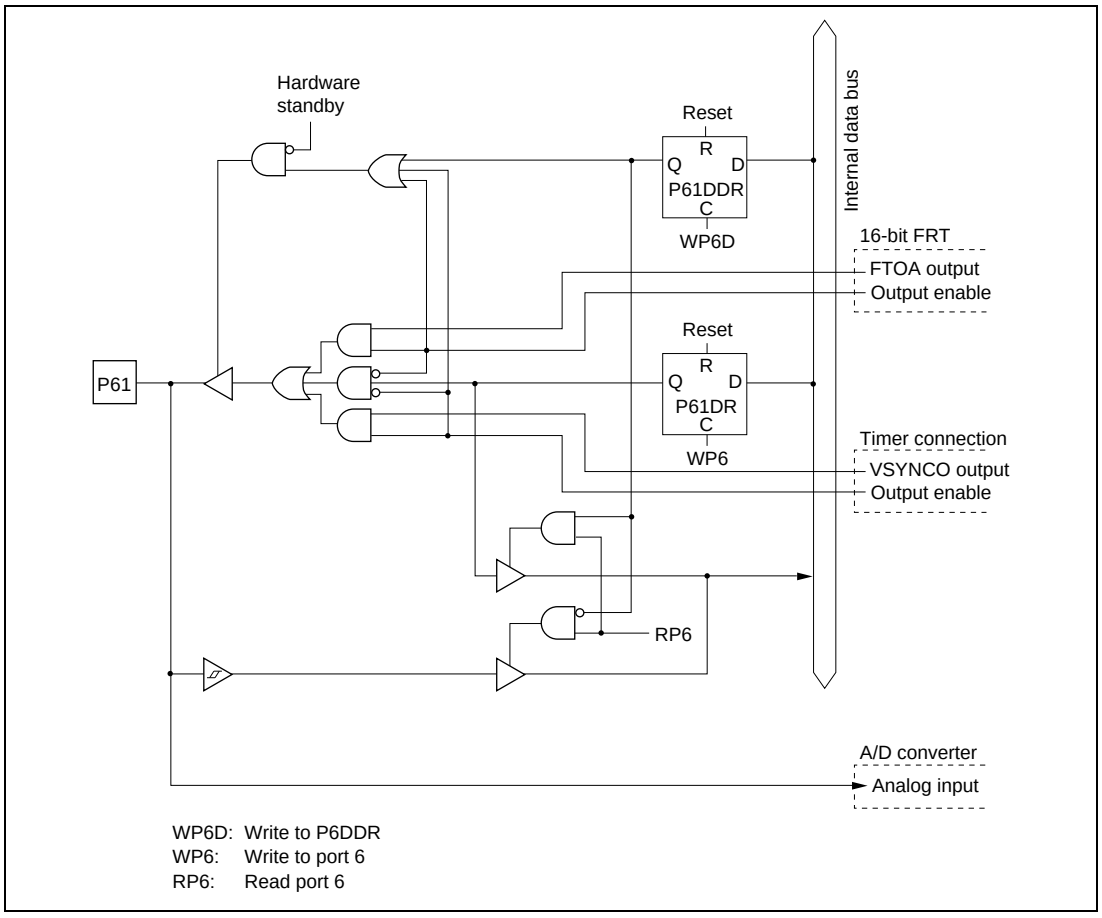


Figure C.19 Port 6 Block Diagram (Pin P61)

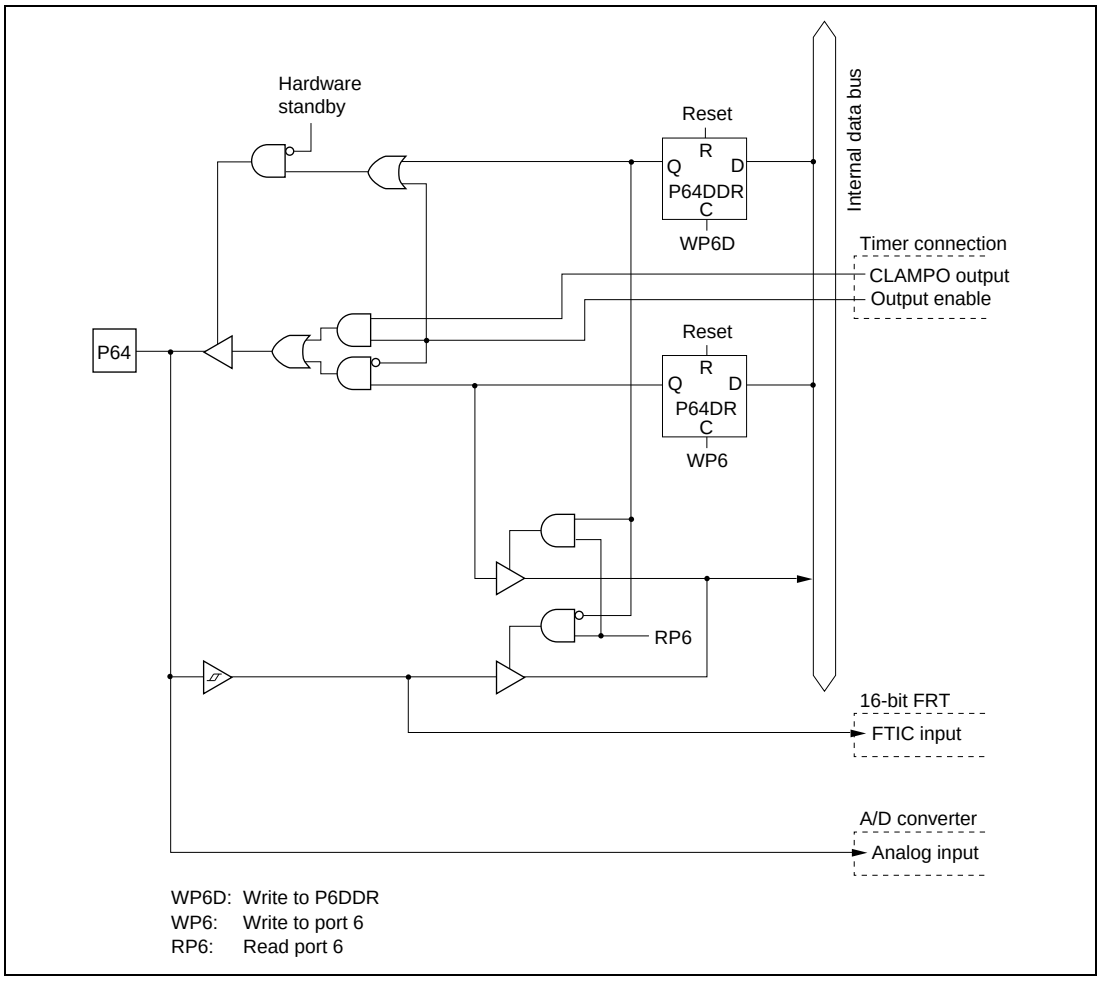


Figure C.20 Port 6 Block Diagram (Pin P64)

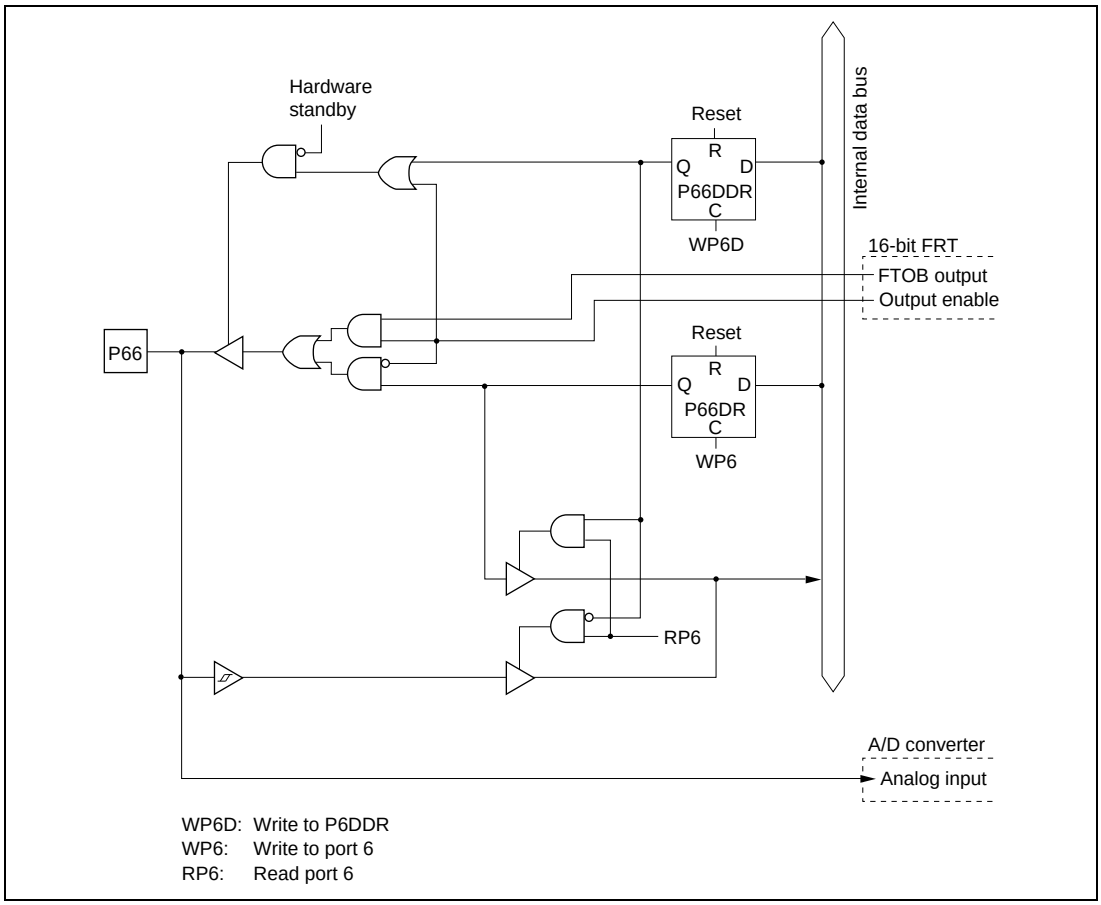


Figure C.21 Port 6 Block Diagram (Pin P66)

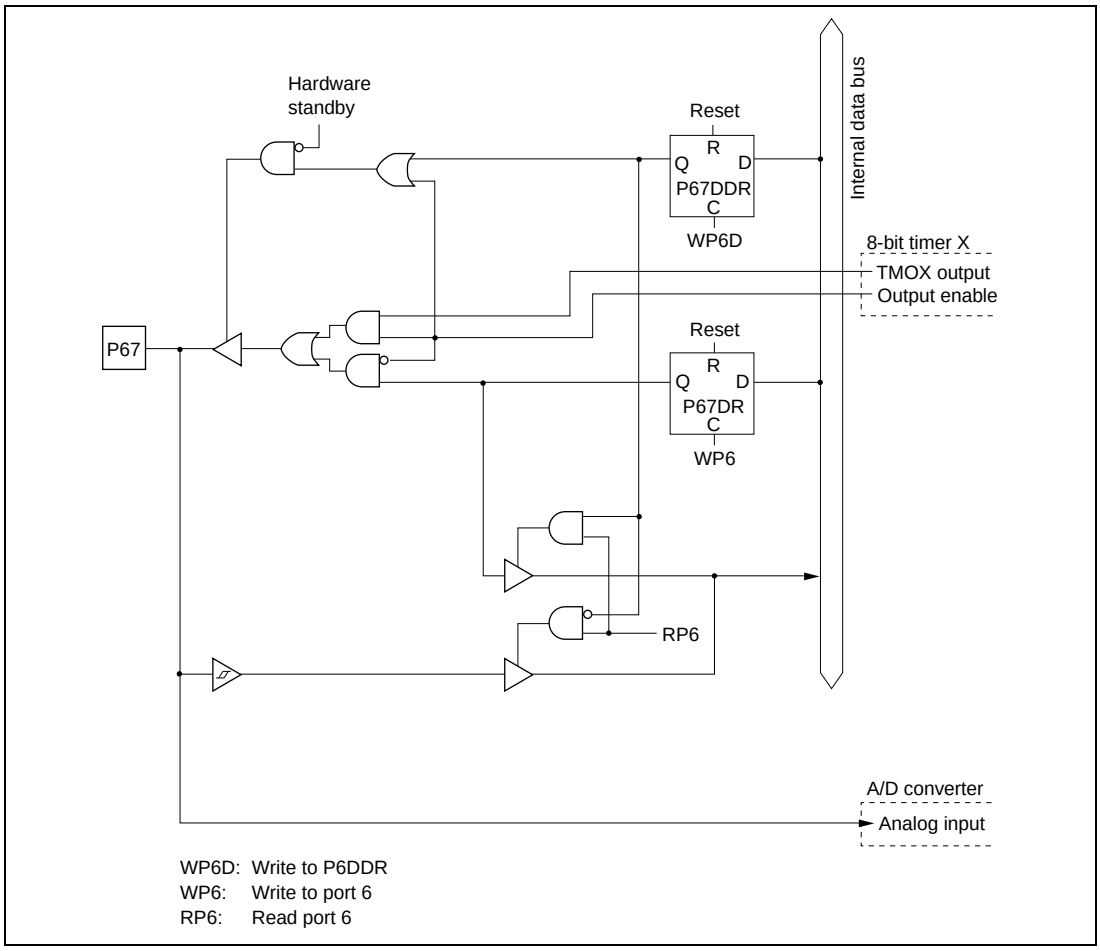


Figure C.22 Port 6 Block Diagram (Pin P67)

C.7 Port 7 Block Diagrams

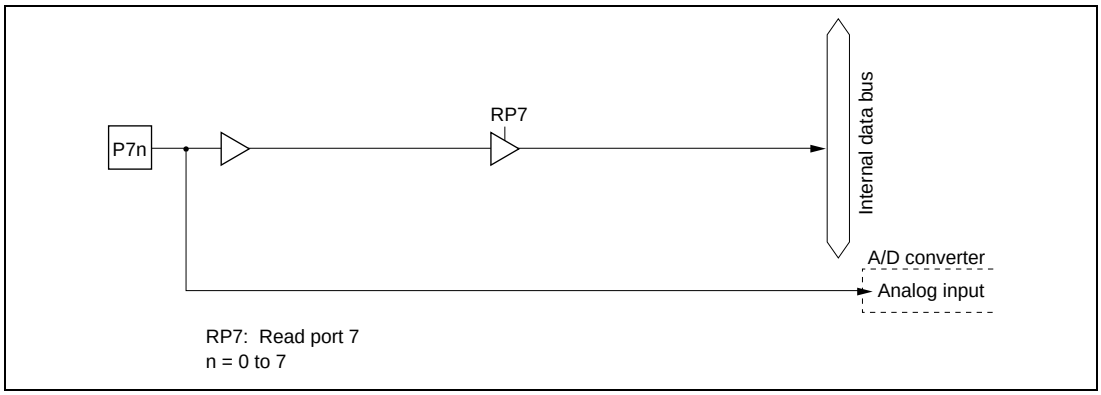


Figure C.23 Port 7 Block Diagram (Pins P70 to P77)

Appendix D Pin States

D.1 Port States in Each Processing State

Table D.1 I/O Port States in Each Processing State

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Watch Mode	Sleep Mode	Sub-sleep Mode	Subactive Mode	Program Execution State
Port 1 A7 to A0	1	L	T	keep*	keep*	keep*	keep*	A7 to A0	A7 to A0
	2, 3 (EXPE = 1)	T						Address output/ input port	Address output/ input port
	2, 3 (EXPE = 0)							I/O port	I/O port
Port 2 A15 to A8	1	L	T	keep*	keep*	keep*	keep*	A15 to A8	A15 to A8
	2, 3 (EXPE = 1)	T						Address output/ input port	Address output/ input port
	2, 3 (EXPE = 0)							I/O port	I/O port
Port 3 D7 to D0	1	T	T	T	T	T	T	D7 to D0	D7 to D0
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)			keep	keep	keep	keep	I/O port	I/O port
Port 47 $\overline{WAIT}$	1	T	T	T/keep	T/keep	T/keep	T/keep	$\overline{WAIT}$/ I/O port	$\overline{WAIT}$/ I/O port
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)			keep	keep	keep	keep	I/O port	I/O port
Port 46 ø EXCL	1	Clock output	T	[DDR = 1] H [DDR = 0] T	EXCL input	[DDR = 1] clock output	EXCL input	EXCL input	Clock output/ EXCL input/ input port
	2, 3 (EXPE = 1)	T				[DDR = 0] T			
	2, 3 (EXPE = 0)								
Port 45 to 43 $\overline{AS}$, $\overline{WR}$, $\overline{RD}$	1	H	T	H	H	H	H	$\overline{AS}$, $\overline{WR}$, $\overline{RD}$	$\overline{AS}$, $\overline{WR}$, $\overline{RD}$
	2, 3 (EXPE = 1)	T							
	2, 3 (EXPE = 0)			keep	keep	keep	keep	I/O port	I/O port
Port 42 to 40	1	T	T	keep	keep	keep	keep	I/O port	I/O port
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)								
Port 5	1	T	T	keep	keep	keep	keep	I/O port	I/O port
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)								

Table D.1 I/O Port States in Each Processing State (cont)

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Watch Mode	Sleep Mode	Sub-sleep Mode	Subactive Mode	Program Execution State
Port 6	1	T	T	keep	keep	keep	keep	I/O port	I/O port
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)								
Port 7	1	T	T	T	T	T	T	Input port	Input port
	2, 3 (EXPE = 1)								
	2, 3 (EXPE = 0)								

Legend:

H: High

L: Low

T: High-impedance state

keep: Input ports are in the high-impedance state (when DDR = 0 and PCR = 1, MOS input pull-ups remain on).

Output ports maintain their previous state.

Depending on the pins, the on-chip supporting modules may be initialized and the I/O port function determined by DDR and DR used.

DDR: Data direction register

Note: * In the case of address output, the last address accessed is retained.

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

E.1 Timing of Transition to Hardware Standby Mode

(1) To retain RAM contents with the RAME bit set to 1 in SYSCR, drive the $\overline{RES}$ signal low 10 system clock cycles before the $\overline{STBY}$ signal goes low, as shown in figure E.1. $\overline{RES}$ must remain low until $\overline{STBY}$ signal goes low (minimum delay from $\overline{STBY}$ low to $\overline{RES}$ high: 0 ns).

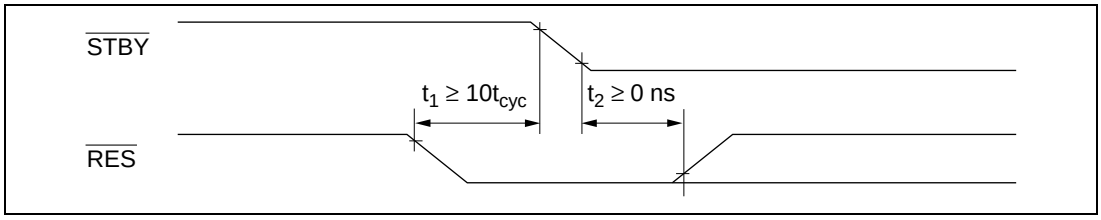


Figure E.1 Timing of Transition to Hardware Standby Mode

(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM contents do not need to be retained, $\overline{RES}$ does not have to be driven low as in (1).

E.2 Timing of Recovery from Hardware Standby Mode

Drive the $\overline{RES}$ signal low at least 100 ns before $\overline{STBY}$ goes high to execute a reset.

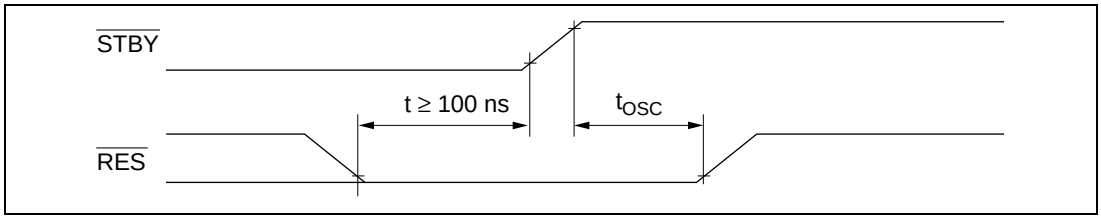


Figure E.2 Timing of Recovery from Hardware Standby Mode

Appendix F Product Code Lineup

Table F.1 H8S/2128 Series and H8S/2124 Series Product Code Lineup — Preliminary —

Product Type				Product Code	Mark Code	Package (Hitachi Package Code)	Notes
H8S/2128 Series	H8S/2128	Mask ROM version	Standard product (5 V version, 4 V version, 3 V version)	HD6432128	HD6432128(***)PS	64-pin shrink DIP (DP-64S)	In planning stage
					HD6432128(***)FA	64-pin QFP (FP-64A)	
					HD6432128(***)TF	80-pin TQFP (TFP-80C)	
			Version with on-chip I^2C bus interface (5 V version, 4 V version, 3 V version)	HD6432128W	HD6432128W(***)PS	64-pin shrink DIP (DP-64S)	
					HD6432128W(***)FA	64-pin QFP (FP-64A)	
					HD6432128W(***)TF	80-pin TQFP (TFP-80C)	
		F-ZTAT version	Standard product (5 V/4 V version)	HD64F2128	HD64F2128PS20	64-pin shrink DIP (DP-64S)	Under development
					HD64F2128FA20	64-pin QFP (FP-64A)	
					HD64F2128TF20	80-pin TQFP (TFP-80C)	
			Low-voltage version (3 V version)	HD64F2128V	HD64F2128VPS10	64-pin shrink DIP (DP-64S)	Under development
					HD64F2128VFA10	64-pin QFP (FP-64A)	
					HD64F2128VTF10	80-pin TQFP (TFP-80C)	
	H8S/2127	Mask ROM version	Standard product (5 V version, 4 V version, 3 V version)	HD6432127R	HD6432127R(***)PS	64-pin shrink DIP (DP-64S)	Under development
					HD6432127R(***)FA	64-pin QFP (FP-64A)	
					HD6432127R(***)TF	80-pin TQFP (TFP-80C)	
			Version with on-chip I^2C bus interface (5 V version, 4 V version, 3 V version)	HD6432127RW	HD6432127RW(***)PS	64-pin shrink DIP (DP-64S)	
					HD6432127RW(***)FA	64-pin QFP (FP-64A)	
					HD6432127RW(***)TF	80-pin TQFP (TFP-80C)	

Table F.1 H8S/2128 Series and H8S/2124 Series Product Code Lineup (cont)

— Preliminary —

Product Type				Product Code	Mark Code	Package (Hitachi Package Code)	Notes
H8S/2128 Series	H8S/2126	Mask ROM version	Standard product (5 V version, 4 V version, 3 V version)	HD6432126R	HD6432126R(***)PS	64-pin shrink DIP (DP-64S)	Under development
					HD6432126R(***)FA	64-pin QFP (FP-64A)	
					HD6432126R(***)TF	80-pin TQFP (TFP-80C)	
			Version with on-chip I^2C bus interface (5 V version, 4 V version, 3 V version)	HD6432126RW	HD6432126RW(***)PS	64-pin shrink DIP (DP-64S)	
					HD6432126RW(***)FA	64-pin QFP (FP-64A)	
					HD6432126RW(***)TF	80-pin TQFP (TFP-80C)	
H8S/2124 Series	H8S/2124	Mask ROM version	Standard product (5 V version, 4 V version, 3 V version)	HD6432124	HD6432124(***)PS	64-pin shrink DIP (DP-64S)	In planning stage
					HD6432124(***)FA	64-pin QFP (FP-64A)	
					HD6432124(***)TF	80-pin TQFP (TFP-80C)	
	H8S/2123	Mask ROM version	Standard product (5 V version, 4 V version, 3 V version)	HD6432123	HD6432123(***)PS	64-pin shrink DIP (DP-64S)	In planning stage
					HD6432123(***)FA	64-pin QFP (FP-64A)	
					HD6432123(***)TF	80-pin TQFP (TFP-80C)	
	H8S/2122	Mask ROM version	Standard product (5 V version, 4 V version, 3 V version)	HD6432122	HD6432122(***)PS	64-pin shrink DIP (DP-64S)	
					HD6432122(***)FA	64-pin QFP (FP-64A)	
					HD6432122(***)TF	80-pin TQFP (TFP-80C)	
	H8S/2120	Mask ROM version	Standard product (5 V version, 4 V version, 3 V version)	HD6432120	HD6432120(***)PS	64-pin shrink DIP (DP-64S)	
					HD6432120(***)FA	64-pin QFP (FP-64A)	
					HD6432120(***)TF	80-pin TQFP (TFP-80C)	

Note: (***) is the ROM code.

The F-ZTAT version of the H8S/2128 has an on-chip I^2C bus interface as standard.

The F-ZTAT 5 V/4 V version supports the operating ranges of the 5 V version and the 4 V version.

The operating range of the F-ZTAT low-voltage version will be decided later.

The above table includes products in the planning stage or under development. Information on the status of individual products can be obtained from Hitachi's sales offices.

Appendix G Package Dimensions

Figures G.1, G.2 and G.3 show the package dimensions of the H8S/2128 Series and H8S/2124 Series.

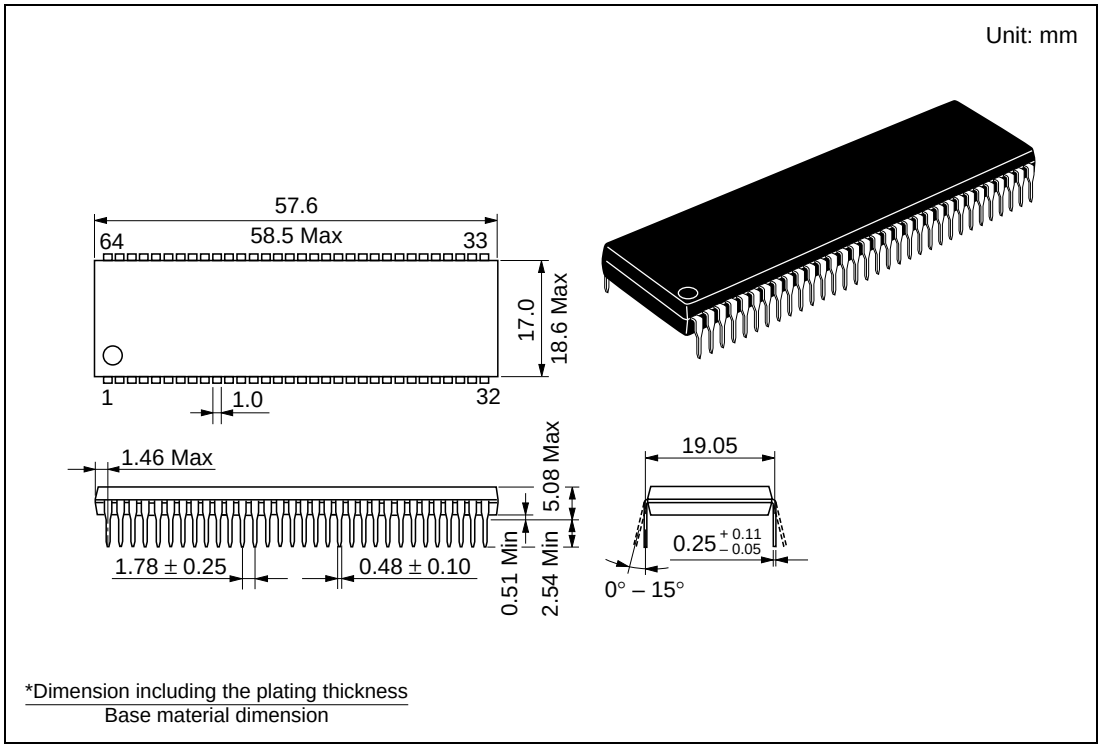


Figure G.1 Package Dimensions (DP-64S)

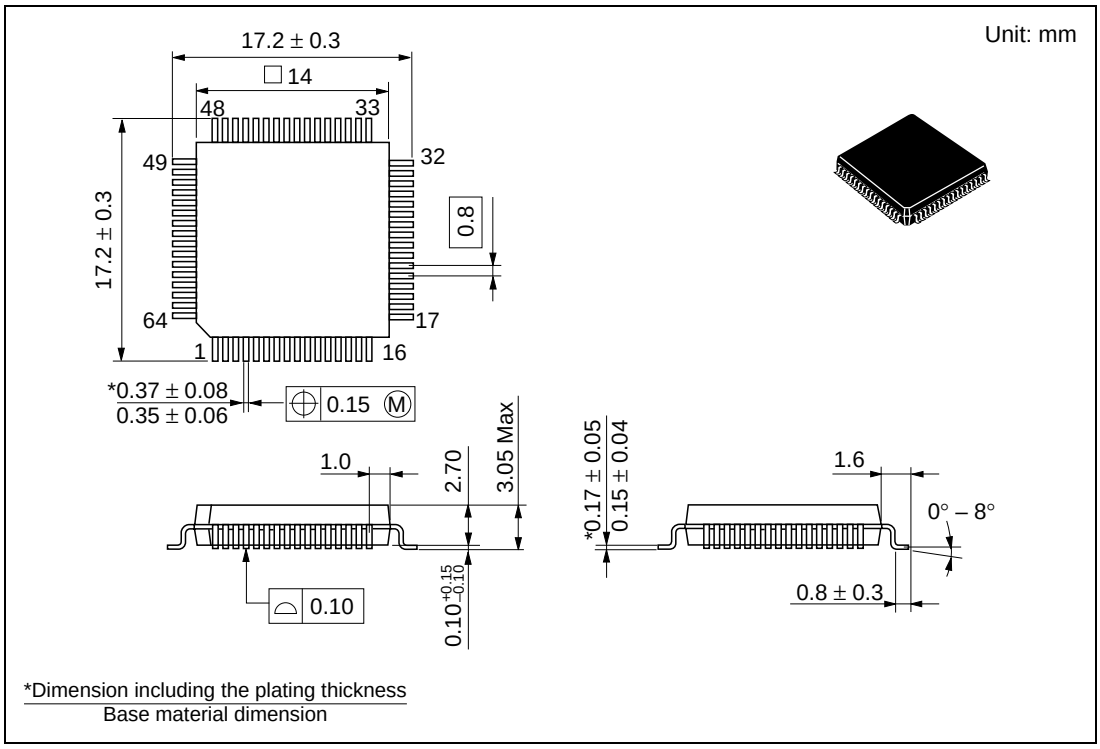


Figure G.2 Package Dimensions (FP-64A)

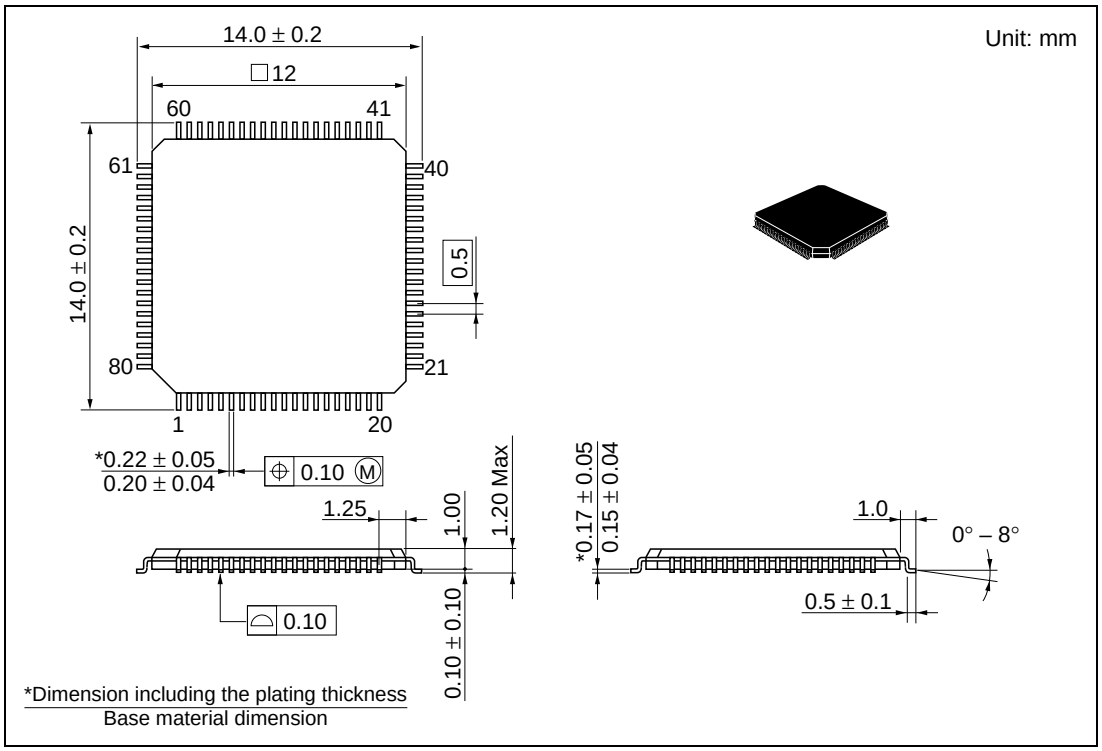


Figure G.3 Package Dimensions (TFP-80C)