Hitachi Single-Chip Microcomputer
H8/3534
HD6433534
H8/3522
HD6433522

Hardware Manual

HITACHI

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 H8/3534 and H8/3522 are high-performance microcontrollers with a fast H8/300 CPU core and a set of on-chip supporting functions optimized for embedded control. The 80-pin H8/3534 includes 32-kbyte ROM, 1-kbyte RAM, four types of timers, a serial communication interface, A/D converter, I/O ports, and other on-chip supporting functions needed in control system configurations, so that compact, high-performance systems can be implemented easily. The 64-pin H8/3522 is a subset version of the H8/3534, with 16-kbyte ROM, 512-byte RAM, and fewer on-chip supporting functions.

In program development for the H8/3534 and H8/3522, development tools should be used that support the H8/3394 and H8/3292-functional supersets of the H8/3534 and H8/3522- respectively. The H8/3534 and H8/3522 are also available in ZTAT™*[1] (zero turn-around time) versions-the H8/3334Y and H8/3294-which are functional supersets of the H8/3534 and H8/3522, respectively. These can be used effectively in program development and the initial stage of volume production.

The H8/3534 and H8/3522 have functional supersets in the H8/3397 Series and H8/3297 Series, respectively. The differences in terms of functions are that the lineup includes only one kind of ROM and one kind of RAM, the maximum operating frequency is 10 MHz, and the current dissipation values are not guaranteed in standby mode, one of the modes in the power-down state.

The H8/3534 and H8/3522 are single-chip microcomputers designed for consumer applications. If there is a need for ZTAT version or F-ZTAT™*[2] version for larger ROM/RAM capacity, for fast processing up to a maximum of 16 MHz, for drastically lower power consumption in standby mode in portable application systems, etc., or for high reliability in automotive or industrial applications, for instance, then the H8/3397 Series or H8/3297 Series should be used.

This manual describes the hardware of the H8/3534 and H8/3522. Refer to the H8/300 Series Programming Manual for a detailed description of the instruction set, and to the H8/3397 Series, H8/3337 Series, and H8/3334Y F-ZTAT Hardware Manual and the H8/3297 Series Hardware Manual for details of the high-end product series including the ZTAT version.

Notes: 1. ZTAT is a trademark of Hitachi, Ltd.
2. F-ZTAT is a trademark of Hitachi, Ltd.

Section 1 Overview

1.1 Overview

The H8/3534 and H8/3522 single-chip microcomputers (MCUs: microcomputer units) feature an H8/300 CPU core and a complement of on-chip supporting modules required for system configuration.

The H8/300 CPU is a high-speed processor with an architecture featuring powerful bit-manipulation instructions, ideally suited to realtime control applications. On-chip supporting modules in the 80-pin H8/3534 include 32-kbyte ROM, 1-kbyte RAM, four types of timers (16-bit free-running timer, two 8-bit timer channels, two PWM timer channels, and a watchdog timer), two serial communication interface (SCI) channels, an A/D converter, and I/O ports. The 64-pin H8/3522 is a subset version of the H8/3534, with 16-kbyte ROM and 512-byte RAM, no PWM timer, and only one SCI channel.

These microcomputers can operate in single-chip mode or two expanded modes, depending on the requirements of the application.

In program development for the H8/3534 and H8/3522, development tools should be used that support the functional-superset H8/3394 and H8/3292, respectively. As ZTAT™*[1] (zero turn-around time) versions, the H8/3334Y and H8/3294 ZTAT versions should be used. In this case, registers, etc., related to superset functions should not be accessed. In particular, take care not to write 1 to the HIE bit in SYSCR, the IICS, IICD, IICX, IICE, and STAC bits in STCR, and the RAMS and RAM0 bits in WSCR.

Note: 1. ZTAT is a trademark of Hitachi, Ltd.

Table 1-1 Features

Item	Specification
CPU	Two-way general register configuration • Eight 16-bit registers, or • Sixteen 8-bit registers High-speed operation • Maximum clock rate (ϕ clock): 10 MHz at 5 V • 8- or 16-bit register-register add/subtract: 200 ns (10 MHz) • 8 × 8-bit multiply: 1400 ns (10 MHz) • 16 ÷ 8-bit divide: 1400 ns (10 MHz) Streamlined, concise instruction set • Instruction length: 2 or 4 bytes • Register-register arithmetic and logic operations • MOV instruction for data transfer between registers and memory Instruction set features • Multiply instruction (8 bits × 8 bits) • Divide instruction (16 bits ÷ 8 bits) • Bit-accumulator instructions • Register-indirect specification of bit positions
Memory	• H8/3534: 32-kbyte ROM; 1-kbyte RAM • H8/3522: 16-kbyte ROM; 512-byte RAM
16-bit free-running timer (1 channel)	• One 16-bit free-running counter (can also count external events) • Two output-compare lines • Four input capture lines (can be buffered)
8-bit timer (2 channels)	Each channel has • One 8-bit up-counter (can also count external events) • Two time constant registers
PWM timer (2 channels) (H8/3534 only)	• Duty cycle can be set from 0 to 100% • Resolution: 1/250
Watchdog timer (WDT) (1 channel)	• Overflow can generate a reset or NMI interrupt • Also usable as interval timer
Serial communication interface (SCI) (H8/3534: 2 channels H8/3522: 1 channel)	• Asynchronous or synchronous mode (selectable) • Full duplex: can transmit and receive simultaneously • On-chip baud rate generator

Table 1-1 Features (cont)

Item	Specification			
Keyboard controller (H8/3534 only)	• Controls a matrix-scan keyboard by providing a keyboard scan function with wake-up interrupts and sense ports			
A/D converter	• 10-bit resolution • Eight channels: single or scan mode (selectable) • Start of A/D conversion can be externally triggered • Sample-and-hold function			
I/O ports (H8/3534)	• 58 input/output lines (16 of which can drive LEDs) • 8 input-only lines			
I/O ports (H8/3522)	• 43 input/output lines (16 of which can drive LEDs) • 8 input-only lines			
Interrupts (H8/3534)	• Nine external interrupt lines: $\overline{NMI}$, $\overline{IRQ}_0$ to $\overline{IRQ}_7$ • 26 on-chip interrupt sources			
Interrupts (H8/3522)	• Four external interrupt lines: $\overline{NMI}$, $\overline{IRQ}_0$ to $\overline{IRQ}_2$ • 19 on-chip interrupt sources			
Wait control	• Three selectable wait modes			
Operating modes	• Expanded mode with on-chip ROM disabled (mode 1) • Expanded mode with on-chip ROM enabled (mode 2) • Single-chip mode (mode 3)			
Power-down modes	• Sleep mode • Software standby mode • Hardware standby mode			
Other features	• On-chip oscillator			
Series lineup	**Product Name**	**Part Number**	**Package**	**ROM**
	H8/3534	HD6433534F10	80-pin QFP (FP-80A)	Mask ROM
	H8/3522	HD6433522F10	64-pin QFP (FP-64A)	
		HD6433522P10	64-pin shrink DIP (DP-64S)	

1.2 Block Diagram

Figure 1-1 shows a block diagram of the H8/3534. Figure 1-2 shows a block diagram of the H8/3522.

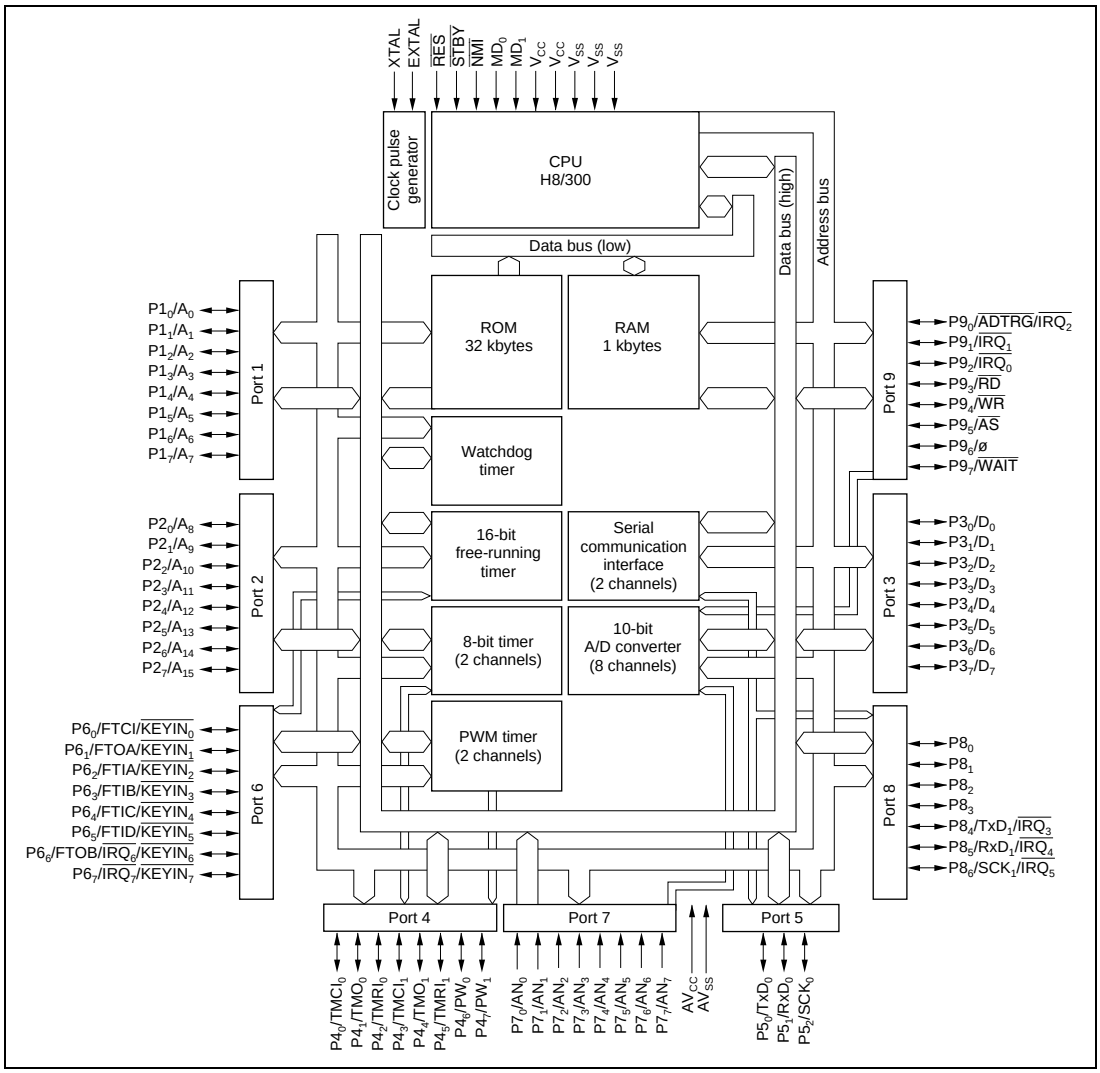


Figure 1-1 Block Diagram for H8/3534

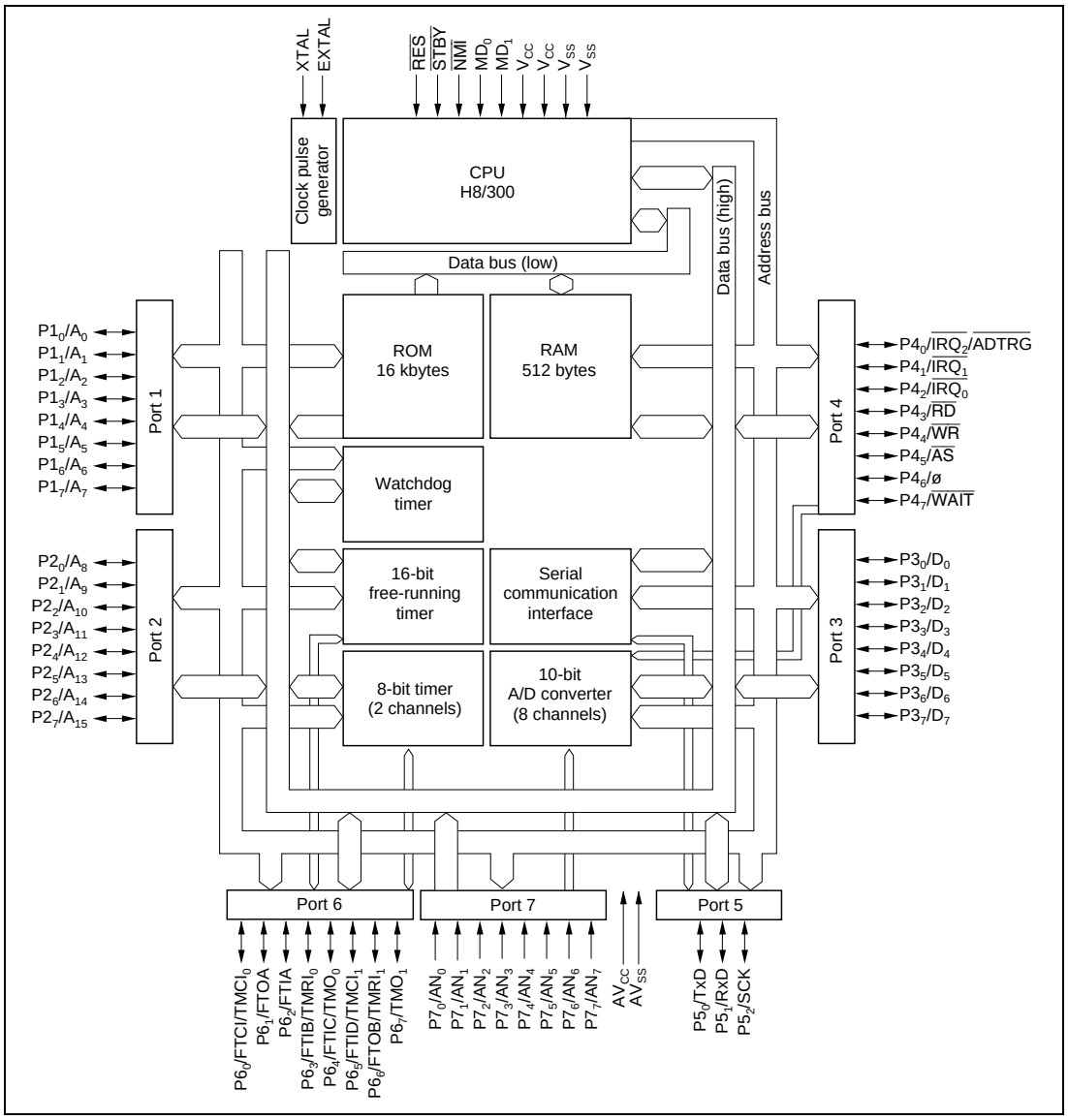


Figure 1-2 Block Diagram for H8/3522

1.3 Pin Assignments and Functions

1.3.1 Pin Arrangement

Figure 1-3 shows the pin arrangement of the FP-80A package for the H8/3534.

Figure 1-4 and 1-5 show the pin arrangement of the FP-64A and DP-64S packages for the H8/3522.

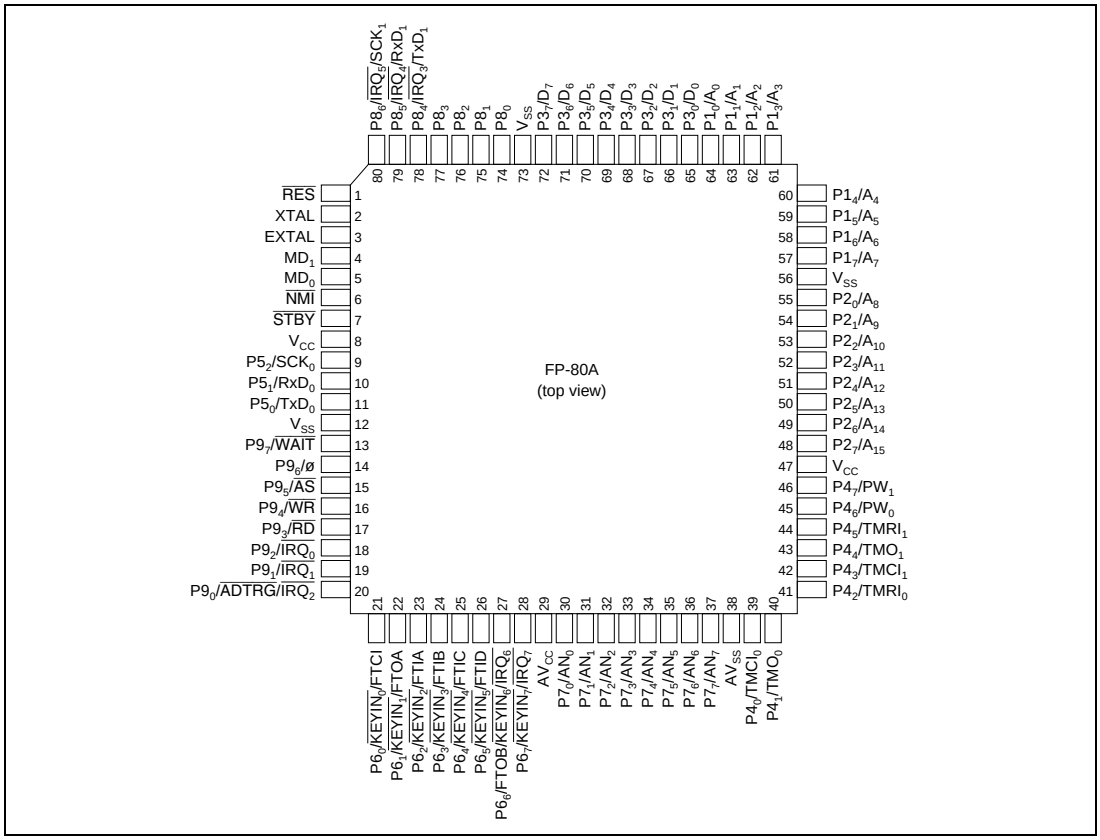


Figure 1-3 Pin Arrangement for H8/3534 (FP-80A, Top View)

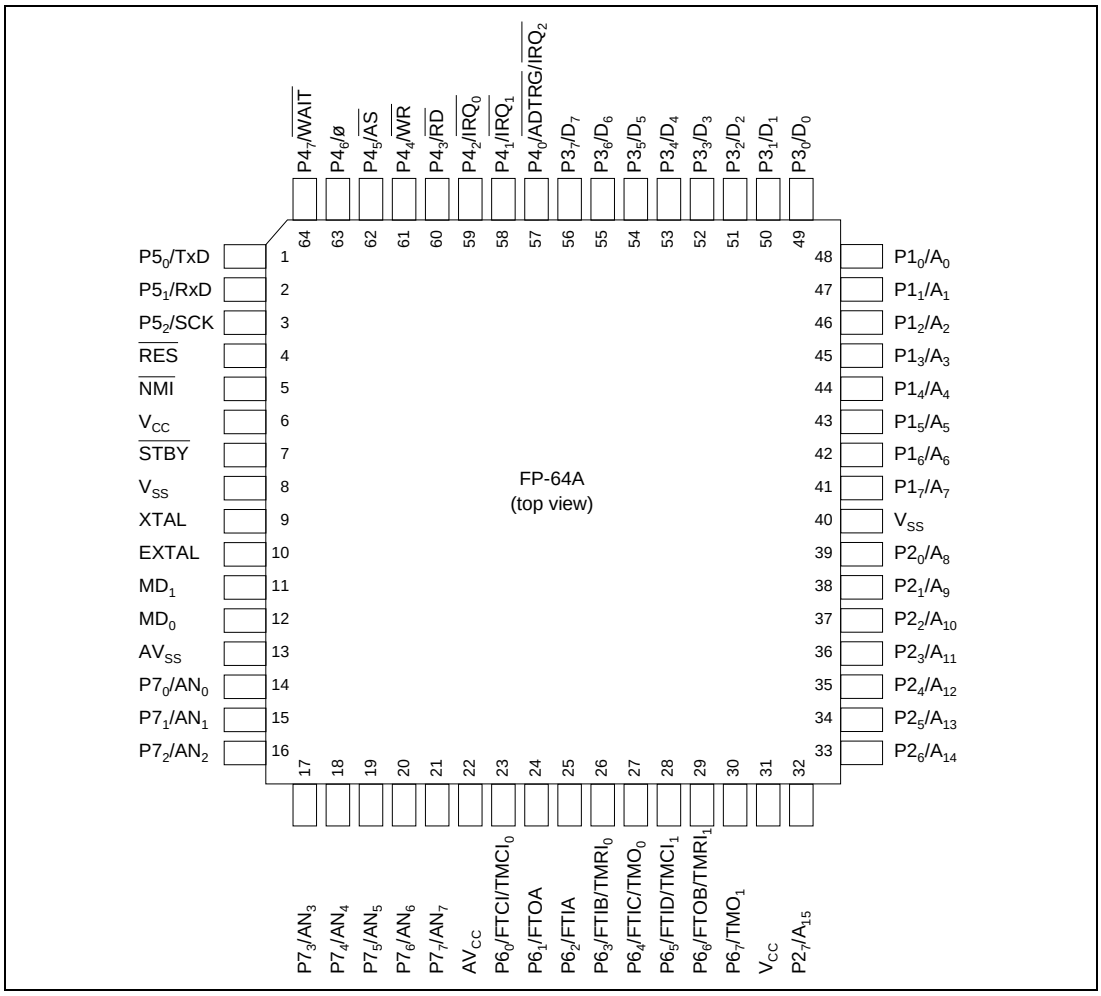


Figure 1-4 Pin Arrangement for H8/3522 (FP-64A, Top View)

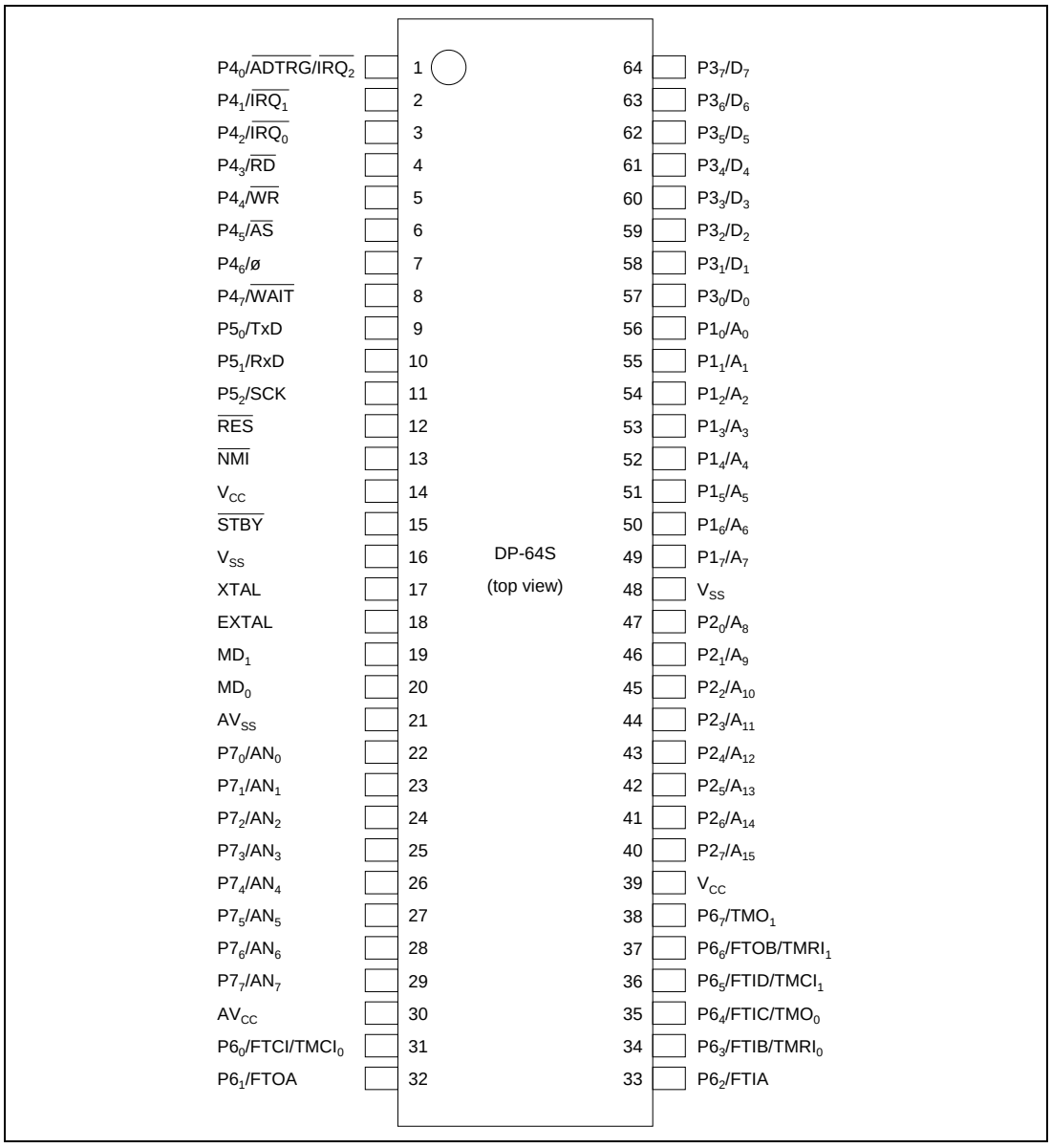


Figure 1-5 Pin Arrangement for H8/3522 (DP-64S, Top View)

1.3.2 Pin Functions

(1) Pin Assignments in Each Operating Mode: Tables 1-2 and 1-3 list the assignments of the pins of the H8/3534's FP-80A package and the H8/3522's FP-64A and DP-64S packages in each operating mode.

Table 1-2 Pin Assignments for H8/3534 in Each Operating Mode

Pin No.	Expanded Modes		Single-Chip Mode
FP-80A	**Mode 1**	**Mode 2**	**Mode 3**
1	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$
2	XTAL	XTAL	XTAL
3	EXTAL	EXTAL	EXTAL
4	MD_1	MD_1	MD_1
5	MD_0	MD_0	MD_0
6	$\overline{NMI}$	$\overline{NMI}$	$\overline{NMI}$
7	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$
8	V_{CC}	V_{CC}	V_{CC}
9	$P5_2/SCK_0$	$P5_2/SCK_0$	$P5_2/SCK_0$
10	$P5_1/RxD_0$	$P5_1/RxD_0$	$P5_1/RxD_0$
11	$P5_0/TxD_0$	$P5_0/TxD_0$	$P5_0/TxD_0$
12	V_{SS}	V_{SS}	V_{SS}
13	$P9_7/\overline{WAIT}$	$P9_7/\overline{WAIT}$	$P9_7$
14	ϕ	ϕ	$P9_6/\phi$
15	$\overline{AS}$	$\overline{AS}$	$P9_5$
16	$\overline{WR}$	$\overline{WR}$	$P9_4$
17	$\overline{RD}$	$\overline{RD}$	$P9_3$
18	$P9_2/\overline{IRQ}_0$	$P9_2/IRQ_0$	$P9_2/\overline{IRQ}_0$
19	$P9_1/\overline{IRQ}_1$	$P9_1/\overline{IRQ}_1$	$P9_1/\overline{IRQ}_1$
20	$P9_0/\overline{ADTRG}/IRQ_2$	$P9_0/\overline{ADTRG}/IRQ_2$	$P9_0/\overline{ADTRG}/IRQ_2$
21	$P6_0/\overline{KEYIN}_0$/FTCI	$P6_0/\overline{KEYIN}_0$/FTCI	$P6_0/\overline{KEYIN}_0$/FTCI
22	$P6_1/\overline{KEYIN}_1$/FTOA	$P6_1/\overline{KEYIN}_1$/FTOA	$P6_1/\overline{KEYIN}_1$/FTOA
23	$P6_2/\overline{KEYIN}_2$/FTIA	$P6_2/\overline{KEYIN}_2$/FTIA	$P6_2/\overline{KEYIN}_2$/FTIA
24	$P6_3/\overline{KEYIN}_3$/FTIB	$P6_3/\overline{KEYIN}_3$/FTIB	$P6_3/\overline{KEYIN}_3$/FTIB
25	$P6_4/\overline{KEYIN}_4$/FTIC	$P6_4/\overline{KEYIN}_4$/FTIC	$P6_4/\overline{KEYIN}_4$/FTIC

Table 1-2 Pin Assignments for H8/3534 in Each Operating Mode (cont)

Pin No.	Expanded Modes		Single-Chip Mode
FP-80A	**Mode 1**	**Mode 2**	**Mode 3**
26	$P6_5/\overline{KEYIN}_5$/FTID	$P6_5/\overline{KEYIN}_5$/FTID	$P6_5/\overline{KEYIN}_5$/FTID
27	$P6_6$/FTOB/$\overline{KEYIN}_6$/ $\overline{IRQ}_6$	$P6_6$/FTOB/$\overline{KEYIN}_6$/ $\overline{IRQ}_6$	$P6_6$/FTOB/$\overline{KEYIN}_6$/ $\overline{IRQ}_6$
28	$P6_7/\overline{KEYIN}_7/\overline{IRQ}_7$	$P6_7/\overline{KEYIN}_7/\overline{IRQ}_7$	$P6_7/\overline{KEYIN}_7/\overline{IRQ}_7$
29	AV_{CC}	AV_{CC}	AV_{CC}
30	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$
31	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$
32	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$
33	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$
34	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$
35	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$
36	$P7_6/AN_6$	$P7_6/AN_6$	$P7_6/AN_6$
37	$P7_7/AN_7$	$P7_7/AN_7$	$P7_7/AN_7$
38	AV_{SS}	AV_{SS}	AV_{SS}
39	$P4_0/TMCI_0$	$P4_0/TMCI_0$	$P4_0/TMCI_0$
40	$P4_1/TMO_0$	$P4_1/TMO_0$	$P4_1/TMO_0$
41	$P4_2/TMRI_0$	$P4_2/TMRI_0$	$P4_2/TMRI_0$
42	$P4_3/TMCI_1$	$P4_3/TMCI_1$	$P4_3/TMCI_1$
43	$P4_4/TMO_1$	$P4_4/TMO_1$	$P4_4/TMO_1$
44	$P4_5/TMRI_1$	$P4_5/TMRI_1$	$P4_5/TMRI_1$
45	$P4_6/PW_0$	$P4_6/PW_0$	$P4_6/PW_0$
46	$P4_7/PW_1$	$P4_7/PW_1$	$P4_7/PW_1$
47	V_{CC}	V_{CC}	V_{CC}
48	A_{15}	$P2_7/A_{15}$	$P2_7$
49	A_{14}	$P2_6/A_{14}$	$P2_6$

Table 1-2 Pin Assignments for H8/3534 in Each Operating Mode (cont)

Pin No.	Expanded Modes		Single-Chip Mode
FP-80A	**Mode 1**	**Mode 2**	**Mode 3**
50	A_{13}	$P2_5/A_{13}$	$P2_5$
51	A_{12}	$P2_4/A_{12}$	$P2_4$
52	A_{11}	$P2_3/A_{11}$	$P2_3$
53	A_{10}	$P2_2/A_{10}$	$P2_2$
54	A_9	$P2_1/A_9$	$P2_1$
55	A_8	$P2_0/A_8$	$P2_0$
56	V_{SS}	V_{SS}	V_{SS}
57	A_7	$P1_7/A_7$	$P1_7$
58	A_6	$P1_6/A_6$	$P1_6$
59	A_5	$P1_5/A_5$	$P1_5$
60	A_4	$P1_4/A_4$	$P1_4$
61	A_3	$P1_3/A_3$	$P1_3$
62	A_2	$P1_2/A_2$	$P1_2$
63	A_1	$P1_1/A_1$	$P1_1$
64	A_0	$P1_0/A_0$	$P1_0$
65	D_0	D_0	$P3_0$
66	D_1	D_1	$P3_1$
67	D_2	D_2	$P3_2$
68	D_3	D_3	$P3_3$
69	D_4	D_4	$P3_4$
70	D_5	D_5	$P3_5$
71	D_6	D_6	$P3_6$
72	D_7	D_7	$P3_7$
73	V_{SS}	V_{SS}	V_{SS}
74	$P8_0$	$P8_0$	$P8_0$

Table 1-2 Pin Assignments for H8/3534 in Each Operating Mode (cont)

Pin No.	Expanded Modes		Single-Chip Mode
FP-80A	**Mode 1**	**Mode 2**	**Mode 3**
75	$P8_1$	$P8_1$	$P8_1$
76	$P8_2$	$P8_2$	$P8_2$
77	$P8_3$	$P8_3$	$P8_3$
78	$P8_4/\overline{IRQ}_3/TxD_1$	$P8_4/\overline{IRQ}_3/TxD_1$	$P8_4/\overline{IRQ}_3/TxD_1$
79	$P8_5/\overline{IRQ}_4/RxD_1$	$P8_5/\overline{IRQ}_4/RxD_1$	$P8_5/\overline{IRQ}_4/RxD_1$
80	$P8_6/\overline{IRQ}_5/SCK_1$	$P8_6/\overline{IRQ}_5/SCK_1$	$P8_6/\overline{IRQ}_5/SCK_1$

Table 1-3 Pin Assignments for H8/3522 in Each Operating Mode

Pin No.		Expanded Modes		Single-Chip Mode
DP-64S	**FP-64A**	**Mode 1**	**Mode 2**	**Mode 3**
1	57	$P4_0/\overline{ADTRG}/\overline{IRQ}_2$	$P4_0/\overline{ADTRG}/\overline{IRQ}_2$	$P4_0/\overline{ADTRG}/\overline{IRQ}_2$
2	58	$P4_1/\overline{IRQ}_1$	$P4_1/\overline{IRQ}_1$	$P4_1/\overline{IRQ}_1$
3	59	$P4_2/\overline{IRQ}_0$	$P4_2/\overline{IRQ}_0$	$P4_2/\overline{IRQ}_0$
4	60	$\overline{RD}$	$\overline{RD}$	$P4_3$
5	61	$\overline{WR}$	$\overline{WR}$	$P4_4$
6	62	$\overline{AS}$	$\overline{AS}$	$P4_5$
7	63	ϕ	ϕ	$P4_6/\phi$
8	64	$P4_7/\overline{WAIT}$	$P4_7/\overline{WAIT}$	$P4_7$
9	1	$P5_0$/TxD	$P5_0$/TxD	$P5_0$/TxD
10	2	$P5_1$/RxD	$P5_1$/RxD	$P5_1$/RxD
11	3	$P5_2$/SCK	$P5_2$/SCK	$P5_2$/SCK
12	4	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$
13	5	$\overline{NMI}$	$\overline{NMI}$	$\overline{NMI}$
14	6	V_{CC}	V_{CC}	V_{CC}
15	7	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$
16	8	V_{SS}	V_{SS}	V_{SS}
17	9	XTAL	XTAL	XTAL
18	10	EXTAL	EXTAL	EXTAL
19	11	MD_1	MD_1	MD_1
20	12	MD_0	MD_0	MD_0
21	13	AV_{SS}	AV_{SS}	AV_{SS}
22	14	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$
23	15	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$
24	16	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$
25	17	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$
26	18	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$
27	19	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$
28	20	$P7_6/AN_6$	$P7_6/AN_6$	$P7_6/AN_6$
29	21	$P7_7/AN_7$	$P7_7/AN_7$	$P7_7/AN_7$
30	22	AV_{CC}	AV_{CC}	AV_{CC}
31	23	$P6_0$/FTCI/$TMCI_0$	$P6_0$/FTCI/$TMCI_0$	$P6_0$/FTCI/$TMCI_0$
32	24	$P6_1$/FTOA	$P6_1$/FTOA	$P6_1$/FTOA

Table 1-3 Pin Assignments for H8/3522 in Each Operating Mode (cont)

Pin No.		Expanded Modes		Single-Chip Mode
DP-64S	**FP-64A**	**Mode 1**	**Mode 2**	**Mode 3**
33	25	$P6_2$/FTIA	$P6_2$/FTIA	$P6_2$/FTIA
34	26	$P6_3$/FTIB/$TMRI_0$	$P6_3$/FTIB/$TMRI_0$	$P6_3$/FTIB/$TMRI_0$
35	27	$P6_4$/FTIC/TMO_0	$P6_4$/FTIC/TMO_0	$P6_4$/FTIC/TMO_0
36	28	$P6_5$/FTID/$TMCI_1$	$P6_5$/FTID/$TMCI_1$	$P6_5$/FTID/$TMCI_1$
37	29	$P6_6$/FTOB/$TMRI_1$	$P6_6$/FTOB/$TMRI_1$	$P6_6$/FTOB/$TMRI_1$
38	30	$P6_7$/TMO_1	$P6_7$/TMO_1	$P6_7$/TMO_1
39	31	V_{CC}	V_{CC}	V_{CC}
40	32	A_{15}	$P2_7$/A_{15}	$P2_7$
41	33	A_{14}	$P2_6$/A_{14}	$P2_6$
42	34	A_{13}	$P2_5$/A_{13}	$P2_5$
43	35	A_{12}	$P2_4$/A_{12}	$P2_4$
44	36	A_{11}	$P2_3$/A_{11}	$P2_3$
45	37	A_{10}	$P2_2$/A_{10}	$P2_2$
46	38	A_9	$P2_1$/A_9	$P2_1$
47	39	A_8	$P2_0$/A_8	$P2_0$
48	40	V_{SS}	V_{SS}	V_{SS}
49	41	A_7	$P1_7$/A_7	$P1_7$
50	42	A_6	$P1_6$/A_6	$P1_6$
51	43	A_5	$P1_5$/A_5	$P1_5$
52	44	A_4	$P1_4$/A_4	$P1_4$
53	45	A_3	$P1_3$/A_3	$P1_3$
54	46	A_2	$P1_2$/A_2	$P1_2$
55	47	A_1	$P1_1$/A_1	$P1_1$
56	48	A_0	$P1_0$/A_0	$P1_0$
57	49	D_0	D_0	$P3_0$
58	50	D_1	D_1	$P3_1$
59	51	D_2	D_2	$P3_2$
60	52	D_3	D_3	$P3_3$
61	53	D_4	D_4	$P3_4$
62	54	D_5	D_5	$P3_5$
63	55	D_6	D_6	$P3_6$
64	56	D_7	D_7	$P3_7$

Table 1-4 Pin Functions

		Pin No.				
		H8/3534	H8/3522			
Type	**Symbol**	**FP-80A**	**FP-64A**	**DP-64S**	**I/O**	**Name and Function**
Power	V_{CC}	8, 47	6, 31	14, 39	I	**Power:** Connected to the power supply. Connect both V_{CC} pins to the system power supply.
	V_{SS}	12, 56, 73	8, 40	16, 48	I	**Ground:** Connected to ground (0 V). Connect all V_{SS} pins to systemground (0 V).
Clock	XTAL	2	9	17	I	**Crystal:** Connected to a crystal oscillator. The crystal frequency should be the same as the desired system clock frequency. If an external clock is input at the EXTAL pin, a reverse-phase clock should be input at the XTAL pin.
	EXTAL	3	10	18	I	**External crystal:** Connected to a crystal oscillator or external clock. The frequency of the external clock should be the same as the desired system clock frequency. See section 6.2, Oscillator Circuit, for examples of connections to a crystal and external clock.
	ϕ	14	63	7	O	**System clock:** Supplies the system clock to peripheral devices.
System Control	$\overline{RES}$	1	4	12	I	**Reset:** A low input causes the chip to reset.
	$\overline{STBY}$	7	7	15	I	**Standby:** A transition to the hardware standby mode occurs when a low input is received at the $\overline{STBY}$ pin.
Address bus	A_{15} to A_0	48 to 55, 57 to 64	32 to 39 41 to 48	40 to 47 49 to 56	O	**Address bus:** Address output pins.
Data bus	D_7 to D_0	72 to 65	56 to 49	64 to 57	I/O	**Data bus:** 8-bit bidirectional data bus.

Table 1-4 Pin Functions (cont)

Type	Symbol	Pin No. H8/3534 FP-80A	Pin No. H8/3522 FP-64A	Pin No. H8/3522 DP-64S	I/O	Name and Function
Bus control	$\overline{WAIT}$	13	64	8	I	**Wait**: Requests the CPU to insert wait states into the bus cycle when an external address is accessed.
	$\overline{RD}$	17	60	4	O	**Read**: Goes low to indicate that the CPU is reading an external address.
	$\overline{WR}$	16	61	5	O	**Write**: Goes low to indicate that the CPU is writing to an external address.
	$\overline{AS}$	15	62	6	O	**Address strobe**: Goes low to indicate that there is a valid address on the address bus.
Interrupt signals	$\overline{NMI}$	6	5	13	I	**Nonmaskable interrupt**: Highest-priority interrupt request. The NMIEG bit in the system control register (SYSCR) determines whether the interrupt is recognized at the rising or falling edge of the NMI input.
	$\overline{IRQ}_0$ to $\overline{IRQ}_7$, $\overline{IRQ}_0$ to $\overline{IRQ}_2$	18 to 20, 78 to 80, 27, 28	57 to 59	1 to 3	I	**Interrupt request 0 to 7:** [H8/3534] **Interrupt request 0 to 2:** [H8/3522] Maskable interrupt request pins.
Operating control	MD_1 MD_0	4, 5	11 12	19 20	I	**Mode:** Input pins for setting the MCU mode operating mode according to the table below.

MD_1	MD_0	Mode	Description
0	1	Mode 1	Expanded mode with on-chip ROM disabled
1	0	Mode 2	Expanded mode with on-chip ROM enabled
1	1	Mode 3	Single-chip mode

Table 1-4 Pin Functions (cont)

		Pin No.				
		H8/3534		H8/3522		
Type	**Symbol**	**FP-80A**	**FP-64A**	**DP-64S**	**I/O**	**Name and Function**
16-bit free-running timer (FRT)	FTCI	21	23	31	I	**FRT counter clock input:** Input pin for an external clock signal for the free-running timer.
	FTOA	22	24	32	O	**FRT output compare A output:** Output compare A output pin.
	FTOB	27	29	37	O	**FRT output compare B output**: Output compare B output pin.
	FTIA	23	25	33	I	**FRT input capture A input:** Input capture A input pin.
	FTIB	24	26	34	I	**FRT input capture B input:** Input capture B input pin.
	FTIC	25	27	35	I	**FRT input capture C input:** Input capture C input pin.
	FTID	26	28	36	I	**FRT input capture D input:** Input capture D input pin.
8-bit timer	TMO_0 TMO_1	40 43	27 30	35 38	O	**8-bit timer output (channels 0 and 1):** Compare-match output pins for the 8-bit timers.
	$TMCI_0$ $TMCI_1$	39 42	23 28	31 36	I	**8-bit timer counter clock input (channels 0 and 1):** External clock input pins for the 8-bit timer counters.
	$TMRI_0$ $TMRI_1$	41 44	26 29	34 37	I	**8-bit timer counter reset input (channels 0 and 1):** A high input at these pins resets the 8-bit timer counters.
PWM timer [H8/3534 only]	PW_0 PW_1	45 46	—	—	O	**PWM timer output (channels 0 and 1):** Pulse-width modulation timer output pins.
Serial communi-cation interface(SCI)	TxD_0 TxD_1	11 78	1 —	9 —	O	**Transmit data (channels 0 and 1):** Data output pins for the serial communication interface.
	RxD_0 RxD_1	10 79	2 —	10 —	I	**Receive data (channels 0 and 1):** Data input pins for the serial communication interface.
	SCK_0 SCK_1	9 80	3 —	11 —	I/O	**Serial clock (channels 0 and 1):** Input/output pins for the serial clock.

Table 1-4 Pin Functions (cont)

		Pin No.				
		H8/3534	H8/3522			
Type	**Symbol**	**FP-80A**	**FP-64A**	**DP-64S**	**I/O**	**Name and Function**
Keyboard [H8/3534 only]	$\overline{KEYIN}_0$ to $\overline{KEYIN}_7$	21 to 28	—	—	I	**Keyboard input:** Input pins from a control matrix keyboard. (Keyboard scan signals are normally output from $P1_0$ to $P1_7$ and $P2_0$ to $P2_7$, allowing a maximum 16 × 8 key matrix. The number of keys can be further increased by use of other output ports.)
A/D converter	AN_7 to AN_0	37 to 30	21 to 14	29 to 22	I	**Analog input:** Analog signal input pins for the A/D converter.
	$\overline{ADTRG}$	20	57	1	I	**A/D trigger:** External trigger input for starting the A/D converter.
	AV_{CC}	29	22	30	I	**Analog reference voltage:** Reference voltage pin for the A/D converter. If the A/D converter are not used, connect AV_{CC} to the system power supply.
	AV_{SS}	38	13	21	I	**Analog ground:** Ground pin for the A/D converter. Connect to system ground (0 V).
I/O ports	$P1_7$ to $P1_0$	57 to 64	41 to 48	49 to 56	I/O	**Port 1:** An 8-bit input/output port with programmable MOS input pull-ups and LED driving capability. The direction of each bit can be selected in the port 1 data direction register (P1DDR).
	$P2_7$ to $P2_0$	48 to 55	32 to 39	40 to 47	I/O	**Port 2:** An 8-bit input/output port with programmable MOS input pull-ups and LED driving capability. The direction of each bit can be selected in the port 2 data direction register (P2DDR).
	$P3_7$ to $P3_0$	72 to 65	56 to 49	64 to 57	I/O	**Port 3:** An 8-bit input/output port with programmable MOS input pull-ups. The direction of each bit can be selected in the port 3 data direction register (P3DDR).

Table 1-4 Pin Functions (cont)

		Pin No.				
		H8/3534	H8/3522			
Type	**Symbol**	**FP-80A**	**FP-64A**	**DP-64S**	**I/O**	**Name and Function**
I/O ports	$P4_7$ to $P4_0$	46 to 39	—	—	I/O	**Port 4 [H8/3534]:** An 8-bit input/output port. The direction of each bit can be selected in the port 4 data direction register (P4DDR).
	$P5_2$ to $P5_0$	9 to 11	3 to 1	11 to 9	I/O	**Port 5:** A 3-bit input/output port. The direction of each bit can be selected in the port 5 data direction register (P5DDR).
	$P6_7$ to $P6_0$	28 to 21	30 to 23	38 to 31	I/O	**Port 6:** An 8-bit input/output port with programmable MOS input pull-ups. The direction of each bit can be selected in the port 6 data direction register (P6DDR).
	$P7_7$ to $P7_0$	37 to 30	21 to 14	29 to 22	I	**Port 7:** An 8-bit input port.
	$P8_6$ to $P8_0$	80 to 74	—	—	I/O	**Port 8:** A 7-bit input/output port. The direction of each bit can be selected in the port 8 data direction register (P8DDR).
	$P9_7$ to $P9_0$ [H8/3534] $P4_7$ to $P4_0$ [H8/3522]	13 to 20	64 to 57	8 to 1	I/O	**Port 9 [H8/3534]:** An 8-bit input/ output port. The direction of each bit (except for $P9_6$) can be selected in the port 9 data direction register (P9DDR). **Port 4 [H8/3522]:** An 8-bit input/output port. The direction of each bit (except for $P4_6$) can be selected in the port 4 data direction register (P4DDR).

Section 2 CPU

2.1 Overview

The H8/300 CPU is a fast central processing unit with eight 16-bit general registers (also configurable as 16 eight-bit registers) and a concise instruction set designed for high-speed operation.

2.1.1 Features

The main features of the H8/300 CPU are listed below.

- Two-way register configuration
  - — Sixteen 8-bit general registers, or
  - — Eight 16-bit general registers
- Instruction set with 57 basic instructions, including:
  - — Multiply and divide instructions
  - — Powerful bit-manipulation instructions
- Eight addressing modes
  - — Register direct (Rn)
  - — Register indirect (@Rn)
  - — Register indirect with displacement (@(d:16, Rn))
  - — Register indirect with post-increment or pre-decrement (@Rn+ or @–Rn)
  - — Absolute address (@aa:8 or @aa:16)
  - — Immediate (#xx:8 or #xx:16)
  - — PC-relative (@(d:8, PC))
  - — Memory indirect (@@aa:8)
- Maximum 64-kbyte address space
- High-speed operation
  - — All frequently-used instructions are executed in two to four states
- Maximum clock rate (ϕ clock): 10 MHz at 5 V
  - — 8- or 16-bit register-register add or subtract: 200 ns (10 MHz)
  - — 8 × 8-bit multiply: 1400 ns (10 MHz)
  - — 16 ÷ 8-bit divide: 1400 ns (10 MHz)
- Power-down mode
  - — SLEEP instruction

2.1.2 Address Space

The H8/300 CPU supports an address space with a maximum size of 64 kbytes for program code and data combined. The memory map differs depending on the mode (mode 1, 2, or 3). For details, see section 3.4, Address Space Map in Each Operating Mode.

2.1.3 Register Configuration

Figure 2-1 shows the internal register structure of the H8/300 CPU. There are two groups of registers: the general registers and control registers.

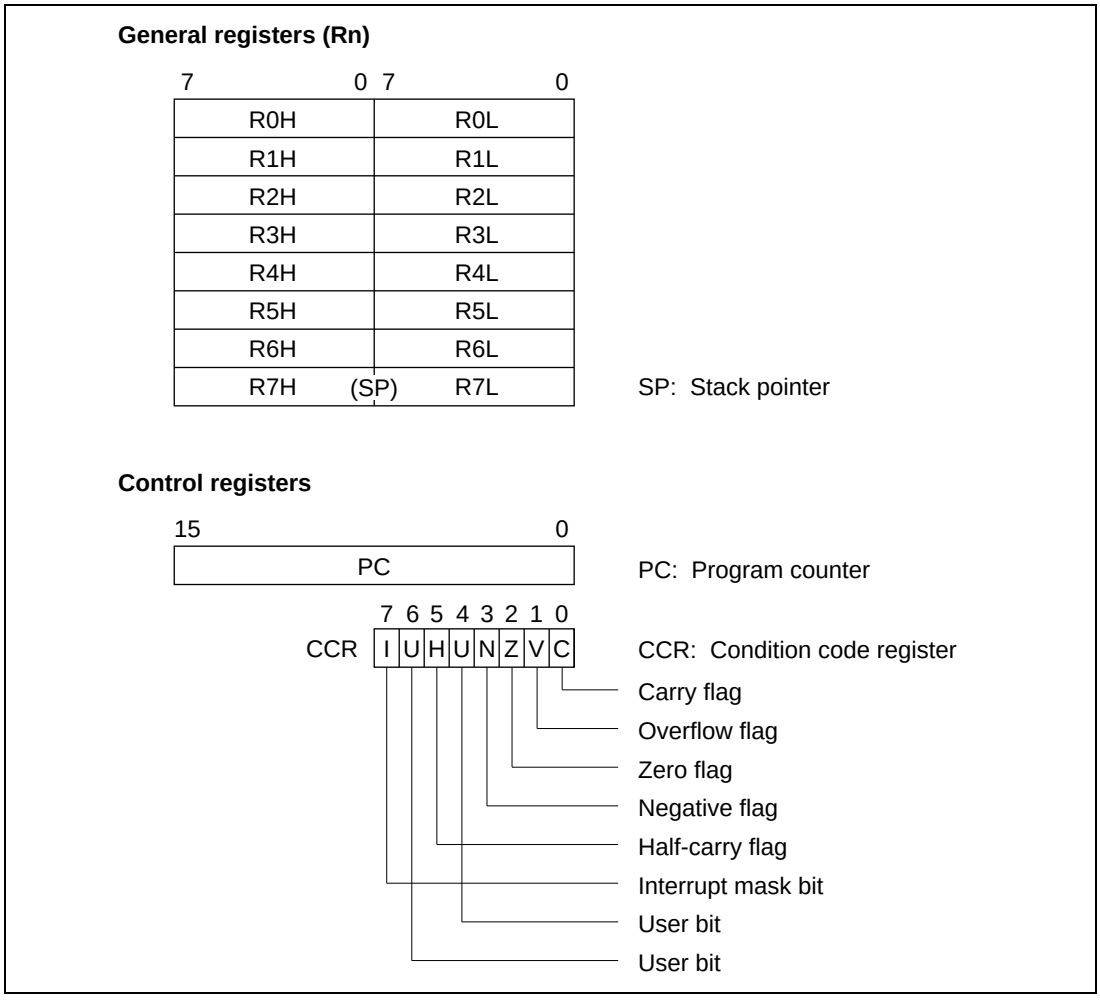


Figure 2-1 CPU Registers

2.2 Register Descriptions

2.2.1 General Registers

All the general registers can be used as both data registers and address registers. When used as address registers, the general registers are accessed as 16-bit registers (R0 to R7). When used as data registers, they can be accessed as 16-bit registers, or the high and low bytes can be accessed separately as 8-bit registers (R0H to R7H and R0L to R7L).

R7 also functions as the stack pointer, used implicitly by hardware in processing interrupts and subroutine calls. In assembly-language coding, R7 can also be denoted by the letters SP. As indicated in figure 2-2, R7 (SP) points to the top of the stack.

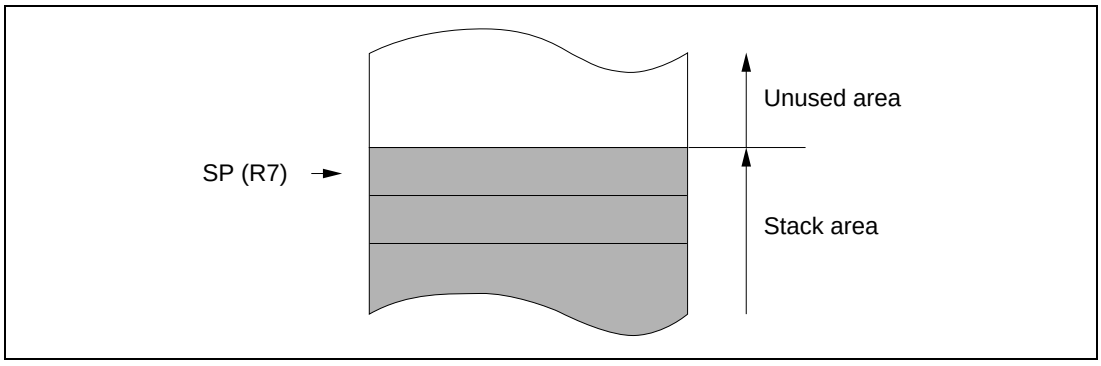


Figure 2-2 Stack Pointer

2.2.2 Control Registers

The CPU control registers include a 16-bit program counter (PC) and an 8-bit condition code register (CCR).

(1) Program Counter (PC): This 16-bit register indicates the address of the next instruction the CPU will execute. Each instruction is accessed in 16 bits (1 word), so the least significant bit of the PC is ignored (always regarded as 0).

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

Bit 7—Interrupt Mask Bit (I): When this bit is set to 1, all interrupts except NMI are masked. This bit is set to 1 automatically by a reset and at the start of interrupt handling.

Bit 6—User Bit (U): This bit can be written and read by software (using the LDC, STC, ANDC, ORC, and XORC instructions).

Bit 5—Half-Carry Flag (H): This flag is set to 1 when the ADD.B, ADDX.B, SUB.B, SUBX.B, NEG.B, or CMP.B instruction causes a carry or borrow out of bit 3, and is cleared to 0 otherwise. Similarly, it is set to 1 when the ADD.W, SUB.W, or CMP.W instruction causes a carry or borrow out of bit 11, and cleared to 0 otherwise. It is used implicitly in the DAA and DAS instructions.

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

Bit 3—Negative Flag (N): This flag indicates the most significant bit (sign bit) of the result of an instruction.

Bit 2—Zero Flag (Z): This flag is set to 1 to indicate a zero result and cleared to 0 to indicate a nonzero result.

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

Bit 0—Carry Flag (C): This flag is used by:

— Add and subtract instructions, to indicate a carry or borrow at the most significant bit of the result

— Shift and rotate instructions, to store the value shifted out of the most significant or least significant bit

— Bit manipulation and bit load instructions, as a bit accumulator

The LDC, STC, ANDC, ORC, and XORC instructions enable the CPU to load and store the CCR, and to set or clear selected bits by logic operations. The N, Z, V, and C flags are used in conditional branching instructions (B*cc*).

For the action of each instruction on the flag bits, see the *H8/300 Series Programming Manual*.

2.2.3 Initial Register Values

When the CPU is reset, the program counter (PC) is loaded from the vector table and the interrupt mask bit (I) in the CCR is set to 1. The other CCR bits and the general registers are not initialized. In particular, the stack pointer (R7) is not initialized. The stack pointer and CCR should be initialized by software, by the first instruction executed after a reset.

2.3 Data Formats

The H8/300 CPU can process 1-bit data, 4-bit (BCD) data, 8-bit (byte) data, and 16-bit (word) data.

- Bit manipulation instructions operate on 1-bit data specified as bit n (n = 0, 1, 2, ..., 7) in a byte operand.
- All arithmetic and logic instructions except ADDS and SUBS can operate on byte data.
- The DAA and DAS instruction perform decimal arithmetic adjustments on byte data in packed BCD form. Each nibble of the byte is treated as a decimal digit.
- The MOV.W, ADD.W, SUB.W, CMP.W, ADDS, SUBS, MULXU (8 bits × 8 bits), and DIVXU (16 bits ÷ 8 bits) instructions operate on word data.

2.3.1 Data Formats in General Registers

Data of all the sizes above can be stored in general registers as shown in figure 2-3.

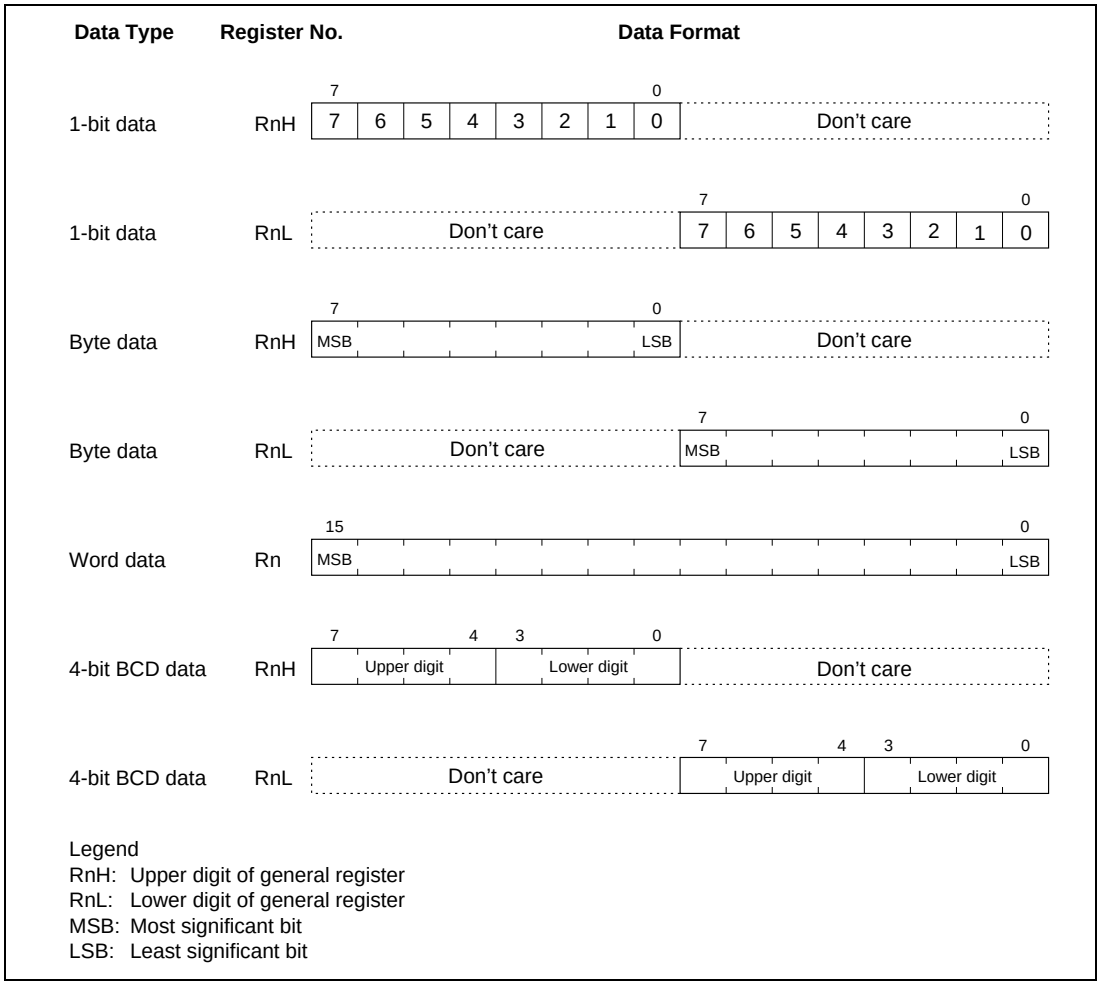


Figure 2-3 Register Data Formats

2.3.2 Memory Data Formats

Figure 2-4 indicates the data formats in memory.

Word data stored in memory must always begin at an even address. In word access the least significant bit of the address is regarded as 0. If an odd address is specified, no address error occurs but the access is performed at the preceding even address. This rule affects MOV.W instructions and branching instructions, and implies that only even addresses should be stored in the vector table.

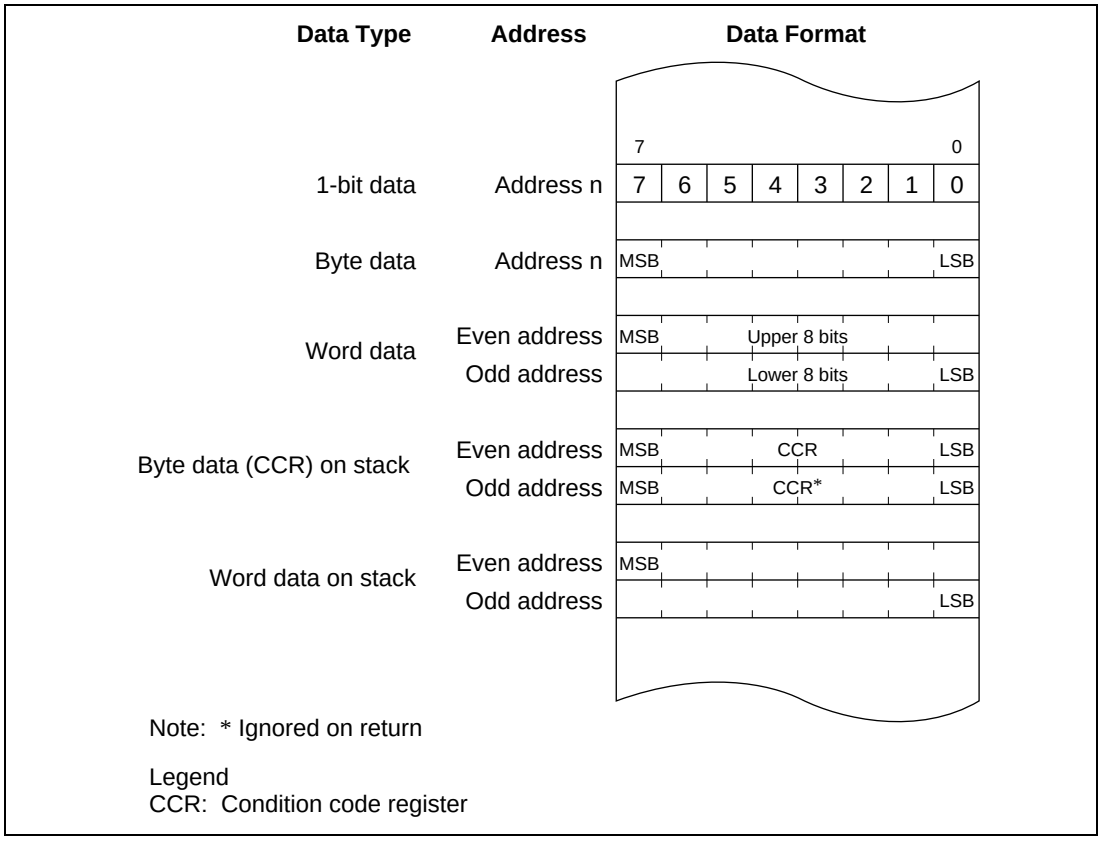


Figure 2-4 Memory Data Formats

When the stack is addressed by register R7, it must always be accessed a word at a time. When the CCR is pushed on the stack, two identical copies of the CCR are pushed to make a complete word. When they are restored, the lower byte is ignored.

2.4 Addressing Modes

2.4.1 Addressing Mode

The H8/300 CPU supports eight addressing modes. Each instruction uses a subset of these addressing modes.

Table 2-1 Addressing Modes

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

(1) Register Direct—Rn: The register field of the instruction specifies an 8- or 16-bit general register containing the operand. In most cases the general register is accessed as an 8-bit register. Only the MOV.W, ADD.W, SUB.W, CMP.W, ADDS, SUBS, MULXU (8 bits × 8 bits), and DIVXU (16 bits ÷ 8 bits) instructions have 16-bit operands.

(2) Register Indirect—@Rn: The register field of the instruction specifies a 16-bit general register containing the address of the operand.

(3) Register Indirect with Displacement—@(d:16, Rn): This mode, which is used only in MOV instructions, is similar to register indirect but the instruction has a second word (bytes 3 and 4) which is added to the contents of the specified general register to obtain the operand address. For the MOV.W instruction, the resulting address must be even.

(4) Register Indirect with Post-Increment or Pre-Decrement—@Rn+ or @–Rn:

— Register indirect with Post-Increment—@Rn+

The @Rn+ mode is used with MOV instructions that load registers from memory.

It is similar to the register indirect mode, but the 16-bit general register specified in the register field of the instruction is incremented after the operand is accessed. The size of the increment is 1 or 2 depending on the size of the operand: 1 for MOV.B; 2 for MOV.W. For MOV.W, the original contents of the 16-bit general register must be even.

— Register Indirect with Pre-Decrement—@–Rn

The @–Rn mode is used with MOV instructions that store register contents to memory.

It is similar to the register indirect mode, but the 16-bit general register specified in the register field of the instruction is decremented before the operand is accessed. The size of the decrement is 1 or 2 depending on the size of the operand: 1 for MOV.B; 2 for MOV.W. For MOV.W, the original contents of the 16-bit general register must be even.

(5) Absolute Address—@aa:8 or @aa:16: The instruction specifies the absolute address of the operand in memory. The MOV.B instruction uses an 8-bit absolute address of the form H'FFxx. The upper 8 bits are assumed to be 1, so the possible address range is H'FF00 to H'FFFF (65280 to 65535). The MOV.B, MOV.W, JMP, and JSR instructions can use 16-bit absolute addresses.

(6) Immediate—#xx:8 or #xx:16: The instruction contains an 8-bit operand in its second byte, or a 16-bit operand in its third and fourth bytes. Only MOV.W instructions can contain 16-bit immediate values.

The ADDS and SUBS instructions implicitly contain the value 1 or 2 as immediate data. Some bit manipulation instructions contain 3-bit immediate data (#xx:3) in the second or fourth byte of the instruction, specifying a bit number.

(7) Program-Counter-Relative—@(d:8, PC): This mode is used to generate branch addresses in the Bcc and BSR instructions. An 8-bit value in byte 2 of the instruction code is added as a sign-extended value to the program counter contents. The result must be an even number. The possible branching range is –126 to +128 bytes (–63 to +64 words) from the current address.

(8) Memory Indirect—@@aa:8: This mode can be used by the JMP and JSR instructions. The second byte of the instruction code specifies an 8-bit absolute address from H'0000 to H'00FF (0 to 255). The word located at this address contains the branch address. The upper 8 bits of the absolute address are 0 (H'00), thus the branch address is limited to values from 0 to 255 (H'0000 to H'00FF). Note that some of the addresses in this range are also used in the vector table. Refer to section 3.4, Address Space Map in Each Operating Mode.

If an odd address is specified as a branch destination or as the operand address of a MOV.W instruction, the least significant bit is regarded as 0, causing word access to be performed at the address preceding the specified address. See section 2.3.2, Memory Data Formats, for further information.

2.4.2 Calculation of Effective Address

Table 2-2 shows how the H8/300 calculates effective addresses in each addressing mode.

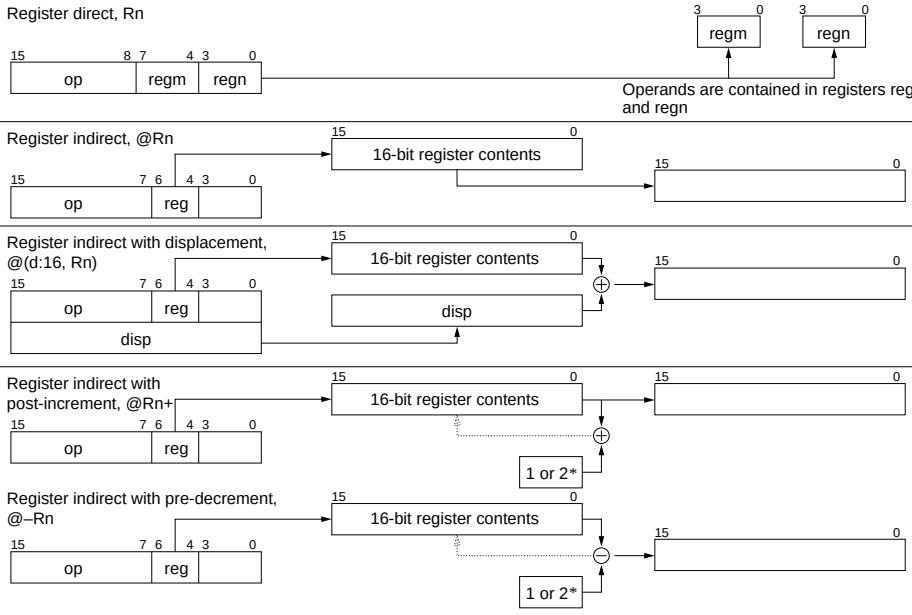
Arithmetic, logic, and shift instructions use register direct addressing (1). The ADD.B, ADDX.B, SUBX.B, CMP.B, AND.B, OR.B, and XOR.B instructions can also use immediate addressing (6).

The MOV instruction uses all the addressing modes except program-counter relative (7) and memory indirect (8).

Bit manipulation instructions use register direct (1), register indirect (2), or 8-bit absolute (5) addressing to identify a byte operand, and 3-bit immediate addressing to identify a bit within the byte. The BSET, BCLR, BNOT, and BTST instructions can also use register direct addressing (1) to identify the bit.

Table 2-2 Effective Address Calculation

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address
1	Register direct, Rn		Operands are contained in registers regm and regn
2	Register indirect, @Rn	16-bit register contents	
3	Register indirect with displacement, @(d:16, Rn)	16-bit register contents + disp	
4	Register indirect with post-increment, @Rn+	16-bit register contents + 1 or 2*	
	Register indirect with pre-decrement, @–Rn	16-bit register contents – 1 or 2*	



Note: * 1 for a byte operand, 2 for a word operand

Table 2-2 Effective Address Calculation

Table 2-2 Effective Address Calculation (cont)

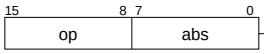
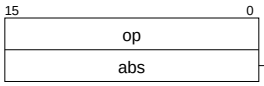
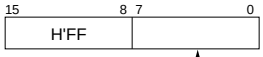
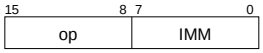
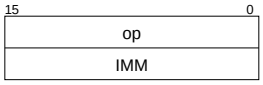
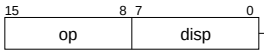
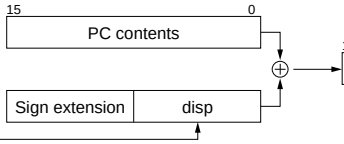
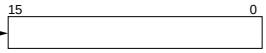
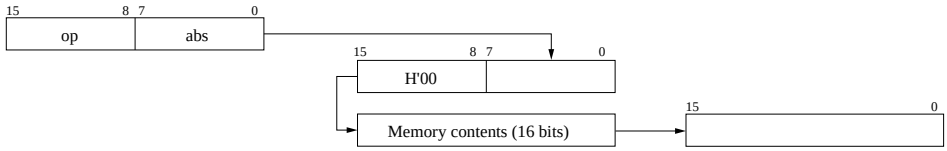
No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective address
5	Absolute address @aa:8  @aa:16 		 
6	Immediate #xx:8  #xx:16 		Operand is 1- or 2-byte immediate data
7	PC-relative @(d:8, PC) 		

Table 2-2 Effective Address Calculation (cont)

Table 2-2 Effective Address Calculation (cont)

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

Legend

reg, regm, regn:	Register field
op:	Operation field
disp:	Displacement
IMM:	Immediate data
abs:	Absolute address

Table 2-2 Effective Address Calculation (cont)

2.5 Instruction Set

The H8/300 CPU has 57 types of instructions, which are classified by function in table 2-3.

Table 2-3 Instruction Classification

Function	Instructions	Types
Data transfer	MOV, MOVTPE[*3], MOVFPE[*3], PUSH[*1], POP[*1]	3
Arithmetic operations	ADD, SUB, ADDX, SUBX, INC, DEC, ADDS, SUBS, DAA, DAS, MULXU, DIVXU, CMP, NEG	14
Logic operations	AND, OR, XOR, NOT	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST	14
Branch	Bcc[*2], JMP, BSR, JSR, RTS	5
System control	RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	8
Block data transfer	EEPMOV	1
		Total 57

Notes: 1. USH Rn is equivalent to MOV.W Rn, @–SP.
POP Rn is equivalent to MOV.W @SP+, Rn.
2. Bcc is a conditional branch instruction in which cc represents a condition code.
3. Not supported by the H8/3534 and H8/3522.

The following sections give a concise summary of the instructions in each category, and indicate the bit patterns of their object code. The notation used is defined next.

Operation Notation

Rd	General register (destination)
Rs	General register (source)
Rn	General register
(EAd)	Destination operand
(EAs)	Source operand
SP	Stack pointer
PC	Program counter
CCR	Condition code register
N	N (negative) flag of CCR
Z	Z (zero) flag of CCR
V	V (overflow) flag of CCR
C	C (carry) flag of CCR
#imm	Immediate data
#xx:3	3-bit immediate data
#xx:8	8-bit immediate data
#xx:16	16-bit immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Exclusive Logical OR
→	Move
¬	NOT (logical complement)

2.5.1 Data Transfer Instructions

Table 2-4 describes the data transfer instructions. Figure 2-5 shows their object code formats.

Table 2-4 Data Transfer Instructions

Instruction	Size*	Function
MOV	B/W	(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. The Rn, @Rn, @(d:16, Rn), @aa:16, #xx:8 or #xx:16, @–Rn, and @Rn+ addressing modes are available for byte or word data. The @aa:8 addressing mode is available for byte data only. The @–R7 and @R7+ modes require word operands. Do not specify byte size for these two modes.
MOVTPE	B	Not supported by the H8/3534 and H8/3522.
MOVFPE	B	Not supported by the H8/3534 and H8/3522.
PUSH	W	Rn → @–SP Pushes a 16-bit general register onto the stack. Equivalent to MOV.W Rn, @–SP.
POP	W	@SP+ → Rn Pops a 16-bit general register from the stack. Equivalent to MOV.W @SP+, Rn.

Note: * Size: Operand size
B: Byte
W: Word

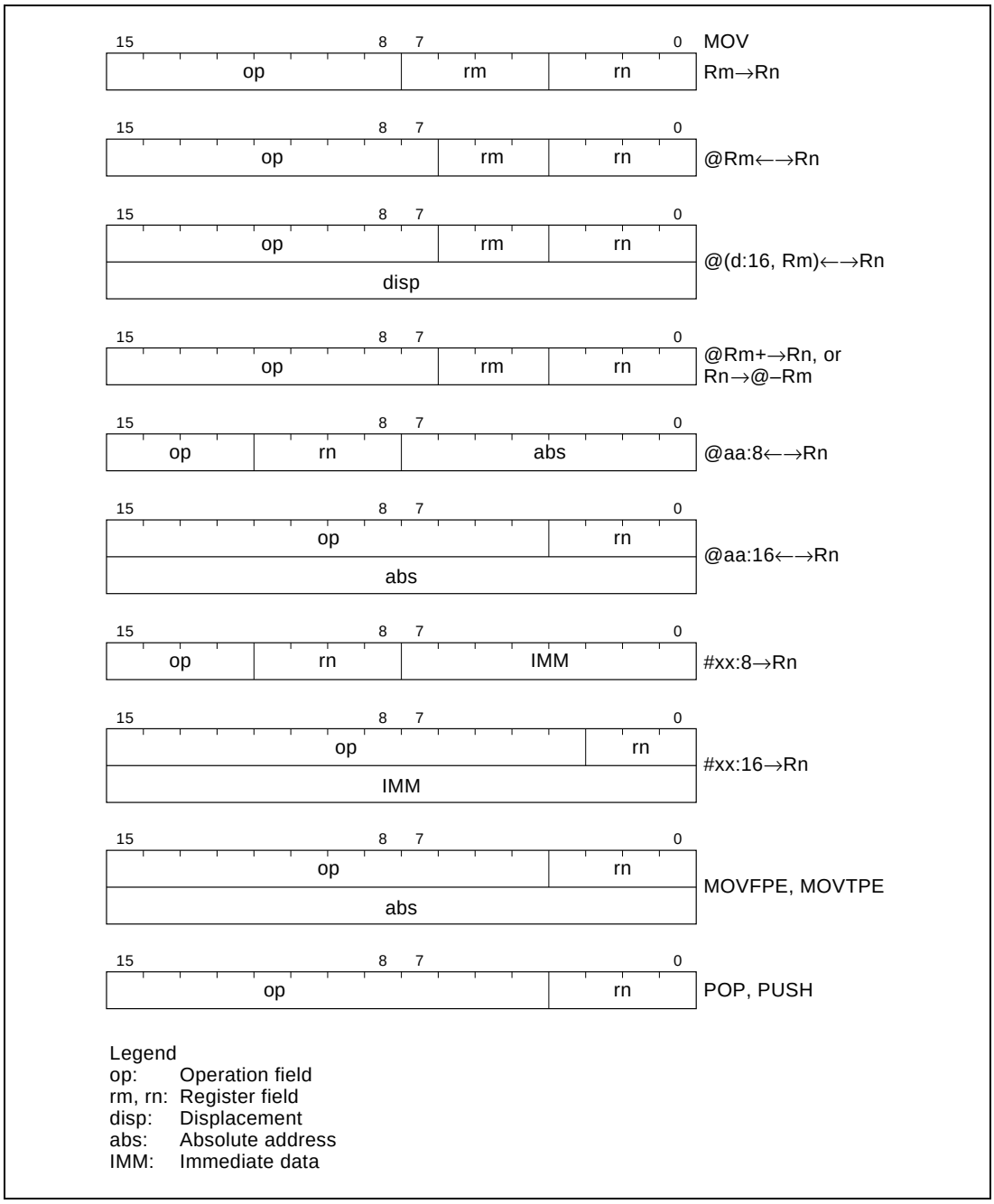


Figure 2-5 Data Transfer Instruction Codes

2.5.2 Arithmetic Operations

Table 2-5 describes the arithmetic instructions. See figure 2-6 in section 2.5.4, Shift Operations, for their object codes.

Table 2-5 Arithmetic Instructions

Instruction	Size*	Function
ADD SUB	B/W	Rd ± Rs → Rd, Rd + #imm → Rd Performs addition or subtraction on data in two general registers, or addition on immediate data and data in a general register. Immediate data cannot be subtracted from data in a general register. Word data can be added or subtracted only when both words are in general registers.
ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #imm ± C → Rd Performs addition or subtraction with carry or borrow on byte data in two general registers, or addition or subtraction on immediate data and data in a general register.
INC DEC	B	Rd ± #1 → Rd Increments or decrements a general register.
ADDS SUBS	W	Rd ± #imm → Rd Adds or subtracts immediate data to or from data in a general register. The immediate data must be 1 or 2.
DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts (adjusts to packed BCD) an addition or subtraction result in a general register by referring to the CCR.
MULXU	B	Rd × Rs → Rd Performs 8-bit × 8-bit unsigned multiplication on data in two general registers, providing a 16-bit result.
DIVXU	B	Rd ÷ Rs → Rd Performs 16-bit ÷ 8-bit unsigned division on data in two general registers, providing an 8-bit quotient and 8-bit remainder.
CMP	B/W	Rd – Rs, Rd – #imm Compares data in a general register with data in another general register or with immediate data. Word data can be compared only between two general registers.
NEG	B	0 – Rd → Rd Obtains the two's complement (arithmetic complement) of data in a general register.

Note: * Size: Operand size
B: Byte
W: Word

2.5.3 Logic Operations

Table 2-6 describes the four instructions that perform logic operations. See figure 2-6 in section 2.5.4, Shift Operations, for their object codes.

Table 2-6 Logic Operation Instructions

Instruction	Size*	Function
AND	B	Rd ∧ Rs → Rd, Rd ∧ #imm → Rd Performs a logical AND operation on a general register and another general register or immediate data.
OR	B	Rd ∨ Rs → Rd, Rd ∨ #imm → Rd Performs a logical OR operation on a general register and another general register or immediate data.
XOR	B	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	~ (Rd) → (Rd) Obtains the one's complement (logical complement) of general register contents.

Note: * Size: Operand size
B: Byte

2.5.4 Shift Operations

Table 2-7 describes the eight shift instructions. Figure 2-6 shows the object code formats of the arithmetic, logic, and shift instructions.

Table 2-7 Shift Instructions

Instruction	Size*	Function
SHAL SHAR	B	Rd shift → Rd Performs an arithmetic shift operation on general register contents.
SHLL SHLR	B	Rd shift → Rd Performs a logical shift operation on general register contents.
ROTL ROTR	B	Rd rotate → Rd Rotates general register contents.
ROTXL ROTXR	B	Rd rotate through carry → Rd Rotates general register contents through the C (carry) bit.

Note: * Size: Operand size
B: Byte

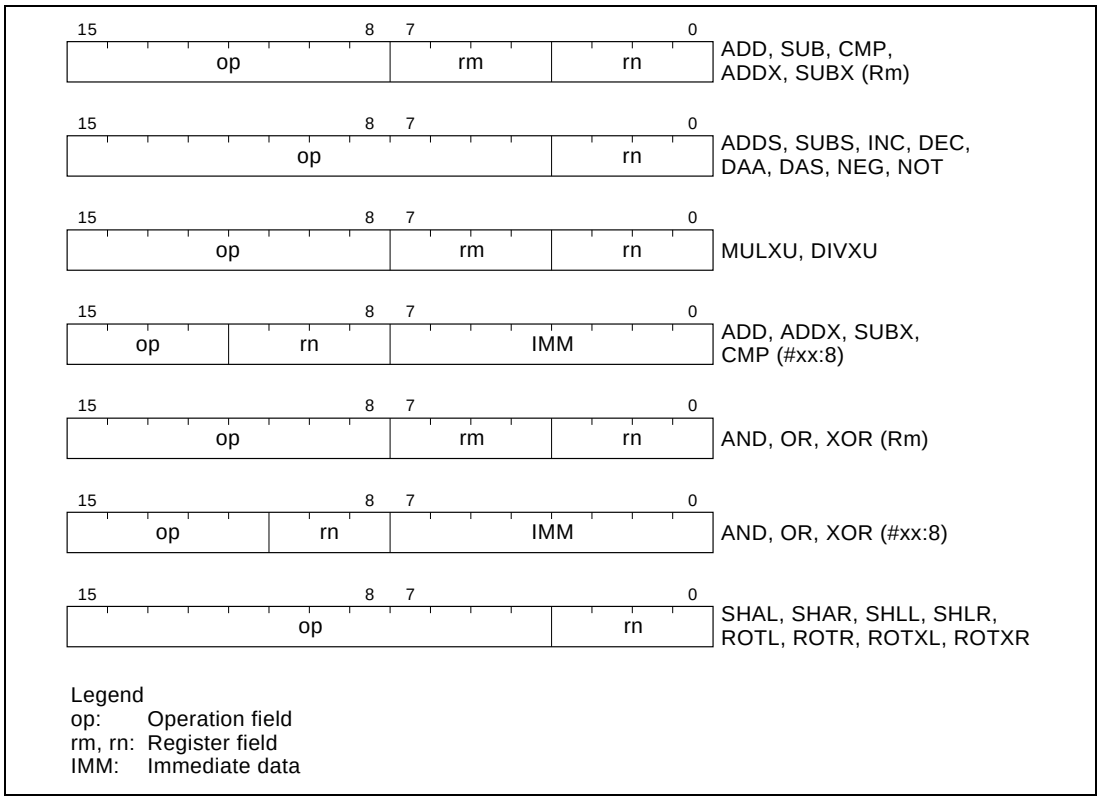


Figure 2-6 Arithmetic, Logic, and Shift Instruction Codes

2.5.5 Bit Manipulations

Table 2-8 describes the bit-manipulation instructions. Figure 2-7 shows their object code formats.

Table 2-8 Bit-Manipulation Instructions

Instruction	Size*	Function
BSET	B	1 → (<bit no.> of <EAd>) Sets a specified bit in a general register or memory to 1. The bit is specified by a bit number, given in 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 to 0. The bit is specified by a bit number, given in 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. The bit is specified by a bit number, given in 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 and sets or clears the Z flag accordingly. The bit is specified by a bit number, given in 3-bit immediate data or the lower three bits of a general register.
BAND	B	C ∧ (<bit no.> of <EAd>) → C ANDs the C flag with a specified bit in a general register or memory.
BIAND		C ∧ [¬ (<bit no.> of <EAd>)] → C ANDs the C flag with the inverse of a specified bit in a general register or memory. The bit number is specified by 3-bit immediate data.
BOR	B	C ∨ (<bit no.> of <EAd>) → C ORs the C flag with a specified bit in a general register or memory.
BIOR		C ∨ [¬ (<bit no.> of <EAd>)] → C ORs the C flag with the inverse of a specified bit in a general register or memory. The bit number is specified by 3-bit immediate data.
BXOR	B	C ⊕ (<bit no.> of <EAd>) → C XORs the C flag with a specified bit in a general register or memory.

Note: * Size: Operand size
B: Byte

Table 2-8 Bit-Manipulation Instructions (cont)

Instruction	Size*	Function
BIXOR	B	C ⊕ ¬ [(<bit no.> of <EAd>)] → C XORs the C flag with the inverse of a specified bit in a general register or memory.
		The bit number is specified by 3-bit immediate data.
BLD	B	(<bit no.> of <EAd>) → C Copies a specified bit in a general register or memory to the C flag.
BILD		¬ (<bit no.> of <EAd>) → C Copies the inverse of a specified bit in a general register or memory to the C flag. The bit number is specified by 3-bit immediate data.
BST	B	C → (<bit no.> of <EAd>) Copies the C flag to a specified bit in a general register or memory.
BIST		¬ C → (<bit no.> of <EAd>) Copies the inverse of the C flag to a specified bit in a general register or memory. The bit number is specified by 3-bit immediate data.

Note: * Size: Operand size
B: Byte

Notes on Bit Manipulation Instructions: BSET, BCLR, BNOT, BST, and BIST are read-modify-write instructions. They read a byte of data, modify one bit in the byte, then write the byte back. Care is required when these instructions are applied to registers with write-only bits and to the I/O port registers.

Step		Description
1	Read	Read one data byte at the specified address
2	Modify	Modify one bit in the data byte
3	Write	Write the modified data byte back to the specified address

Example 1: BCLR is executed to clear bit 0 in the port 4 data direction register (P4DDR) under the following conditions.

$P4_7$: Input pin, low
$P4_6$: Input pin, high
$P4_5$ – $P4_0$: Output pins, low

The intended purpose of this BCLR instruction is to switch $P4_0$ from output to input.

Before Execution of BCLR Instruction

	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low	High	Low	Low	Low	Low	Low	Low
DDR	0	0	1	1	1	1	1	1
DR	1	0	0	0	0	0	0	0

Execution of BCLR Instruction

```
BCLR  #0, @P4DDR      ;Clear bit 0 in data direction register
```

After Execution of BCLR Instruction

	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
Input/output	Output	Output	Output	Output	Output	Output	Output	Input
Pin state	Low	High	Low	Low	Low	Low	Low	High
DDR	1	1	1	1	1	1	1	0
DR	1	0	0	0	0	0	0	0

Explanation: To execute the BCLR instruction, the CPU begins by reading P4DDR. Since P4DDR is a write-only register, it is read as H'FF, even though its true value is H'3F.

Next the CPU clears bit 0 of the read data, changing the value to H'FE.

Finally, the CPU writes this value (H'FE) back to P4DDR to complete the BCLR instruction.

As a result, $P4_0$DDR is cleared to 0, making $P4_0$ an input pin. In addition, $P4_7$DDR and $P4_6$DDR are set to 1, making $P4_7$ and $P4_6$ output pins.

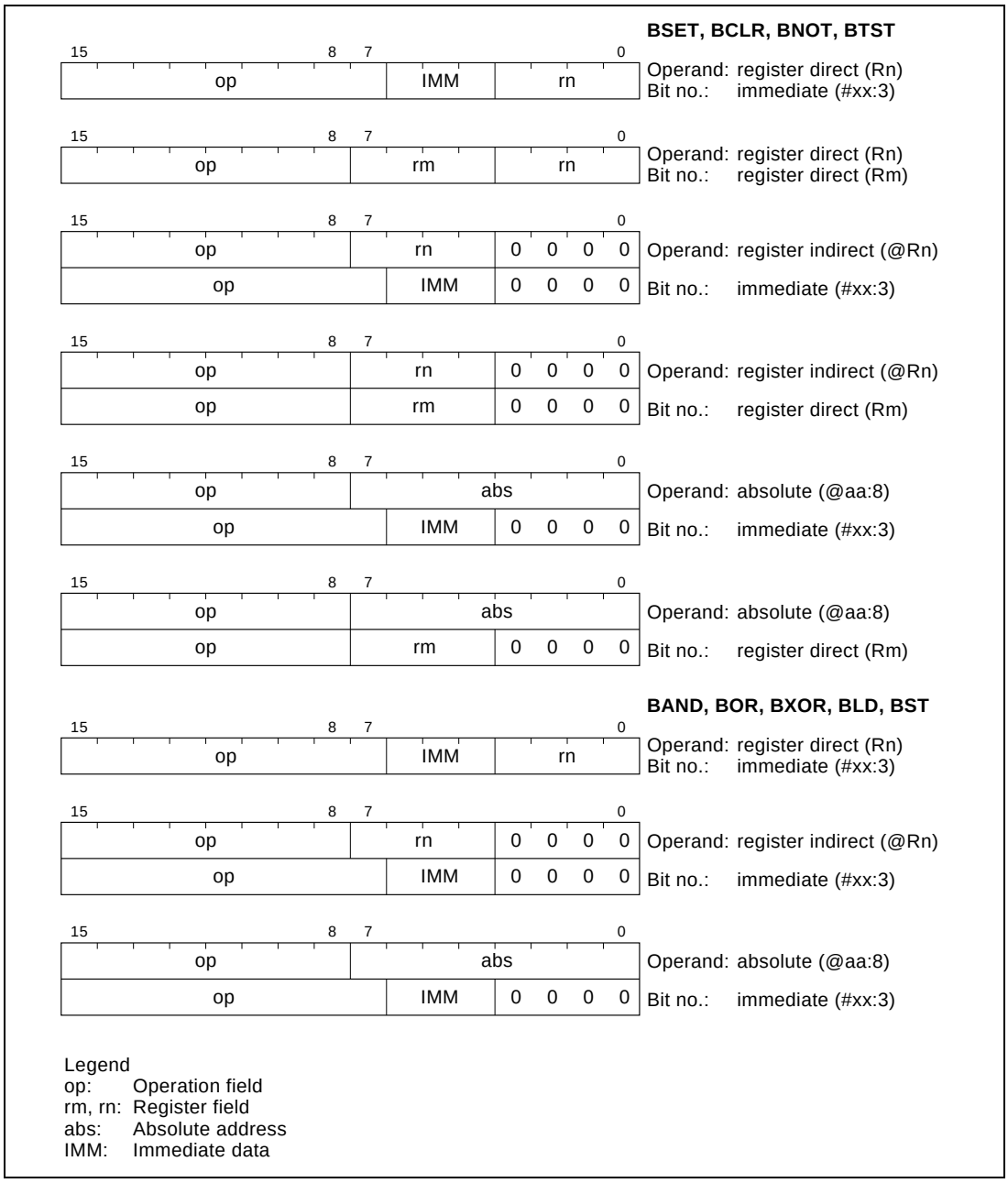


Figure 2-7 Bit Manipulation Instruction Codes (1)

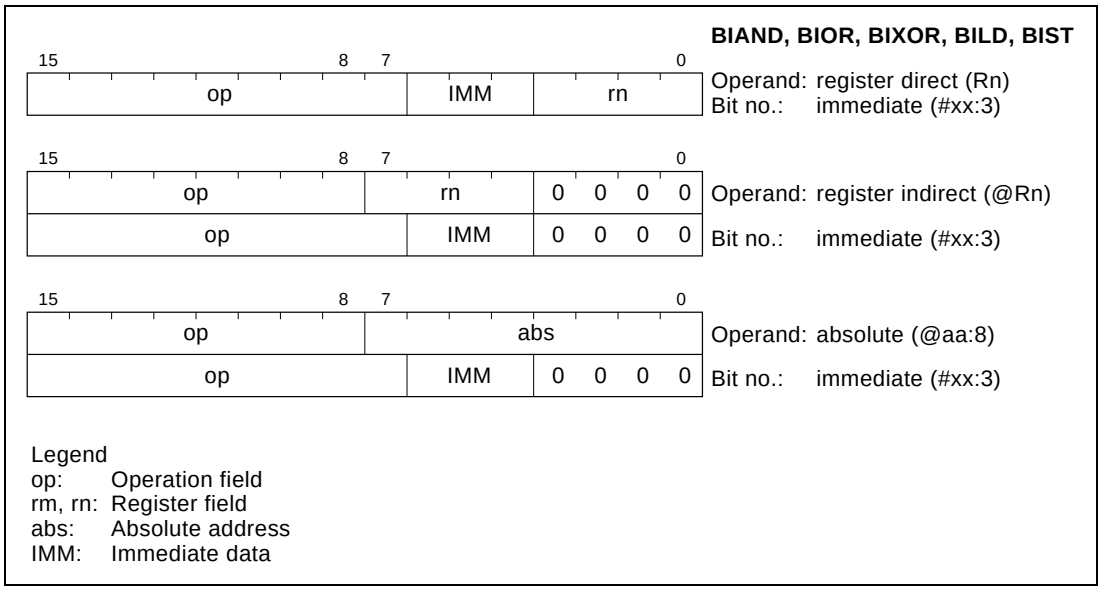


Figure 2-7 Bit Manipulation Instruction Codes (2)

2.5.6 Branching Instructions

Table 2-9 describes the branching instructions. Figure 2-8 shows their object code formats.

Table 2-9 Branching Instructions

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

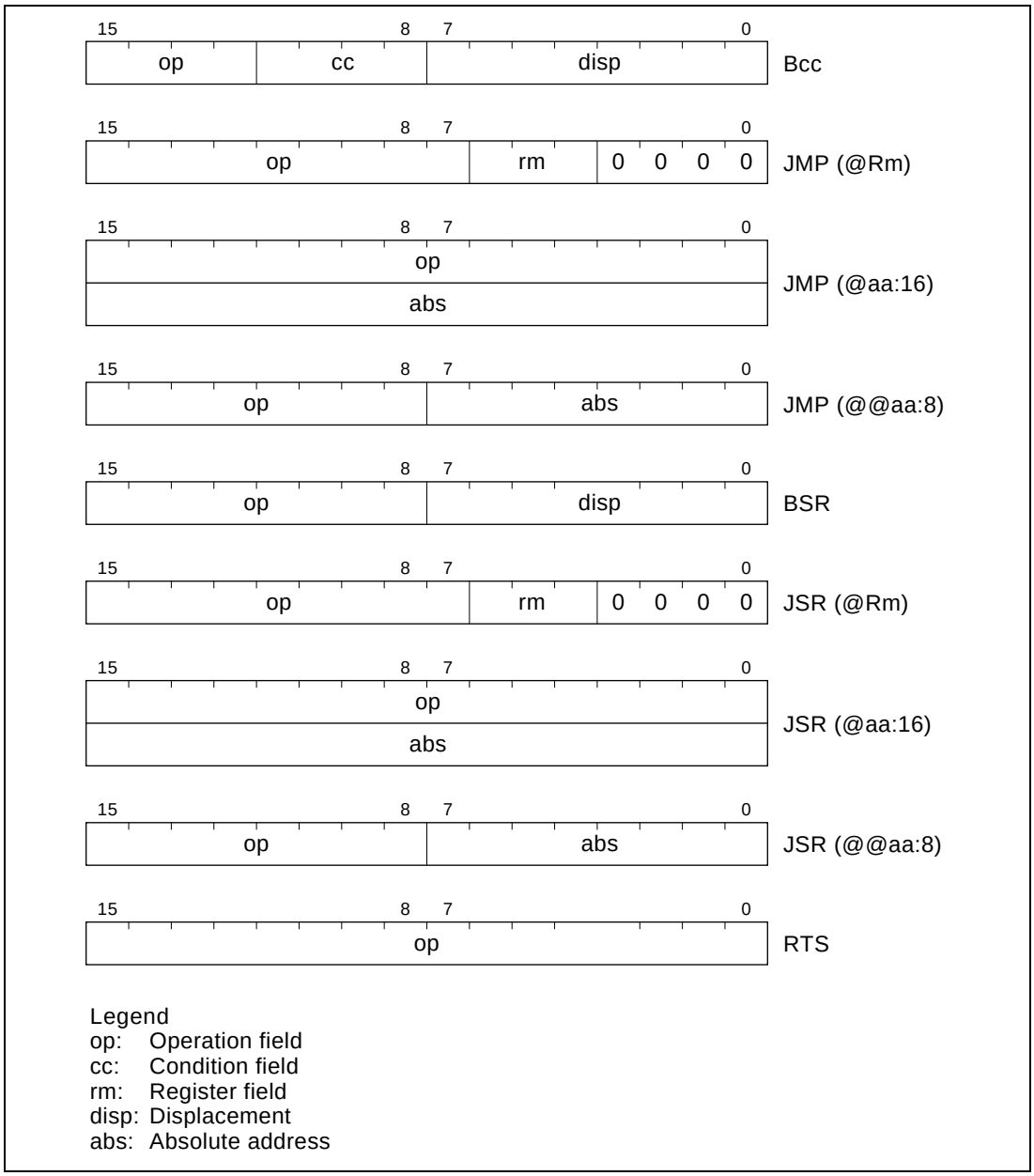


Figure 2-8 Branching Instruction Codes

2.5.7 System Control Instructions

Table 2-10 describes the system control instructions. Figure 2-9 shows their object code formats.

Table 2-10 System Control Instructions

Instruction	Size	Function
RTE	—	Returns from an exception-handling routine.
SLEEP	—	Causes a transition to the power-down state.
LDC	B	Rs → CCR, #imm → CCR Moves immediate data or general register contents to the condition code register.
STC	B	CCR → Rd Copies the condition code register to a specified general register.
ANDC	B	CCR ∧ #imm → CCR Logically ANDs the condition code register with immediate data.
ORC	B	CCR ∨ #imm → CCR Logically ORs the condition code register with immediate data.
XORC	B	CCR ⊕ #imm → CCR Logically exclusive-ORs the condition code register with immediate data.
NOP	—	PC + 2 → PC Only increments the program counter.

Note: * Size: Operand size
B: Byte

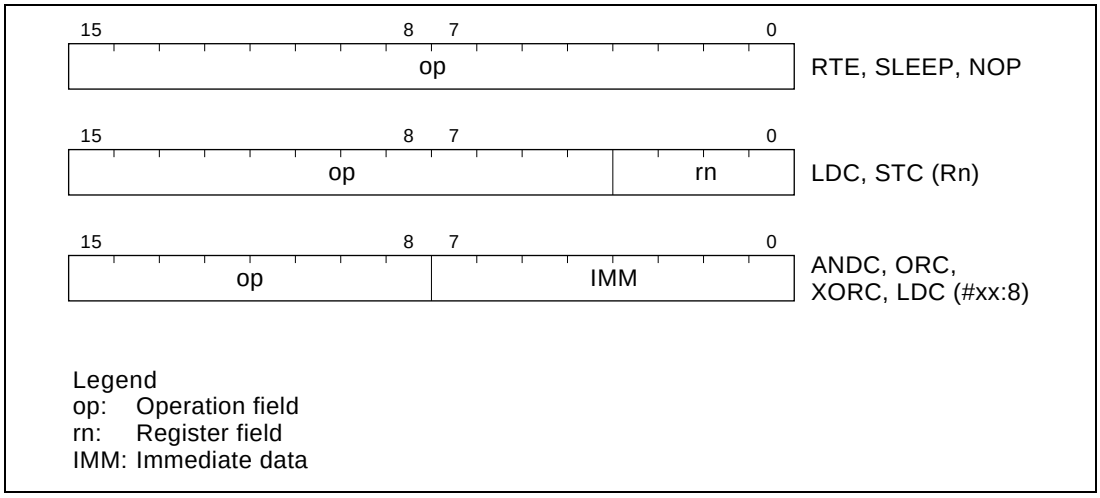


Figure 2-9 System Control Instruction Codes

2.5.8 Block Data Transfer Instruction

Table 2-11 describes the EEPMOV instruction. Figure 2-10 shows its object code format.

Table 2-11 Block Data Transfer Instruction

Instruction	Size	Function
EEPMOV	—	if R4L ≠ 0 then repeat @R5+ → @R6+ R4L – 1 → R4L until R4L = 0 else next; Moves a data block according to parameters set in general registers R4L, R5, and R6. R4L: size of block (bytes) R5: starting source address R6: starting destination address Execution of the next instruction starts as soon as the block transfer is completed.

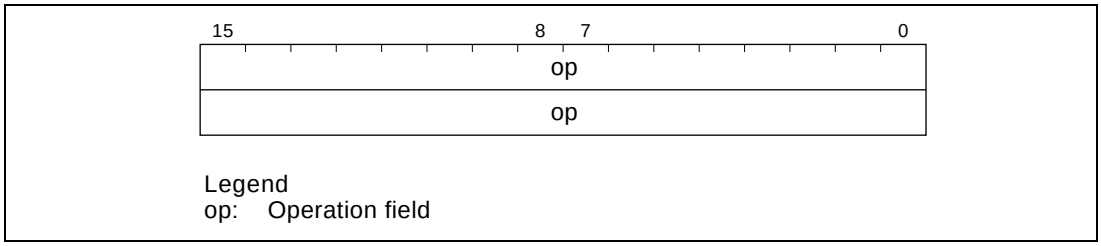
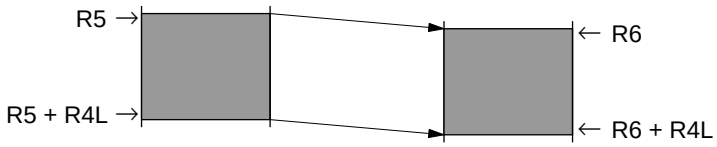


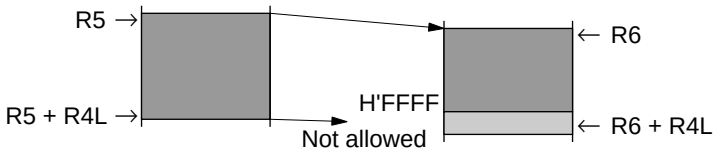
Figure 2-10 Block Data Transfer Instruction

Notes on EEPMOV Instruction

1. The EEPMOV instruction is a block data transfer instruction. It moves the number of bytes specified by R4L from the address specified by R5 to the address specified by R6.



2. When setting R4L and R6, make sure that the final destination address (R6 + R4L) does not exceed H'FFFF. The value in R6 must not change from H'FFFF to H'0000 during execution of the instruction



2.6 CPU States

2.6.1 Overview

The CPU has three states: the program execution state, exception-handling state, and power-down state. The power-down state is further divided into three modes: sleep mode, software standby mode, and hardware standby mode. Figure 2-11 summarizes these states, and figure 2-12 shows a map of the state transitions.

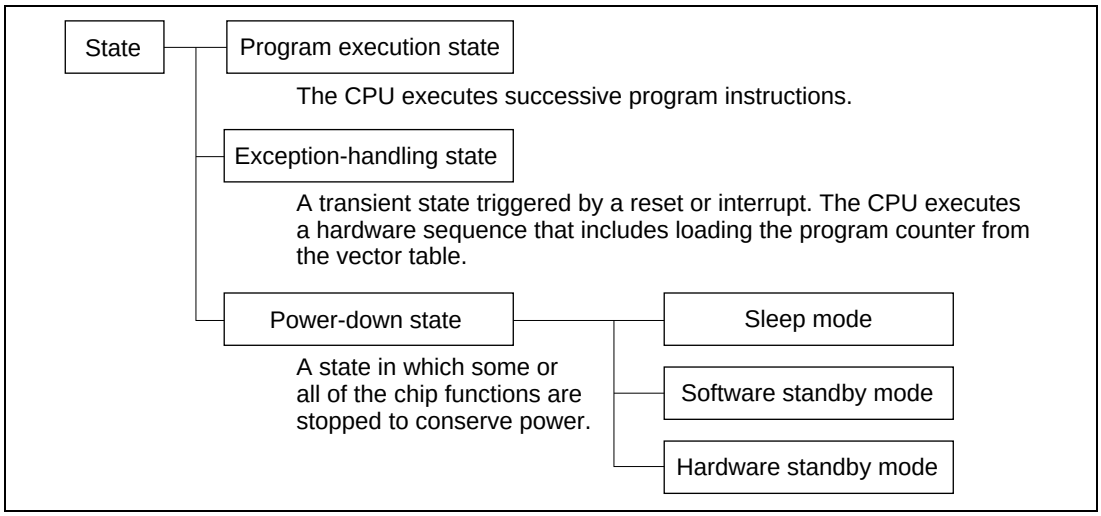
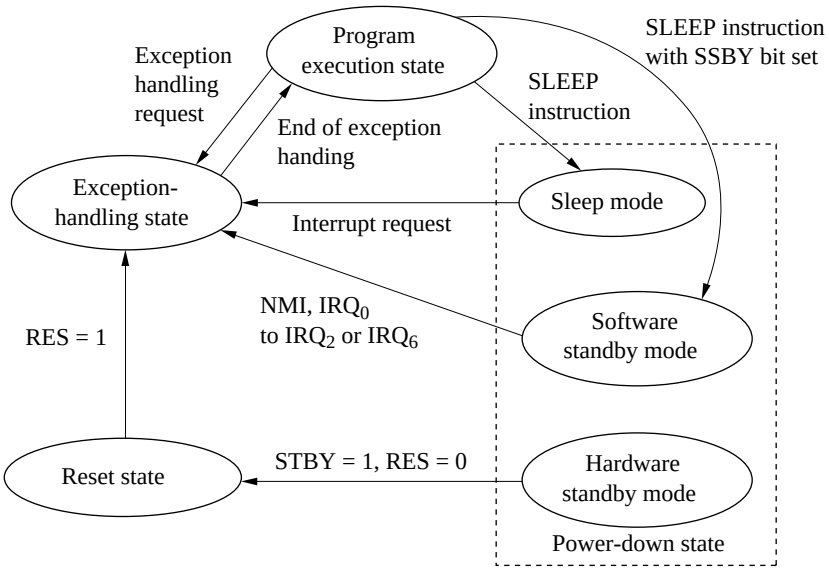


Figure 2-11 Operating States



Notes: 1. A transition to the reset state occurs when RES goes low, except when the chip is in the hardware standby mode.
2. A transition from any state to the hardware standby mode occurs when STBY goes low.

Figure 2-12 State Transitions

2.6.2 Program Execution State

In this state the CPU executes program instructions.

2.6.3 Exception-Handling State

The exception-handling state is a transient state that occurs when the CPU is reset or interrupted and changes its normal processing flow. In interrupt exception handling, the CPU references the stack pointer (R7) and saves the program counter and condition code register on the stack. For further details see section 4, Exception Handling.

2.6.4 Power-Down State

The power-down state includes three modes: sleep mode, software standby mode, and hardware standby mode.

(1) Sleep Mode: Is entered when a SLEEP instruction is executed. The CPU halts, but CPU register contents remain unchanged and the on-chip supporting modules continue to function.

(2) Software Standby Mode: Is entered if the SLEEP instruction is executed while the SSBY (Software Standby) bit in the system control register (SYSCR) is set. The CPU and all on-chip supporting modules halt. The on-chip supporting modules are initialized, but the contents of the on-chip RAM and CPU registers remain unchanged as long as a specified voltage is supplied. I/O port outputs also remain unchanged.

(3) Hardware Standby Mode: Is entered when the input at the $\overline{\text{STBY}}$ pin goes low. All chip functions halt, including I/O port output. The on-chip supporting modules are initialized, but on-chip RAM contents are held.

See section 19, Power-Down State, for further information.

2.7 Access Timing and Bus Cycle

The CPU is driven by the system clock (ϕ). The period from one rising edge of the system clock to the next is referred to as a "state." Memory access is performed in a two- or three-state bus cycle. On-chip memory, on-chip supporting modules, and external devices are accessed in different bus cycles as described below.

2.7.1 Access to On-Chip Memory (RAM and ROM)

On-chip ROM and RAM are accessed in a cycle of two states designated T_1 and T_2. Either byte or word data can be accessed, via a 16-bit data bus. Figure 2-13 shows the on-chip memory access cycle. Figure 2-14 shows the associated pin states.

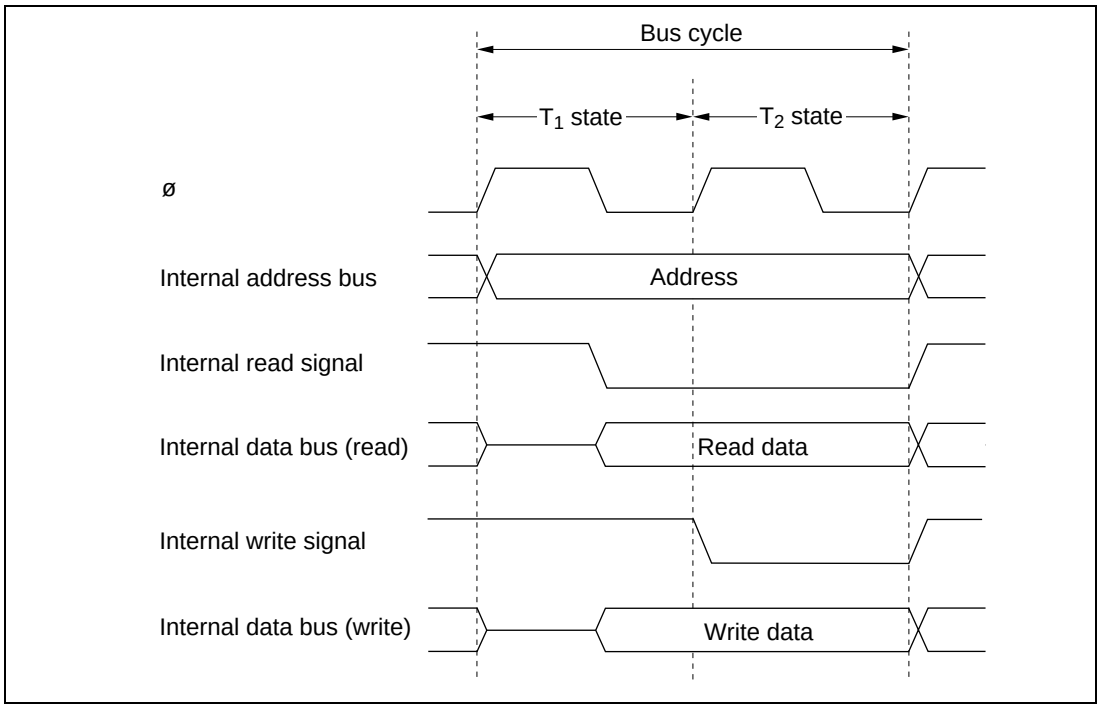


Figure 2-13 On-Chip Memory Access Cycle

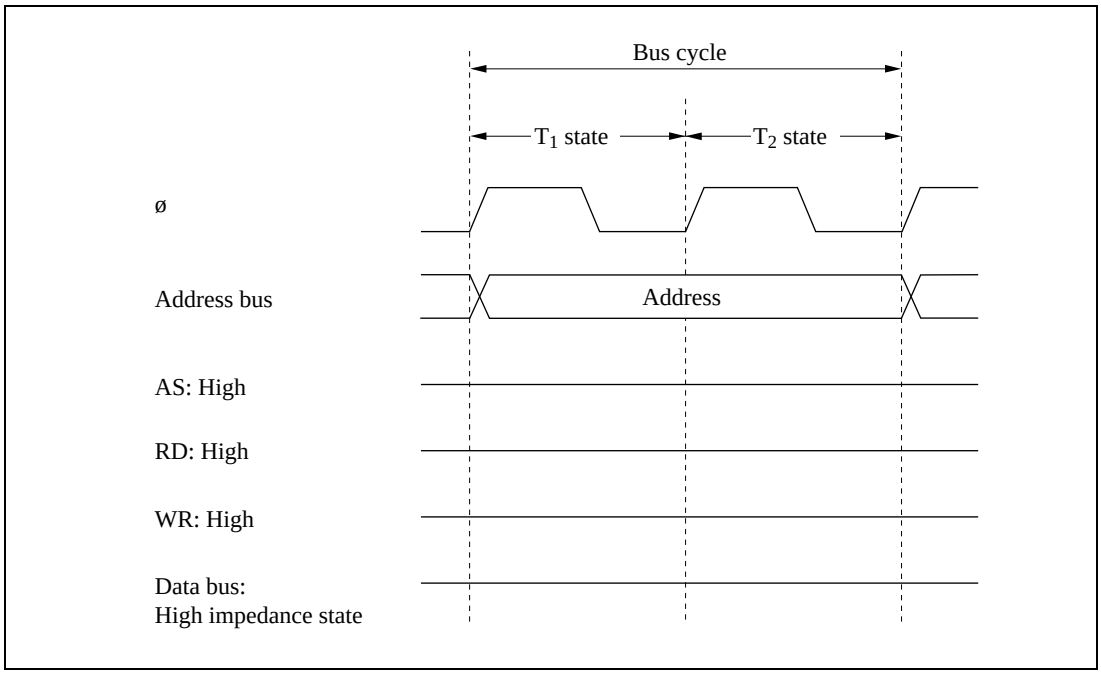


Figure 2-14 Pin States during On-Chip Memory Access Cycle

2.7.2 Access to On-Chip Supporting Modules and External Devices

The on-chip supporting module registers and external devices are accessed in a cycle consisting of three states: T_1, T_2, and T_3. Only one byte of data can be accessed per cycle, via an 8-bit data bus. Access to word data or instruction codes requires two consecutive cycles (six states).

Figure 2-15 shows the access cycle for the on-chip supporting modules. Figure 2-16 shows the associated pin states. Figures 2-17 (a) and (b) show the read and write access timing for external devices.

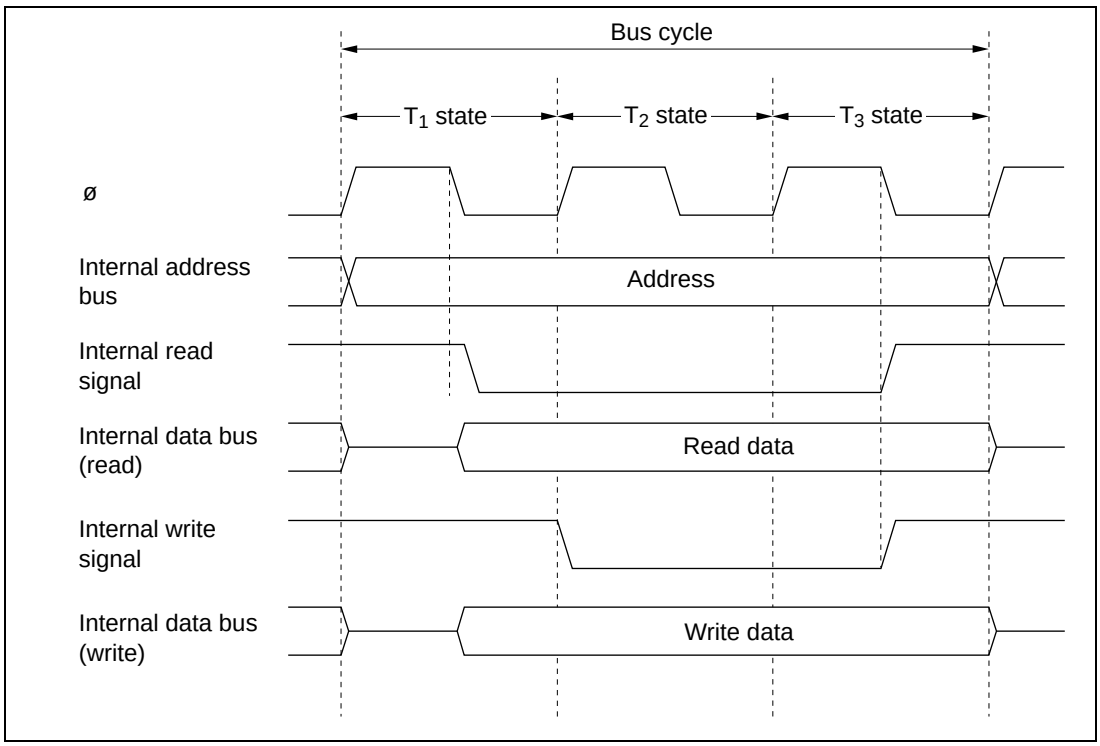


Figure 2-15 On-Chip Supporting Module Access Cycle

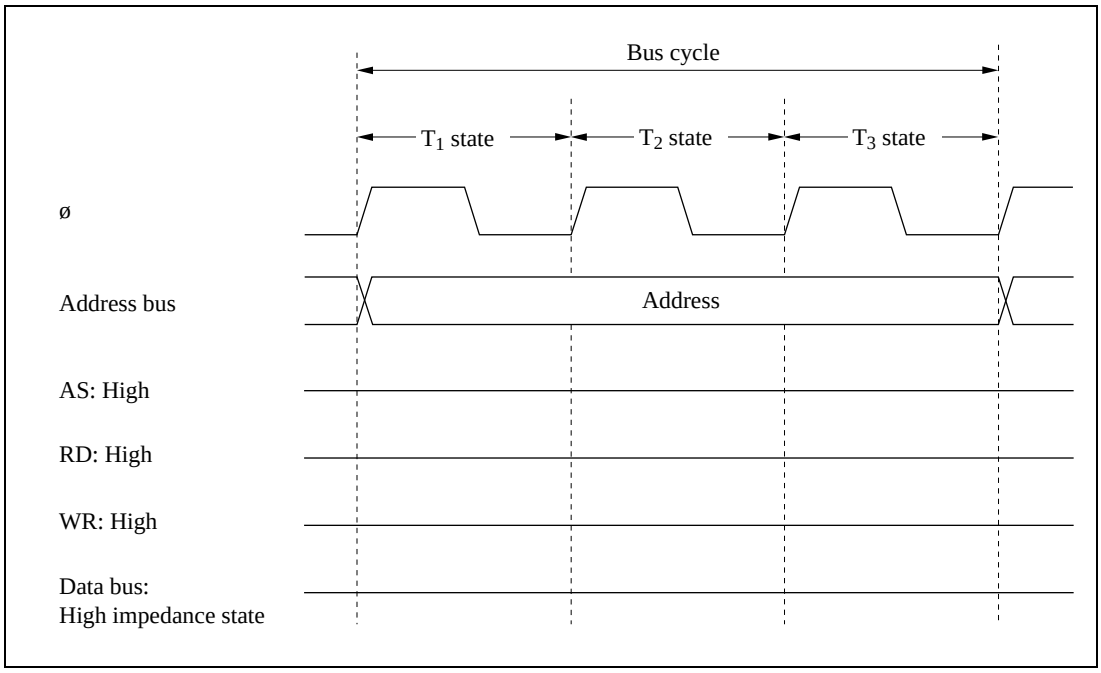


Figure 2-16 Pin States during On-Chip Supporting Module Access Cycle

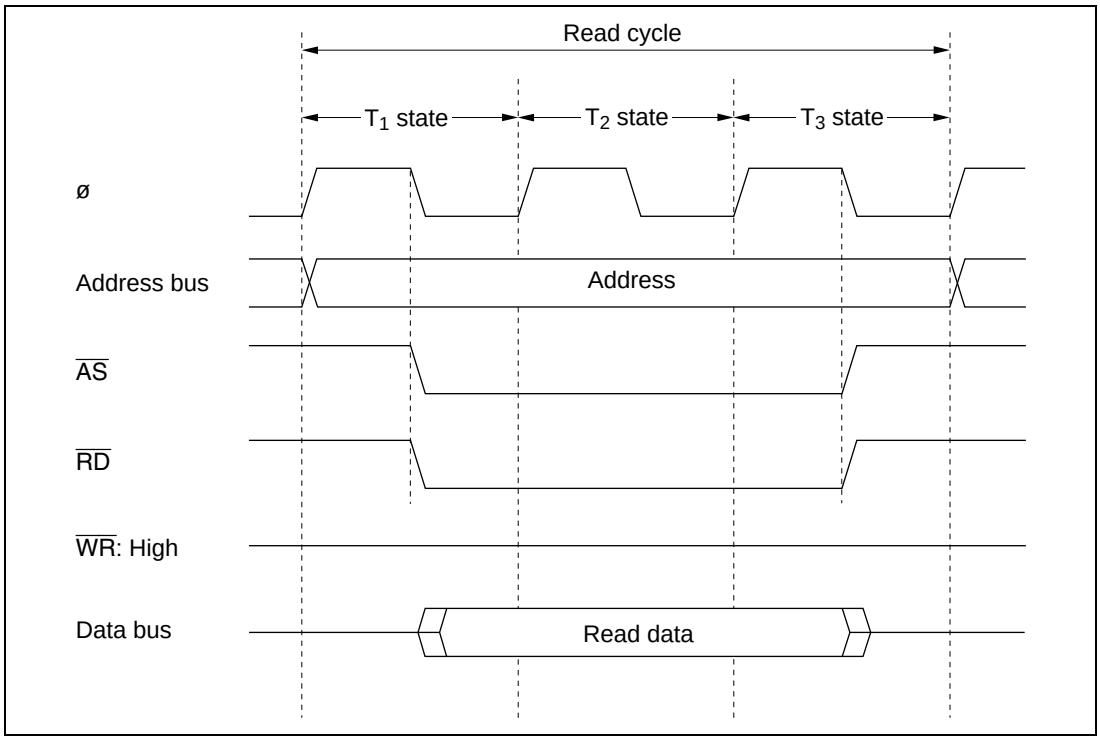


Figure 2-17 (a) External Device Access Timing (Read)

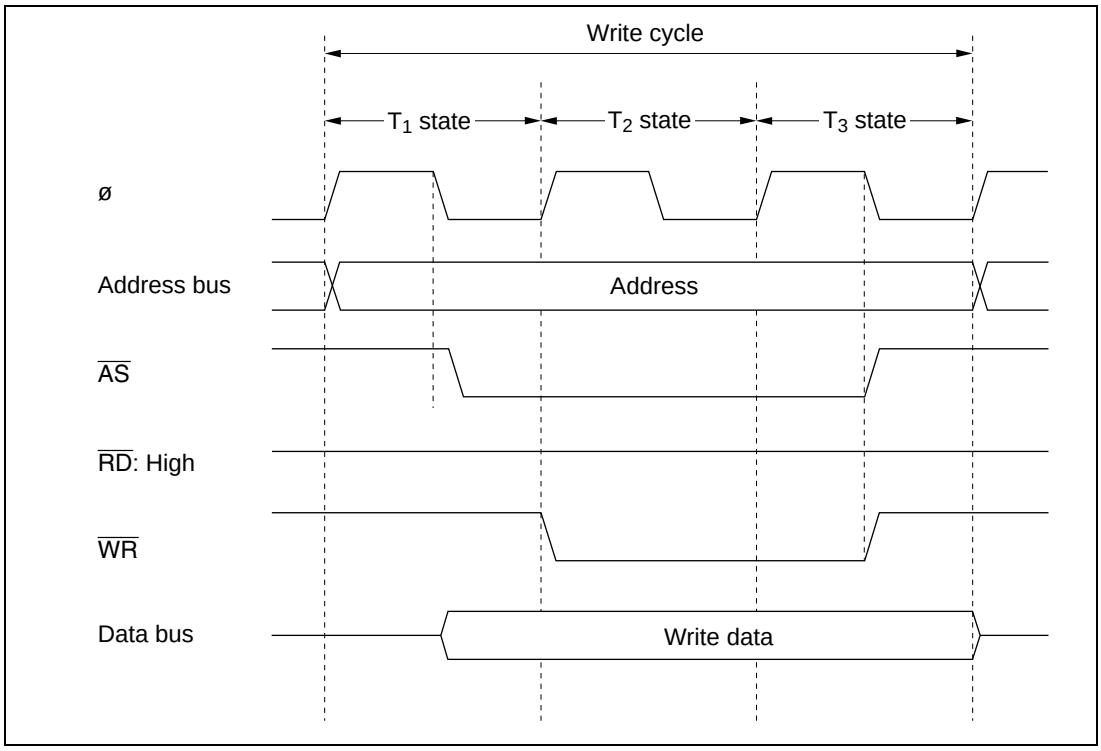


Figure 2-17 (b) External Device Access Timing (Write)

Section 3 MCU Operating Modes and Address Space

3.1 Overview

3.1.1 Mode Selection

The H8/3534 and H8/3522 operate in three modes numbered 1, 2, and 3. The mode is selected by the inputs at the mode pins (MD_1 and MD_0). See table 3-1.

Table 3-1 Operating Modes

Mode	MD_1	MD_0	Address space	On-chip ROM	On-chip RAM
Mode 0	Low	Low	—	—	—
Mode 1	Low	High	Expanded	Disabled	Enabled*
Mode 2	High	Low	Expanded	Enabled	Enabled*
Mode 3	High	High	Single-chip	Enabled	Enabled

Note: * If the RAME bit in the system control register (SYSCR) is cleared to 0, off-chip memory can be accessed instead.

Modes 1 and 2 are expanded modes that permit access to off-chip memory and peripheral devices. The maximum address space supported by these externally expanded modes is 64 kbytes.

In mode 3 (single-chip mode), only on-chip ROM and RAM and the on-chip register field are used. All ports are available for general-purpose input and output.

Mode 0 is inoperative in the H8/3534 and H8/3522. Avoid setting the mode pins to mode 0.

3.1.2 Mode and System Control Registers

Table 3-2 lists the registers related to the chip's operating mode: the system control register (SYSCR) and mode control register (MDCR). The mode control register indicates the inputs to the mode pins MD_1 and MD_0.

Table 3-2 Mode and System Control Registers

Name	Abbreviation	Read/Write	Address
System control register	SYSCR	R/W	H'FFC4
Mode control register	MDCR	R	H'FFC5

3.2 System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
[H8/3534]	SSBY	STS2	STS1	STS0	XRST	NMIEG	(HIE)	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
[H8/3522]	SSBY	STS2	STS1	STS0	XRST	NMIEG	—	RAME
Initial value	0	0	0	0	1	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	—	R/W

The system control register (SYSCR) is an 8-bit register that controls the operation of the chip.

Bit 7—Software Standby (SSBY): Enables transition to the software standby mode. For details, see section 16, Power-Down State.

On recovery from software standby mode by an external interrupt, the SSBY bit remains set to 1. It can be cleared by writing 0.

Bit 7 SSBY	Description	
0	The SLEEP instruction causes a transition to sleep mode.	(Initial value)
1	The SLEEP instruction causes a transition to software standby mode.	

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the clock settling time when the chip recovers from the software standby mode by an external interrupt. During the selected time the CPU and on-chip supporting modules continue to stand by. These bits should be set according to the clock frequency so that the settling time is at least 8 ms. For specific settings, see section 16.3.3, Clock Settling Time for Exit from Software Standby Mode.

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description	
0	0	0	Settling time = 8,192 states	(Initial value)
0	0	1	Settling time = 16,384 states	
0	1	0	Settling time = 32,768 states	
0	1	1	Settling time = 65,536 states	
1	0	—	Settling time = 131,072 states	
1	1	—	Unused	

Bit 3—External Reset (XRST): Indicates the source of a reset. A reset can be generated by input of an external reset signal, or by a watchdog timer overflow when the watchdog timer is used. 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	Reset was caused by watchdog timer overflow.	
1	Reset was caused by external input.	(Initial value)

Bit 2—NMI Edge (NMIEG): Selects the valid edge of the $\overline{\text{NMI}}$ input.

Bit 2 NMIEG	Description	
0	An interrupt is requested on the falling edge of the $\overline{\text{NMI}}$ input.	(Initial value)
1	An interrupt is requested on the rising edge of the $\overline{\text{NMI}}$ input.	

Bit 1—Host Interface Enable (HIE): [H8/3534] Reserved. Do not set this bit to 1.

Bit 1—Reserved: [H8/3522] This bit cannot be modified and is always read as 1.

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized by a reset, but is not initialized in the software standby mode.

Bit 0 RAME	Description	
0	The on-chip RAM is disabled.	
1	The on-chip RAM is enabled.	(Initial value)

3.3 Mode Control Register (MDCR)

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

Note: * Initialized according to MD_1 and MD_0 inputs.

The mode control register (MDCR) is an 8-bit register that indicates the operating mode of the chip.

Bits 7 to 5—Reserved: These bits cannot be modified and are always read as 1.

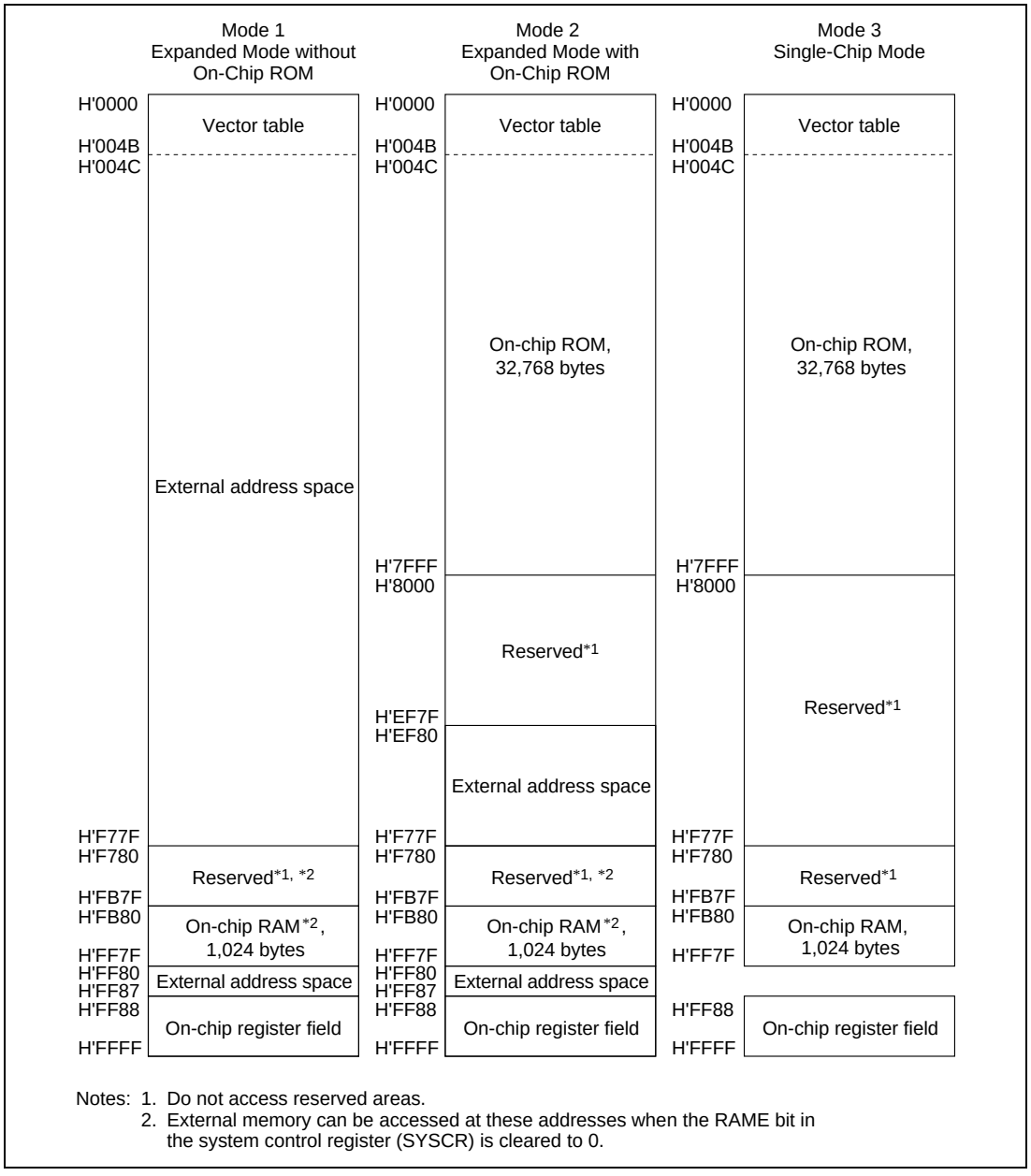
Bits 4 and 3—Reserved: These bits cannot be modified and are always read as 0.

Bit 2—Reserved: This bit cannot be modified and is always read as 1.

Bits 1 and 0—Mode Select 1 and 0 (MDS1 and MDS0): These bits indicate the values of the mode pins (MD_1 and MD_0), thereby indicating the current operating mode of the chip. MDS1 corresponds to MD_1 and MDS0 to MD_0. These bits can be read but not written. When the mode control register is read, the levels at the mode pins (MD_1 and MD_0) are latched in these bits.

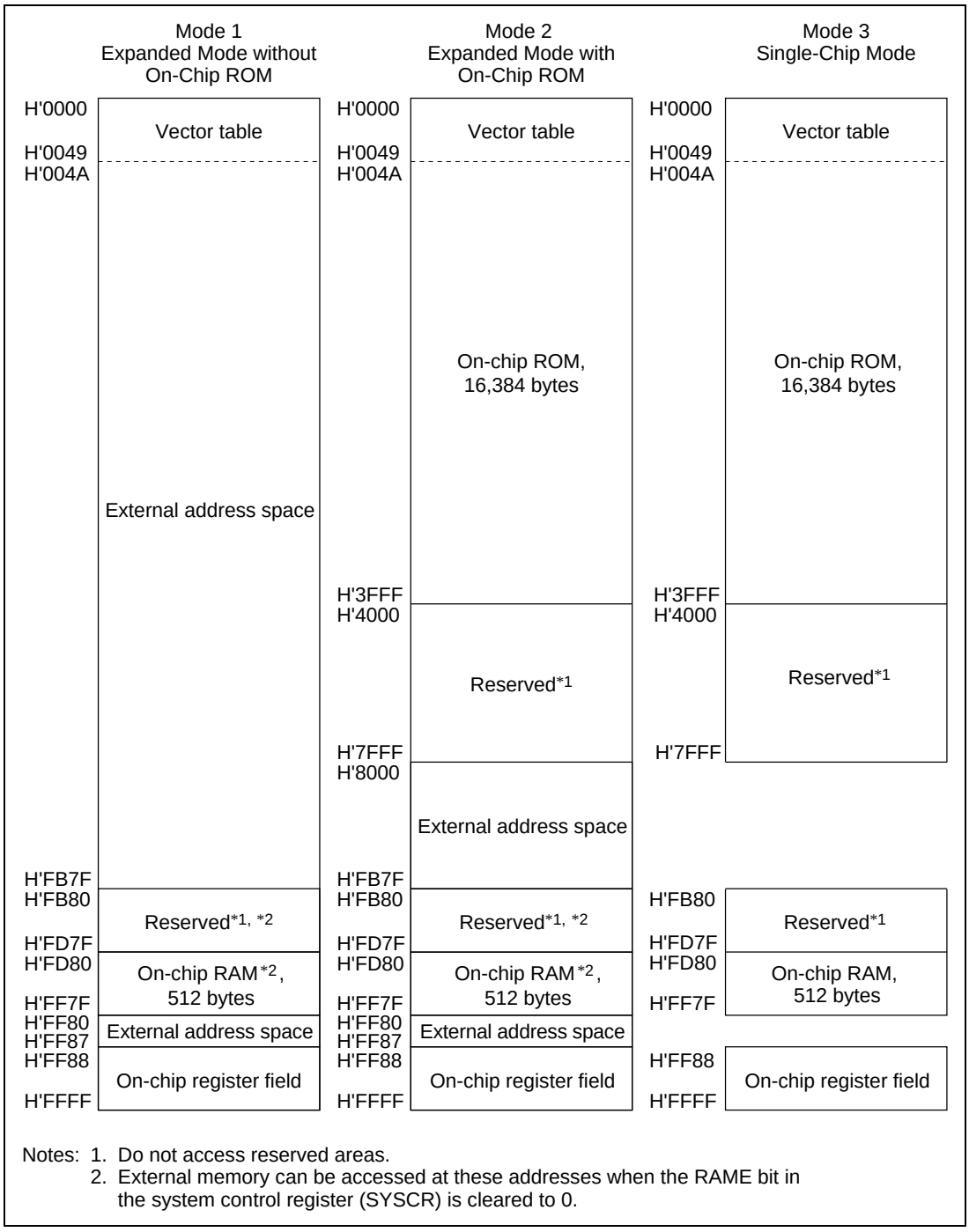
3.4 Address Space Map in Each Operating Mode

Figures 3-1, 3-2 show memory maps of the H8/3534, H8/3522 in modes 1, 2, and 3.



Notes: 1. Do not access reserved areas.
2. External memory can be accessed at these addresses when the RAME bit in the system control register (SYSCR) is cleared to 0.

Figure 3-1 H8/3534 Address Space Map



Notes: 1. Do not access reserved areas.
2. External memory can be accessed at these addresses when the RAME bit in the system control register (SYSCR) is cleared to 0.

Figure 3-2 H8/3522 Address Space Map

Section 4 Exception Handling

4.1 Overview

The H8/3534 and H8/3522 recognize two kinds of exceptions: interrupts and the reset. Table 4-1 indicates their priority and the timing of their hardware exception-handling sequence.

Table 4-1 Hardware Exception-Handling Sequences and Priority

Priority	Type of Exception	Detection Timing	Timing of Exception-Handling Sequence
High	Reset	Synchronized with clock	The hardware exception-handling sequence begins as soon as $\overline{\text{RES}}$ changes from low to high.
↓	Interrupt	End of instruction execution*	When an interrupt is requested, the hardware exception-handling sequence begins at the end of the current instruction, or at the end of the current hardware exception-handling sequence.
Low			

Note: * Not detected after ANDC, ORC, XORC, and LDC instructions.

4.2 Reset

4.2.1 Overview

A reset has the highest exception-handling priority. When the $\overline{\text{RES}}$ pin goes low or when there is a watchdog timer reset (when the reset option is selected for watchdog timer overflow), all current processing stops and the chip enters the reset state. The internal state of the CPU and the registers of the on-chip supporting modules are initialized. The reset exception-handling sequence starts when $\overline{\text{RES}}$ returns from low to high, or at the end of a watchdog reset pulse.

4.2.2 Reset Sequence

The reset state begins when $\overline{\text{RES}}$ goes low or a watchdog reset is generated. To ensure correct resetting, at power-on the $\overline{\text{RES}}$ pin should be held low for at least 20 ms. In a reset during operation, the $\overline{\text{RES}}$ pin should be held low for at least 10 system clock cycles. The watchdog reset pulse width is always 518 system clocks. For the pin states during a reset, see appendix D, Port States in Each Mode.

The following sequence is carried out when reset exception handling begins.

(1) The internal state of the CPU and the registers of the on-chip supporting modules are initialized, and the I bit in the condition code register (CCR) is set to 1.
(2) The CPU loads the program counter with the first word in the vector table (stored at addresses H'0000 and H'0001) and starts program execution.

The $\overline{\text{RES}}$ pin should be held low when power is switched off, as well as when power is switched on.

Figure 4-1 indicates the timing of the reset sequence in modes 2 and 3. Figure 4-2 indicates the timing in mode 1.

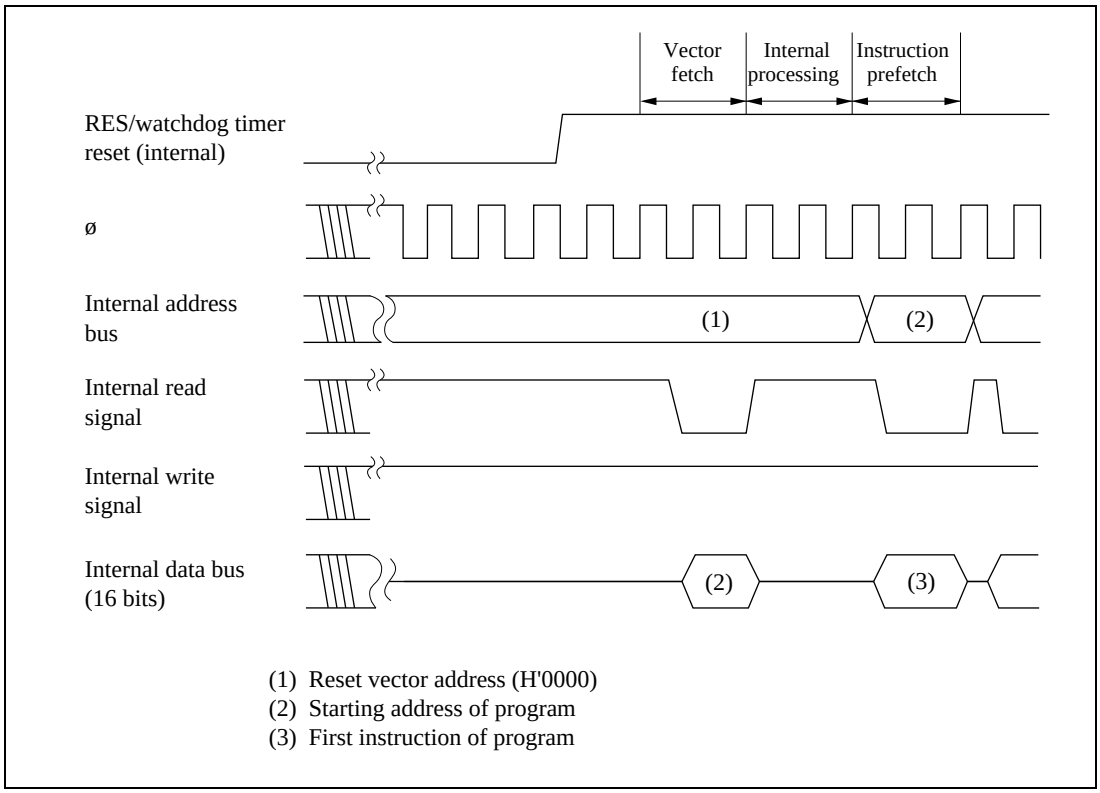
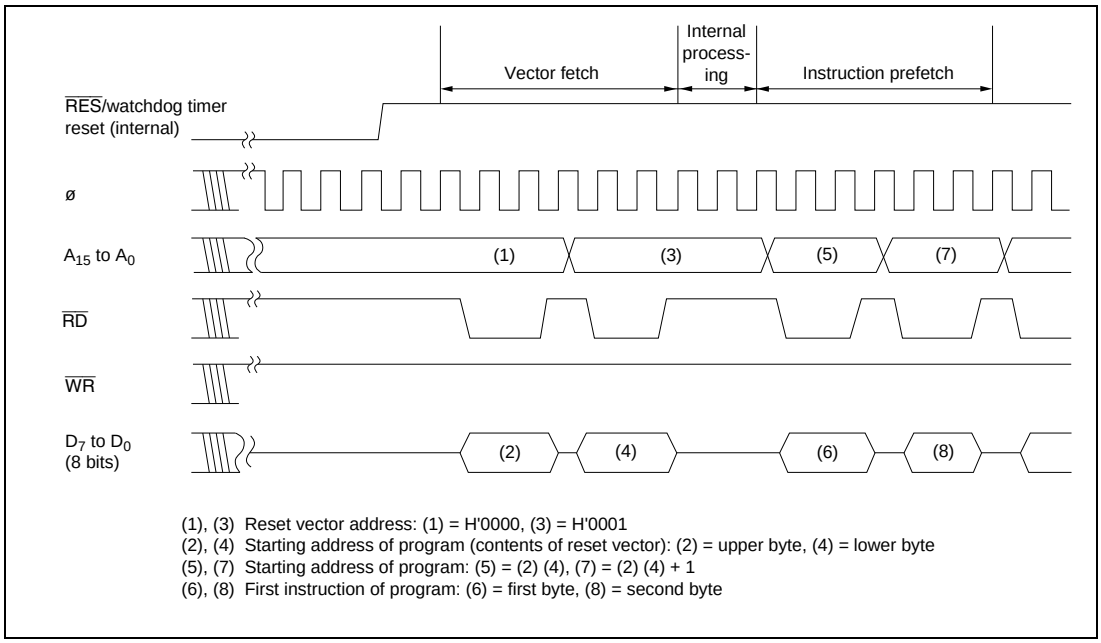


Figure 4-1 Reset Sequence (Mode 2 or 3, Program Stored in On-Chip ROM)



(1), (3) Reset vector address: (1) = H'0000, (3) = H'0001
(2), (4) Starting address of program (contents of reset vector): (2) = upper byte, (4) = lower byte
(5), (7) Starting address of program: (5) = (2) (4), (7) = (2) (4) + 1
(6), (8) First instruction of program: (6) = first byte, (8) = second byte

Figure 4-2 Reset Sequence (Mode 1)

4.2.3 Disabling of Interrupts after Reset

After a reset, if an interrupt were to be accepted before initialization of the stack pointer (SP: R7), the program counter and condition code register might not be saved correctly, leading to a program crash. To prevent this, all interrupts, including NMI, are disabled immediately after a reset. The first program instruction is therefore always executed. This instruction should initialize the stack pointer (example: MOV.W #xx:16, SP).

After reset exception handling, in order to initialize the contents of CCR, a CCR manipulation instruction can be executed before an instruction to initialize the stack pointer. Immediately after execution of a CCR manipulation instruction, all interrupts including NMI are disabled. Use the next instruction to initialize the stack pointer.

4.3 Interrupts

4.3.1 Overview

The interrupt sources for the H8/3534 include nine external sources from 23 input pins (NMI, IRQ_0 to IRQ_7, and $KEYIN_0$ to $KEYIN_7$), and 23 internal sources in the on-chip supporting modules. The H8/3522 has four external sources (NMI, and IRQ_0 to IRQ_2) and 19 internal sources in the on-chip supporting modules. Table 4-2 lists the interrupt sources in priority order and gives their vector addresses. When two or more interrupts are requested, the interrupt with highest priority is served first.

The features of these interrupts are:

- NMI has the highest priority and is always accepted. All internal and external interrupts except NMI can be masked by the I bit in the CCR. When the I bit is set to 1, interrupts other than NMI are not accepted.
- IRQ_0 to IRQ_7 [H8/3534]/IRQ_0 to IRQ_2 [H8/3522] can be sensed on the falling edge of the input signal, or level-sensed. The type of sensing can be selected for each interrupt individually. NMI is edge-sensed, and either the rising or falling edge can be selected.
- All interrupts are individually vectored. The software interrupt-handling routine does not have to determine what type of interrupt has occurred.
- IRQ_6 [H8/3534] is multiplexed with 8 external sources ($KEYIN_0$ to $KEYIN_7$). $KEYIN_0$ to $KEYIN_7$ can be masked individually by user software.
- The watchdog timer can generate either an NMI or overflow interrupt, depending on the needs of the application. For details, see section 11, Watchdog Timer.

Table 4-2 (a) H8/3534 Interrupts

Interrupt source		No.	Vector Table Address	Priority
NMI		3	H'0006 to H'0007	High
IRQ_0		4	H'0008 to H'0009	
IRQ_1		5	H'000A to H'000B	
IRQ_2		6	H'000C to H'000D	
IRQ_3		7	H'000E to H'000F	
IRQ_4		8	H'0010 to H'0011	
IRQ_5		9	H'0012 to H'0013	
IRQ_6		10	H'0014 to H'0015	
IRQ_7		11	H'0016 to H'0017	
16-bit free-running timer	ICIA (Input capture A)	12	H'0018 to H'0019	
	ICIB (Input capture B)	13	H'001A to H'001B	
	ICIC (Input capture C)	14	H'001C to H'001D	
	ICID (Input capture D)	15	H'001E to H'001F	
	OCIA (Output compare A)	16	H'0020 to H'0021	
	OCIB (Output compare B)	17	H'0022 to H'0023	
	FOVI (Overflow)	18	H'0024 to H'0025	
8-bit timer 0	CMI0A (Compare-match A)	19	H'0026 to H'0027	↓
	CMI0B (Compare-match B)	20	H'0028 to H'0029	
	OVI0 (Overflow)	21	H'002A to H'002B	
8-bit timer 1	CMI1A (Compare-match A)	22	H'002C to H'002D	
	CMI1B (Compare-match B)	23	H'002E to H'002F	
	OVI1 (Overflow)	24	H'0030 to H'0031	
Reserved		25	H'0032 to H'0033	
		26	H'0034 to H'0035	
Serial communication interface 0	ERI0 (Receive error)	27	H'0036 to H'0037	
	RXI0 (Receive end)	28	H'0038 to H'0039	
	TXI0 (TDR empty)	29	H'003A to H'003B	
	TEI0 (TSR empty)	30	H'003C to H'003D	
Serial communication interface 1	ERI1 (Receive error)	31	H'003E to H'003F	
	RXI1 (Receive end)	32	H'0040 to H'0041	
	TXI1 (TDR empty)	33	H'0042 to H'0043	
	TEI1 (TSR empty)	34	H'0044 to H'0045	
A/D converter	ADI (Conversion end)	35	H'0046 to H'0047	
Watchdog timer	WOVF (WDT overflow)	36	H'0048 to H'0049	Low

Notes: 1. H'0000 and H'0001 contain the reset vector.
2. H'0002 to H'0005 are reserved in the H8/3534 and are not available to the user.

Table 4-2 (b) H8/3522 Interrupts

Interrupt source		No.	Vector Table Address	Priority
NMI		3	H'0006 to H'0007	High
IRQ_0		4	H'0008 to H'0009	
IRQ_1		5	H'000A to H'000B	
IRQ_2		6	H'000C to H'000D	
Reserved		7	H'000E to H'000F	
		8	H'0010 to H'0011	
		9	H'0012 to H'0013	
		10	H'0014 to H'0015	
		11	H'0016 to H'0017	
16-bit free-running timer	ICIA (Input capture A)	12	H'0018 to H'0019	
	ICIB (Input capture B)	13	H'001A to H'001B	
	ICIC (Input capture C)	14	H'001C to H'001D	
	ICID (Input capture D)	15	H'001E to H'001F	
	OCIA (Output compare A)	16	H'0020 to H'0021	
	OCIB (Output compare B)	17	H'0022 to H'0023	
	FOVI (Overflow)	18	H'0024 to H'0025	
8-bit timer 0	CMI0A (Compare-match A)	19	H'0026 to H'0027	↓
	CMI0B (Compare-match B)	20	H'0028 to H'0029	
	OVI0 (Overflow)	21	H'002A to H'002B	
8-bit timer 1	CMI1A (Compare-match A)	22	H'002C to H'002D	
	CMI1B (Compare-match B)	23	H'002E to H'002F	
	OVI1 (Overflow)	24	H'0030 to H'0031	
Reserved		25	H'0032 to H'0033	
		26	H'0034 to H'0035	
Serial communication interface	ERI (Receive error)	27	H'0036 to H'0037	
	RXI (Receive end)	28	H'0038 to H'0039	
	TXI (TDR empty)	29	H'003A to H'003B	
	TEI (TSR empty)	30	H'003C to H'003D	
Reserved		31	H'003E to H'003F	
		32	H'0040 to H'0041	
		33	H'0042 to H'0043	
		34	H'0044 to H'0045	
A/D converter	ADI (Conversion end)	35	H'0046 to H'0047	
Watchdog timer	WOVF (WDT overflow)	36	H'0048 to H'0049	Low

Notes: 1. H'0000 and H'0001 contain the reset vector.
2. H'0002 to H'0005 are reserved in the H8/3522 and are not available to the user.

4.3.2 Interrupt-Related Registers

The interrupt-related registers are the system control register (SYSCR), IRQ sense control register (ISCR), IRQ enable register (IER), and keyboard matrix interrupt mask register (KMIMR).

Table 4-3 Registers Read by Interrupt Controller

Name	Abbreviation	Read/write	Address
System control register	SYSCR	R/W	H'FFC4
IRQ sense control register	ISCR	R/W	H'FFC6
IRQ enable register	IER	R/W	H'FFC7
Keyboard matrix interrupt mask register	KMIMR	R/W	H'FFF1

System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
[H8/3534]	SSBY	STS2	STS1	STS0	XRST	NMIEG	(HIE)	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
[H8/3522]	SSBY	STS2	STS1	STS0	XRST	NMIEG	—	RAME
Initial value	0	0	0	0	1	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	—	R/W

The valid edge on the $\overline{\text{NMI}}$ line is controlled by bit 2 (NMIEG) in the system control register.

Bit 2—NMI Edge (NMIEG): Determines whether a nonmaskable interrupt is generated on the falling or rising edge of the $\overline{\text{NMI}}$ input signal.

Bit 2 NMIEG	Description	
0	An interrupt is generated on the falling edge of $\overline{\text{NMI}}$.	(Initial value)
1	An interrupt is generated on the rising edge of $\overline{\text{NMI}}$.	

See section 3.2, System Control Register, for information on the other SYSCR bits.

IRQ Sense Control Register (ISCR)

Bit	7	6	5	4	3	2	1	0
[H8/3534]	IRQ7SC	IRQ6SC	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC
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
[H8/3522]	—	—	—	—	—	IRQ2SC	IRQ1SC	IRQ0SC
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

[H8/3534]

Bits 0 to 7—IRQ0 to IRQ7 Sense Control (IRQ0SC to IRQ7SC): These bits determine whether $\overline{IRQ_0}$ to $\overline{IRQ_7}$ are level-sensed or sensed on the falling edge.

Bits 0 to 7 IRQ0SC to IRQ7SC	Description
0	An interrupt is generated when $\overline{IRQ_0}$ to $\overline{IRQ_7}$ inputs are low. (Initial value)
1	An interrupt is generated by the falling edge of the $\overline{IRQ_0}$ to $\overline{IRQ_7}$ inputs.

[H8/3522]

Bits 3 to 7—Reserved: These bits cannot be modified and are always read as 1.

Bits 0 to 2—IRQ0 to IRQ2 Sense Control (IRQ0SC to IRQ2SC): These bits determine whether $\overline{IRQ_0}$ to $\overline{IRQ_2}$ are level-sensed or sensed on the falling edge.

Bits 0 to 2 IRQ0SC to IRQ2SC	Description
0	An interrupt is generated when $\overline{IRQ_0}$ to $\overline{IRQ_2}$ inputs are low. (Initial value)
1	An interrupt is generated by the falling edge of the $\overline{IRQ_0}$ to $\overline{IRQ_2}$ inputs.

IRQ Enable Register (IER)

Bit	7	6	5	4	3	2	1	0
[H8/3534]	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
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
[H8/3522]	—	—	—	—	—	IRQ2E	IRQ1E	IRQ0E
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

[H8/3534]

Bits 0 to 7—IRQ0 to IRQ7 Enable (IRQ0E to IRQ7E): These bits enable or disable the IRQ_0 to IRQ_7 interrupts individually.

Bits 0 to 7 IRQ0E to IRQ7E	Description	
0	IRQ_0 to IRQ_7 interrupt requests are disabled.	(Initial value)
1	IRQ_0 to IRQ_7 interrupt requests are enabled.	

[H8/3522]

Bits 3 to 7—Reserved: These bits cannot be modified and are always read as 1.

Bits 0 to 2—IRQ0 to IRQ2 Enable (IRQ0E to IRQ2E): These bits enable or disable the IRQ_0 to IRQ_2 interrupts individually.

Bits 0 to 2 IRQ0E to IRQ2E	Description	
0	IRQ_0 to IRQ_2 interrupt requests are disabled.	(Initial value)
1	IRQ_0 to IRQ_2 interrupt requests are enabled.	

When edge sensing is selected (by setting bits IRQ0SC to IRQ7SC* to 1), it is possible for an interrupt-handling routine to be executed even though the corresponding enable bit (IRQ0E to IRQ7E*) is cleared to 0 and the interrupt is disabled. If an interrupt is requested while the enable bit (IRQ0E to IRQ7E) is set to 1, the request will be held pending until served. If the enable bit is cleared to 0 while the request is still pending, the request will remain pending, although new requests will not be recognized. If the interrupt mask bit (I) in the CCR is cleared to 0, the interrupt-handling routine can be executed even though the enable bit is now 0.

If execution of interrupt-handling routines under these conditions is not desired, it can be avoided by using the following procedure to disable and clear interrupt requests.

1. Set the I bit to 1 in the CCR, masking interrupts. Note that the I bit is set to 1 automatically when execution jumps to an interrupt vector.
2. Clear the desired bits from IRQ0E to IRQ7E to 0 to disable new interrupt requests.
3. Clear the corresponding IRQ0SC to IRQ7SC bits to 0, then set them to 1 again. Pending IRQn interrupt requests are cleared when I = 1 in the CCR, IRQnSC = 0, and IRQnE = 0.

Note: * For the H8/3522, read "IRQ0SC to IRQ2SC bits " and "IRQ0E to IRQ2E bits", respectively.

Keyboard Matrix Interrupt Mask Register (KMIMR) [H8/3534 only]

KMIMR is provided as a register for keyboard matrix interrupt masking. This register controls interrupts from the $KEYIN_0$ to $KEYIN_7$ key sense input pins for a 16×8 matrix keyboard.

Bits KMIMR0 to KMIMR7 of KMIMR correspond to key sense inputs $KEYIN_0$ to $KEYIN_7$.

In interrupt mask bit initialization, bit KMIMR6 corresponding to the IRQ_6/$KEYIN_6$ pin is set to enable interrupt requests, while the other mask bits are set to disable interrupts.

KMIMR is an 8-bit readable/writable register used in keyboard matrix scan/sense. This register initializes to a state in which only the input at the IRQ_6 pin is enabled. To enable key sense input interrupts from two or more pins in keyboard matrix scanning and sensing, clear the corresponding mask bits to 0.

Bit	7	6	5	4	3	2	1	0
[H8/3534]	KMIMR7	KMIMR6	KMIMR5	KMIMR4	KMIMR3	KMIMR2	KMIMR1	KMIMR0
Initial value	1	0	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 0 to 7—Keyboard Matrix Interrupt Mask (KMIMR7 to KMIMR0): These bits control key sense input interrupt requests $KEYIN_7$ to $KEYIN_0$.

Bits 0 to 7 KMIMR0 to KMIMR7	Description	
0	Key sense input interrupt request is enabled.	
1	Key sense input interrupt request is disabled.	(Initial value)*

Note: * Except KMIMR6, which is initially 0.

Figure 4-3 shows the relationship between the IRQ_6 interrupt and KMIMR.

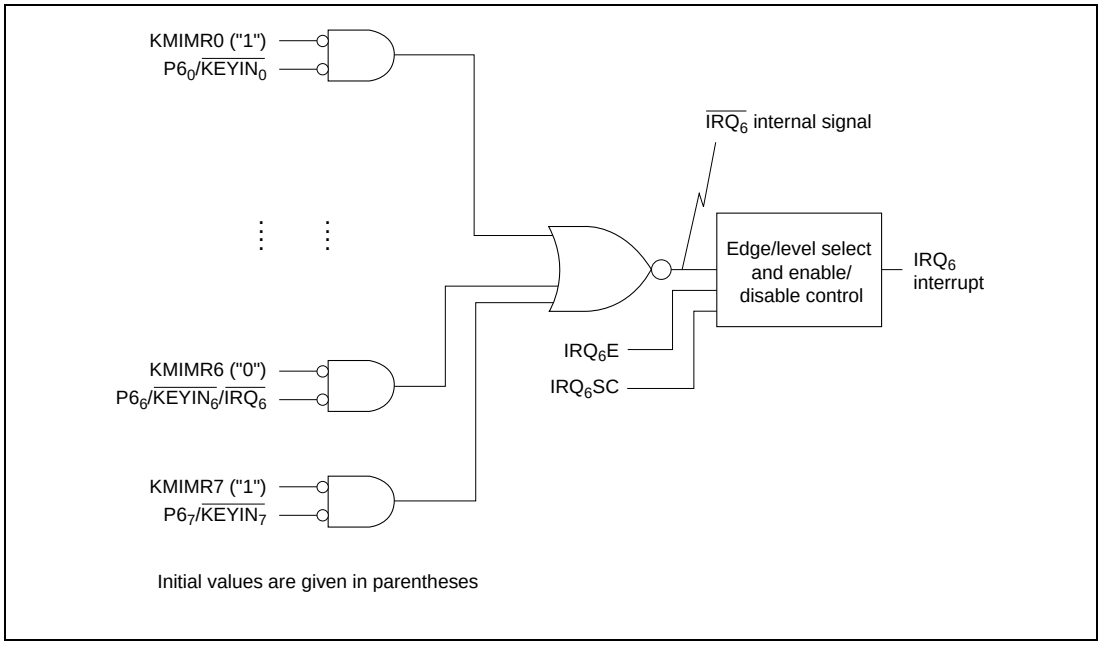


Figure 4-3 KMIMR and IRQ_6 Interrupt

4.3.3 External Interrupts

NMI, IRQ_0, IRQ_1, IRQ_2, and IRQ_6 can be used to recover from software standby mode.

(1) NMI: A nonmaskable interrupt is generated on the rising or falling edge of the $\overline{NMI}$ input signal regardless of whether the I (interrupt mask) bit is set in the CCR. The valid edge is selected by the NMIEG bit in the system control register. The NMI vector number is 3. In the NMI hardware exception-handling sequence the I bit in the CCR is set to 1.

(2) IRQ_0 to IRQ_7*: These interrupt signals are level-sensed or sensed on the falling edge of the input, as selected by ISCR bits IRQ0SC to IRQ7SC*. These interrupts can be masked collectively by the I bit in the CCR, and can be enabled and disabled individually by setting and clearing bits IRQ0E to IRQ7E* in the IRQ enable register.

In the H8/3534, the IRQ_6 input signal can be logically ORed internally with the key sense input signals.

When $KEYIN_0$ to $KEYIN_7$ pins ($P6_0$ to $P6_7$) are used for key sense input, the corresponding KMIMR bits should be cleared to 0 to enable the corresponding key sense input interrupts. KMIMR bits corresponding to unused key sense inputs should be set to 1 to disable the interrupts. All 8 key sense input interrupts are combined into a single IRQ_6 interrupt.

When one of these interrupts is accepted, the I bit is set to 1. IRQ_0 to IRQ_7 have interrupt vector numbers 4 to 11. They are prioritized in order from IRQ_7 (low) to IRQ_0 (high). For details, see table 4-2.

Interrupts IRQ_0 to IRQ_7 do not depend on whether pins IRQ_0 to IRQ_7* are input or output pins. When using external interrupts IRQ_0 to IRQ_7, clear the corresponding DDR bits to 0 to set these pins to the input state, and do not use these pins as input or output pins for the timers, serial communication interface, or A/D converter.

Note: * For the H8/3522, read "IRQ_0 to IRQ_2", "bits IRQ0SC to IRQ2SC", "bits IRQ0E to IRQ2E", and "pins IRQ_0 to IRQ_2", respectively.

4.3.4 Internal Interrupts

Twenty-three [H8/3534] or nineteen [H8/3522] internal interrupts can be requested by the on-chip supporting modules. Each interrupt source has its own vector number, so the interrupt-handling routine does not have to determine which interrupt has occurred. All internal interrupts are masked when the I bit in the CCR is set to 1. When one of these interrupts is accepted, the I bit is set to 1 to mask further interrupts (except NMI). The vector numbers are 12 to 36. For the priority order, see table 4-2.

4.3.5 Interrupt Handling

Interrupts are controlled by an interrupt controller that arbitrates between simultaneous interrupt requests, commands the CPU to start the hardware interrupt exception-handling sequence, and furnishes the necessary vector number. Figure 4-4 shows a block diagram of the interrupt controller.

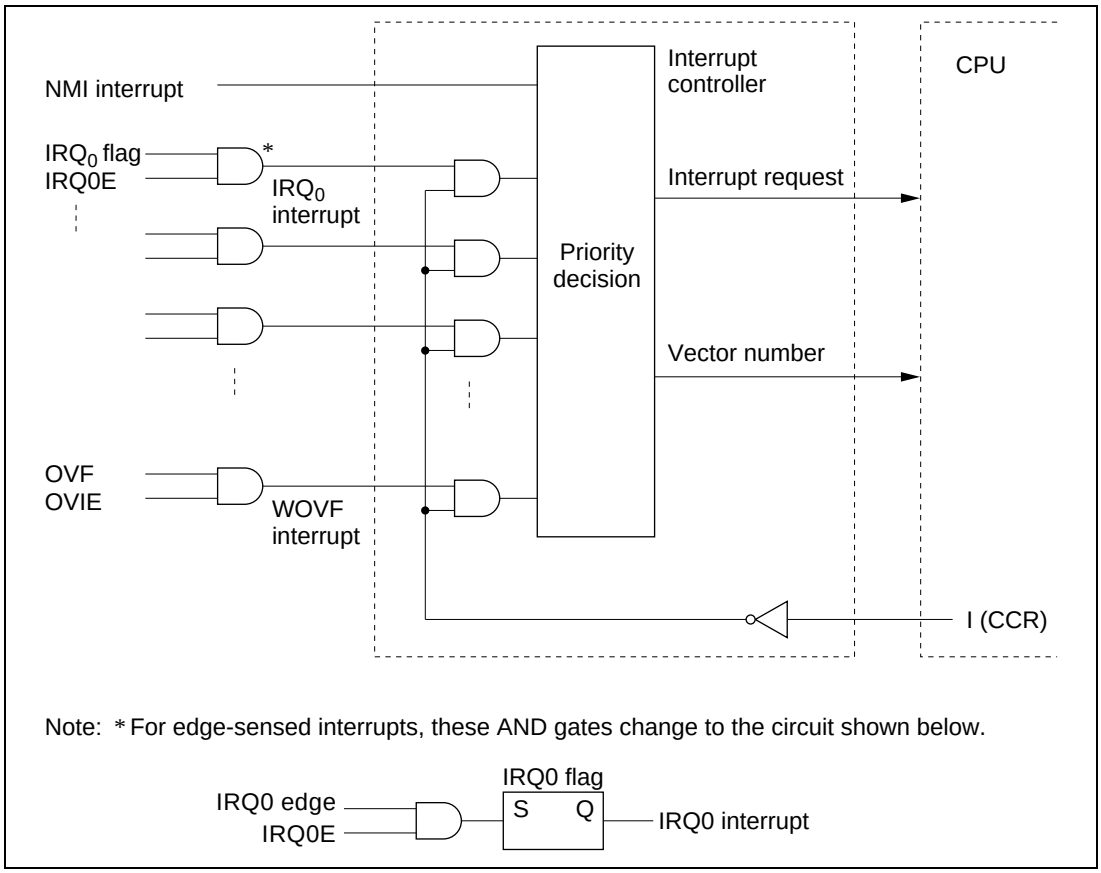


Figure 4-4 Block Diagram of Interrupt Controller

The IRQ interrupts and interrupts from the on-chip supporting modules (except for reset selected for a watchdog timer overflow) all have corresponding enable bits. When the enable bit is cleared to 0, the interrupt signal is not sent to the interrupt controller, so the interrupt is ignored. These interrupts can also all be masked by setting the CPU's interrupt mask bit (I) to 1. Accordingly, these interrupts are accepted only when their enable bit is set to 1 and the I bit is cleared to 0.

The nonmaskable interrupt (NMI) is always accepted, except in the reset state and hardware standby mode.

When an NMI or another enabled interrupt is requested, the interrupt controller transfers the interrupt request to the CPU and indicates the corresponding vector number. (When two or more interrupts are requested, the interrupt controller selects the vector number of the interrupt with the highest priority.) When notified of an interrupt request, at the end of the current instruction or current hardware exception-handling sequence, the CPU starts the hardware exception-handling sequence for the interrupt and latches the vector number.

Figure 4-5 shows the interrupt-handling sequence.

(1) An interrupt request is sent to the interrupt controller when an NMI interrupt occurs, and when an interrupt occurs on an IRQ input line or in an on-chip supporting module provided the enable bit of that interrupt is set to 1.
(2) The interrupt controller checks the I bit in CCR and accepts the interrupt request if the I bit is cleared to 0. If the I bit is set to 1 only NMI requests are accepted; other interrupt requests remain pending.
(3) Among all accepted interrupt requests, the interrupt controller selects the request with the highest priority and passes it to the CPU. Other interrupt requests remain pending.
(4) When it receives the interrupt request, the CPU waits until completion of the current instruction or hardware exception-handling sequence, then starts the hardware exception-handling sequence for the interrupt and latches the interrupt vector number.
(5) In the hardware exception-handling sequence, the CPU first pushes the PC and CCR onto the stack. See figure 4-6. The stacked PC indicates the address of the first instruction that will be executed on return from the software interrupt-handling routine.
(6) Next the I bit in CCR is set to 1, masking all further interrupts except NMI.
(7) The vector address corresponding to the vector number is generated, the vector table entry at this vector address is loaded into the program counter, and execution branches to the software interrupt-handling routine at the address indicated by that entry.

Figure 4-7 shows the interrupt timing sequence for the case in which the software interrupt-handling routine is in on-chip ROM and the stack is in on-chip RAM.

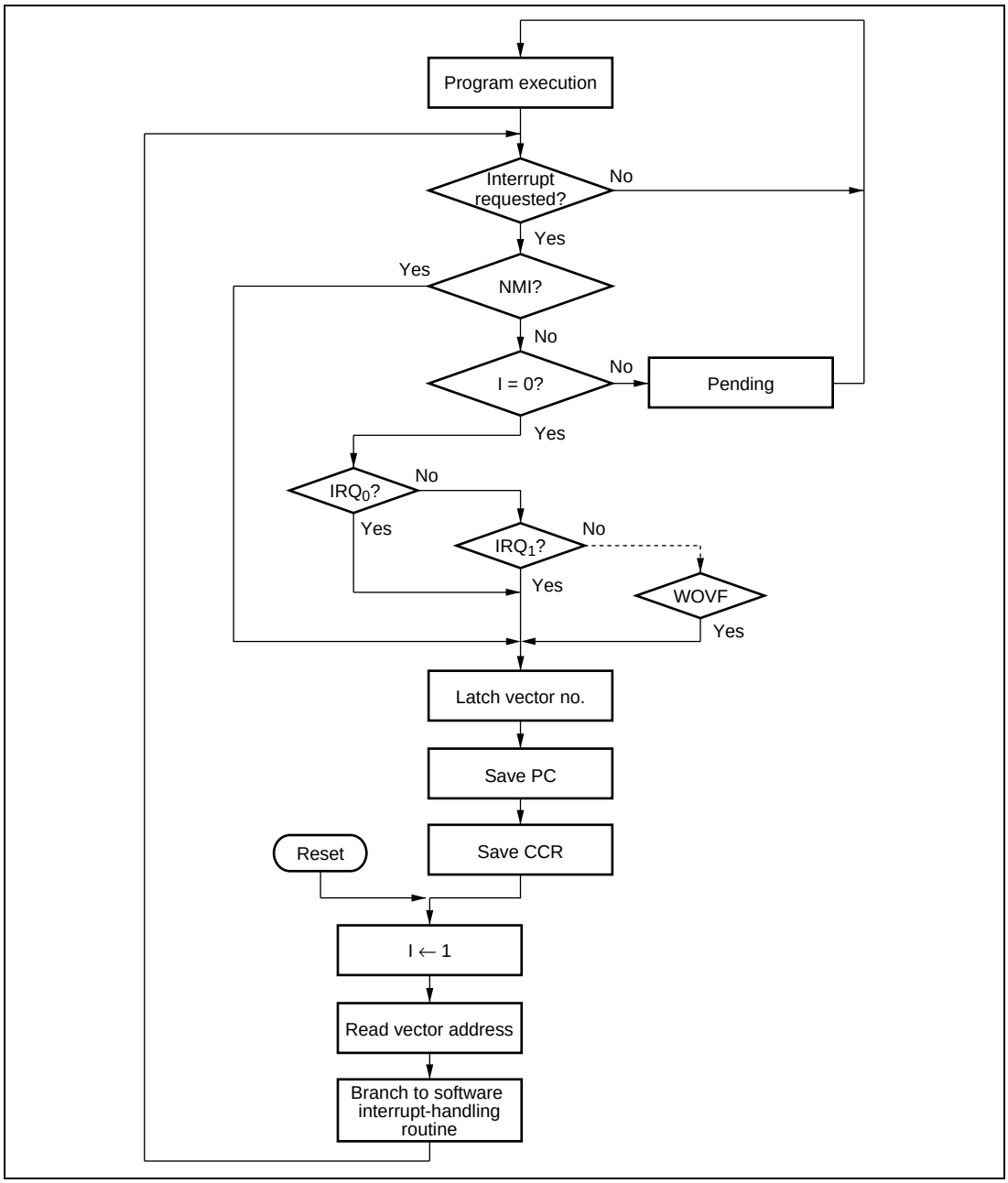


Figure 4-5 Hardware Interrupt-Handling Sequence

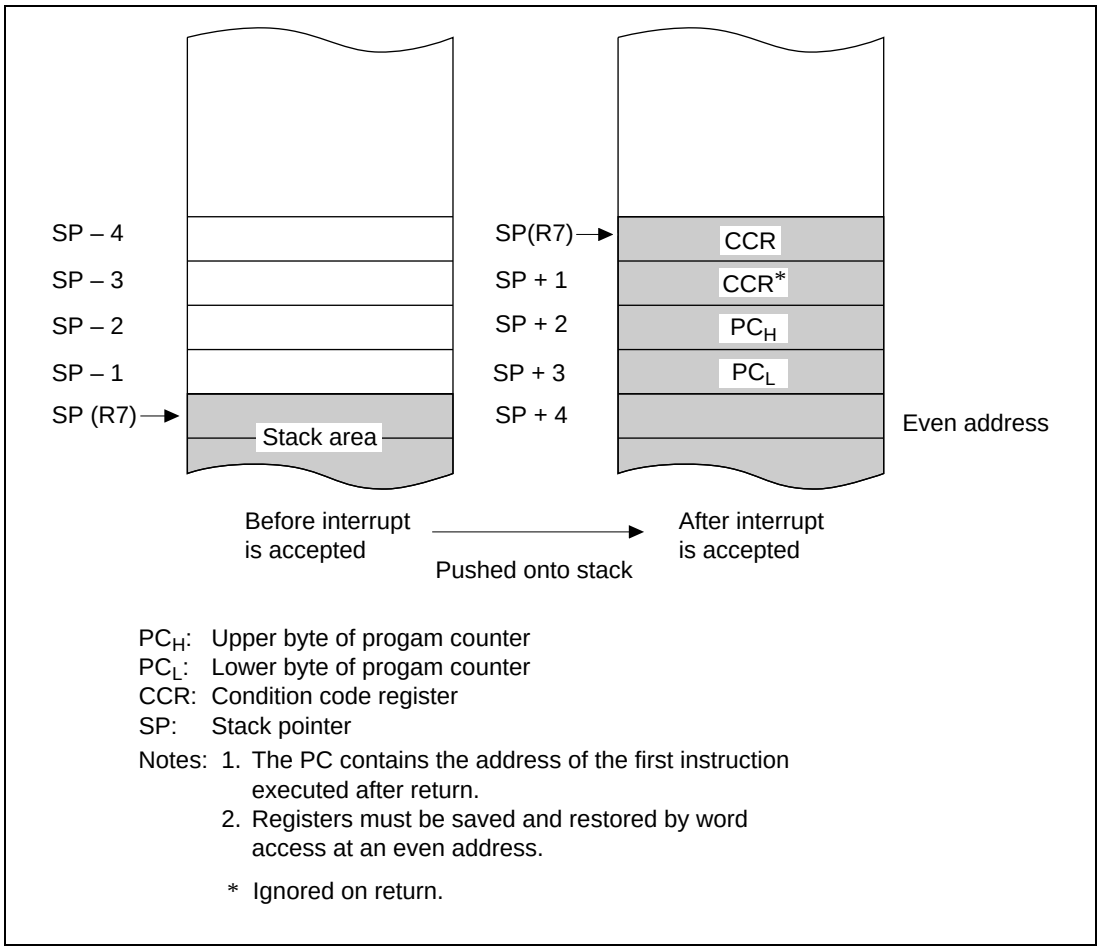
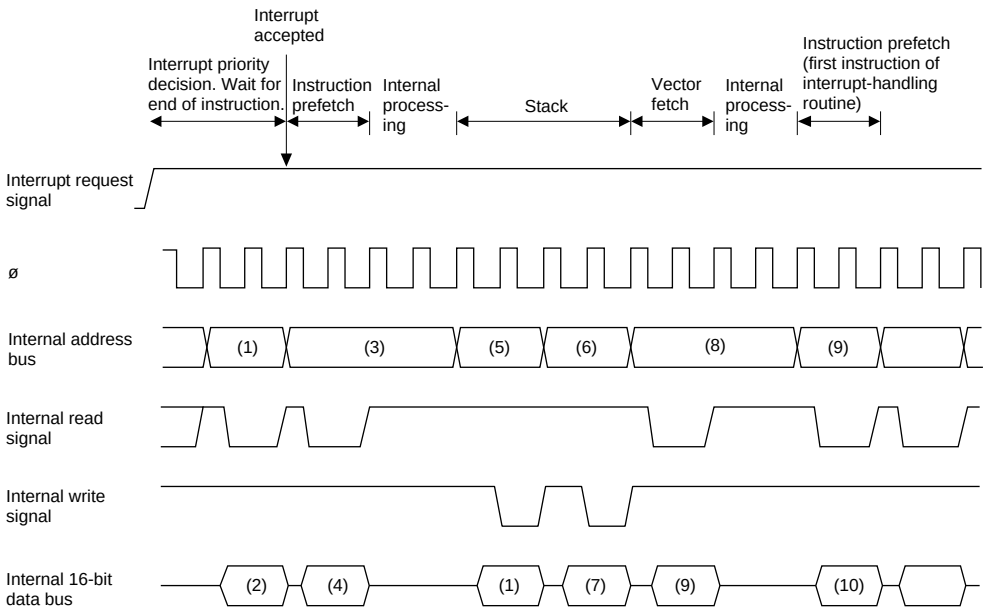


Figure 4-6 Usage of Stack in Interrupt Handling

The CCR is comprised of one byte, but when it is saved to the stack, it is treated as one word of data. During interrupt processing, two identical bytes of CCR data are saved to the stack to create one word of data. When the RTE instruction is executed to restore the value from the stack, the byte located at the even address is loaded into CCR, and the byte located at the odd address is ignored.



(1) Instruction prefetch address (Pushed on stack. Instruction is executed on return from interrupt-handling routine.)
(2) (4) Instruction code (Not executed)
(3) Instruction prefetch address (Not executed)
(5) SP–2
(6) SP–4
(7) CCR
(8) Address of vector table entry
(9) Vector table entry (address of first instruction of interrupt-handling routine)
(10) First instruction of interrupt-handling routine

Figure 4-7 Timing of Interrupt Sequence

4.3.6 Interrupt Response Time

Table 4-4 indicates the number of states that elapse from an interrupt request signal until the first instruction of the software interrupt-handling routine is executed. Since on-chip memory is accessed 16 bits at a time, very fast interrupt service can be obtained by placing interrupt-handling routines in on-chip ROM and the stack in on-chip RAM.

Table 4-4 Number of States before Interrupt Service

		Number of States	
No.	**Reason for Wait**	**On-Chip Memory**	**External Memory**
1	Interrupt priority decision	2*3	2*3
2	Wait for completion of current instruction*1	1 to 13	5 to 17*2
3	Save PC and CCR	4	12*2
4	Fetch vector	2	6*2
5	Fetch instruction	4	12*2
6	Internal processing	4	4
	Total	17 to 29	41 to 53 *2

Notes: 1. These values do not apply if the current instruction is EEPMOV.
2. If wait states are inserted in external memory access, add the number of wait states.
3. 1 for internal interrupts.

4.3.7 Precaution

Note that the following type of contention can occur in interrupt handling.

When software clears the enable bit of an interrupt to 0 to disable the interrupt, the interrupt becomes disabled after execution of the clearing instruction. If an enable bit is cleared by a BCLR or MOV instruction, for example, and the interrupt is requested during execution of that instruction, at the instant when the instruction ends the interrupt is still enabled, so after execution of the instruction, the hardware exception-handling sequence is executed for the interrupt. If a higher-priority interrupt is requested at the same time, however, the hardware exception-handling sequence is executed for the higher-priority interrupt and the interrupt that was disabled is ignored.

Similar considerations apply when an interrupt request flag is cleared to 0.

Figure 4-8 shows an example in which the OCIAE bit is cleared to 0.

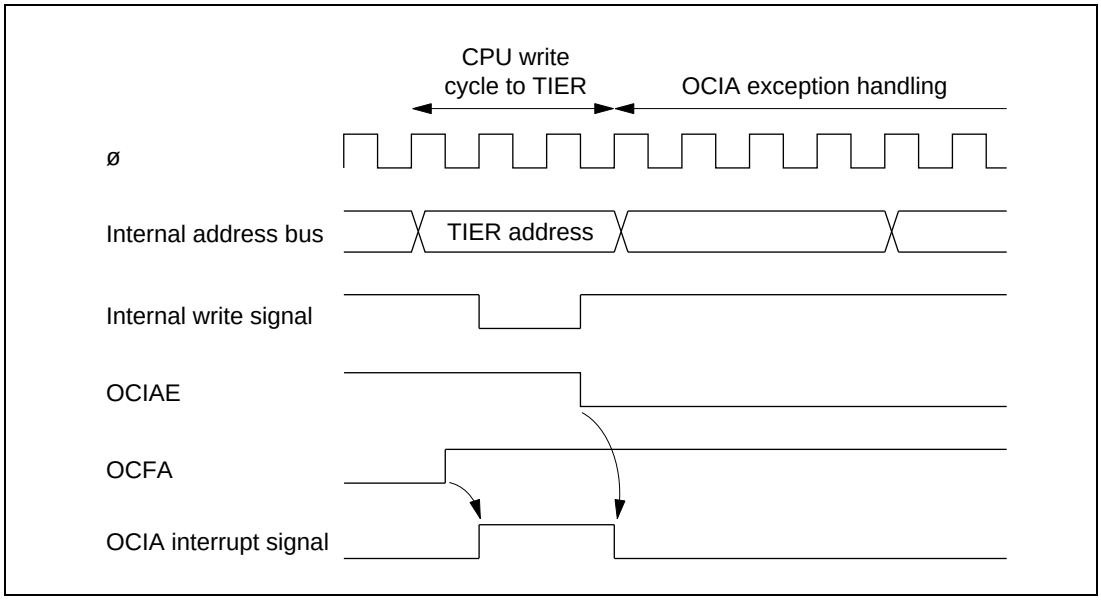


Figure 4-8 Contention between Interrupt and Disabling Instruction

The above contention does not occur if the enable bit or flag is cleared to 0 while the interrupt mask bit (I) is set to 1.

4.4 Note on Stack Handling

In word access, the least significant bit of the address is always assumed to be 0. The stack is always accessed by word access. Care should be taken to keep an even value in the stack pointer (general register R7). Use the PUSH Rn and POP Rn (or MOV.W Rn, @–SP and MOV.W @SP+, Rn) instructions to push and pop registers on the stack.

Setting the stack pointer to an odd value can cause programs to crash. Figure 4-9 shows an example of damage caused when the stack pointer contains an odd address.

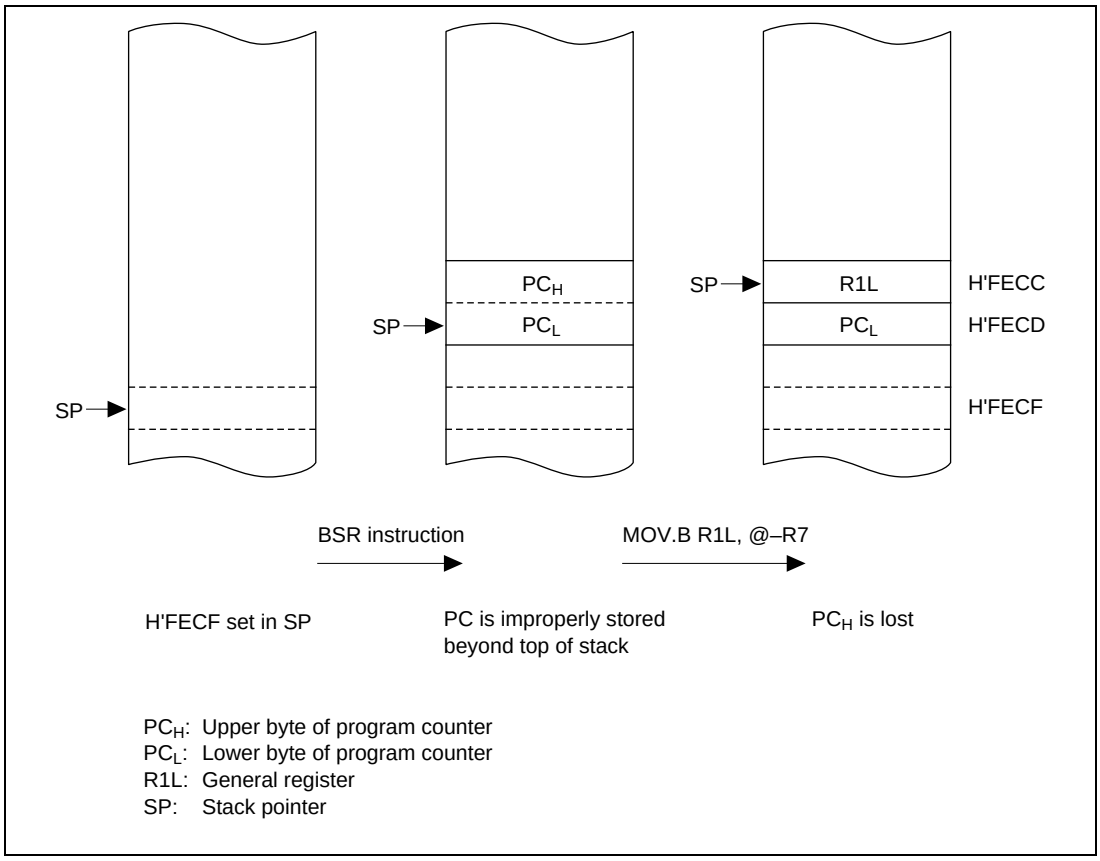


Figure 4-9 Example of Damage Caused by Setting an Odd Address in SP

Section 5 Wait-State Controller

5.1 Overview

The H8/3534 and H8/3522 have an on-chip wait-state controller that enables insertion of wait states into bus cycles for interfacing to low-speed external devices.

5.1.1 Features

Features of the wait-state controller are listed below.

- Three selectable wait modes: programmable wait mode, pin auto-wait mode, and pin wait mode
- Automatic insertion of zero to three wait states

5.1.2 Block Diagram

Figure 5-1 shows a block diagram of the wait-state controller.

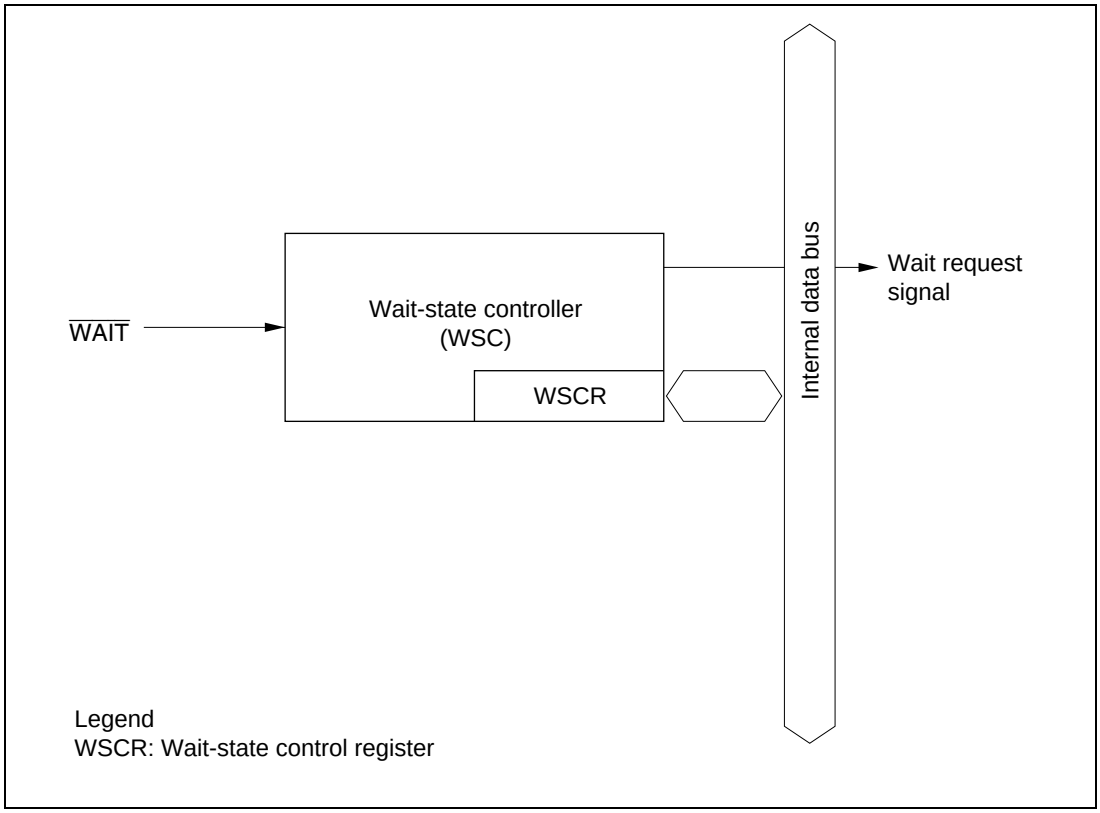


Figure 5-1 Block Diagram of Wait-State Controller

5.1.3 Input/Output Pins

Table 5-1 summarizes the wait-state controller's input pin.

Table 5-1 Wait-State Controller Pins

Name	Abbreviation	I/O	Function
Wait	$\overline{\text{WAIT}}$	Input	Wait request signal for access to external addresses

5.1.4 Register Configuration

Table 5-2 summarizes the wait-state controller's register.

Table 5-2 Register Configuration

Address	Name	Abbreviation	R/W	Initial Value
H'FFC2	Wait-state control register	WSCR	R/W	H'08

5.2 Register Description

5.2.1 Wait-State Control Register (WSCR)

WSCR is an 8-bit readable/writable register that selects the wait mode for the wait-state controller (WSC) and specifies the number of wait states. It also controls frequency division of the clock signals supplied to the supporting modules.

Bit	7	6	5	4	3	2	1	0
[H8/3534]	(RAMS)	(RAM0)	CKDBL	—	WMS1	WMS0	WC1	WC0
Initial value	0	0	0	0	1	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
[H8/3522]	—	—	CKDBL	—	WMS1	WMS0	WC1	WC0
Initial value	0	0	0	0	1	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WSCR is initialized to H'08 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—RAM Select (RAMS) [H8/3534]

Bit 6—RAM Area Select (RAM0) [H8/3534]

These bits are reserved. They should not be set to 1.

Bits 7 and 6—Reserved: [H8/3522] These bits are reserved, but they can be written and read. Their initial value is 0.

Bit 5—Clock Double (CKDBL): Controls frequency division of clock signals supplied to supporting modules. For details, see section 6, Clock Pulse Generator.

Bit 4—Reserved: This bit is reserved, but it can be written and read. Its initial value is 0.

Bits 3 and 2—Wait Mode Select 1 and 0 (WMS1/0): These bits select the wait mode.

Bit 3 WMS1	Bit 2 WMS0	Description	
0	0	Programmable wait mode	
	1	No wait states inserted by wait-state controller	
1	0	Pin wait mode	(Initial value)
	1	Pin auto-wait mode	

Bits 1 and 0—Wait Count 1 and 0 (WC1/0): These bits select the number of wait states inserted in access to external address areas.

Bit 1 WC1	Bit 0 WC0	Description	
0	0	No wait states inserted by wait-state controller	(Initial value)
	1	1 state inserted	
1	0	2 states inserted	
	1	3 states inserted	

5.3 Wait Modes

Programmable Wait Mode: The number of wait states (T_w) selected by bits WC1 and WC0 are inserted in all accesses to external addresses. Figure 5-2 shows the timing when the wait count is 1 (WC1 = 0, WC0 = 1).

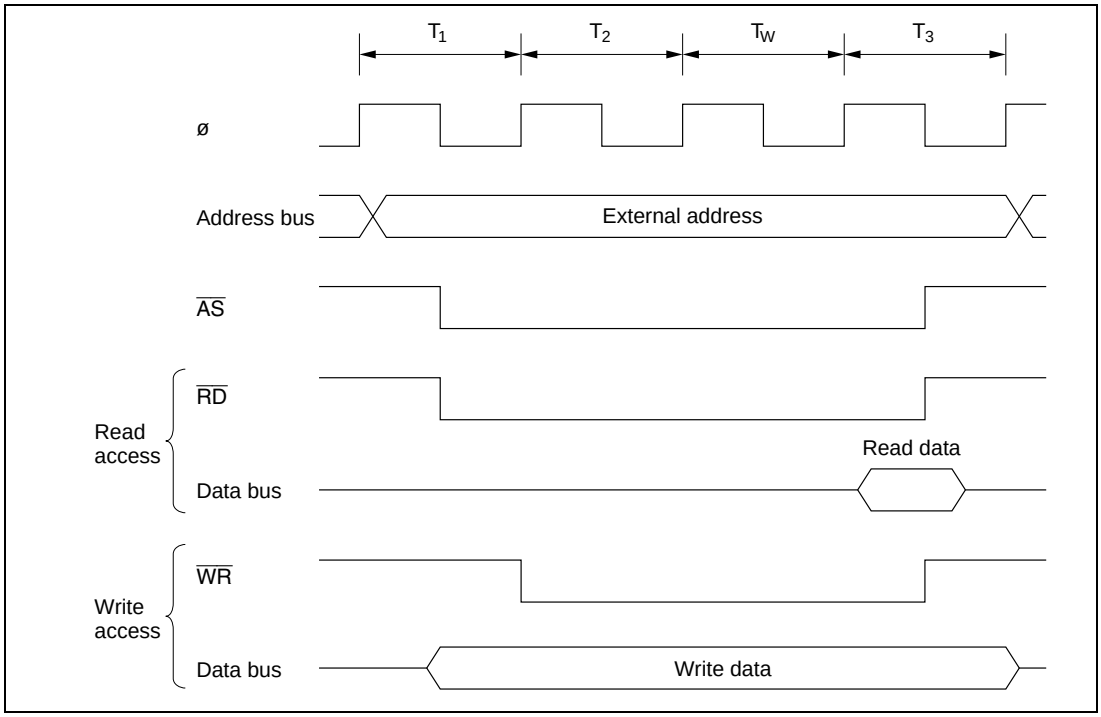


Figure 5-2 Programmable Wait Mode

Pin Wait Mode: In all accesses to external addresses, the number of wait states (T_w) selected by bits WC1 and WC0 are inserted. If the $\overline{\text{WAIT}}$ pin is low at the fall of the system clock (ϕ) in the last of these wait states, an additional wait state is inserted. If the $\overline{\text{WAIT}}$ pin remains low, wait states continue to be inserted until the $\overline{\text{WAIT}}$ signal goes high.

Pin wait mode is useful for inserting four or more wait states, or for inserting different numbers of wait states for different external devices.

Figure 5-3 shows the timing when the wait count is 1 (WC1 = 0, WC0 = 1) and one additional wait state is inserted by $\overline{\text{WAIT}}$ input.

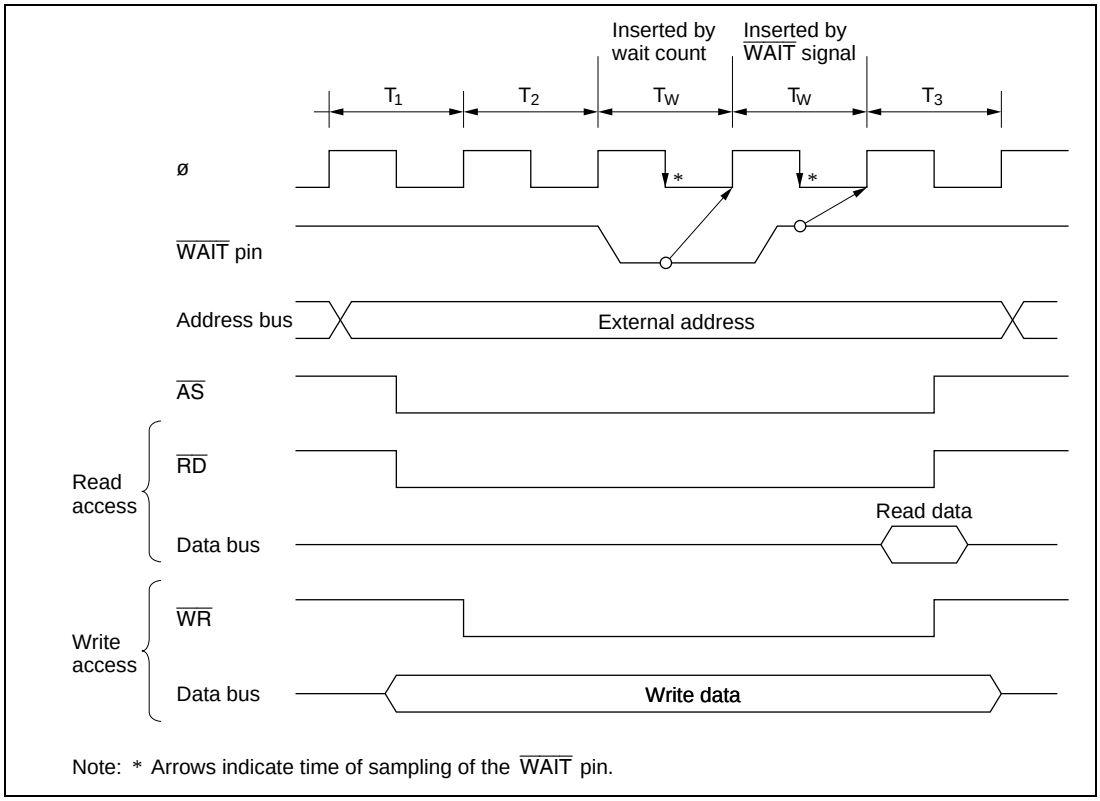


Figure 5-3 Pin Wait Mode

Pin Auto-Wait Mode: If the $\overline{WAIT}$ pin is low, the number of wait states (T_w) selected by bits WC1 and WC0 are inserted.

In pin auto-wait mode, if the $\overline{WAIT}$ pin is low at the fall of the system clock (ϕ) in the T_2 state, the number of wait states (T_w) selected by bits WC1 and WC0 are inserted. No additional wait states are inserted even if the $\overline{WAIT}$ 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 $\overline{WAIT}$ pin.

Figure 5-4 shows the timing when the wait count is 1.

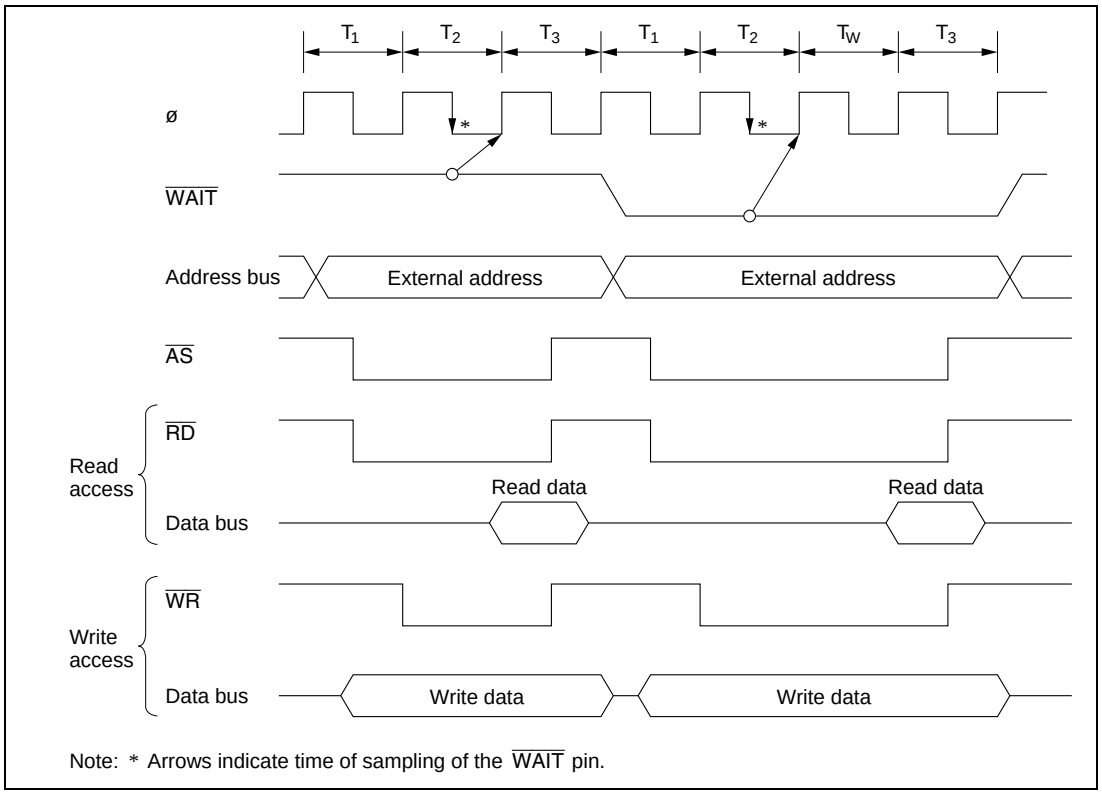


Figure 5-4 Pin Auto-Wait Mode

Section 6 Clock Pulse Generator

6.1 Overview

The H8/3534 and H8/3522 have a built-in clock pulse generator (CPG) consisting of an oscillator circuit, a duty adjustment circuit, and a divider and a prescaler that generates clock signals for the on-chip supporting modules.

6.1.1 Block Diagram

Figure 6-1 shows a block diagram of the clock pulse generator.

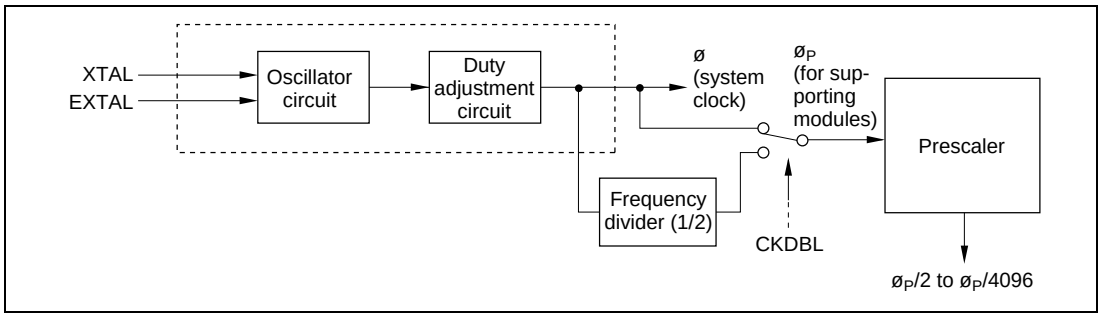


Figure 6-1 Block Diagram of Clock Pulse Generator

Input an external clock signal to the EXTAL pin, or connect a crystal resonator to the XTAL and EXTAL pins. The system clock frequency (ϕ) will be the same as the input frequency. This same system clock frequency (ϕ_p) can be supplied to timers and other supporting modules, or it can be divided by two. The selection is made by software, by controlling the CKDBL bit.

6.1.2 Wait-State Control Register (WSCR)

WSCR is an 8-bit readable/writable register that controls frequency division of the clock signals supplied to the supporting modules. It also controls wait-state insertion and emulation of flash memory by RAM.

WSCR is initialized to H'08 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit	7	6	5	4	3	2	1	0
[H8/3534]	(RAMS)	(RAM0)	CKDBL	—	WMS1	WMS0	WC1	WC0
Initial value	0	0	0	0	1	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
[H8/3522]	—	—	CKDBL	—	WMS1	WMS0	WC1	WC0
Initial value	0	0	0	0	1	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit 7—RAM Select (RAMS) [H8/3534]

Bit 6—RAM Area Select (RAM0) [H8/3534]

These bits are reserved. They should not be set to 1.

Bits 7 and 6—Reserved: [H8/3522] These bits are reserved, but they can be written and read. Their initial value is 0.

Bit 5—Clock Double (CKDBL): Controls the frequency division of clock signals supplied to supporting modules.

Bit 5 CKDBL	Description
0	The undivided system clock (ϕ) is supplied as the clock (ϕ_P) for supporting modules. (Initial value)
1	The system clock (ϕ) is divided by two and supplied as the clock (ϕ_P) for supporting modules.

Bit 4—Reserved: This bit is reserved, but it can be written and read. Its initial value is 0.

Bits 3 and 2—Wait Mode Select 1 and 0 (WMS1/0)

Bits 1 and 0—Wait Count 1 and 0 (WC1/0)

These bits control wait-state insertion. For details, see section 5, Wait-State Controller.

6.2 Oscillator Circuit

If an external crystal is connected across the EXTAL and XTAL pins, the on-chip oscillator circuit generates a system clock signal. Alternatively, an external clock signal can be applied to the EXTAL pin.

1. Connecting an External Crystal

(1) Circuit Configuration: An external crystal can be connected as in the example in figure 6-2. Table 6-1 indicates the appropriate damping resistance Rd. An AT-cut parallel resonance crystal should be used.

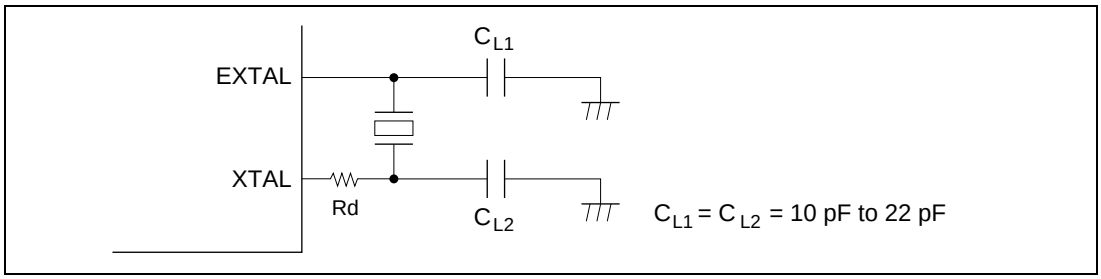


Figure 6-2 Connection of Crystal Oscillator (Example)

Table 6-1 Damping Resistance

Frequency (MHz)	4	8	10
Rd (Ω)	500	200	0

(2) Crystal Oscillator: Figure 6-3 shows an equivalent circuit of the crystal resonator. The crystal resonator should have the characteristics listed in table 6-2.

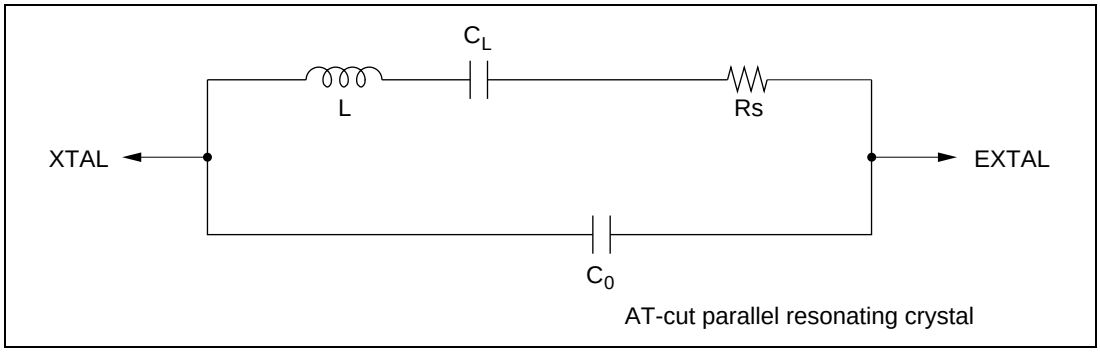


Figure 6-3 Equivalent Circuit of External Crystal

Table 6-2 External Crystal Parameters

Frequency (MHz)	4	8	10
Rs max (Ω)	120	80	70
C_0 (pF)		7 pF max	

Use a crystal with the same frequency as the desired system clock frequency (ϕ).

(3) Note on Board Design: When an external crystal is connected, other signal lines should be kept away from the crystal circuit to prevent induction from interfering with correct oscillation. See figure 6-4. The crystal and its load capacitors should be placed as close as possible to the XTAL and EXTAL pins.

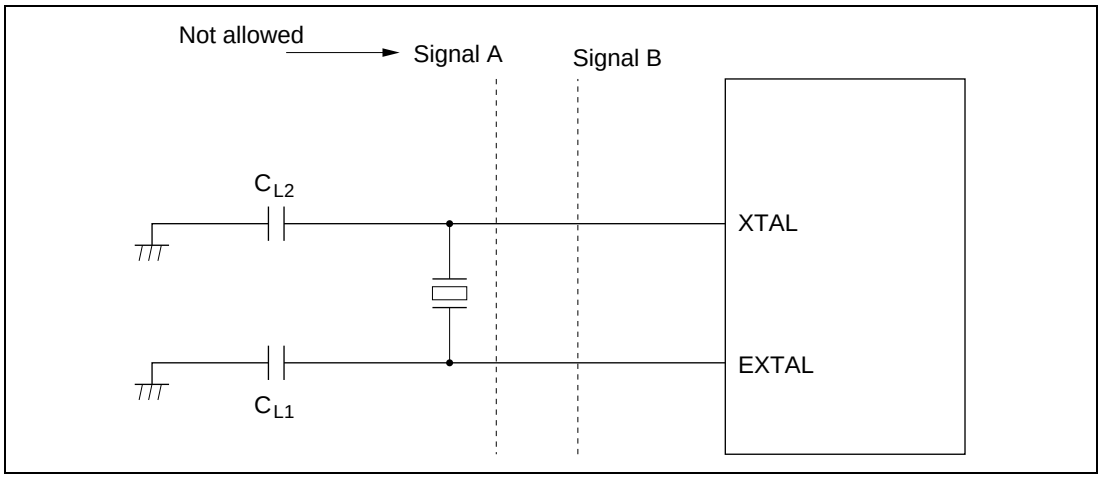


Figure 6-4 Board Design around External Crystal

2. Input of External Clock Signal

(1) Circuit Configuration: An external clock signal can be input as shown in the examples in figure 6-5. In example (b) in figure 6-5, the external clock signal should be kept high during standby.

If the XTAL pin is left open, make sure the stray capacitance does not exceed 10 pF.

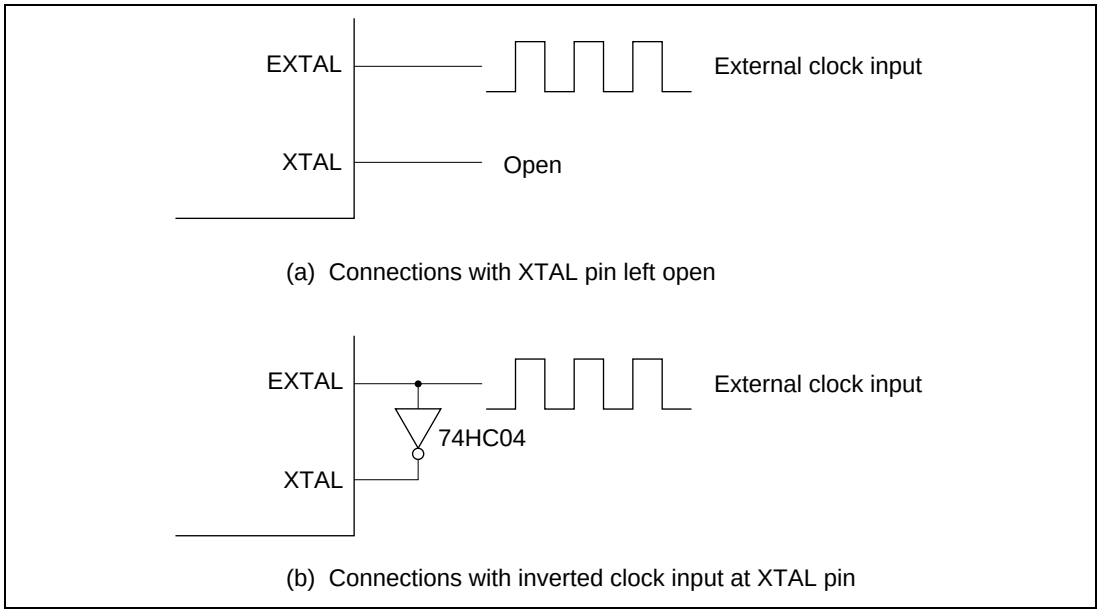


Figure 6-5 External Clock Input (Example)

(2) External Clock Input

The external clock signal should have the same frequency as the desired system clock (ϕ). Clock timing parameters are given in table 6-3 and figure 6-6.

Table 6-3 Clock Timing

Item	Symbol	V_{cc} = 5.0 V ±10% Min	Max	Unit	Test Conditions	
Low pulse width of external clock input	t_{EXL}	40	—	ns	Figure 6-6	
High pulse width of external clock input	t_{EXH}	40	—	ns		
External clock rise time	t_{EXr}	—	10	ns		
External clock fall time	t_{EXf}	—	10	ns		
Clock pulse width low	t_{CL}	0.3	0.7	t_{cyc}	ϕ ≥ 5 MHz	Figure 17-4
		0.4	0.6	t_{cyc}	ϕ < 5 MHz	
Clock pulse width high	t_{CH}	0.3	0.7	t_{cyc}	ϕ ≥ 5 MHz	
		0.4	0.6	t_{cyc}	ϕ < 5 MHz	

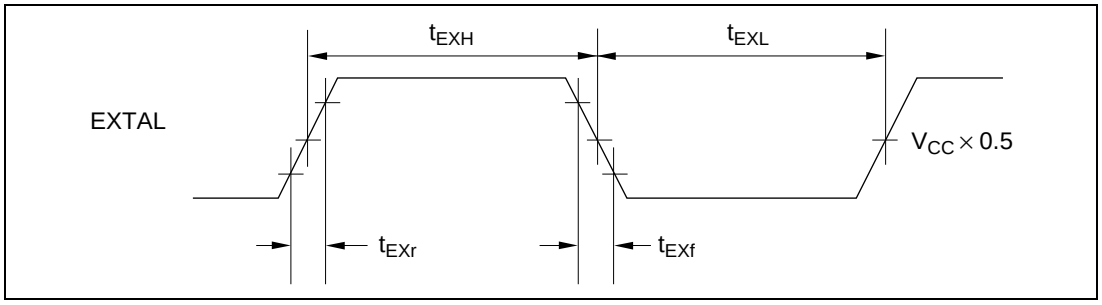


Figure 6-6 External Clock Input Timing

Table 6-4 shows the external clock output settling delay time. Figure 6-7 shows the timing for the external clock output settling delay time. The oscillator and duty correction circuit have the function of regulating the waveform of the external clock input to the EXTAL pin. When the specified clock signal is input to the EXTAL pin, internal clock signal output is confirmed after the elapse of the external clock output settling delay time (t_{DEXT}). As clock signal output is not confirmed during the t_{DEXT} period, the reset signal should be driven low and the reset state maintained during this time.

Table 6-4 External Clock Output Settling Delay Time

Conditions: V_{CC} = 4.5 to 5.5 V, AV_{CC} = 4.5 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 6-7

Note: * t_{DEXT} includes a 10 t_{cyc} $\overline{RES}$ pulse width (t_{RESW}).

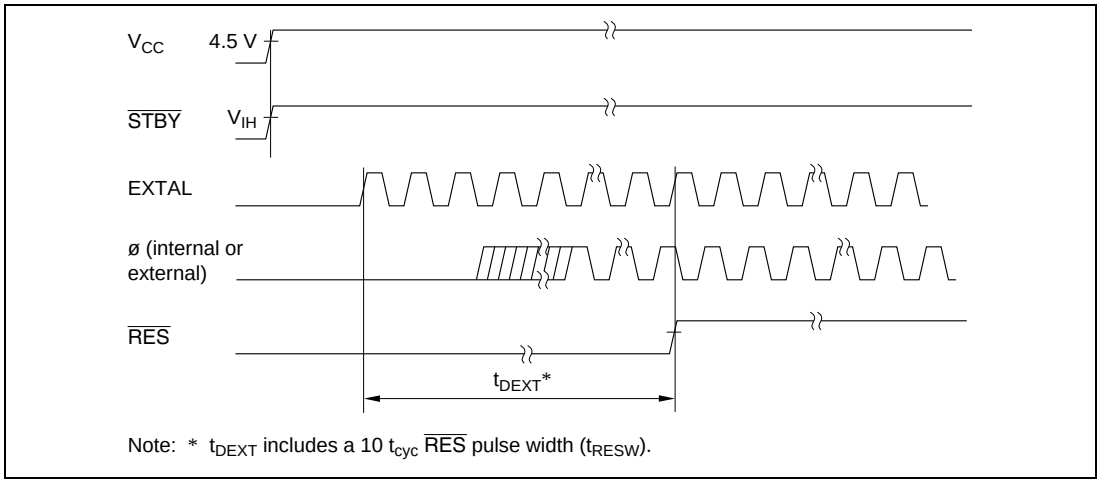


Figure 6-7 External Clock Output Settling Delay Time

6.3 Duty Adjustment Circuit

When the clock frequency is 5 MHz or above, the duty adjustment circuit adjusts the duty cycle of the signal from the oscillator circuit to generate the system clock (ϕ).

6.4 Prescaler

The clock for the on-chip supporting modules (ϕ_P) has either the same frequency as the system clock (ϕ) or this frequency divided by two, depending on the CKDBL bit. The prescaler divides the frequency of ϕ_P to generate internal clock signals with frequencies from $\phi_P/2$ to $\phi_P/4096$.

Section 7 I/O Ports

7.1 Overview

The H8/3534 has six 8-bit input/output ports, one 7-bit input/output port, one 3-bit input/output port, and one 8-bit dedicated input port. The H8/3522 has five 8-bit input/output ports, one 3-bit input/output port, and one 8-bit dedicated input port.

Table 7-1 lists the functions of each port in each operating mode. As table 7-1 indicates, the port pins are multiplexed, and the pin functions differ depending on the operating mode.

Each port has a data direction register (DDR) that selects input or output, and a data register (DR) that stores output data. If bit manipulation instructions will be executed on the port data direction registers, see “Notes on Bit Manipulation Instructions” in section 2.5.5, Bit Manipulation Instructions.

Ports 1, 2, 3, 4, 6, and 9 can drive one TTL load and a 90-pF capacitive load. Ports 5 and 8 can drive one TTL load and a 30-pF capacitive load. Ports 1 and 2 can drive LEDs (with 10-mA current sink). Ports 1 to 6, 8, and 9 can drive a darlington pair. Ports 1 to 3, and 6 have built-in MOS pull-up transistors.

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

Table 7-1 (a) Port Functions for H8/3534

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2**	**Mode 3 Master Mode**
Port 1	• 8-bit I/O port • Can drive LEDs • Built-in input pull-ups	$P1_7$ to $P1_0$/A_7 to A_0	Lower address output (A_7 to A_0)	Lower address output (A_7 to A_0) or general input	General input/output (Can also be used as Key scan output port)
Port 2	• 8-bit I/O port • Can drive LEDs • Built-in input pull-ups	$P2_7$ to $P2_0$/A_{15} to A_8	Upper address output (A_{15} to A_8)	Upper address output (A_{15} to A_8) or general input	General input/output (Can also be used as Key-scan output port)
Port 3	• 8-bit I/O port • Built-in input pull-ups	$P3_7$ to $P3_0$ / D_7 to D_0	Data bus (D_7 to D_0)		General input/ output
Port 4	• 8-bit I/O port	$P4_7$/PW_1 $P4_6$/PW_0	PWM timer 0/1 output (PW_0, PW_1), or general input/output		
		$P4_5$/$TMRI_1$ $P4_4$/TMO_1 $P4_3$/$TMCI_1$	8-bit timer 1 input/output ($TMCI_1$, TMO_1, $TMRI_1$), or general input/output		
		$P4_2$/$TMRI_0$ $P4_1$/TMO_0 $P4_0$/$TMCI_0$	8-bit timer 0 input/output ($TMCI_0$, TMO_0, $TMRI_0$) or general input/output		
Port 5	• 3-bit I/O port	$P5_2$/SCK_0 $P5_1$/RxD_0 $P5_0$/TxD_0	Serial communication interface 0 input/output (TxD_0, RxD_0, SCK_0) or general input/output		
Port 6	• 8-bit I/Oport • Built-in input pull-ups	$P6_7$/$\overline{IRQ}_7$/$\overline{KEYIN}_7$ $P6_6$/FTOB/$\overline{IRQ}_6$/$\overline{KEYIN}_6$ $P6_5$/FTID/$\overline{KEYIN}_5$ $P6_4$/FTIC/$\overline{KEYIN}_4$ $P6_3$/FTIB/$\overline{KEYIN}_3$ $P6_2$/FTIA/$\overline{KEYIN}_2$ $P6_1$/FTOA/$\overline{KEYIN}_1$ $P6_0$/FTCI/$KEYIN_0$	16-bit free-running timer input/output (FTCI, FTOA, FTIA, FTIB, FTIC, FTID, FTOB), key sense interrupt input($\overline{KEYIN}_7$ to $\overline{KEYIN}_0$), external interrupt input ($\overline{IRQ}_7$, $\overline{IRQ}_6$), or general input/output		
Port 7	• 8-bit input port	$P7_7$~$P7_0$/AN_7~AN_0	A/D converter analog input (AN_7 to AN_0) or general input		

Table 7-1 (a) Port Functions for H8/3534 (cont)

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2**	**Mode 3**
Port 8	• 7-bit I/O port	$P8_6/\overline{IRQ_6}/SCK_1$ $P8_5/\overline{IRQ_4}/RxD_1$ $P8_4/\overline{IRQ_3}/TxD_1$	Serial communication interface 1 input/output (TxD_1, RxD_1, SCK_1), external interrupt input ($\overline{IRQ_5}$, $\overline{IRQ_4}$, $\overline{IRQ_3}$), or general input/output		
		$P8_3$ $P8_2$ $P8_1$ $P8_0$	General input/output		
Port 9	• 8-bit I/O port	$P9_7/\overline{WAIT}$	Expanded data bus control input($\overline{WAIT}$), or general input/output		General input/ output
		$P9_6/\phi$	System clock (ϕ) output		ϕ output or general input
		$P9_5/\overline{AS}$ $P9_4/\overline{WR}$ $P9_3/\overline{RD}$	Expanded data bus control output($\overline{RD}$, $\overline{WR}$, $\overline{AS}$)		General input/ output
		$P9_2/\overline{IRQ_0}$ $P9_1/\overline{IRQ_1}$	External interrupt ($\overline{IRQ_0}$, $\overline{IRQ_1}$) or general input/output		
		$P9_0/\overline{ADTRG}/\overline{IRQ_2}$	A/D converter external trigger input ($\overline{ADTRG}$), external interrupt input ($\overline{IRQ_2}$), or general input/output		

Table 7-1 (b) Port Functions for H8/3522

			Expanded Modes		Single-Chip Mode
Port	**Description**	**Pins**	**Mode 1**	**Mode 2**	**Mode 3 Master Mode**
Port 1	• 8-bit I/O port • Can drive LEDs • Built-in input pull-ups	$P1_7$ to $P1_0$/A_7 to A_0	Lower address output (A_7 to A_0)	When DDR = 0(after reset): general input When DDR = 1: lower address output (A_7 to A_0)	General input/output
Port 2	• 8-bit I/O port • Can drive LEDs • Built-in input pull-ups	$P2_7$ to $P2_0$/A_{15} to A_8	Upper address output (A_{15} to A_8)	When DDR = 0 (after reset): general input When DDR = 1: upper address output (A_{15} to A_8)	General input/output
Port 3	• 8-bit I/O port • Built-in input pull-ups	$P3_7$ to $P3_0$/ D_7 to D_0	Data bus (D_7 to D_0)		General input/ output
Port 4	• 8-bit I/O port	$P4_7$/$\overline{WAIT}$	Expanded data bus control input ($\overline{WAIT}$) or general input/output		General input/ output
		$P4_6$/ϕ	System clock (ϕ) output		When DDR=0: (after reset): When DDR=1:general input ϕ output
		$P4_5$/$\overline{AS}$ $P4_4$/$\overline{WR}$ $P4_3$/$\overline{RD}$	Expanded data bus control output($\overline{RD}$, $\overline{WR}$, $\overline{AS}$)		General input/ output
		$P4_2$/IRQ_0 $P4_1$/IRQ_1	External interrupt ($\overline{IRQ_0}$, $\overline{IRQ_1}$) or general input/output		
		$P4_0$/IRQ_2/$\overline{ADTRG}$	A/D converter external trigger input ($\overline{ADTRG}$), external interrupt ($\overline{IRQ_2}$), or general input/output		
Port 5	• 3-bit I/O port	$P5_2$/SCK $P5_1$/RxD $P5_0$/TxD	Serial communication interface input/output (TxD, RxD, SCK) or general input/output		
Port 6	• 8-bit I/O port	$P6_7$/TMO_1 $P6_6$/FTOB/$TMRI_1$ $P6_5$/FTID/$TMCI_1$ $P6_4$/FTIC/TMO_0 $P6_3$/FTIB/$TMRI_0$ $P6_2$/FTIA $P6_1$/FTOA $P6_0$/FTCI/$TMCI_0$	16-bit free-running timer input/output (FTCI, FTOA, FTIA, FTIB, FTIC, FTID, FTOB), 8-bit timer 0 and 1 input/ output ($TMCI_0$, $TMRI_0$, TMO_0, $TMCI_1$, $TMRI_1$, TMO_1), or general input/output		
Port 7	• 8-bit input port	$P7_7$ to $P7_0$/AN_7 to AN_0	A/D converter analog input (AN_7 to AN_0) or general input		

7.2 Port 1

7.2.1 Overview

Port 1 is an 8-bit input/output port with the pin configuration shown in figure 7-1. The pin functions differ depending on the operating mode.

Port 1 has built-in, software-controllable MOS input pull-up transistors that can be used in modes 2 and 3.

Pins in port 1 can drive one TTL load and a 90-pF capacitive load. They can also drive LEDs and darlington transistors.

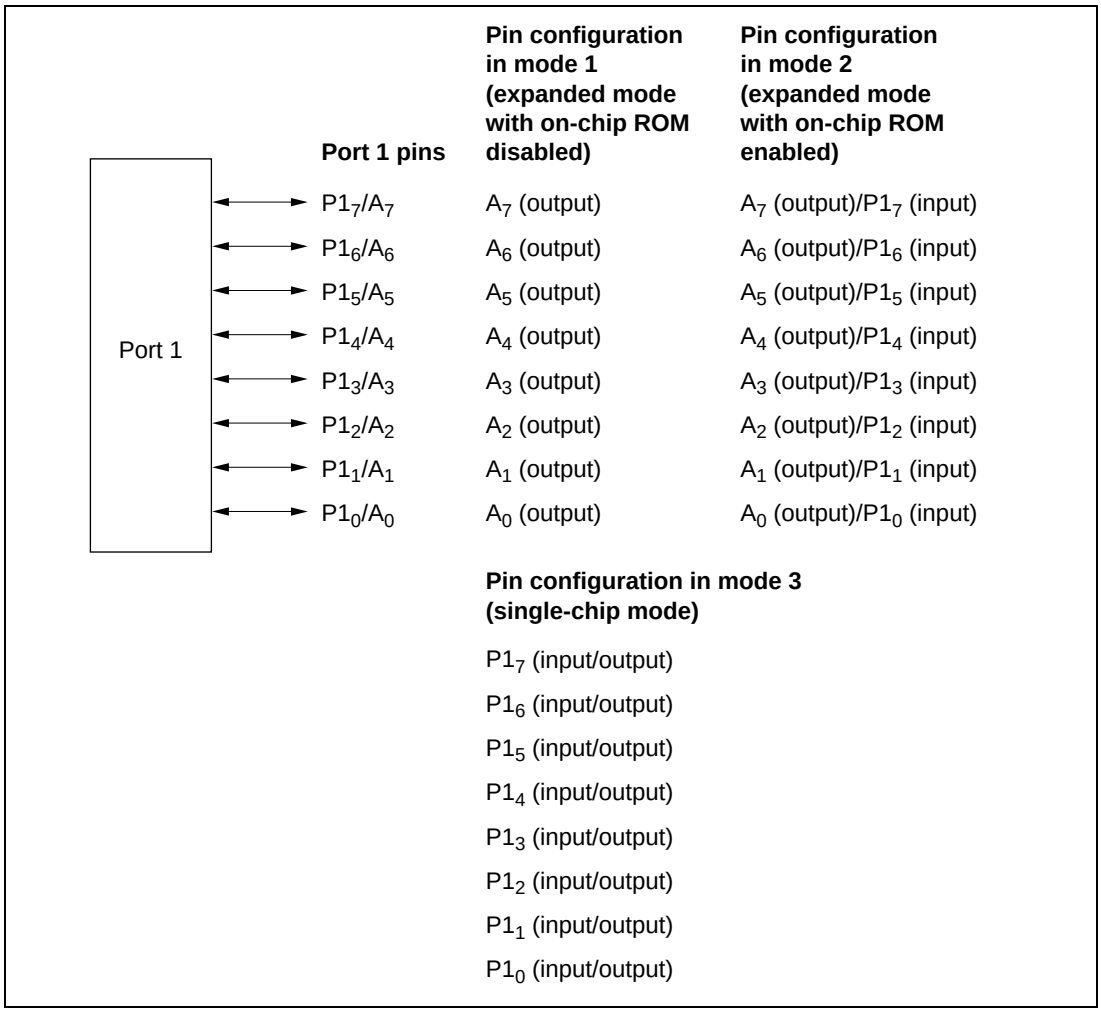


Figure 7-1 Port 1 Pin Configuration

7.2.2 Register Configuration and Descriptions

Table 7-2 summarizes the port 1 registers.

Table 7-2 Port 1 Registers

Name	Abbreviation	Read/Write	Initial Value	Address
Port 1 data direction register	P1DDR	W	H'FF (mode 1) H'00 (modes 2 and 3)	H'FFB0
Port 1 data register	P1DR	R/W	H'00	H'FFB2
Port 1 input pull-up control register	P1PCR	R/W	H'00	H'FFAC

Port 1 Data Direction Register (P1DDR)

Bit	7	6	5	4	3	2	1	0
	$P1_7DDR$	$P1_6DDR$	$P1_5DDR$	$P1_4DDR$	$P1_3DDR$	$P1_2DDR$	$P1_1DDR$	$P1_0DDR$
Mode 1								
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—
Modes 2 and 3								
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

P1DDR controls the input/output direction of each pin in port 1.

Mode 1: The P1DDR values are fixed at 1. Port 1 consists of lower address output pins. P1DDR values cannot be modified and are always read as 1.

In hardware standby mode, the address bus is in the high-impedance state.

Mode 2: A pin in port 1 is used for address output if the corresponding P1DDR bit is set to 1, and for general input if this bit is cleared to 0.

Mode 3: A pin in port 1 is used for general output if the corresponding P1DDR bit is set to 1, and for general input if this bit is cleared to 0.

In modes 2 and 3, P1DDR is a write-only register. Read data is invalid. If read, all bits always read 1. P1DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values, so if a transition to software standby mode occurs while a P1DDR bit is set to 1, the corresponding pin remains in the output state.

Port 1 Data Register (P1DR)

Bit	7	6	5	4	3	2	1	0
	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
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

P1DR is an 8-bit register that stores data for pins $P1_7$ to $P1_0$. When a P1DDR bit is set to 1, if port 1 is read, the value in P1DR is obtained directly, regardless of the actual pin state. When a P1DDR bit is cleared to 0, if port 1 is read the pin state is obtained.

P1DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

Port 1 Input Pull-Up Control Register (P1PCR)

Bit	7	6	5	4	3	2	1	0
	$P1_7$PCR	$P1_6$PCR	$P1_5$PCR	$P1_4$PCR	$P1_3$PCR	$P1_2$PCR	$P1_1$PCR	$P1_0$PCR
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

P1PCR is an 8-bit readable/writable register that controls the input pull-up transistors in port 1. If a P1DDR bit is cleared to 0 (designating input) and the corresponding P1PCR bit is set to 1, the input pull-up transistor is turned on.

P1PCR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

7.2.3 Pin Functions in Each Mode

Port 1 has different pin functions in different modes. A separate description for each mode is given below.

Pin Functions in Mode 1: In mode 1 (expanded mode with on-chip ROM disabled), port 1 is automatically used for lower address output (A_7 to A_0). Figure 7-2 shows the pin functions in mode 1.

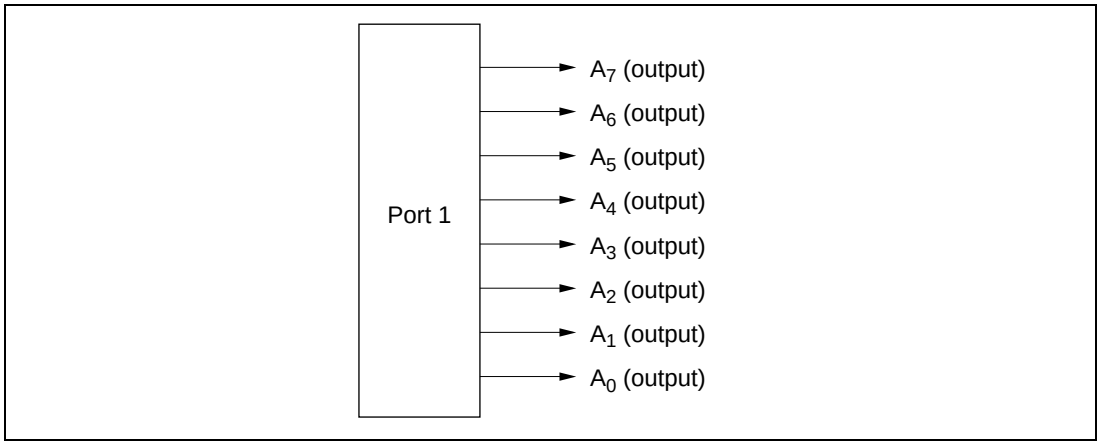


Figure 7-2 Pin Functions in Mode 1 (Port 1)

Mode 2: In mode 2 (expanded mode with on-chip ROM enabled), port 1 can provide lower address output pins and general input pins. Each pin becomes a lower address output pin if its P1DDR bit is set to 1, and a general input pin if this bit is cleared to 0. Following a reset, all pins are input pins. To be used for address output, their P1DDR bits must be set to 1. Figure 7-3 shows the pin functions in mode 2.

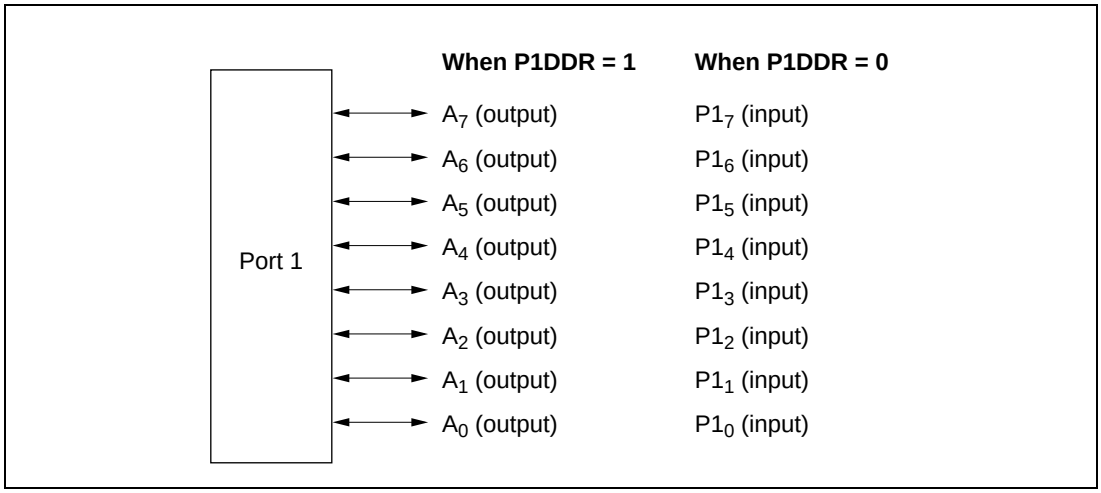


Figure 7-3 Pin Functions in Mode 2 (Port 1)

Mode 3: In mode 3 (single-chip mode), the input or output direction of each pin can be selected individually. A pin becomes a general input pin when its P1DDR bit is cleared to 0 and a general output pin when this bit is set to 1. Figure 7-4 shows the pin functions in mode 3.

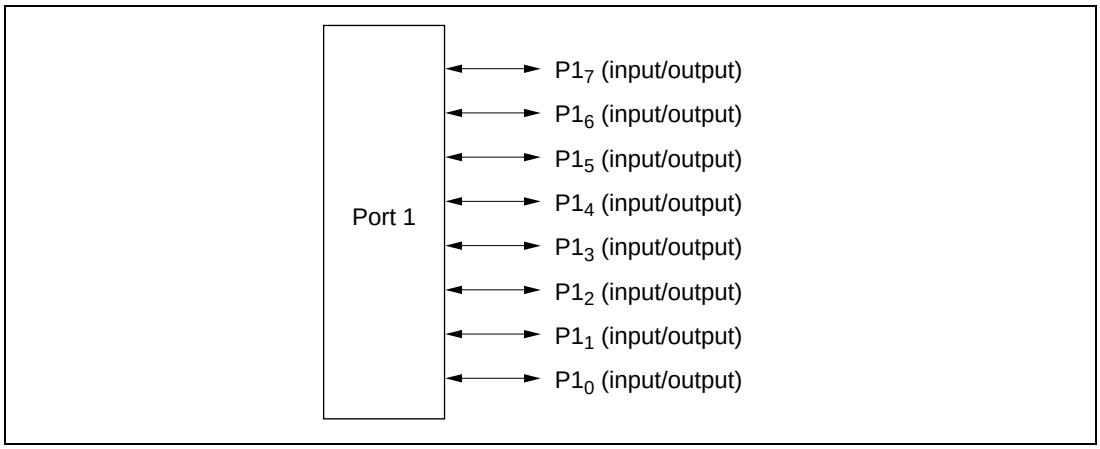


Figure 7-4 Pin Functions in Mode 3 (Port 1)

7.2.4 Input Pull-Up Transistors

Port 1 has built-in programmable input pull-up transistors that are available in modes 2 and 3. The pull-up for each bit can be turned on and off individually. To turn on an input pull-up in mode 2 or 3, set the corresponding P1PCR bit to 1 and clear the corresponding P1DDR bit to 0. P1PCR is cleared to H'00 by a reset and in hardware standby mode, turning all input pull-ups off. In software standby mode, the previous state is maintained.

Table 7-3 indicates the states of the input pull-up transistors in each operating mode.

Table 7-3 States of Input Pull-Up Transistors (Port 1)

Mode	Reset	Hardware Standby	Software Standby	Other Operating Modes
1	Off	Off	Off	Off
2	Off	Off	On/off	On/off
3	Off	Off	On/off	On/off

Notes: Off: The input pull-up transistor is always off.
On/off: The input pull-up transistor is on if P1PCR = 1 and P1DDR = 0, but offotherwise.

7.3 Port 2

7.3.1 Overview

Port 2 is an 8-bit input/output port with the pin configuration shown in figure 7-5. The pin functions differ depending on the operating mode.

Port 2 has built-in, software-controllable MOS input pull-up transistors that can be used in modes 2 and 3.

Pins in port 2 can drive one TTL load and a 90-pF capacitive load. They can also drive LEDs and darlington transistors.

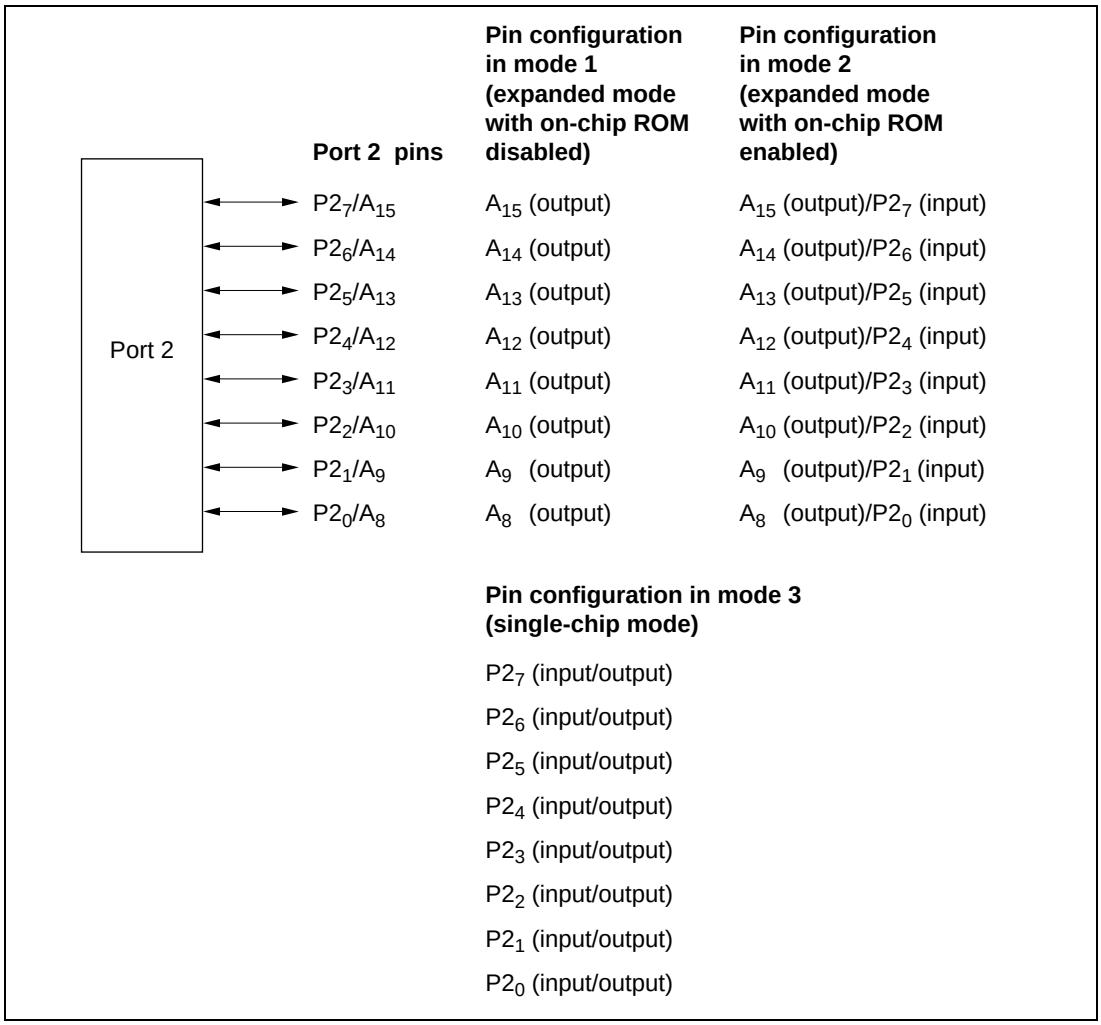


Figure 7-5 Port 2 Pin Configuration

7.3.2 Register Configuration and Descriptions

Table 7-4 summarizes the port 2 registers.

Table 7-4 Port 2 Registers

Name	Abbreviation	Read/Write	Initial Value	Address
Port 2 data direction register	P2DDR	W	H'FF (mode 1) H'00 (modes 2 and 3)	H'FFB1
Port 2 data register	P2DR	R/W	H'00	H'FFB3
Port 2 input pull-up control register	P2PCR	R/W	H'00	H'FFAD

Port 2 Data Direction Register (P2DDR)

Bit	7	6	5	4	3	2	1	0
	$P2_7DDR$	$P2_6DDR$	$P2_5DDR$	$P2_4DDR$	$P2_3DDR$	$P2_2DDR$	$P2_1DDR$	$P2_0DDR$
Mode 1								
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—
Modes 2 and 3								
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

P2DDR controls the input/output direction of each pin in port 2.

Mode 1: The P2DDR values are fixed at 1. Port 2 consists of upper address output pins. P2DDR values cannot be modified and are always read as 1.

In hardware standby mode, the address bus is in the high-impedance state.

Mode 2: A pin in port 2 is used for address output if the corresponding P2DDR bit is set to 1, and for general input if this bit is cleared to 0.

Mode 3: A pin in port 2 is used for general output if the corresponding P2DDR bit is set to 1, and for general input if this bit is cleared to 0.

In modes 2 and 3, P2DDR is a write-only register. Read data is invalid. If read, all bits always read 1. P2DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values, so if a transition to software standby mode occurs while a P2DDR bit is set to 1, the corresponding pin remains in the output state.

Port 2 Data Register (P2DR)

Bit	7	6	5	4	3	2	1	0
	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
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

P2DR is an 8-bit register that stores data for pins $P2_7$ to $P2_0$. When a P2DDR bit is set to 1, if port 2 is read, the value in P2DR is obtained directly, regardless of the actual pin state. When a P2DDR bit is cleared to 0, if port 2 is read the pin state is obtained.

P2DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

Port 2 Input Pull-Up Control Register (P2PCR)

Bit	7	6	5	4	3	2	1	0
	$P2_7$PCR	$P2_6$PCR	$P2_5$PCR	$P2_4$PCR	$P2_3$PCR	$P2_2$PCR	$P2_1$PCR	$P2_0$PCR
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

P2PCR is an 8-bit readable/writable register that controls the input pull-up transistors in port 2. If a P2DDR bit is cleared to 0 (designating input) and the corresponding P2PCR bit is set to 1, the input pull-up transistor is turned on.

P2PCR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

7.3.3 Pin Functions in Each Mode

Port 2 has different pin functions in different modes. A separate description for each mode is given below.

Pin Functions in Mode 1: In mode 1 (expanded mode with on-chip ROM disabled), port 2 is automatically used for upper address output (A_{15} to A_8). Figure 7-6 shows the pin functions in mode 1.

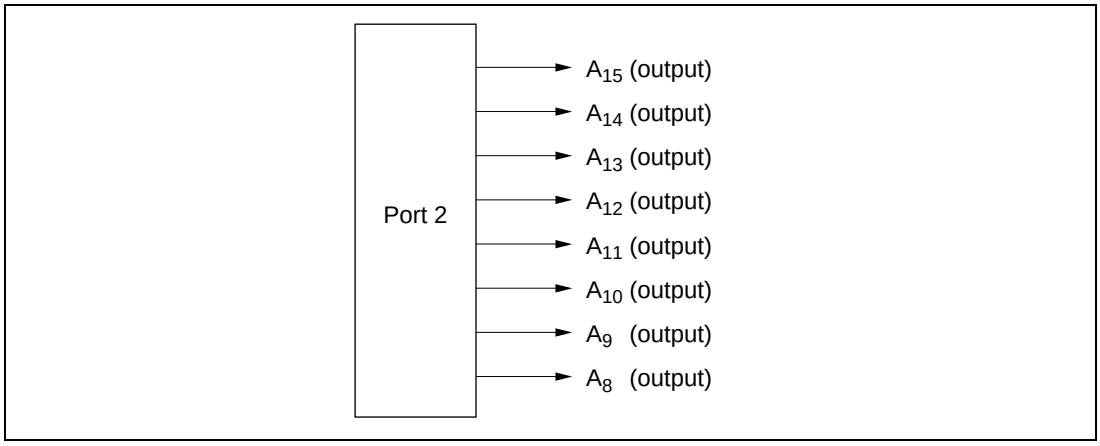


Figure 7-6 Pin Functions in Mode 1 (Port 2)

Mode 2: In mode 2 (expanded mode with on-chip ROM enabled), port 2 can provide upper address output pins and general input pins. Each pin becomes an upper address output pin if its P2DDR bit is set to 1, and a general input pin if this bit is cleared to 0. Following a reset, all pins are input pins. To be used for address output, their P2DDR bits must be set to 1. Figure 7-7 shows the pin functions in mode 2.

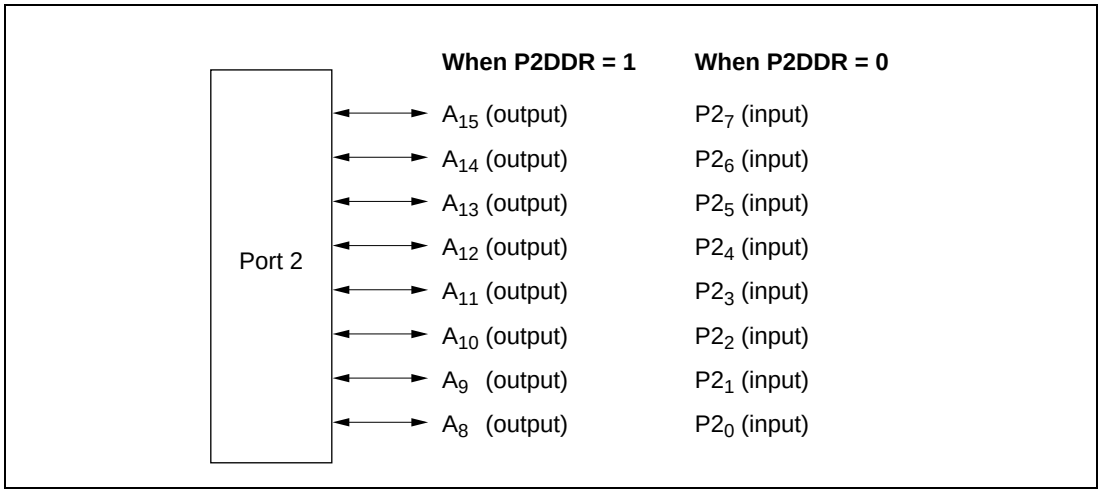


Figure 7-7 Pin Functions in Mode 2 (Port 2)

Mode 3: In mode 3 (single-chip mode), the input or output direction of each pin can be selected individually. A pin becomes a general input pin when its P2DDR bit is cleared to 0, and a general output pin when this bit is set to 1. Figure 7-8 shows the pin functions in mode 3.

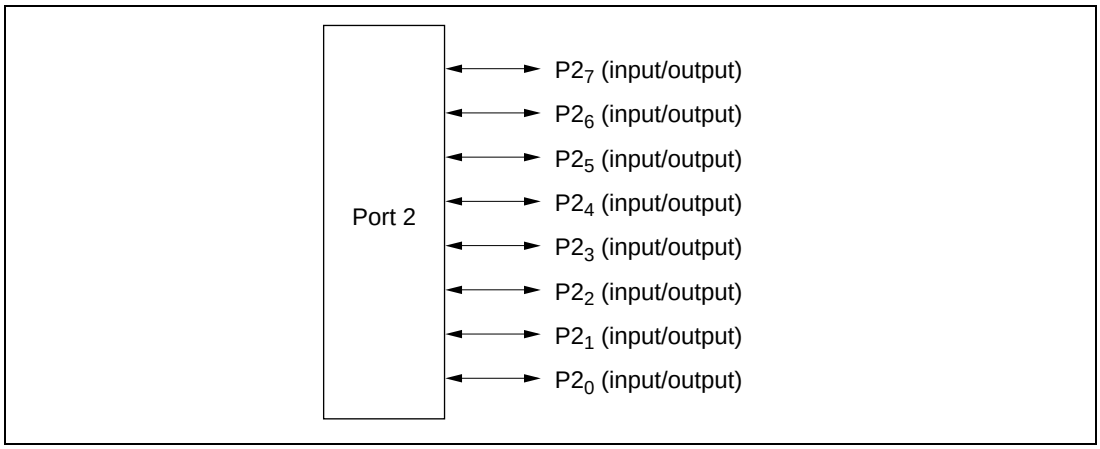


Figure 7-8 Pin Functions in Mode 3 (Port 2)

7.3.4 Input Pull-Up Transistors

Port 2 has built-in programmable input pull-up transistors that are available in modes 2 and 3. The pull-up for each bit can be turned on and off individually. To turn on an input pull-up in mode 2 or 3, set the corresponding P2PCR bit to 1 and clear the corresponding P2DDR bit to 0. P2PCR is cleared to H'00 by a reset and in hardware standby mode, turning all input pull-ups off. In software standby mode, the previous state is maintained.

Table 7-5 indicates the states of the input pull-up transistors in each operating mode.

Table 7-5 States of Input Pull-Up Transistors (Port 2)

Mode	Reset	Hardware Standby	Software Standby	Other Operating Modes
1	Off	Off	Off	Off
2	Off	Off	On/off	On/off
3	Off	Off	On/off	On/off

Notes: Off: The input pull-up transistor is always off.
On/off: The input pull-up transistor is on if P2PCR = 1 and P2DDR = 0, but off otherwise.

7.4 Port 3

7.4.1 Overview

Port 3 is an 8-bit input/output port that is multiplexed with the data bus. Figure 7-9 shows the pin configuration of port 3. The pin functions differ depending on the operating mode.

Port 3 has built-in, software-controllable MOS input pull-up transistors that can be used in mode 3.

Pins in port 3 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington pair.

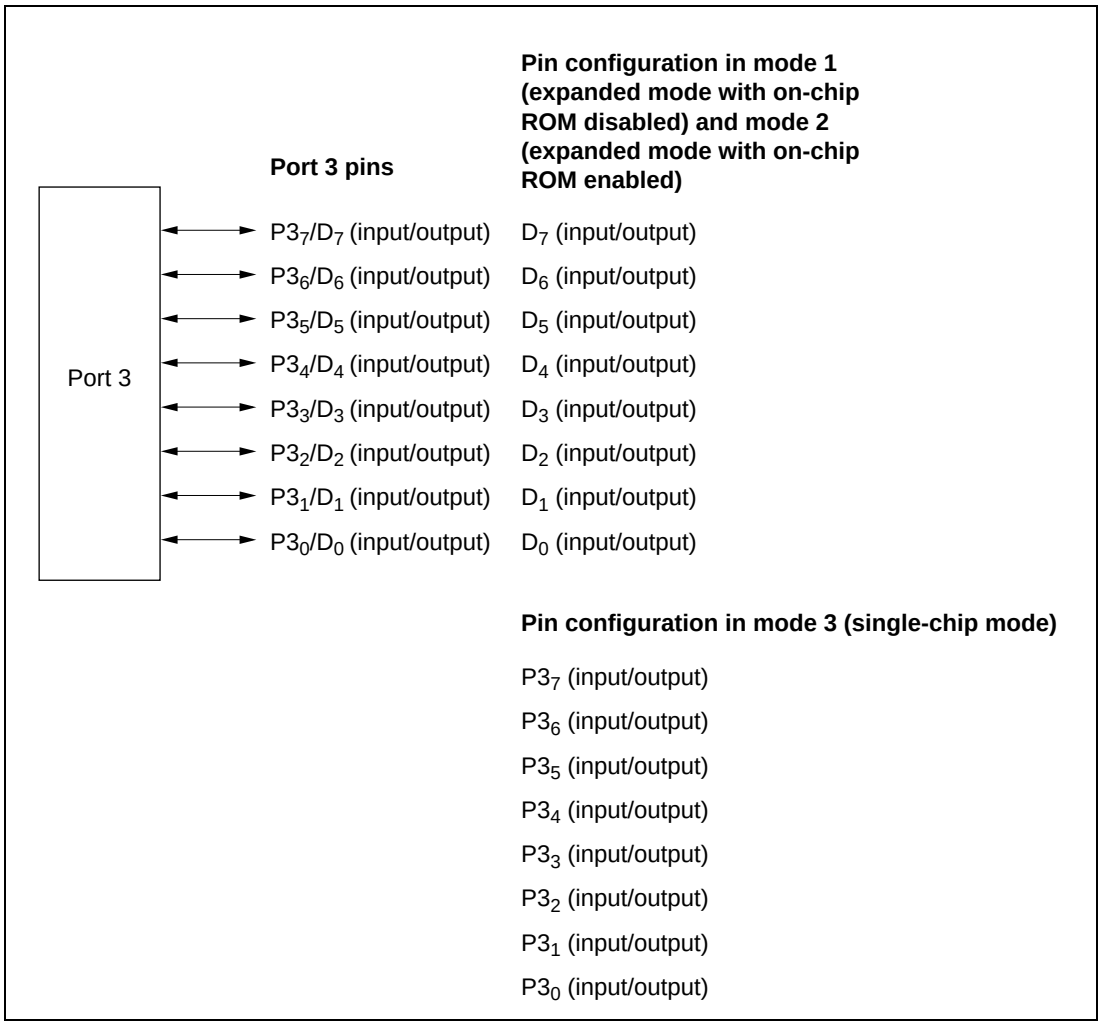


Figure 7-9 Port 3 Pin Configuration

7.4.2 Register Configuration and Descriptions

Table 7-6 summarizes the port 3 registers.

Table 7-6 Port 3 Registers

Name	Abbreviation	Read/Write	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 input pull-up control register	P3PCR	R/W	H'00	H'FFAE

Port 3 Data Direction Register (P3DDR)

Bit	7	6	5	4	3	2	1	0
	$P3_7DDR$	$P3_6DDR$	$P3_5DDR$	$P3_4DDR$	$P3_3DDR$	$P3_2DDR$	$P3_1DDR$	$P3_0DDR$
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 readable/writable register that controls the input/output direction of each pin in port 3. P3DDR is a write-only register. Read data is invalid. If read, all bits always read 1.

Modes 1 and 2: In mode 1 (expanded mode with on-chip ROM disabled) and mode 2 (expanded mode with on-chip ROM enabled), the input/output directions designated by P3DDR are ignored. Port 3 automatically consists of the input/output pins of the 8-bit data bus (D_7 to D_0).

The data bus is in the high-impedance state during reset, and during hardware and software standby.

Mode 3: A pin in port 3 is used for general output if the corresponding P3DDR bit is set to 1, and for general input if this bit is cleared to 0. P3DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values, so if a transition to software standby mode occurs while a P3DDR bit is set to 1, the corresponding pin remains in the output state.

Port 3 Data Register (P3DR)

Bit	7	6	5	4	3	2	1	0
	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
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

P3DR is an 8-bit register that stores data for pins $P3_7$ to $P3_0$. When a P3DDR bit is set to 1, if port 3 is read, the value in P3DR is obtained directly, regardless of the actual pin state. When a P3DDR bit is cleared to 0, if port 3 is read the pin state is obtained.

P3DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

Port 3 Input Pull-Up Control Register (P3PCR)

Bit	7	6	5	4	3	2	1	0
	$P3_7$PCR	$P3_6$PCR	$P3_5$PCR	$P3_4$PCR	$P3_3$PCR	$P3_2$PCR	$P3_1$PCR	$P3_0$PCR
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 input pull-up MOStransistors in port 3. If a P3DDR bit is cleared to 0 (designating input) and the corresponding P3PCR bit is set to 1, the input pull-up transistor is turned on.

P3PCR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

7.4.3 Pin Functions in Each Mode

Port 3 has different pin functions in different modes. A separate description for each mode is given below.

Pin Functions in Modes 1 and 2: In mode 1 (expanded mode with on-chip ROM disabled) and mode 2 (expanded mode with on-chip ROM enabled), port 3 is automatically used for the input/output pins of the 8-bit data bus (D_7 to D_0). Figure 7-10 shows the pin functions in modes 1 and 2.

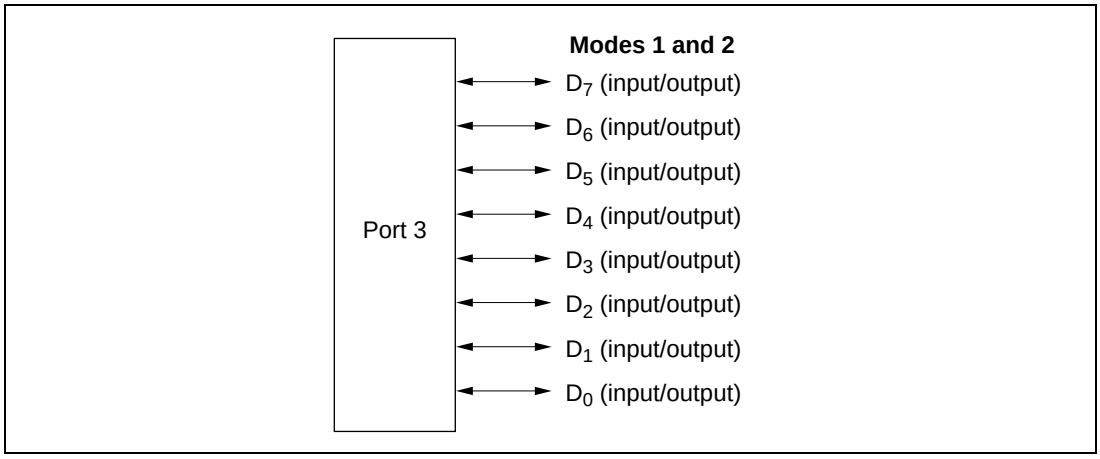


Figure 7-10 Pin Functions in Modes 1 and 2 (Port 3)

Mode 3: In mode 3 (single-chip mode), port 3 is a general-purpose input/output port. A pin becomes an output pin when its P3DDR bit is set to 1, and an input pin when this bit is cleared to 0.

Figure 7-11 shows the pin functions in mode 3.

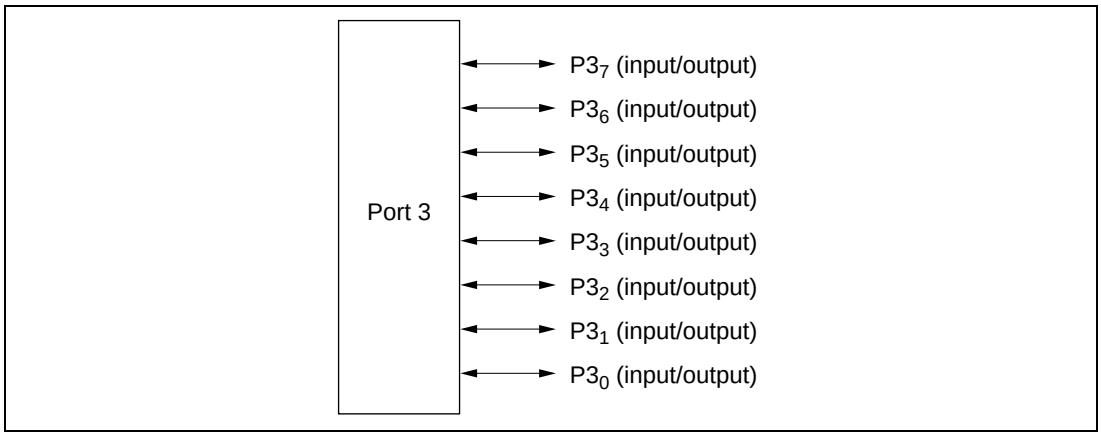


Figure 7-11 Pin Functions in Mode 3 (Port 3)

7.4.4 Input Pull-Up Transistors

Port 3 has built-in programmable input pull-up transistors that are available in mode 3. The pull-up for each bit can be turned on and off individually. To turn on an input pull-up in mode 3, set the corresponding P3PCR bit to 1 and clear the corresponding P3DDR bit to 0. P3PCR is cleared to H'00 by a reset and in hardware standby mode, turning all input pull-ups off. In software standby mode, the previous state is maintained.

Table 7-7 indicates the states of the input pull-up transistors in each operating mode.

Table 7-7 States of Input Pull-Up Transistors (Port 3)

Mode	Reset	Hardware Standby	Software Standby	Other Operating Modes
1	Off	Off	Off	Off
2	Off	Off	Off	Off
3	Off	Off	On/off	On/off

Notes: Off: The input pull-up transistor is always off.
On/off: The input pull-up transistor is on if P3PCR = 1 and P3DDR = 0, but off otherwise.

7.5 Port 4

7.5.1 Overview

Port 4 is an 8-bit input/output port that is multiplexed with input/output pins ($TMRI_0$, $TMRI_1$, $TMCI_0$, $TMCI_1$, TMO_0, TMO_1) of 8-bit timers 0 and 1 and output pins (PW_0, PW_1) of PWM timers 0 and 1. Port 4 pin functions are the same in all modes.

Figure 7-12 shows the pin configuration of port 4.

Pins in port 4 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington pair.

Port 4 in the H8/3522 has the same functions as port 9 in the H8/3534. For details, see section 7.10, Port 9 (H8/3534)/Port 4 (H8/3522).

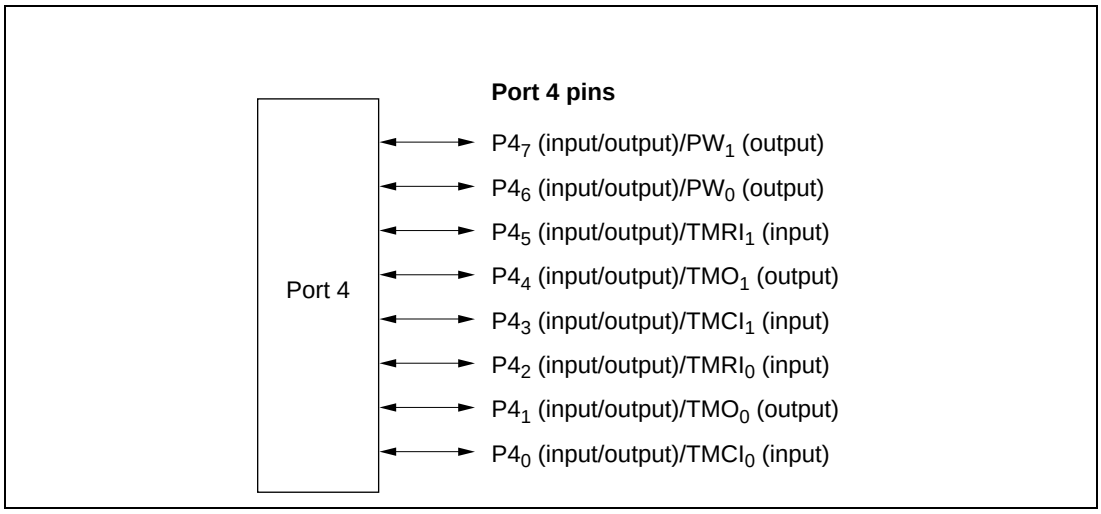


Figure 7-12 Port 4 Pin Configuration

7.5.2 Register Configuration and Descriptions

Table 7-8 summarizes the port 4 registers.

Table 7-8 Port 4 Registers

Name	Abbreviation	Read/Write	Initial Value	Address
Port 4 data direction register	P4DDR	W	H'00	H'FFB5
Port 4 data register	P4DR	R/W	H'00	H'FFB7

Port 4 Data Direction Register (P4DDR)

Bit	7	6	5	4	3	2	1	0
	$P4_7DDR$	$P4_6DDR$	$P4_5DDR$	$P4_4DDR$	$P4_3DDR$	$P4_2DDR$	$P4_1DDR$	$P4_0DDR$
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 readable/writable register that controls the input/output direction of each pin in port 4. A pin functions as an output pin if the corresponding P4DDR bit is set to 1, and as an input pin if this bit is cleared to 0.

P4DDR is a write-only register. Read data is invalid. If read, all bits always read 1.

P4DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values, so if a transition to software standby mode occurs while a P4DDR bit is set to 1, the corresponding pin remains in the output state.

If a transition to software standby mode occurs while port 4 is being used by an on-chip supporting module (for example, for 8-bit timer output), the on-chip supporting module will be initialized, so the pin will revert to general-purpose input/output, controlled by P4DDR and P4DR.

Port 4 Data Register (P4DR)

Bit	7	6	5	4	3	2	1	0
	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
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

P4DR is an 8-bit register that stores data for pins $P4_7$ to $P4_0$. When a P4DDR bit is set to 1, if port 4 is read, the value in P4DR is obtained directly, regardless of the actual pin state. When a P4DDR bit is cleared to 0, if port 4 is read the pin state is obtained. This also applies to pins used by on-chip supporting modules.

P4DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

7.5.3 Pin Functions

Port 4 has different pin functions depending on whether the chip is or is not operating in slave mode. Table 7-9 indicates the pin functions of port 4.

Table 7-9 Port 4 Pin Functions

Pin	Pin Functions and Selection Method				
$P4_7/PW_1$	Bit OE in TCR of PWM timer 1 and bit $P4_7$DDR select the pin function as follows				
	OE	0		1	
	$P4_7$DDR	0	1	0	1
	Pin function	$P4_7$ input	$P4_7$ output	PW_1 output	
$P4_6/PW_0$	Bit OE in TCR of PWM timer 0 and bit $P4_6$DDR select the pin function as follows				
	OE	0		1	
	$P4_6$DDR	0	1	0	1
	Pin function	$P4_6$ input	$P4_6$ output	PW_0 output	
$P4_5/TMRI_1$					
	$P4_5$DDR	0		1	
	Pin function	$P4_5$ input		$P4_5$ output	
		$TMRI_1$ input			
	$TMRI_1$ input is usable when bits CCLR1 and CCLR0 are both set to 1 in TCR of 8-bit timer 1				
$P4_4/TMO_1$	Bits OS3 to OS0 in TCSR of 8-bit timer 1, bit $P4_4$DDR, and the operating mode select the pin function as follows				
	OS3 to 0	All 0		Not all 0	
	$P4_4$DDR	0	1	—	
	Pin function	$P4_4$ input	$P4_4$ output	TMO_1 output	

Table 7-9 Port 4 Pin Functions (cont)

Pin	Pin Functions and Selection Method		
$P4_3/TMCI_1$			
	$P4_3DDR$	0	1
	Pin function	$P4_3$ input	$P4_3$ output
		$TMCI_1$ input	
	$TMCI_1$ input is usable when bits CKS2 to CKS0 in TCR of 8-bit timer 1 select an external clock source		
$P4_2/TMRI_0$			
	$P4_2DDR$	0	1
	Pin function	$P4_2$ input	$P4_2$ output
		$TMRI_0$ input	
	$TMRI_0$ input is usable when bits CCLR1 and CCLR0 are both set to 1 in TCR of 8-bit timer 0		
$P4_1/TMO_0$	Bits OS3 to OS0 in TCSR of 8-bit timer 0 and bit $P4_1DDR$ select the pin function as follows		
	OS3 to 0	All 0	Not all 0
	$P4_1DDR$	0 1	—
	Pin function	$P4_1$ input $P4_1$ output	TMO_0 output
$P4_0/TMCI_0$			
	$P4_0DDR$	0	1
	Pin function	$P4_0$ input	$P4_0$ output
		$TMCI_0$ input	
	$TMCI_0$ input is usable when bits CKS2 to CKS0 in TCR of 8-bit timer 0 select an external clock source		

7.6 Port 5

7.6.1 Overview

Port 5 is a 3-bit input/output port that is multiplexed with serial communication interface 0 input/output pins (TxD_0, RxD_0, SCK_0 in the H8/3534; TxD, RxD, SCK in the H8/3522). The port 5 pin functions are the same in all operating modes.

Pins in port 5 can drive one TTL load and a 30-pF capacitive load. They can also drive a darlington pair.

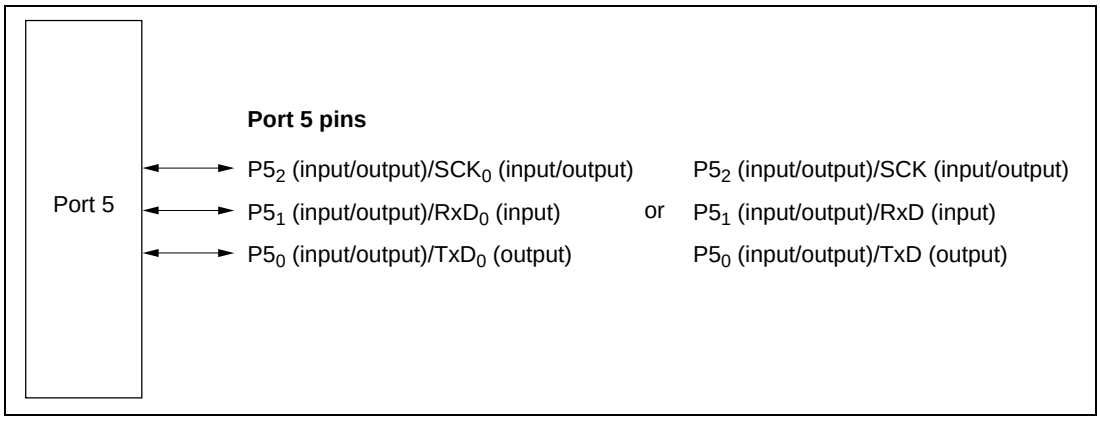


Figure 7-13 Port 5 Pin Configuration

7.6.2 Register Configuration and Descriptions

Table 7-10 summarizes the port 5 registers.

Table 7-10 Port 5 Registers

Name	Abbreviation	Read/Write	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

Port 5 Data Direction Register (P5DDR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	W	W	W

P5DDR is an 8-bit register that controls the input/output direction of each pin in port 5. A pin functions as an output pin if the corresponding P5DDR bit is set to 1, and as an input pin if this bit is cleared to 0.

P5DDR is a write-only register. Read data is invalid. Bits 7 to 3 are reserved. If read, these bits always read 1.

P5DDR is initialized to H'F8 by a reset and in hardware standby mode. In software standby mode it retains its existing values, so if a transition to software standby mode occurs while a P5DDR bit is set to 1, the corresponding pin remains in the output state.

If a transition to software standby mode occurs while port 5 is being used by the SCI, the SCI will be initialized, so the pin will revert to general-purpose input/output, controlled by P5DDR and P5DR.

Port 5 Data Register (P5DR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	$P5_2$	$P5_1$	$P5_0$
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

P5DR is an 8-bit register that stores data for pins $P5_2$ to $P5_0$. Bits 7 to 3 are reserved. They cannot be modified, and are always read as 1.

When a P5DDR bit is set to 1, if port 5 is read, the value in P5DR is obtained directly, regardless of the actual pin state. When a P5DDR bit is cleared to 0, if port 5 is read the pin state is obtained. This also applies to pins used as SCI pins.

P5DR is initialized to H'F8 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

7.6.3 Pin Functions

Port 5 has the same pin functions in each operating mode. All pins can also be used as SCI input/output pins. Table 7-11 indicates the pin functions of port 5.

Table 7-11 Port 5 Pin Functions

Pin	Pin Functions and Selection Method				
$P5_2/SCK_0$ or $P5_2/SCK$	Bit C/$\overline{A}$ in SMR of SCI0, bits CKE0 and CKE1 in SCR of SCI0, and bit $P5_2$DDR select the pin function as follows				
	CKE1	0			1
	C/$\overline{A}$	0		1	—
	CKE0	0	1	—	—
	$P5_2$DDR	0	1	—	—
	Pin function	$P5_2$ input	$P5_2$ output	SCK_0 (SCK) output	SCK_0 (SCK) output	SCK_0 (SCK) input
$P5_1/RxD_0$ or $P5_1/RxD$	Bit RE in SCR of SCI0 and bit $P5_1$DDR select the pin function as follows				
	RE	0		1	
	$P5_1$DDR	0	1	—	
	Pin function	$P5_1$ input	$P5_1$ output	RxD_0 (RxD) input	
$P5_0/TxD_0$ or $P5_0/TxD$	Bit TE in SCR of SCI0 and bit $P5_0$DDR select the pin function as follows				
	TE	0		1	
	$P5_0$DDR	0	1	—	
	Pin function	$P5_0$ input	$P5_0$ output	TxD_0 (TxD) output	

7.7 Port 6

7.7.1 Overview

Port 6 is an 8-bit input/output port that is multiplexed with input/output pins (FTOA, FTOB, FTIA to FTID, FTCI) of the 16-bit free-running timer (FRT), with key-sense input pins ($\overline{KEYIN}_0$ to $\overline{KEYIN}_7$) [H8/3534], with $\overline{IRQ}_6$ and $\overline{IRQ}_7$ input pins [H8/3534], and input/output pins ($TMCI_0$, $TMRI_0$, TMO_0, $TMCI_1$, $TMRI_1$, TMO_1) of 8-bit timer 0 and timer 1 [H8/3522]. The port 6 pin functions are the same in all operating modes. Figure 7-14 shows the pin configuration of port 6.

Port 6 has built-in, software-controllable MOS input pull-up transistors. [H8/3534]

Pins in port 6 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington pair.

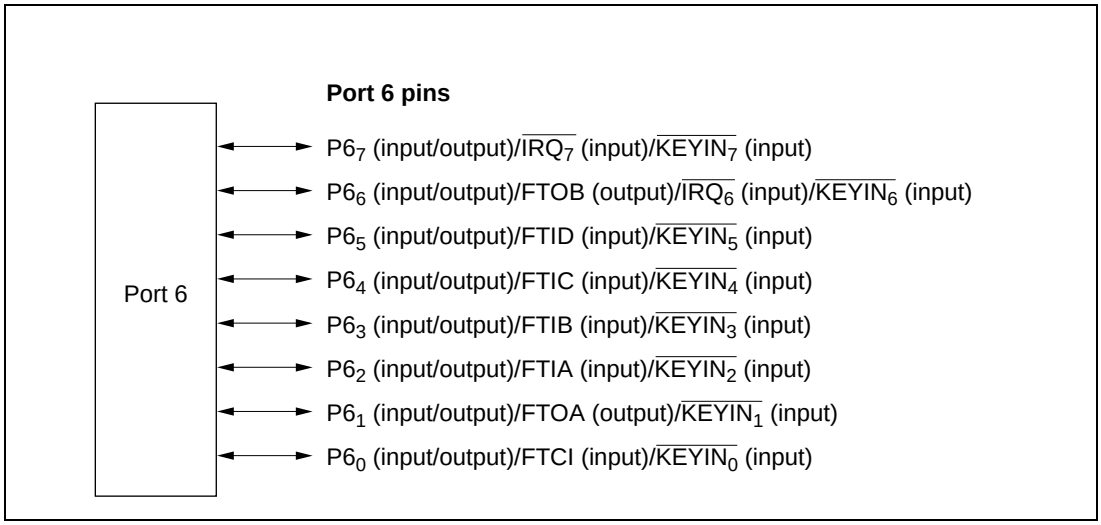


Figure 7-14 (a) H8/3534 Port 6 Pin Configuration

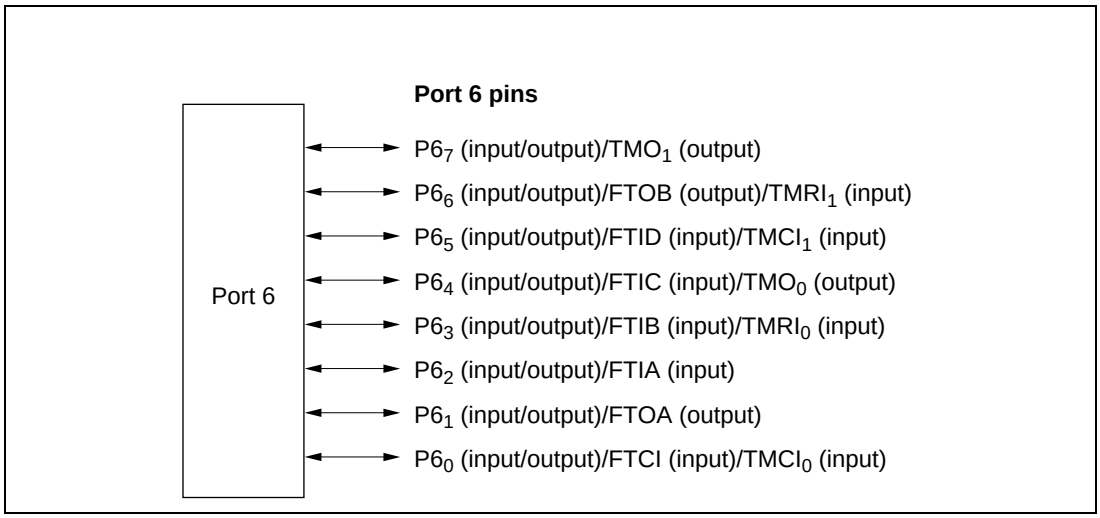


Figure 7-14 (b) H8/3522 Port 6 Pin Configuration

7.7.2 Register Configuration and Descriptions

Table 7-12 summarizes the port 6 registers.

Table 7-12 (a) H8/3534 Port 6 Registers

Name	Abbreviation	Read/Write	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
Port 6 input pull-up control register	KMPCR	R/W	H'00	H'FFF2

Table 7-12 (b) H8/3522 Port 6 Registers

Name	Abbreviation	Read/Write	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

Port 6 Data Direction Register (P6DDR)

Bit	7	6	5	4	3	2	1	0
	$P6_7DDR$	$P6_6DDR$	$P6_5DDR$	$P6_4DDR$	$P6_3DDR$	$P6_2DDR$	$P6_1DDR$	$P6_0DDR$
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 readable/writable register that controls the input/output direction of each pin in port 6. A pin functions as an output pin if the corresponding P6DDR bit is set to 1, and as an input pin if this bit is cleared to 0.

P6DDR is a write-only register. Read data is invalid. If read, all bits always read 1.

P6DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values, so if a transition to software standby mode occurs while a P6DDR bit is set to 1, the corresponding pin remains in the output state.

If a transition to software standby mode occurs while port 6 is being used by the free-running timer, or other timer the timer will be initialized, so the pin will revert to general-purpose input/output, controlled by P6DDR and P6DR.

Port 6 Data Register (P6DR)

Bit	7	6	5	4	3	2	1	0
	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
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

P6DR is an 8-bit register that stores data for pins $P6_7$ to $P6_0$. When a P6DDR bit is set to 1, if port 6 is read, the value in P6DR is obtained directly, regardless of the actual pin state. When a P6DDR bit is cleared to 0, if port 6 is read the pin state is obtained. This also applies to pins used as FRT and timer pins.

P6DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

Port 6 Input Pull-Up Control Register (KMPCR) [H8/3534]

Bit	7	6	5	4	3	2	1	0
	KM_7PCR	KM_6PCR	KM_5PCR	KM_4PCR	KM_3PCR	KM_2PCR	KM_1PCR	KM_0PCR
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

KMPCR is an 8-bit readable/writable register that controls the input pull-up transistors in port 6. If a P6DDR bit is cleared to 0 (designating input) and the corresponding KMPCR bit is set to 1, the input pull-up transistor is turned on.

KMPCR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

7.7.3 Pin Functions

Port 6 has the same pin functions in all operating modes. The pins are multiplexed with FRT input/output, $\overline{IRQ}_6$ and $\overline{IRQ}_7$ input [H8/3534], key-sense input, and input/output of 8-bit timer 0 and timer 1 [H8/3522]. Table 7-13 indicates the pin functions of port 6.

Table 7-13 (a) H8/ 3534 Port 6 Pin Functions

Pin	Pin Functions and Selection Method				
$P6_7/\overline{IRQ}_7/\overline{KEYIN}_7$					
	$P6_7$DDR	0		1	
	Pin function	$P6_7$ input		$P6_7$ output	
		$\overline{IRQ}_7$ input or $\overline{KEYIN}_7$ input			
	$\overline{IRQ}_7$ input is usable when bit IRQ7E is set to 1 in IER				
$P6_6$/FTOB/ $\overline{IRQ}6/\overline{KEYIN}6$	Bit OEB in TOCR of the FRT and bit $P6_6$DDR select the pin function as follows				
	OEB	0		1	
	$P6_6$DDR	0	1	0	1
	Pin function	$P6_6$ input	$P6_6$ output	FTOB output	
		$\overline{IRQ}_6$ input or $\overline{KEYIN}_6$ input			
	$\overline{IRQ}_6$ input is usable when bit IRQ6E is set to 1 in IER				
$P6_5$/FTID/					
$\overline{KEYIN}_5$	$P6_5$DDR	0		1	
	Pin function	$P6_5$ input		$P6_5$ output	
		FTID input or $\overline{KEYIN}_5$ input			
$P6_4$/FTIC/					
$\overline{KEYIN}_4$	$P6_4$DDR	0		1	
	Pin function	$P6_4$ input		$P6_4$ output	
		FTIC input or $\overline{KEYIN}_4$ input			

Table 7-13 (a) H8/ 3534 Port 6 Pin Functions (cont)

Pin	Pin Functions and Selection Method				
$P6_3$/FTIB/					
$\overline{KEYIN}_3$	$P6_3DDR$	0		1	
	Pin function	$P6_3$ input		$P6_3$ output	
		FTIB input or $\overline{KEYIN}_3$ input			
$P6_2$/FTIA/					
$\overline{KEYIN}_2$	$P6_2DDR$	0		1	
	Pin function	$P6_2$ input		$P6_2$ output	
		FTIA input or $\overline{KEYIN}_2$ input			
$P6_1$/FTOA/ $\overline{KEYIN}_1$	Bit OEA in TOCR of the FRT and bit $P6_1DDR$ select the pin function as follows				
	OEA	0		1	
	$P6_1DDR$	0	1	0	1
	Pin function	$P6_1$ input	$P6_1$ output	FTOA output	
		$\overline{KEYIN}_1$ input			
$P6_0$/FTCI/					
$\overline{KEYIN}_0$	$P6_0DDR$	0		1	
	Pin function	$P6_0$ input		$P6_0$ output	
		FTCI input or $\overline{KEYIN}_0$ input			
	FTCI input is usable when bits CKS1 and CKS0 in TCR of the FRT select an external clock source				

Table 7-13 (b) H8/3522 Port 6 Pin Functions

Pin	Pin Functions and Selection Method			
$P6_7$/TMO_1	Bits OS3 to OS0 in TCSR of 8-bit timer 1 and bit $P6_7$DDR select the pin function as follows			
	OS3 to OS0	All 0		Not all 1
	$P6_7$DDR	0	1	—
	Pin function	$P6_7$ input	$P6_7$ output	TMO_1 output
$P6_6$/FTOB/ $TMRI_1$	Bit OEB in TOCR of the FRT and bit $P6_6$DDR select the pin function as follows			
	OEB	0		1
	$P6_6$DDR	0	1	—
	Pin function	$P6_6$ input	$P6_6$ output	FTOB output
		$TMRI_1$ input		
	$TMRI_1$ input is usable when bit CCLR1 or CCLR0 is set to 1 in TCR of 8-bit timer 1			
$P6_5$/FTID/$TMCI_1$				
	$P6_5$DDR	0		1
	Pin function	$P6_5$ input		$P6_5$ output
		FTID input, $TMCI_1$ input		
	$TMCI_1$ input is usable when an external clock is selected by bits CKS2 to CKS0 in TCSR of 8-bit timer 1			
$P6_4$/FTIC/ TMO_0	Bits OS3 to OS0 in TCSR of 8-bit timer 0 and bit $P6_4$DDR select the pin function as follows			
	OS3 to OS0	All 0		Not all 1
	$P6_4$DDR	0	1	—
	Pin function	$P6_4$ input	$P6_4$ output	TMO_0 output
		FTIC input		

Table 7-13 (b) H8/3522 Port 6 Pin Functions (cont)

Pin	Pin Functions and Selection Method			
$P6_3$/FTIB/$TMRI_0$				
	$P6_3$DDR	0		1
	Pin function	$P6_3$ input		$P6_3$ output
		FTIB input, $TMRI_0$ input		
	$TMRI_0$ input is usable when bit CCLR1 or CCLR0 is set to 1 in TCR of 8-bit timer 0			
$P6_2$/FTIA				
	$P6_2$DDR	0		1
	Pin function	$P6_2$ input		$P6_2$ output
	FTIA input			
$P6_1$/FTOA	Bit OEA in TOCR of the free-running timer and bit $P6_1$DDR select the pin function as follows			
	OEA	0		1
	$P6_1$DDR	0	1	—
	Pin function	$P6_1$ input	$P6_1$ output	FTOA output
$P6_0$/FTCI/$TMRI_0$				
	$P6_0$DDR	0		1
	Pin function	$P6_0$ input		$P6_0$ output
		FTCI input, $TMCI_0$ input		
	FTCI input is usable when an external clock is selected by bits CKS1 and CKS0 in TCR of the free-running timer $TMCI_0$ input is usable when an external clock is selected by bits CKS2 to CKS0 in TCR of 8-bit timer 0			

7.7.4 Input Pull-Up Transistors [H8/3534]

Port 6 has built-in programmable input pull-up transistors. The pull-up for each bit can be turned on and off individually. To turn on an input pull-up, set the corresponding KMPCR bit to 1 and clear the corresponding P6DDR bit to 0. KMPCR is cleared to H'00 by a reset and in hardware standby mode, turning all input pull-ups off. In software standby mode, the previous state is maintained.

Table 7-14 indicates the states of the input pull-up transistors in each operating mode.

Table 7-14 States of Input Pull-Up Transistors (Port 6)

Mode	Reset	Hardware Standby	Software Standby	Other Operating Modes
1	Off	Off	On/off	On/off
2	Off	Off	On/off	On/off
3	Off	Off	On/off	On/off

Notes: Off: The input pull-up transistor is always off.
On/off: The input pull-up transistor is on if KMPCR = 1 and P6DDR = 0, but off otherwise.

7.8 Port 7

7.8.1 Overview

Port 7 is an 8-bit input port that also provides the analog input pins for the A/D converter. The pin functions are the same in all modes. Figure 7-15 shows the pin configuration of port 7.

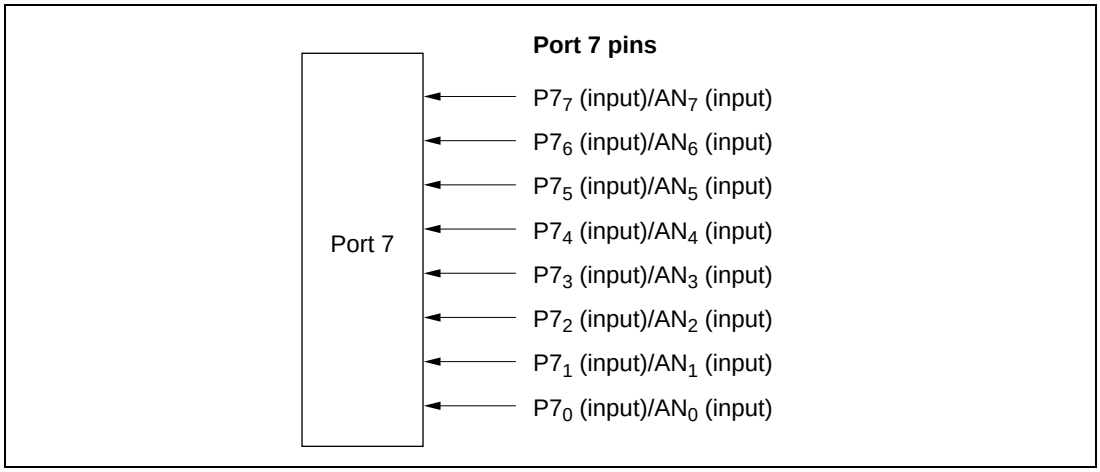


Figure 7-15 Port 7 Pin Configuration

7.8.2 Register Configuration and Descriptions

Table 7-15 summarizes the port 7 registers. Port 7 is a dedicated input port, and has no data direction register.

Table 7-15 Port 7 Register

Name	Abbreviation	Read/Write	Initial Value	Address
Port 7 input data register	P7PIN	R	Undetermined	H'FFBE

Port 7 Input Data Register (P7PIN)

Bit	7	6	5	4	3	2	1	0
	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Note: * Depends on the levels of pins $P7_7$ to $P7_0$.

When P7PIN is read, the pin states are always read.

P7PIN is a read-only register and cannot be modified.

7.9 Port 8 [H8/3534]

7.9.1 Overview

Port 8 is a 7-bit input/output port that is multiplexed with input/output pins (TxD_1, RxD_1, SCK_1) of serial communication interface 1, and with interrupt input pins (IRQ_5 to IRQ_3).

Port 8 pin functions are the same in all operating modes.

Figure 7-16 shows the pin configuration of port 8.

Pins in port 8 can drive one TTL load and a 30-pF capacitive load. They can also drive a darlington pair.

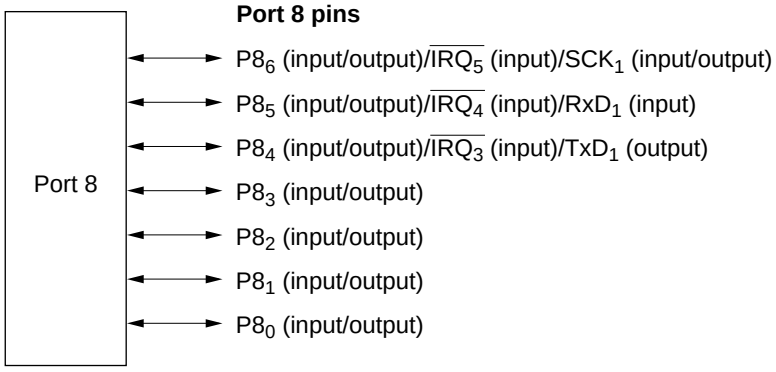


Figure 7-16 Port 8 Pin Configuration

7.9.2 Register Configuration and Descriptions

Table 7-16 summarizes the port 8 registers.

Table 7-16 Port 8 Registers

Name	Abbreviation	Read/Write	Initial Value	Address
Port 8 data direction register	P8DDR	W	H'80	H'FFBD
Port 8 data register	P8DR	R/W	H'80	H'FFBF

Port 8 Data Direction Register (P8DDR)

	7	6	5	4	3	2	1	0
Bit	—	$P8_6DDR$	$P8_5DDR$	$P8_4DDR$	$P8_3DDR$	$P8_2DDR$	$P8_1DDR$	$P8_0DDR$
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	W	W	W	W	W	W	W

P8DDR is an 8-bit readable/writable register that controls the input/output direction of each pin in port 8. A pin functions as an output pin if the corresponding P8DDR bit is set to 1, and as an input pin if this bit is cleared to 0. P8DDR is a write-only register. Read data is invalid. If read, all bits always read 1. Bit 7 is a reserved bit that always reads 1.

P8DDR is initialized to H'80 by a reset and in hardware standby mode. In software standby mode P8DDR retains its existing values, so if a transition to software standby mode occurs while a P8DDR bit is set to 1, the corresponding pin remains in the output state.

Port 8 Data Register (P8DR)

Bit	7	6	5	4	3	2	1	0
	—	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P8DR is an 8-bit register that stores data for pins $P8_6$ to $P8_0$. Bit 7 is a reserved bit that always reads 1.

When a P8DDR bit is set to 1, if port 8 is read, the value in P8DR is obtained directly, regardless of the actual pin state. When a P8DDR bit is cleared to 0, if port 8 is read the pin state is obtained. This also applies to pins used by on-chip supporting modules.

P8DR is initialized to H'80 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

7.9.3 Pin Functions

Pins $P8_6$ to $P8_4$ are multiplexed with SCI1 input/output and IRQ_5 to IRQ_3 input. Table 7-17 indicates the functions of pins $P8_6$ to $P8_4$.

Table 7-17 Pin Functions of Port 8 Pins $P8_6$ to $P8_4$

Pin	Pin Functions and Selection Method					
$P8_6/\overline{IRQ_5}/SCK_1$	Bit C/$\overline{A}$ in SMR of SCI1, bits CKE0 and CKE1 in SCR of SCI1, and bit $P8_6$DDR select the pin function as follows					
	CKE1	0				1
	C/$\overline{A}$	0			1	—
	CKE0	0		1	—	—
	$P8_6$DDR	0	1	—	—	—
	Pin function	$P8_6$ input	$P8_6$ output	SCK_1 output	SCK_1 output	SCK_1 intput
		$\overline{IRQ_5}$ input				
	$\overline{IRQ_5}$ input is usable when bit IRQ5E is set to 1 in IER					
$P8_5/\overline{IRQ_4}/RxD_1$	Bit RE in SCR of SCI1, and bit $P8_5$DDR select the pin function as follows					
	RE	0			1	
	$P8_5$DDR	0	1		—	
	Pin function	$P8_5$ input	$P8_5$ output		RxD_1 input	
		$\overline{IRQ_4}$ input				
	$\overline{IRQ_4}$ input is usable when bit IRQ4E is set to 1 in IER					
$P8_4/\overline{IRQ_3}/TxD_1$	Bit TE in SCR of SCI1, and bit $P8_4$DDR select the pin function as follows					
	TE	0			1	
	$P8_4$DDR	0	1		—	
	Pin function	$P8_4$ input	$P8_4$ output		TxD_1 output	
		$\overline{IRQ_3}$ input				
	$\overline{IRQ_3}$ input is usable when bit IRQ3E is set to 1 in IER					

7.10 Port 9 [H8/3534] · Port 4 [H8/3522]

7.10.1 Overview

Port 9 in the H8/3534 and port 4 in the H8/3522 have the same functions except that the DR and DDR addresses are different. Port 9 in the H8/3534 is described below.

Port 9 is an 8-bit input/output port that is multiplexed with interrupt input pins ($\overline{IRQ_0}$ to $\overline{IRQ_2}$), input/output pins for bus control signals ($\overline{RD}$, $\overline{WR}$, $\overline{AS}$, $\overline{WAIT}$), an input pin ($\overline{ADTRG}$) for the A/D converter, and an output pin (ϕ) for the system clock. The function of pins $P9_2$ to $P9_0$ is the same in all operating modes, while the function of pins $P9_7$ to $P9_3$ depends on the operating mode. Figure 7-17 shows the pin configuration of port 9.

Pins in port 9 can drive one TTL load and a 90-pF capacitive load. They can also drive a darlington pair.

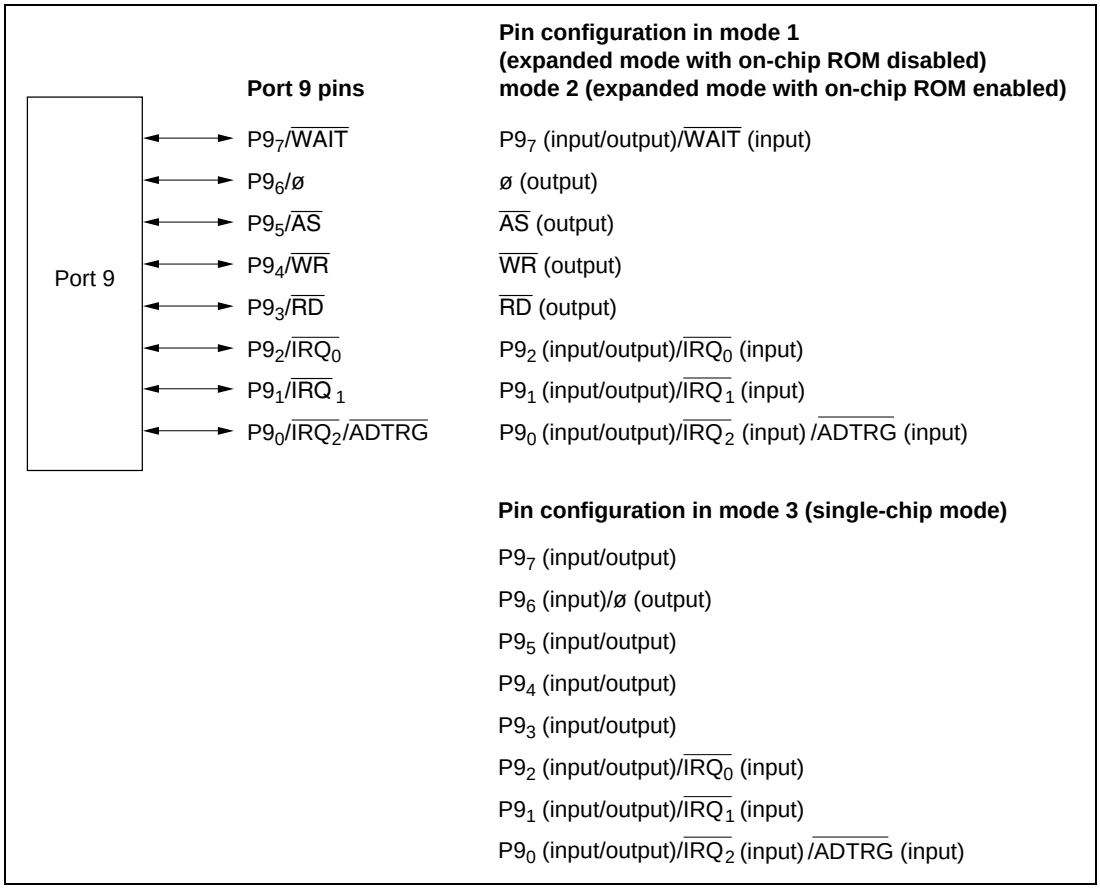


Figure 7-17 Port 9 Pin Configuration

7.10.2 Register Configuration and Descriptions

Table 7-18 summarizes the port 9 registers.

Table 7-18 (a) H8/3534 Port 9 Registers

Name	Abbreviation	Read/Write	Initial Value	Address
Port 9 data direction register	P9DDR	W	H'40 (modes 1 and 2) H'00 (mode 3)	H'FFC0
Port 9 data register	P9DR	R/W*1	Undetermined*2	H'FFC1

Table 7-18 (b) H8/3522 Port 4 Registers

Name	Abbreviation	Read/Write	Initial Value	Address
Port 4 data direction register	P4DDR	W	H'40 (modes 1 and 2) H'00 (mode 3)	H'FFB5
Port 4 data register	P4DR	R/W*1	Undetermined*2	H'FFB7

Notes: 1. Bit 6 is read-only.
2. Bit 6 is undetermined. Other bits are initially 0.

Port 9 Data Direction Register (P9DDR)

Bit	7	6	5	4	3	2	1	0
	$P9_7DDR$	$P9_6DDR$	$P9_5DDR$	$P9_4DDR$	$P9_3DDR$	$P9_2DDR$	$P9_1DDR$	$P9_0DDR$
Modes 1, 2								
Initial value	0	1	0	0	0	0	0	0
Read/Write	W	—	W	W	W	W	W	W
Mode 3								
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

P9DDR is an 8-bit readable/writable register that controls the input/output direction of each pin in port 9. A pin functions as an output pin if the corresponding P9DDR bit is set to 1, and as an input pin if this bit is cleared to 0. In modes 1 and 2, $P9_6DDR$ is fixed at 1 and cannot be modified.

P9DDR is a write-only register. Read data is invalid. If read, all bits always read 1.

P9DDR is initialized by a reset and in hardware standby mode. The initial value is H'40 in modes 1 and 2, and H'00 in mode 3. In software standby mode P9DDR retains its existing values, so if a transition to software standby mode occurs while a P9DDR bit is set to 1, the corresponding pin remains in the output state.

Port 9 Data Register (P9DR)

Bit	7	6	5	4	3	2	1	0
	$P9_7$	$P9_6$	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
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 level at pin $P9_6$.

P9DR is an 8-bit register that stores data for pins $P9_7$ to $P9_0$. When a P9DDR bit is set to 1, if port 9 is read, the value in P9DR is obtained directly, regardless of the actual pin state, except for $P9_6$. When a P9DDR bit is cleared to 0, if port 9 is read the pin state is obtained. This also applies to pins used by on-chip supporting modules and for bus control signals. $P9_6$ always returns the pin state.

Except for bit $P9_6$, P9DR bits are initialized to 0 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

7.10.3 Pin Functions

Port 9 has one set of pin functions in modes 1 and 2, and a different set of pin functions in mode 3. The pins are multiplexed with IRQ_0 to IRQ_2 input, bus control signal input/output, A/D converter input and system clock (ϕ) output. Table 7-19 indicates the pin functions of port 9.

Table 7-19 Port 9 Pin Functions

Pin	Pin Functions and Selection Method				
$P9_7/\overline{WAIT}$	Bit $P9_7$DDR, the wait mode as determined by WSCR, and the operating mode select the pin function as follows				
	Operating mode	Modes 1 and 2		Mode 3	
	Wait mode	$\overline{WAIT}$ used	$\overline{WAIT}$ not used	—	
	$P9_7$DDR	—	0	1	0	1
	Pin function	$\overline{WAIT}$ input pin	$P9_7$ input pin	$P9_7$ output pin	$P9_7$ input pin	$P9_7$ output pin
$P9_6/\phi$	Bit $P9_6$DDR and the operating mode select the pin function as follows				
	Operating mode	Modes 1 and 2		Mode 3	
	$P9_6$DDR	Always 1		0	1
	Pin function	ϕ output		$P9_6$ input	ϕ output
$P9_5/\overline{AS}$	Bit $P9_5$DDR and the operating mode select the pin function as follows				
	Operating mode	Modes 1 and 2		Mode 3	
	$P9_5$DDR	—		0	1
	Pin function	$\overline{AS}$ output		$P9_5$ input	$P9_5$ output
$P9_4/\overline{WR}$	Bit $P9_4$DDR and the operating mode select the pin function as follows				
	Operating mode	Modes 1 and 2		Mode 3	
	$P9_4$DDR	—		0	1
	Pin function	$\overline{WR}$ output		$P9_4$ input	$P9_4$ output

Table 7-19 Port 9 Pin Functions (cont)

Pin	Pin Functions and Selection Method			
$P9_3/\overline{RD}$	Bit $P9_3$DDR and the operating mode select the pin function as follows			
	Operating mode	Modes 1 and 2	Mode 3	
	$P9_3$DDR	—	0	1
	Pin function	$\overline{RD}$ output	$P9_3$ input	$P9_3$ output
$P9_2/\overline{IRQ_0}$	$P9_2$DDR	0	1	
	Pin function	$P9_2$ input	$P9_2$ output	
		$\overline{IRQ_0}$ input		
	$\overline{IRQ_0}$ input can be used when bit IRQ0E is set to 1 in IER			
$P9_1/\overline{IRQ_1}$				
	$P9_1$DDR	0	1	
	Pin function	$P9_1$ input	$P9_1$ output	
		$\overline{IRQ_1}$ input		
	$\overline{IRQ_1}$ input can be used when bit IRQ1E is set to 1 in IER			
$P9_0/\overline{IRQ_2}/$ $\overline{ADTRG}$				
	$P9_0$DDR	0	1	
	Pin function	$P9_0$ input	$P9_0$ output	
		$\overline{IRQ_2}$ input and $\overline{ADTRG}$ input		
	$\overline{IRQ_2}$ input can be used when bit IRQ2E is set to 1 in IER $\overline{ADTRG}$ input can be used when bit TRGE is set to 1 in ADCR			

Section 8 16-Bit Free-Running Timer

8.1 Overview

The H8/3534 and H8/3522 have an 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.

8.1.1 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 ($\phi_p/2$, $\phi_p/8$, or $\phi_p/32$), 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
  Compare-match A and B, input capture A to D, and overflow interrupts are requested independently.

8.1.2 Block Diagram

Figure 8-1 shows a block diagram of the free-running timer.

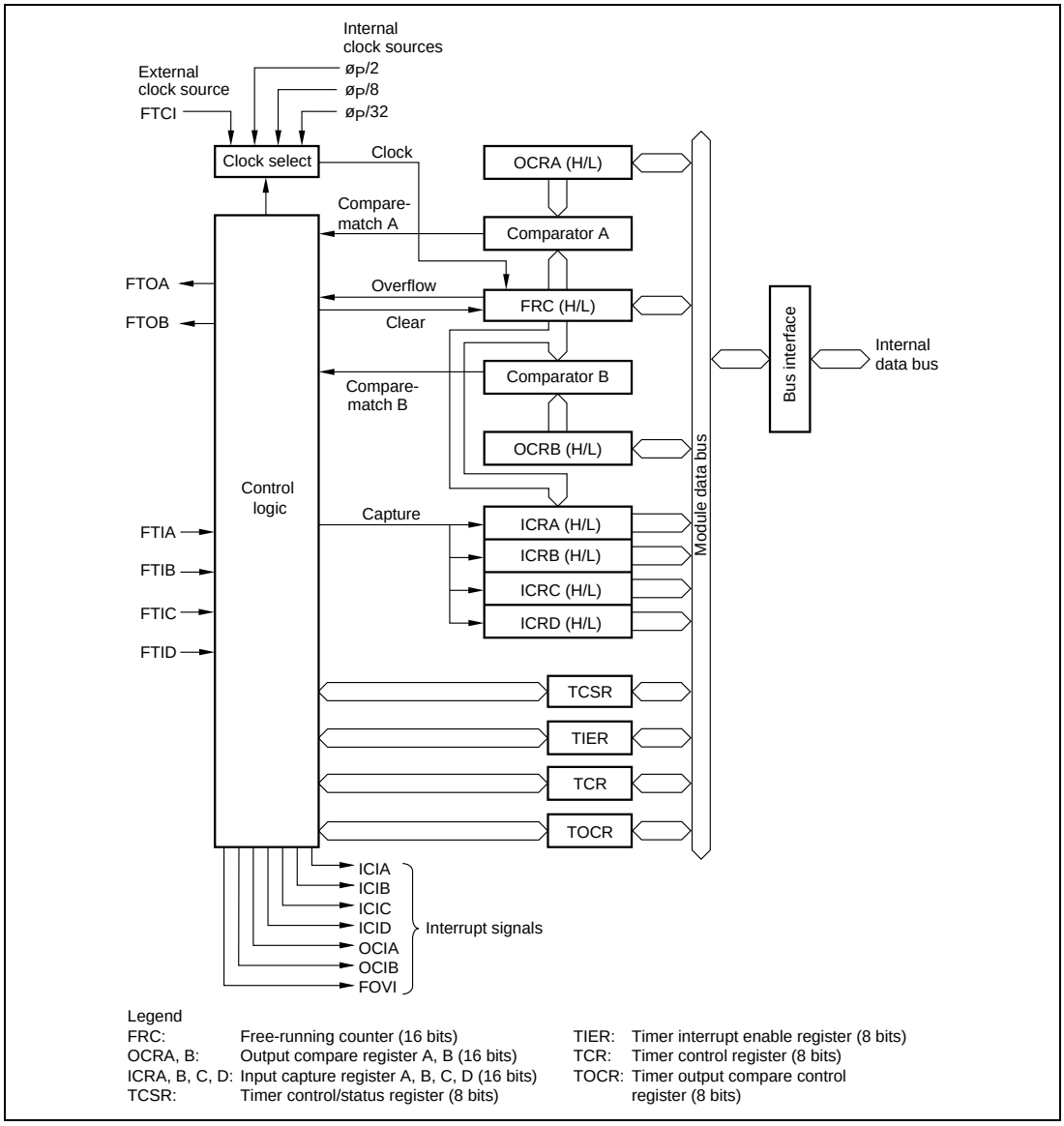


Figure 8-1 Block Diagram of 16-Bit Free-Running Timer

8.1.3 Input and Output Pins

Table 8-1 lists the input and output pins of the free-running timer module.

Table 8-1 Input and Output Pins of Free-Running Timer Module

Name	Abbreviation	I/O	Function
Counter clock input	FTCI	Input	Input of external free-running counter clock signal
Output compare A	FTOA	Output	Output controlled by comparator A
Output compare B	FTOB	Output	Output controlled by comparator B
Input capture A	FTIA	Input	Trigger for capturing current count into input capture register A
Input capture B	FTIB	Input	Trigger for capturing current count into input capture register B
Input capture C	FTIC	Input	Trigger for capturing current count into input capture register C
Input capture D	FTID	Input	Trigger for capturing current count into input capture register D

8.1.4 Register Configuration

Table 8-2 lists the registers of the free-running timer module.

Table 8-2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address
Timer interrupt enable register	TIER	R/W	H'01	H'FF90
Timer control/status register	TCSR	R/(W)*1	H'00	H'FF91
Free-running counter (high)	FRC (H)	R/W	H'00	H'FF92
Free-running counter (low)	FRC (L)	R/W	H'00	H'FF93
Output compare register A/B (high)*2	OCRA/B (H)	R/W	H'FF	H'FF94*2
Output compare register A/B (low)*2	OCRA/B (L)	R/W	H'FF	H'FF95*2
Timer control register	TCR	R/W	H'00	H'FF96
Timer output compare control register	TOCR	R/W	H'E0	H'FF97
Input capture register A (high)	ICRA (H)	R	H'00	H'FF98
Input capture register A (low)	ICRA (L)	R	H'00	H'FF99

Notes: 1. Software can write a 0 to clear bits 7 to 1, but cannot write a 1 in these bits. Bit 0 can be read and written to.

2. OCRA and OCRB share the same addresses. Access is controlled by the OCRS bit in TOCR.

Table 8-2 Register Configuration (cont.)

Name	Abbreviation	R/W	Initial Value	Address
Input capture register B (high)	ICRB (H)	R	H'00	H'FF9A
Input capture register B (low)	ICRB (L)	R	H'00	H'FF9B
Input capture register C (high)	ICRC (H)	R	H'00	H'FF9C
Input capture register C (low)	ICRC (L)	R	H'00	H'FF9D
Input capture register D (high)	ICRD (H)	R	H'00	H'FF9E
Input capture register D (low)	ICRD (L)	R	H'00	H'FF9F

8.2 Register Descriptions

8.2.1 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 the clock select 1 and 0 bits (CKS1 and CKS0) of the timer control register (TCR).

When FRC overflows from H'FFFF to H'0000, the overflow flag (OVF) in the timer control/status register (TCSR) is set to 1.

Because FRC is a 16-bit register, a temporary register (TEMP) is used when FRC is written or read. See section 8.3, CPU Interface, for details.

FRC is initialized to H'0000 by a reset and in the standby modes.

8.2.2 Output Compare Registers A and B (OCRA and 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 flag (OCFA or OCFB) is set in the timer control/status register (TCSR).

In addition, if the output enable bit (OEA or OEB) in the timer output compare control register (TOCR) is set to 1, when the output compare register 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.

OCRA and OCRB share the same address. They are differentiated by the OCRS bit in TOCR. A temporary register (TEMP) is used for write access, as explained in section 8.3, CPU Interface.

OCRA and OCRB are initialized to H'FFFF by a reset and in the standby modes.

8.2.3 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 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 the timer control/status register (TCSR) is set to 1. The input capture edge is selected by the input edge select bits (IEDGA to IEDGD) in the timer control register (TCR).

Note: *The FRC contents are transferred to the input capture register regardless of the value of the input capture flag (ICFA/B/C/D).

Input capture can be buffered by using the input capture registers in pairs. When the BUFEA bit in TCR is set to 1, ICRC is used as a buffer register for ICRA as shown in figure 8-2. When an FTIA input is received, the old ICRA contents are moved into ICRC, and the new FRC count is copied into ICRA.

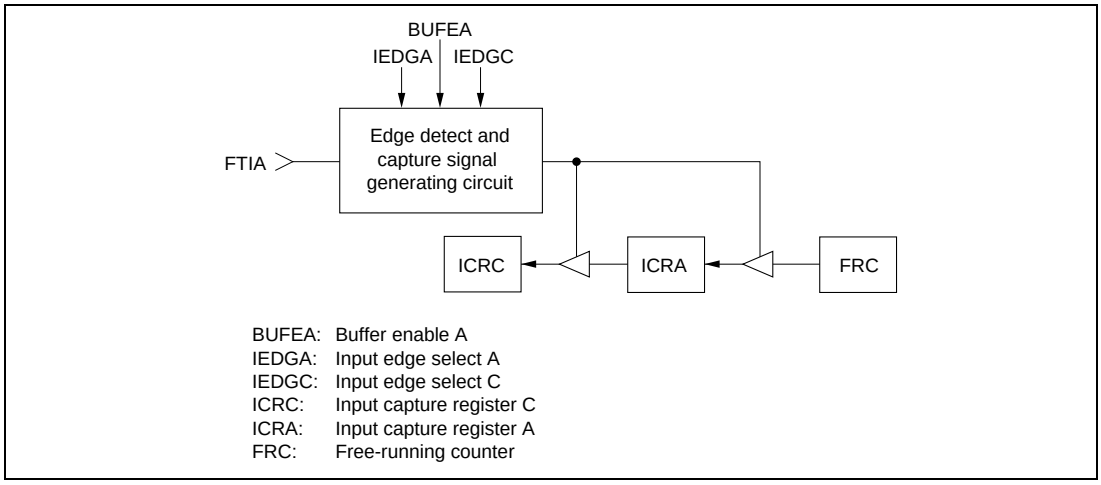


Figure 8-2 Input Capture Buffering (Example)

Similarly, when the BUFEB bit in TCR is set to 1, ICRD is used as a buffer register for ICRB.

When input capture is buffered, if the two input edge bits are set to different values (IEDGA ≠ IEDGC or IEDGB ≠ IEDGD), then input capture is triggered on both the rising and falling edges of the FTIA or FTIB input signal. If the two input edge bits are set to the same value (IEDGA = IEDGC or IEDGB = IEDGD), then input capture is triggered on only one edge. See table 8-3.

Table 8-3 Buffered Input Capture Edge Selection (Example)

IEDGA	IEDGC	Input Capture Edge
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	1	Captured on rising edge of input capture A (FTIA)

Because the input capture registers are 16-bit registers, a temporary register (TEMP) is used when they are read. See section 8.3, CPU Interface, for details.

To ensure input capture, the width of the input capture pulse should be at least 1.5 system clock (ϕ) periods. When triggering is enabled on both edges, the input capture pulse width should be at least 2.5 system clock periods.

The input capture registers are initialized to H'0000 by a reset and in the standby modes.

8.2.4 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 the standby modes.

Bit 7—Input Capture Interrupt A Enable (ICIAE): This bit selects whether to request input capture interrupt A (ICIA) when input capture flag A (ICFA) in the timer status/control register (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): This bit 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): This bit 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): This bit 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): This bit 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): This bit 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): This bit 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.

8.2.5 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: * Software can write a 0 in bits 7 to 1 to clear the flags, but cannot write a 1 in these bits.

TCSR is an 8-bit readable and partially writable register that contains the seven interrupt flags and specifies whether to clear the counter on compare-match A (when the FRC and OCRA values match).

TCSR is initialized to H'00 by a reset and in the standby modes.

Timing is described in section 8.4, Operation.

Bit 7—Input Capture Flag A (ICFA): This status bit is set to 1 to flag an input capture A event. If BUFEA = 0, ICFA indicates that the FRC value has been copied to ICRA. If BUFEA = 1, ICFA indicates that the old ICRA value has been moved into ICRC and the new FRC value has been copied 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	To clear ICFA, the CPU must read ICFA after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when an FTIA input signal causes the FRC value to be copied to ICRA.	

Bit 6—Input Capture Flag B (ICFB): This status bit is set to 1 to flag an input capture B event. If BUFEB = 0, ICFB indicates that the FRC value has been copied to ICRB. If BUFEB = 1, ICFB indicates that the old ICRB value has been moved into ICRD and the new FRC value has been copied 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	To clear ICFB, the CPU must read ICFB after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when an FTIB input signal causes the FRC value to be copied to ICRB.	

Bit 5—Input Capture Flag C (ICFC): This status bit is set to 1 to flag input of a rising or falling edge of FTIC as selected by the IEDGC bit. When BUFEA = 0, this indicates capture of the FRC count in ICRC. When BUFEA = 1, however, the FRC count is not captured, so ICFC becomes simply an external interrupt flag. In other words, the buffer mode frees FTIC for use as a general-purpose interrupt signal (which can be enabled or disabled by the ICICE bit).

ICFC must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 5 ICFC	Description	
0	To clear ICFC, the CPU must read ICFC after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when an FTIC input signal is received.	

Bit 4—Input Capture Flag D (ICFD): This status bit is set to 1 to flag input of a rising or falling edge of FTID as selected by the IEDGD bit. When BUFEB = 0, this indicates capture of the FRC count in ICRD. When BUFEB = 1, however, the FRC count is not captured, so ICFD becomes simply an external interrupt flag. In other words, the buffer mode frees FTID for use as a general-purpose interrupt signal (which can be enabled or disabled by the ICIDE bit).

ICFD must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 4 ICFD	Description	
0	To clear ICFD, the CPU must read ICFD after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when an FTID input signal is received	

Bit 3—Output Compare Flag A (OCFA): This status flag is set to 1 when 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	To clear OCFA, the CPU must read OCFA after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when FRC = OCRA.	

Bit 2—Output Compare Flag B (OCFB): This status flag is set to 1 when 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	To clear OCFB, the CPU must read OCFB after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when FRC = OCRB.	

Bit 1—Timer Overflow Flag (OVF): This status flag is set to 1 when FRC overflows (changes 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	To clear OVF, the CPU must read OVF after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when FRC changes from H'FFFF to H'0000.	

Bit 0—Counter Clear A (CCLRA): This bit selects whether to clear FRC at compare-match A (when the FRC and OCRA values match).

Bit 0 CCLRA	Description	
0	The FRC is not cleared.	(Initial value)
1	The FRC is cleared at compare-match A.	

8.2.6 Timer Control Register (TCR)

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

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 the standby modes.

Bit 7—Input Edge Select A (IEDGA): This bit selects the rising or falling edge of the input capture A signal (FTIA).

Bit 7 IEDGA	Description	
0	Input capture A events are recognized on the falling edge of FTIA.	(Initial value)
1	Input capture A events are recognized on the rising edge of FTIA.	

Bit 6—Input Edge Select B (IEDGB): This bit selects the rising or falling edge of the input capture B signal (FTIB).

Bit 6 IEDGB	Description	
0	Input capture B events are recognized on the falling edge of FTIB.	(Initial value)
1	Input capture B events are recognized on the rising edge of FTIB.	

Bit 5—Input Edge Select C (IEDGC): This bit selects the rising or falling edge of the input capture C signal (FTIC).

Bit 5 IEDGC	Description	
0	Input capture C events are recognized on the falling edge of FTIC.	(Initial value)
1	Input capture C events are recognized on the rising edge of FTIC.	

Bit 4—Input Edge Select D (IEDGD): This bit selects the rising or falling edge of the input capture D signal (FTID).

Bit 4 IEDGD	Description	
0	Input capture D events are recognized on the falling edge of FTID.	(Initial value)
1	Input capture D events are recognized on the rising edge of FTID.	

Bit 3—Buffer Enable A (BUFEA): This bit selects whether to use ICRC as a buffer register for ICRA.

Bit 3 BUFEA	Description	
0	ICRC is not used as a buffer register for ICRA.	(Initial value)
1	ICRC is used as a buffer register for ICRA.	

Bit 2—Buffer Enable B (BUFEB): This bit selects whether to use ICRD as a buffer register for ICRB.

Bit 2 BUFEB	Description	
0	ICRD is not used as a buffer register for ICRB.	(Initial value)
1	ICRD is used as a buffer register for ICRB.	

Bits 1 and 0—Clock Select (CKS1 and CKS0): These bits select external clock input or one of three internal clock sources for FRC. External clock pulses are counted on the rising edge of signals input to pin FTCI.

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	ϕ_P/2 internal clock source	(Initial value)
0	1	ϕ_P/8 internal clock source	
1	0	ϕ_P/32 internal clock source	
1	1	External clock source (rising edge)	

8.2.7 Timer Output Compare Control Register (TOCR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	OCRS	OEA	OEB	OLVLA	OLVLB
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	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, and switches access between output compare registers A and B.

TOCR is initialized to H'E0 by a reset and in the standby modes.

Bits 7 to 5—Reserved: These bits cannot be modified and are always read as 1.

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	OCRA is selected.	(Initial value)
1	OCRB is selected.	

Bit 3—Output Enable A (OEA): This bit 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): This bit 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): This bit selects the logic level to be output at the FTOA pin when the FRC and OCRA values match.

Bit 1 OLVLA	Description	
0	A 0 logic level is output for compare-match A.	(Initial value)
1	A 1 logic level is output for compare-match A.	

Bit 0—Output Level B (OLVLB): This bit selects the logic level to be output at the FTOB pin when the FRC and OCRB values match.

Bit 0 OLVLB	Description	
0	A 0 logic level is output for compare-match B.	(Initial value)
1	A 1 logic level is output for compare-match B.	

8.3 CPU Interface

The free-running counter (FRC), output compare registers (OCRA and OCRB), and input capture registers (ICRA to ICRD) are 16-bit registers, but they are connected to an 8-bit data bus. When the CPU accesses these registers, to ensure that both bytes are written or read simultaneously, the access is performed using an 8-bit temporary register (TEMP).

These registers are written and read as follows:

- **Register Write**

  When the CPU writes to the upper byte, the byte of write data is placed in TEMP. Next, when the CPU writes to the lower byte, this byte of data is combined with the byte in TEMP and all 16 bits are written in the register simultaneously.
- **Register Read**

  When the CPU reads the upper byte, the upper byte of data is sent to the CPU and the lower byte is placed in TEMP. When the CPU reads the lower byte, it receives the value in TEMP.

Programs that access these registers should normally use word access. Equivalently, they may access first the upper byte, then the lower byte by two consecutive byte accesses. Data will not be transferred correctly if the bytes are accessed in reverse order, or if only one byte is accessed.

Figure 8-3 shows the data flow when FRC is accessed. The other registers are accessed in the same way. As an exception, when the CPU reads OCRA or OCRB, it reads both the upper and lower bytes directly, without using TEMP.

Coding Examples

To write the contents of general register R0 to OCRA:

```
MOV.W  R0, @OCRA
```

To transfer the contents of ICRA to general register R0:

```
MOV.W  @ICRA, R0
```

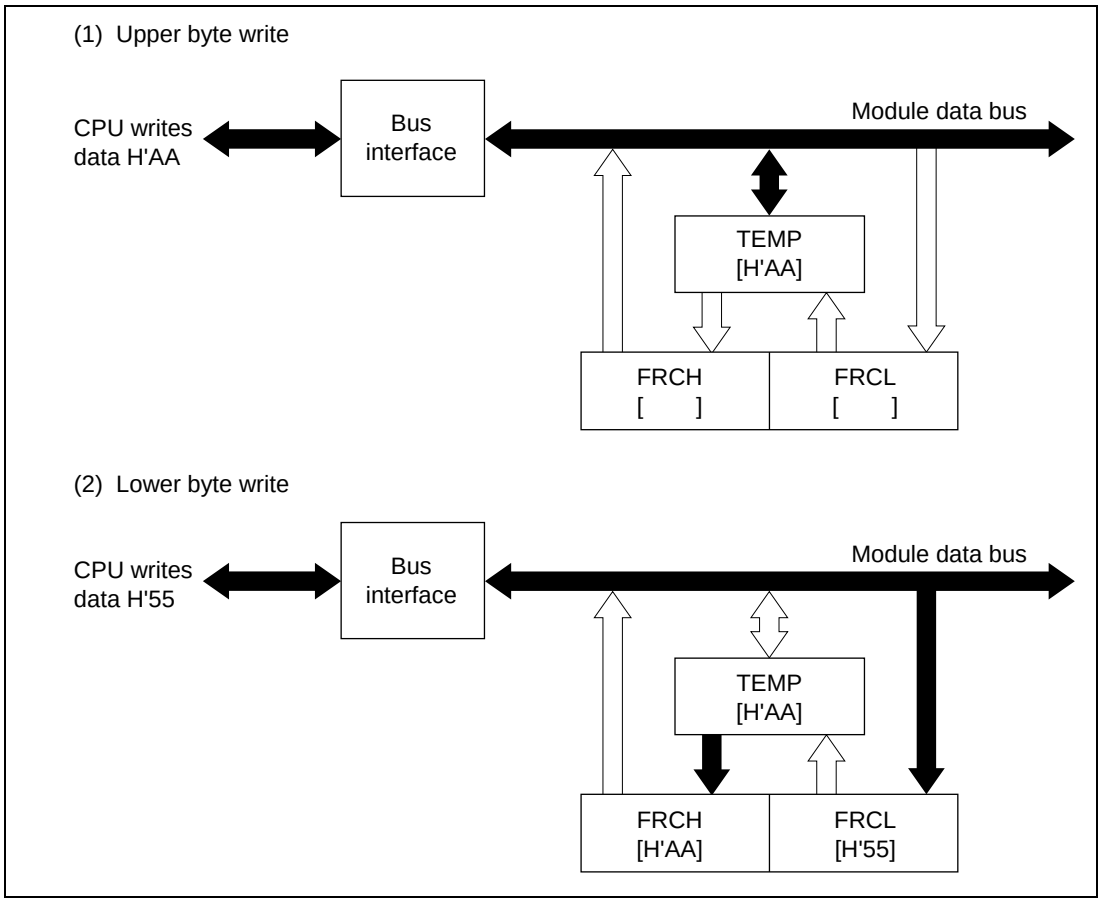


Figure 8-3 (a) Write Access to FRC (when CPU Writes H'AA55)

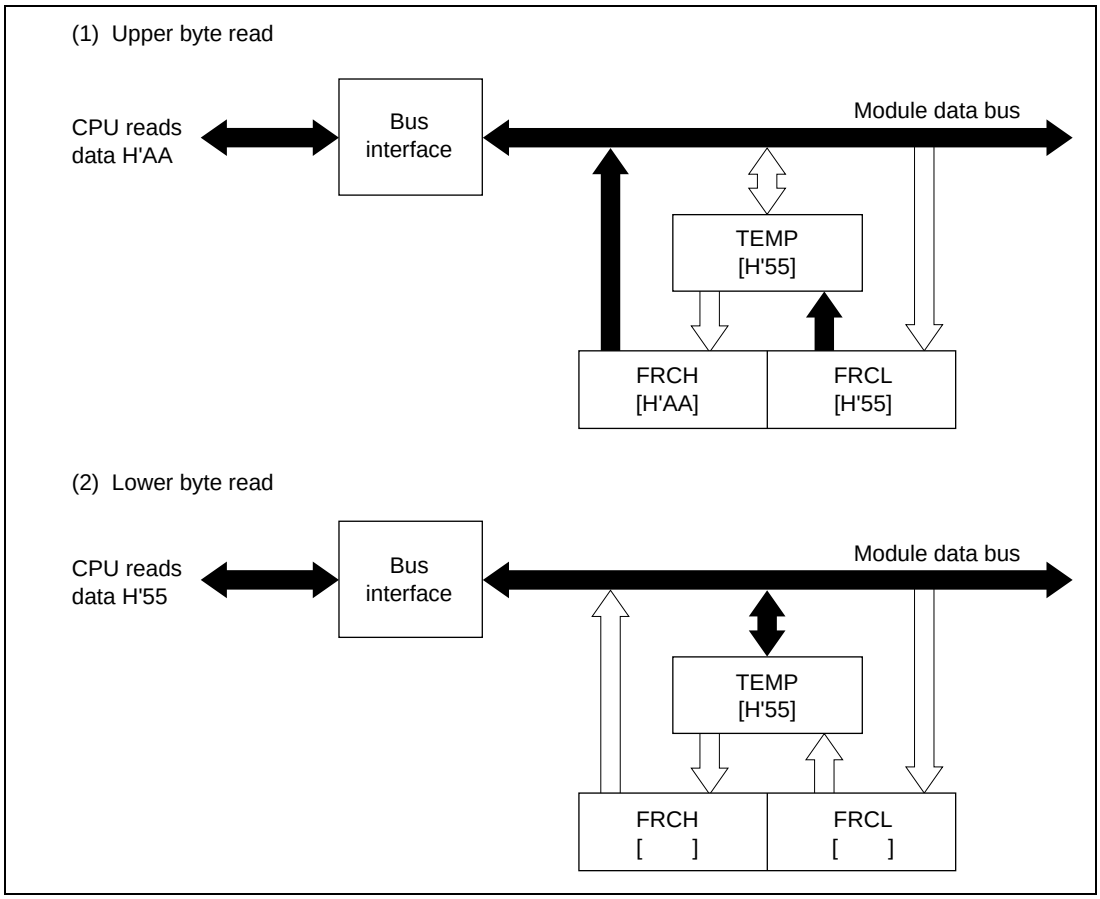


Figure 8-3 (b) Read Access to FRC (when FRC Contains H'AA55)

8.4 Operation

8.4.1 FRC Increment Timing

FRC increments on a pulse generated once for each period of the selected (internal or external) clock source. The clock source is selected by bits CKS0 and CKS1 in TCR.

Internal Clock: The internal clock sources ($\phi_p/2$, $\phi_p/8$, $\phi_p/32$) are created from the system clock (ϕ) by a prescaler. FRC increments on a pulse generated from the falling edge of the prescaler output. See figure 8-4.

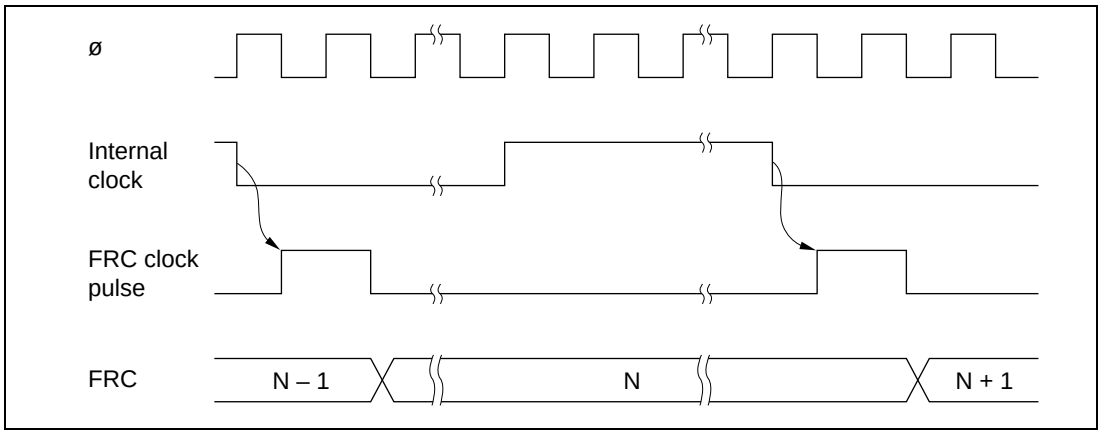


Figure 8-4 Increment Timing for Internal Clock Source

External Clock: If external clock input is selected, FRC increments on the rising edge of the FTCI clock signal. Figure 8-5 shows the increment timing.

The pulse width of the external clock signal must be at least 1.5 system clock (ϕ) periods. The counter will not increment correctly if the pulse width is shorter than 1.5 system clock periods.

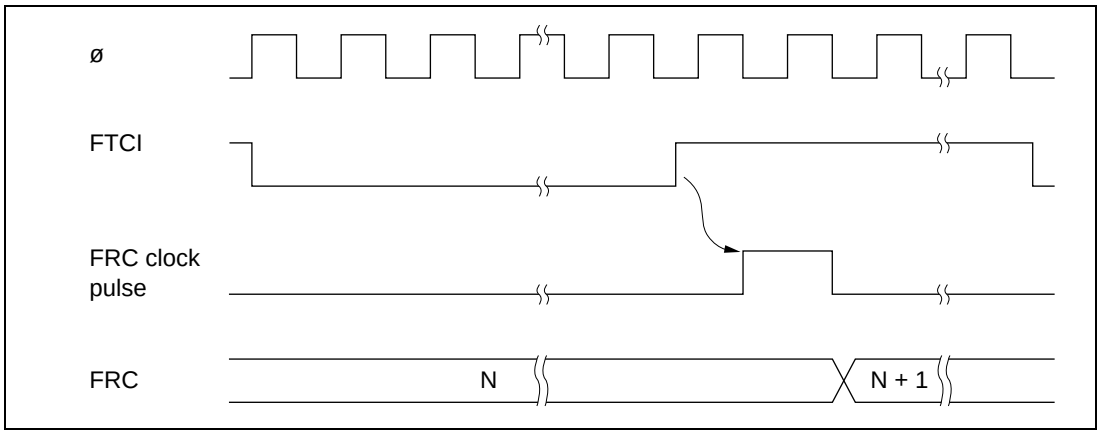


Figure 8-5 Increment Timing for External Clock Source

8.4.2 Output Compare 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 8-6 shows the timing of this operation for compare-match A.

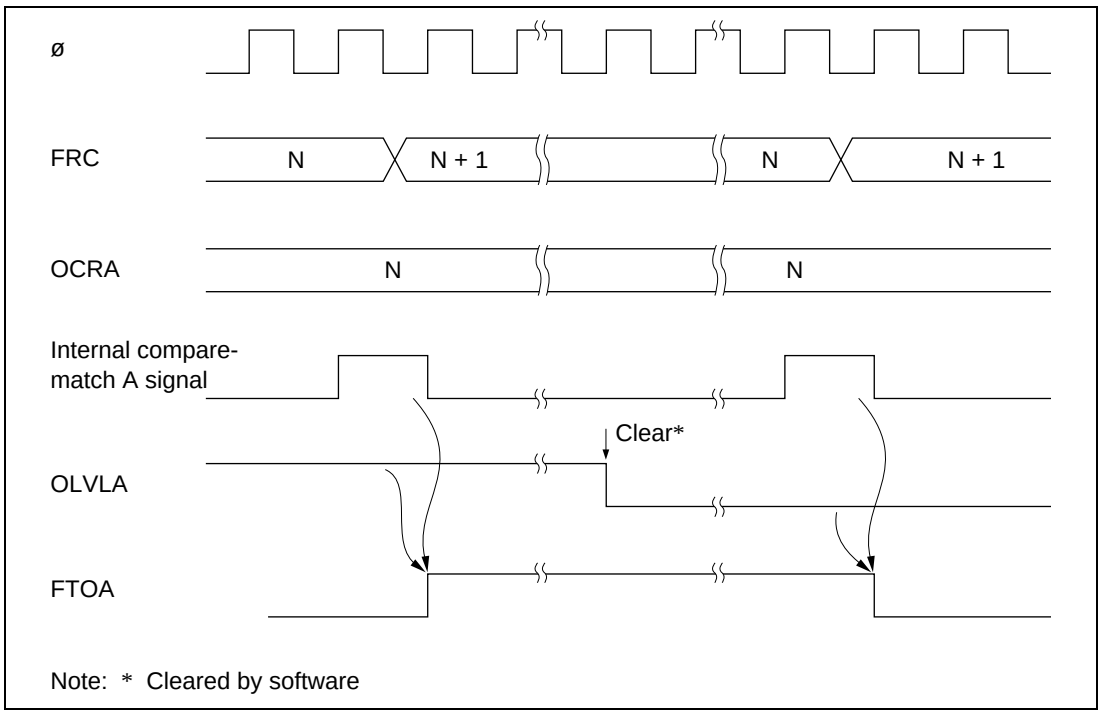


Figure 8-6 Timing of Output Compare A

8.4.3 FRC Clear Timing

If the CCLRA bit in TCSR is set to 1, the FRC is cleared when compare-match A occurs. Figure 8-7 shows the timing of this operation.

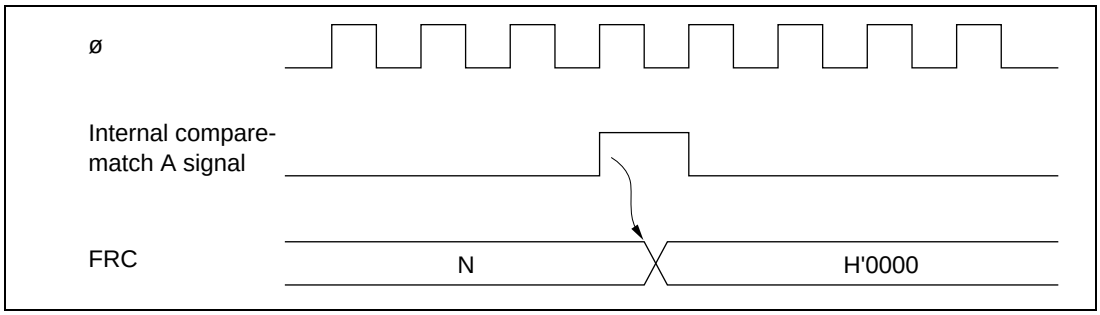


Figure 8-7 Clearing of FRC by Compare-Match A

8.4.4 Input Capture Timing

(1) Input Capture Timing: An internal input capture signal is generated from the rising or falling edge of the signal at the input capture pin FTIx (x = A, B, C, D), as selected by the corresponding IEDGx bit in TCR. Figure 8-8 shows the usual input capture timing when the rising edge is selected (IEDGx = 1).

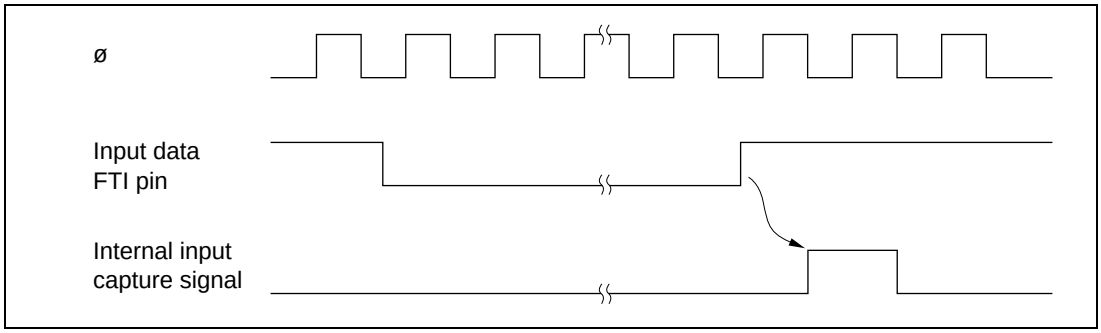


Figure 8-8 Input Capture Timing (Usual Case)

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 state. Figure 8-9 shows the timing for this case.

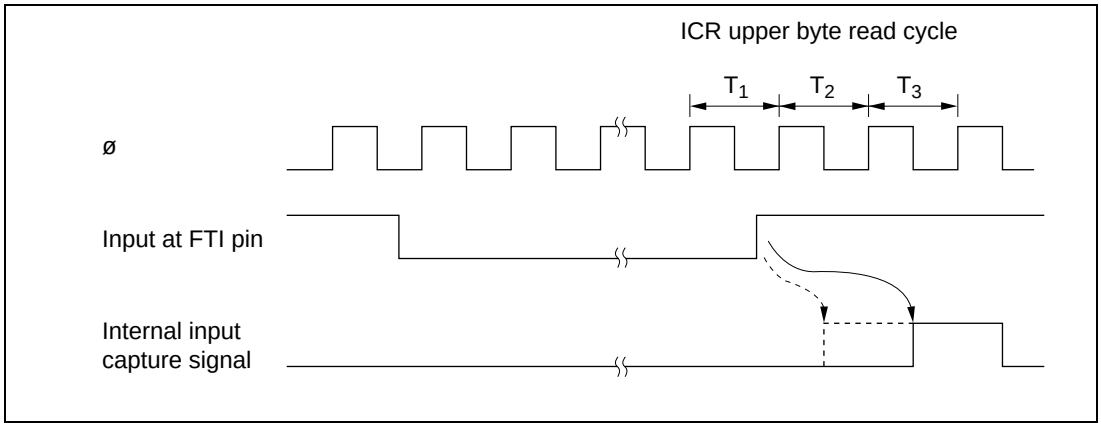


Figure 8-9 Input Capture Timing (1-State Delay Due to ICRA/B/C/D Read)

(2) Buffered Input Capture Timing: ICRC and ICRD can operate as buffers for ICRA and ICRB.

Figure 8-10 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.

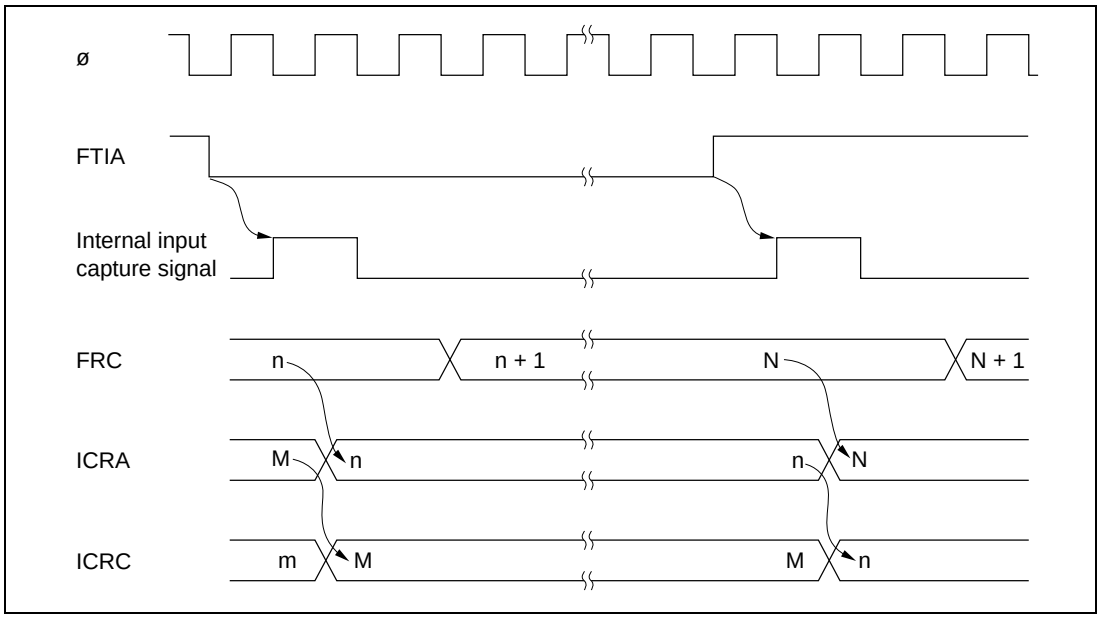


Figure 8-10 Buffered Input Capture with Both Edges Selected

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 (ϕ). Figure 8-11 shows the timing when BUFEA = 1.

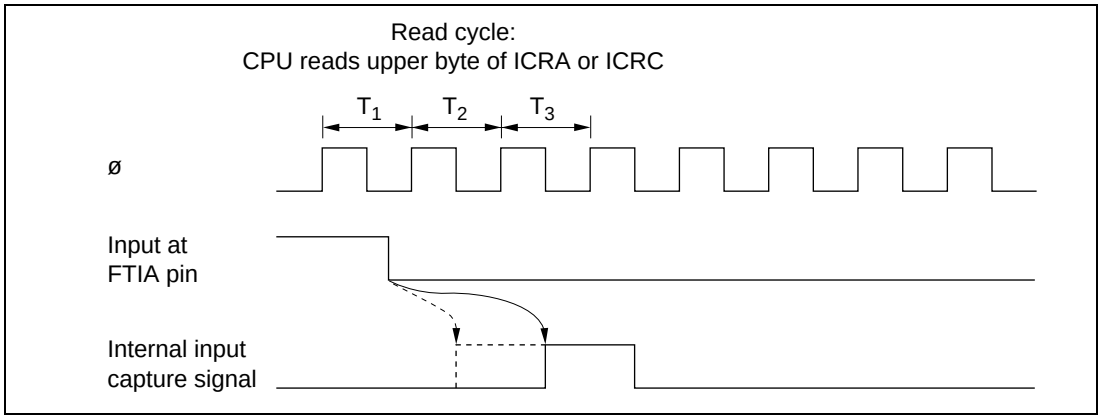


Figure 8-11 Input Capture Timing (1-State Delay, Buffer Mode)

8.4.5 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. Figure 8-12 shows the timing of this operation.

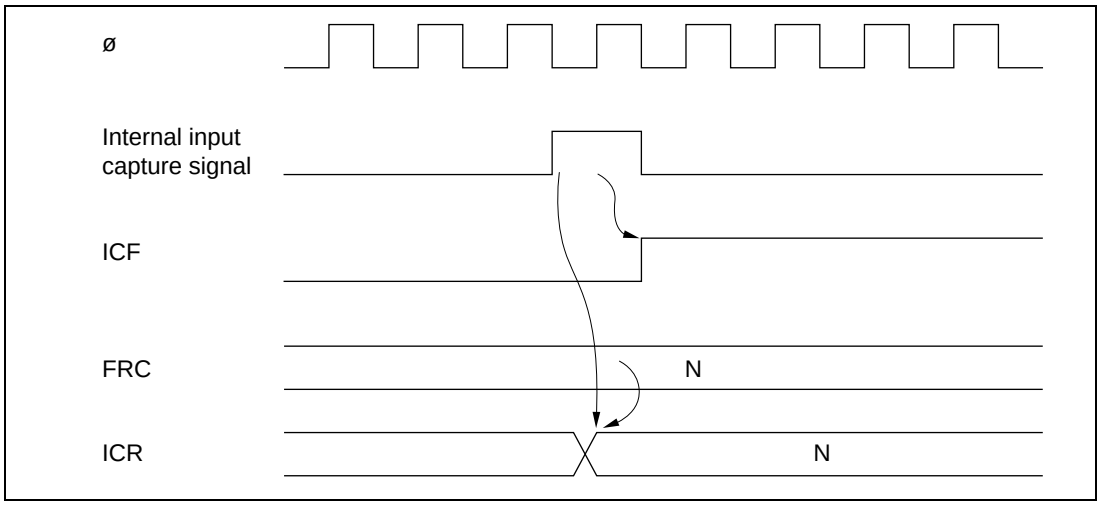


Figure 8-12 Setting of Input Capture Flag

8.4.6 Setting of Output Compare Flags A and B (OCFA and 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 8-13 shows the timing of the setting of the output compare flags.

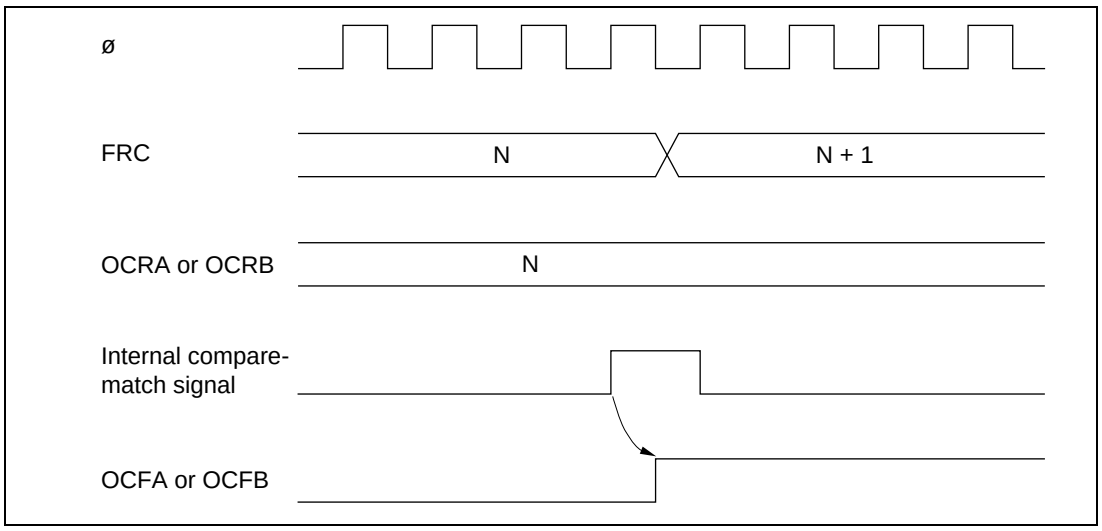


Figure 8-13 Setting of Output Compare Flags

8.4.7 Setting of Timer Overflow Flag (OVF)

The timer overflow flag (OVF) is set to 1 when FRC overflows (changes from H'FFFF to H'0000). Figure 8-14 shows the timing of this operation.

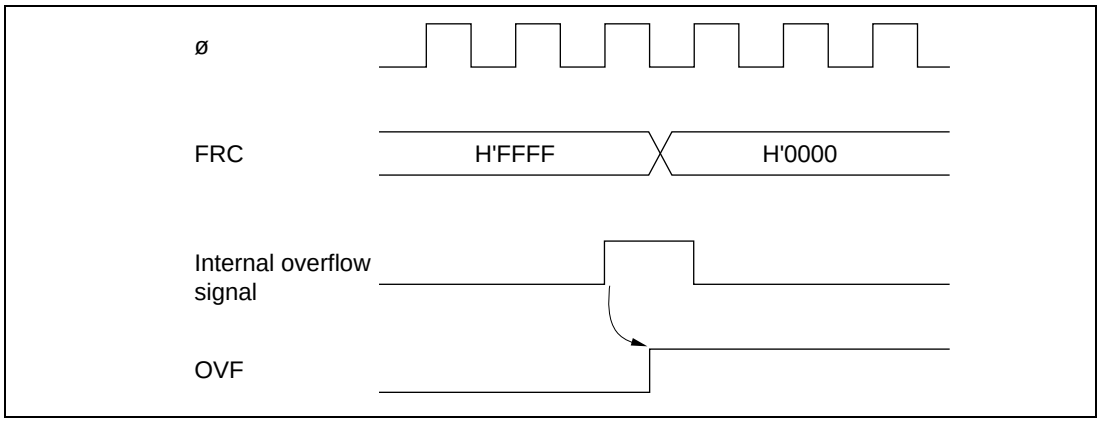


Figure 8-14 Setting of Timer Overflow Flag (OVF)

8.5 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 8-4 lists information about these interrupts.

Table 8-4 Free-Running Timer Interrupts

Interrupt	Description	Priority
ICIA	Requested by ICFA	High
ICIB	Requested by ICFB	
ICIC	Requested by ICFC	
ICID	Requested by ICFD	↓
OCIA	Requested by OCFA	
OCIB	Requested by OCFB	
FOVI	Requested by OVF	Low

8.6 Sample Application

In the example below, the free-running timer is used to generate two square-wave outputs with a 50% duty cycle and arbitrary phase relationship. The programming is as follows:

(1) The CCLRA bit in TCSR is set to 1.

(2) Each time a compare-match interrupt occurs, software inverts the corresponding output level bit in TOCR (OLVLA or OLVLB).

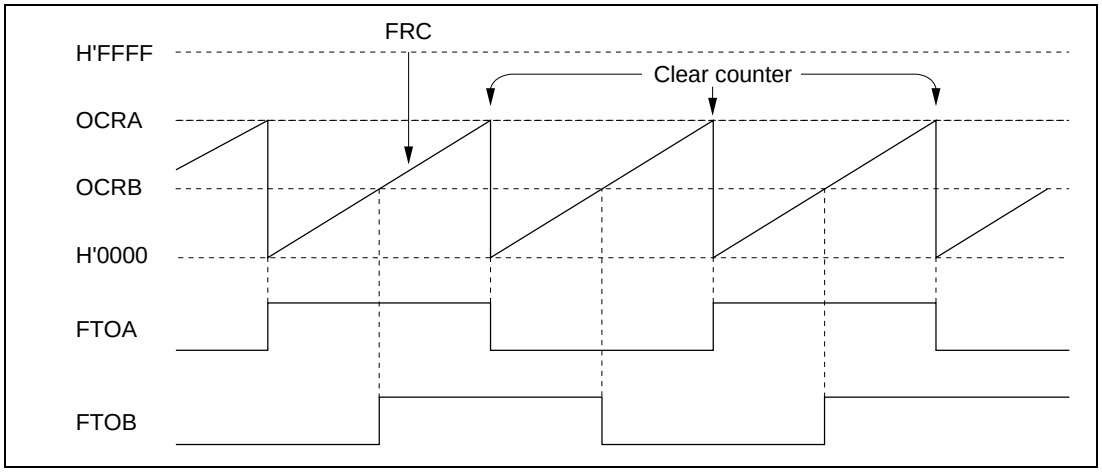


Figure 8-15 Square-Wave Output (Example)

8.7 Application Notes

Application programmers should note that the following types of contention can occur in the free-running timer.

(1) Contention between FRC Write and Clear: If an internal counter clear signal is generated during the T_3 state of a write cycle to the lower byte of the free-running counter, the clear signal takes priority and the write is not performed.

Figure 8-16 shows this type of contention.

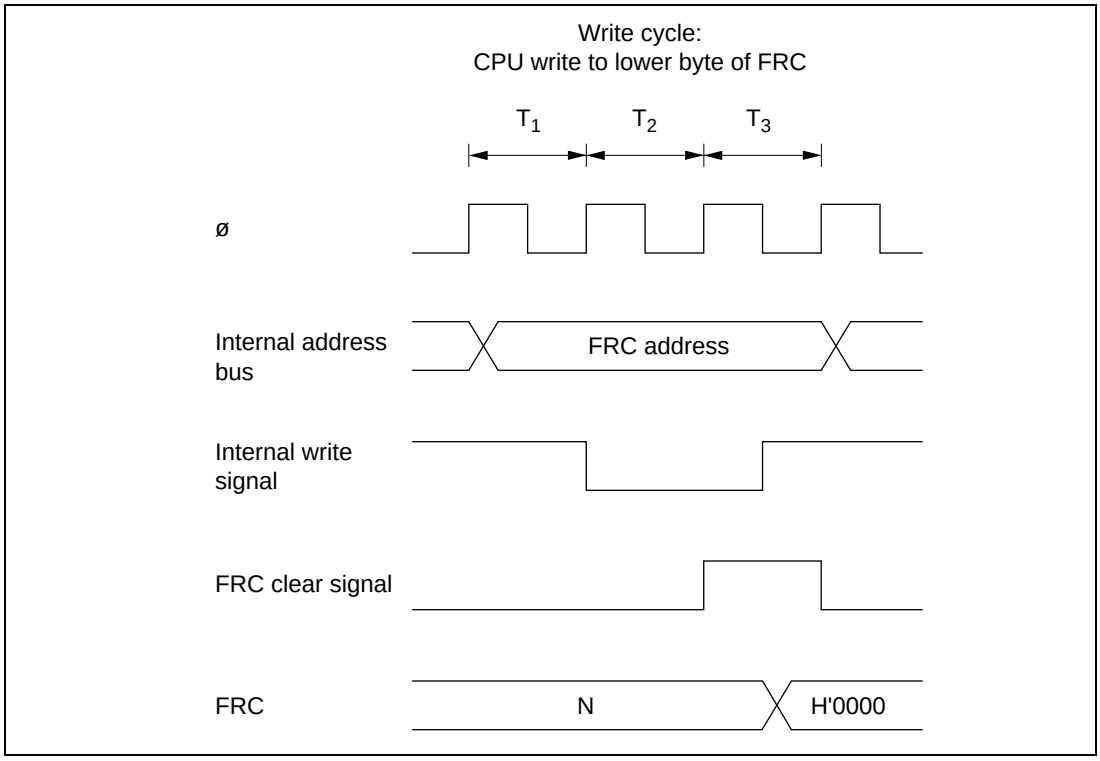


Figure 8-16 FRC Write-Clear Contention

(2) Contention between FRC Write and Increment: If an FRC increment pulse is generated during the T_3 state of a write cycle to the lower byte of the free-running counter, the write takes priority and FRC is not incremented.

Figure 8-17 shows this type of contention.

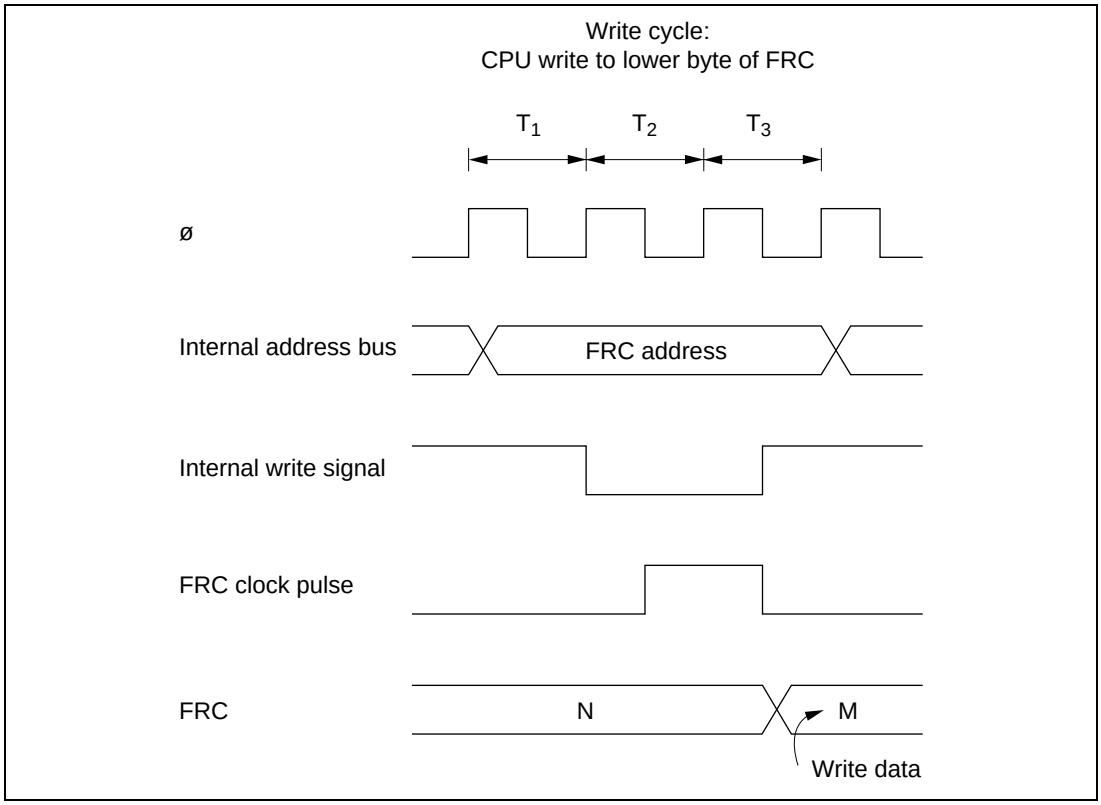


Figure 8-17 FRC Write-Increment Contention

(3) Contention between OCR Write and Compare-Match: If a compare-match occurs during the T_3 state of a write cycle to the lower byte of OCRA or OCRB, the write takes priority and the compare-match signal is inhibited.

Figure 8-18 shows this type of contention.

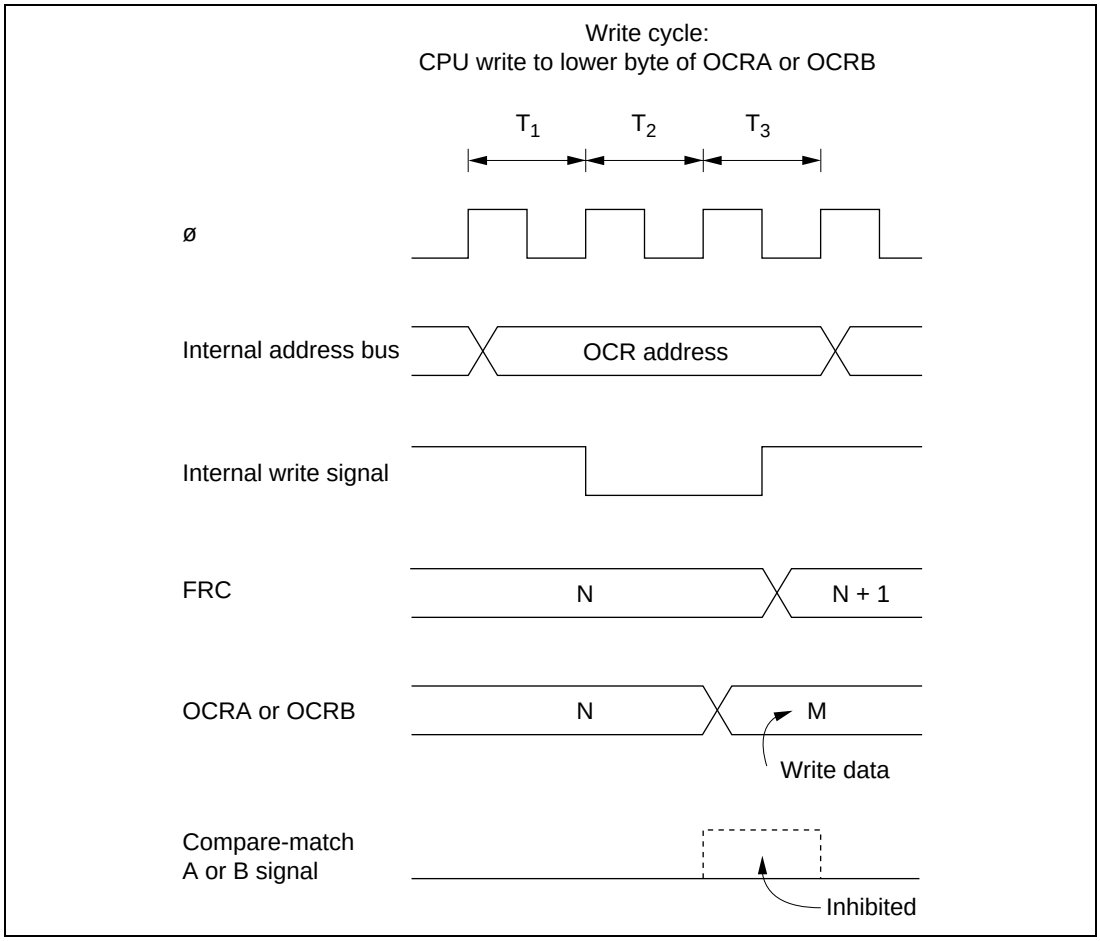


Figure 8-18 Contention between OCR Write and Compare-Match

(4) Increment Caused by Changing of Internal Clock Source: When an internal clock source 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 8-5.

The pulse that increments FRC is generated at the falling edge of the internal clock source. If clock sources are changed when the old source is high and the new source is low, as in case no. 3 in table 8-5, the changeover generates a falling edge that triggers the FRC increment clock pulse.

Switching between an internal and external clock source can also cause FRC to increment.

Table 8-5 Effect of Changing Internal Clock Sources

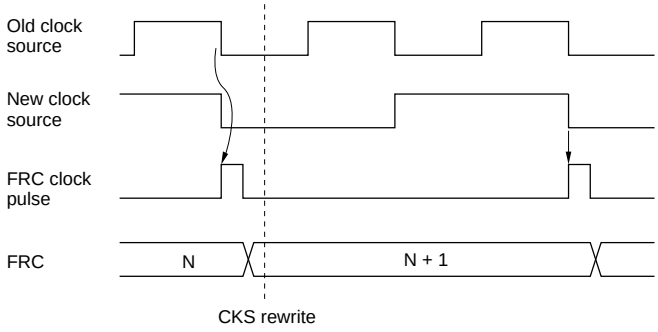
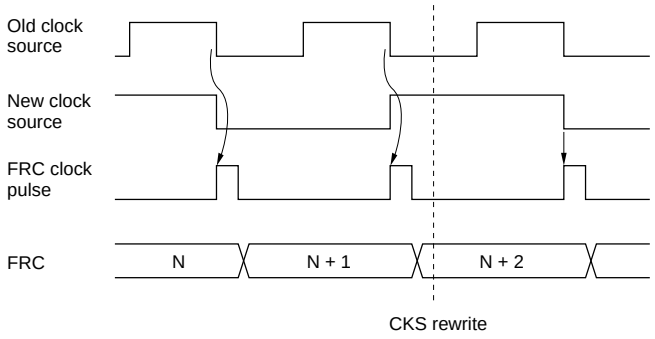
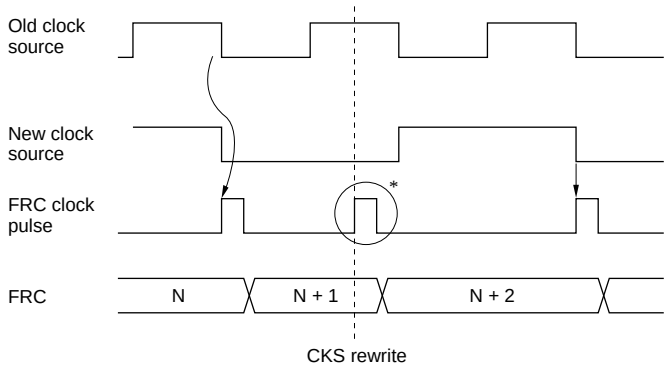
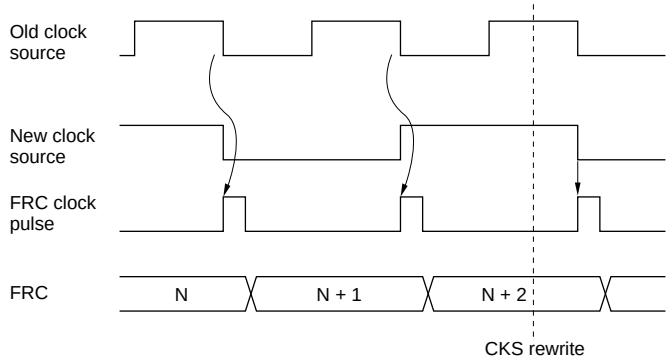
No.	Description	Timing
1	Low → low: CKS1 and CKS0 are rewritten while both clock sources are low.	
2	Low → high: CKS1 and CKS0 are rewritten while old clock source is low and new clock source is high.	

Table 8-5 Effect of Changing Internal Clock Sources (cont)

No.	Description	Timing
3	High → low: CKS1 and CKS0 are rewritten while old clock source is high and new clock source is low.	
4	High → high: CKS1 and CKS0 are rewritten while both clock sources are high.	

Note: * The switching of clock sources is regarded as a falling edge that increments FRC.

Section 9 8-Bit Timers

9.1 Overview

The H8/3534 and H8/3522 include an 8-bit timer module with two channels (numbered 0 and 1). 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-match events. One of the many applications of the 8-bit timer module is to generate a rectangular-wave output with an arbitrary duty cycle.

9.1.1 Features

The features of the 8-bit timer module are listed below.

- Selection of seven clock sources
  The counters can be driven by one of six internal clock signals or an external clock input (enabling use as an external event counter).
- Selection of three ways to clear the counters
  The counters can be cleared on compare-match A or B, or by an external reset 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 generate output waveforms with an arbitrary duty cycle, or PWM waveforms.
- Three independent interrupts
  Compare-match A and B and overflow interrupts can be requested independently.

9.1.2 Block Diagram

Figure 9-1 shows a block diagram of one channel in the 8-bit timer module.

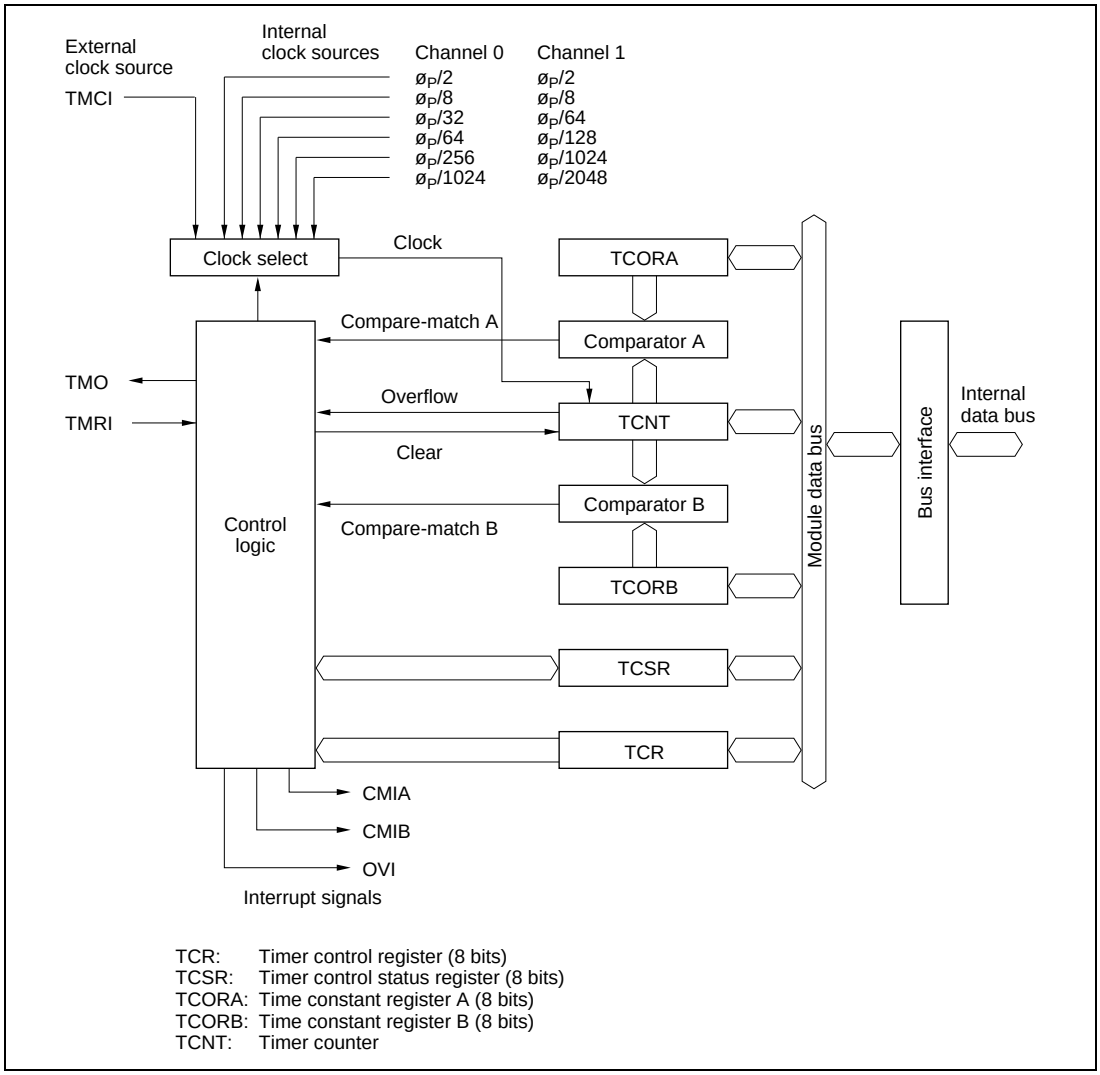


Figure 9-1 Block Diagram of 8-Bit Timer (1 Channel)

9.1.3 Input and Output Pins

Table 9-1 lists the input and output pins of the 8-bit timer.

Table 9-1 Input and Output Pins of 8-Bit Timer

	Abbreviation*			
Name	**Channel 0**	**Channel 1**	**I/O**	**Function**
Timer output	TMO_0	TMO_1	Output	Output controlled by compare-match
Timer clock input	$TMCI_0$	$TMCI_1$	Input	External clock source for the counter
Timer reset input	$TMRI_0$	$TMRI_1$	Input	External reset signal for the counter

Note: * In this manual, the channel subscript has been deleted, and only TMO, TMCI, and TMRI are used.

9.1.4 Register Configuration

Table 9-2 lists the registers of the 8-bit timer module.

Table 9-2 8-Bit Timer Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address
0	Timer control register	TCR	R/W	H'00	H'FFC8
	Timer control/status register	TCSR	R/(W)*	H'10	H'FFC9
	Time constant register A	TCORA	R/W	H'FF	H'FFCA
	Time constant register B	TCORB	R/W	H'FF	H'FFCB
	Timer counter	TCNT	R/W	H'00	H'FFCC
1	Timer control register	TCR	R/W	H'00	H'FFD0
	Timer control/status register	TCSR	R/(W)*	H'10	H'FFD1
	Time constant register A	TCORA	R/W	H'FF	H'FFD2
	Time constant register B	TCORB	R/W	H'FF	H'FFD3
	Timer counter	TCNT	R/W	H'00	H'FFD4
0, 1	Serial/timer control register	STCR	R/W	H'00	H'FFC3

Note: * Software can write a 0 to clear bits 7 to 5, but cannot write a 1 in these bits.

9.2 Register Descriptions

9.2.1 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

Each timer counter (TCNT) is an 8-bit up-counter that increments on a pulse generated from an internal or external clock source selected by clock select bits 2 to 0 (CKS2 to CKS0) of the timer control register (TCR). The CPU can always read or write the timer counter.

The timer counter can be cleared by an external reset input or by an internal compare-match signal generated at a compare-match event. Clock clear bits 1 and 0 (CCLR1 and CCLR0) of the timer control register select the method of clearing.

When a timer counter overflows from H'FF to H'00, the overflow flag (OVF) in the timer control/status register (TCSR) is set to 1.

The timer counters are initialized to H'00 by a reset and in the standby modes.

9.2.2 Time Constant Registers A and B (TCORA and TCORB)

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 and TCORB are 8-bit readable/writable registers. The timer count is continually compared with the constants written in these registers (except during the T_3 state of a write cycle to TCORA or TCORB). When a match is detected, the corresponding compare-match flag (CMFA or CMFB) is set in the timer control/status register (TCSR).

The timer output signal is controlled by these compare-match signals as specified by output select bits 3 to 0 (OS3 to OS0) in the timer control/status register (TCSR).

TCORA and TCORB are initialized to H'FF by a reset and in the standby modes.

9.2.3 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 the timer counter is cleared, and enables interrupts.

TCR is initialized to H'00 by a reset and in the standby modes.

For timing diagrams, see section 9.3, Operation.

Bit 7—Compare-Match Interrupt Enable B (CMIEB): This bit selects whether to request compare-match interrupt B (CMIB) when compare-match flag B (CMFB) in the timer control/status register (TCSR) is set to 1.

Bit 7 CMIEB	Description	
0	Compare-match interrupt request B (CMIB) is disabled.	(Initial value)
1	Compare-match interrupt request B (CMIB) is enabled.	

Bit 6—Compare-Match Interrupt Enable A (CMIEA): This bit selects whether to request compare-match interrupt A (CMIA) when compare-match flag A (CMFA) in TCSR is set to 1.

Bit 6 CMIEA	Description	
0	Compare-match interrupt request A (CMIA) is disabled.	(Initial value)
1	Compare-match interrupt request A (CMIA) is enabled.	

Bit 5—Timer Overflow Interrupt Enable (OVIE): This bit selects whether to request a timer overflow interrupt (OVI) when the overflow flag (OVF) in TCSR is set to 1.

Bit 5 OVIE	Description	
0	The timer overflow interrupt request (OVI) is disabled.	Initial value)
1	The timer overflow interrupt request (OVI) is enabled.	

Bits 4 and 3—Counter Clear 1 and 0 (CCLR1 and CCLR0): These bits select how the timer counter is cleared: by compare-match A or B or by an external reset input (TMRI).

Bit 4 CCLR1	Bit 3 CCLR0	Description	
0	0	Not cleared.	(Initial value)
0	1	Cleared on compare-match A.	
1	0	Cleared on compare-match B.	
1	1	Cleared on rising edge of external reset input signal.	

Bits 2, 1, and 0—Clock Select (CKS2, CKS1, and CKS0): These bits and bits ICKS1 and ICKS0 in the serial/timer control register (STCR) select the internal or external clock source for the timer counter. Six internal clock sources, derived by prescaling the system clock, are available for each timer channel. For internal clock sources the counter is incremented on the falling edge of the internal clock. For an external clock source, these bits can select whether to increment the counter on the rising or falling edge of the clock input (TMCI), or on both edges.

	TCR			STCR		
	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
Channel	CKS2	CKS1	CKS0	ICKS1	ICKS0	Description
0	0	0	0	—	—	No clock source (timer stopped) (Initial value)
	0	0	1	—	0	ϕ_P/8 internal clock, counted on falling edge
	0	0	1	—	1	ϕ_P/2 internal clock, counted on falling edge
	0	1	0	—	0	ϕ_P/64 internal clock, counted on falling edge
	0	1	0	—	1	ϕ_P/32 internal clock, counted on falling edge
	0	1	1	—	0	ϕ_P/1024 internal clock, counted on falling edge
	0	1	1	—	1	ϕ_P/256 internal clock, counted on falling edge
	1	0	0	—	—	No clock source (timer stopped)
	1	0	1	—	—	External clock source, counted on rising edge
	1	1	0	—	—	External clock source, counted on falling edge
	1	1	1	—	—	External clock source, counted on both rising and falling edges
1	0	0	0	—	—	No clock source (timer stopped) (Initial value)
	0	0	1	0	—	ϕ_P/8 internal clock, counted on falling edge
	0	0	1	1	—	ϕ_P/2 internal clock, counted on falling edge
	0	1	0	0	—	ϕ_P/64 internal clock, counted on falling edge
	0	1	0	1	—	ϕ_P/128 internal clock, counted on falling edge
	0	1	1	0	—	ϕ_P/1024 internal clock, counted on falling edge
	0	1	1	1	—	ϕ_P/2048 internal clock, counted on falling edge
	1	0	0	—	—	No clock source (timer stopped)
	1	0	1	—	—	External clock source, counted on rising edge
	1	1	0	—	—	External clock source, counted on falling edge
	1	1	1	—	—	External clock source, counted on both rising and falling edges

9.2.4 Timer Control/Status Register (TCSR)

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

Note: * Software can write a 0 in bits 7 to 5 to clear the flags, but cannot write a 1 in these bits.

TCSR is an 8-bit readable and partially writable register that indicates compare-match and overflow status and selects the effect of compare-match events on the timer output signal.

TCSR is initialized to H'10 by a reset and in the standby modes.

Bit 7—Compare-Match Flag B (CMFB): This status flag is set to 1 when the timer count matches the time constant set in TCORB. CMFB must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 7 CMFB	Description	
0	To clear CMFB, the CPU must read CMFB after it has been set to 1 then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when TCNT = TCORB.	

Bit 6—Compare-Match Flag A (CMFA): This status flag is set to 1 when the timer count matches the time constant set in TCORA. CMFA must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 6 CMFA	Description	
0	To clear CMFA, the CPU must read CMFA after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when TCNT = TCORA.	

Bit 5—Timer Overflow Flag (OVF): This status flag is set to 1 when the timer count overflows (changes from H'FF to H'00). OVF must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 5 OVF	Description	
0	To clear OVF, the CPU must read OVF after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when TCNT changes from H'FF to H'00.	

Bit 4—Reserved: This bit is always read as 1. It cannot be written.

Bits 3 to 0—Output Select 3 to 0 (OS3 to OS0): These bits specify the effect of TCOR–TCNT compare-match events on the timer output signal (TMO). Bits OS3 and OS2 control the effect of compare-match B on the output level. Bits OS1 and OS0 control the effect of compare-match A on the output level.

If compare-match A and B occur simultaneously, any conflict is resolved according to the following priority order: toggle > 1 output > 0 output.

When all four output select bits are cleared to 0 the timer output signal is disabled.

After a reset, the timer output is 0 until the first compare-match event.

Bit 3 OS3	Bit 2 OS2	Description	
0	0	No change when compare-match B occurs.	(Initial value)
0	1	Output changes to 0 when compare-match B occurs.	
1	0	Output changes to 1 when compare-match B occurs.	
1	1	Output inverts (toggles) when compare-match B occurs	

. Bit 1 OS1	Bit 0 OS0	Description	
0	0	No change when compare-match A occurs.	(Initial value)
0	1	Output changes to 0 when compare-match A occurs.	
1	0	Output changes to 1 when compare-match A occurs.	
1	1	Output inverts (toggles) when compare-match A occurs.	

9.2.5 Serial/Timer Control Register (STCR)

Bit	7	6	5	4	3	2	1	0
[H8/3534]	(IICS)	(IICD)	(IICX)	(IICE)	(STAC)	MPE	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

Bit	7	6	5	4	3	2	1	0
[H8/3522]	—	—	—	—	—	MPE	ICKS1	ICKS0
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

STCR is an 8-bit readable/writable register that controls the operating mode of the serial communication interface and selects internal clock sources for the timer counters.

STCR is initialized to H'00 [H8/3534]/H'F8 [H8/3522] by a reset.

Bits 7 to 4—I^{2}C Control (IICS, IICD, IICX, IICE) [H8/3534]: These bits are reserved. They should not be set to 1.

Bit 3—Slave Input Switch (STAC) [H8/3534]: This bit is reserved. It should not be set to 1.

Bits 7 to 3—Reserved [H8/3522]: These bits cannot be modified and are always read as 1.

Bit 2—Multiprocessor Enable (MPE): Controls the operating mode of serial communication interfaces 0 and 1. For details, see section 12, Serial Communication Interface.

Bits 1 and 0—Internal Clock Source Select 1 and 0 (ICKS1 and ICKS0): These bits and bits CKS2 to CKS0 in TCR select clock sources for the timer counters. For details, see section 9.2.3, Timer Control Register.

9.3 Operation

9.3.1 TCNT Increment Timing

The timer counter increments on a pulse generated once for each period of the selected (internal or external) clock source.

Internal Clock: Internal clock sources are created from the system clock by a prescaler. The counter increments on an internal TCNT clock pulse generated from the falling edge of the prescaler output, as shown in figure 9-2. Bits CKS2 to CKS0 of TCR and bits ICKS1 and ICKS0 of STCR can select one of the six internal clocks.

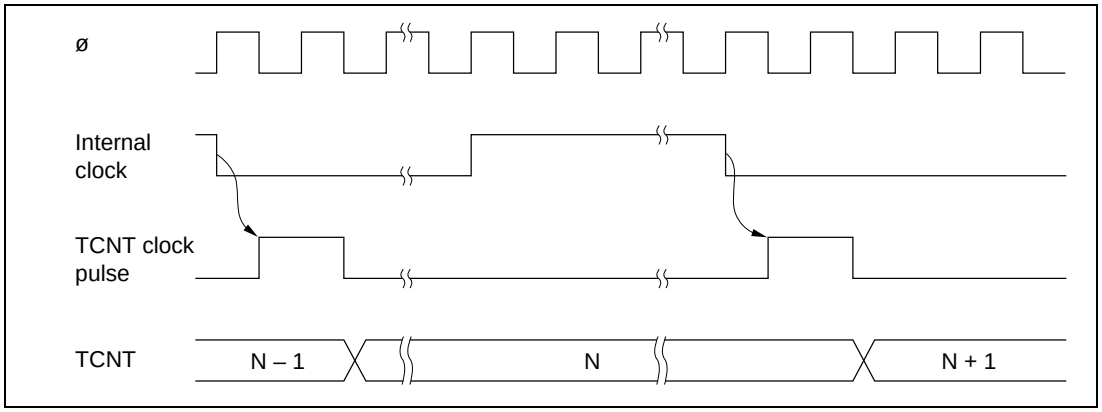


Figure 9-2 Increment Timing for Internal Clock Input

External Clock: If external clock input (TMCI) is selected, the timer counter can increment on the rising edge, the falling edge, or both edges of the external clock signal. Figure 9-3 shows incrementation on both edges of the external clock signal.

The external clock pulse width must be at least 1.5 system clock (ϕ) periods for incrementation on a single edge, and at least 2.5 system clock periods for incrementation on both edges. The counter will not increment correctly if the pulse width is shorter than these values.

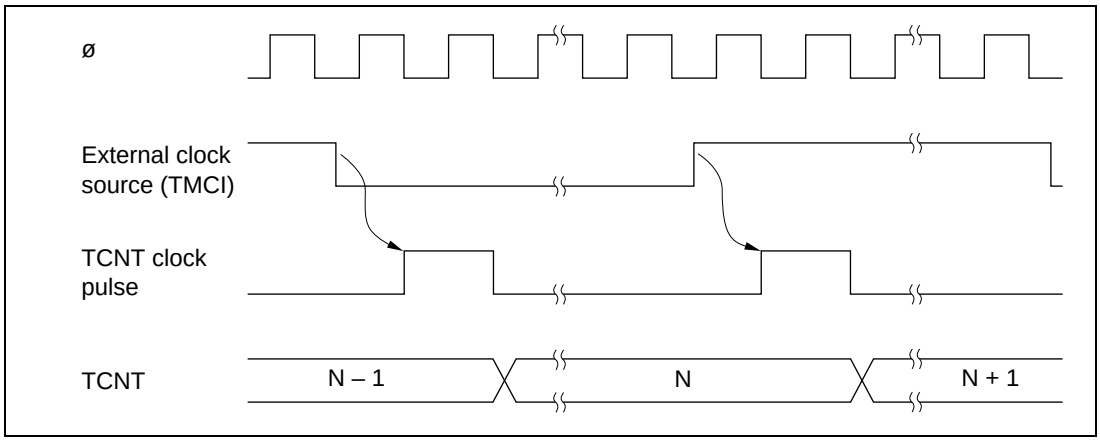


Figure 9-3 Increment Timing for External Clock Input

9.3.2 Compare-Match Timing

1. **Setting of Compare-Match Flags A and B (CMFA and CMFB):** The compare-match flags are set to 1 by an internal compare-match signal generated when the timer count matches the time constant in TCORA or TCORB. The compare-match signal is generated at the last state in which the match is true, just before the timer counter increments to a new value.

   Accordingly, when the timer count matches one of the time constants, the compare-match signal is not generated until the next period of the clock source. Figure 9-4 shows the timing of the setting of the compare-match flags.

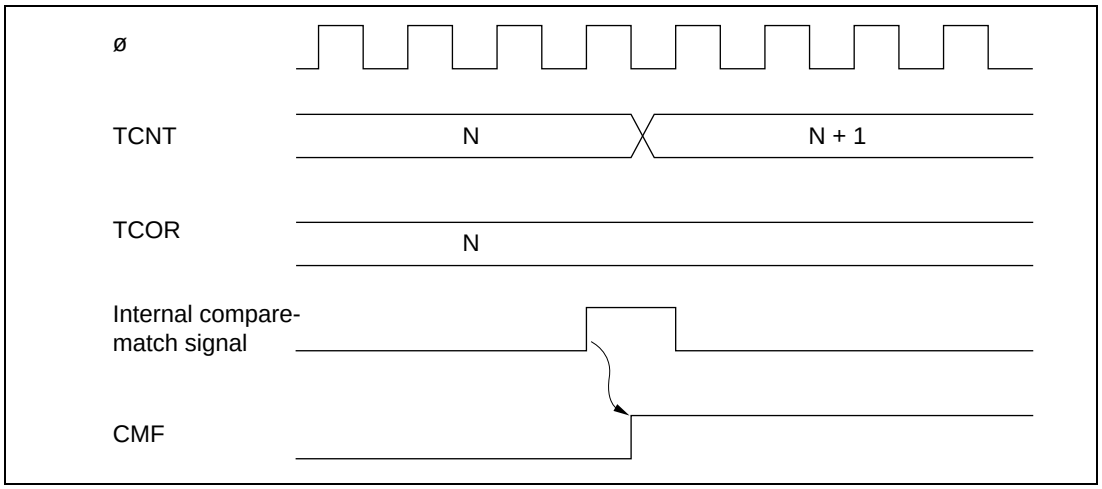


Figure 9-4 Setting of Compare-Match Flags

2. **Output Timing:** When a compare-match event occurs, the timer output changes as specified by the output select bits (OS3 to OS0) in the TCSR. Depending on these bits, the output can remain the same, change to 0, change to 1, or toggle.

   Figure 9-5 shows the timing when the output is set to toggle on compare-match A.

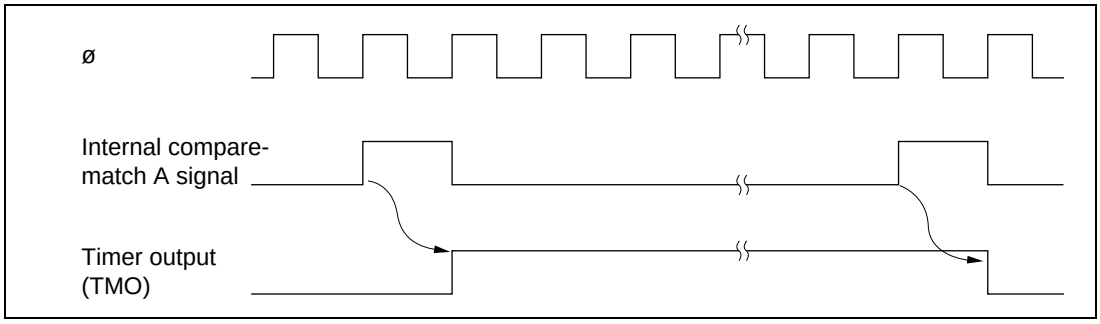


Figure 9-5 Timing of Timer Output

3. **Timing of Compare-Match Clear:** Depending on the CCLR1 and CCLR0 bits in TCR, the timer counter can be cleared when compare-match A or B occurs. Figure 9-6 shows the timing of this operation.

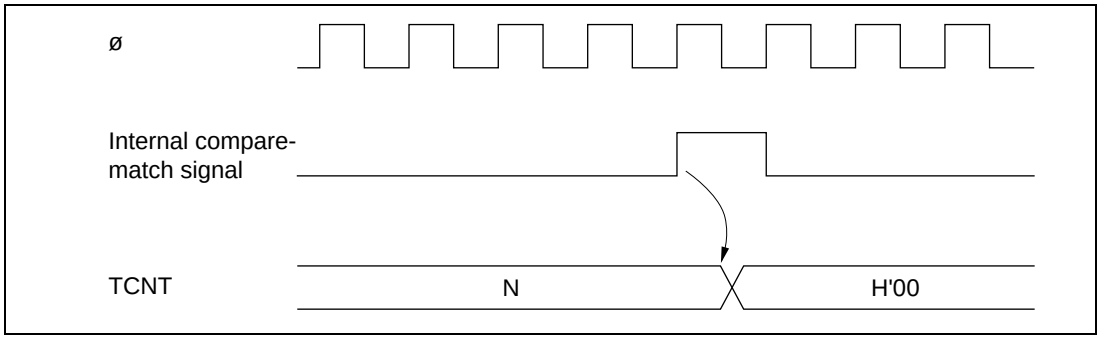


Figure 9-6 Timing of Compare-Match Clear

9.3.3 External Reset of TCNT

When the CCLR1 and CCLR0 bits in TCR are both set to 1, the timer counter is cleared on the rising edge of an external reset input. Figure 9-7 shows the timing of this operation. The timer reset pulse width must be at least 1.5 system clock (ϕ) periods.

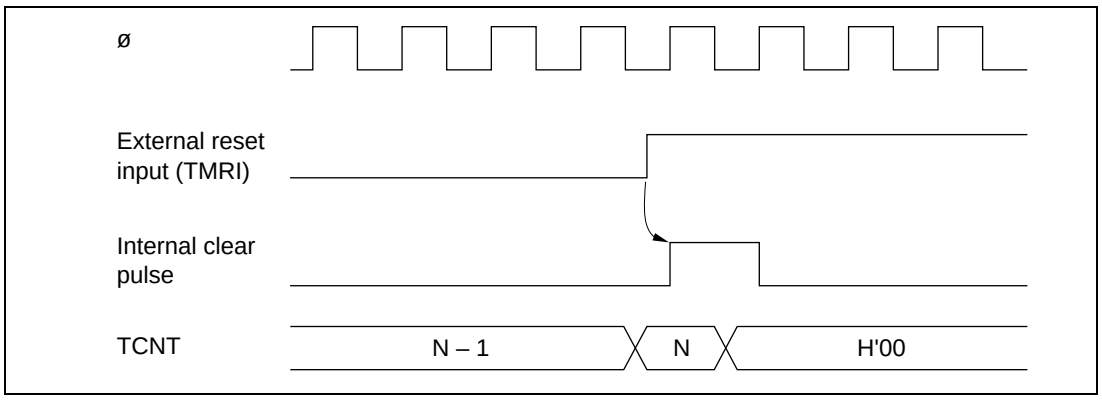


Figure 9-7 Timing of External Reset

9.3.4 Setting of Overflow Flag (OVF)

The overflow flag (OVF) in TCSR is set to 1 when the timer count overflows (changes from H'FF to H'00). Figure 9-8 shows the timing of this operation.

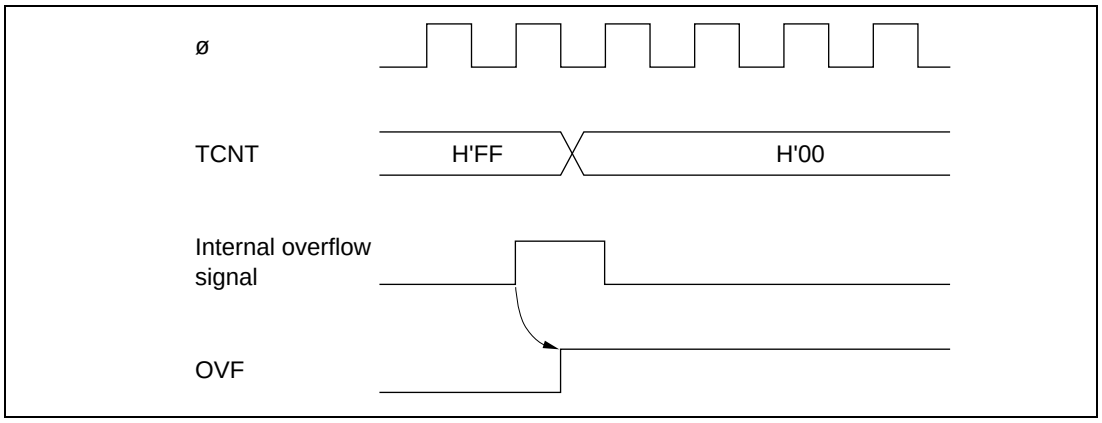


Figure 9-8 Setting of Overflow Flag (OVF)

9.4 Interrupts

Each channel in the 8-bit timer can generate three types of interrupts: compare-match A and B (CMIA and CMIB), and overflow (OVI). Each interrupt can be enabled or disabled by an enable bit in TCR. Independent signals are sent to the interrupt controller for each interrupt. Table 9-3 lists information about these interrupts.

Table 9-3 8-Bit Timer Interrupts

Interrupt	Description	Priority
CMIA	Requested by CMFA	High
CMIB	Requested by CMFB	↓
OVI	Requested by OVF	Low

9.5 Sample Application

In the example below, the 8-bit timer is used to generate a pulse output with a selected duty cycle. The control bits are set as follows:

(1) In TCR, CCLR1 is cleared to 0 and CCLR0 is set to 1 so that the timer counter is cleared when its value matches the constant in TCORA.
(2) In TCSR, bits OS3 to OS0 are set to 0110, causing the output to change to 1 on compare-match A and to 0 on compare-match B.

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.

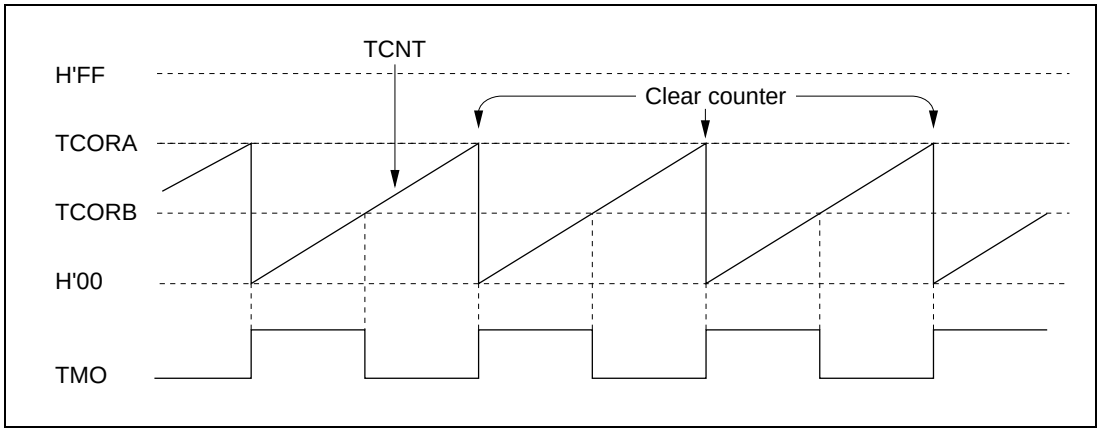


Figure 9-9 Example of Pulse Output

9.6 Application Notes

Application programmers should note that the following types of contention can occur in the 8-bit timer.

9.6.1 Contention between TCNT Write and Clear

If an internal counter clear signal is generated during the T_3 state of a write cycle to the timer counter, the clear signal takes priority and the write is not performed.

Figure 9-10 shows this type of contention.

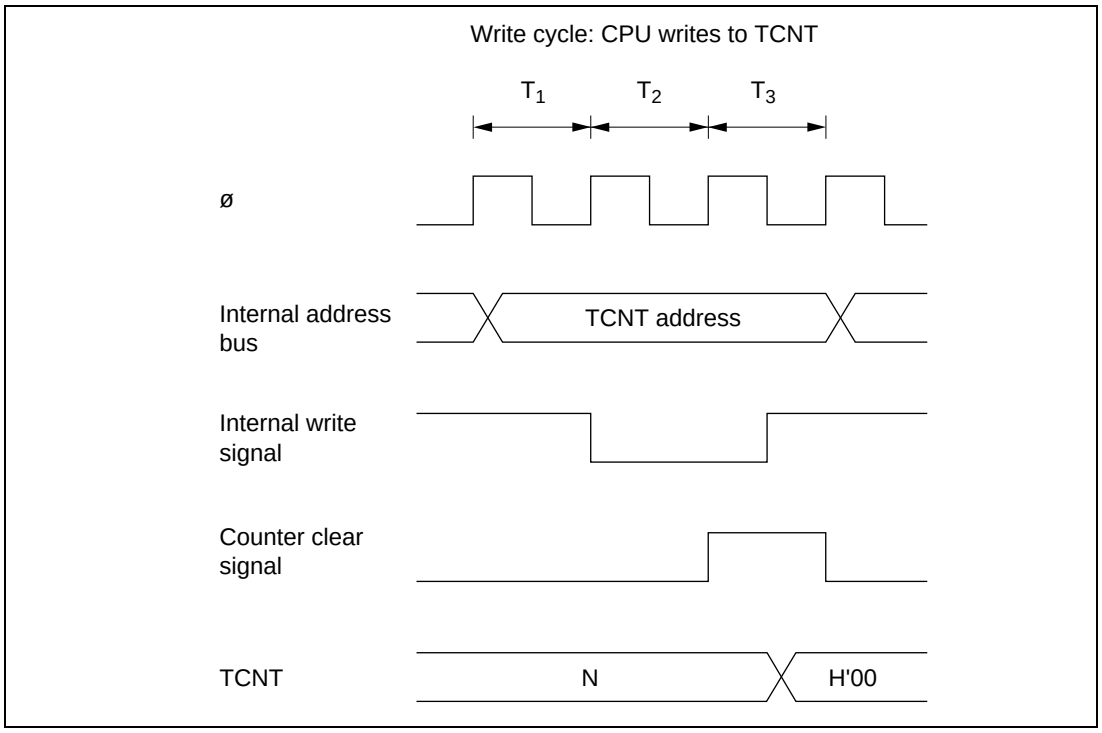


Figure 9-10 TCNT Write-Clear Contention

9.6.2 Contention between TCNT Write and Increment

If a timer counter increment pulse is generated during the T_3 state of a write cycle to the timer counter, the write takes priority and the timer counter is not incremented.

Figure 9-11 shows this type of contention.

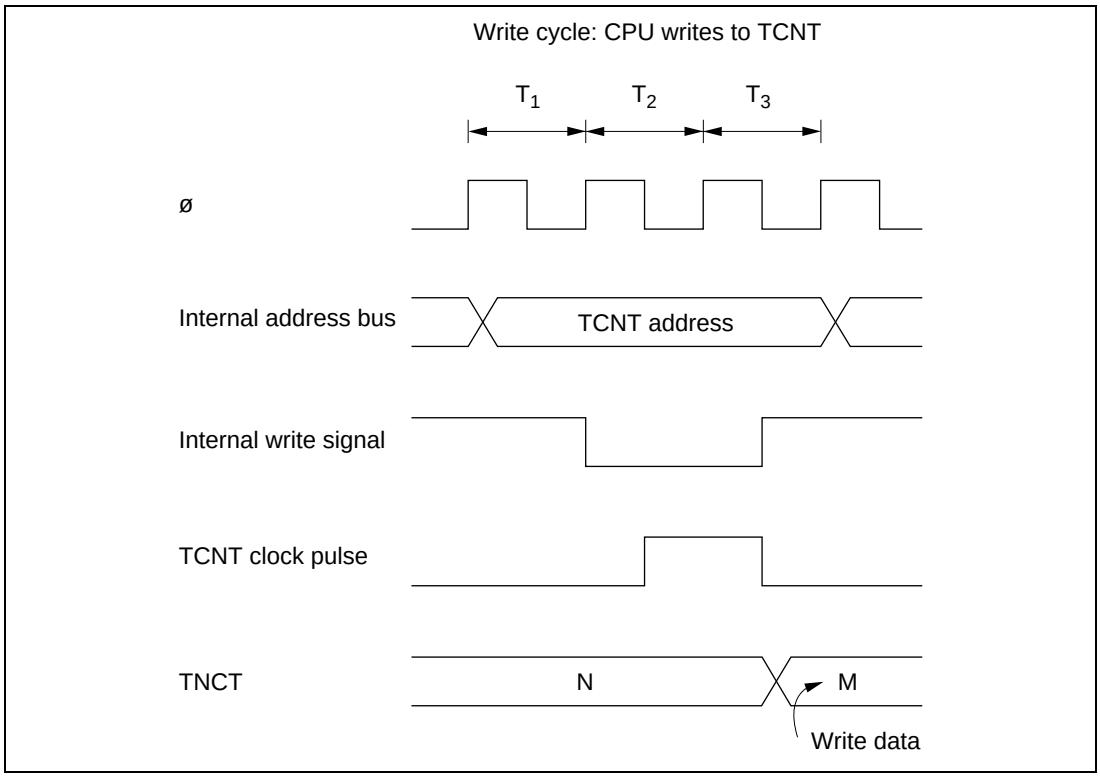


Figure 9-11 TCNT Write-Increment Contention

9.6.3 Contention between TCOR Write and Compare-Match

If a compare-match occurs during the T_3 state of a write cycle to TCOR, the write takes priority and the compare-match signal is inhibited.

Figure 9-12 shows this type of contention.

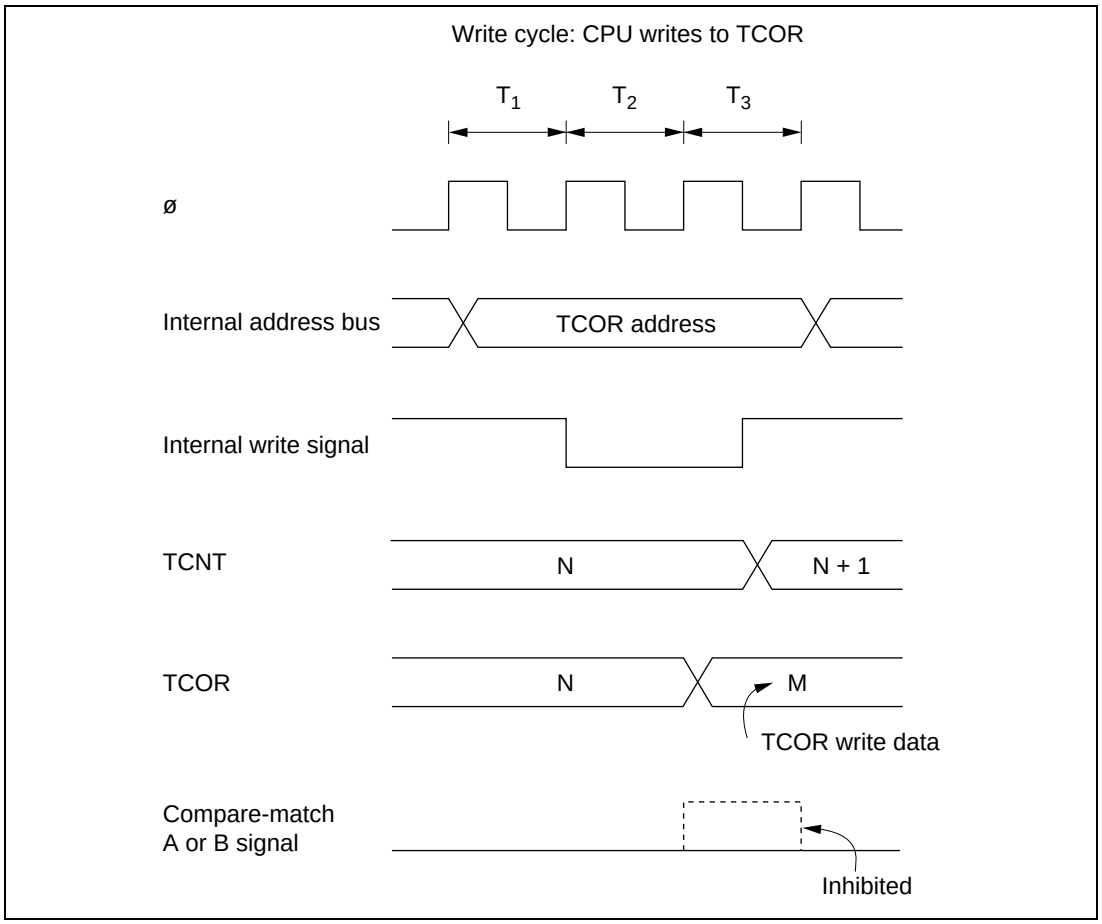


Figure 9-12 Contention between TCOR Write and Compare-Match

9.6.4 Contention between Compare-Match A and Compare-Match B

If identical time constants are written in TCORA and TCORB, causing compare-match A and B to occur simultaneously, any conflict between the output selections for compare-match A and B is resolved by following the priority order in table 9-4.

Table 9-4 Priority of Timer Output

Output Selection	Priority
Toggle	High
1 output	↓
0 output	
No change	Low

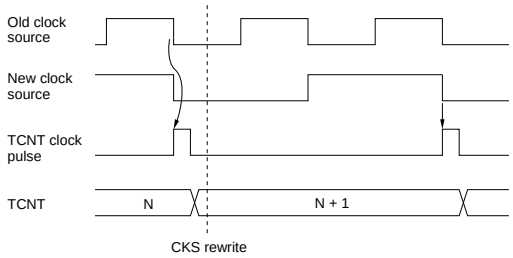
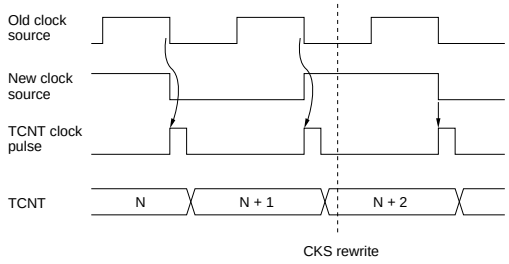
9.6.5 Increment Caused by Changing of Internal Clock Source

When an internal clock source is changed, the changeover may cause the timer counter to increment. This depends on the time at which the clock select bits (CKS1, CKS0) are rewritten, as shown in table 9-5.

The pulse that increments the timer counter is generated at the falling edge of the internal clock source signal. If clock sources are changed when the old source is high and the new source is low, as in case no. 3 in table 9-5, the changeover generates a falling edge that triggers the TCNT clock pulse and increments the timer counter.

Switching between an internal and external clock source can also cause the timer counter to increment.

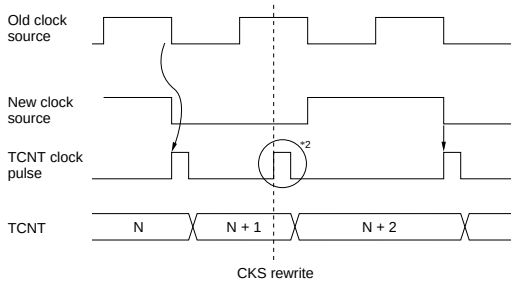
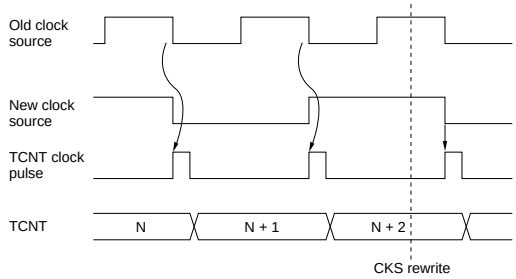
Table 9-5 Effect of Changing Internal Clock Sources

No.	Description	Timing
1	Low → low[*1]	
2	Low → high[*2]	

Notes: 1. Including a transition from low to the stopped state (CKS1 = 0, CKS0 = 0), or a transition from the stopped state to low.

2. Including a transition from the stopped state to high.

Table 9-5 Effect of Changing Internal Clock Sources (cont)

No.	Description	Timing chart
3	High → low[*1]	
4	High → high	

Notes: 1. Including a transition from high to the stopped state.
2. The switching of clock sources is regarded as a falling edge that increments TCNT.

Section 10 PWM Timers (H8/3534 Only)

10.1 Overview

The H8/3534 has an on-chip pulse-width modulation (PWM) timer module with two independent channels (PWM0 and PWM1). Both channels are functionally identical. Each PWM channel generates a rectangular output pulse with a duty cycle of 0 to 100%. The duty cycle is specified in an 8-bit duty register (DTR).

10.1.1 Features

The PWM timer module has the following features:

- Selection of eight clock sources
- Duty cycles from 0 to 100% with 1/250 resolution
- Direct or inverted PWM output, and software enable/disable control

10.1.2 Block Diagram

Figure 10-1 shows a block diagram of one PWM timer channel.

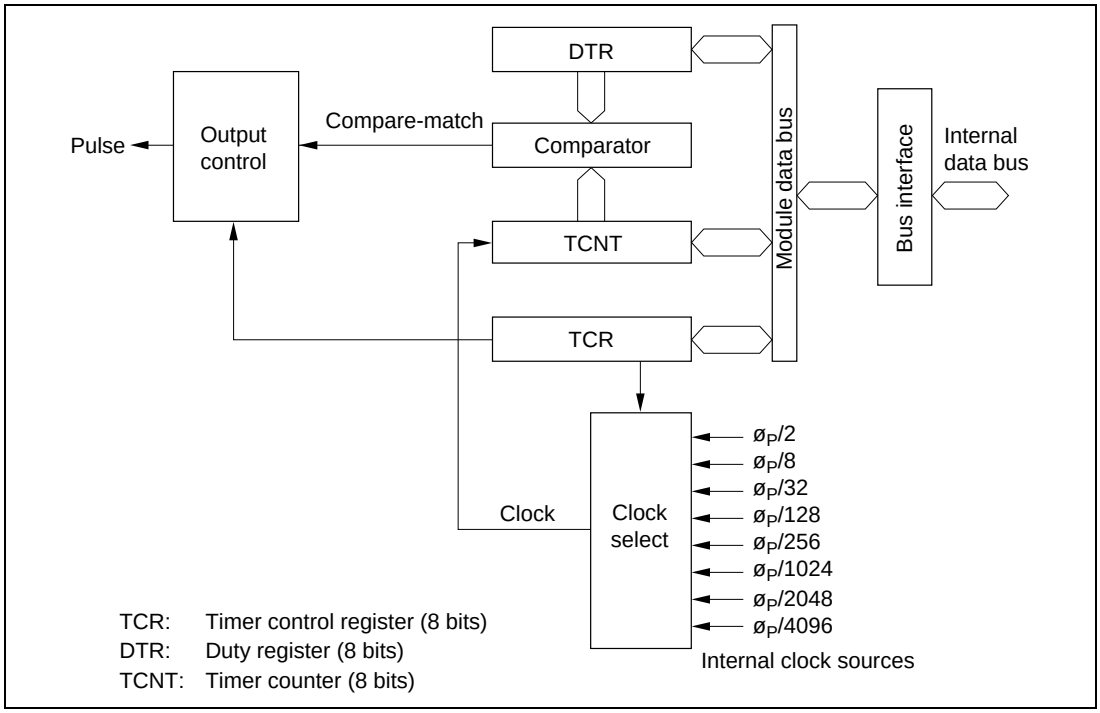


Figure 10-1 Block Diagram of PWM Timer (One Channel)

10.1.3 Input and Output Pins

Table 10-1 lists the output pins of the PWM timer module. There are no input pins.

Table 10-1 Output Pins of PWM Timer Module

Name	Abbreviation	I/O	Function
PWM0 output	PW_0	Output	Pulse output from PWM timer channel 0.
PWM1 output	PW_1	Output	Pulse output from PWM timer channel 1.

10.1.4 Register Configuration

The PWM timer module has three registers for each channel as listed in table 10-2.

Table 10-2 PWM Timer Registers

			Initial	Address	
Name	Abbreviation	R/W	Value	PWM0	PWM1
Timer control register	TCR	R/W	H'38	H'FFA0	H'FFA4
Duty register	DTR	R/W	H'FF	H'FFA1	H'FFA5
Timer counter	TCNT	R/W	H'00	H'FFA2	H'FFA6

10.2 Register Descriptions

10.2.1 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 output enable bit (OE) is set to 1 in TCR, TCNT starts counting pulses of an internal clock source selected by clock select bits 2 to 0 (CKS2 to CKS0). After counting from H'00 to H'F9, the count repeats from H'00. When TCNT changes from H'00 to to H'01, the PWM output is placed in the 1 state, unless the DTR value is H'00, in which case the duty cycle is 0% and the PWM output remains in the 0 state.

TCNT is initialized to H'00 at a reset and in the standby modes, and when the OE bit is cleared to 0.

10.2.2 Duty Register (DTR)

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

DTR is an 8-bit readable/writable register that specifies the duty cycle of the output pulse. Any duty cycle from 0% to 100% can be output by setting the corresponding value in DTR. The resolution is 1/250. Writing 0 (H'00) in DTR gives a 0% duty cycle. Writing 125 (H'7D) gives a 50% duty cycle. Writing 250 (H'FA) gives a 100% duty cycle.

The DTR and TCNT values are always compared. When the values match, the PWM output is placed in the 0 state.

DTR is double-buffered. A new value written in DTR does not become valid until after the timer count changes from H'F9 to H'00. While the OE bit is cleared to 0 in TCR, however, new values written in DTR become valid immediately. When DTR is read, the value read is the currently valid value.

DTR is initialized to H'FF by a reset and in the standby modes.

10.2.3 Timer Control Register (TCR)

Bit	7	6	5	4	3	2	1	0
	OE	OS	—	—	—	CKS2	CKS1	CKS0
Initial value	0	0	1	1	1	0	0	0
Read/Write	R/W	R/W	—	—	—	R/W	R/W	R/W

TCR is an 8-bit readable/writable register that selects the clock input to TCNT and controls PWM output.

TCR is initialized to H'38 by a reset and in standby mode.

Bit 7—Output Enable (OE): This bit enables the timer counter and the PWM output.

Bit 7 OE	Description	
0	PWM output is disabled. TCNT is cleared to H'00 and stopped.	(Initial value)
1	PWM output is enabled. TCNT runs.	

Bit 6—Output Select (OS): This bit selects positive or negative logic for the PWM output.

Bit 6 OS	Description	
0	Positive logic; positive-going PWM pulse, 1 = high	(Initial value)
1	Negative logic; negative-going PWM pulse, 1 = low	

Bits 5 to 3—Reserved: These bits cannot be modified and are always read as 1.

Bits 2, 1, and 0—Clock Select (CKS2, CKS1, and CKS0): These bits select one of eight internal clock sources obtained by dividing the supporting-module clock (ϕ_P).

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	0	$\phi_P/2$	(Initial value)
0	0	1	$\phi_P/8$	
0	1	0	$\phi_P/32$	
0	1	1	$\phi_P/128$	
1	0	0	$\phi_P/256$	
1	0	1	$\phi_P/1024$	
1	1	0	$\phi_P/2048$	
1	1	1	$\phi_P/4096$	

From the clock source frequency, the resolution, period, and frequency of the PWM output can be calculated as follows.

Resolution = 1/clock source frequency

PWM period = resolution × 250

PWM frequency = 1/PWM period

If the ϕ_P clock frequency is 10 MHz, then the resolution, period, and frequency of the PWM output for each clock source are as shown in table 10-3.

Table 10-3 PWM Timer Parameters for 10 MHz System Clock

Internal Clock Frequency	Resolution	PWM Period	PWM Frequency
$\phi_P/2$	200 ns	50 µs	20 kHz
$\phi_P/8$	800 ns	200 µs	5 kHz
$\phi_P/32$	3.2 µs	800 µs	1.25 kHz
$\phi_P/128$	12.8 µs	3.2 ms	312.5 Hz
$\phi_P/256$	25.6 µs	6.4 ms	156.3 Hz
$\phi_P/1024$	102.4 µs	25.6 ms	39.1 Hz
$\phi_P/2048$	204.8 µs	51.2 ms	19.5 Hz
$\phi_P/4096$	409.6 µs	102.4 ms	9.8 Hz

10.3 Operation

10.3.1 Timer Increment

The PWM clock source is created by dividing the system clock (ϕ). The timer counter increments on a TCNT clock pulse generated from the falling edge of the prescaler output as shown in figure 10-2.

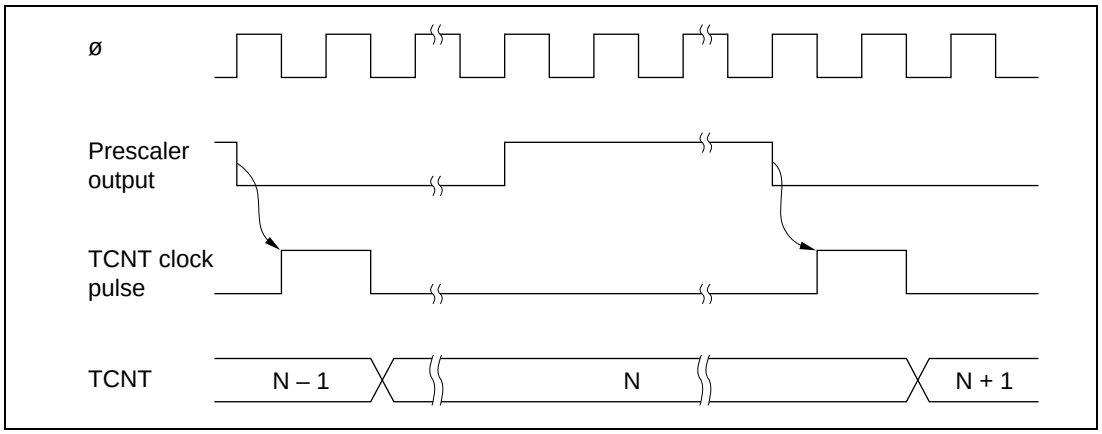


Figure 10-2 TCNT Increment Timing

10.3.2 PWM Operation

Figure 10-3 is a timing chart of the PWM operation.

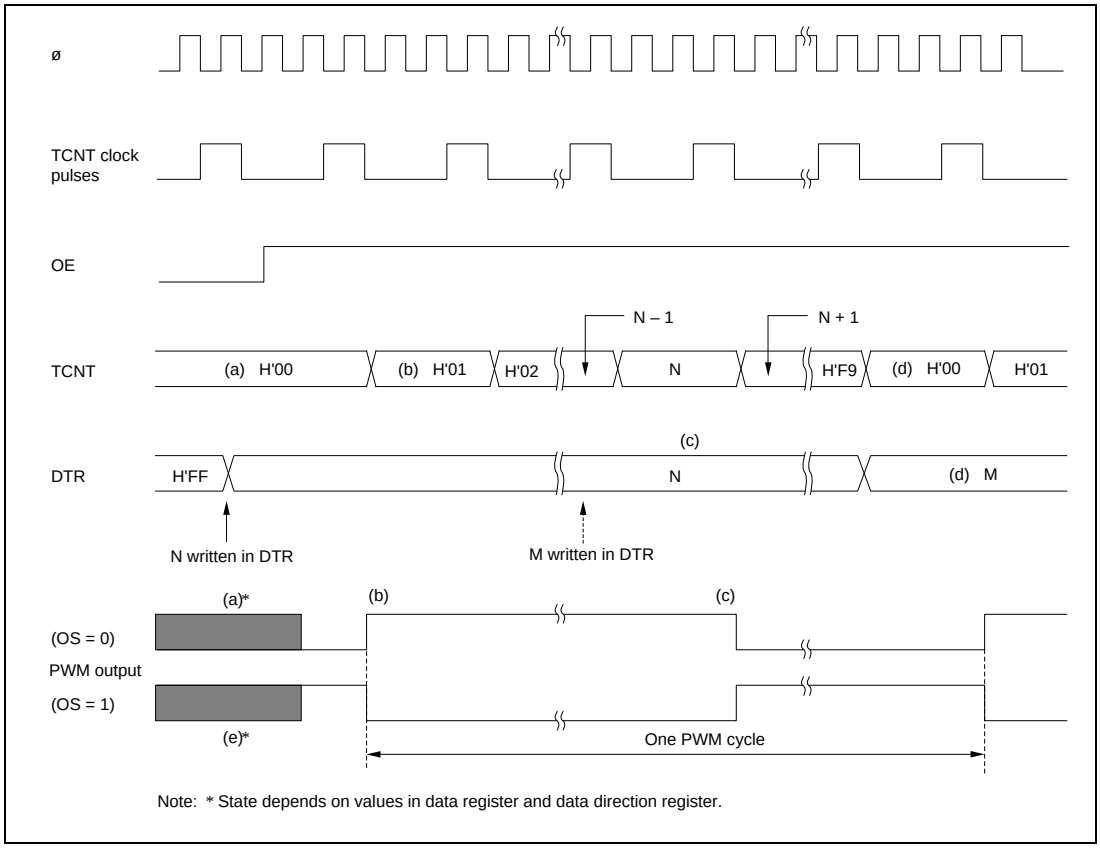


Figure 10-3 PWM Timing

1. Direct Output (OS = 0)

(1) When (OE = 0)—(a) in Figure 10-3: The timer count is held at H'00 and PWM output is inhibited. [Pin 4_6 (for PW0) or pin 4_7 (for PW1) is used for port 4 input/output, and its state depends on the corresponding port 4 data register and data direction register.] Any value (such as N in figure 10-3) written in the DTR becomes valid immediately.

(2) When (OE = 1)

i) The timer counter begins incrementing. The PWM output goes high when TCNT changes from H'00 to H'01, unless DTR = H'00. [(b) in figure 10-3]

ii) When the count passes the DTR value, the PWM output goes low. [(c) in figure 10-3]

iii) If the DTR value is changed (by writing the data "M" in figure 10-3), the new value becomes valid after the timer count changes from H'F9 to H'00. [(d) in figure 10-3]

2. Inverted Output (OS = 1)—(e) in Figure 10-3: The operation is the same except that high and low are reversed in the PWM output. [(e) in figure 10-3]

10.4 Application Notes

Some notes on the use of the PWM timer module are given below.

(1) Any necessary changes to the clock select bits (CKS2 to CKS0) and output select bit (OS) should be made before the output enable bit (OE) is set to 1.

(2) If the DTR value is H'00, the duty cycle is 0% and PWM output remains constant at 0.

If the DTR value is H'FA to H'FF, the duty cycle is 100% and PWM output remains constant at 1.

(For direct output, 0 is low and 1 is high. For inverted output, 0 is high and 1 is low.)

Section 11 Watchdog Timer

11.1 Overview

The H8/3534 and H8/3522 have an on-chip watchdog timer (WDT) that can monitor system operation by resetting the CPU or generating a nonmaskable interrupt if a system crash allows the timer count to overflow.

When this watchdog function is not needed, the watchdog timer module can be used as an interval timer. In interval timer mode, it requests an OVF interrupt at each counter overflow.

11.1.1 Features

- Selection of eight clock sources
- Selection of two modes:
  - Watchdog timer mode
  - Interval timer mode
- Counter overflow generates an interrupt request or reset:
  - Reset or NMI request in watchdog timer mode
  - OVF interrupt request in interval timer mode

11.1.2 Block Diagram

Figure 11-1 is a block diagram of the watchdog timer.

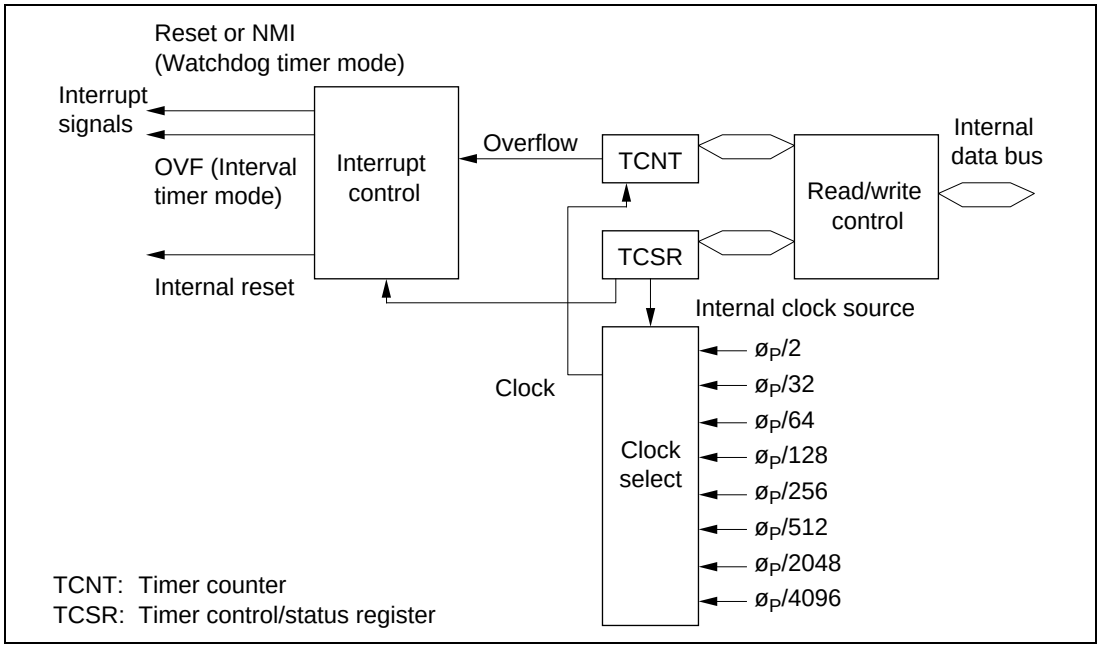


Figure 11-1 Block Diagram of Watchdog Timer

11.1.3 Register Configuration

Table 11-1 lists information on the watchdog timer registers.

Table 11-1 Register Configuration

				Addresses	
Name	**Abbreviation**	**R/W**	**Initial Value**	**Write**	**Read**
Timer control/status register	TCSR	R/(W)*	H'18	H'FFA8	H'FFA8
Timer counter	TCNT	R/W	H'00	H'FFA8	H'FFA9

Note: * Software can write a 0 to clear the status flag bits, but cannot write 1.

11.2 Register Descriptions

11.2.1 Timer Counter (TCNT)

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
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 timer enable bit (TME) in the timer control/status register (TCSR) is set to 1, the timer counter starts counting pulses of an internal clock source selected by clock select bits 2 to 0 (CKS2 to CKS0) in TCSR. When the count overflows (changes from H'FF to H'00), an overflow flag (OVF) in TCSR is set to 1.

TCNT is initialized to H'00 by a reset and when the TME bit is cleared to 0.

Note: TCNT is write-protected by a password. See Section 11.2.3, Register Access, for details.

11.2.2 Timer Control/Status Register (TCSR)

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	—	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
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

Note: * Software can write a 0 in bit 7 to clear the flag, but cannot write a 1 in this bit.

TCSR is an 8-bit readable/writable register that selects the timer mode and clock source and performs other functions.

Bits 7 to 5 and bit 3 are initialized to 0 by a reset and in the standby modes. Bits 2 to 0 are initialized to 0 by a reset, but retain their values in the standby modes.

Note: TCSR is write-protected by a password. See section 11.2.3, Register Access, for details.

Bit 7—Overflow Flag (OVF): Indicates that the watchdog timer count has overflowed from H’FF to H’00.

Bit 7 OVF	Description	
0	To clear OVF, the CPU must read OVF after it has been set to 1, then write a 0 in this bit	(Initial value)
1	Set to 1 when TCNT changes from H'FF to H'00	

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether to operate in watchdog timer mode or interval timer mode. When TCNT overflows, an OVF interrupt request is sent to the CPU in interval timer mode. For watchdog timer mode, a reset or NMI interrupt is requested.

Bit 6 WT/$\overline{IT}$	Description	
0	Interval timer mode (OVF request)	(Initial value)
1	Watchdog timer mode (reset or NMI request)	

Bit 5—Timer Enable (TME): Enables or disables the timer.

Bit 5 TME	Description	
0	TCNT is initialized to H'00 and stopped	(Initial value)
1	TCNT runs and requests a reset or an interrupt when it overflows	

Bit 4—Reserved: This bit cannot be modified and is always read as 1.

Bit 3: Reset or NMI Select (RST/$\overline{NMI}$): Selects either an internal reset or the NMI function at watchdog timer overflow.

Bit 3 RST/$\overline{NMI}$	Description	
0	NMI function enabled	(Initial value)
1	Reset function enabled	

Bits 2 to 0— Clock Select (CKS2–CKS0): These bits select one of eight clock sources obtained by dividing the system clock (ϕ).

The overflow interval is the time from when the watchdog timer counter begins counting from H'00 until an overflow occurs. In interval timer mode, OVF interrupts are requested at this interval.

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Clock Source	Overflow Interval (ϕ_P = 10 MHz)	
0	0	0	ϕ_P/2	51.2 µs	(Initial value)
0	0	1	ϕ_P/32	819.2 µs	
0	1	0	ϕ_P/64	1.6 ms	
0	1	1	ϕ_P/128	3.3 ms	
1	0	0	ϕ_P/256	6.6 ms	
1	0	1	ϕ_P/512	13.1 ms	
1	1	0	ϕ_P/2048	52.4 ms	
1	1	1	ϕ_P/4096	104.9 ms	

11.2.3 Register Access

The watchdog timer's TCNT and TCSR registers are more difficult to write to than other registers. The procedures for writing and reading these registers are given below.

Writing to TCNT and TCSR: Word access is required. Byte data transfer instructions cannot be used for write access.

The TCNT and TCSR registers have the same write address. The write data must be contained in the lower byte of a word written at this address. The upper byte must contain H'5A (password for TCNT) or H'A5 (password for TCSR). See figure 11-2. The result of the access depicted in figure 11-2 is to transfer the write data from the lower byte to TCNT or TCSR.

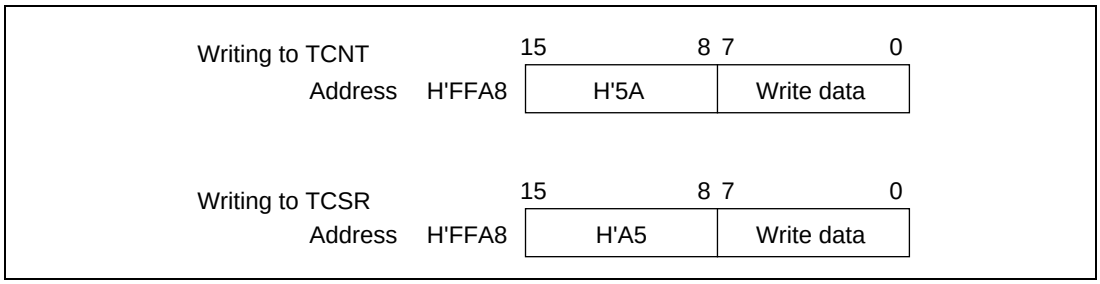


Figure 11-2 Writing to TCNT and TCSR

Reading TCNT and TCSR: The read addresses are H'FFA8 for TCSR and H'FFA9 for TCNT, as indicated in table 11-2.

These two registers are read like other registers. Byte access instructions can be used.

Table 11-2 Read Addresses of TCNT and TCSR

Read Address	Register
H'FFA8	TCSR
H'FFA9	TCNT

11.3 Operation

11.3.1 Watchdog Timer Mode

The watchdog timer function begins operating when software sets the WT/$\overline{IT}$ and TME bits to 1 in TCSR. Thereafter, software should periodically rewrite the contents of the timer counter (normally by writing H'00) to prevent the count from overflowing. If a program crash allows the timer count to overflow, the entire chip is reset for 518 system clocks (518 ϕ), or an NMI interrupt is requested. Figure 11-3 shows the operation.

NMI requests from the watchdog timer have the same vector as NMI requests from the $\overline{NMI}$ pin. Avoid simultaneous handling of watchdog timer NMI requests and NMI requests from pin $\overline{NMI}$.

A reset from the watchdog timer has the same vector as an external reset from the $\overline{RES}$ pin. The reset source can be determined by the XRST bit in SYSCR.

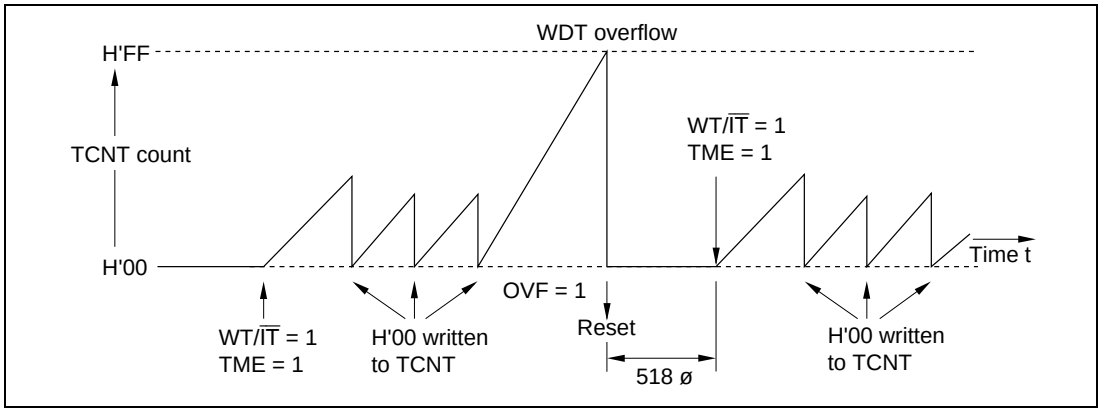


Figure 11-3 Operation in Watchdog Timer Mode

11.3.2 Interval Timer Mode

Interval timer operation begins when the WT/$\overline{IT}$ bit is cleared to 0 and the TME bit is set to 1.

In interval timer mode, an OVF request is generated each time the timer count overflows. This function can be used to generate OVF requests at regular intervals. See figure 11-4.

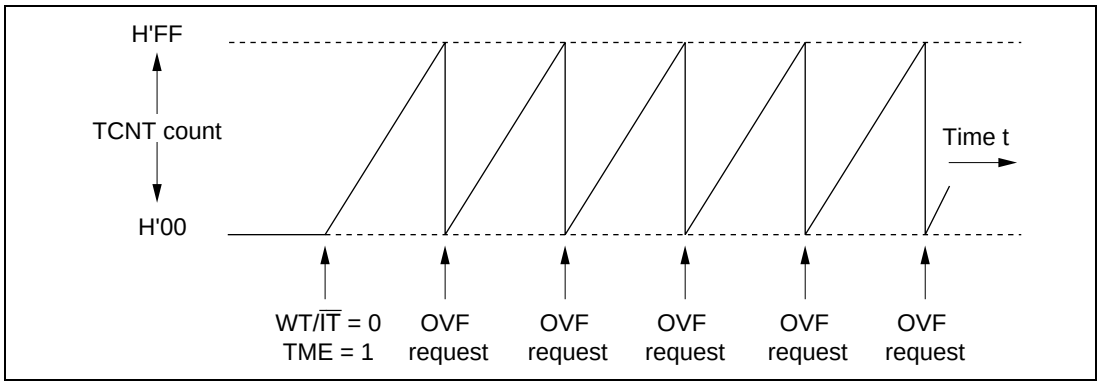


Figure 11-4 Operation in Interval Timer Mode

11.3.3 Setting the Overflow Flag

The OVF bit is set to 1 when the timer count overflows. Simultaneously, the WDT module requests an internal reset, NMI, or OVF interrupt. The timing is shown in figure 11-5.

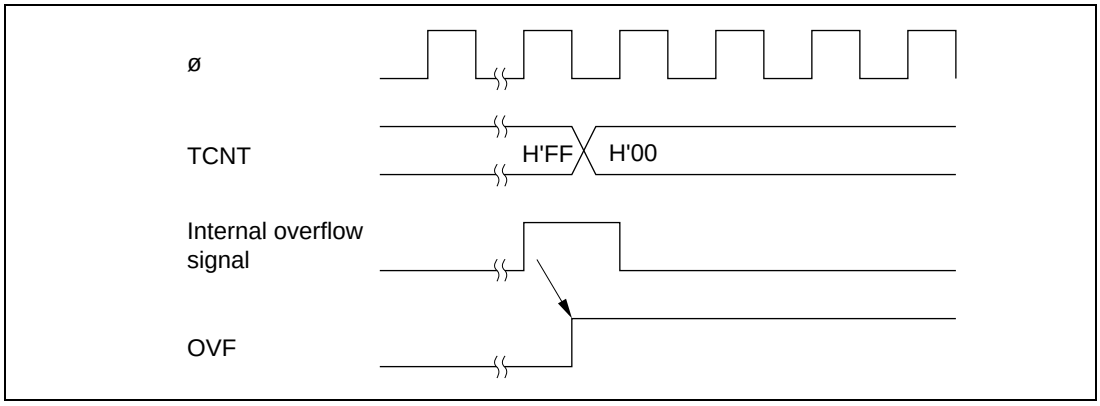


Figure 11-5 Setting the OVF Bit

11.4 Application Notes

11.4.1 Contention between TCNT Write and Increment

If a timer counter clock pulse is generated during the T_3 state of a write cycle to the timer counter, the write takes priority and the timer counter is not incremented. See figure 11-6.

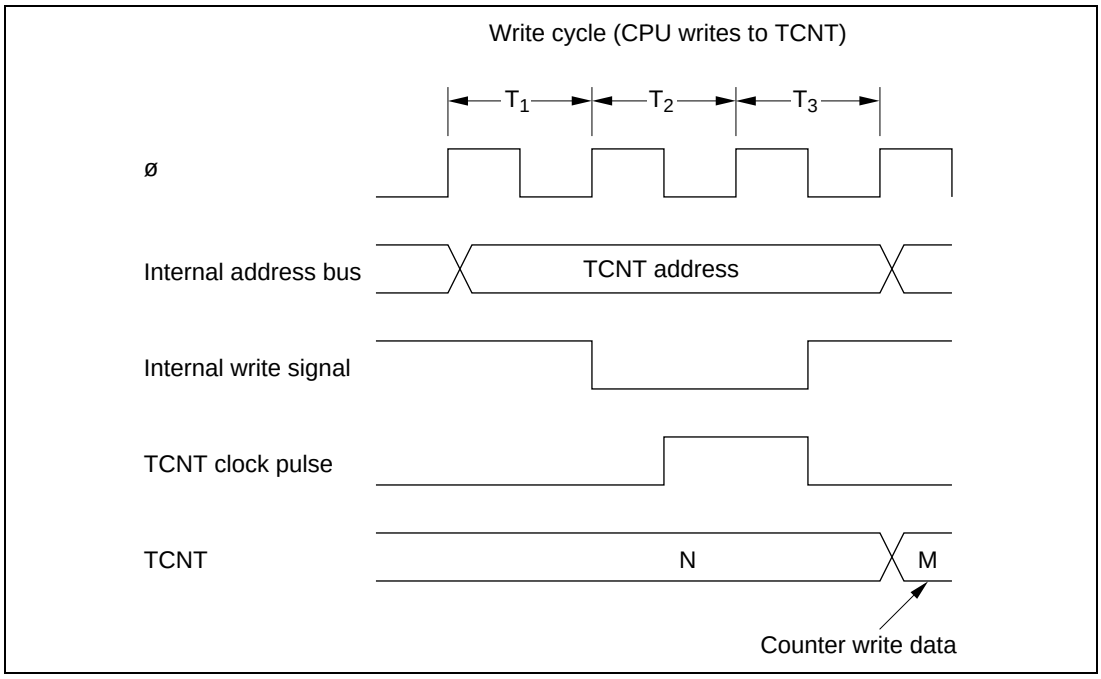


Figure 11-6 TCNT Write-Increment Contention

11.4.2 Changing the Clock Select Bits (CKS2 to CKS0)

Software should stop the watchdog timer (by clearing the TME bit to 0) before changing the value of the clock select bits. If the clock select bits are modified while the watchdog timer is running, the timer count may be incremented incorrectly.

11.4.3 Recovery from Software Standby Mode

TCSR bits, except bits 0 to 2, and the TCNT counter are reset when the chip recovers from software standby mode. Re-initialize the watchdog timer as necessary to resume normal operation.

Section 12 Serial Communication Interface

12.1 Overview

The H8/3534 includes two serial communication interface channels (SCI0 and SCI1), and the H8/3522 one channel, for transferring serial data to and from other chips. Either synchronous or asynchronous communication can be selected.

12.1.1 Features

The features of the on-chip serial communication interface are:

- Asynchronous mode

  The H8/3534 and H8/3522 can communicate with a UART (Universal Asynchronous Receiver/Transmitter), ACIA (Asynchronous Communication Interface Adapter), or other chip that employs standard asynchronous serial communication. A multiprocessor communication function is also provided for communication with other processors. Twelve data formats are available.

  — Data length: 7 or 8 bits
  — Stop bit length: 1 or 2 bits
  — Parity: Even, odd, or none
  — Multiprocessor bit: 1 or 0
  — Error detection: Parity, overrun, and framing errors
  — Break detection: When a framing error occurs, the break condition can be detected by reading the level of the RxD line directly.
- Synchronous mode

  The SCI can communicate with chips able to perform clocked synchronous data transfer.

  — Data length: 8 bits
  — Error detection: Overrun errors
- Full duplex communication

  The transmitting and receiving sections are independent, so each channel can transmit and receive simultaneously. Both the transmit and receive sections use double buffering, so continuous data transfer is possible in either direction.
- Built-in baud rate generator

  Any specified bit rate can be generated.

- Internal or external clock source

  The SCI can operate on an internal clock signal from the baud rate generator, or an external clock signal input at the SCK0 or SCK1 pin.
- Four interrupts

  TDR-empty, TSR-empty, receive-end, and receive-error interrupts are requested independently.

12.1.2 Block Diagram

Figure 12-1 shows a block diagram of one serial communication interface channel.

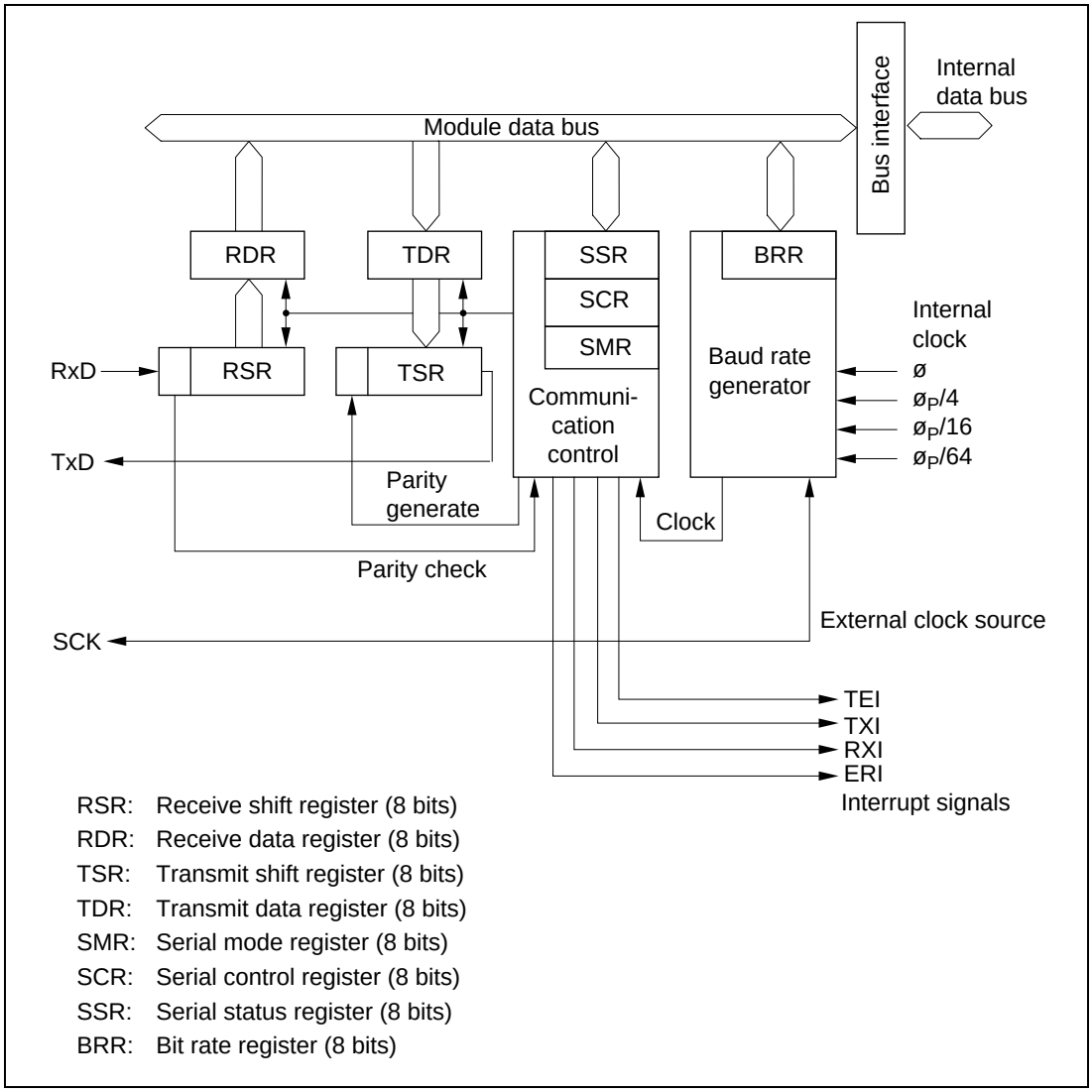


Figure 12-1 Block Diagram of Serial Communication Interface

12.1.3 Input and Output Pins

Table 12-1 lists the input and output pins used by the SCI module.

Table 12-1 SCI Input/Output Pins

Channel	Name	Abbr.	I/O	Function
0	Serial clock input/output	SCK_0 (SCK)	Input/output	SCI0 clock input and output
	Receive data input	RxD_0 (RxD)	Input	SCI0 receive data input
	Transmit data output	TxD_0 (TxD)	Output	SCI0 transmit data output
1	Serial clock input/output	SCK_1	Input/output	SCI1 clock input and output
[H8/3534	Receive data input	RxD_1	Input	SCI1 receive data input
only]	Transmit data output	TxD_1	Output	SCI1 transmit data output

Note: In this manual, the channel subscript has been deleted, and only SCK, RxD, and TxD are used.

12.1.4 Register Configuration

Table 12-2 lists the SCI registers. These registers specify the operating mode (synchronous or asynchronous), data format and bit rate, and control the transmit and receive sections.

Table 12-2 SCI Registers

Channel	Name	Abbr.	R/W	Value	Address
0	Receive shift register	RSR	—*[1]	—	—
	Receive data register	RDR	R	H'00	H'FFDD
	Transmit shift register	TSR	—*[1]	—	—
	Transmit data register	TDR	R/W	H'FF	H'FFDB
	Serial mode register	SMR	R/W	H'00	H'FFD8
	Serial control register	SCR	R/W	H'00	H'FFDA
	Serial status register	SSR	R/(W)*[2]	H'84	H'FFDC
	Bit rate register	BRR	R/W	H'FF	H'FFD9
1	Receive shift register	RSR	—*[1]	—	—
[H8/3534	Receive data register	RDR	R	H'00	H'FF8D
only]	Transmit shift register	TSR	—*[1]	—	—
	Transmit data register	TDR	R/W	H'FF	H'FF8B
	Serial mode register	SMR	R/W	H'00	H'FF88
	Serial control register	SCR	R/W	H'00	H'FF8A
	Serial status register	SSR	R/(W)*[2]	H'84	H'FF8C
	Bit rate register	BRR	R/W	H'FF	H'FF89
0 and 1	Serial/timer control register	STCR	R/W	H'00	H'FFC3

Note: 1. Cannot be read or written to.
2. Software can write a 0 to clear the flags in bits 7 to 3, but cannot write 1 in these bits.

12.2 Register Descriptions

12.2.1 Receive Shift Register (RSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

RSR is a shift register that converts incoming serial data to parallel data. When one data character has been received, it is transferred to the receive data register (RDR).

The CPU cannot read or write RSR directly.

12.2.2 Receive Data Register (RDR)

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

RDR stores received data. As each character is received, it is transferred from RSR to RDR, enabling RSR to receive the next character. This double-buffering allows the SCI to receive data continuously.

RDR is a read-only register. RDR is initialized to H'00 by a reset and in the standby modes.

12.2.3 Transmit Shift Register (TSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

TSR is a shift register that converts parallel data to serial transmit data. When transmission of one character is completed, the next character is moved from the transmit data register (TDR) to TSR and transmission of that character begins. If the TDRE bit is still set to 1, however, nothing is transferred to TSR.

The CPU cannot read or write TSR directly.

12.2.4 Transmit Data Register (TDR)

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

TDR is an 8-bit readable/writable register that holds the next data to be transmitted. When TSR becomes empty, the data written in TDR is transferred to TSR. Continuous data transmission is possible by writing the next data in TDR while the current data is being transmitted from TSR.

TDR is initialized to H'FF by a reset and in the standby modes.

12.2.5 Serial Mode Register (SMR)

Bit	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SMR is an 8-bit readable/writable register that controls the communication format and selects the clock source of the on-chip baud rate generator. It is initialized to H'00 by a reset and in the standby modes. For further information on the SMR settings and communication formats, see tables 12-5 and 12-7 in section 12.3, Operation.

Bit 7—Communication Mode (C/$\overline{A}$): This bit selects asynchronous or synchronous communication mode.

Bit 7 C/$\overline{A}$	Description	
0	Asynchronous communication	(Initial value)
1	Synchronous communication	

Bit 6—Character Length (CHR): This bit selects the character length in asynchronous mode. It is ignored in synchronous mode.

Bit 6 CHR	Description	
0	8 bits per character	(Initial value)
1	7 bits per character (Bits 0 to 6 of TDR and RDR are used for transmitting and receiving, respectively.)	

Bit 5—Parity Enable (PE): This bit selects whether to add a parity bit in asynchronous mode. It is ignored in synchronous mode, and when a multiprocessor format is used.

Bit 5 PE	Description	
0	Transmit: No parity bit is added. Receive: Parity is not checked.	(Initial value)
1	Transmit: A parity bit is added. Receive: Parity is checked.	

Bit 4—Parity Mode (O/$\overline{E}$): In asynchronous mode, when parity is enabled (PE = 1), this bit selects even or odd parity.

Even parity means that a parity bit is added to the data bits for each character to make the total number of 1’s even. Odd parity means that the total number of 1’s is made odd.

This bit is ignored when PE = 0, or when a multiprocessor format is used. It is also ignored in synchronous mode.

Bit 4 O/$\overline{E}$	Description	
0	Even parity	(Initial value)
1	Odd parity	

Bit 3—Stop Bit Length (STOP): This bit selects the number of stop bits. It is ignored in synchronous mode.

Bit 3 STOP	Description	
0	One stop bit Transmit: One stop bit is added. Receive: One stop bit is checked to detect framing errors.	(Initial value)
1	Two stop bits Transmit: Two stop bits are added. Receive: The first stop bit is checked to detect framing errors. If the second stop bit is a space (0), it is regarded as the next start bit.	

Bit 2—Multiprocessor Mode (MP): This bit selects the multiprocessor format in asynchronous communication. When multiprocessor format is selected, the parity settings of the parity enable bit (PE) and parity mode bit (O/$\overline{E}$) are ignored. The MP bit is ignored in synchronous communication.

The MP bit is valid only when the MPE bit in the serial/timer control register (STCR) is set to 1. When the MPE bit is cleared to 0, the multiprocessor communication function is disabled regardless of the setting of the MP bit.

Bit 2 MP	Description	
0	Multiprocessor communication function is disabled.	(Initial value)
1	Multiprocessor communication function is enabled.	

Bits 1 and 0—Clock Select 1 and 0 (CKS1 and CKS0): These bits select the clock source of the on-chip baud rate generator.

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	ϕ clock	(Initial value)
0	1	ϕ_P/4 clock	
1	0	ϕ_P/16 clock	
1	1	ϕ_P/64 clock	

12.2.6 Serial Control Register (SCR)

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR is an 8-bit readable/writable register that enables or disables various SCI functions. It is initialized to H'00 by a reset and in the standby modes.

Bit 7—Transmit Interrupt Enable (TIE): This bit enables or disables the TDR-empty interrupt (TXI) requested when the transmit data register empty (TDRE) bit in the serial status register (SSR) is set to 1.

Bit 7 TIE	Description	
0	The TDR-empty interrupt request (TXI) is disabled.	(Initial value)
1	The TDR-empty interrupt request (TXI) is enabled.	

Bit 6—Receive Interrupt Enable (RIE): This bit enables or disables the receive-end interrupt (RXI) requested when the receive data register full (RDRF) bit in the serial status register (SSR) is set to 1, and the receive error interrupt (ERI) requested when the overrun error (ORER), framing error (FER), or parity error (PER) bit in the serial status register (SSR) is set to 1.

Bit 6 RIE	Description	
0	The receive-end interrupt (RXI) and receive-error (ERI) requests are disabled.	(Initial value)
1	The receive-end interrupt (RXI) and receive-error (ERI) requests are enabled.	

Bit 5—Transmit Enable (TE): This bit enables or disables the transmit function. When the transmit function is enabled, the TxD pin is automatically used for output. When the transmit function is disabled, the TxD pin can be used as a general-purpose I/O port.

Bit 5 TE	Description	
0	The transmit function is disabled. The TxD pin can be used for general-purpose I/O.	(Initial value)
1	The transmit function is enabled. The TxD pin is used for output.	

Bit 4—Receive Enable (RE): This bit enables or disables the receive function. When the receive function is enabled, the RxD pin is automatically used for input. When the receive function is disabled, the RxD pin is available as a general-purpose I/O port.

Bit 4 RE	Description	
0	The receive function is disabled. The RxD pin can be used for general-purpose I/O.	(Initial value)
1	The receive function is enabled. The RxD pin is used for input.	

Bit 3—Multiprocessor Interrupt Enable (MPIE): When serial data is received in a multiprocessor format, this bit enables or disables the receive-end interrupt (RXI) and receive-error interrupt (ERI) until data with the multiprocessor bit set to 1 is received. It also enables or disables the transfer of received data from RSR to RDR, and enables or disables setting of the RDRF, FER, PER, and ORER bits in the serial status register (SSR).

The MPIE bit is ignored when the MP bit is cleared to 0, and in synchronous mode.

Clearing the MPIE bit to 0 disables the multiprocessor receive interrupt function. In this condition data is received regardless of the value of the multiprocessor bit in the receive data.

Setting the MPIE bit to 1 enables the multiprocessor receive interrupt function. In this condition, if the multiprocessor bit in the receive data is 0, the receive-end interrupt (RXI) and receive-error interrupt (ERI) are disabled, the receive data is not transferred from RSR to RDR, and the RDRF, FER, PER, and ORER bits in the serial status register (SSR) are not set. If the multiprocessor bit is 1, however, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0, the receive data is transferred from RSR to RDR, the FER, PER, and ORER bits can be set, and the receive-end and receive-error interrupts are enabled.

Bit 3 MPIE	Description	
0	The multiprocessor receive interrupt function is disabled. (Normal receive operation)	(Initial value)
1	The multiprocessor receive interrupt function is enabled. During the interval before data with the multiprocessor bit set to 1 is received, the receive interrupt request (RXI) and receive-error interrupt request (ERI) are disabled, the RDRF, FER, PER, and ORER bits are not set in the serial status register (SSR), and no data is transferred from the RSR to the RDR. The MPIE bit is cleared at the following times: (1) When 0 is written in MPIE. (2) When data with the multiprocessor bit set to 1 is received	

Bit 2—Transmit-End Interrupt Enable (TEIE): This bit enables or disables the TSR-empty interrupt (TEI) requested when the transmit-end bit (TEND) in the serial status register (SSR) is set to 1.

Bit 2 TEIE	Description	
0	The TSR-empty interrupt request (TEI) is disabled.	(Initial value)
1	The TSR-empty interrupt request (TEI) is enabled.	

Bit 1—Clock Enable 1 (CKE1): This bit selects the internal or external clock source for the baud rate generator. When the external clock source is selected, the SCK pin is automatically used for input of the external clock signal.

Bit 1 CKE1	Description	
0	Internal clock source When C/$\overline{A}$ = 1, the serial clock signal is output at the SCK pin. When C/$\overline{A}$ = 0, output depends on the CKE0 bit.	(Initial value)
1	External clock source. The SCK pin is used for input.	

Bit 0—Clock Enable 0 (CKE0): When an internal clock source is used in asynchronous mode, this bit enables or disables serial clock output at the SCK pin.

This bit is ignored when the external clock is selected, or when synchronous mode is selected.

For further information on the communication format and clock source selection, see table 12-6 in section 12.3, Operation.

Bit 0 CKE0	Description	
0	The SCK pin is not used by the SCI (and is available as a general-purpose I/O port).	(Initial value)
1	The SCK pin is used for serial clock output.	

12.2.7 Serial Status Register (SSR)

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * Software can write a 0 to clear the flags, but cannot write a 1 in these bits.

SSR is an 8-bit register that indicates transmit and receive status. It is initialized to H'84 by a reset and in the standby modes.

Bit 7—Transmit Data Register Empty (TDRE): This bit indicates when transmit data can safely be written in TDR.

Bit 7 TDRE	Description
0	To clear TDRE, the CPU must read TDRE after it has been set to 1, then write a 0 in this bit.
1	This bit is set to 1 at the following times: (Initial value) (1) When TDR contents are transferred to TSR. (2) When the TE bit in SCR is cleared to 0.

Bit 6—Receive Data Register Full (RDRF): This bit indicates when one character has been received and transferred to RDR.

Bit 6 RDRF	Description
0	To clear RDRF, the CPU must read RDRF after it has been set to 1, then write a 0 in this bit. (Initial value)
1	This bit is set to 1 when one character is received without error and transferred from RSR to RDR.

Bit 5—Overrun Error (ORER): This bit indicates an overrun error during reception.

Bit 5 ORER	Description	
0	To clear ORER, the CPU must read ORER after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 if reception of the next character ends while the receive data register is still full (RDRF = 1).	

Bit 4—Framing Error (FER): This bit indicates a framing error during data reception in asynchronous mode. It has no meaning in synchronous mode.

Bit 4 FER	Description	
0	To clear FER, the CPU must read FER after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 if a framing error occurs (stop bit = 0).	

Bit 3—Parity Error (PER): This bit indicates a parity error during data reception in the asynchronous mode, when a communication format with parity bits is used.

This bit has no meaning in the synchronous mode, or when a communication format without parity bits is used.

Bit 3 PER	Description	
0	To clear PER, the CPU must read PER after it has been set to 1, then write a 0 in this bit.	(Initial value)
1	This bit is set to 1 when a parity error occurs (the parity of the received data does not match the parity selected by the O/$\overline{E}$ bit in SMR).	

Bit 2—Transmit End (TEND): This bit indicates that the serial communication interface has stopped transmitting because there was no valid data in TDR when the last bit of the current character was transmitted. The TEND bit is also set to 1 when the TE bit in the serial control register (SCR) is cleared to 0.

The TEND bit is a read-only bit and cannot be modified directly. To use the TEI interrupt, first start transmitting data, which clears TEND to 0, then set TEIE to 1.

Bit 2 TEND	Description	
0	To clear TEND, the CPU must read TDRE after TDRE has been set to 1, then write a 0 in TDRE	
1	This bit is set to 1 when: (1) TE = 0 (2) TDRE = 1 at the end of transmission of a character	(Initial value)

Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in data received in a multiprocessor format in asynchronous communication mode. This bit retains its previous value in synchronous mode, when a multiprocessor format is not used, or when the RE bit is cleared to 0 even if a multiprocessor format is used.

MPB can be read but not written.

Bit 1 MPB	Description	
0	Multiprocessor bit = 0 in receive data.	(Initial value)
1	Multiprocessor bit = 1 in receive data.	

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit inserted in transmit data when a multiprocessor format is used in asynchronous communication mode. The MPBT bit is double-buffered in the same way as TSR and TDR. The MPBT bit has no effect in synchronous mode, or when a multiprocessor format is not used.

Bit 0 MPBT	Description	
0	Multiprocessor bit = 0 in transmit data.	(Initial value)
1	Multiprocessor bit = 1 in transmit data.	

12.2.8 Bit Rate Register (BRR)

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
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, together with the CKS1 and CKS0 bits in SMR, determines the bit rate output by the baud rate generator.

BRR is initialized to H'FF by a reset and in the standby modes.

Tables 12-3 and 12-4 show examples of BRR settings.

Table 12-3 Examples of BRR Settings in Asynchronous Mode (When $\phi_P = \phi$)

	ϕ Frequency (MHz)					
	2			2.097152		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	1	141	+0.03	1	148	–0.04
150	1	103	+0.16	1	108	+0.21
300	0	207	+0.16	0	217	+0.21
600	0	103	+0.16	0	108	+0.21
1200	0	51	+0.16	0	54	–0.70
2400	0	25	+0.16	0	26	+1.14
4800	0	12	+0.16	0	13	–2.48
9600	—	—	—	0	6	–2.48
19200	—	—	—	—	—	—
31250	0	1	0	—	—	—
38400	—	—	—	—	—	—

Note: If possible, the error should be within 1%.

In the shaded section, if $\phi_P = \phi/2$, the bit rate is cut in half. In this case, BRR settings for the desired bit rate should be referenced from the column of one-half the actual system clock frequency (ϕ).

Table 12-3 Examples of BRR Settings in Asynchronous Mode (When $\phi_P = \phi$) (cont)

	ϕ Frequency (MHz)											
	2.4576			3			3.6864			4		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	174	–0.26	2	52	+0.50	2	64	+0.70	2	70	+0.03
150	1	127	0	1	155	+0.16	1	191	0	1	207	+0.16
300	0	255	0	1	77	+0.16	1	95	0	1	103	+0.16
600	0	127	0	0	155	+0.16	0	191	0	0	207	+0.16
1200	0	63	0	0	77	+0.16	0	95	0	0	103	+0.16
2400	0	31	0	0	38	+0.16	0	47	0	0	51	+0.16
4800	0	15	0	0	19	–2.34	0	23	0	0	25	+0.16
9600	0	7	0	0	9	–2.34	0	11	0	0	12	+0.16
19200	0	3	0	0	4	–2.34	0	5	0	—	—	—
31250	—	—	—	0	2	0	—	—	—	0	3	0
38400	0	1	0	—	—	—	0	2	0	—	—	—

Table 12-3 Examples of BRR Settings in Asynchronous Mode (When $\phi_P = \phi$) (cont)

	ϕ Frequency (MHz)											
	4.9152			5			6			6.144		
Bit Rate	Error		Error	Error		Error						
(bit/s)	n	N	(%)	n	N	(%)	n	N	(%)	n	N	(%)
110	2	86	+0.31	2	88	–0.25	2	106	–0.44	2	108	+0.08
150	1	255	0	2	64	+0.16	2	77	+0.16	2	79	0
300	1	127	0	1	129	+0.16	1	155	+0.16	1	159	0
600	0	255	0	1	64	+0.16	1	77	+0.16	1	79	0
1200	0	127	0	0	129	+0.16	0	155	+0.16	0	159	0
2400	0	63	0	0	64	+0.16	0	77	+0.16	0	79	0
4800	0	31	0	0	32	–1.36	0	38	+0.16	0	39	0
9600	0	15	0	0	15	+1.73	0	19	–2.34	0	19	0
19200	0	7	0	0	7	+1.73	0	9	–2.34	0	9	0
31250	0	4	–1.70	0	4	0	0	5	0	0	5	+2.40
38400	0	3	0	0	3	+1.73	0	4	–2.34	0	4	0

Note: If possible, the error should be within 1%.

In the shaded section, if $\phi_P = \phi/2$, the bit rate is cut in half. In this case, BRR settings for the desired bit rate should be referenced from the column of one-half the actual system clock frequency (ϕ).

Table 12-3 Examples of BRR Settings in Asynchronous Mode (When $\phi_P = \phi$) (cont)

	ϕ Frequency (MHz)											
	7.3728			8			9.8304			10		
Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	130	–0.07	2	141	+0.03	2	174	–0.26	2	177	–0.25
150	2	95	0	2	103	+0.16	2	127	0	2	129	+0.16
300	1	191	0	1	207	+0.16	1	255	0	2	64	+0.16
600	1	95	0	1	103	+0.16	1	127	0	1	129	+0.16
1200	0	191	0	0	207	+0.16	0	255	0	1	64	+0.16
2400	0	95	0	0	103	+0.16	0	127	0	0	129	+0.16
4800	0	47	0	0	51	+0.16	0	63	0	0	64	+0.16
9600	0	23	0	0	25	+0.16	0	31	0	0	32	–1.36
19200	0	11	0	0	12	+0.16	0	15	0	0	15	+1.73
31250	—	—	—	0	7	0	0	9	–1.70	0	9	0
38400	0	5	0	—	—	—	0	7	0	0	7	+1.73

$$B = F \times 10^6/[64 \times 2^{2n-1} \times (N + 1)] \rightarrow N = F \times 10^6/[64 \times 2^{2n-1} \times B] - 1$$

B: Bit rate (bits/second)
N: BRR value ($0 \leq N \leq 255$)
F: ϕ_P (MHz) when $n \neq 0$, or ϕ (MHz) when $n = 0$
n: Internal clock source (0, 1, 2, or 3)
The meaning of n is given by the table below:

n	CKS1	CKS0	Clock
0	0	0	ϕ
1	0	1	$\phi_P/4$
2	1	0	$\phi_P/16$
3	1	1	$\phi_P/64$

Bit rate error can be calculated with the formula below.

$$\text{Error (\%)} = \left\{[(F \times 10^6)/\{(N + 1) \times B \times 64 \times 2^{2n-1}\}] - 1\right\} \times 100$$

Table 12-4 Examples of BRR Settings in Synchronous Mode (When $\phi_P = \phi$)

	ϕ Frequency (MHz)									
Bit Rate	**2**		**4**		**5**		**8**		**10**	
(bit/s)	**n**	**N**	**n**	**N**	**n**	**N**	**n**	**N**	**n**	**N**
100	—	—	—	—	—	—	—	—	—	—
250	2	124	2	249	—	—	3	124	—	—
500	1	249	2	124	—	—	2	249	—	—
1 k	1	124	1	249	—	—	2	124	—	—
2.5 k	0	199	1	99	1	124	1	199	1	249
5 k	0	99	0	199	0	249	1	99	1	124
10 k	0	49	0	99	0	124	0	199	0	249
25 k	0	19	0	39	0	49	0	79	0	99
50 k	0	9	0	19	0	24	0	39	0	49
100 k	0	4	0	9	—	—	0	19	0	24
250 k	0	1	0	3	0	4	0	7	0	9
500 k	0	0*	0	1	—	—	0	3	0	4
1 M			0	0*	—	—	0	1	—	—
2.5 M									0	0*
4 M										

Notes: In the shaded section, if $\phi_P = \phi/2$, the bit rate is cut in half. In this case, BRR settings for the desired bit rate should be referenced from the column of one-half the actual system clock frequency (ϕ).

Blank: No setting is available.

—: A setting is available, but the bit rate is inaccurate.

*: Continuous transfer is not possible.

$B = F \times 10^6/[8 \times 2^{2n-1} \times (N + 1)] \rightarrow N = F \times 10^6/[8 \times 2^{2n-1} \times B] - 1$

B: Bit rate (bits per second)

N: BRR value ($0 \leq N \leq 255$)

F: ϕ_P (MHz) when $n \neq 0$, or ϕ (MHz) when $n = 0$

n: Internal clock source (0, 1, 2, or 3)

The meaning of n is given by the table below:

n	CKS1	CKS0	Clock
0	0	0	ϕ
1	0	1	$\phi_P/4$
2	1	0	$\phi_P/16$
3	1	1	$\phi_P/64$

12.2.9 Serial/Timer Control Register (STCR)

Bit	7	6	5	4	3	2	1	0
[H8/3534]	(IICS)	(IICD)	(IICX)	(IICE)	(STAC)	MPE	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

Bit	7	6	5	4	3	2	1	0
[H8/3522]	—	—	—	—	—	MPE	ICKS1	ICKS0
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

STCR is an 8-bit readable/writable register that controls the SCI operating mode and selects the TCNT clock source in the 8-bit timers. STCR is initialized to H'00 [H8/3534]/H'F8 [H8/3522] by a reset.

Bits 7 to 4—I^{2}C Control (IICS, IICD, IICX, IICE) [H8/3534]: These bits are reserved. They should not be set to 1.

Bit 3—Slave Input Switch (STAC) [H8/3534]: This bit is reserved. It should not be set to 1.

Bits 7 to 3—Reserved [H8/3522]: These bits cannot be modified and are always read as 1.

Bit 2—Multiprocessor Enable (MPE): Enables or disables the multiprocessor communication function on channels SCI0 and SCI1.

Bit 2 MPE	Description
0	The multiprocessor communication function is disabled, regardless of the setting of the MP bit in SMR. (Initial value)
1	The multiprocessor communication function is enabled. The multiprocessor format can be selected by setting the MP bit in SMR to 1.

Bits 1 and 0—Internal Clock Source Select 1 and 0 (ICKS1, ICKS0): These bits select the clock input to the timer counters (TCNT) in the 8-bit timers. For details, see section 9, 8-Bit Timers.

12.3 Operation

12.3.1 Overview

The SCI supports serial data transfer in two modes. In asynchronous mode each character is synchronized individually. In synchronous mode communication is synchronized with a clock signal.

The selection of asynchronous or synchronous mode and the communication format depend on SMR settings as indicated in table 12-5. The clock source depends on the settings of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR as indicated in table 12-6.

Asynchronous Mode

- Data length: 7 or 8 bits can be selected.
- A parity bit or multiprocessor bit can be added, and stop bit lengths of 1 or 2 bits can be selected. (These selections determine the communication format and character length.)
- Framing errors (FER), parity errors (PER), and overrun errors (ORER) can be detected in receive data, and the line-break condition can be detected.
- SCI clock source: An internal or external clock source can be selected.

  Internal clock: The SCI is clocked by the on-chip baud rate generator and can output a clock signal at the bit-rate frequency.

  External clock: The external clock frequency must be 16 times the bit rate. (The on-chip baud rate generator is not used.)

Synchronous Mode

- Communication format: The data length is 8 bits.
- Overrun errors (ORER) can be detected in receive data.
- SCI clock source: An internal or external clock source can be selected.

  Internal clock: The SCI is clocked by the on-chip baud rate generator and outputs a serial clock signal to external devices.

  External clock: The on-chip baud rate generator is not used. The SCI operates on the input serial clock.

Table 12-5 Communication Formats Used by SCI

SMR Settings						Communication Format			
Bit 7 C/$\overline{A}$	**Bit 6 CHR**	**Bit 2 MP**	**Bit 5 PE**	**Bit 3 STOP**	**Mode**	**Data Length**	**Multi-processor Bit**	**Parity Bit**	**Stop-Bit Length**
0	0	0	0	0	Asynchronous mode	8 bits	None	None	1 bit
				1					2 bits
			1	0				Present	1 bit
				1					2 bits
	1		0	0		7 bits		None	1 bit
				1					2 bits
			1	0				Present	1 bit
				1					2 bits
	0	1	—	0	Asynchronous mode (multi-processor format)	8 bits	Present	None	1 bit
				1					2 bits
	1			0		7 bits			1 bit
				1					2 bits
1	—	—	—	—	Synchronous mode	8 bits	None		None

Table 12-6 SCI Clock Source Selection

SMR	SCR			Serial Transmit/Receive Clock	
Bit 7	**Bit 1**	**Bit 0**			
C/$\overline{A}$	**CKE1**	**CKE0**	**Mode**	**Clock Source**	**SCK Pin Function**
0	0	0	Async	Internal	Input/output port (not used by SCI)
		1			Serial clock output at bit rate
	1	0		External	Serial clock input at 16 × bit rate
		1			
1	0	0	Sync	Internal	Serial clock output
		1			
	1	0		External	Serial clock input
		1			

12.3.2 Asynchronous Mode

In asynchronous mode, each transmitted or received character is individually synchronized by framing it with a start bit and stop bit.

Full duplex data transfer is possible because the SCI has independent transmit and receive sections. Double buffering in both sections enables the SCI to be programmed for continuous data transfer.

Figure 12-2 shows the general format of one character sent or received in asynchronous mode. The communication channel is normally held in the mark state (high). Character transmission or reception starts with a transition to the space state (low).

The first bit transmitted or received is the start bit (low). It is followed by the data bits, in which the least significant bit (LSB) comes first. The data bits are followed by the parity or multiprocessor bit, if present, then the stop bit or bits (high) confirming the end of the frame.

In receiving, the SCI synchronizes on the falling edge of the start bit, and samples each bit at the center of the bit (at the 8th cycle of the internal serial clock, which runs at 16 times the bit rate).

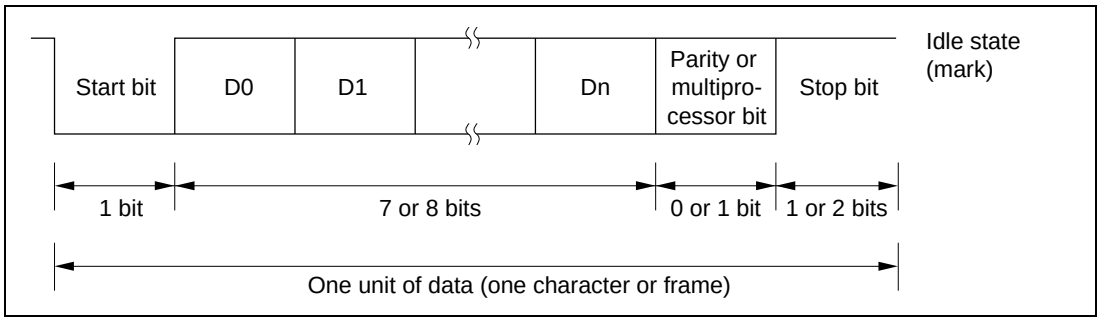


Figure 12-2 Data Format in Asynchronous Mode (Example of 8-Bit Data with Parity Bit and Two Stop Bits)

1. **Data Format:** Table 12-7 lists the data formats that can be sent and received in asynchronous mode. Twelve formats can be selected by bits in the serial mode register (SMR).

Table 12-7 Data Formats in Asynchronous Mode

	SMR Bits													
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	

Notes: SMR: Serial mode register
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

2. **Clock:** In asynchronous mode it is possible to select either an internal clock created by the on-chip baud rate generator, or an external clock input at the SCK pin. The selection is made by the C/$\overline{A}$ bit in the serial mode register (SMR) and the CKE1 and CKE0 bits in the serial control register (SCR). Refer to table 12-6.

   If an external clock is input at the SCK pin, its frequency should be 16 times the desired bit rate.

   If the internal clock provided by the on-chip baud rate generator is selected and the SCK pin is used for clock output, the output clock frequency is equal to the bit rate, and the clock pulse rises at the center of the transmit data bits. Figure 12-3 shows the phase relationship between the output clock and transmit data.

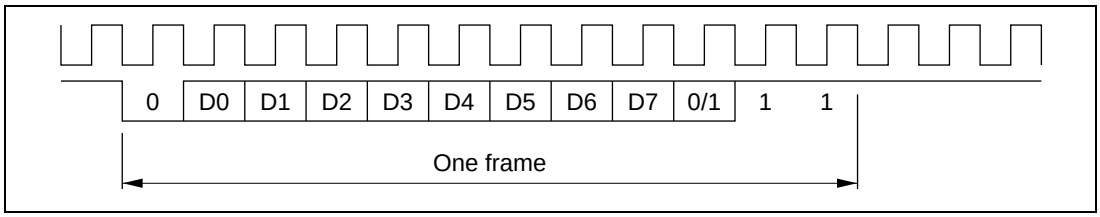


Figure 12-3 Phase Relationship between Clock Output and Transmit Data (Asynchronous Mode)

3. Transmitting and Receiving Data

— **SCI Initialization:** Before transmitting or receiving, software must clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI following the procedure in figure 12-4.

Note: When changing the communication mode or format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.

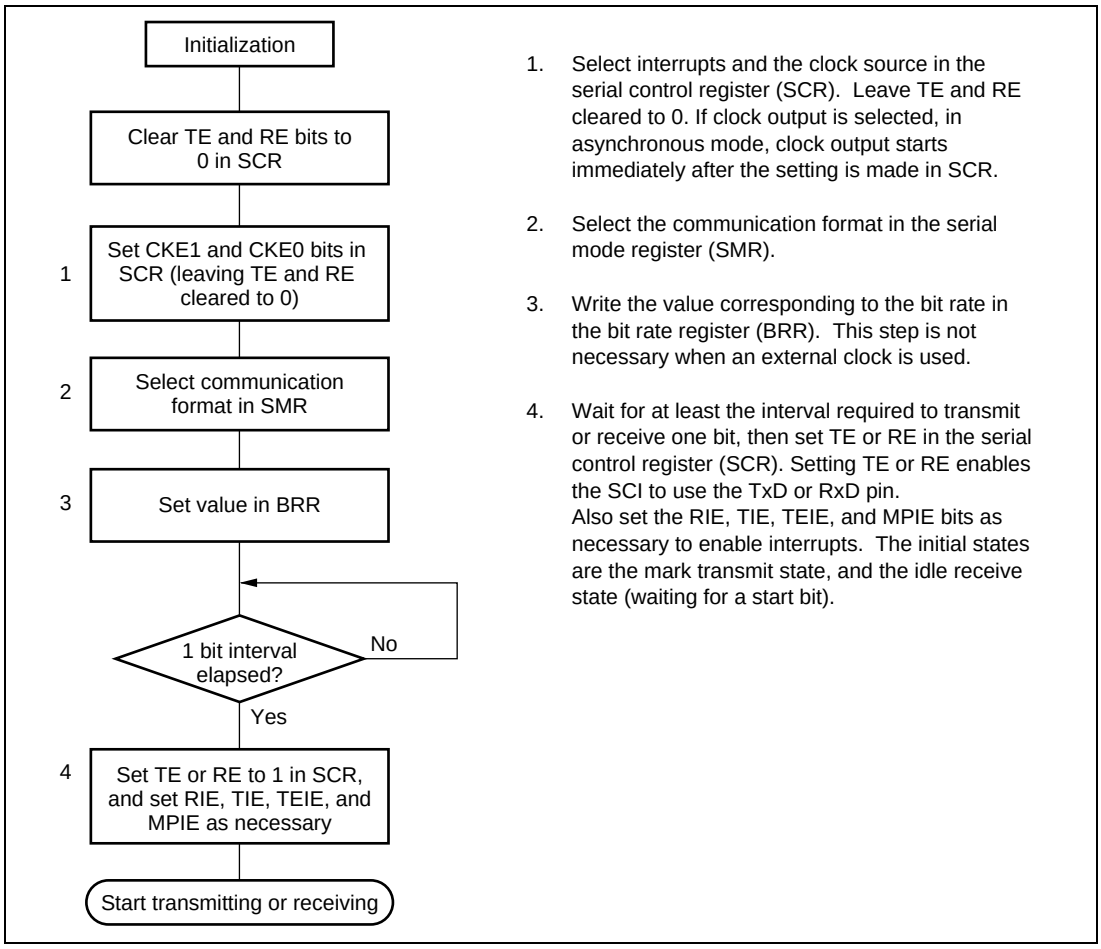
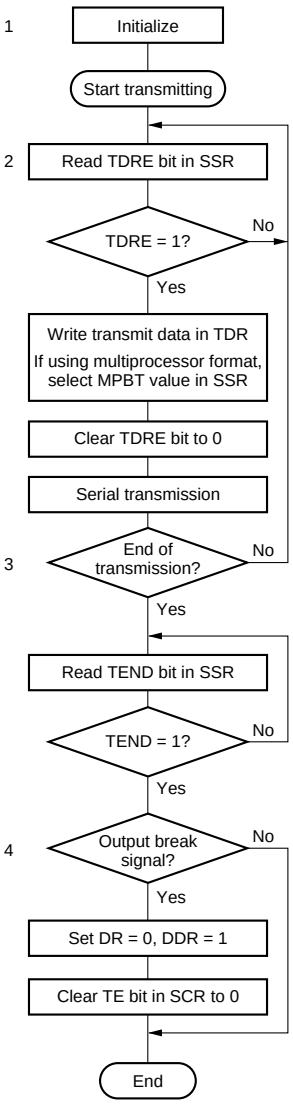


Figure 12-4 Sample Flowchart for SCI Initialization

— **Transmitting Serial Data:** Follow the procedure in figure 12-5 for transmitting serial data.



1. SCI initialization: the transmit data output function of the TxD pin is selected automatically.

2. SCI status check and transmit data write: read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0. If a multiprocessor format is selected, after writing the transmit data write 0 or 1 in the multiprocessor bit transfer (MPBT) in SSR. Transition of the TDRE bit from 0 to 1 can be reported by an interrupt.

3. (a) To continue transmitting serial data: read the TDRE bit to check whether it is safe to write; if TDRE = 1, write data in TDR, then clear TDRE to 0.
   (b) To end serial transmission: end of transmission can be confirmed by checking transition of the TEND bit from 0 to 1. This can be reported by a TEI interrupt.

4. To output a break signal at the end of serial transmission: set the DDR bit to 1 and clear the DR bit to 0 (DDR and DR are I/O port registers), then clear TE to 0 in SCR.

Figure 12-5 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows.

1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).
2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the TIE bit (TDR-empty interrupt enable) is set to 1 in SCR, the SCI requests a TXI interrupt (TDR-empty interrupt) at this time.

   Serial transmit data are transmitted in the following order from the TxD pin:

   (a) Start bit: One 0 bit is output.

   (b) Transmit data: Seven or eight bits are output, LSB first.

   (c) Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.

   (d) Stop bit: One or two 1 bits (stop bits) are output.

   (e) Mark state: Output of 1 bits continues until the start bit of the next transmit data.
3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, after loading new data from TDR into TSR and transmitting the stop bit, the SCI begins serial transmission of the next frame. If TDRE is 1, after setting the TEND bit to 1 in SSR and transmitting the stop bit, the SCI continues 1-level output in the mark state, and if the TEIE bit (TSR-empty interrupt enable) in SCR is set to 1, the SCI generates a TEI interrupt request (TSR-empty interrupt).

Figure 12-6 shows an example of SCI transmit operation in asynchronous mode.

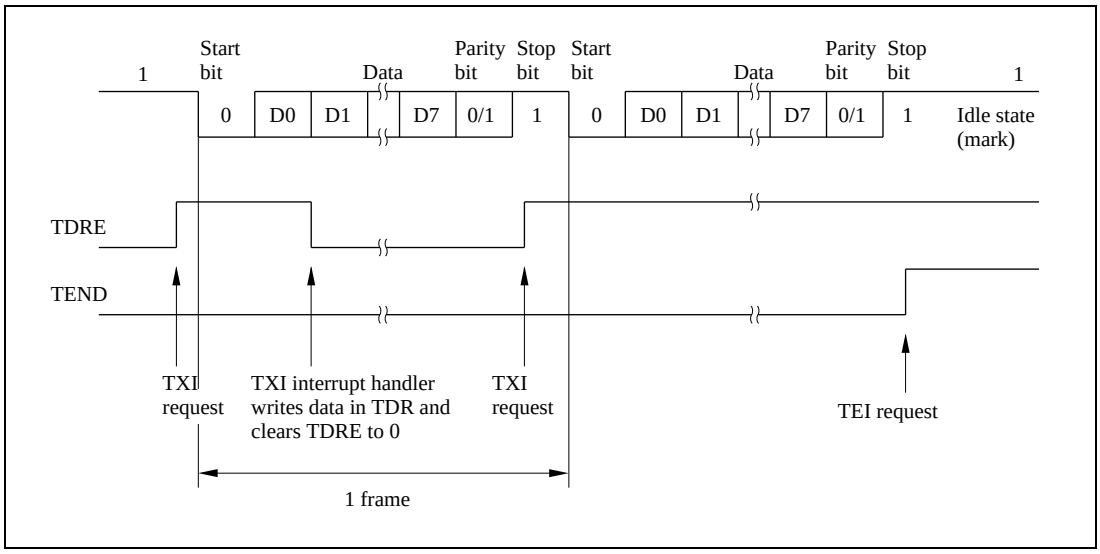


Figure 12-6 Example of SCI Transmit Operation (8-Bit Data with Parity and One Stop Bit)

— **Receiving Serial Data:** Follow the procedure in figure 12-7 for receiving serial data.

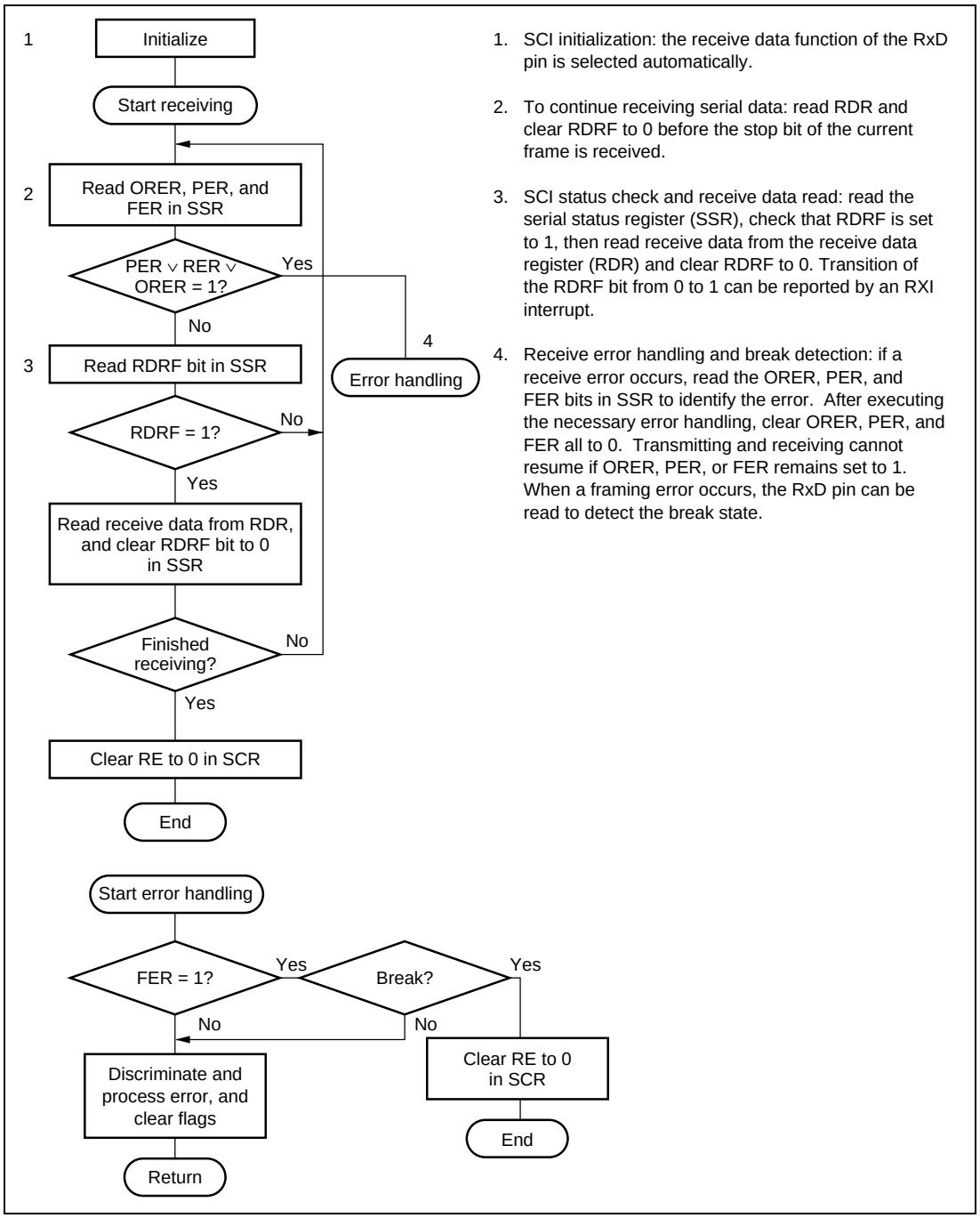


Figure 12-7 Sample Flowchart for Receiving Serial Data

In receiving, the SCI operates as follows.

1. The SCI monitors the receive data line and synchronizes internally when it detects a start bit.
2. Receive data is shifted into RSR in order from LSB to MSB.
3. The parity bit and stop bit are received.

   After receiving these bits, the SCI makes the following checks:

   (a) Parity check: The number of 1s in the receive data must match the even or odd parity setting of the O/$\overline{E}$ bit in SMR.

   (b) Stop bit check: The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.

   (c) Status check: RDRF must be 0 so that receive data can be loaded from RSR into RDR.

   If these checks all pass, the SCI sets RDRF to 1 and stores the received data in RDR. If one of the checks fails (receive error), the SCI operates as indicated in table 12-8.

   Note: When a receive error flag is set, further receiving is disabled. The RDRF bit is not set to 1. Be sure to clear the error flags.
4. After setting RDRF to 1, if the RIE bit (receive-end interrupt enable) is set to 1 in SCR, the SCI requests an RXI (receive-end) interrupt. If one of the error flags (ORER, PER, or FER) is set to 1 and the RIE bit in SCR is also set to 1, the SCI requests an ERI (receive-error) interrupt.

   Figure 12-8 shows an example of SCI receive operation in asynchronous mode.

Table 12-8 Receive Error Conditions and SCI Operation

Receive error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SSR	Receive data not loaded from RSR into RDR
Framing error	FER	Stop bit is 0	Receive data loaded from RSR into RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data loaded from RSR into RDR

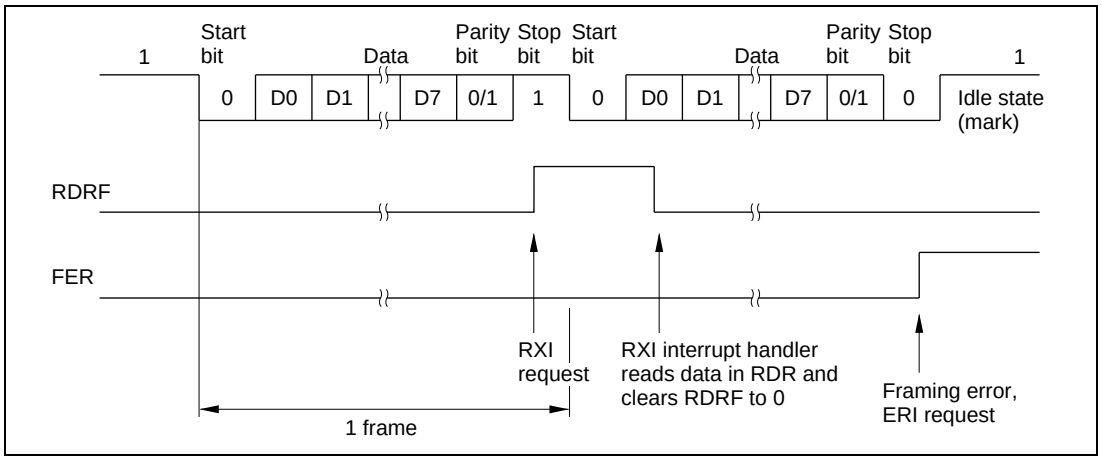


Figure 12-8 Example of SCI Receive Operation (8-Bit Data with Parity and One Stop Bit)

4. Multiprocessor Communication

The multiprocessor communication function enables several processors to share a single serial communication line. The processors communicate in asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by an ID.

A serial communication cycle consists of two cycles: an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles.

The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1.

After receiving data with the multiprocessor bit set to 1, the receiving processor with an ID matching the received data continues to receive further incoming data. Multiple processors can send and receive data in this way.

Four formats are available. Parity-bit settings are ignored when a multiprocessor format is selected. For details see table 12-7.

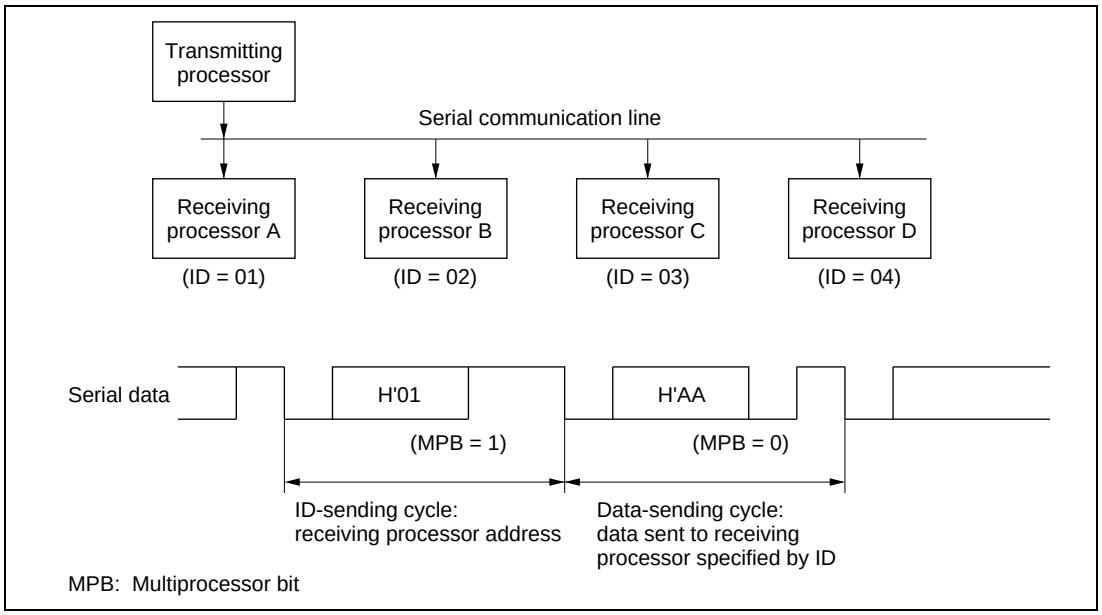


Figure 12-9 Example of Communication among Processors using Multiprocessor Format (Sending Data H'AA to Receiving Processor A)

— **Transmitting Multiprocessor Serial Data:** See figures 12-5 and 12-6.

— **Receiving Multiprocessor Serial Data:** Follow the procedure in figure 12-10 for receiving multiprocessor serial data.

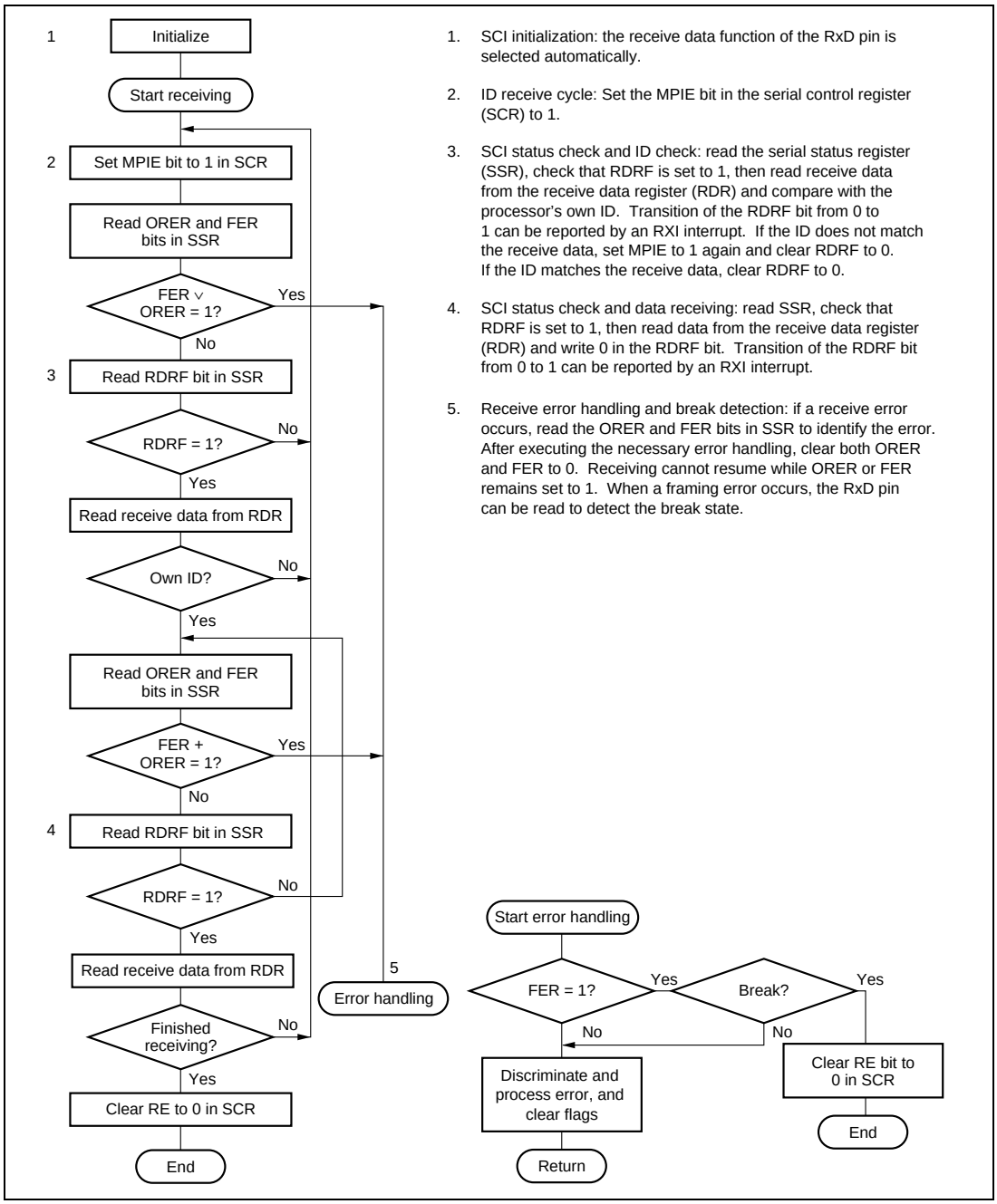


Figure 12-10 Sample Flowchart for Receiving Multiprocessor Serial Data

Figure 12-11 shows an example of an SCI receive operation using a multiprocessor format (8-bit data with multiprocessor bit and one stop bit).

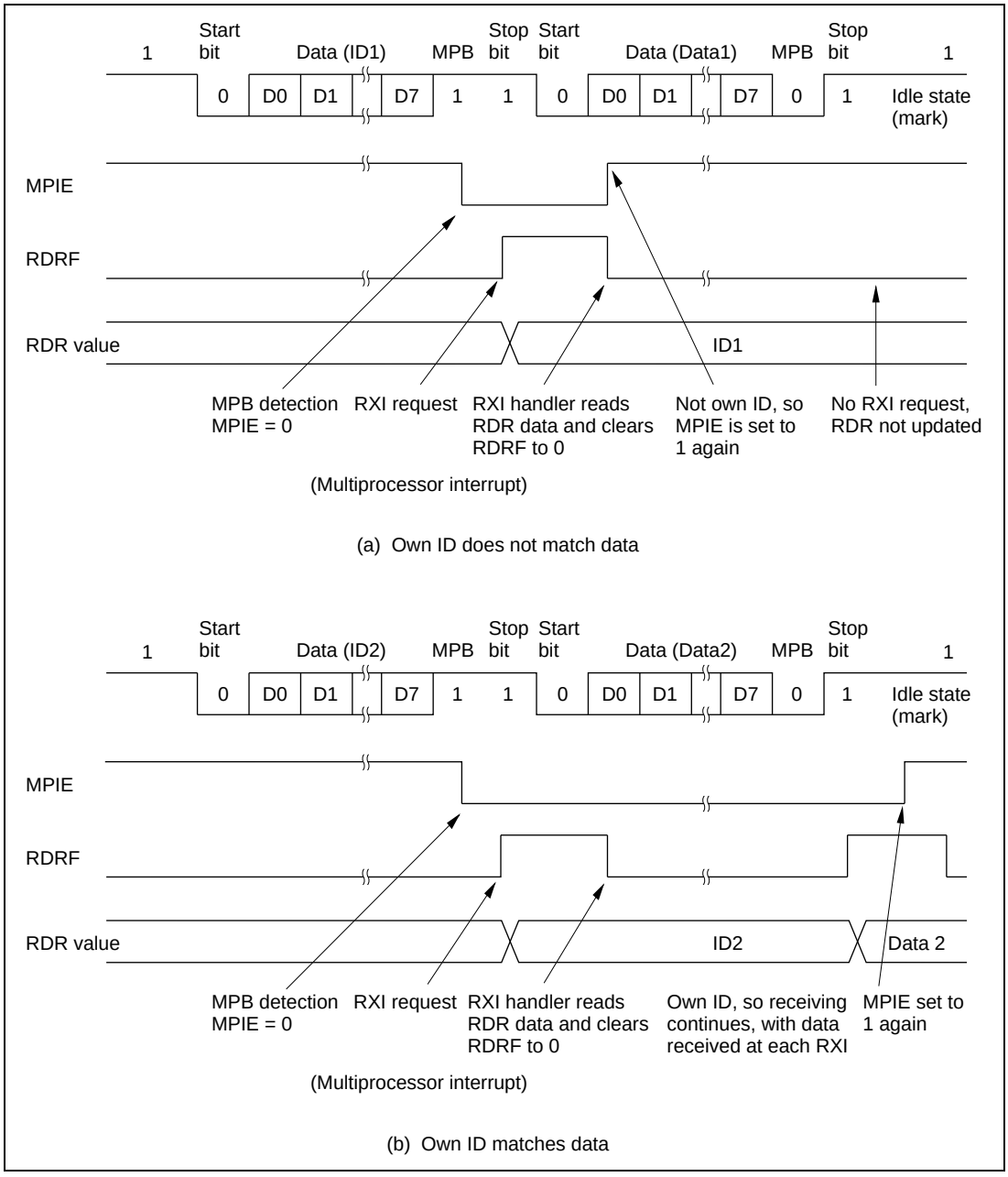


Figure 12-11 Example of SCI Receive Operation (8-Bit Data with Multiprocessor Bit and One Stop Bit)

12.3.3 Synchronous Mode

(1) Overview: In synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver share the same clock but are otherwise independent, so full duplex communication is possible. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 12-12 shows the general format in synchronous serial communication.

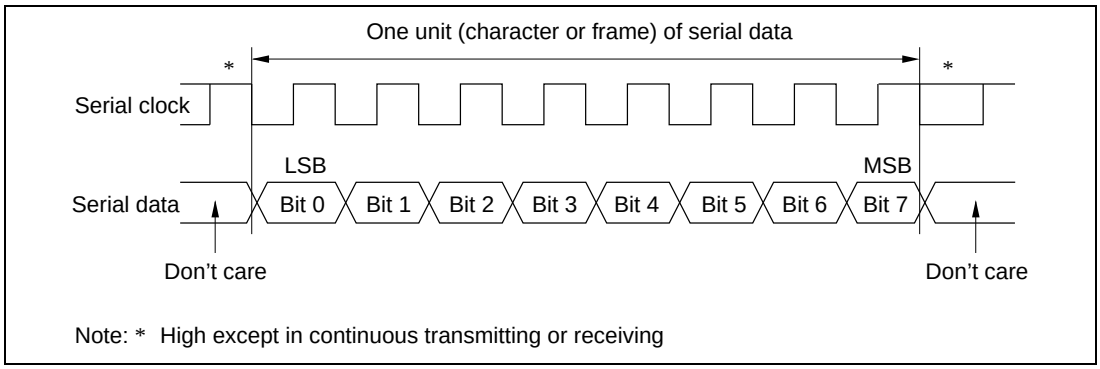


Figure 12-12 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is sent on the communication line from one falling edge of the serial clock to the next. Data is received in synchronization with the rising edge of the serial clock.

In each character, the serial data bits are transmitted in order from LSB (first) to MSB (last). After output of the MSB, the communication line remains in the state of the MSB.

— **Communication Format:** The data length is fixed at eight bits. No parity bit or multiprocessor bit can be added.

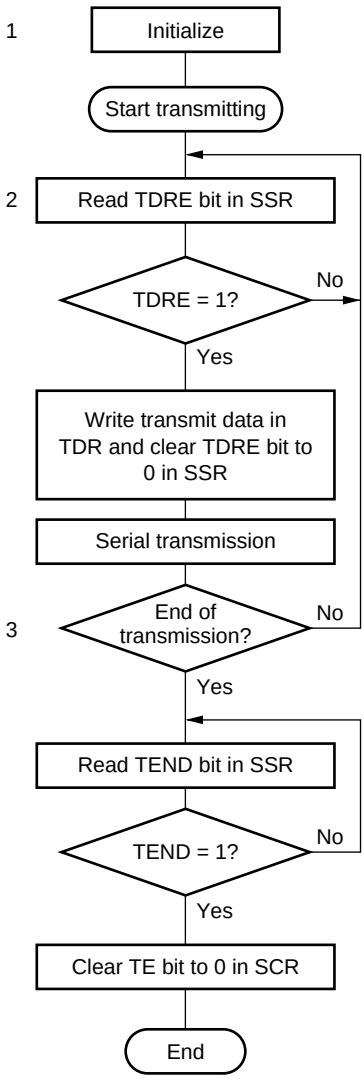
— **Clock:** An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected by clearing or setting the C/$\overline{A}$ bit in the serial mode register (SMR) and the CKE1 and CKE0 bits in the serial control register (SCR). See table 12-6.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains at the high level.

(2) Transmitting and Receiving Data

— **SCI Initialization:** The SCI must be initialized in the same way as in asynchronous mode. See figure 12-4. When switching from asynchronous mode to synchronous mode, check that the ORER, FER, and PER bits are cleared to 0. Transmitting and receiving cannot begin if ORER, FER, or PER is set to 1.

— **Transmitting Serial Data:** Follow the procedure in figure 12-13 for transmitting serial data.



1. SCI initialization: the transmit data output function of the TxD pin is selected automatically.

2. SCI status check and transmit data write: read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0. Transition of the TDRE bit from 0 to 1 can be reported by a TXI interrupt.

3. (a) To continue transmitting serial data: read the TDRE bit to check whether it is safe to write; if TDRE = 1, write data in TDR, then clear TDRE to 0.
   (b) To end serial transmission: end of transmission can be confirmed by checking transition of the TEND bit from 0 to 1. This can be reported by a TEI interrupt.

Figure 12-13 Sample Flowchart for Serial Transmitting

In transmitting serial data, the SCI operates as follows.

1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).
2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the TIE bit (TDR-empty interrupt enable) in SCR is set to 1, the SCI requests a TXI interrupt (TDR-empty interrupt) at this time.

   If clock output is selected the SCI outputs eight serial clock pulses, triggered by the clearing of the TDRE bit to 0. If an external clock source is selected, the SCI outputs data in synchronization with the input clock.

   Data is output from the TxD pin in order from LSB (bit 0) to MSB (bit 7).
3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the SCI loads data from TDR into TSR, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in SSR to 1, transmits the MSB, then holds the output in the MSB state. If the TEIE bit (transmit-end interrupt enable) in SCR is set to 1, a TEI interrupt (TSR-empty interrupt) is requested at this time.
4. After the end of serial transmission, the SCK pin is held at the high level.

Figure 12-14 shows an example of SCI transmit operation.

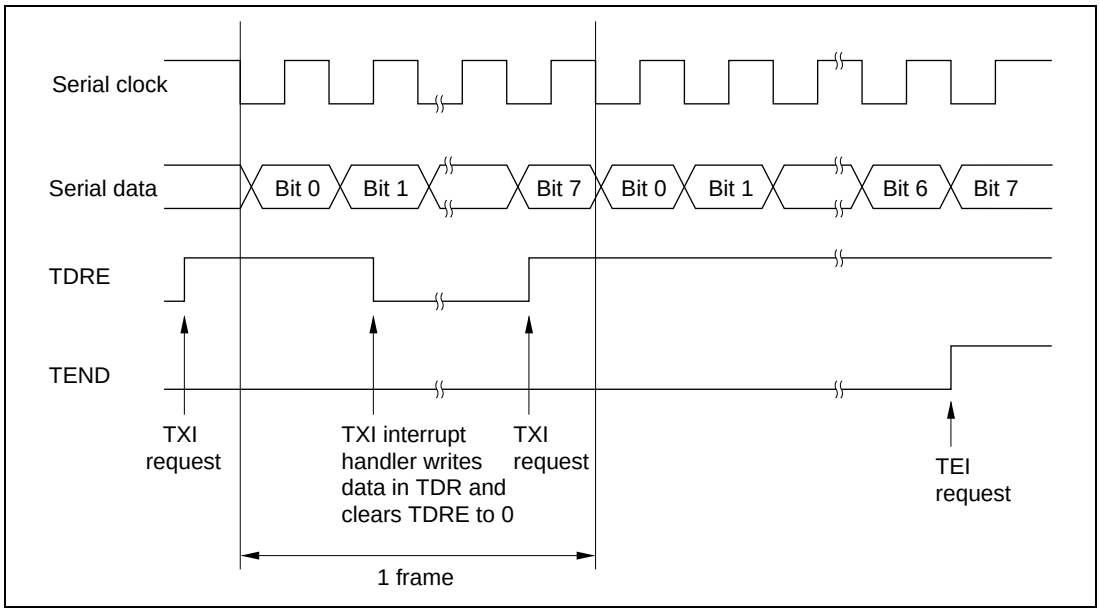


Figure 12-14 Example of SCI Transmit Operation

— **Receiving Serial Data:** Follow the procedure in figure 12-15 for receiving serial data. When switching from asynchronous mode to synchronous mode, be sure to check that PER and FER are cleared to 0. If PER or FER is set to 1 the RDRF bit will not be set and both transmitting and receiving will be disabled.

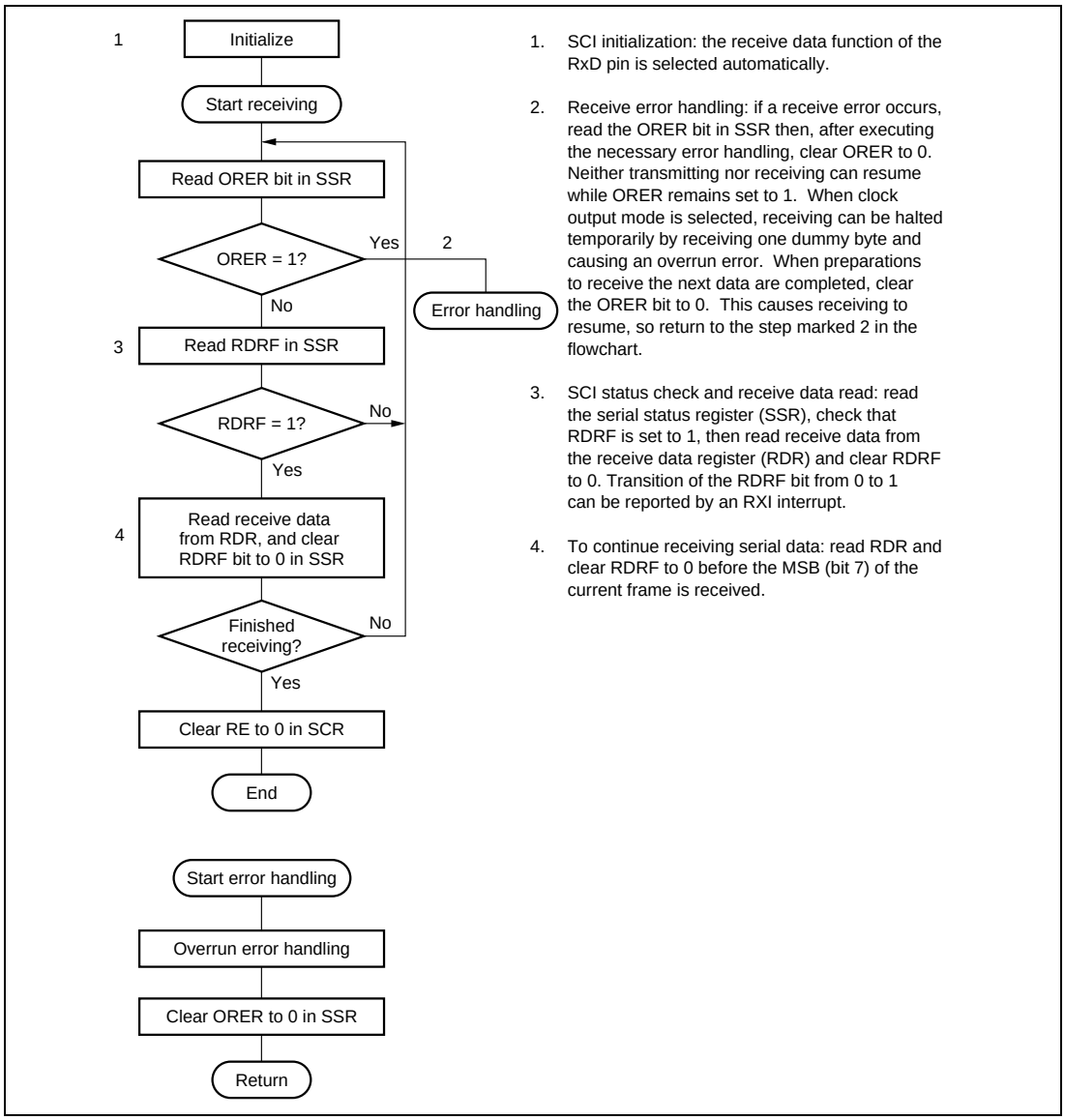


Figure 12-15 Sample Flowchart for Serial Receiving

In receiving, the SCI operates as follows.

1. If an external clock is selected, data is input in synchronization with the input clock. If clock output is selected, as soon as the RE bit is set to 1 the SCI begins outputting the serial clock and inputting data. If clock output is stopped because the ORER bit is set to 1, output of the serial clock and input of data resume as soon as the ORER bit is cleared to 0.
2. Receive data is shifted into RSR in order from LSB to MSB.

   After receiving the data, the SCI checks that RDRF is 0 so that receive data can be loaded from RSR into RDR. If this check passes, the SCI sets RDRF to 1 and stores the received data in RDR. If the check does not pass (receive error), the SCI operates as indicated in
   table 12-8.

   Note: Both transmitting and receiving are disabled while a receive error flag is set. The RDRF bit is not set to 1. Be sure to clear the error flag.
3. After setting RDRF to 1, if the RIE bit (receive-end interrupt enable) is set to 1 in SCR, the SCI requests an RXI (receive-end) interrupt. If the ORER bit is set to 1 and the RIE bit in SCR is set to 1, the SCI requests an ERI (receive-error) interrupt.

   When clock output mode is selected, clock output stops when the RE bit is cleared to 0 or the ORER bit is set to 1. To prevent clock count errors, it is safest to receive one dummy byte and generate an overrun error.

   Figure 12-16 shows an example of SCI receive operation.

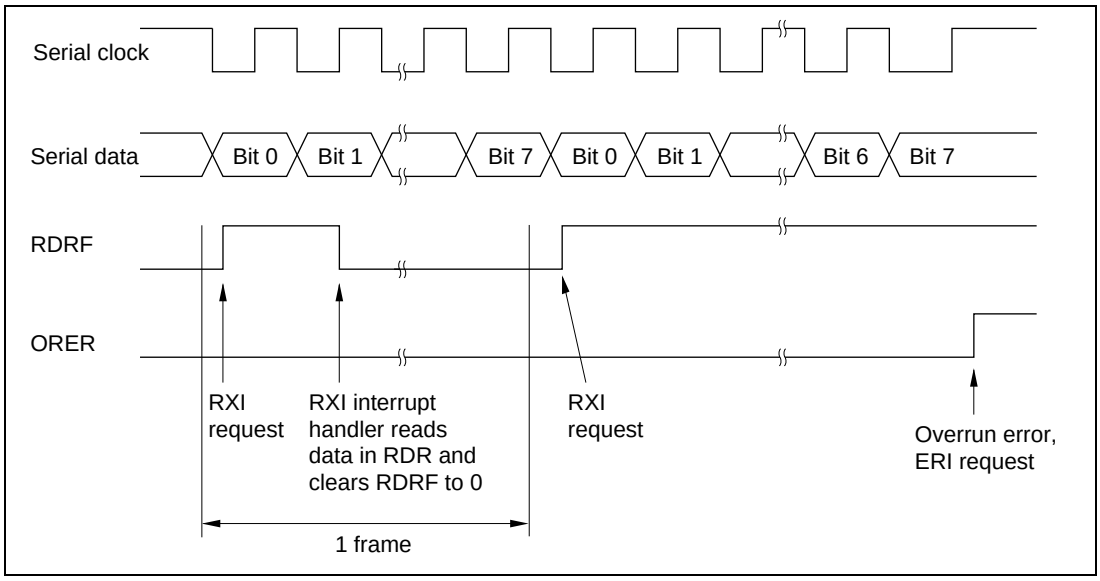


Figure 12-16 Example of SCI Receive Operation

— **Transmitting and Receiving Serial Data Simultaneously:** Follow the procedure in figure
12-17 for transmitting and receiving serial data simultaneously. If clock output mode is selected, output of the serial clock begins simultaneously with serial transmission.

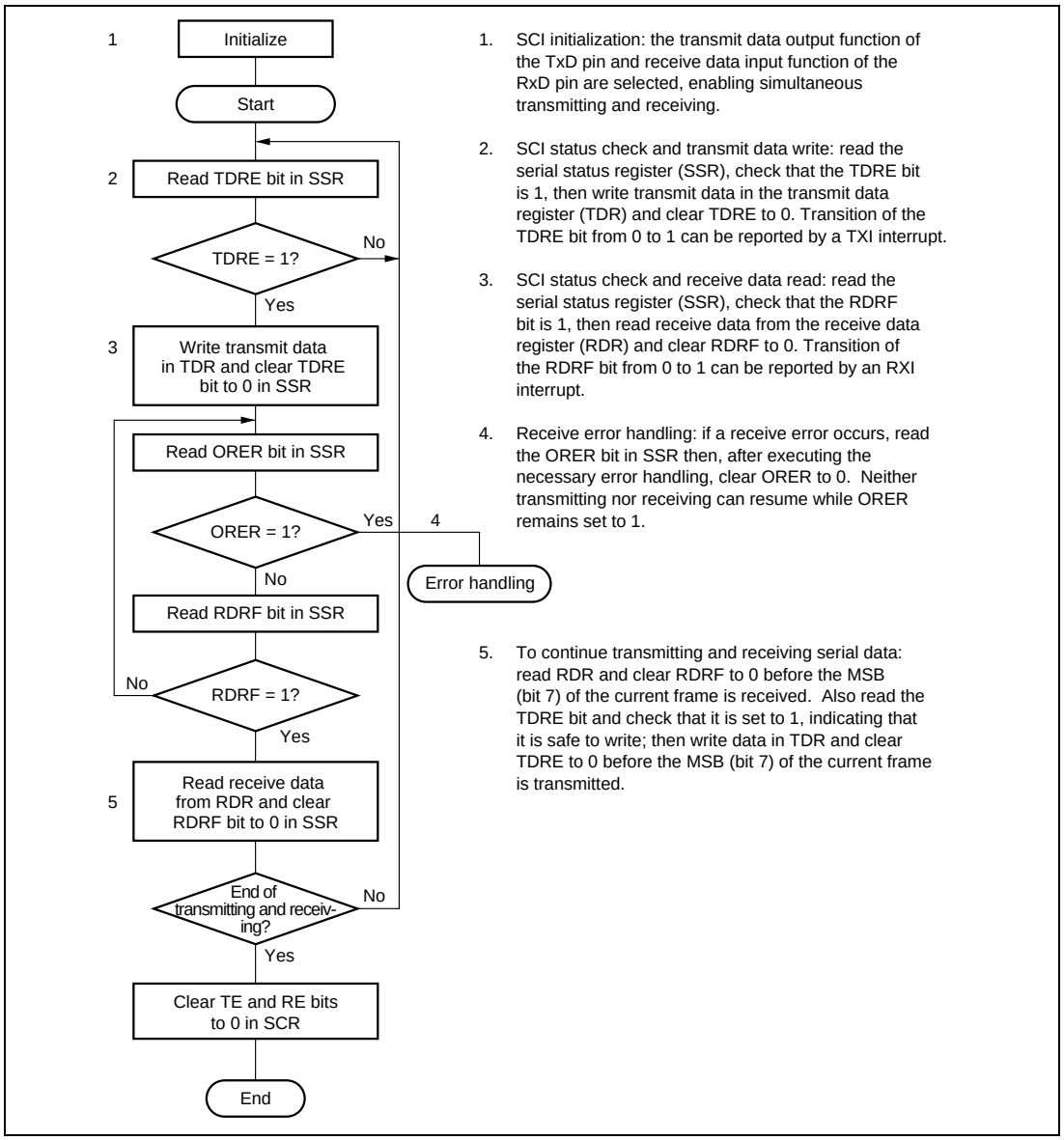


Figure 12-17 Sample Flowchart for Serial Transmitting and Receiving

Note: In switching from transmitting or receiving to simultaneous transmitting and receiving, clear both TE and RE to 0, then set both TE and RE to 1.

12.4 Interrupts

The SCI can request four types of interrupts: ERI, RXI, TXI, and TEI. Table 12-9 indicates the source and priority of these interrupts. The interrupt sources can be enabled or disabled by the TIE, RIE, and TEIE bits in the SCR. Independent signals are sent to the interrupt controller for each interrupt source, except that the receive-error interrupt (ERI) is the logical OR of three sources: overrun error, framing error, and parity error.

The TXI interrupt indicates that the next transmit data can be written. The TEI interrupt indicates that the SCI has stopped transmitting data.

Table 12-9 SCI Interrupt Sources

Interrupt	Description	Priority
ERI	Receive-error interrupt (ORER, FER, or PER)	High
RXI	Receive-end interrupt (RDRF)	
TXI	TDR-empty interrupt (TDRE)	
TEI	TSR-empty interrupt (TEND)	Low

12.5 Application Notes

Application programmers should note the following features of the SCI.

(1) TDR Write: The TDRE bit in SSR is simply a flag that indicates that the TDR contents have been transferred to TSR. The TDR contents can be rewritten regardless of the TDRE value. If a new byte is written in TDR while the TDRE bit is 0, before the old TDR contents have been moved into TSR, the old byte will be lost. Software should check that the TDRE bit is set to 1 before writing to TDR.

(2) Multiple Receive Errors: Table 12-10 lists the values of flag bits in the SSR when multiple receive errors occur, and indicates whether the RSR contents are transferred to RDR.

Table 12-10 SSR Bit States and Data Transfer when Multiple Receive Errors Occur

	SSR Bits				RSR →
Receive error	**RDRF**	**ORER**	**FER**	**PER**	**RDR[*2]**
Overrun error	1[*1]	1	0	0	No
Framing error	0	0	1	0	Yes
Parity error	0	0	0	1	Yes
Overrun and framing errors	1[*1]	1	1	0	No
Overrun and parity errors	1[*1]	1	0	1	No
Framing and parity errors	0	0	1	1	Yes
Overrun, framing, and parity errors	1[*1]	1	1	1	No

Notes: 1. Set to 1 before the overrun error occurs.
2. Yes: The RSR contents are transferred to RDR.
No: The RSR contents are not transferred to RDR.

(3) Line Break Detection: When the RxD pin receives a continuous stream of 0's in asynchronous mode (line-break state), a framing error occurs because the SCI detects a 0 stop bit. The value H'00 is transferred from RSR to RDR. Software can detect the line-break state as a framing error accompanied by H'00 data in RDR.

The SCI continues to receive data, so if the FER bit is cleared to 0 another framing error will occur.

(4) Sampling Timing and Receive Margin in Asynchronous Mode: The serial clock used by the SCI in asynchronous mode runs at 16 times the bit rate. The falling edge of the start bit is detected by sampling the RxD input on the falling edge of this clock. After the start bit is detected, each bit of receive data in the frame (including the start bit, parity bit, and stop bit or bits) is sampled on the rising edge of the serial clock pulse at the center of the bit. See figure 12-18.

It follows that the receive margin can be calculated as in equation (1).

When the absolute frequency deviation of the clock signal is 0 and the clock duty cycle is 0.5, data can theoretically be received with distortion up to the margin given by equation (2). This is a theoretical limit, however. In practice, system designers should allow a margin of 20% to 30%.

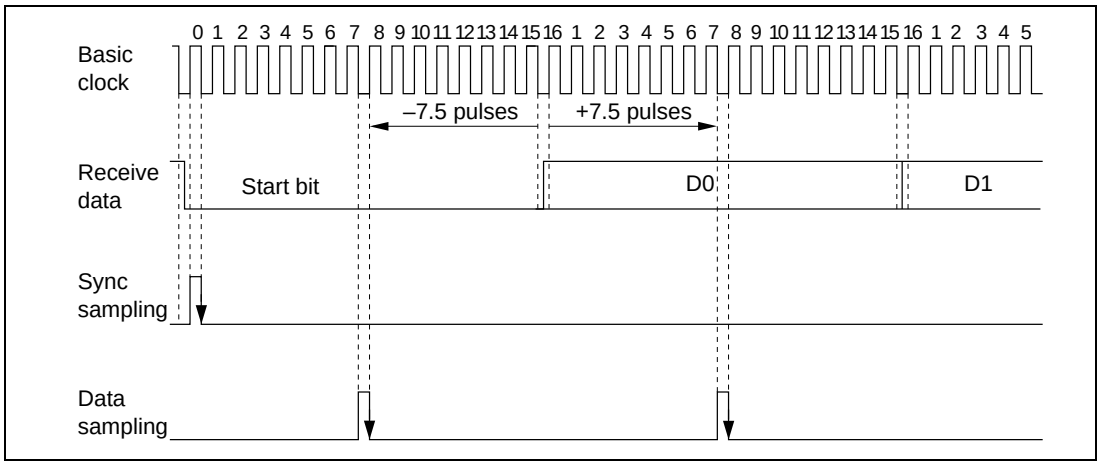


Figure 12-18 Sampling Timing (Asynchronous Mode)

$$M = \{ (0.5 - 1/2N) - (D - 0.5)/N - (L - 0.5)F\} \times 100\ [\%] \qquad (1)$$

M: Receive margin

N: Ratio of basic clock to bit rate (N=16)

D: Duty factor of clock—ratio of high pulse width to low width (0.5 to 1.0)

L: Frame length (9 to 12)

F: Absolute clock frequency deviation

When D = 0.5 and F = 0

$$M = (0.5 - 1/2 \times 16) \times 100\ [\%] = 46.875\% \qquad (2)$$

Section 13 A/D Converter

13.1 Overview

The H8/3534 and H8/3522 include a 10-bit successive-approximations A/D converter with a selection of up to eight analog input channels.

13.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Eight input channels
- High-speed conversion
  Conversion time: minimum 13.4 µs per channel (with 10-MHz system clock)
- Two conversion modes
  Single mode: A/D conversion of one channel
  Scan mode: continuous conversion on one to four channels
- Four 16-bit data registers
  A/D conversion results are transferred for storage into data registers corresponding to the channels.
- Sample-and-hold function
- A/D conversion can be externally triggered
- A/D interrupt requested at end of conversion
  At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.

13.1.2 Block Diagram

Figure 13-1 shows a block diagram of the A/D converter.

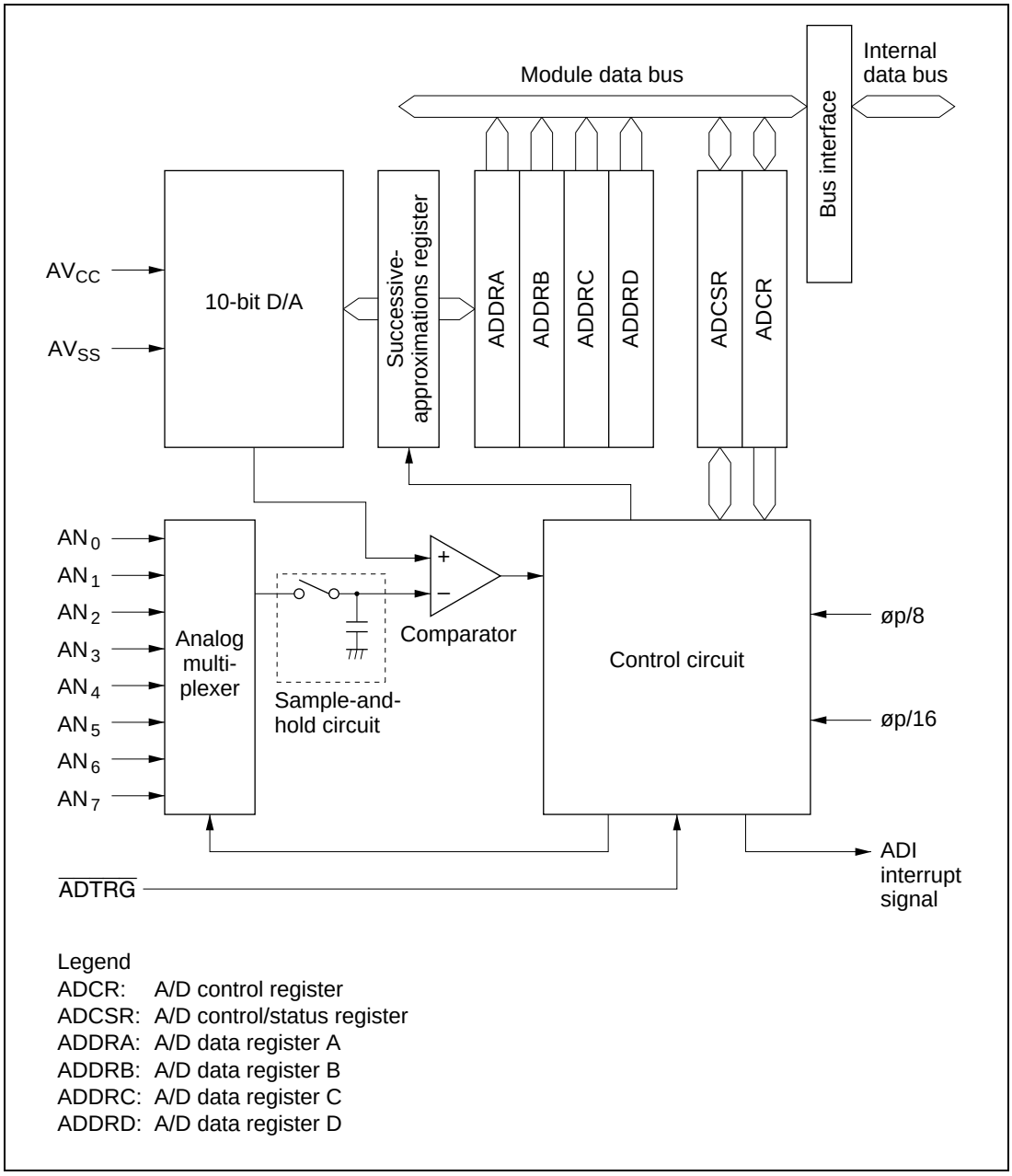


Figure 13-1 A/D Converter Block Diagram

13.1.3 Input Pins

Table 13-1 lists the A/D converter's input pins. The eight analog input pins are divided into two groups: group 0 (AN_0 to AN_3), and group 1 (AN_4 to AN_7). AV_{CC} and AV_{SS} are the power supply for the analog circuits in the A/D converter.

Table 13-1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog power supply
Analog ground pin	AV_{SS}	Input	Analog ground and reference voltage
Analog input pin 0	AN_0	Input	Group 0 analog inputs
Analog input pin 1	AN_1	Input	
Analog input pin 2	AN_2	Input	
Analog input pin 3	AN_3	Input	
Analog input pin 4	AN_4	Input	Group 1 analog inputs
Analog input pin 5	AN_5	Input	
Analog input pin 6	AN_6	Input	
Analog input pin 7	AN_7	Input	
A/D external trigger input pin	$\overline{ADTRG}$	Input	External trigger input for starting A/D conversion

13.1.4 Register Configuration

Table 13-2 summarizes the A/D converter's registers.

Table 13-2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address
A/D data register A (high)	ADDRAH	R	H'00	H'FFE0
A/D data register A (low)	ADDRAL	R	H'00	H'FFE1
A/D data register B (high)	ADDRBH	R	H'00	H'FFE2
A/D data register B (low)	ADDRBL	R	H'00	H'FFE3
A/D data register C (high)	ADDRCH	R	H'00	H'FFE4
A/D data register C (low)	ADDRCL	R	H'00	H'FFE5
A/D data register D (high)	ADDRDH	R	H'00	H'FFE6
A/D data register D (low)	ADDRDL	R	H'00	H'FFE7
A/D control/status register	ADCSR	R/W*	H'00	H'FFE8
A/D control register	ADCR	R/W	H'7F	H'FFE9

Note: * Only 0 can be written in bit 7, to clear the flag.

13.2 Register Descriptions

13.2.1 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
ADDRn	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
(n = A~D)																

The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The upper 8 bits of the result are stored in the upper byte of the A/D data register. The lower 2 bits are stored in the lower byte. Bits 5 to 0 of an A/D data register are reserved bits that always read 0. Table 13-3 indicates the pairings of analog input channels and A/D data registers.

The CPU can always read and write the A/D data registers. The upper byte can be read directly, but the lower byte is read through a temporary register (TEMP). For details see section 13.3, CPU Interface.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 13-3 Analog Input Channels and A/D Data Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
AN_0	AN_4	ADDRA
AN_1	AN_5	ADDRB
AN_2	AN_6	ADDRC
AN_3	AN_7	ADDRD

13.2.2 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, to clear the flag.

ADCSR is an 8-bit readable/writable register that selects the mode and controls the A/D converter. ADCSR is initialized to H'00 by a reset and in standby mode.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7 ADF	Description	
0	[Clearing condition] Cleared by reading ADF while ADF = 1, then writing 0 in ADF	(Initial value)
1	[Setting conditions] 1.Single mode: A/D conversion ends 2.Scan mode: A/D conversion ends in all selected channels	

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the interrupt (ADI) requested at the end of A/D conversion.

Bit 6 ADIE	Description	
0	A/D end interrupt request (ADI) is disabled	(Initial value)
1	A/D end interrupt request (ADI) is enabled	

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin.

Bit 5 ADST	Description	
0	A/D conversion is stopped	(Initial value)
1	1. Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends 2. Scan mode: A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode	

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode. For further information on operation in these modes, see section 13.4, Operation. Clear the ADST bit to 0 before switching the conversion mode.

Bit 4 SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

Bit 3—Clock Select (CKS): Selects the A/D conversion time. When $\phi_p = \phi/2$, the conversion time doubles. Clear the ADST bit to 0 before switching the conversion time.

Bit 3 CKS	Description	
0	Conversion time = 266 states (maximum) (when $\phi_p = \phi$)	(Initial value)
1	Conversion time = 134 states (maximum) (when $\phi_p = \phi$)	

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits and the SCAN bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection.

Group Selection	Channel Selection		Description	
CH2	**CH1**	**CH0**	**Single Mode**	**Scan Mode**
0	0	0	AN_0 (initial value)	AN_0
	0	1	AN_1	AN_0, AN_1
	1	0	AN_2	AN_0 to AN_2
	1	1	AN_3	AN_0 to AN_3
1	0	0	AN_4	AN_4
	0	1	AN_5	AN_4, AN_5
	1	0	AN_6	AN_4 to AN_6
	1	1	AN_7	AN_4 to AN_7

13.2.3 A/D Control Register (ADCR)

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

ADCR is an 8-bit readable/writable register that enables or disables external triggering of A/D conversion. ADCR is initialized to H'7F by a reset and in standby mode.

Bit 7—Trigger Enable (TRGE): Enables or disables external triggering of A/D conversion.

Bit 7 TRGE	Description	
0	A/D conversion cannot be externally triggered	(Initial value)
1	A/D conversion is enabled by the external trigger signal ($\overline{\text{ADTRG}}$) (A/D conversion can also be enabled by software)	

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

13.3 CPU Interface

ADDRA to ADDRD are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, although the upper byte can be be accessed directly by the CPU, the lower byte is read through an 8-bit temporary register (TEMP).

An A/D data register is read as follows. When the upper byte is read, the upper-byte value is transferred directly to the CPU and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 13-2 shows the data flow for access to an A/D data register.

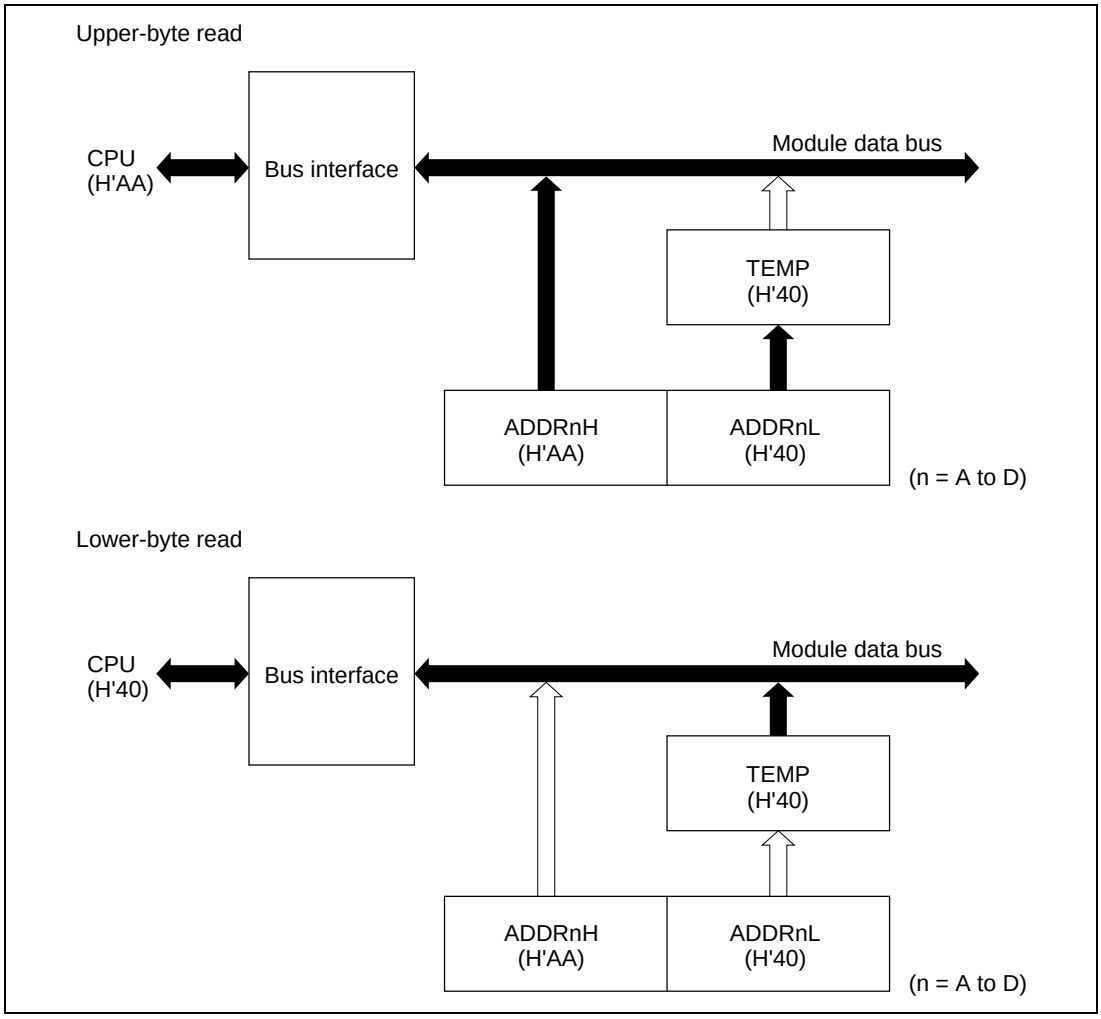


Figure 13-2 A/D Data Register Access Operation (Reading H'AA40)

13.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

13.4.1 Single Mode (SCAN = 0)

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts 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.

When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in ADF.

When the mode or analog input channel must be switched 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 mode or channel is changed.

Typical operations when channel 1 (AN_1) is selected in single mode are described next. Figure 13-3 shows a timing diagram for this example.

1. Single mode is selected (SCAN = 0), input channel AN_1 is selected (CH2 = CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into 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 in 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.

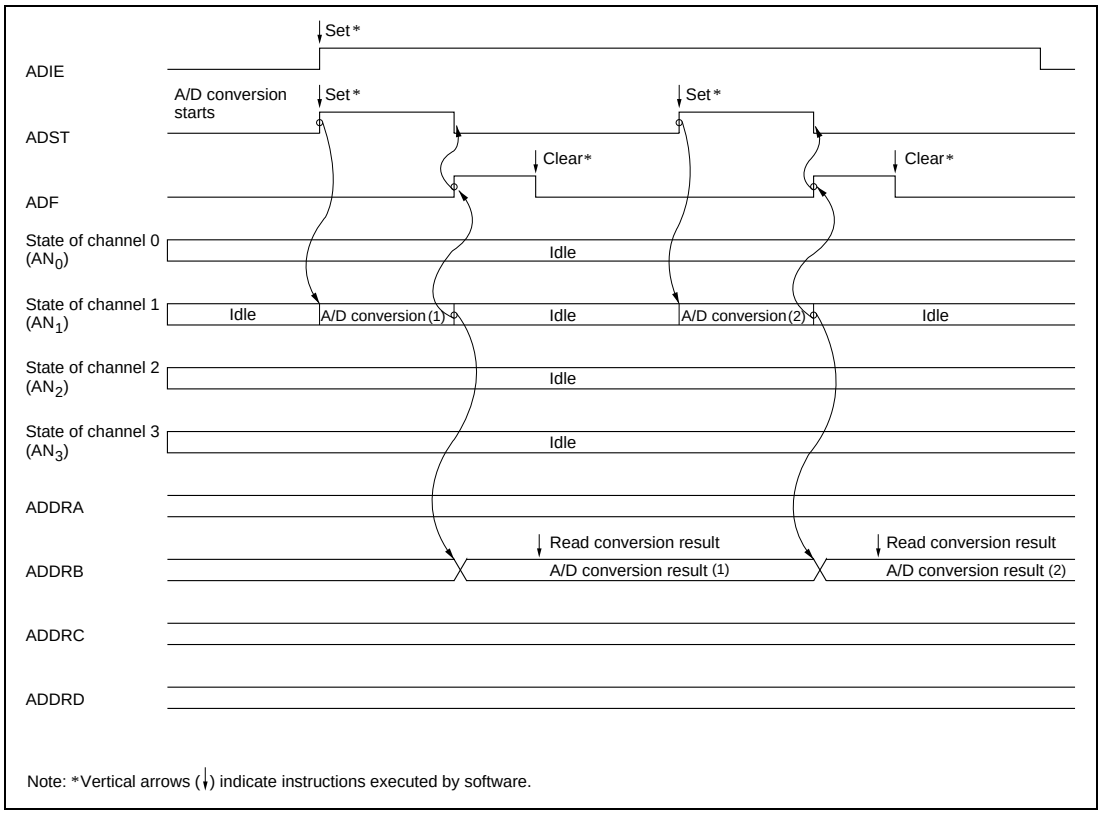


Figure 13-3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

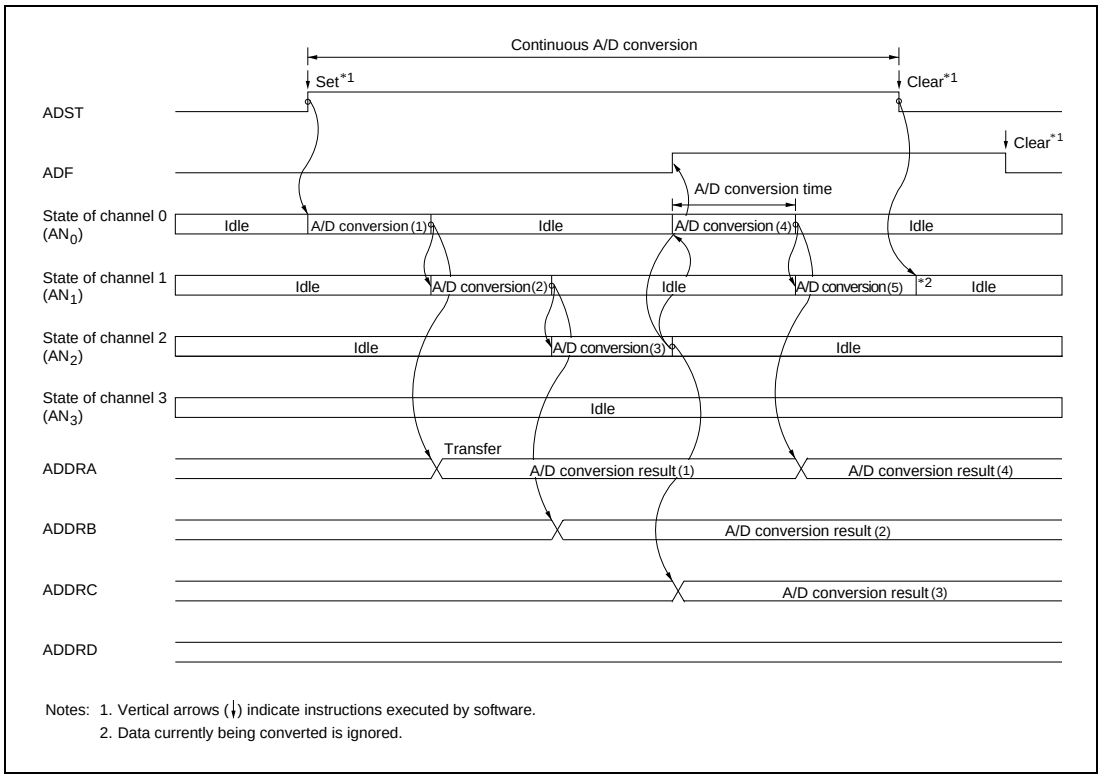
13.4.2 Scan Mode (SCAN = 1)

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 external trigger input, A/D conversion starts on the first channel in the group (AN_0 when CH2 = 0, AN_4 when CH2 = 1). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN_1 or AN_5) 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 A/D data registers corresponding to the channels.

When the mode or analog input channel selection 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. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in group 0 (AN_0 to AN_2) are selected in scan mode are described next. Figure 13-4 shows a timing diagram for this example.

1. Scan mode is selected (SCAN = 1), scan group 0 is selected (CH2 = 0), analog input channels AN_0 to AN_2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN_0) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN_1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN_2).
4. When conversion of all selected channels (AN_0 to AN_2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN_0) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.
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 (AN_0).



Notes: 1. Vertical arrows (↓) indicate instructions executed by software.
2. Data currently being converted is ignored.

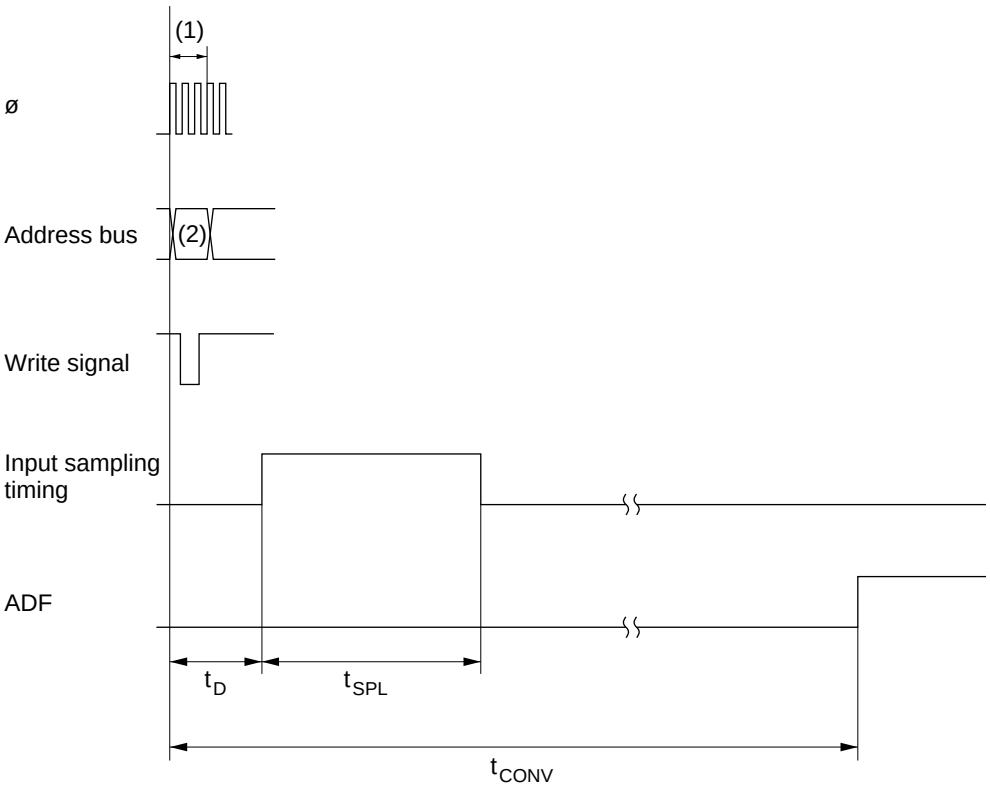
Figure 13-4 Example of A/D Converter Operation (Scan Mode, Channels AN_0 to AN_2 Selected)

13.4.3 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 13-5 shows the A/D conversion timing. Table 13-4 indicates the A/D conversion time.

As indicated in figure 13-5, 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 13-4.

In scan mode, the values given in table 13-4 apply to the first conversion. In the second and subsequent conversions the conversion time is fixed at 256 states when CKS = 0 or 128 states when CKS = 1. (when $\phi_P = \phi$)



Legend
(1): ADCSR write cycle
(2): ADCSR address
t_D: Synchronization delay
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 13-5 A/D Conversion Timing

Table 13-4 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1		
	Symbol	Min	Typ	Max	Min	Typ	Max
Synchronization delay	t_D	10	—	17	6	—	9
Input sampling time*	t_{SPL}	—	80	—	—	40	—
A/D conversion time*	t_{CONV}	259	—	266	131	—	134

Note: * Values in the table are numbers of states. Values are for $\phi_P = \phi$. When $\phi = \phi/2$, the values are doubled.

13.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE bit is set to 1 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A high-to-low transition 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 if the ADST bit had been set to 1 by software. Figure 13-6 shows the timing.

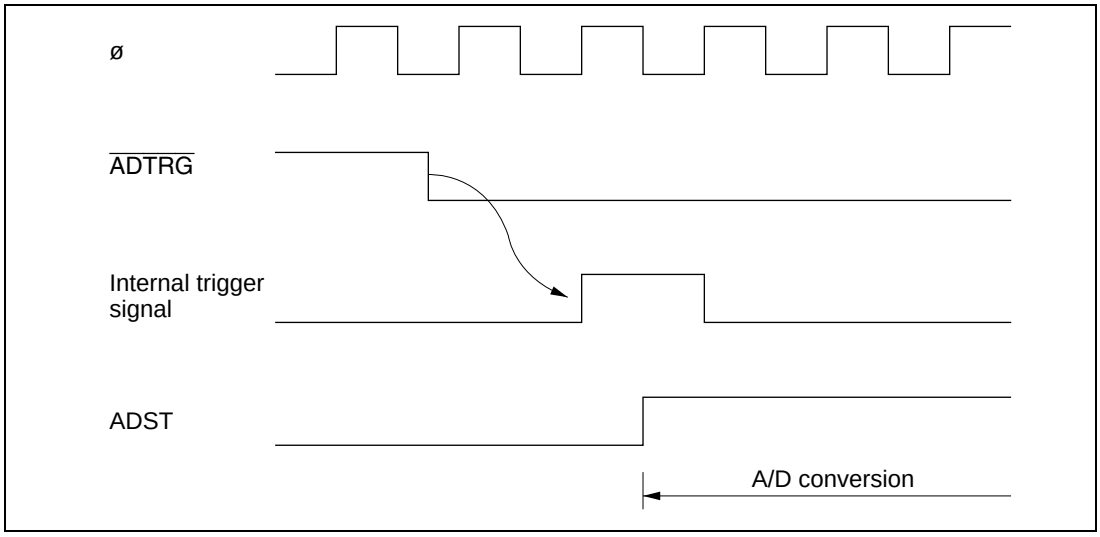


Figure 13-6 External Trigger Input Timing

13.5 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.

13.6 Usage Notes

When using the A/D converter, note the following points:

Analog Input Voltage Range: During A/D conversion, the voltages input to the analog input pins ANn should be in the range $AV_{SS} \le AN_n \le AV_{CC}$. (n = 0 to 7)

AV_{CC} and AV_{SS} Input Voltage: AV_{SS} should have the following value: $AV_{SS} = V_{SS}$. If the A/D converter is not used, the values should be $AV_{CC} = V_{CC}$ and $AV_{SS} = V_{SS}$.

Section 14 RAM

14.1 Overview

The H8/3534 has 1 kbyte of on-chip static RAM, and the H8/3522 has 512 bytes. The RAM is connected to the CPU by a 16-bit data bus. Both byte and word access to the on-chip RAM are performed in two states, enabling rapid data transfer and instruction execution.

The on-chip RAM is assigned to addresses H'FB80 to H'FF7F in the address space of the H8/3534 and addresses H'FD80 to H'FF7F in the address space of the H8/3522. The RAME bit in the system control register (SYSCR) can enable or disable the on-chip RAM.

14.1.1 Block Diagram

Figure 14-1 is a block diagram of the on-chip RAM.

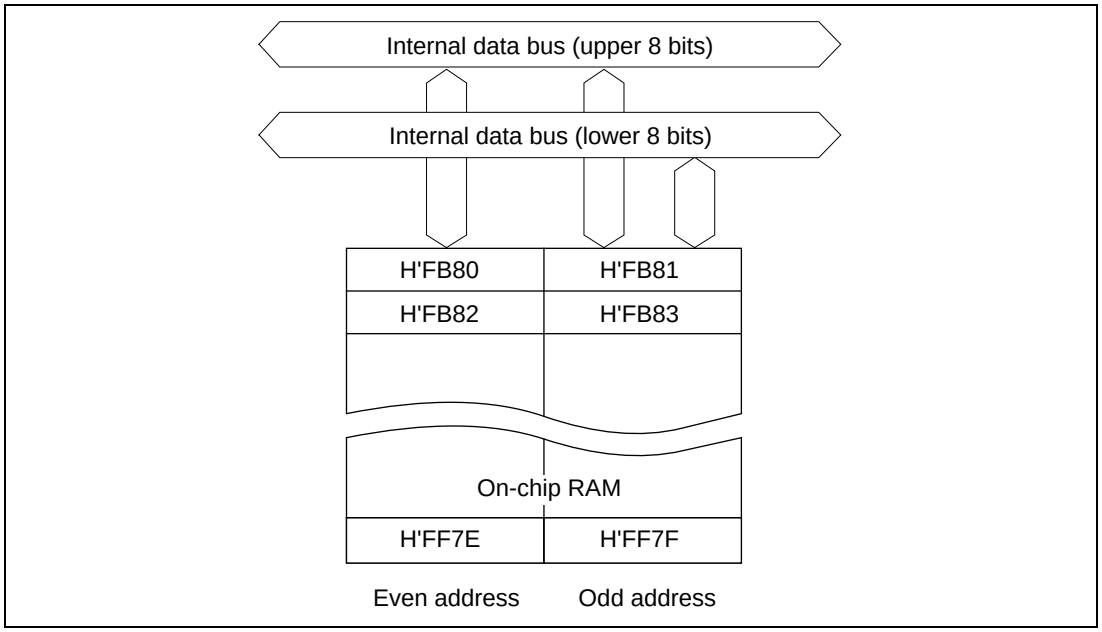


Figure 14-1 Block Diagram of On-Chip RAM (H8/3534)

14.1.2 RAM Enable Bit (RAME) in System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
[H8/3534]	SSBY	STS2	STS1	STS0	XRST	NMIEG	(HIE)	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
[H8/3522]	SSBY	STS2	STS1	STS0	XRST	NMIEG	—	RAME
Initial value	0	0	0	0	1	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	—	R/W

The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. See section 3.2, System Control Register, for the other SYSCR bits.

Bit 0—RAM Enable (RAME): This bit enables or disables the on-chip RAM. The RAME bit is initialized to 1 on the rising edge of the $\overline{\text{RES}}$ signal. The RAME bit is not initialized in software standby mode.

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

14.2 Operation

14.2.1 Expanded Modes (Modes 1 and 2)

If the RAME bit is set to 1, accesses to addresses H'FB80 to H'FF7F in the H8/3534 and addresses H'FD80 to H'FF7F in the H8/3522 are directed to the on-chip RAM.

If the RAME bit is cleared to 0, accesses to these addresses are directed to the external data bus.

14.2.2 Single-Chip Mode (Mode 3)

If the RAME bit is set to 1, accesses to addresses H'FB80 to H'FF7F in the H8/3534 and addresses H'FD80 to H'FF7F in the H8/3522 are directed to the on-chip RAM.

If the RAME bit is cleared to 0, the on-chip RAM data cannot be accessed. Attempted write access has no effect. Attempted read access always results in H'FF data being read.

Section 15 ROM

15.1 Overview

The size of the on-chip ROM is 32 kbytes in the H8/3534, and 16 kbytes in the H8/3522. The on-chip ROM is connected to the CPU via a 16-bit data bus. Both byte data and word data are accessed in two states, enabling rapid data transfer.

The on-chip ROM is enabled or disabled depending on the inputs at the mode pins (MD_1 and MD_0). See table 15-1.

Table 15-1 On-Chip ROM Usage in Each MCU Operating Mode

	Mode Pins		
Mode	**MD_1**	**MD_0**	**On-chip ROM**
Mode 1 (expanded mode)	0	1	Disabled (external addresses)
Mode 2 (expanded mode)	1	0	Enabled
Mode 3 (single-chip mode)	1	1	Enabled

15.1.1 Block Diagram

Figure 15-1 is a block diagram of the on-chip ROM.

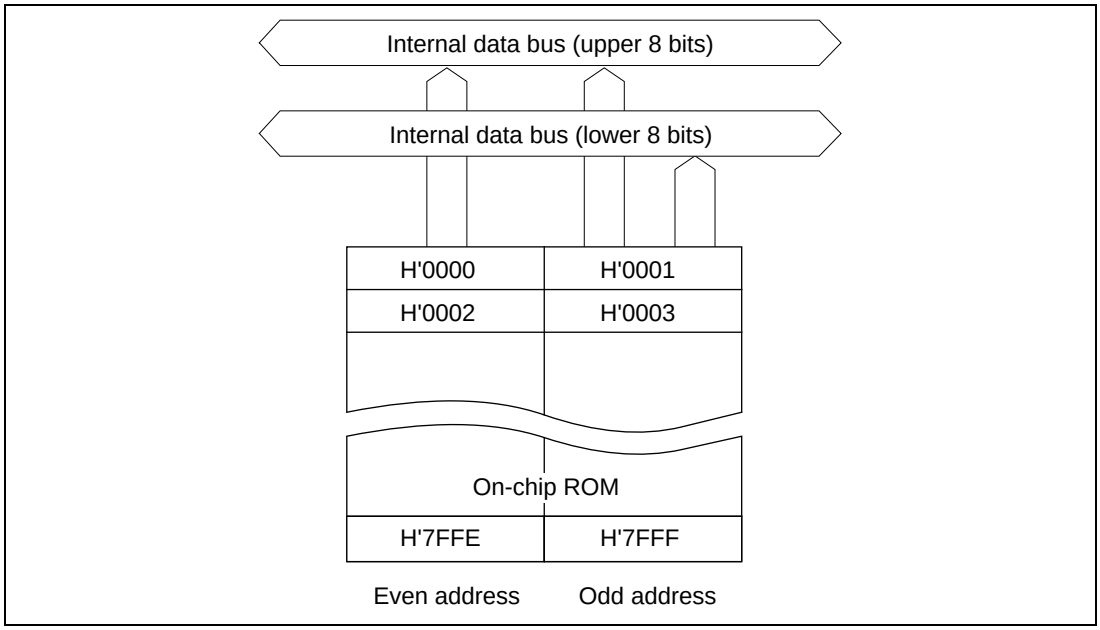


Figure 15-1 Block Diagram of On-Chip ROM (H8/3534 Single-Chip Mode)

Section 16 Power-Down State

16.1 Overview

The H8/3534 and H8/3522 have a sleep mode that greatly reduces power consumption by stopping CPU functions. The following two standby modes can also be set in addition to sleep mode, but since a guaranteed value is not set for power consumption in the standby modes, use of these modes is not recommended.

(1) Sleep mode

(2) Software standby mode

(3) Hardware standby mode

Table 16-1 lists the conditions for entering and leaving the power-down modes. It also indicates the status of the CPU, on-chip supporting modules, etc. in each power-down mode.

Table 16-1 Power-Down State

		State						
Mode	**Entering Procedure**	**Clock**	**CPU**	**CPU Reg's.**	**Sup. Mod.**	**RAM**	**I/O Ports**	**Exiting Methods**
Sleep mode	Execute SLEEP instruction	Active	Halted	Held	Active	Held	Held	• Interrupt • $\overline{\text{RES}}$ • $\overline{\text{STBY}}$
Software standby mode	Set SSBY bit in SYSCR 1, then execute SLEEP instruction	Halted	Halted	Held	Halted and initialized	Held	Held	• NMI IRQ0 to IRQ2 IRQ6 (include. KEYIN0 to KEYIN7) • RES • STBY
Hardware standby mode	Set $\overline{\text{STBY}}$ pin to low level	Halted	Halted	Undeter-mined	Halted and initialized	Held	High impe-dance state	• $\overline{\text{STBY}}$ and $\overline{\text{RES}}$

Notes: 1. SYSCR: System control register
2. SSBY: Software standby bit

16.1.1 System Control Register (SYSCR)

Four of the eight bits in the system control register (SYSCR) control the power-down state. These are bit 7 (SSBY) and bits 6 to 4 (STS2 to STS0). See table 16-2.

Table 16-2 System Control Register

Name	Abbreviation	R/W	Address
System control register	SYSCR	R/W	H'FFC4

Bit	7	6	5	4	3	2	1	0
[H8/3534]	SSBY	STS2	STS1	STS0	XRST	NMIEG	(HIE)	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
[H8/3522]	SSBY	STS2	STS1	STS0	XRST	NMIEG	—	RAME
Initial value	0	0	0	0	1	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	—	R/W

Bit 7—Software Standby (SSBY): This bit enables or disables the transition to software standby mode.

On recovery from the software standby mode by an external interrupt, SSBY remains set to 1. To clear this bit, software must write a 0.

Bit 7 SSBY	Description	
0	The SLEEP instruction causes a transition to sleep mode.	(Initial value)
1	The SLEEP instruction causes a transition to software standby mode.	

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the clock settling time when the chip recovers from software standby mode by an external interrupt. During the selected time, the clock oscillator runs but the CPU and on-chip supporting modules remain in standby. Set bits STS2 to STS0 according to the clock frequency to obtain a settling time of at least 8 ms. See table 16-3.

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description	
0	0	0	Settling time = 8,192 states	(Initial value)
0	0	1	Settling time = 16,384 states	
0	1	0	Settling time = 32,768 states	
0	1	1	Settling time = 65,536 states	
1	0	—	Settling time = 131,072 states	
1	1	—	Unused	

16.2 Sleep Mode

16.2.1 Transition to Sleep Mode

When the SSBY bit in the system control register is cleared to 0, execution of the SLEEP instruction causes a transition from the program execution state to sleep mode. After executing the SLEEP instruction, the CPU halts, but the contents of its internal registers remain unchanged. The on-chip supporting modules continue to operate normally.

16.2.2 Exit from Sleep Mode

The chip exits sleep mode when it receives an internal or external interrupt request, or a low input at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

(1) Exit by Interrupt: An interrupt releases sleep mode and starts the CPU's interrupt-handling sequence.

If an interrupt from an on-chip supporting module is disabled by the corresponding enable/disable bit in the module's control register, the interrupt cannot be requested, so it cannot wake the chip up. Similarly, the CPU cannot be awakened by an interrupt other than NMI if the I (interrupt mask) bit is set when the SLEEP instruction is executed.

(2) Exit by $\overline{\text{RES}}$ pin: When the $\overline{\text{RES}}$ pin goes low, the chip exits from sleep mode to the reset state.

(3) Exit by $\overline{\text{STBY}}$ pin: When the $\overline{\text{STBY}}$ pin goes low, the chip exits from sleep mode to hardware standby mode.

16.3 Software Standby Mode

16.3.1 Transition to Software Standby Mode

To enter software standby mode, set the standby bit (SSBY) in the system control register (SYSCR) to 1, then execute the SLEEP instruction.

In software standby mode, the system clock stops and chip functions halt, including both CPU functions and the functions of the on-chip supporting modules. The on-chip supporting modules and their registers are reset to their initial states, but as long as a minimum necessary voltage supply is maintained, the contents of the CPU registers and on-chip RAM remain unchanged. I/O ports retain their states.

16.3.2 Exit from Software Standby Mode

The chip can be brought out of software standby mode by an $\overline{\text{RES}}$ input, $\overline{\text{STBY}}$ input, or external interrupt input at the $\overline{\text{NMI}}$ pin, $\overline{\text{IRQ}_0}$ to $\overline{\text{IRQ}_2}$ pins, or $\overline{\text{IRQ}_6}$ pin* (including $\overline{\text{KEYIN}_0}$ to $\overline{\text{KEYIN}_7}$).

(1) Exit by Interrupt: When an NMI, IRQ_0, IRQ_1, IRQ_2, or IRQ_6* interrupt request signal is input, the clock oscillator begins operating. After the waiting time set in bits STS2 to STS0 of SYSCR, a stable clock is supplied to the entire chip, software standby mode is released, and interrupt exception-handling begins. IRQ_3, IRQ_4, IRQ_5, and IRQ_7 interrupts should be disabled before the transition to software standby (clear IRQ3E, IRQ4E, IRQ5E, and IRQ7E to 0)*.

Note: * Applies to the H8/3534 only.

(2) Exit by $\overline{\text{RES}}$ Pin: When the $\overline{\text{RES}}$ input goes low, the clock oscillator begins operating. When $\overline{\text{RES}}$ is brought to the high level (after allowing time for the clock oscillator to settle), the CPU starts reset exception handling. Be sure to hold $\overline{\text{RES}}$ low long enough for clock oscillation to stabilize.

(3) Exit by $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ input goes low, the chip exits from software standby mode to hardware standby mode.

16.3.3 Clock Settling Time for Exit from Software Standby Mode

Set bits STS2 to STS0 in SYSCR as follows:

- Crystal oscillator

  Set STS2 to STS0 for a settling time of at least 8 ms. Table 16-3 lists the settling times selected by these bits at several clock frequencies.
- External clock

  The STS bits can be set to any value.

Table 16-3 Times Set by Standby Timer Select Bits (Unit: ms)

				System Clock Frequency (MHz)			
STS2	**STS1**	**STS0**	**Settling Time(States)**	**10**	**8**	**6**	**4**
0	0	0	8,192	0.8	1.0	1.3	2.0
0	0	1	16,384	1.6	2.0	2.7	4.1
0	1	0	32,768	3.3	4.1	5.5	8.2
0	1	1	65,536	6.6	8.2	10.9	16.4
1	0	—	131,072	13.1	16.4	21.8	32.8

Note: Recommended values are printed in boldface.

16.3.4 Sample Application of Software Standby Mode

In this example the chip enters the software standby mode when $\overline{\text{NMI}}$ goes low and exits when $\overline{\text{NMI}}$ goes high, as shown in figure 16-1.

The NMI edge bit (NMIEG) in the system control register is originally cleared to 0, selecting the falling edge. When $\overline{\text{NMI}}$ goes low, the $\overline{\text{NMI}}$ interrupt handling routine sets NMIEG to 1, sets SSBY to 1 (selecting the rising edge), then executes the SLEEP instruction. The chip enters software standby mode. It recovers from software standby mode on the next rising edge of $\overline{\text{NMI}}$.

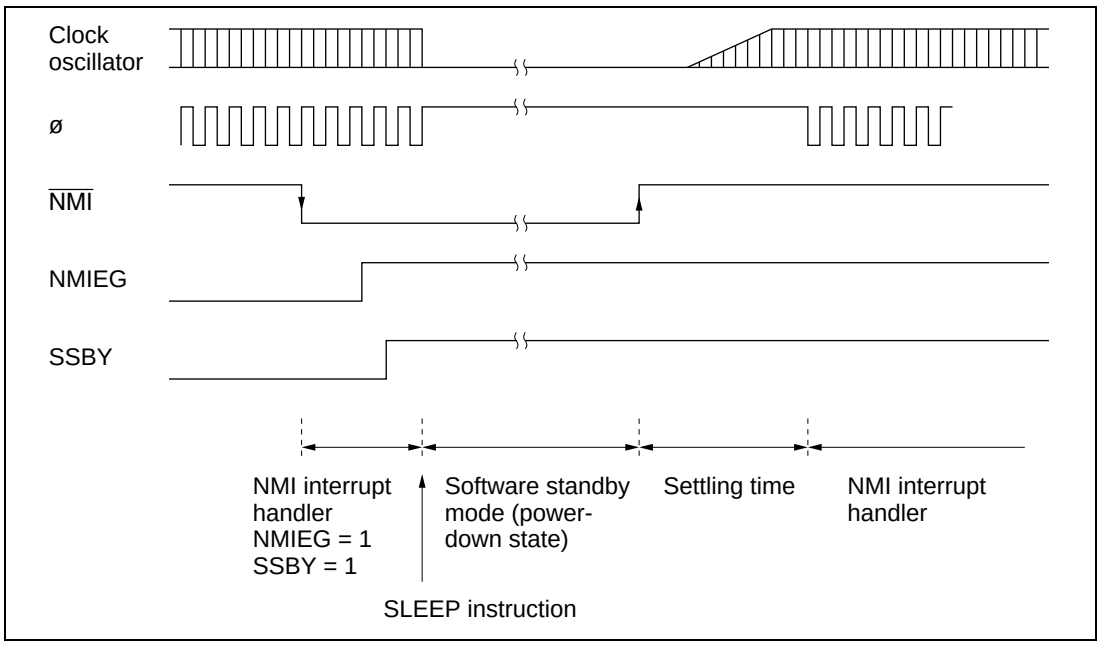


Figure 16-1 NMI Timing in Software Standby Mode (Application Example)

16.3.5 Application Notes

The I/O ports retain their present states in software standby mode. Thus, current dissipation caused by the output current is not reduced.

16.4 Hardware Standby Mode

16.4.1 Transition to Hardware Standby Mode

Regardless of its current state, the chip enters hardware standby mode whenever the $\overline{\text{STBY}}$ pin goes low.

Hardware standby mode halts the CPU, stopping all the functions of the on-chip supporting modules, and placing I/O ports in the high-impedance state. The registers of the on-chip supporting modules are reset to their initial values. Only the on-chip RAM is held unchanged, provided the minimum necessary voltage supply is maintained.

Notes: 1. The RAME bit in the system control register should be cleared to 0 before the $\overline{\text{STBY}}$ pin goes low.
2. Do not change the inputs at the mode pins (MD_1, MD_0) during hardware standby mode.

16.4.2 Recovery from Hardware Standby Mode

Recovery from the hardware standby mode requires inputs at both the $\overline{\text{STBY}}$ and $\overline{\text{RES}}$ pins. When the $\overline{\text{STBY}}$ pin goes high, the clock oscillator begins running. The $\overline{\text{RES}}$ pin should be low at this time and should be held low long enough for the clock to stabilize. When the $\overline{\text{RES}}$ pin changes from low to high, the reset sequence is executed and the chip returns to the program execution state.

16.4.3 Timing Relationships in Hardware Standby Mode

Figure 16-2 shows the timing relationships in hardware standby mode.

In the sequence shown, first $\overline{\text{RES}}$ goes low, then $\overline{\text{STBY}}$ goes low, at which point the chip enters hardware standby mode. To recover, first $\overline{\text{STBY}}$ goes high, then after the clock settling time, $\overline{\text{RES}}$ goes high.

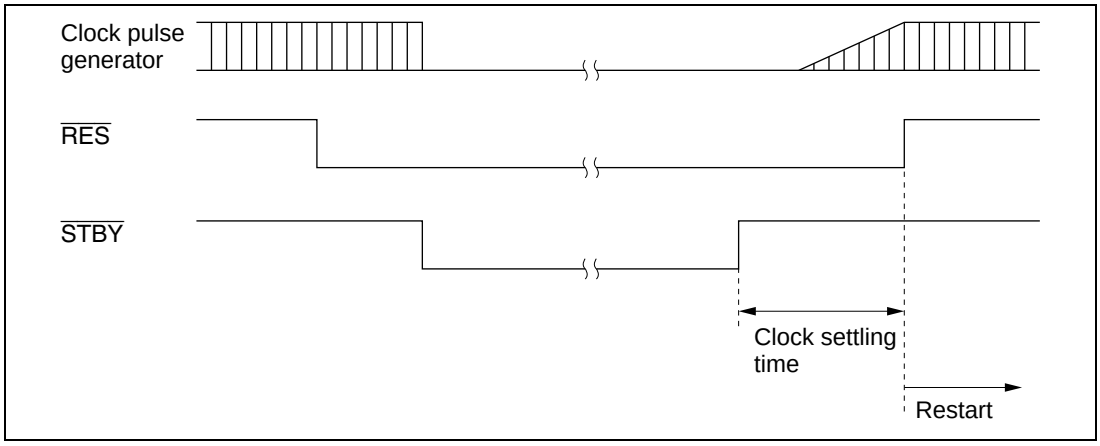


Figure 16-2 Hardware Standby Mode Timing

Section 17 Electrical Specifications

17.1 Absolute Maximum Ratings

Table 17-1 lists the absolute maximum ratings.

Table 17-1 Absolute Maximum Ratings

Item		Symbol	Rating	Unit
Supply voltage		V_{CC}	–0.3 to +7.0	V
Input voltage	Pins other than Ports7	V_{in}	–0.3 to V_{CC} + 0.3	V
	Port 7	V_{in}	–0.3 to AV_{CC} + 0.3	V
Analog 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}	–20 to +75	°C
Storage temperature		T_{stg}	–55 to +125	°C

Note: Exceeding the absolute maximum ratings shown in table 17-1 can permanently destroy the chip.

17.2 Electrical Characteristics

17.2.1 DC Characteristics

Table 17-2 lists the DC characteristics and Table 17-3 gives the allowable current output values.

Table 17-2 (a) H8/3534 DC Characteristics – Preliminary –
Conditions: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%[*1], V_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	$P6_7$ to $P6_0$[*3], (1)	V_T^-	1.0	—	—	V	
	$\overline{IRQ}_2$ to $\overline{IRQ}_0$[*4],	V_{T+}	—	—	$V_{CC} \times 0.7$		
	$\overline{IRQ}_7$ to $\overline{IRQ}_3$	$V_{T+} - V_{T-}$	0.4	—	—		
Input high voltage	$\overline{RES}$, $\overline{STBY}$, (2) MD_1, MD_0, EXTAL, $\overline{NMI}$	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	$P7_7$ to $P7_0$		2.0	—	AV_{CC} + 0.3		
	All input pins other than (1) and (2) above		2.0	—	V_{CC} + 0.3		
Input low voltage	$\overline{RES}$, $\overline{STBY}$, (3) MD_1, MD_0	V_{IL}	–0.3	—	0.5	V	
	All input pins other than (1) and (3) above		–0.3	—	0.8		
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—		I_{OH} = –1.0mA

Table 17-2 (a) H8/3534 DC Characteristics (cont) – Preliminary –

Conditions: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%[*1], V_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	$P1_7$ to $P1_0$, $P2_7$ to $P2_0$		—	—	1.0		I_{OL} = 10.0 mA
Input leakage current	$\overline{RES}$, $\overline{STBY}$	$\lvert I_{in} \rvert$	—	—	10.0	µA	Vin = 0.5 V to
	$\overline{NMI}$, MD_1, MD_0		—	—	1.0		V_{CC} – 0.5 V
	$P7_7$ to $P7_0$		—	—	1.0		Vin = 0.5 V to AV_{CC} – 0.5 V
Leakage current in three-state (off state)	Ports 1 to 6, 8, 9,	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	Vin = 0.5 V to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1 to 3	$-I_P$	30	—	250	µA	Vin = 0 V
	Port 6		60	—	500		
Input capacitance	$\overline{RES}$, $\overline{STBY}$ (4)	C_{in}	—	—	60	pF	Vin = 0 V,
	$\overline{NMI}$, MD_1		—	—	50		f = 1 MHz,
	$P9_7$, $P8_6$		—	—	20		Ta = 25°C
	All input pins other than (4)		—	—	15		
Current dissipation*2	Normal operation	I_{CC}	—	23	40	mA	f = 10 MHz
	Sleep mode		—	15	25		

Table 17-2 (a) H8/3534 DC Characteristics (cont) – Preliminary –

Conditions: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%[*1], V_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Analog supply current	During A/D conversion	AI_{CC}	—	2.0	5.0	mA	
Analog supply voltage[*1]		AV_{CC}	4.5	—	5.5	V	During operation
			2.0	—	5.5		While idle or when not in use

Notes: 1. Even when the A/D converter is not used, connect AV_{CC} to power supply V_{CC} and keep the applied voltage between 2.0 V and 5.5 V.

2. Current dissipation values assume that $V_{IH\ min}$ = V_{CC} – 0.5 V, $V_{IL\ max}$ = 0.5 V, all output pins are in the no-load state, and all input pull-up transistors are off.
3. $P6_7$ to $P6_0$ include supporting module inputs multiplexed with them.
4. IRQ_2 includes $\overline{ADTRG}$ multiplexed with it.

Table 17-2 (b) H8/3522 DC Characteristics – Preliminary –

Conditions: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%[*1], V_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	$P6_7$ to $P6_0$[*3], (1)	V_T^-	1.0	—	—	V	
	$\overline{IRQ}2$ to $\overline{IRQ}0$*4,	VT+	—	—	VCC × 0.7		
		VT+ – VT–	0.4	—	—		
Input high voltage	$\overline{RES}$, $\overline{STBY}$, (2) MD_1, MD_0, EXTAL, $\overline{NMI}$	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	$P7_7$ to $P7_0$		2.0	—	AV_{CC} + 0.3		
	All input pins other than (1) and (2) above		2.0	—	V_{CC} + 0.3		
Input low voltage	$\overline{RES}$, $\overline{STBY}$, (3) MD_1, MD_0	V_{IL}	–0.3	—	0.5	V	
	All input pins other than (1) and (3) above		–0.3	—	0.8		
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—		IOH = –1.0mA

Table 17-2 (b) H8/3522 DC Characteristics (cont) – Preliminary –

Conditions: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%[*1], V_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	P17 to P10, $P2_7$ to $P2_0$		—	—	1.0		I_{OL} = 10.0 mA
Input leakage current	$\overline{RES}$, $\overline{STBY}$	| I_{in} |	—	—	10.0	µA	Vin = 0.5 V to
	$\overline{NMI}$, MD_1, MD_0		—	—	1.0		VCC – 0.5 V
	$P7_7$ to $P7_0$		—	—	1.0		Vin = 0.5 V to AV_{CC} – 0.5 V
Leakage current in three-state (off state)	Ports 1 to 6	| I_{TSI} |	—	—	1.0	µA	Vin = 0.5 V to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1 to 3	$–I_P$	30	—	250	µA	Vin = 0 V
Input capacitance	$\overline{RES}$, $\overline{STBY}$ (4)	C_{in}	—	—	60	pF	Vin = 0 V, f = 1 MHz, Ta = 25°C
	$\overline{NMI}$, MD_1		—	—	30		
	All input pins other than (4)		—	—	15		
Current dissipation*2	Normal operation	I_{CC}	—	23	40	mA	f = 10 MHz
	Sleep mode		—	15	25		
Analog supply current	During A/D conversion	AI_{CC}	—	2.0	5.0	mA	
Analog supply voltage[*1]		AV_{CC}	4.5	—	5.5	V	During operation
			2.0	—	5.5		While idle or when not in use

Notes: 1. Even when the A/D converter is not used, connect AV_{CC} to power supply V_{CC} and keep the applied voltage between 2.0 V and 5.5 V.

2. Current dissipation values assume that $V_{IH\ min}$ = V_{CC} – 0.5 V, $V_{IL\ max}$ = 0.5 V, all output pins are in the no-load state, and all input pull-up transistors are off.

3. $P6_7$ to $P6_0$ include supporting module inputs multiplexed with them.

4. $\overline{IRQ_2}$ includes $\overline{ADTRG}$ multiplexed with it.

Table 17-3 Allowable Output Current Values – Preliminary –

Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, Ta = –20°C to +75°C

Item		Symbol	Min	Typ	Max	Unit
Allowable output low current (per pin)	Ports 1 and 2	I_{OL}	—	—	10	mA
	Other output pins		—	—	2	
Allowable output low current (total)	Ports 1 and 2, total	ΣI_{OL}	—	—	80	
	Total of all output		—	—	120	
Allowable output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	
Allowable output high current (total)	Total of all output	$\Sigma -I_{OH}$	—	—	40	

Note: To avoid degrading the reliability of the chip, be careful not to exceed the output current values in tables 17-3. In particular, when driving a darlington transistor pair or LED directly, be sure to insert a current-limiting resistor in the output path. See figures 17-1 and 17-2.

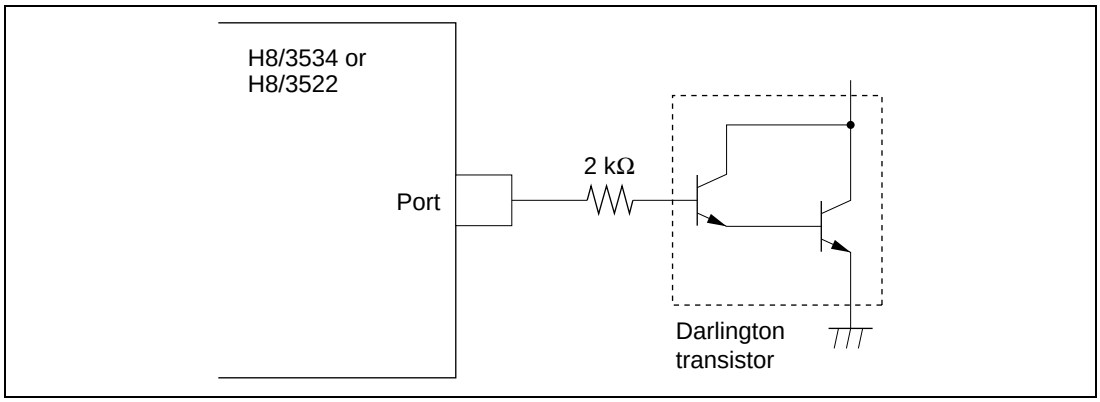


Figure 17-1 Example of Circuit for Driving a Darlington Transistor

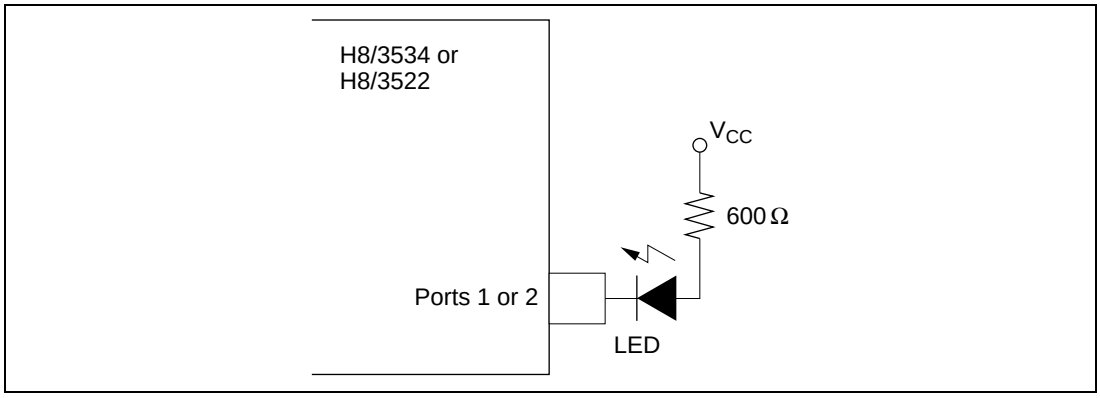


Figure 17-2 Example of Circuit for Driving an LED

17.2.2 AC Characteristics

The AC characteristics are listed in following tables. Bus timing parameters are given in table 17-4, control signal timing parameters in table 17-5, timing parameters of the on-chip supporting modules in table 17-6, and external clock output delay timing parameters in table 17-7.

Table 17-4 Bus Timing – Preliminary –
Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, ϕ = 4.0 MHz to maximum operating frequency, Ta = –20°C to +75°C

		10 MHz			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
Clock cycle time	t_{cyc}	100	250	ns	Fig. 17-4
Clock pulse width low	t_{CL}	35	–		
Clock pulse width high	t_{CH}	35	–		
Clock rise time	t_{Cr}	–	15		
Clock fall time	t_{Cf}	–	15		
Address delay time	t_{AD}	–	50		
Address hold time	t_{AH}	20	–		
Address strobe delay time	t_{ASD}	–	40		
Write strobe delay time	t_{WSD}	–	50		
Strobe delay time	t_{SD}	–	50		
Write strobe pulse width*	t_{WSW}	120	–		
Address setup time 1*	t_{AS1}	15	–		
Address setup time 2*	t_{AS2}	65	–		
Read data setup time	t_{RDS}	35	–		
Read data hold time*	t_{RDH}	0	–		
Read data access time*	t_{ACC}	–	170		
Write data delay time	t_{WDD}	–	75		
Write data setup time	t_{WDS}	5	–		
Write data hold time	t_{WDH}	20	–		
Wait setup time	t_{WTS}	40	–		Fig. 17-5
Wait hold time	t_{WTH}	10	–		

Note: * Values at maximum operating frequency

Table 17-5 Control Signal Timing – Preliminary –

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, ϕ = 4.0 MHz to maximum operating frequency, Ta = –20°C to +75°C

		10 MHz			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{RES}$ setup time	t_{RESS}	200	–	ns	Fig. 17-6
$\overline{RES}$ pulse width	t_{RESW}	10	–	t_{cyc}	
$\overline{NMI}$ setup time ($\overline{NMI}$, $\overline{IRQ_0}$ to $\overline{IRQ_7}$)	t_{NMIS}	150	–	ns	Fig. 17-7
$\overline{NMI}$ hold time ($\overline{NMI}$, $\overline{IRQ_0}$ to $\overline{IRQ_7}$)	t_{NMIH}	10	–		
Interrupt pulse width for recovery from software standby mode ($\overline{NMI}$, $\overline{IRQ_0}$ to $\overline{IRQ_2}$, $\overline{IRQ_6}$)	t_{NMIW}	200	–		
Crystal oscillator settling time (reset)	t_{OSC1}	20	–	ms	Fig. 17-8
Crystal oscillator settling time (software standby)	t_{OSC2}	8	–		Fig. 17-9

- Measurement Conditions for AC Characteristics

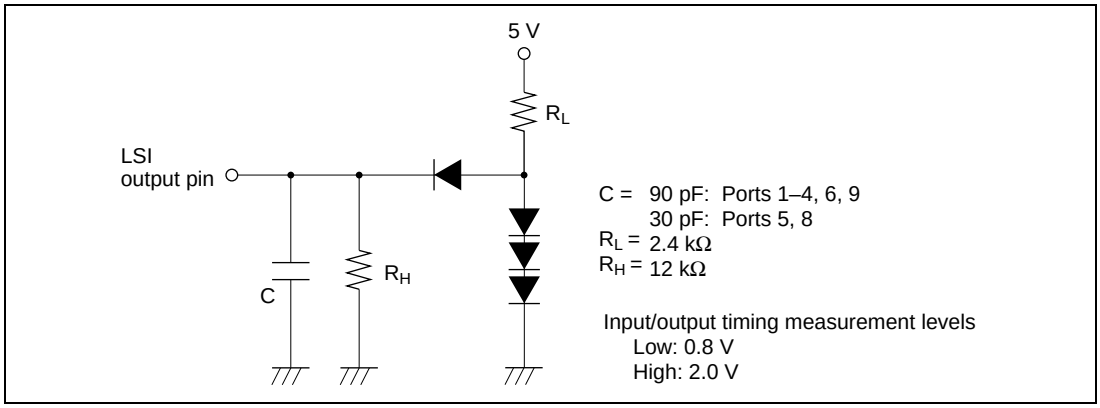


Figure 17-3 Output Load Circuit

Table 17-6 Timing Conditions of On-Chip Supporting Modules – Preliminary –

Conditions: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, ϕ = 4.0 MHz to maximum operating frequency, Ta = –20°C to +75°C

Item		Symbol	10 MHz Min	10 MHz Max	Unit	Test Conditions
FRT	Timer output delay time	t_{FTOD}	–	100	ns	Fig. 17-10
	Timer input setup time	t_{FTIS}	50	–		
	Timer clock input setup time	t_{FTCS}	50	–		Fig. 17-11
	Timer clock pulse width	t_{FTCWH} t_{FTCWL}	1.5	–	t_{cyc}	
TMR	Timer output delay time	t_{TMOD}	–	100	ns	Fig. 17-12
	Timer reset input setup time	t_{TMRS}	50	–		Fig. 17-14
	Timer clock input setup time	t_{TMCS}	50	–		Fig. 17-13
	Timer clock pulse width (single edge)	t_{TMCWH}	1.5	–	t_{cyc}	
	Timer clock pulse width (both edges)	t_{TMCWL}	2.5	–		
PWM	Timer output delay time [H8/3534]	t_{PWOD}	–	100	ns	Fig. 17-15
SCI	Input clock cycle (Async)	t_{Scyc}	4	–	t_{cyc}	Fig. 17-16
	(Sync)	t_{Scyc}	6	–		
	Transmit data delay time (Sync)	t_{TXD}	–	100	ns	Fig. 17-16
	Receive data setup time (Sync)	t_{RXS}	100	–		
	Receive data hold time (Sync)	t_{RXH}	100	–		
	Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	Fig. 17-17
Ports	Output data delay time	t_{PWD}	–	100	ns	Fig. 17-18
	Input data setup time	t_{PRS}	50	–		
	Input data hold time	t_{PRH}	50	–		

Table 17-7 External clock output delay Timing – Preliminary –
Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0V, Ta = –20°C to +75°C

Item	Symbol	Min	Max	Unit	Notes
External clock output delay time	t_{DEXT}*	500	—	µs	Fig. 17-19

Note: * t_{DEXT} includes to $\overline{RES}$ pulse width t_{RESW} (10 tcyc).

17.2.3 A/D Converter Characteristics

Table 17-8 lists the characteristics of the on-chip A/D converter.

Table 17-8 A/D Converter Characteristics – Preliminary –
Conditions: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%, V_{SS} = AV_{SS} = 0 V, ϕ = 4.0 MHz to maximum operating frequency, Ta = –20°C to +75°C

	10 MHz			
Item	Min	Typ	Max	Unit
Resolution	10	10	10	Bits
Conversion (single mode)*	—	—	13.4	µs
Analog input capacitance	—	—	20	pF
Allowable signal source impedance	—	—	10	kΩ
Nonlinearity error	—	—	±3.0	LSB
Offset error	—	—	±3.5	
Full-scale error	—	—	±3.5	
Quantizing error	—	—	±0.5	
Absolute accuracy	—	—	±4.0	

Note: * Values at maximum operating frequency

17.3 MCU Operational Timing

This section provides the following timing charts:

17.3.1	Bus Timing	Figures 17-4 and 17-5
17.3.2	Control Signal Timing	Figures 17-6 to 17-9
17.3.3	16-Bit Free-Running Timer Timing	Figures 17-10 and 17-11
17.3.4	8-Bit Timer Timing	Figures 17-12 to 17-14
17.3.5	PWM Timer Timing	Figure 17-15
17.3.6	SCI Timing	Figures 17-16 and 17-17
17.3.7	I/O Port Timing	Figure 17-18
17.3.8	External Clock Output Timing	Figure 17-19

17.3.1 Bus Timing

(1) Basic Bus Cycle (without Wait States) in Expanded Modes

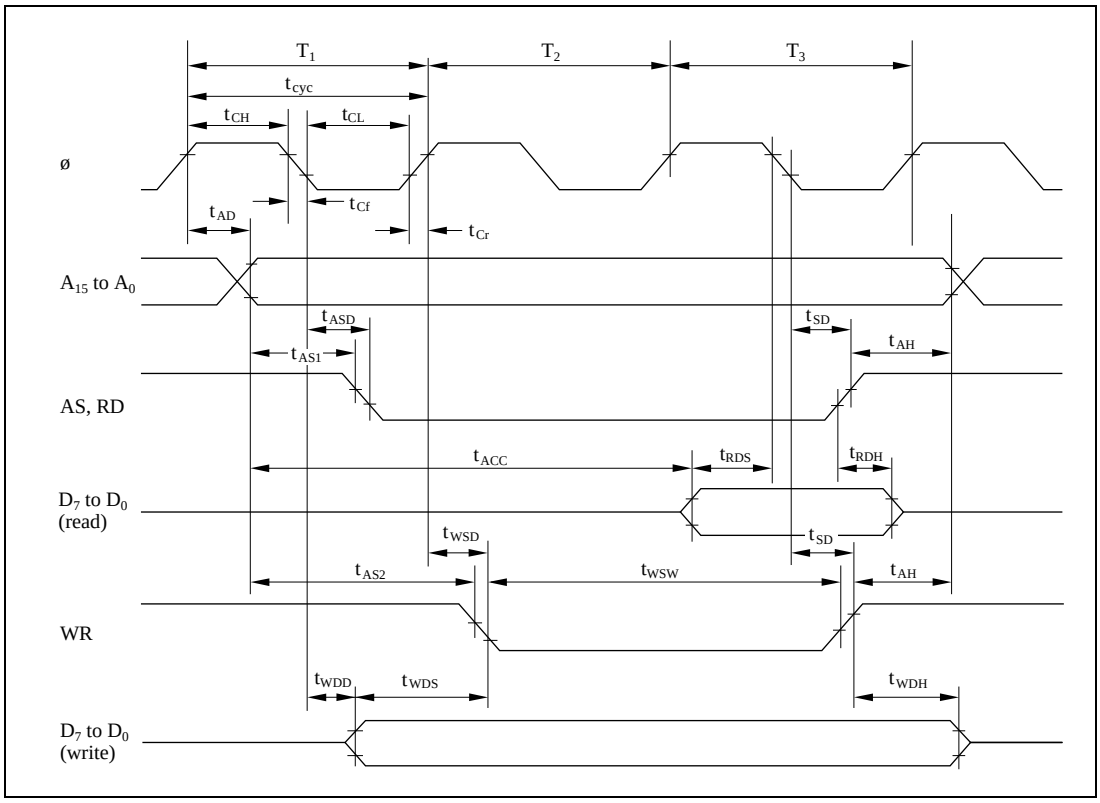


Figure 17-4 Basic Bus Cycle (without Wait States) in Expanded Modes

(2) Basic Bus Cycle (with 1 Wait State) in Expanded Modes

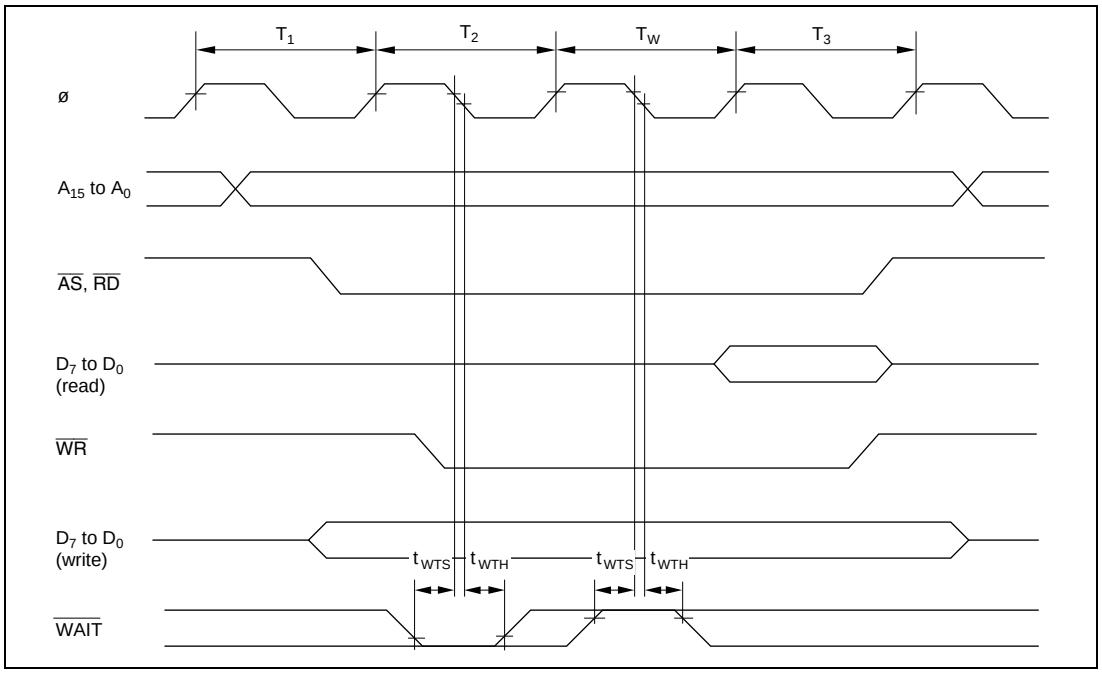


Figure 17-5 Basic Bus Cycle (with 1 Wait State) in Expanded Modes (Modes 1 and 2)

17.3.2 Control Signal Timing

(1) Reset Input Timing

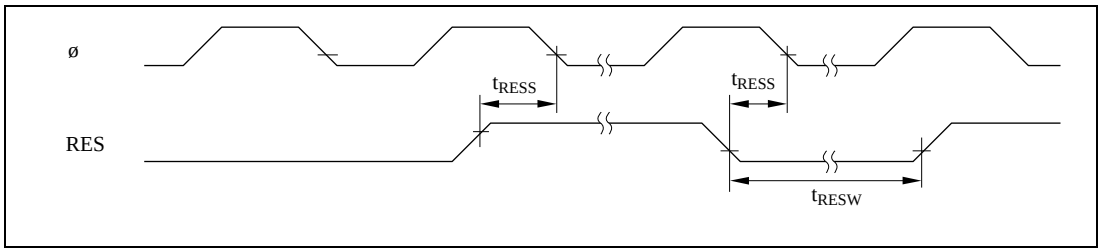


Figure 17-6 Reset Input Timing

(2) Interrupt Input Timing

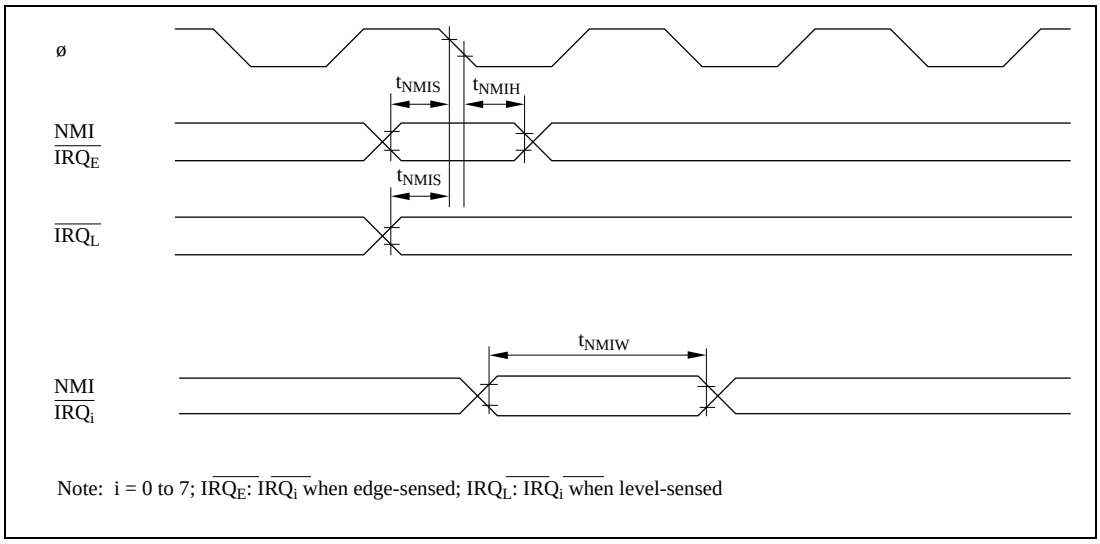


Figure 17-7 Interrupt Input Timing

(3) Clock Settling Timing

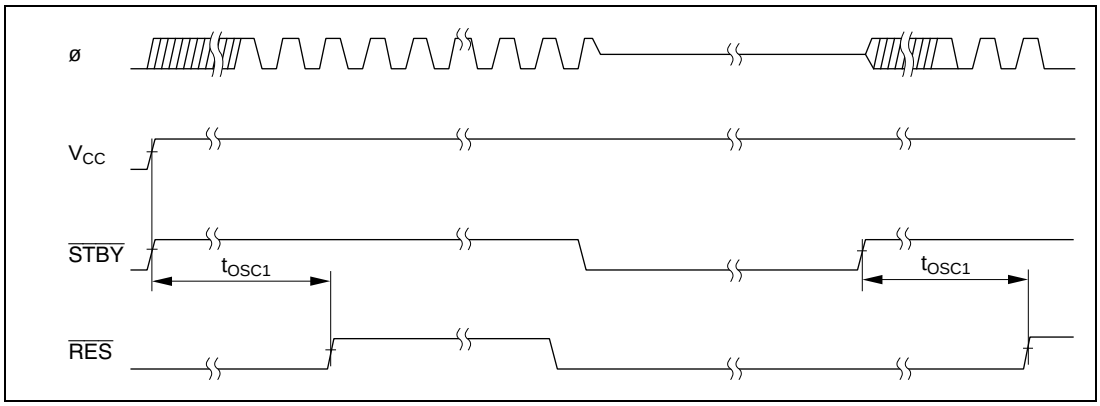


Figure 17-8 Clock Settling Timing

(4) Clock Settling Timing for Recovery from Software Standby Mode

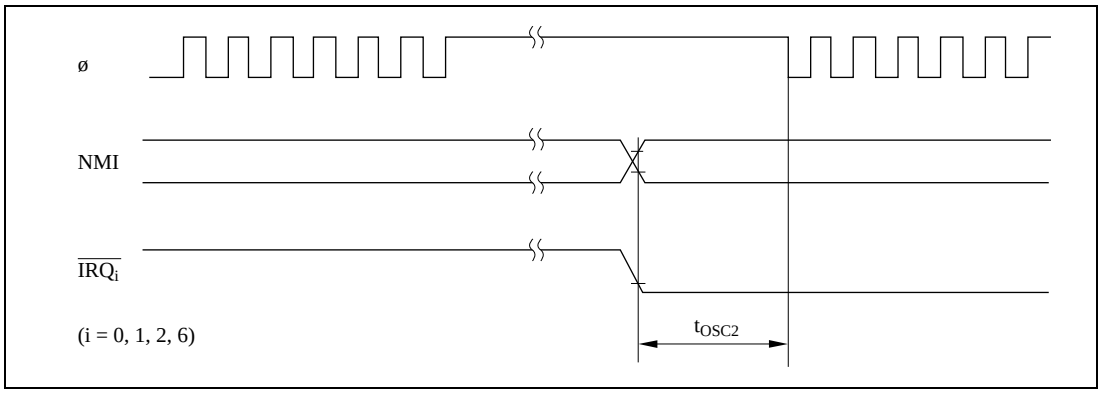


Figure 17-9 Clock Settling Timing for Recovery from Software Standby Mode

17.3.3 16-Bit Free-Running Timer Timing

(1) Free-Running Timer Input/Output Timing

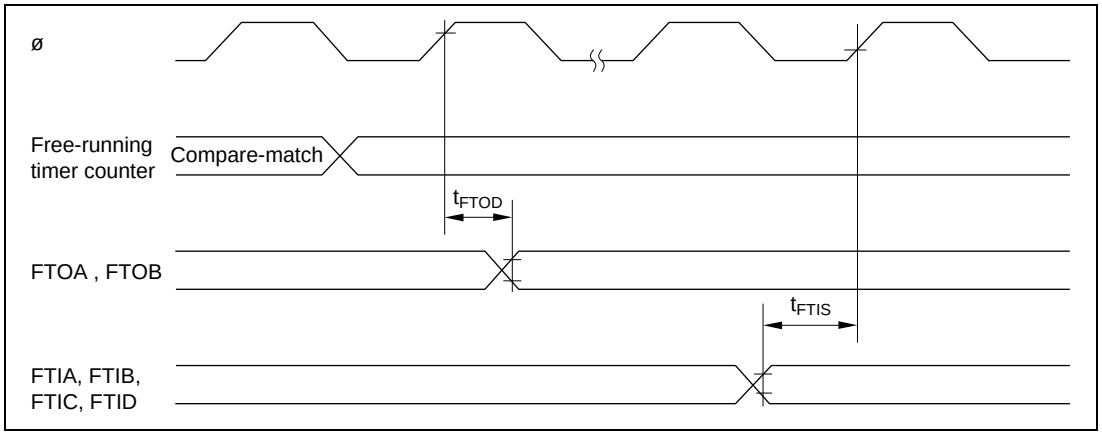


Figure 17-10 Free-Running Timer Input/Output Timing

(2) External Clock Input Timing for Free-Running Timer

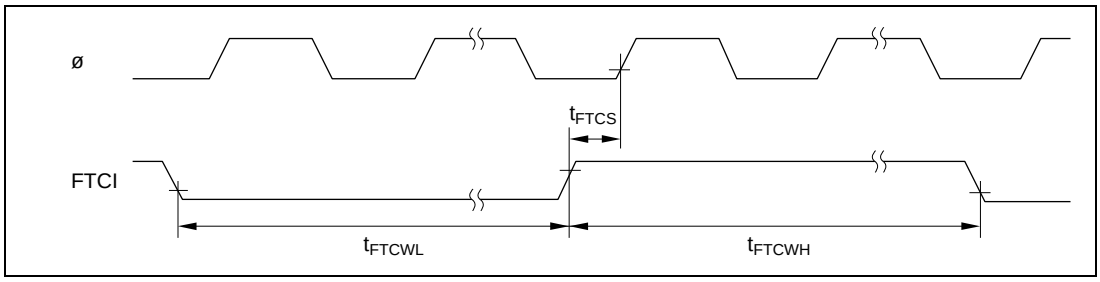


Figure 17-11 External Clock Input Timing for Free-Running Timer

17.3.4 8-Bit Timer Timing

(1) 8-Bit Timer Output Timing

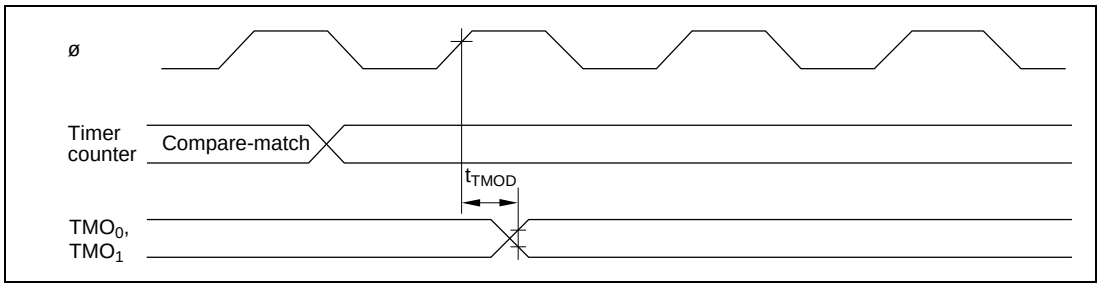


Figure 17-12 8-Bit Timer Output Timing

(2) 8-Bit Timer Clock Input Timing

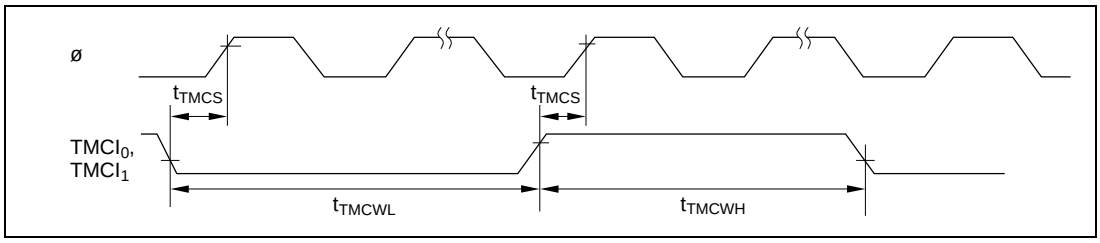


Figure 17-13 8-Bit Timer Clock Input Timing

(3) 8-Bit Timer Reset Input Timing

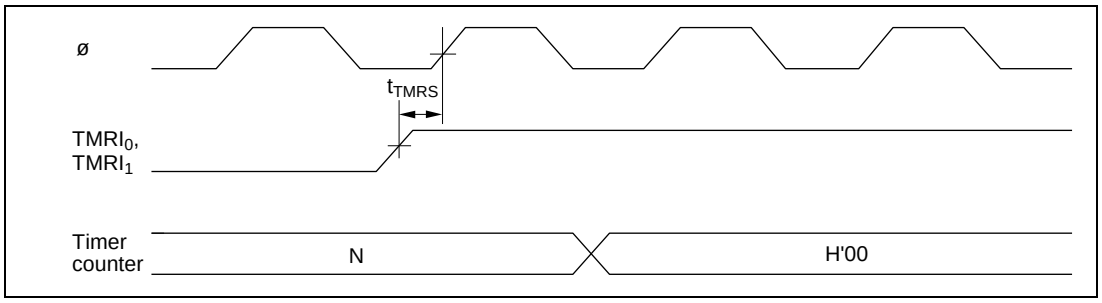


Figure 17-14 8-Bit Timer Reset Input Timing

17.3.5 Pulse Width Modulation Timer Timing [H8/3534]

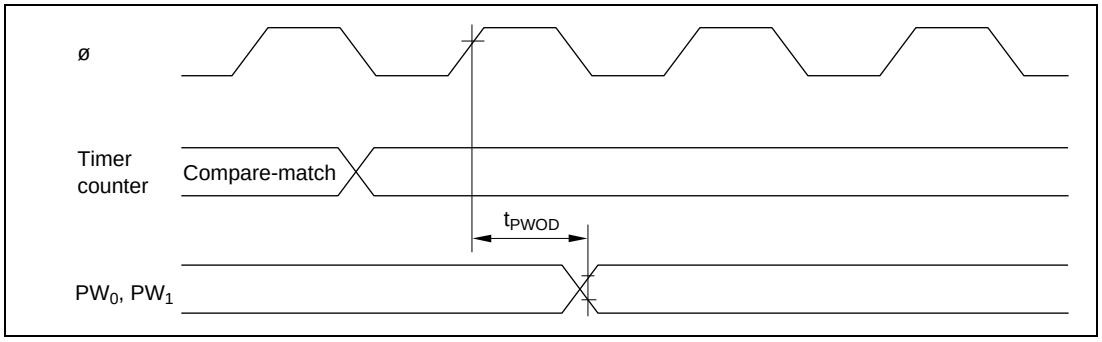


Figure 17-15 PWM Timer Output Timing

17.3.6 Serial Communication Interface Timing

(1) SCI Input/Output Timing

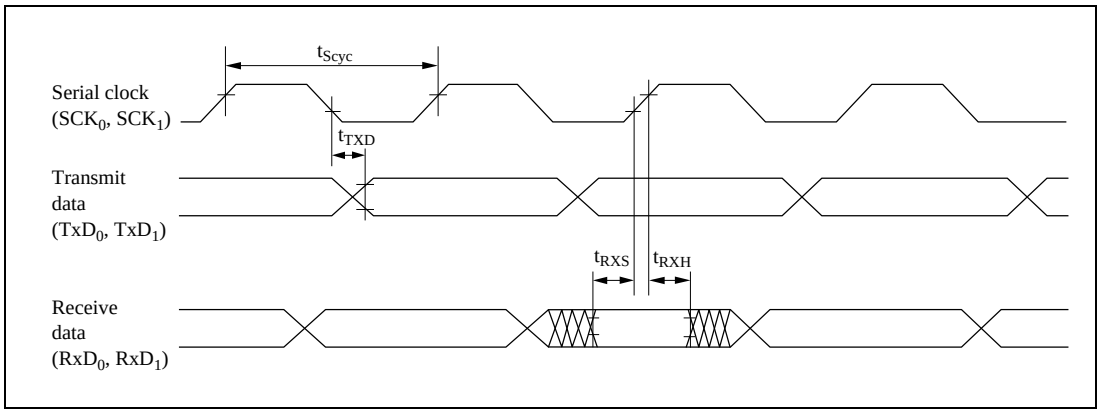


Figure 17-16 SCI Input/Output Timing (Synchronous Mode)

(2) SCI Input Clock Timing

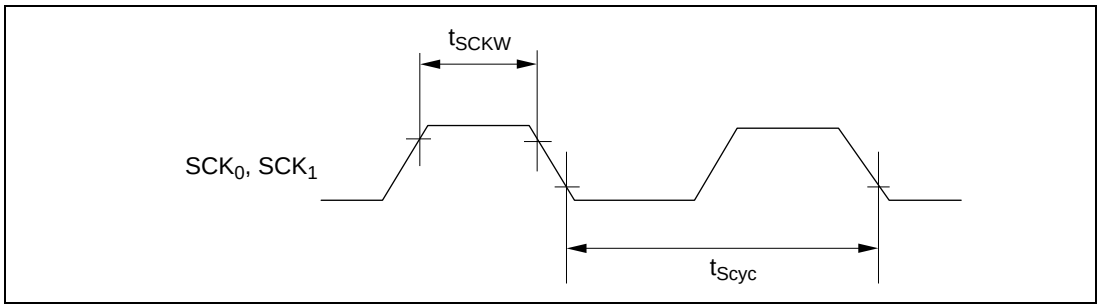
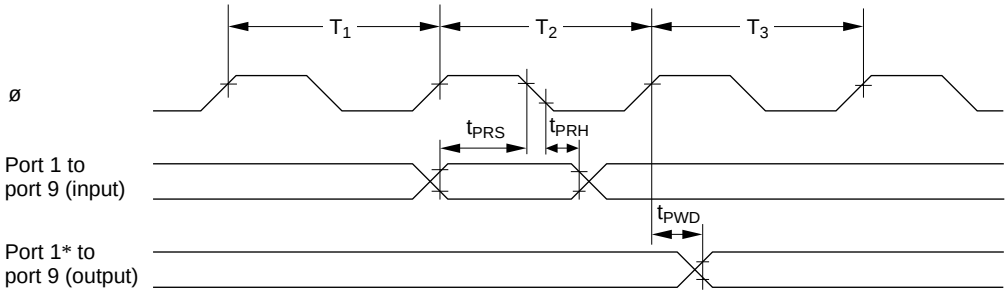


Figure 17-17 SCI Input Clock Timing

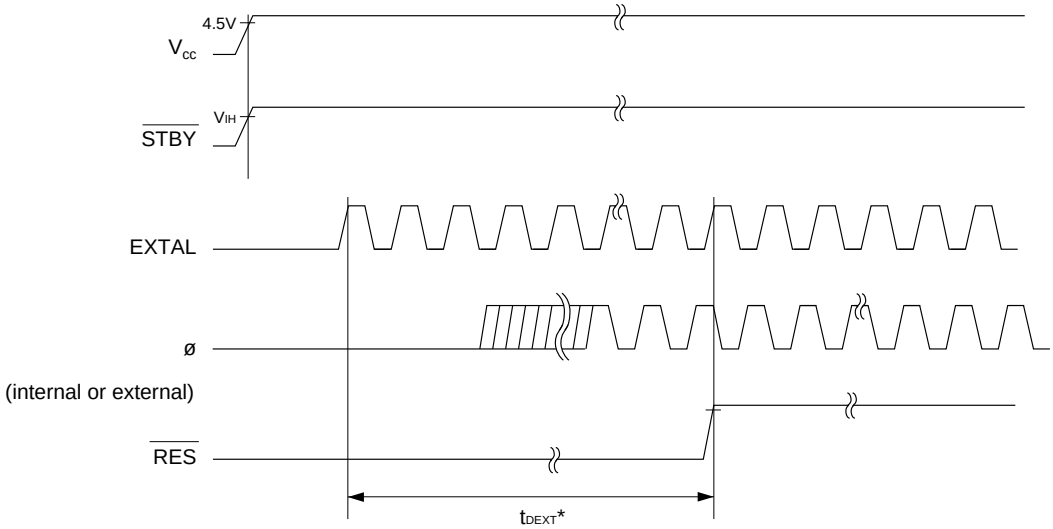
17.3.7 I/O Port Timing



Note: * Except $P9_6$ and $P7_7$ to $P7_0$ [H8/3534]; except $P7_7$ to $P7_0$ and $P4_6$ [H8/3522].

Figure 17-18 I/O Port Input/Output Timing

17.3.8 External Clock Output Timing



Note: * t_{DEXT} includes an $\overline{RES}$ pulse width (t_{RESW}) of 10 t_{cyc}.

Figure 17-19 External clock output delay Timing

Appendix A CPU Instruction Set

A.1 Instruction Set List

Operation Notation

Rd8/16	General register (destination) (8 or 16 bits)
Rs8/16	General register (source) (8 or 16 bits)
Rn8/16	General register (8 or 16 bits)
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
#xx:3/8/16	Immediate data (3, 8, or 16 bits)
d:8/16	Displacement (8 or 16 bits)
@aa:8/16	Absolute address (8 or 16 bits)
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Exclusive logical OR
→	Move
¬	NOT (logical complement)

Condition Code Notation

Δ	Modified according to the instruction result
*	Undetermined (unpredictable)
0	Always cleared to 0
—	Not affected by the instruction result

Table A-1 Instruction Set

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16,Rn)	@–Rn/@Rn+	@aa:8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
MOV.B #xx:8, Rd	B	#xx:8 → Rd8	2									—	—	Δ	Δ	0	—	2
MOV.B Rs, Rd	B	Rs8 → Rd8		2								—	—	Δ	Δ	0	—	2
MOV.B @Rs, Rd	B	@Rs16 → Rd8			2							—	—	Δ	Δ	0	—	4
MOV.B @(d:16, Rs), Rd	B	@(d:16, Rs16)→ Rd8				4						—	—	Δ	Δ	0	—	6
MOV.B @Rs+, Rd	B	@Rs16 → Rd8 Rs16+1 → Rs16					2					—	—	Δ	Δ	0	—	6
MOV.B @aa:8, Rd	B	@aa:8 → Rd8						2				—	—	Δ	Δ	0	—	4
MOV.B @aa:16, Rd	B	@aa:16 → Rd8						4				—	—	Δ	Δ	0	—	6
MOV.B Rs, @Rd	B	Rs8 → @Rd16			2							—	—	Δ	Δ	0	—	4
MOV.B Rs, @(d:16, Rd)	B	Rs8 → @(d:16, Rd16)				4						—	—	Δ	Δ	0	—	6
MOV.B Rs, @–Rd	B	Rd16–1 → Rd16 Rs8 → @Rd16					2					—	—	Δ	Δ	0	—	6
MOV.B Rs, @aa:8	B	Rs8 → @aa:8						2				—	—	Δ	Δ	0	—	4
MOV.B Rs, @aa:16	B	Rs8 → @aa:16						4				—	—	Δ	Δ	0	—	6
MOV.W #xx:16, Rd	W	#xx:16 → Rd16	4									—	—	Δ	Δ	0	—	4
MOV.W Rs, Rd	W	Rs16 → Rd16		2								—	—	Δ	Δ	0	—	2
MOV.W @Rs, Rd	W	@Rs16 → Rd16			2							—	—	Δ	Δ	0	—	4
MOV.W @(d:16, Rs), Rd	W	@(d:16, Rs16) → Rd16				4						—	—	Δ	Δ	0	—	6
MOV.W @Rs+, Rd	W	@Rs16 → Rd16 Rs16+2 → Rs16					2					—	—	Δ	Δ	0	—	6
MOV.W @aa:16, Rd	W	@aa:16 → Rd16						4				—	—	Δ	Δ	0	—	6
MOV.W Rs, @Rd	W	Rs16 → @Rd16			2							—	—	Δ	Δ	0	—	4
MOV.W Rs, @(d:16, Rd)	W	Rs16 → @(d:16, Rd16)				4						—	—	Δ	Δ	0	—	6
MOV.W Rs, @–Rd	W	Rd16–2 → Rd16 Rs16 → @Rd16					2					—	—	Δ	Δ	0	—	6
MOV.W Rs, @aa:16	W	Rs16 → @aa:16						4				—	—	Δ	Δ	0	—	6
POP Rd	W	@SP → Rd16 SP+2 → SP					2					—	—	Δ	Δ	0	—	6
PUSH Rs	W	SP–2 → SP Rs16 → @SP					2					—	—	Δ	Δ	0	—	6

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16,Rn)	@–Rn/@Rn+	@aa:8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
MOVFPE @aa:16, Rd	B	Not supported																(5)
MOVTPE Rs, @aa:16	B	Not supported																(5)
EEPMOV	—	if R4L≠0 then Repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L Until R4L=0 else next									4	—	—	—	—	—	—	(4)
ADD.B #xx:8, Rd	B	Rd8+#xx:8 → Rd8	2									—	Δ	Δ	Δ	Δ	Δ	2
ADD.B Rs, Rd	B	Rd8+Rs8 → Rd8		2								—	Δ	Δ	Δ	Δ	Δ	2
ADD.W Rs, Rd	W	Rd16+Rs16 → Rd16		2								—	(1)	Δ	Δ	Δ	Δ	2
ADDX.B #xx:8, Rd	B	Rd8+#xx:8 +C → Rd8	2									—	Δ	Δ	(2)	Δ	Δ	2
ADDX.B Rs, Rd	B	Rd8+Rs8 +C → Rd8		2								—	Δ	Δ	(2)	Δ	Δ	2
ADDS.W #1, Rd	W	Rd16+1 → Rd16		2								—	—	—	—	—	—	2
ADDS.W #2, Rd	W	Rd16+2 → Rd16		2								—	—	—	—	—	—	2
INC.B Rd	B	Rd8+1 → Rd8		2								—	—	Δ	Δ	Δ	—	2
DAA.B Rd	B	Rd8 decimal adjust → Rd8		2								—	*	Δ	Δ	*	(3)	2
SUB.B Rs, Rd	B	Rd8–Rs8 → Rd8		2								—	Δ	Δ	Δ	Δ	Δ	2
SUB.W Rs, Rd	W	Rd16–Rs16 → Rd16		2								—	(1)	Δ	Δ	Δ	Δ	2
SUBX.B #xx:8, Rd	B	Rd8–#xx:8 –C → Rd8	2									—	Δ	Δ	(2)	Δ	Δ	2
SUBX.B Rs, Rd	B	Rd8–Rs8 –C → Rd8		2								—	Δ	Δ	(2)	Δ	Δ	2
SUBS.W #1, Rd	W	Rd16–1 → Rd16		2								—	—	—	—	—	—	2
SUBS.W #2, Rd	W	Rd16–2 → Rd16		2								—	—	—	—	—	—	2
DEC.B Rd	B	Rd8–1 → Rd8		2								—	—	Δ	Δ	Δ	—	2
DAS.B Rd	B	Rd8 decimal adjust → Rd8		2								—	*	Δ	Δ	*	—	2
NEG.B Rd	B	0–Rd8 → Rd8		2								—	Δ	Δ	Δ	Δ	Δ	2
CMP.B #xx:8, Rd	B	Rd8–#xx:8	2									—	Δ	Δ	Δ	Δ	Δ	2
CMP.B Rs, Rd	B	Rd8–Rs8		2								—	Δ	Δ	Δ	Δ	Δ	2
CMP.W Rs, Rd	W	Rd16–Rs16		2								—	(1)	Δ	Δ	Δ	Δ	2

Table A-1 Instruction Set (cont)

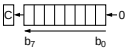
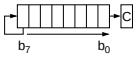
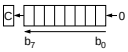
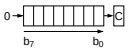
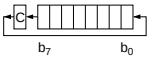
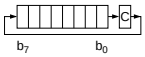
Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16,Rn)	@–Rn/@Rn+	@aa:8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
MULXU.B Rs, Rd	B	Rd8 × Rs8→Rd16		2								—	—	—	—	—	—	14
DIVXU.B Rs, Rd	B	Rd16÷Rs8→Rd16 (RdH:remainder, RdL:quotient)		2								—	—	(6)	(7)	—	—	14
AND.B#xx:8,Rd	B	Rd8∧#xx:8→ Rd8	2									—	—	Δ	Δ	0	—	2
AND.B Rs, Rd	B	Rd8∧Rs8 → Rd8		2								—	—	Δ	Δ	0	—	2
OR.B #xx:8, Rd	B	Rd8∨#xx:8 → Rd8	2									—	—	Δ	Δ	0	—	2
OR.B Rs, Rd	B	Rd8∨Rs8 → Rd8		2								—	—	Δ	Δ	0	—	2
XOR.B #xx:8, Rd	B	Rd8⊕#xx:8 → Rd8	2									—	—	Δ	Δ	0	—	2
XOR.B Rs, Rd	B	Rd8⊕Rs8 → Rd8		2								—	—	Δ	Δ	0	—	2
NOT.B Rd	B	$\overline{Rd8}$ → Rd8		2								—	—	Δ	Δ	0	—	2
SHAL.B Rd	B			2								—	—	Δ	Δ	Δ	Δ	2
SHAR.B Rd	B			2								—	—	Δ	Δ	0	Δ	2
SHLL.B Rd	B			2								—	—	0	Δ	0	Δ	2
SHLR.B Rd	B			2								—	—	0	Δ	0	Δ	2
ROTXL.B Rd	B			2								—	—	Δ	Δ	0	Δ	2
ROTXR.B Rd	B			2								—	—	Δ	Δ	0	Δ	2

Table A-1 Instruction Set (cont)

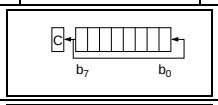
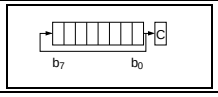
Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16,Rn)	@–Rn/@Rn+	@aa:8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
ROTL.BRd	B			2								—	—	Δ	Δ	0	Δ	2
ROTR.BRd	B			2								—	—	Δ	Δ	0	Δ	2
BSET #xx:3,Rd	B	(#xx:3 of Rd8) ← 1		2								—	—	—	—	—	—	2
BSET #xx:3, @Rd	B	(#xx:3 of @Rd16) ← 1			4							—	—	—	—	—	—	8
BSET #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 1						4				—	—	—	—	—	—	8
BSET Rn, Rd	B	(Rn8 of Rd8) ← 1		2								—	—	—	—	—	—	2
BSET Rn, @Rd	B	(Rn8 of @Rd16) ← 1			4							—	—	—	—	—	—	8
BSET Rn, @aa:8	B	(Rn8 of @aa:8) ← 1						4				—	—	—	—	—	—	8
BCLR #xx:3, Rd	B	(#xx:3 of Rd8) ← 0		2								—	—	—	—	—	—	2
BCLR #xx:3, @Rd	B	(#xx:3 of @Rd16) ← 0			4							—	—	—	—	—	—	8
BCLR #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 0						4				—	—	—	—	—	—	8
BCLR Rn, Rd	B	(Rn8 of Rd8) ← 0		2								—	—	—	—	—	—	2
BCLR Rn, @Rd	B	(Rn8 of @Rd16) ← 0			4							—	—	—	—	—	—	8
BCLR Rn, @aa:8	B	(Rn8 of @aa:8) ← 0						4				—	—	—	—	—	—	8
BNOT #xx:3, Rd	B	(#xx:3 of Rd8) ← ($\overline{\#xx:3}$ of $\overline{Rd8}$)		2								—	—	—	—	—	—	2
BNOT #xx:3, @Rd	B	(#xx:3 of @Rd16) ← ($\overline{\#xx:3}$ of $\overline{@Rd16}$)			4							—	—	—	—	—	—	8
BNOT #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← ($\overline{\#xx:3}$ of $\overline{@aa:8}$)						4				—	—	—	—	—	—	8
BNOT Rn, Rd	B	(Rn8 of Rd8) ← ($\overline{Rn8}$ of $\overline{Rd8}$)		2								—	—	—	—	—	—	2
BNOT Rn, @Rd	B	(Rn8 of @Rd16) ← ($\overline{Rn8}$ of $\overline{@Rd16}$)			4							—	—	—	—	—	—	8
BNOT Rn, @aa:8	B	(Rn8 of @aa:8) ← ($\overline{Rn8}$ of $\overline{@aa:8}$)						4				—	—	—	—	—	—	8

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16,Rn)	@–Rn/@Rn+	@aa:8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
BTST #xx:3, Rd	B	(#xx:3 of Rd8) → Z		2								—	—	—	Δ	—	—	2
BTST #xx:3, @Rd	B	(#xx:3 of @Rd16) → Z			4							—	—	—	Δ	—	—	6
BTST #xx:3, @aa:8	B	(#xx:3 of @aa:8) → Z						4				—	—	—	Δ	—	—	6
BTST Rn, Rd	B	(Rn8 of Rd8) → Z		2								—	—	—	Δ	—	—	2
BTST Rn, @Rd	B	(Rn8 of @Rd16) → Z			4							—	—	—	Δ	—	—	6
BTST Rn, @aa:8	B	(Rn8 of @aa:8) → Z						4				—	—	—	Δ	—	—	6
BLD #xx:3, Rd	B	(#xx:3 of Rd8) → C		2								—	—	—	—	—	Δ	2
BLD #xx:3, @Rd	B	(#xx:3 of @Rd16) → C			4							—	—	—	—	—	Δ	6
BLD #xx:3, @aa:8	B	(#xx:3 of @aa:8) → C						4				—	—	—	—	—	Δ	6
BILD #xx:3, Rd	B	(#xx:3 of Rd8) → C		2								—	—	—	—	—	Δ	2
BILD #xx:3, @Rd	B	(#xx:3 of @Rd16) → C			4							—	—	—	—	—	Δ	6
BILD #xx:3, @aa:8	B	(#xx:3 of @aa:8) → C						4				—	—	—	—	—	Δ	6
BST #xx:3, Rd	B	C → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2
BST #xx:3, @Rd	B	C → (#xx:3 of @Rd16)			4							—	—	—	—	—	—	8
BST #xx:3, @aa:8	B	C → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8
BIST #xx:3, Rd	B	$\overline{C}$ → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2
BIST #xx:3, @Rd	B	$\overline{C}$ → (#xx:3 of @Rd16)			4							—	—	—	—	—	—	8
BIST #xx:3, @aa:8	B	$\overline{C}$ → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8
BAND #xx:3, Rd	B	C∧(#xx:3 of Rd8) → C		2								—	—	—	—	—	Δ	2
BAND #xx:3, @Rd	B	C∧(#xx:3 of @Rd16) → C			4							—	—	—	—	—	Δ	6
BAND #xx:3, @aa:8	B	C∧(#xx:3 of @aa:8) → C						4				—	—	—	—	—	Δ	6
BIAND #xx:3, Rd	B	C∧($\overline{\text{#xx:3 of Rd8}}$) → C		2								—	—	—	—	—	Δ	2
BIAND #xx:3, @Rd	B	C∧($\overline{\text{#xx:3 of @Rd16}}$) → C			4							—	—	—	—	—	Δ	6
BIAND #xx:3, @aa:8	B	C∧($\overline{\text{#xx:3 of @aa:8}}$) → C						4				—	—	—	—	—	Δ	6
BOR #xx:3, Rd	B	C∨(#xx:3 of Rd8) → C		2								—	—	—	—	—	Δ	2
BOR #xx:3, @Rd	B	C∨(#xx:3 of @Rd16) → C			4							—	—	—	—	—	Δ	6
BOR #xx:3, @aa:8	B	C∨(#xx:3 of @aa:8) → C						4				—	—	—	—	—	Δ	6
BIOR #xx:3, Rd	B	C∨($\overline{\text{#xx:3 of Rd8}}$) → C		2								—	—	—	—	—	Δ	2
BIOR #xx:3, @Rd	B	C∨($\overline{\text{#xx:3 of @Rd16}}$) → C			4							—	—	—	—	—	Δ	6

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Branching Condition	Addressing Mode/ Instruction Length								Condition Code						No. of States
				#xx: 8/16	Rn	@Rn	@(d:16,Rn)	@–Rn/@Rn+	@aa:8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
BIOR #xx:3, @aa:8	B	C∨(#xx:3 of @aa:8) → C							4				—	—	—	—	—	Δ	6
BXOR #xx:3, Rd	B	C⊕(#xx:3 of Rd8) → C			2								—	—	—	—	—	Δ	2
BXOR #xx:3, @Rd	B	C⊕(#xx:3 of @Rd16) → C				4							—	—	—	—	—	Δ	6
BXOR #xx:3, @aa:8	B	C⊕(#xx:3 of @aa:8) → C							4				—	—	—	—	—	Δ	6
BIXOR #xx:3, Rd	B	C⊕(#xx:3 of Rd8) → C			2								—	—	—	—	—	Δ	2
BIXOR #xx:3, @Rd	B	C⊕(#xx:3 of @Rd16) → C				4							—	—	—	—	—	Δ	6
BIXOR #xx:3, @aa:8	B	C⊕(#xx:3 of @aa:8) → C							4				—	—	—	—	—	Δ	6
BRA d:8 (BT d:8)	—	PC ← PC+d:8								2			—	—	—	—	—	—	4
BRN d:8 (BF d:8)	—	PC ← PC+2								2			—	—	—	—	—	—	4
BHI d:8	—	If condition	C ∨ Z = 0							2			—	—	—	—	—	—	4
BLS d:8	—	is true then	C ∨ Z = 1							2			—	—	—	—	—	—	4
BCC d:8 (BHS d:8)	—	PC ←	C = 0							2			—	—	—	—	—	—	4
BCS d:8 (BLO d:8)	—	PC + d:8	C = 1							2			—	—	—	—	—	—	4
BNE d:8	—	else next;	Z = 0							2			—	—	—	—	—	—	4
BEQ d:8	—		Z = 1							2			—	—	—	—	—	—	4
BVC d:8	—		V = 0							2			—	—	—	—	—	—	4
BVS d:8	—		V = 1							2			—	—	—	—	—	—	4
BPL d:8	—		N = 0							2			—	—	—	—	—	—	4
BMI d:8	—		N = 1							2			—	—	—	—	—	—	4
BGE d:8	—		N⊕V = 0							2			—	—	—	—	—	—	4
BLT d:8	—		N⊕V = 1							2			—	—	—	—	—	—	4
BGT d:8	—		Z ∨ (N⊕V) = 0							2			—	—	—	—	—	—	4
BLE d:8	—		Z ∨ (N⊕V) = 1							2			—	—	—	—	—	—	4
JMP @Rn	—	PC ← Rn16				2							—	—	—	—	—	—	4
JMP @aa:16	—	PC ← aa:16							4				—	—	—	—	—	—	6
JMP @@aa:8	—	PC ← @aa:8									2		—	—	—	—	—	—	8
BSR d:8	—	SP–2 → SP PC → @SP PC ← PC+d:8								2			—	—	—	—	—	—	6

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16,Rn)	@–Rn/@Rn+	@aa:8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
JSR @Rn	—	SP–2 → SP PC → @SP PC ← Rn16			2							—	—	—	—	—	—	6
JSR @aa:16	—	SP–2 → SP PC → @SP PC ← aa:16						4				—	—	—	—	—	—	8
JSR @@aa:8	—	SP–2 → SP PC → @SP PC ← @aa:8								2		—	—	—	—	—	—	8
RTS	—	PC ← @SP SP+2 → SP									2	—	—	—	—	—	—	8
RTE	—	CCR ← @SP SP+2 → SP PC ← @SP SP+2 → SP									2	Δ	Δ	Δ	Δ	Δ	Δ	10
SLEEP	—	Transition to power-down state.									2	—	—	—	—	—	—	2
LDC #xx:8, CCR	B	#xx:8 → CCR	2									Δ	Δ	Δ	Δ	Δ	Δ	2
LDC Rs, CCR	B	Rs8 → CCR		2								Δ	Δ	Δ	Δ	Δ	Δ	2
STC CCR, Rd	B	CCR → Rd8		2								—	—	—	—	—	—	2
ANDC #xx:8, CCR	B	CCR∧#xx:8 → CCR	2									Δ	Δ	Δ	Δ	Δ	Δ	2
ORC #xx:8, CCR	B	CCR∨#xx:8 → CCR	2									Δ	Δ	Δ	Δ	Δ	Δ	2
XORC #xx:8, CCR	B	CCR⊕#xx:8 → CCR	2									Δ	Δ	Δ	Δ	Δ	Δ	2
NOP	—	PC ← PC+2									2	—	—	—	—	—	—	2

Notes: The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory.

(1) Set to 1 when there is a carry or borrow from bit 11; otherwise cleared to 0.

(2) If the result is zero, the previous value of the flag is retained; otherwise the flag is cleared to 0.

(3) Set to 1 if decimal adjustment produces a carry; otherwise retains its previous value.

(4) The number of states required for execution is 4n+8 (n = value of R4L).

(5) These instructions are not supported by the H8/3534 and H8/3522.

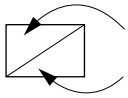
(6) Set to 1 if the divisor is negative; otherwise cleared to 0.

(7) Set to 1 if the divisor is 0; otherwise cleared to 0.

A.2 Operation Code Map

Table A-2 is a map of the operation codes contained in the first byte of the instruction code (bits 15 to 8 of the first instruction word).

Some pairs of instructions have identical first bytes. These instructions are differentiated by the first bit of the second byte (bit 7 of the first instruction word).



Instruction when first bit of byte 2 (bit 7 of first instruction word) is 0.
Instruction when first bit of byte 2 (bit 7 of first instruction word) is 1.

Table A-2 Operation Code Map

High \ Low	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SLEEP	STC	LDC	ORC	XORC	ANDC	LDC	ADD		INC	ADDS	MOV		ADDX	DAA
1	SHLL / SHAL	SHLR / SHAR	ROTXL / ROTL	ROTXR / ROTR	OR	XOR	AND	NOT / NEG	SUB		DEC	SUBS	CMP		SUBX	DAS
2	MOV															
3																
4	BRA*2	BRN*2	BHI	BLS	BCC*2	BCS*2	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU			RTS	BSR	RTE			JMP				JSR		
6	BSET	BNOT	BCLR	BTS				BST / BIST	MOV*1							
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD		MOV		EEPMOV	Bit manipulation instructions			
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

Notes: 1. The MOVFPE and MOVTPE instructions are identical to MOV instructions in the first byte and first bit of the second byte (bits 15 to 7 of the instruction word). The PUSH and POP instructions are identical in machine language to MOV instructions.
2. The BT, BF, BHS, and BLO instructions are identical in machine language to BRA, BRN, BCC, and BCS, respectively.

A.3 Number of States Required for Execution

The tables below can be used to calculate the number of states required for instruction execution. Table A-3 indicates the number of states required for each cycle (instruction fetch, branch address read, stack operation, byte data access, word data access, internal operation). Table A-4 indicates the number of cycles of each type occurring in each instruction. The total number of states required for execution of an instruction can be calculated from these two tables as follows:

Execution states = $I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$

Examples: Mode 1 (on-chip ROM disabled), stack located in external memory, 1 wait state inserted in external memory access.

1. BSET #0, @FFC7
   From table A-4: I = L = 2, J = K = M = N= 0
   From table A-3: $S_I = 8$, $S_L = 3$
   Number of states required for execution: $2 \times 8 + 2 \times 3 = 22$
2. JSR @@30
   From table A-4: I = 2, J = K = 1, L = M = N = 0
   From table A-3: $S_I = S_J = S_K = 8$
   Number of states required for execution: $2 \times 8 + 1 \times 8 + 1 \times 8 = 32$

Table A-3. Number of States Taken by Each Cycle in Instruction Execution

		Access Location		
Execution Status (Instruction Cycle)		**On-Chip Memory**	**On-Chip Supporting Module**	**External Device**
Instruction fetch	S_I	2	6	6 + 2m
Branch address read	S_J			
Stack operation	S_K			
Byte data access	S_L		3	3 + m
Word data access	S_M		6	6 + 2m
Internal operation	S_N	1	1	1

Notes: m: Number of wait states inserted in access to external device.

Table A-4 Number of Cycles in Each Instruction

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. 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 Rs, Rd	1					
ADDS	ADDS.W #1/2, Rd	1					
ADDX	ADDX.B #xx:8, Rd	1					
	ADDX.B Rs, Rd	1					
AND	AND.B #xx:8, Rd	1					
	AND.B Rs, Rd	1					
ANDC	ANDC #xx:8, CCR	1					
BAND	BAND #xx:3, Rd	1					
	BAND #xx:3, @Rd	2			1		
	BAND #xx:3, @aa:8	2			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					
	BGT d:8	2					
	BLE d:8	2					
BCLR	BCLR #xx:3, Rd	1					
	BCLR #xx:3, @Rd	2			2		
	BCLR #xx:3, @aa:8	2			2		
	BCLR Rn, Rd	1					
	BCLR Rn, @Rd	2			2		
	BCLR Rn, @aa:8	2			2		

Note: All values left blank are zero.

Table A-4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @Rd	2			1		
	BIAND #xx:3, @aa:8	2			1		
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @Rd	2			1		
	BILD #xx:3, @aa:8	2			1		
BIOR	BIOR #xx:3, Rd	1					
	BIOR #xx:3, @Rd	2			1		
	BIOR #xx:3, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @Rd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	1					
	BIXOR #xx:3, @Rd	2			1		
	BIXOR #xx:3, @aa:8	2			1		
BLD	BLD #xx:3, Rd	1					
	BLD #xx:3, @Rd	2			1		
	BLD #xx:3, @aa:8	2			1		
BNOT	BNOT #xx:3, Rd	1					
	BNOT #xx:3, @Rd	2			2		
	BNOT #xx:3, @aa:8	2			2		
	BNOT Rn, Rd	1					
	BNOT Rn, @Rd	2			2		
	BNOT Rn, @aa:8	2			2		
BOR	BOR #xx:3, Rd	1					
	BOR #xx:3, @Rd	2			1		
	BOR #xx:3, @aa:8	2			1		
BSET	BSET #xx:3, Rd	1					
	BSET #xx:3, @Rd	2			2		
	BSET #xx:3, @aa:8	2			2		
	BSET Rn, Rd	1					
	BSET Rn, @Rd	2			2		
	BSET Rn, @aa:8	2			2		

Note: All values left blank are zero.

Table A-4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BSR	BSR d:8	2		1			
BST	BST #xx:3, Rd	1					
	BST #xx:3, @Rd	2			2		
	BST #xx:3, @aa:8	2			2		
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @Rd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @Rd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	BXOR #xx:3, @Rd	2			1		
	BXOR #xx:3, @aa:8	2			1		
CMP	CMP.B #xx:8, Rd	1					
	CMP.B Rs, Rd	1					
	CMP.W Rs, Rd	1					
DAA	DAA.B Rd	1					
DAS	DAS.B Rd	1					
DEC	DEC.B Rd	1					
DIVXU	DIVXU.B Rs, Rd	1					12
EEPMOV	EEPMOV	2			2n+2*		1
INC	INC.B Rd	1					
JMP	JMP @Rn	2					
	JMP @aa:16	2					2
	JMP @@aa:8	2	1				2
JSR	JSR @Rn	2		1			
	JSR @aa:16	2		1			2
	JSR @@aa:8	2	1	1			
LDC	LDC #xx:8, CCR	1					
	LDC Rs, CCR	1					
MOV	MOV.B #xx:8, Rd	1					
	MOV.B Rs, Rd	1					
	MOV.B @Rs, Rd	1			1		
	MOV.B @(d:16,Rs), Rd	2			1		

Notes: All values left blank are zero.

* n: Initial value in R4L. Source and destination are accessed n + 1 times each.

Table A-4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	I nternal Operation N
MOV	MOV.B @Rs+, Rd	1			1		2
	MOV.B @aa:8, Rd	1			1		
	MOV.B @aa:16, Rd	2			1		
	MOV.B Rs, @Rd	1			1		
	MOV.B Rs, @(d:16, Rd)	2			1		
	MOV.B Rs, @–Rd	1			1		2
	MOV.B Rs, @aa:8	1			1		
	MOV.B Rs, @aa:16	2			1		
	MOV.W #xx:16, Rd	2					
	MOV.W Rs, Rd	1					
	MOV.W @Rs, Rd	1				1	
	MOV.W @(d:16, Rs), Rd	2				1	
	MOV.W @Rs+, Rd	1				1	2
	MOV.W @aa:16, Rd	2				1	
	MOV.W Rs, @Rd	1				1	
	MOV.W Rs, @(d:16, Rd)	2				1	
	MOV.W Rs, @–Rd	1				1	2
	MOV.W Rs, @aa:16	2				1	
MOVFPE	MOVFPE @aa:16, Rd	Not supported					
MOVTPE	MOVTPERs, @aa:16	Not supported					
MULXU	MULXU.BRs, Rd	1					12
NEG	NEG.B Rd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
OR	OR.B #xx:8, Rd	1					
	OR.B Rs, Rd	1					
ORC	ORC #xx:8, CCR	1					
POP	POP Rd	1			1		2
PUSH	PUSH Rd	1			1		2
ROTL	ROTL.B Rd	1					
ROTR	ROTR.B Rd	1					
ROTXL	ROTXL.B Rd	1					
ROTXR	ROTXR.B Rd	1					
RTE	RTE	2		2			2
RTS	RTS	2		1			2

Note: All values left blank are zero.

Table A-4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
SHAL	SHAL.B Rd	1					
SHAR	SHAR.B Rd	1					
SHLL	SHLL.B Rd	1					
SHLR	SHLR.B Rd	1					
SLEEP	SLEEP	1					
STC	STC CCR, Rd	1					
SUB	SUB.B Rs, Rd	1					
	SUB.W Rs, Rd	1					
SUBS	SUBS.W #1/2, Rd	1					
SUBX	SUBX.B #xx:8, Rd	1					
	SUBX.B Rs, Rd	1					
XOR	XOR.B #xx:8, Rd	1					
	XOR.B Rs, Rd	1					
XORC	XORC #xx:8, CCR	1					

Note: All values left blank are zero.

Appendix B Internal I/O Register

B.1 Addresses

B.1.1 Addresses for H8/3534

Addr. (Last Byte)	Register Name	Bit Names: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'80										External addresses (in expanded modes)
H'81										
H'82										
H'83										
H'84										
H'85										
H'86										
H'87										
H'88	SMR	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI1
H'89	BRR									
H'8A	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'8B	TDR									
H'8C	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'8D	RDR									
H'8E	—	—	—	—	—	—	—	—	—	
H'8F	—	—	—	—	—	—	—	—	—	
H'90	TIER	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	FRT
H'91	TCSR	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA	
H'92	FRC (H)									
H'93	FRC (L)									
H'94	OCRA (H)									
	OCRB (H)									
H'95	OCRA (L)									
	OCRB (L)									
H'96	TCR	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0	
H'97	TOCR	—	—	—	OCRS	OEA	OEB	OLVLA	OLVLB	
H'98	ICRA (H)									
H'99	ICRA (L)									
H'9A	ICRB (H)									
H'9B	ICRB (L)									
H'9C	ICRC (H)									
H'9D	ICRC (L)									
H'9E	ICRD (H)									
H'9F	ICRD (L)									

Notes: FRT: 16-bit free-running timer
SCI1: Serial communication interface 1

(Continued on next page)

(Continued from previous page)

		Bit Names								
Addr. (Last Byte)	**Register Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Module**
H'A0	TCR	OE	OS	—	—	—	CKS2	CKS1	CKS0	PWM0
H'A1	DTR									
H'A2	TCNT									
H'A3	—	—	—	—	—	—	—	—	—	
H'A4	TCR	OE	OS	—	—	—	CKS2	CKS1	CKS0	PWM1
H'A5	DTR									
H'A6	TCNT									
H'A7	—	—	—	—	—	—	—	—	—	
H'A8	TCSR/ TCNT	OVF	WT/$\overline{IT}$	TME	—	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT
H'A9	TCNT									
H'AA	—	—	—	—	—	—	—	—	—	
H'AB	—	—	—	—	—	—	—	—	—	
H'AC	P1PCR	$P1_7$PCR	$P1_6$PCR	$P1_5$PCR	$P1_4$PCR	$P1_3$PCR	$P1_2$PCR	$P1_1$PCR	$P1_0$PCR	Port 1
H'AD	P2PCR	$P2_7$PCR	$P2_6$PCR	$P2_5$PCR	$P2_4$PCR	$P2_3$PCR	$P2_2$PCR	$P2_1$PCR	$P2_0$PCR	Port 2
H'AE	P3PCR	$P3_7$PCR	$P3_6$PCR	$P3_5$PCR	$P3_4$PCR	$P3_3$PCR	$P3_2$PCR	$P3_1$PCR	$P3_0$PCR	Port 3
H'AF	—	—	—	—	—	—	—	—	—	—
H'B0	P1DDR	$P1_7$DDR	$P1_6$DDR	$P1_5$DDR	$P1_4$DDR	$P1_3$DDR	$P1_2$DDR	$P1_1$DDR	$P1_0$DDR	Port 1
H'B1	P2DDR	$P2_7$DDR	$P2_6$DDR	$P2_5$DDR	$P2_4$DDR	$P2_3$DDR	$P2_2$DDR	$P2_1$DDR	$P2_0$DDR	Port 2
H'B2	P1DR	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$	Port 1
H'B3	P2DR	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$	Port 2
H'B4	P3DDR	$P3_7$DDR	$P3_6$DDR	$P3_5$DDR	$P3_4$DDR	$P3_3$DDR	$P3_2$DDR	$P3_1$DDR	$P3_0$DDR	Port 3
H'B5	P4DDR	$P4_7$DDR	$P4_6$DDR	$P4_5$DDR	$P4_4$DDR	$P4_3$DDR	$P4_2$DDR	$P4_1$DDR	$P4_0$DDR	Port 4
H'B6	P3DR	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$	Port 3
H'B7	P4DR	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$	Port 4
H'B8	P5DDR	—	—	—	—	—	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR	Port 5
H'B9	P6DDR	$P6_7$DDR	$P6_6$DDR	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	$P6_2$DDR	$P6_1$DDR	$P6_0$DDR	Port 6
H'BA	P5DR	—	—	—	—	—	$P5_2$	$P5_1$	$P5_0$	Port 5
H'BB	P6DR	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$	Port 6

Notes: PWM0: Pulse-width modulation timer channel 0
PWM1: Pulse-width modulation timer channel 1
WDT: Watchdog timer

(Continued on next page)

(Continued from preceding page)

Addr. (Last Byte)	Register Name	Bit Names: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'BC	—	—	—	—	—	—	—	—	—	
H'BD	P8DDR	—	$P8_6$DDR	$P8_5$DDR	$P8_4$DDR	$P8_3$DDR	$P8_2$DDR	$P8_1$DDR	$P8_0$DDR	Port 8
H'BE	P7PIN	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$	Port 7
H'BF	P8DR	—	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$	Port 8
H'C0	P9DDR	$P9_7$DDR	$P9_6$DDR	$P9_5$DDR	$P9_4$DDR	$P9_3$DDR	$P9_2$DDR	$P9_1$DDR	$P9_0$DDR	Port 9
H'C1	P9DR	$P9_7$	$P9_6$	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$	
H'C2	WSCR	(RAMS)	(RAMO)	CKDBL	—	WMS1	WMS0	WC1	WC0	
H'C3	STCR	(IICS)	(IICD)	(IICX)	(IICE)	(STAC)	MPE	ICKS1	ICKS0	—
H'C4	SYSCR	SSBY	STS2	STS1	STS0	XRST	NMIEG	(HIE)	RAME	
H'C5	MDCR	—	—	—	—	—	—	MDS1	MDS0	
H'C6	ISCR	IRQ7SC	IRQ6SC	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC	
H'C7	IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
H'C8	TCR	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR0
H'C9	TCSR	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
H'CA	TCORA									
H'CB	TCORB									
H'CC	TCNT									
H'CD	—	—	—	—	—	—	—	—	—	
H'CE	—	—	—	—	—	—	—	—	—	
H'CF	—	—	—	—	—	—	—	—	—	
H'D0	TCR	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR1
H'D1	TCSR	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
H'D2	TCORA									
H'D3	TCORB									
H'D4	TCNT									
H'D5	—	—	—	—	—	—	—	—	—	
H'D6	—	—	—	—	—	—	—	—	—	
H'D7	—	—	—	—	—	—	—	—	—	

Notes: TMR0: 8-bit timer channel 0
TMR1: 8-bit timer channel 1

(Continued on next page)

(Continued from preceding page)

Addr. (Last Byte)	Register Name	Bit Names: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'D8	SMR	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI0
H'D9	BRR									
H'DA	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'DB	TDR									
H'DC	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'DD	RDR									
H'DE	—	—	—	—	—	—	—	—	—	
H'DF	—	—	—	—	—	—	—	—	—	
H'E0	ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
H'E1	ADDRAL	AD1	AD0	—	—	—	—	—	—	
H'E2	ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'E3	ADDRBL	AD1	AD0	—	—	—	—	—	—	
H'E4	ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'E5	ADDRCL	AD1	AD0	—	—	—	—	—	—	
H'E6	ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'E7	ADDRDL	AD1	AD0	—	—	—	—	—	—	
H'E8	ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
H'E9	ADCR	TRGE	—	—	—	—	—	—	—	
H'EA	—	—	—	—	—	—	—	—	—	
H'EB	—	—	—	—	—	—	—	—	—	
H'EC	—	—	—	—	—	—	—	—	—	—
H'ED	—	—	—	—	—	—	—	—	—	
H'EE	—	—	—	—	—	—	—	—	—	
H'EF	—	—	—	—	—	—	—	—	—	

Notes: SCI0: Serial communication interface 0
A/D: Analog-to-digital converter

(Continued on next page)

(Continued from preceding page)

		Bit Names								
Addr. (Last Byte)	**Register Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Module**
H'F0										
H'F1	KMIMR	KMIMR7	KMIMR6	KMIMR5	KMIMR4	KMIMR3	KMIMR2	KMIMR1	KMIMR0	
H'F2	KMPCR	KM_7PCR	KM_6PCR	KM_5PCR	KM_4PCR	KM_3PCR	KM_2PCR	KM_1PCR	KM_0PCR	Port6
H'F3										
H'F4										
H'F5										
H'F6										
H'F7										
H'F8										
H'F9										
H'FA										
H'FB										
H'FC										
H'FD										
H'FE										
H'FF										

B.1.2 Addresses for H8/3522

Addr. (Last Byte)	Register Name	Bit Names Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'80										External addresses (in expanded modes)
H'81										
H'82										
H'83										
H'84										
H'85										
H'86										
H'87										
H'88	—	—	—	—	—	—	—	—	—	—
H'89	—	—	—	—	—	—	—	—	—	
H'8A	—	—	—	—	—	—	—	—	—	
H'8B	—	—	—	—	—	—	—	—	—	
H'8C	—	—	—	—	—	—	—	—	—	
H'8D	—	—	—	—	—	—	—	—	—	
H'8E	—	—	—	—	—	—	—	—	—	
H'8F	—	—	—	—	—	—	—	—	—	
H'90	TIER	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	FRT
H'91	TCSR	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA	
H'92	FRCH									
H'93	FRCL									
H'94	OCRAH									
	OCRBH									
H'95	OCRAL									
	OCRBL									
H'96	TCR	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0	
H'97	TOCR	—	—	—	OCRS	OEA	OEB	OLVLA	OLVLB	
H'98	ICRAH									
H'99	ICRAL									
H'9A	ICRBH									
H'9B	ICRBL									
H'9C	ICRCH									
H'9D	ICRCL									
H'9E	ICRDH									
H'9F	ICRDL									

Note: FRT: 16-bit free-running timer

(Continued on next page)

(Continued from preceding page)

Addr. (Last Byte)	Register Name	Bit Names: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'A0	—	—	—	—	—	—	—	—	—	
H'A1	—	—	—	—	—	—	—	—	—	
H'A2	—	—	—	—	—	—	—	—	—	
H'A3	—	—	—	—	—	—	—	—	—	
H'A4	—	—	—	—	—	—	—	—	—	
H'A5	—	—	—	—	—	—	—	—	—	
H'A6	—	—	—	—	—	—	—	—	—	
H'A7	—	—	—	—	—	—	—	—	—	
H'A8	TCSR/ TCNT	OVF	WT/$\overline{IT}$	TME	—	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT
H'A9	TCNT									
H'AA	—	—	—	—	—	—	—	—	—	
H'AB	—	—	—	—	—	—	—	—	—	
H'AC	P1PCR	$P1_7$PCR	$P1_6$PCR	$P1_5$PCR	$P1_4$PCR	$P1_3$PCR	$P1_2$PCR	$P1_1$PCR	$P1_0$PCR	Port 1
H'AD	P2PCR	$P2_7$PCR	$P2_6$PCR	$P2_5$PCR	$P2_4$PCR	$P2_3$PCR	$P2_2$PCR	$P2_1$PCR	$P2_0$PCR	Port 2
H'AE	P3PCR	$P3_7$PCR	$P3_6$PCR	$P3_5$PCR	$P3_4$PCR	$P3_3$PCR	$P3_2$PCR	$P3_1$PCR	$P3_0$PCR	Port 3
H'AF	—	—	—	—	—	—	—	—	—	—
H'B0	P1DDR	$P1_7$DDR	$P1_6$DDR	$P1_5$DDR	$P1_4$DDR	$P1_3$DDR	$P1_2$DDR	$P1_1$DDR	$P1_0$DDR	Port 1
H'B1	P2DDR	$P2_7$DDR	$P2_6$DDR	$P2_5$DDR	$P2_4$DDR	$P2_3$DDR	$P2_2$DDR	$P2_1$DDR	$P2_0$DDR	Port 2
H'B2	P1DR	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$	Port 1
H'B3	P2DR	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$	Port 2
H'B4	P3DDR	$P3_7$DDR	$P3_6$DDR	$P3_5$DDR	$P3_4$DDR	$P3_3$DDR	$P3_2$DDR	$P3_1$DDR	$P3_0$DDR	Port 3
H'B5	P4DDR	$P4_7$DDR	$P4_6$DDR	$P4_5$DDR	$P4_4$DDR	$P4_3$DDR	$P4_2$DDR	$P4_1$DDR	$P4_0$DDR	Port 4
H'B6	P3DR	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$	Port 3
H'B7	P4DR	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$	Port 4
H'B8	P5DDR	—	—	—	—	—	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR	Port 5
H'B9	P6DDR	$P6_7$DDR	$P6_6$DDR	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	$P6_2$DDR	$P6_1$DDR	$P6_0$DDR	Port 6
H'BA	P5DR	—	—	—	—	—	$P5_2$	$P5_1$	$P5_0$	Port 5
H'BB	P6DR	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$	Port 6
H'BC	—	—	—	—	—	—	—	—	—	—
H'BD	—	—	—	—	—	—	—	—	—	—
H'BE	P7PIN	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$	Port 7
H'BF	—	—	—	—	—	—	—	—	—	—

Note: WDT: Watchdog timer

(Continued on next page)

(Continued from preceding page)

Addr. (Last Byte)	Register Name	Bit Names: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'C0	—	—	—	—	—	—	—	—	—	—
H'C1	—	—	—	—	—	—	—	—	—	—
H'C2	WSCR	—	—	CKDBL	—	WMS1	WMS0	WC1	WC0	
H'C3	STCR	—	—	—	—	—	MPE	ICKS1	ICKS0	
H'C4	SYSCR	SSBY	STS2	STS1	STS0	XRST	NMIEG	—	RAME	
H'C5	MDCR	—	—	—	—	—	—	MDS1	MDS0	
H'C6	ISCR	—	—	—	—	—	IRQ2SC	IRQ1SC	IRQ0SC	
H'C7	IER	—	—	—	—	—	IRQ2E	IRQ1E	IRQ0E	
H'C8	TCR	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR0
H'C9	TCSR	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
H'CA	TCORA									
H'CB	TCORB									
H'CC	TCNT									
H'CD	—	—	—	—	—	—	—	—	—	
H'CE	—	—	—	—	—	—	—	—	—	
H'CF	—	—	—	—	—	—	—	—	—	
H'D0	TCR	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR1
H'D1	TCSR	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
H'D2	TCORA									
H'D3	TCORB									
H'D4	TCNT									
H'D5	—	—	—	—	—	—	—	—	—	
H'D6	—	—	—	—	—	—	—	—	—	
H'D7	—	—	—	—	—	—	—	—	—	
H'D8	SMR	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI
H'D9	BRR									
H'DA	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'DB	TDR									
H'DC	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'DD	RDR									
H'DE	—	—	—	—	—	—	—	—	—	
H'DF	—	—	—	—	—	—	—	—	—	

Notes: TMR0: 8-bit timer channel 0
TMR1: 8-bit timer channel 1
SCI: Serial communication interface

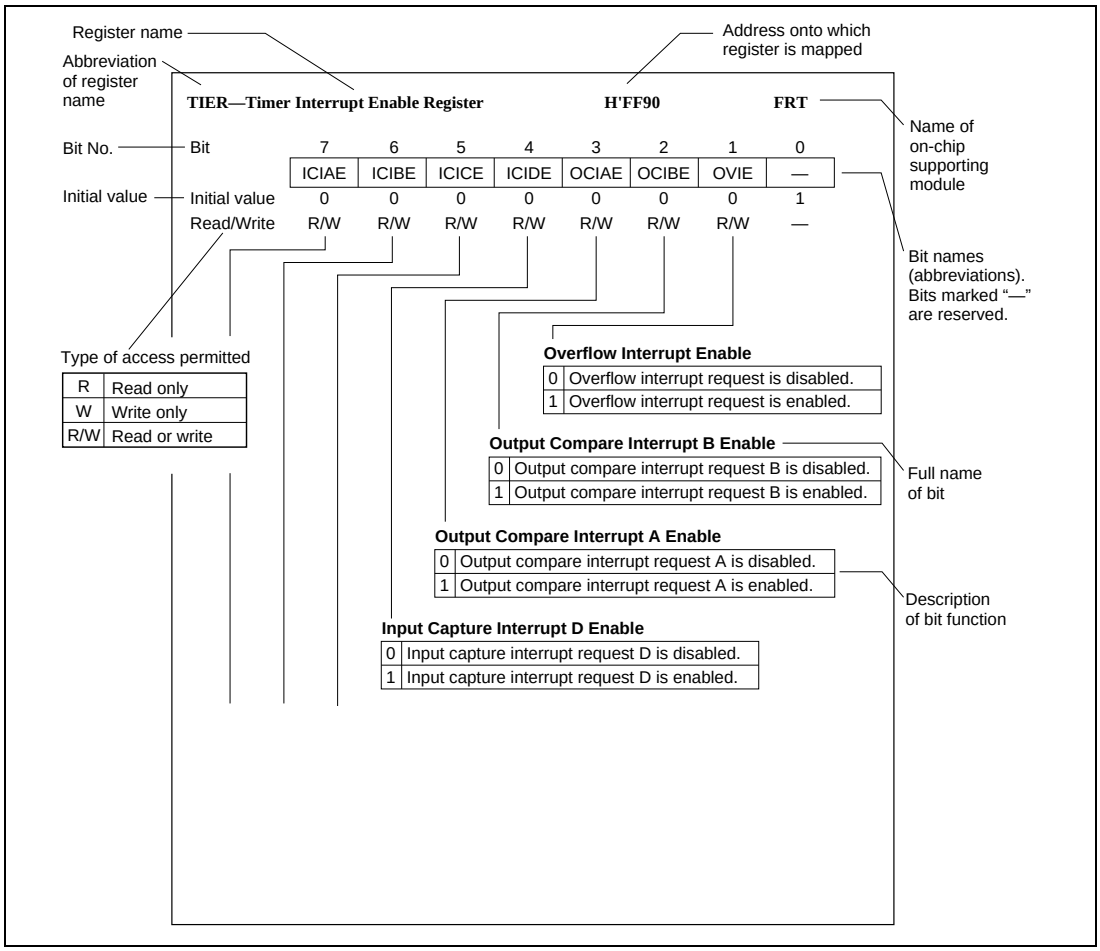
(Continued on next page)

(Continued from preceding page)

Addr. (Last Byte)	Register Name	Bit Names: Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module
H'E0	ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
H'E1	ADDRAL	AD1	AD0	—	—	—	—	—	—	
H'E2	ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'E3	ADDRBL	AD1	AD0	—	—	—	—	—	—	
H'E4	ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'E5	ADDRCL	AD1	AD0	—	—	—	—	—	—	
H'E6	ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'E7	ADDRDL	AD1	AD0	—	—	—	—	—	—	
H'E8	ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
H'E9	ADCR	TRGE	—	—	—	—	—	—	—	
H'EA	—	—	—	—	—	—	—	—	—	
H'EB	—	—	—	—	—	—	—	—	—	
H'EC	—	—	—	—	—	—	—	—	—	—
H'ED	—	—	—	—	—	—	—	—	—	
H'EE	—	—	—	—	—	—	—	—	—	
H'EF	—	—	—	—	—	—	—	—	—	
H'F0	—	—	—	—	—	—	—	—	—	—
H'F1	—	—	—	—	—	—	—	—	—	
H'F2	—	—	—	—	—	—	—	—	—	
H'F3	—	—	—	—	—	—	—	—	—	
H'F4	—	—	—	—	—	—	—	—	—	
H'F5	—	—	—	—	—	—	—	—	—	
H'F6	—	—	—	—	—	—	—	—	—	
H'F7	—	—	—	—	—	—	—	—	—	
H'F8	—	—	—	—	—	—	—	—	—	
H'F9	—	—	—	—	—	—	—	—	—	
H'FA	—	—	—	—	—	—	—	—	—	
H'FB	—	—	—	—	—	—	—	—	—	
H'FC	—	—	—	—	—	—	—	—	—	
H'FD	—	—	—	—	—	—	—	—	—	
H'FE	—	—	—	—	—	—	—	—	—	
H'FF	—	—	—	—	—	—	—	—	—	

Note: A/D: Analog-to-digital converter

B.2 Function


Register name
Abbreviation of register name
Address onto which register is mapped
TIER—Timer Interrupt Enable Register
H'FF90
FRT
Name of on-chip supporting module
Bit No.
Bit
Initial value
Initial value
Read/Write
7 6 5 4 3 2 1 0
ICIAE ICIBE ICICE ICIDE OCIAE OCIBE OVIE —
0 0 0 0 0 0 0 1
R/W R/W R/W R/W R/W R/W R/W —
Bit names (abbreviations). Bits marked "—" are reserved.
Type of access permitted
R Read only
W Write only
R/W Read or write
Overflow Interrupt Enable
0 Overflow interrupt request is disabled.
1 Overflow interrupt request is enabled.
Output Compare Interrupt B Enable
0 Output compare interrupt request B is disabled.
1 Output compare interrupt request B is enabled.
Full name of bit
Output Compare Interrupt A Enable
0 Output compare interrupt request A is disabled.
1 Output compare interrupt request A is enabled.
Description of bit function
Input Capture Interrupt D Enable
0 Input capture interrupt request D is disabled.
1 Input capture interrupt request D is enabled.

SMR—Serial Mode Register [H8/3534 only] H'FF88 SCI1

Bit	7	6	5	4	3	2	1	0
	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	ø clock
0	1	$ø_P$/4 clock
1	0	$ø_P$/16 clock
1	1	$ø_P$/64 clock

Multiprocessor Mode

0	Multiprocessor function disabled
1	Multiprocessor function enabled

Stop Bit Length

0	One stop bit
1	Two stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Transmit:	No parity bit added.
	Receive:	Parity bit not checked.
1	Transmit:	Parity bit added.
	Receive:	Parity bit checked.

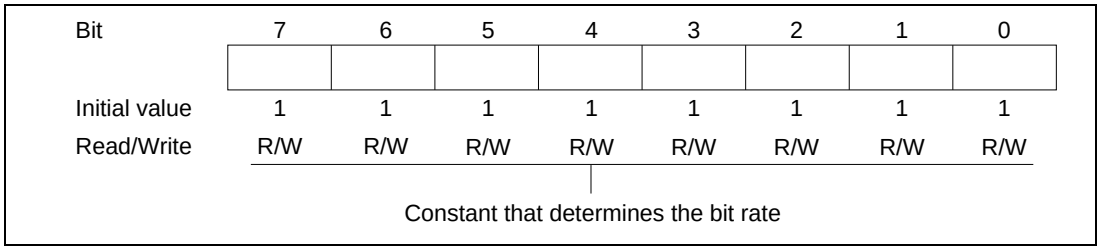
Character Length

0	8-bit per character
1	7-bit per character

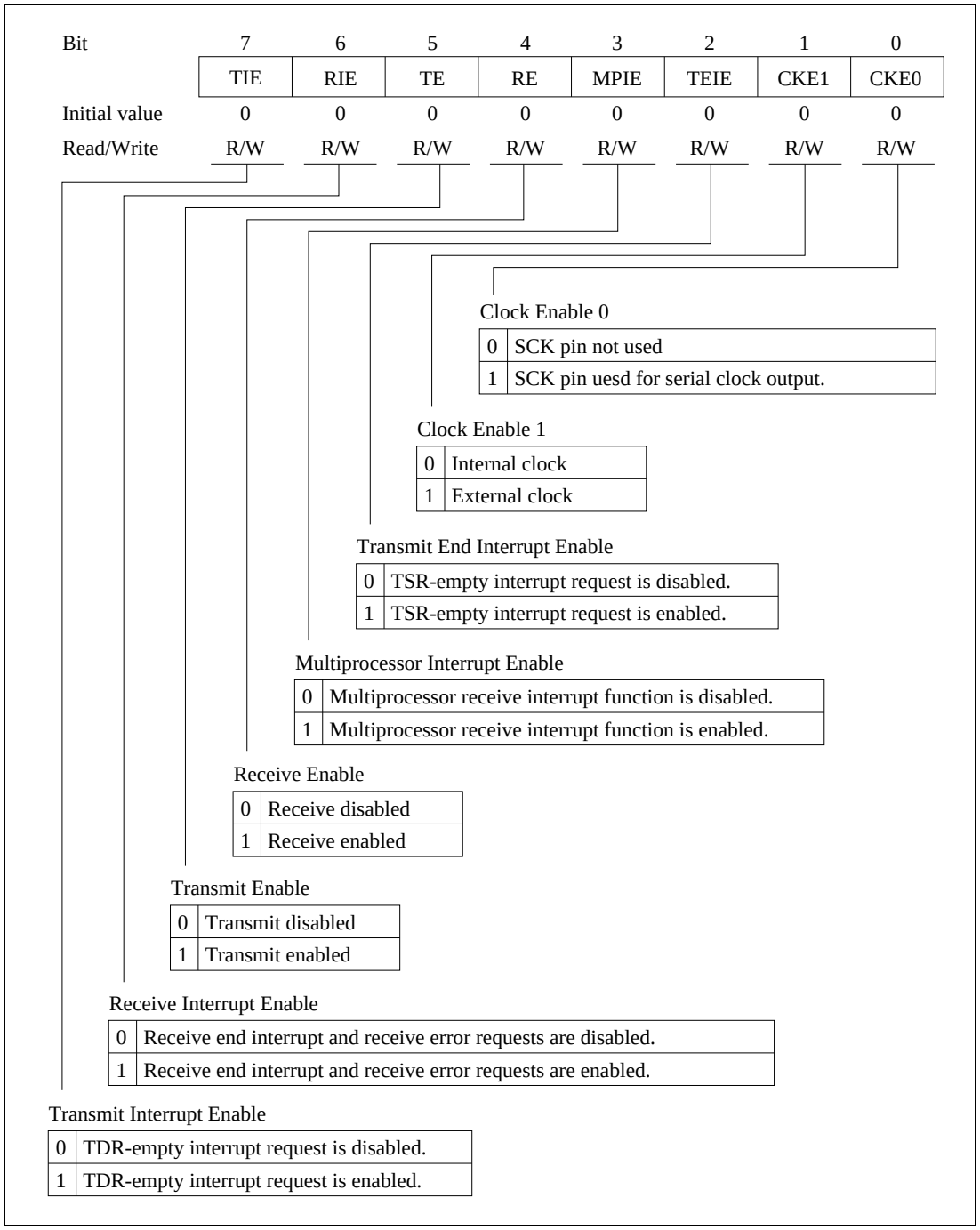
Communication Mode

0	Asynchronous
1	Synchronous

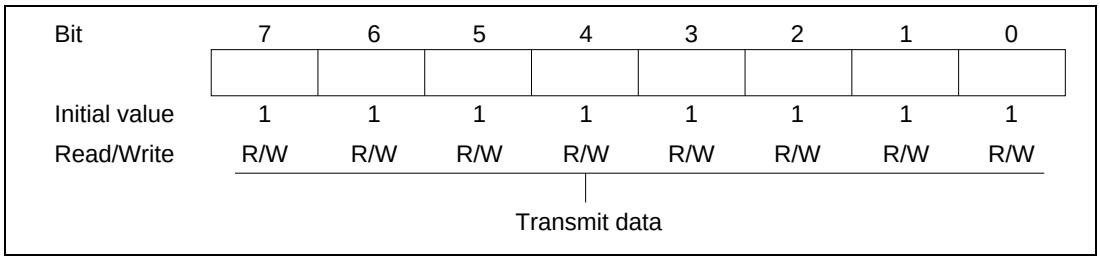
BRR—Bit Rate Register [H8/3534 only] H'FF89 SCI1



SCR—Serial Control Register [H8/3534 only] H'FF8A SCI1



TDR—Transmit Data Register [H8/3534 only] H'FF8B SCI1



SSR—Serial Status Register [H8/3534 only] H'FF8C SCI1

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	Multiprocessor bit = 0 in transmit data. (Initial value)
1	Multiprocessor bit = 1 in transmit data.

Multiprocessor Bit

0	Multiprocessor bit = 0 in receive data. (Initial value)
1	Multiprocessor bit = 1 in receive data.

Transmit End

0	Cleared by reading TDRE = 1, then writing 0 in TDRE.
1	Set to 1 when TE = 0, or when TDRE = 1 at the end of character transmission. (Initial value)

Parity Error

0	Cleared by reading PER = 1, then writing 0 in PER. (Initial value)
1	Set when a parity error occurs (parity of receive data does not match parity selected by O/E bit in SMR).

Framing Error

0	Cleared by reading FER = 1, then writing 0 in FER. (Initial value)
1	Set when a framing error occurs (stop bit is 0).

Overrun Error

0	Cleared by reading ORER = 1, then writing 0 in ORER. (Initial value)
1	Set when an overrun error occurs (next data is completely received while RDRF bit is set to 1).

Receive Data Register Full

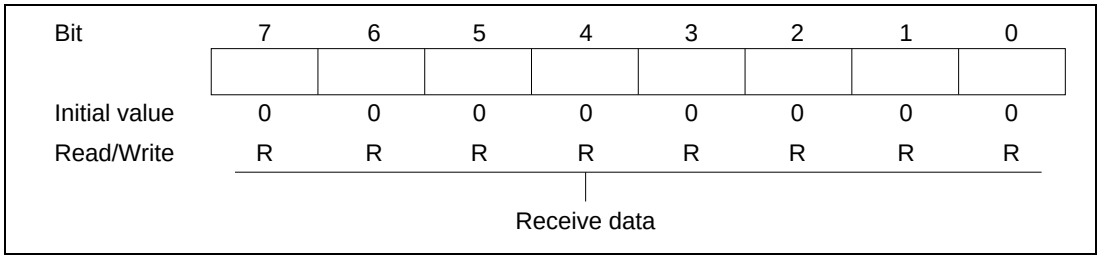
0	Cleared by reading RDRF = 1, then writing 0 in RDRF. (Initial value)
1	Set when one character is received normally and transferred from RSR to RDR.

Transmit Data Register Empty

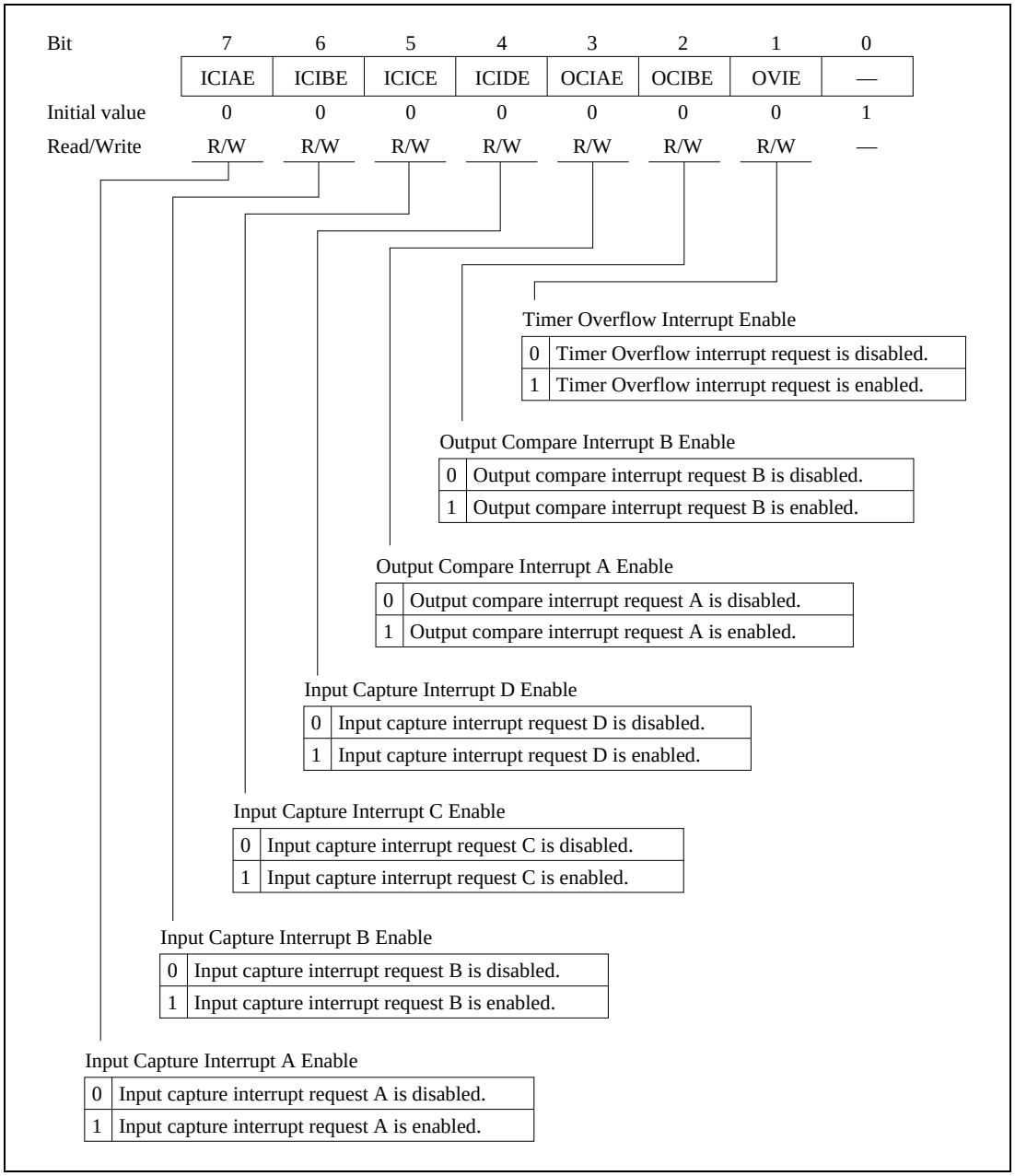
0	Cleared by reading TDRE = 1, then writing 0 in TDRE.
1	Set when: (Initial value) 1. Data is transferred from TDR to TSR. 2. TE is cleared while TDRE = 0.

Note: * Software can write a 0 in bits 7 to 3 to clear the flags, but cannot write a 1 in these bits.

RDR—Receive Data Register [H8/3534 only] H'FF8D SCI1



TIER—Timer Interrupt Enable Register H'FF90 FRT



TCSR—Timer Control/Status Register H'FF91 FRT

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

Counter Clear A

0	FRC count is not cleared.
1	FRC count is cleared by compare-match A.

Timer Overflow Flag

0	Cleared by reading OVF = 1, then writing 0 in OVF.
1	Set when FRC changes from H'FFFF to H'0000.

Output Compare Flag B

0	Cleared by reading OCFB = 1, then writing 0 in OCFB.
1	Set when FRC = OCRB.

Output Compare Flag A

0	Cleared by reading OCFA = 1, then writing 0 in OCFA.
1	Set when FRC = OCRA.

Input Capture Flag D

0	Cleared by reading ICFD = 1, then writing 0 in ICFD.
1	Set when an input capture signal is received.

Input Capture Flag C

0	Cleared by reading ICFC = 1, then writing 0 in ICFC.
1	Set when an input capture signal is received.

Input Capture Flag B

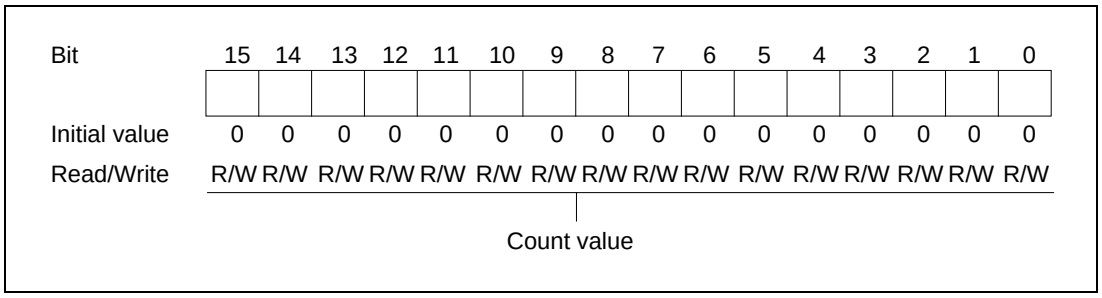
0	Cleared by reading ICFB = 1, then writing 0 in ICFB.
1	Set when FTIB input causes FRC to be copied to ICRB.

Input Capture Flag A

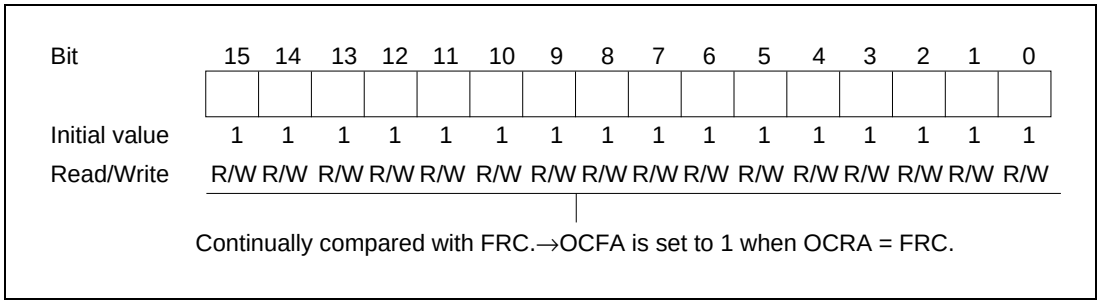
0	Cleared by reading ICFA = 1, then writing 0 in ICFA.
1	Set when FTIA input causes FRC to be copied to ICRA.

Note: * Software can write a 0 in bits 7 to 1 to clear the flags, but cannot write a 1 in these bits.

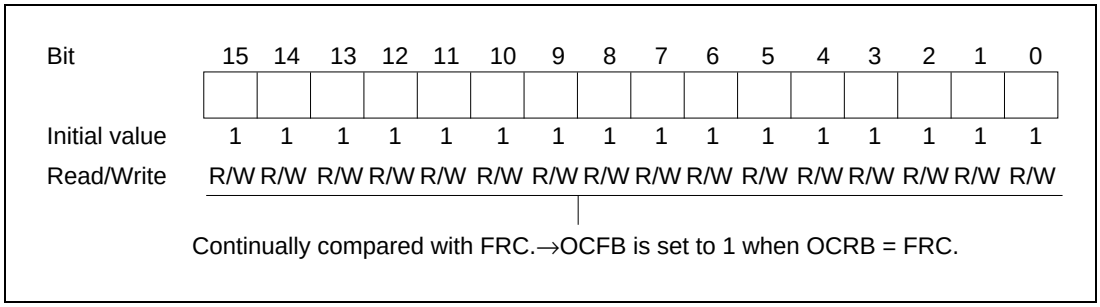
FRC (H and L)—Free-Running Counter H'FF92, H'FF93 FRT



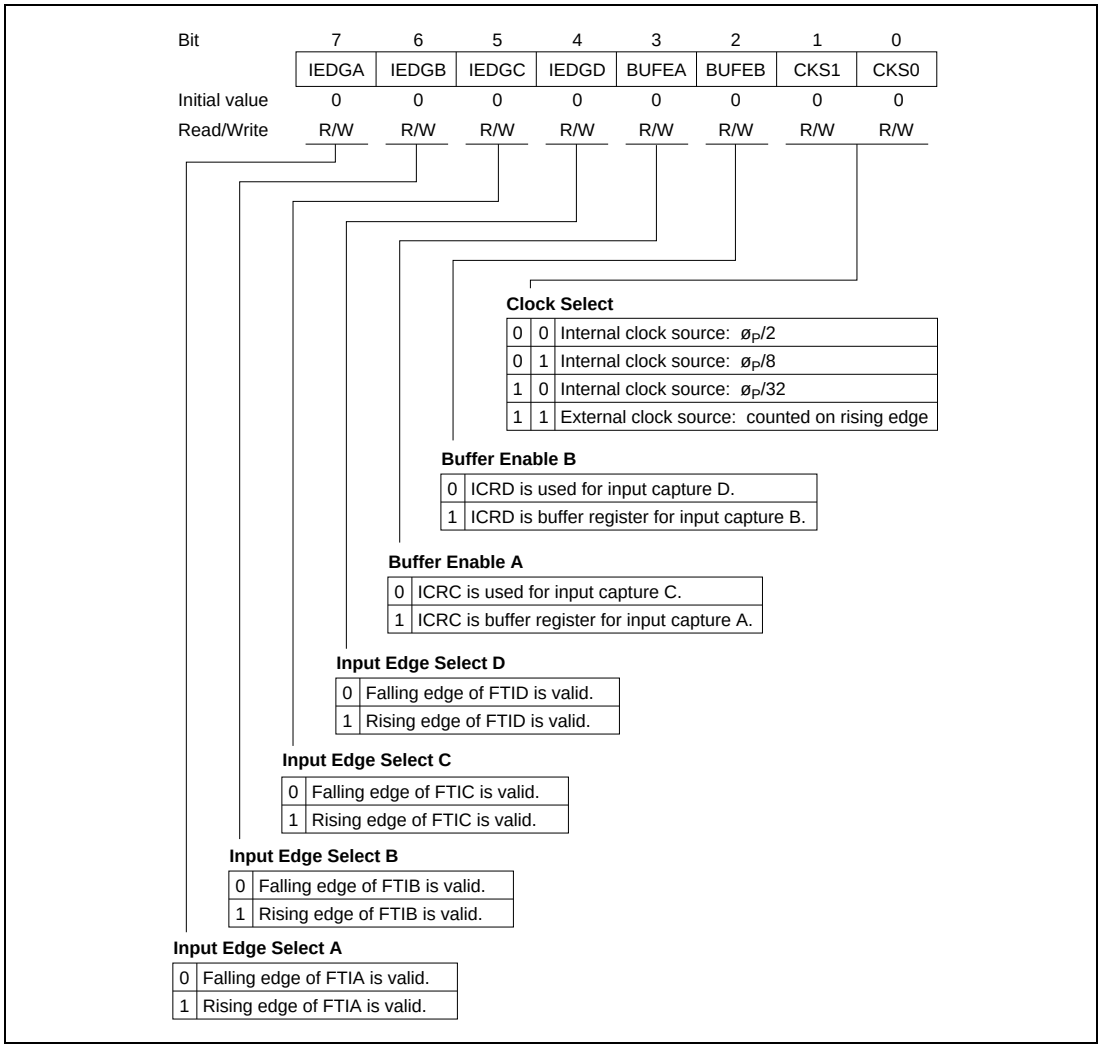
OCRA (H and L)—Output Compare Register A H'FF94, H'FF95 FRT



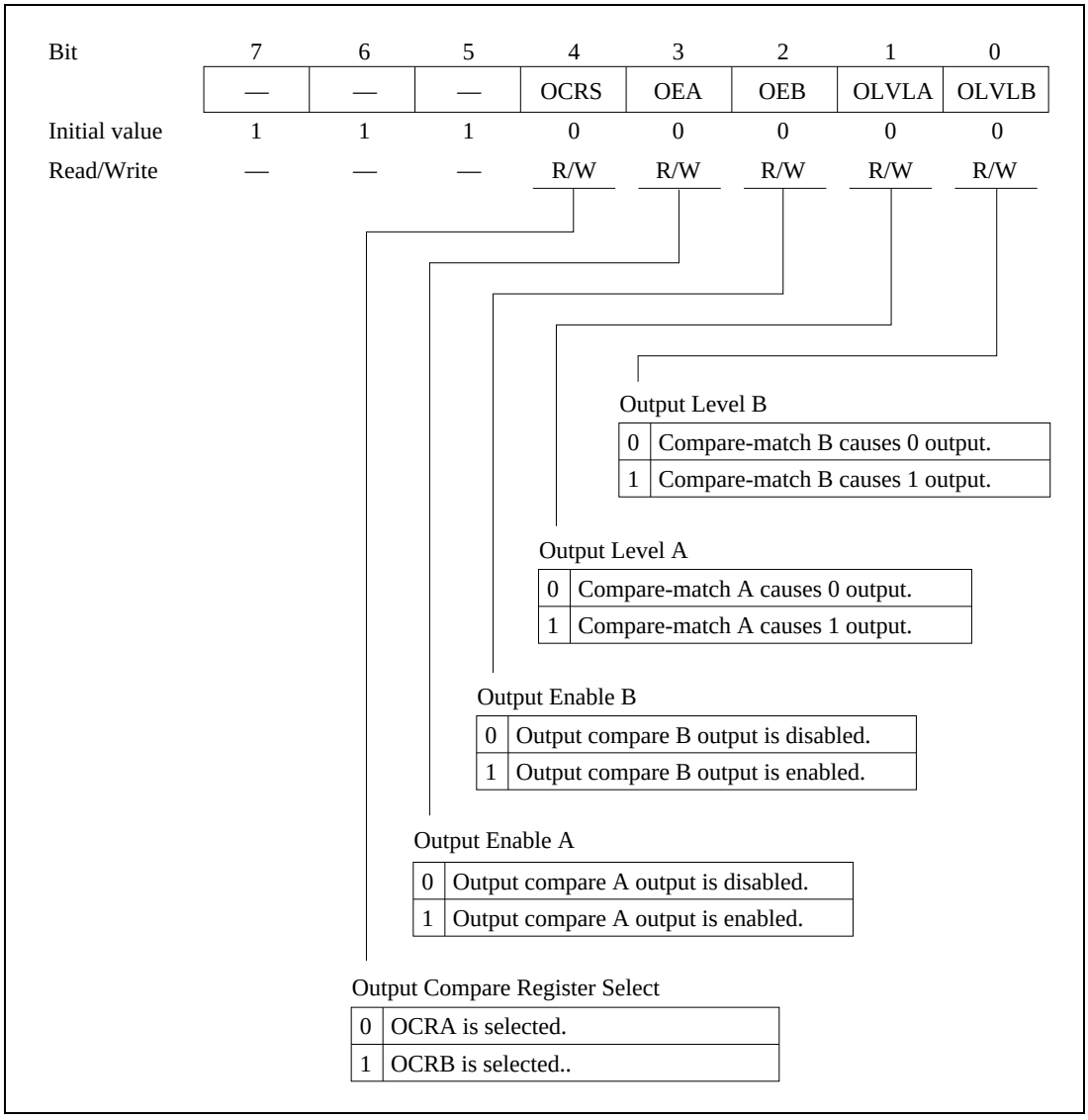
OCRB (H and L)—Output Compare Register B H'FF94, H'FF95 FRT



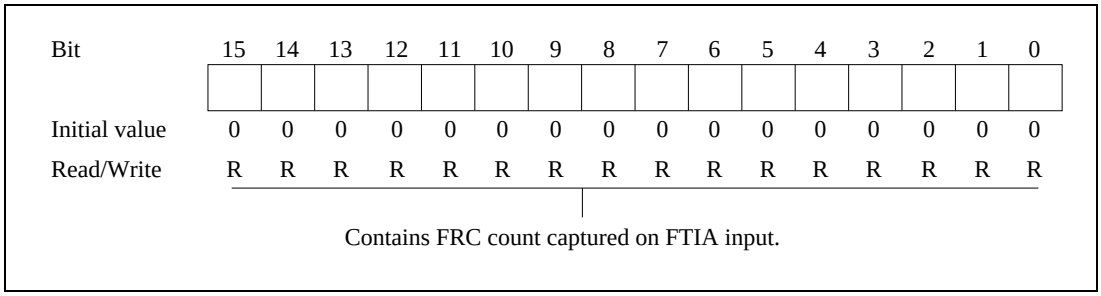
TCR—Timer Control Register H'FF96 FRT



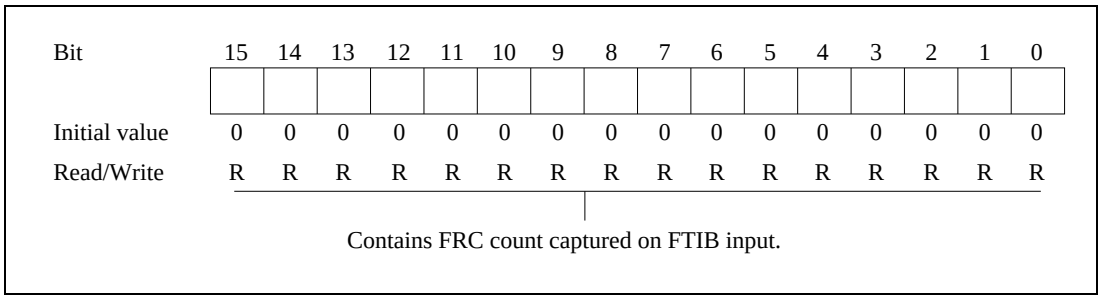
TOCR—Timer Output Compare Control Register H'FF97 FRT



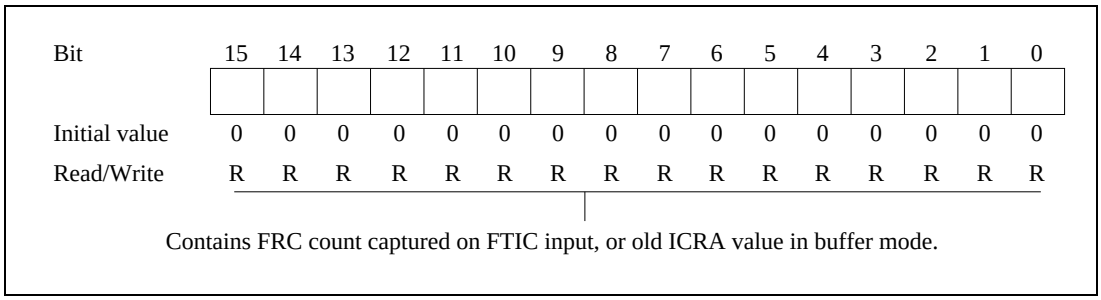
ICRA (H and L)—Input Capture Register A H'FF98, H'FF99 FRT



ICRB (H and L)—Input Capture Register B H'FF9A, H'FF9B FRT



ICRC (H and L)—Input Capture Register C H'FF9C, H'FF9D FRT



ICRD (H and L)—Input Capture Register D H'FF9E, H'FF9F 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	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Contains FRC count captured on FTID input, or old ICRB value in buffer mode.

TCR—Timer Control Register [H8/3534 only] H'FFA0 PWM0

Bit	7	6	5	4	3	2	1	0
	OE	OS	—	—	—	CKS2	CKS1	CKS0
Initial value	0	0	1	1	1	0	0	0
Read/Write	R/W	R/W	—	—	—	R/W	R/W	R/W

Clock Select (Values when ϕ_P = 10 MHz)

			Internal clock freq.	Resolution	PWM period	PWM frequency
0	0	0	$\phi_P/2$	200 ns	50 µs	20 kHz
		1	$\phi_P/8$	800 ns	200 µs	5 kHz
	1	0	$\phi_P/32$	3.2 µs	800 µs	1.25 kHz
		1	$\phi_P/128$	12.8 µs	3.2 ms	312.5 Hz
1	0	0	$\phi_P/256$	25.6 µs	6.4 ms	156.3 Hz
		1	$\phi_P/1024$	102.4 µs	25.6 ms	39.1 Hz
	1	0	$\phi_P/2048$	204.8 µs	51.2 ms	19.5 Hz
		1	$\phi_P/4096$	409.6 µs	102.4 ms	9.8 Hz

Output Select

0	PWM direct output
1	PWM inverse output

Output Enable

0	PWM output disabled; TCNT cleared to H'00 and stops.
1	PWM output enabled; TCNT runs.

DTR—Duty Register [H8/3534 only] H'FFA1 PWM0

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

Pulse duty cycle

TCNT—Timer Counter [H8/3534 only] H'FFA2 PWM0

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

Count value (runs from H'00 to H'F9, then repeats from H'00)

TCR—Timer Control Register [H8/3534 only] H'FFA4 PWM1

Bit	7	6	5	4	3	2	1	0
	OE	OS	—	—	—	CKS2	CKS1	CKS0
Initial value	0	0	1	1	1	0	0	0
Read/Write	R/W	R/W	—	—	—	R/W	R/W	R/W

Note: Bit functions are the same as for PWM0.

DTR—Duty Register [H8/3534 only] H'FFA5 PWM1

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

Note: Bit functions are the same as for PWM0.

TCNT—Timer Counter [H8/3534 only] H'FFA6 PWM1

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

Note: Bit functions are the same as for PWM0.

TCSR—Timer Control/Status Register H'FFA8 WDT

Bit	7	6	5	4	3	2	1	0
	OVF	WT/IT	TME	—	RST/NMI	CKS2	CKS1	CKS0
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

Clock Select 2 to 0

0	0	0	$\phi_P/2$
		1	$\phi_P/32$
	1	0	$\phi_P/64$
		1	$\phi_P/128$
1	0	0	$\phi_P/256$
		1	$\phi_P/512$
	1	0	$\phi_P/2048$
		1	$\phi_P/4096$

Reset or NMI Select

0	Functions as NMI (initial value)
1	Functions as reset

Timer Enable

0	Timer disabled: TCNT is initialized to H'00 and stopped (initial value)
1	Timer enabled: TCNT runs; CPU interrupts can be requested

Timer Mode Select

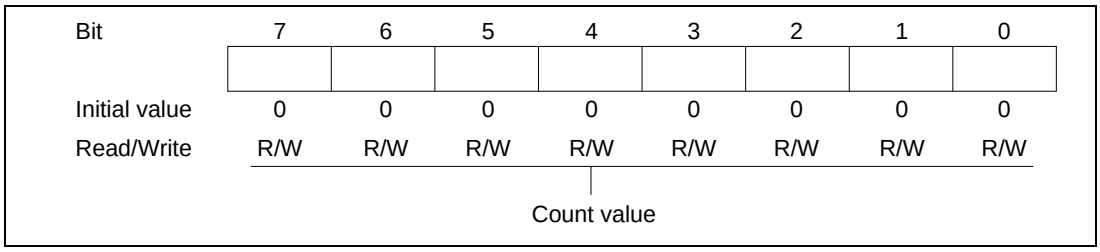
0	Interval timer mode (OVF request)
1	Watchdog timer mode (reset or NMI request)

Overflow Flag

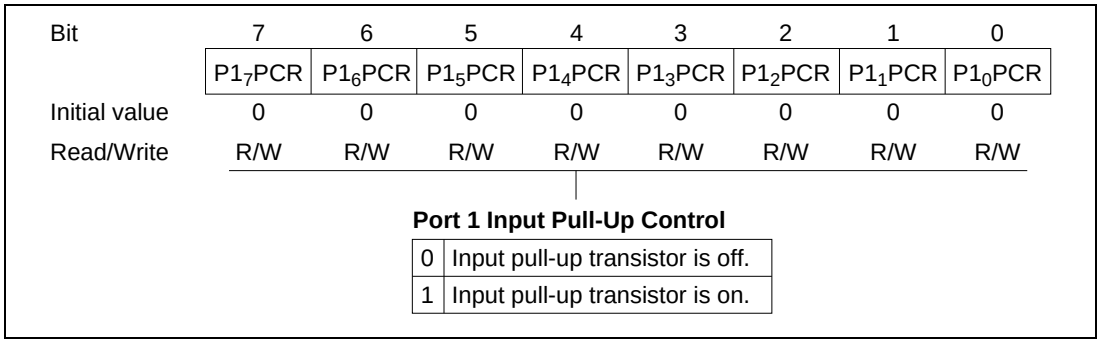
0	Cleared by reading OVF = 1, then writing 1 in OVF (initial value)
1	Set when TCNT changes from H'FF to H'00

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

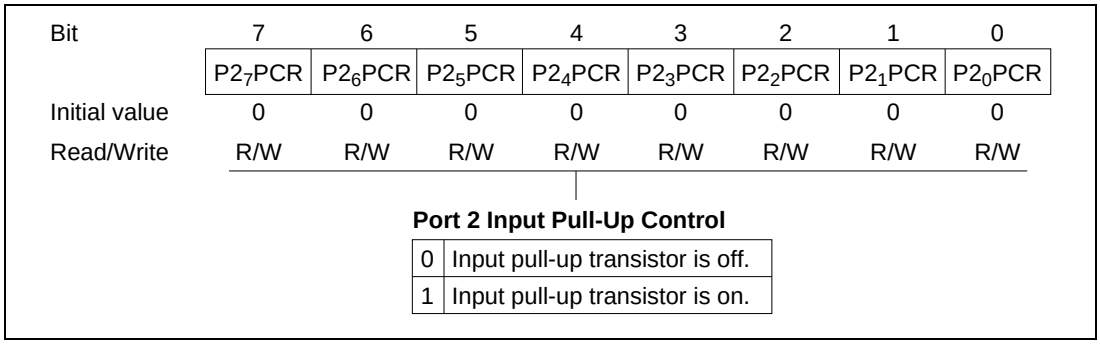
TCNT—Timer Counter H’FFA9 (read), WDT H’FFA8 (write)



P1PCR—Port 1 Input Pull-Up Control Register H'FFAC Port 1



P2PCR—Port 2 Input Pull-Up Control Register H'FFAD Port 2



P3PCR—Port 3 Input Pull-Up Control Register H'FFAE Port 3

Bit	7	6	5	4	3	2	1	0
	$P3_7PCR$	$P3_6PCR$	$P3_5PCR$	$P3_4PCR$	$P3_3PCR$	$P3_2PCR$	$P3_1PCR$	$P3_0PCR$
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

Port 3 Input Pull-Up Control

0	Input pull-up transistor is off.
1	Input pull-up transistor is on.

P1DDR—Port 1 Data Direction Register H'FFB0 Port 1

Bit	7	6	5	4	3	2	1	0
	$P1_7DDR$	$P1_6DDR$	$P1_5DDR$	$P1_4DDR$	$P1_3DDR$	$P1_2DDR$	$P1_1DDR$	$P1_0DDR$
Mode 1								
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—
Modes 2 and 3								
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 1 Input/Output Control

0	Input port
1	Output port

P1DR—Port 1 Data Register H'FFB2 Port 1

Bit	7	6	5	4	3	2	1	0
	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
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

P2DDR—Port 2 Data Direction Register H'FFB1 Port 2

Bit	7	6	5	4	3	2	1	0
	$P2_7DDR$	$P2_6DDR$	$P2_5DDR$	$P2_4DDR$	$P2_3DDR$	$P2_2DDR$	$P2_1DDR$	$P2_0DDR$
Mode 1								
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—
Modes 2 and 3								
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 2 Input/Output Control

0	Input port
1	Output port

P2DR—Port 2 Data Register H'FFB3 Port 2

Bit	7	6	5	4	3	2	1	0
	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
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

P3DDR—Port 3 Data Direction Register H'FFB4 Port 3

Bit	7	6	5	4	3	2	1	0
	$P3_7DDR$	$P3_6DDR$	$P3_5DDR$	$P3_4DDR$	$P3_3DDR$	$P3_2DDR$	$P3_1DDR$	$P3_0DDR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 3 Input/Output Control

0	Input port
1	Output port

P3DR—Port 3 Data Register H'FFB6 Port 3

Bit	7	6	5	4	3	2	1	0
	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
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

P4DDR—Port 4 Data Direction Register [H8/3534] H'FFB5 Port 4

Bit	7	6	5	4	3	2	1	0
	$P4_7$DDR	$P4_6$DDR	$P4_5$DDR	$P4_4$DDR	$P4_3$DDR	$P4_2$DDR	$P4_1$DDR	$P4_0$DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

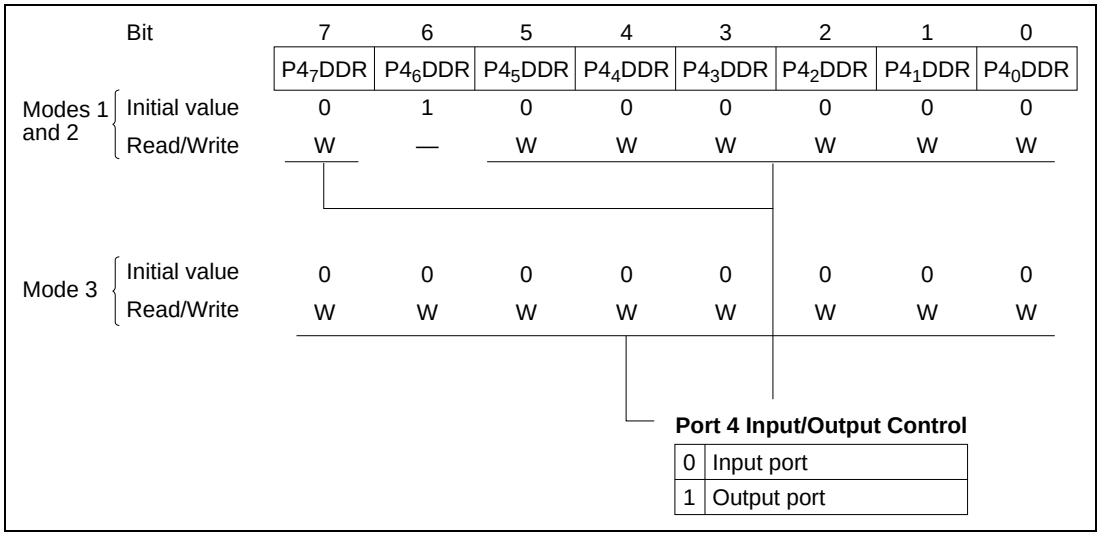
Port 4 Input/Output Control

0	Input port
1	Output port

P4DR—Port 4 Data Register [H8/3534] H'FFB7 Port 4

Bit	7	6	5	4	3	2	1	0
	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
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

P4DDR—Port 4 Data Direction Register [H8/3522] H'FFB5 Port 4



P4DR—Port 4 Data Register [H8/3522] H'FFB7 Port 4

Bit	7	6	5	4	3	2	1	0
	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
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: * Depends on the level at pin $P4_6$.

P5DDR—Port 5 Data Direction Register H'FFB8 Port 5

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	W	W	W

Port 5 Input/Output Control

0	Input port
1	Output port

P5DR—Port 5 Data Register H'FFBA Port 5

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	$P5_2$	$P5_1$	$P5_0$
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

P6DDR—Port 6 Data Direction Register H'FFB9 Port 6

Bit	7	6	5	4	3	2	1	0
	$P6_7$DDR	$P6_6$DDR	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	$P6_2$DDR	$P6_1$DDR	$P6_0$DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 6 Input/Output Control

0	Input port
1	Output port

P6DR—Port 6 Data Register H'FFBB Port 6

Bit	7	6	5	4	3	2	1	0
	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
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

P7PIN—Port 7 Input Data Register H'FFBE Port 7

Bit	7	6	5	4	3	2	1	0
	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Note: *Depends on the levels of pins $P7_7$ to $P7_0$.

P8DDR—Port 8 Data Direction Register [H8/3534 only] H'FFBD Port 8

Bit	7	6	5	4	3	2	1	0
	—	$P8_6$DDR	$P8_5$DDR	$P8_4$DDR	$P8_3$DDR	$P8_2$DDR	$P8_1$DDR	$P8_0$DDR
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	W	W	W	W	W	W	W

Port 8 Input/Output Control

0	Input port
1	Output port

P8DR—Port 8 Data Register [H8/3534 only] H'FFBF Port 8

Bit	7	6	5	4	3	2	1	0
	—	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$
Initial value	1	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P9DDR—Port 9 Data Direction Register [H8/3534 only] H'FFC0 Port 9

Bit	7	6	5	4	3	2	1	0
	$P9_7$DDR	$P9_6$DDR	$P9_5$DDR	$P9_4$DDR	$P9_3$DDR	$P9_2$DDR	$P9_1$DDR	$P9_0$DDR
Modes 1 and 2								
Initial value	0	1	0	0	0	0	0	0
Read/Write	W	—	W	W	W	W	W	W
Mode 3								
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 9 Input/Output Control

0	Input port
1	Output port

P9DR—Port 9 Data Register [H8/3534 only] H'FFC1 Port 9

Bit	7	6	5	4	3	2	1	0
	$P9_7$	$P9_6$	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
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: *Depends on the level of pin $P9_6$.

WSCR—Wait-State Control Register [H8/3534] H'FFC2 System control

Bit	7	6	5	4	3	2	1	0
	(RAMS)	(RAM0)	CKDBL	—	WMS1	WMS0	WC1	WC0
Initial value	0	0	0	0	1	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Wait Count

0 0	No wait states inserted by wait-state controller (initial value)
0 1	1 state inserted
1 0	2 states inserted
1 1	3 states inserted

Wait Mode Select

0 0	Programmable wait mode
0 1	No wait states inserted by wait-state controller
1 0	Pin wait mode (initial value)
1 1	Pin auto-wait mode

Clock Double

0	Supporting module clock frequency is not divided ($ø_P$ = ø) (initial value)
1	Supporting module clock frequency is divided by two ($ø_P$ = ø/2)

Note: Do not write 1 to bits RAMS and RAM0.

WSCR—Wait-State Control Register [H8/3522] H'FFC2 System control

Bit	7	6	5	4	3	2	1	0
	—	—	CKDBL	—	WMS1	WMS0	WC1	WC0
Initial value	0	0	0	0	1	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Wait Count

0 0	No wait states inserted by wait-state controller (initial value)
0 1	1 state inserted
1 0	2 states inserted
1 1	3 states inserted

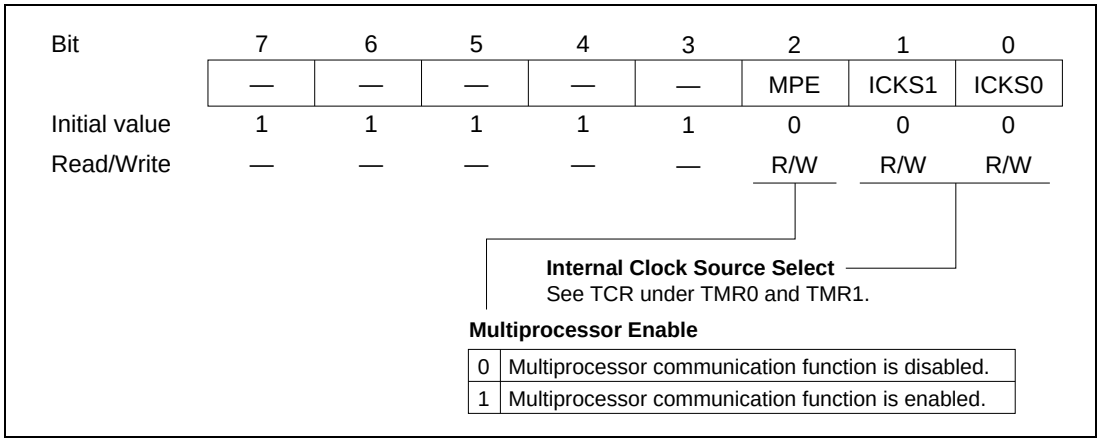
Wait Mode Select

0 0	Programmable wait mode
0 1	No wait states inserted by wait-state controller
1 0	Pin wait mode (initial value)
1 1	Pin auto-wait mode

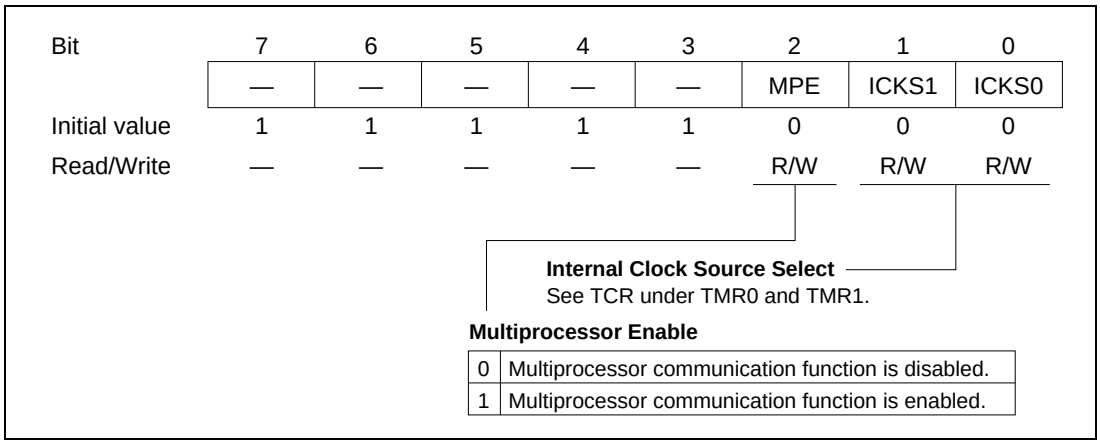
Clock Double

0	Supporting module clock frequency is not divided ($ø_P$ = ø) (initial value)
1	Supporting module clock frequency is divided by two ($ø_P$ = ø/2)

STCR—Serial/Timer Control Register [H8/3534] H'FFC3 System Control



STCR—Serial/Timer Control Register [H8/3522] H'FFC3 System Control



SYSCR—System Control Register [H8/3534] H'FFC4 System Control

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	XRST	NMIEG	(HIE)	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

RAM Enable

0	On-chip RAM is disabled.
1	On-chip RAM is enabled. (initial value)

NMI Edge

0	Falling edge of $\overline{\text{NMI}}$ is detected. (initial value)
1	Rising edge of $\overline{\text{NMI}}$ is detected.

External Reset

0	Reset was caused by watchdog timer overflow
1	Reset was caused by external reset signal (initial value)

Standby Timer Select 2 to 0

0	0	0	Clock settling time = 8,192 states (initial value)
0	0	1	Clock settling time = 16,384 states
0	1	0	Clock settling time = 32,768 states
0	1	1	Clock settling time = 65,536 states
1	0	—	Clock settling time = 131,072 states
1	1	—	Unused

Software Standby

0	SLEEP instruction causes transition to sleep mode. (initial value)
1	SLEEP instruction causes transition to software standby mode.

Note: Do not write 1 to bit HIE.

SYSCR—System Control Register [H8/3522] H'FFC4System Control

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	XRST	NMIEG	—	RAME
Initial value	0	0	0	0	1	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	—	R/W

RAM Enable

0	On-chip RAM is disabled.
1	On-chip RAM is enabled. (initial value)

NMI Edge

0	Falling edge of $\overline{\text{NMI}}$ is detected. (initial value)
1	Rising edge of $\overline{\text{NMI}}$ is detected.

External Reset

0	Reset was caused by watchdog timer overflow
1	Reset was caused by external reset signal (initial value)

Standby Timer Select 2 to 0

0	0	0	Clock settling time = 8,192 states (initial value)
0	0	1	Clock settling time = 16,384 states
0	1	0	Clock settling time = 32,768 states
0	1	1	Clock settling time = 65,536 states
1	0	—	Clock settling time = 131,072 states
1	1	—	Unused

Software Standby

0	SLEEP instruction causes transition to sleep mode. (initial value)
1	SLEEP instruction causes transition to software standby mode.

MDCR—Mode Control Register H'FFC5System Control

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

Mode Select Bits

Value at mode pins.

Note: *Determined by inputs at pins MD_1 and MD_0.

ISCR—IRQ Sense Control Register [H8/3534] H'FFC6System Control

Bit	7	6	5	4	3	2	1	0
	IRQ7SC	IRQ6SC	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC
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

IRQ0 to IRQ7 Sense Control

0	$\overline{IRQ_0}$ to $\overline{IRQ_7}$ are level-sensed (active low).
1	$\overline{IRQ_0}$ to $\overline{IRQ_7}$ are edge-sensed (falling edge).

IER—IRQ Enable Register [H8/3534] H'FFC7System Control

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

IRQ0 to IRQ7 Enable

0	IRQ_0 to IRQ_7 are disabled.
1	IRQ_0 to IRQ_7 are enabled.

ISCR—IRQ Sense Control Register [H8/3522] H'FFC6System Control

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	IRQ2SC	IRQ1SC	IRQ0SC
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

IRQ0 to IRQ2 Sense Control

0	$\overline{IRQ_0}$ to $\overline{IRQ_2}$ are level-sensed (active low).
1	$\overline{IRQ_0}$ to $\overline{IRQ_2}$ are edge-sensed (falling edge).

IER—IRQ Enable Register [H8/3522] H'FFC7System Control

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/W	R/W	R/W

IRQ0 to IRQ2 Enable

0	IRQ_0 to IRQ_2 are disabled.
1	IRQ_0 to IRQ_2 are enabled.

TCR—Timer Control Register H'FFC8TMR0

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

Clock Select

TCR			STCR		Description
Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
CKS2	CKS1	CKS0	ICKS1	ICKS0	
0	0	0	—	—	Timer stopped (Initial value)
0	0	1	—	0	ϕ_P/8 internal clock, falling edge
0	0	1	—	1	ϕ_P/2 internal clock, falling edge
0	1	0	—	0	ϕ_P/64 internal clock, falling edge
0	1	0	—	1	ϕ_P/32 internal clock, falling edge
0	1	1	—	0	ϕ_P/1024 internal clock, falling edge
0	1	1	—	1	ϕ_P/256 internal clock, falling edge
1	0	0	—	—	Timer stopped
1	0	1	—	—	External clock, rising edge
1	1	0	—	—	External clock, falling edge
1	1	1	—	—	External clock, rising and falling edges

Counter Clear

0	0	Counter is not cleared.
0	1	Cleared by compare-match A.
1	0	Cleared by compare-match B.
1	1	Cleared on rising edge of external reset input.

Timer Overflow Interrupt Enable

0	Timer overflow interrupt request is disabled.
1	Timer overflow interrupt request is enabled.

Compare-Match Interrupt Enable A

0	Compare-match A interrupt request is disabled.
1	Compare-match A interrupt request is enabled.

Compare-Match Interrupt Enable B

0	Compare-match B interrupt request is disabled.
1	Compare-match B interrupt request is enabled.

TCSR—Timer Control/Status Register H'FFC9TMR0

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OS3 *1	OS2 *1	OS1 *1	OS0 *1
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/(W)*2	R/(W)*2	R/(W)*2	—	R/W	R/W	R/W	R/W

Output Select

0	0	No change on compare-match A.
0	1	Output 0 on compare-match A.
1	0	Output 1 on compare-match A.
1	1	Invert (toggle) output on compare-match A.

Output Select

0	0	No change on compare-match B.
0	1	Output 0 on compare-match B.
1	0	Output 1 on compare-match B.
1	1	Invert (toggle) output on compare-match B.

Timer Overflow Flag

0	Cleared by reading OVF = 1, then writing 0 in OVF.
1	Set when TCNT changes from H'FF to H'00.

Compare-Match Flag A

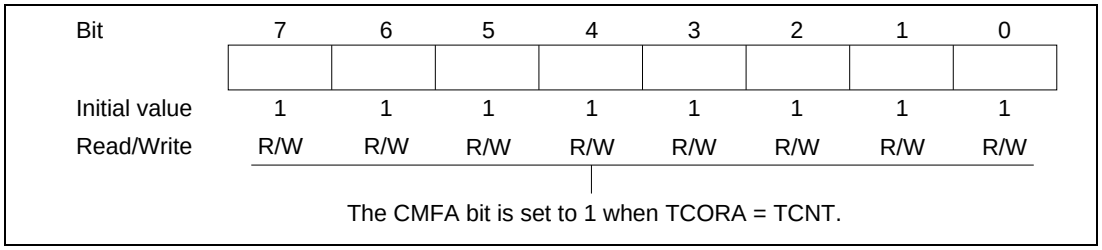
0	Cleared by reading CMFA = 1, then writing 0 in CMFA.
1	Set when TCNT = TCORA.

Compare-Match Flag B

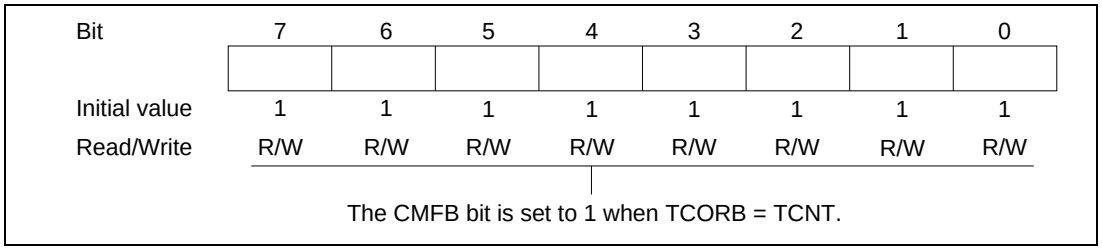
0	Cleared by reading CMFB = 1, then writing 0 in CMFB.
1	Set when TCNT = TCORB.

Notes: 1. When all four bits (OS3 to OS0) are cleared to 0, output is disabled.
2. Software can write a 0 in bits 7 to 5 to clear the flags, but cannot write a 1 in these bits.

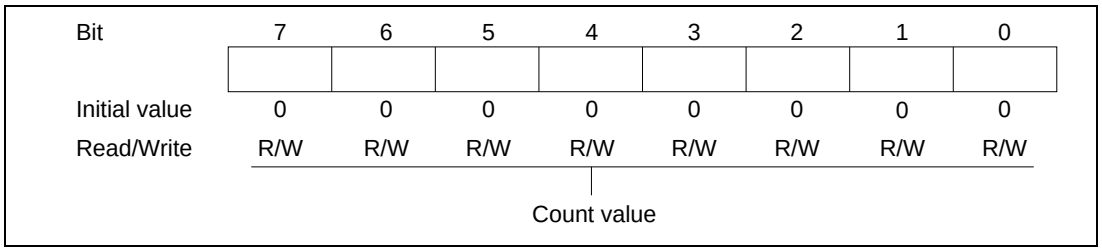
TCORA—Time Constant Register A H'FFCA TMR0



TCORB—Time Constant Register B H'FFCB TMR0



TCNT—Timer Counter H'FFCC TMR0



TCR—Timer Control Register H'FFD0TMR1

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

Clock Select

TCR			STCR		Description
Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
CKS2	CKS1	CKS0	ICKS1	ICKS0	
0	0	0	—	—	Timer stopped
0	0	1	0	—	ϕ_p/8 internal clock, falling edge
0	0	1	1	—	ϕ_p/2 internal clock, falling edge
0	1	0	0	—	ϕ_p/64 internal clock, falling edge
0	1	0	1	—	ϕ_p/128 internal clock, falling edge
0	1	1	0	—	ϕ_p/1024 internal clock, falling edge
0	1	1	1	—	ϕ_p/2048 internal clock, falling edge
1	0	0	—	—	Timer stopped
1	0	1	—	—	External clock, rising edge
1	1	0	—	—	External clock, falling edge
1	1	1	—	—	External clock, rising and falling edges

Counter Clear

0	0	Counter is not cleared.
0	1	Cleared by compare-match A.
1	0	Cleared by compare-match B.
1	1	Cleared on rising edge of external reset input.

Timer Overflow Interrupt Enable

0	Timer overflow interrupt request is disabled.
1	Timer overflow interrupt request is enabled.

Compare-Match Interrupt Enable A

0	Compare-match A interrupt request is disabled.
1	Compare-match A interrupt request is enabled.

Compare-Match Interrupt Enable B

0	Compare-match B interrupt request is disabled.
1	Compare-match B interrupt request is enabled.

TCSR—Timer Control/Status Register H'FFD1TMR1

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OS3 *1	OS2 *1	OS1 *1	OS0 *1
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/(W) *2	R/(W) *2	R/(W) *2	—	R/W	R/W	R/W	R/W

Notes: Bit functions are the same as for TMR0.
1. When all four bits (OS3 to OS0) are cleared to 0, output is disabled.
2. Software can write a 0 in bits 7 to 5 to clear the flags, but cannot write a 1 in these bits.

TCORA—Time Constant Register A H'FFD2TMR1

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

Note: Bit functions are the same as for TMR0.

TCORB—Time Constant Register B H'FFD3TMR1

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

Note: Bit functions are the same as for TMR0.

TCNT—Timer Counter H'FFD4TMR1

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

Note: Bit functions are the same as for TMR0.

SMR—Serial Mode Register H'FFD8SCI0

Bit	7	6	5	4	3	2	1	0
	C/A	CHR	PE	O/E	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	ø clock
0	1	$ø_P$/4 clock
1	0	$ø_P$/16 clock
1	1	$ø_P$/64 clock

Multiprocessor Mode

0	Multiprocessor function disabled
1	Multiprocessor function enabled

Stop Bit Length

0	One stop bit
1	Two stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

0	Transmit:	No parity bit added.
	Receive:	Parity bit not checked.
1	Transmit:	Parity bit added.
	Receive:	Parity bit checked.

Character Length

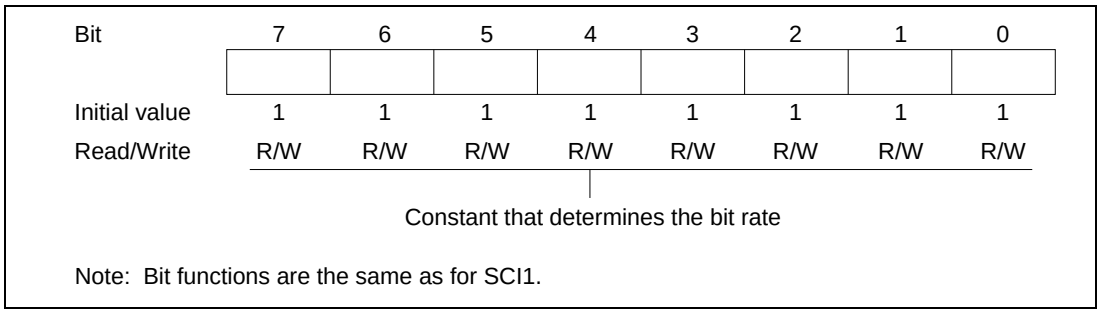
0	8-bit per character
1	7-bit per character

Communication Mode

0	Asynchronous
1	Synchronous

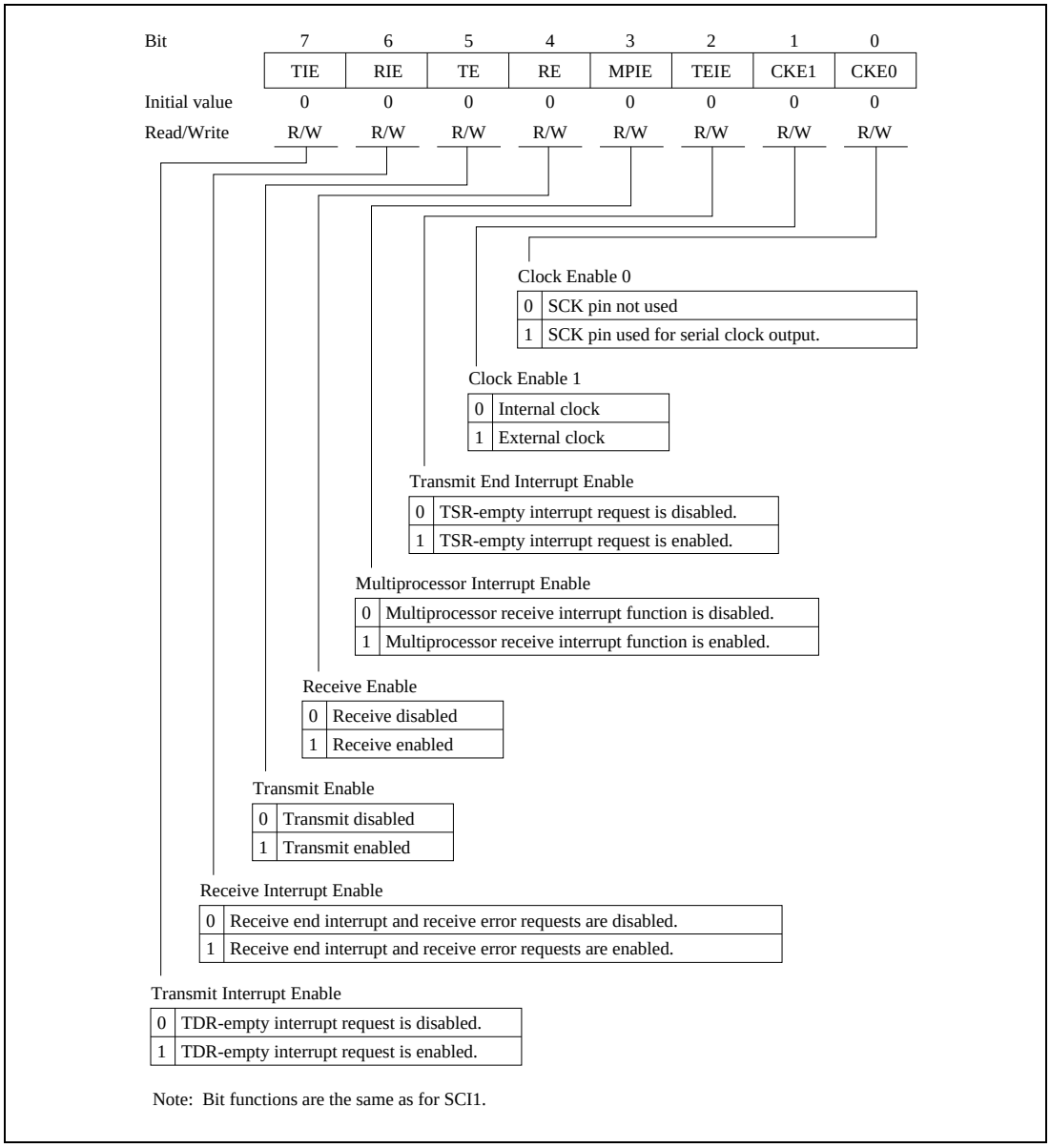
Note: Bit functions are the same as for SCI1.

BRR—Bit Rate Register H'FFD9SCI0



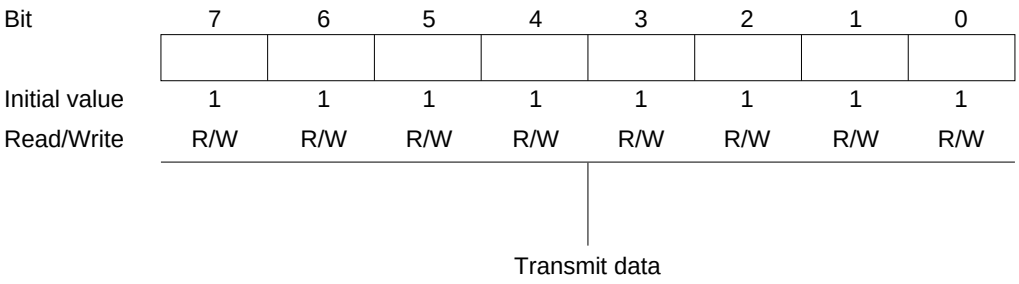
Note: Bit functions are the same as for SCI1.

SCR—Serial Control Register H'FFDA SCI0



Note: Bit functions are the same as for SCI1.

TDR—Transmit Data Register H'FFDB SCI0



Note: Bit functions are the same as for SCI1.

SSR—Serial Status Register 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	Multiprocessor bit = 0 in transmit data. (Initial Value)
1	Multiprocessor bit = 1 in transmit data.

Multiprocessor Bit

0	Multiprocessor bit = 0 in receive data. (Initial Value)
1	Multiprocessor bit = 1 in receive data.

Transmit End

0	Cleared by reading TDRE = 1, then writing 0 in TDRE.
1	Set to 1 when TE = 0, or when TDRE = 1 at the end of character transmission. (Initial Value)

Parity Error

0	Cleared by reading PER = 1, then writing 0 in PER. (Initial Value)
1	Set when a parity error occurs (parity of receive data does not match parity selected by O/E bit in SMR).

Framing Error

0	Cleared by reading FER = 1, then writing 0 in FER. (Initial Value)
1	Set when a framing error occurs (stop bit is 0).

Overrun Error

0	Cleared by reading ORER = 1, then writing 0 in ORER. (Initial Value)
1	Set when an overrun error occurs (next data is completely received while RDRF bit is set to 1).

Receive Data Register Full

0	Cleared by reading RDRF = 1, then writing 0 in RDRF. (Initial Value)
1	Set when one character is received normally and transferred from RSR to RDR.

Transmit Data Register Empty

0	Cleared by reading TDRE = 1, then writing 0 in TDRE.
1	Set when: (Initial Value) 1. Data is transferred from TDR to TSR. 2. TE is cleared to 0 while TDRE = 0.

Note: * Software can write a 0 in bits 7 to 3 to clear the flags, but cannot write a 1 in these bits.
Bit functions are the same as for SCI1.

RDR—Receive Data Register 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

Receive data

Note: Bit functions are the same as for SCI1.

ADDRA (H and L)—A/D Data Register A H'FFE0, H'FFE1 A/D

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

ADDRA H ADDRA L

A/D Conversion Data
10-bit data giving an A/D conversion result

Reserved Bits

ADDRB (H and L)—A/D Data Register B H'FFE2, H'FFE3 A/D

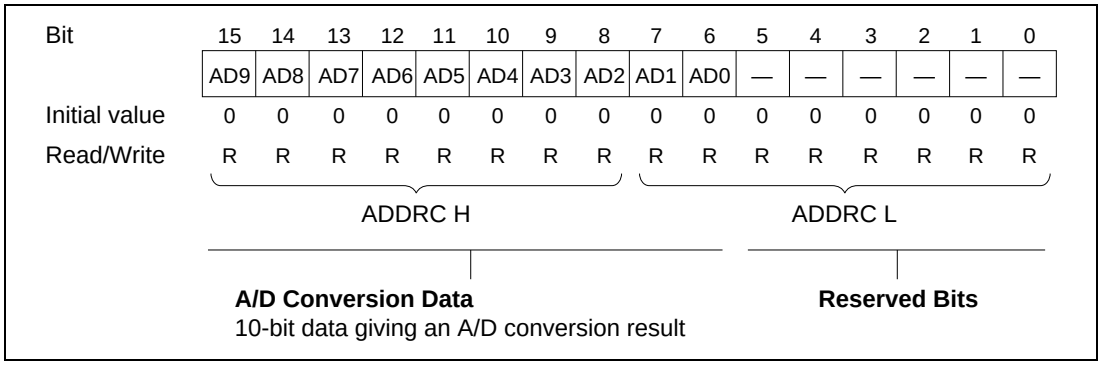
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

ADDRB H ADDRB L

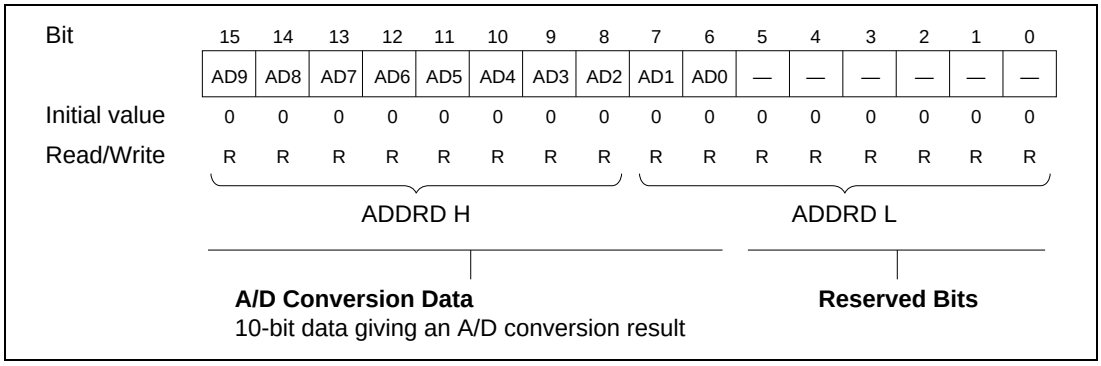
A/D Conversion Data
10-bit data giving an A/D conversion result

Reserved Bits

ADDRC (H and L)—A/D Data Register C H'FFE4, H'FFE5 A/D



ADDRD (H and L)—A/D Data Register D H'FFE6, H'FFE7 A/D



ADCSR—A/D Control/Status Register H'FFE8 A/D

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	AN_0	AN_0
		1	AN_1	AN_0, AN_1
	1	0	AN_2	AN_0 to AN_2
		1	AN_3	AN_0 to AN_3
1	0	0	AN_4	AN_4
		1	AN_5	AN_4, AN_5
	1	0	AN_6	AN_4 to AN_6
		1	AN_7	AN_4 to AN_7

Clock Select

0	Conversion time = 266 states (max)
1	Conversion time = 134 states (max)

Note: $ø_P = ø$

Scan Mode

0	Single mode
1	Scan mode

A/D Start

0	A/D conversion is halted.
1	1. Single mode: One A/D conversion is performed, then this bit is automatically cleared to 0. 2. Scan mode: A/C conversion starts and continues cyclically on all selected channels until 0 is written in this bit.

A/D Interrupt Enable

0	The A/D interrupt request (ADI) is disabled.
1	The A/D interrupt request (ADI) is enabled.

A/D End Flag

0	Cleared from 1 to 0 when CPU reads ADF = 1, then writes 0 in ADF.
1	Set to 1 at the following times: 1. Single mode: at the completion of A/D conversion 2. Scan mode: when all selected channels have been converted.

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

ADCR—A/D Control Register H'FFE9 A/D

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

Trigger Enable

0	$\overline{\text{ADTRG}}$ is disabled.
1	$\overline{\text{ADTRG}}$ is enabled. A/D conversion can be started by external trigger, or by software.

KMIMR—Keyboard Matrix Interrupt Mask Register H'FFF1 HIF
[H8/3534 only]

Bit	7	6	5	4	3	2	1	0
	KMIMR7	KMIMR6	KMIMR5	KMIMR4	KMIMR3	KMIMR2	KMIMR1	KMIMR0
Initial value	1	0	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Keyboard Matrix Interrupt Mask

0	Key-sense input interrupt request enabled
1	Key-sense input interrupt request disabled (initial value)*

Note: * Initial value of KMIMR6 is 0.

KMPCR—Port 6 Input Pull-Up Control Register H'FFF2 HIF
[H8/3534 only]

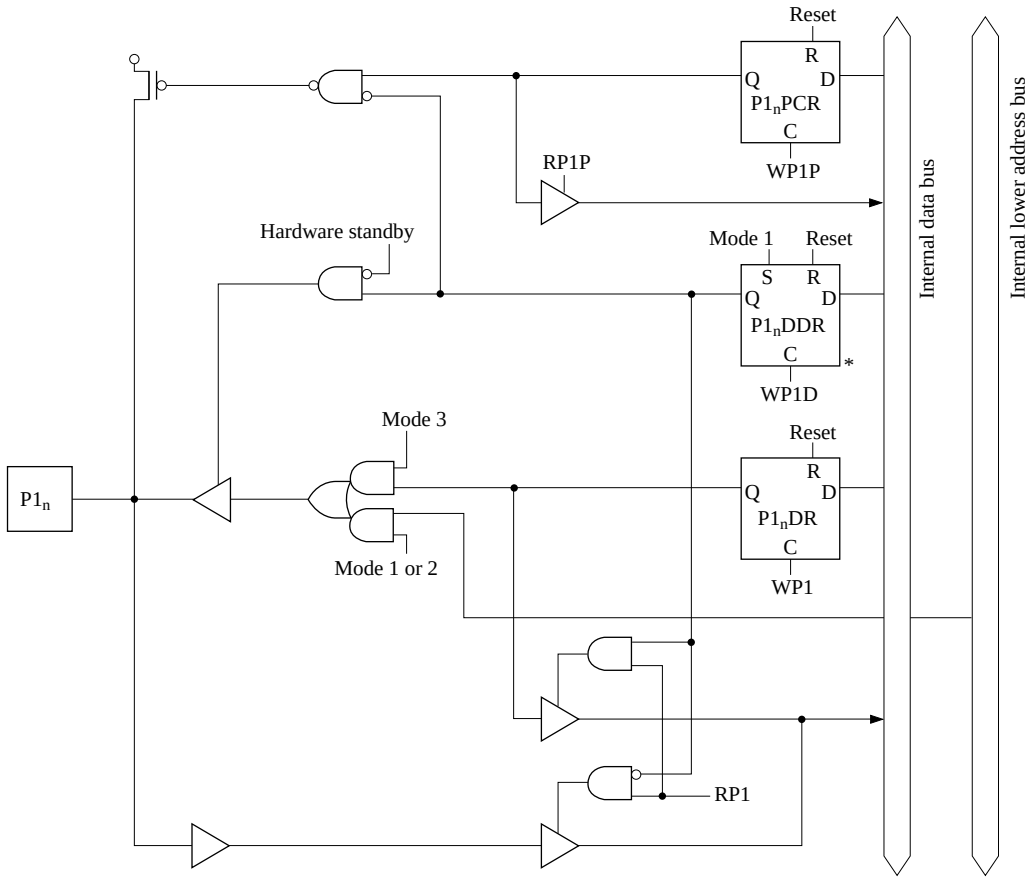
Bit	7	6	5	4	3	2	1	0
	KM_7PCR	KM_6PCR	KM_5PCR	KM_4PCR	KM_3PCR	KM_2PCR	KM_1PCR	KM_0PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 6 Input Pull-Up Control

0	Input pull-up transistor is off. (Initial value)
1	Input pull-up transistor is on.

Appendix C I/O Port Block Diagrams

C.1 Port 1 Block Diagram

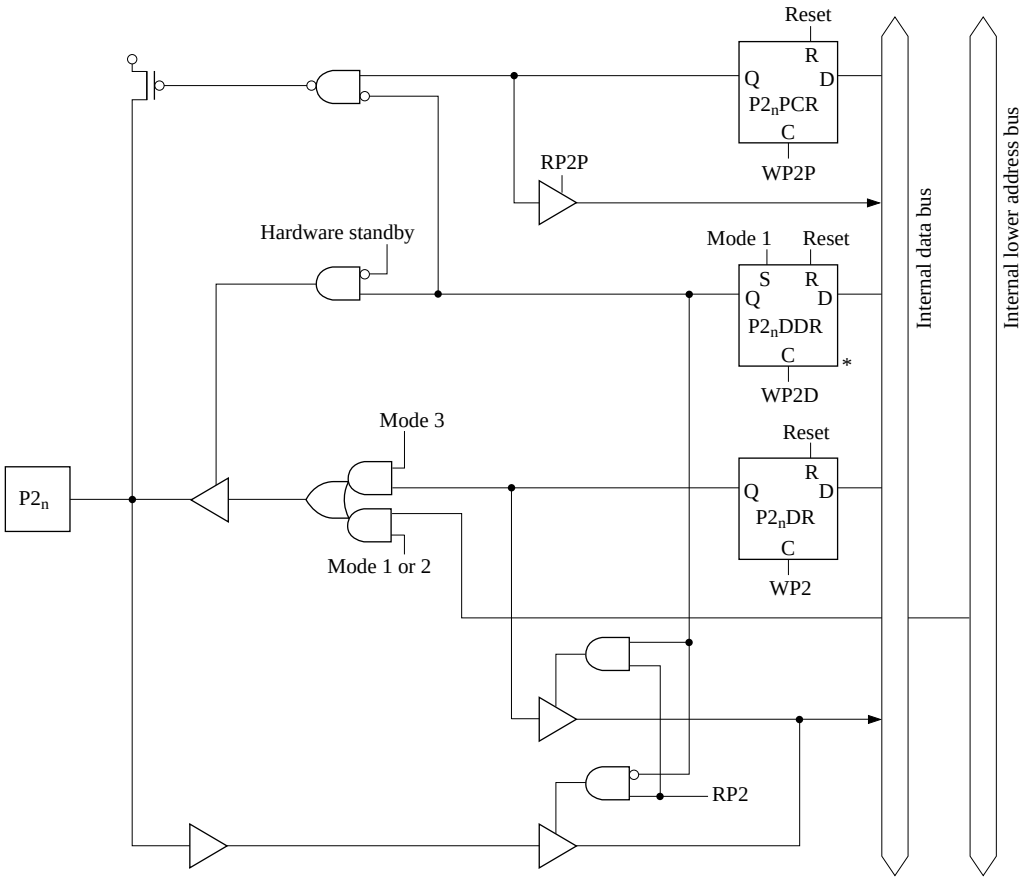


WP1P: Write to P1PCR
WP1D: Write to P1DDR
WP1: Write to port 1
RP1P: Read P1PCR
RP1: Read port 1
n = 0 to 7

Note: * Set priority

Figure C-1 Port 1 Block Diagram

C.2 Port 2 Block Diagram



WP2P: Write to P2PCR
WP2D: Write to P2DDR
WP2: Write to port 2
RP2P: Read P2PCR
RP2: Read port 2
n = 0 to 7

Note: * Set priority

Figure C-2 Port 2 Block Diagram

C.3 Port 3 Block Diagram

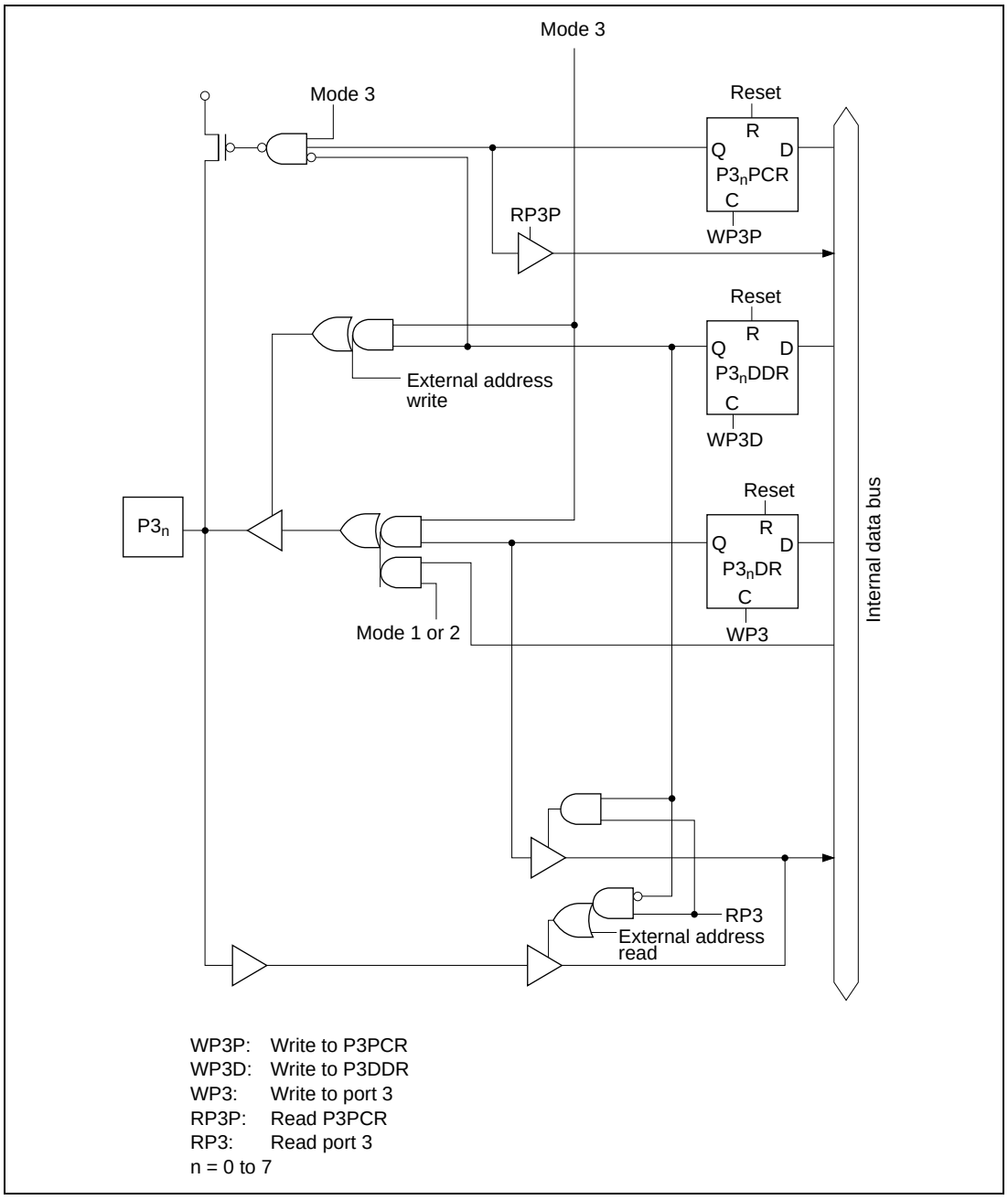


Figure C-3 Port 3 Block Diagram

C.4 Port 4 Block Diagrams [H8/3534]

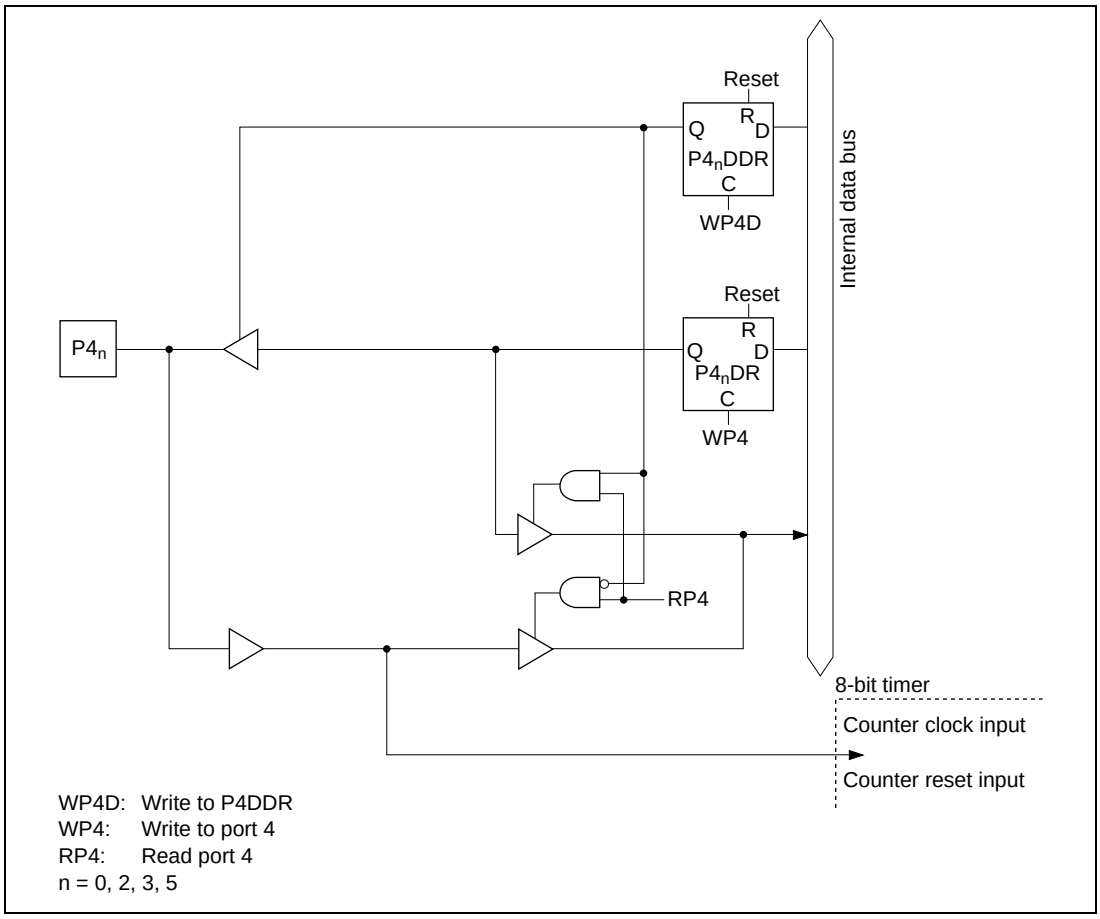


Figure C-4 (a) Port 4 Block Diagram (Pins $P4_0$, $P4_2$, $P4_3$, $P4_5$) [H8/3534]

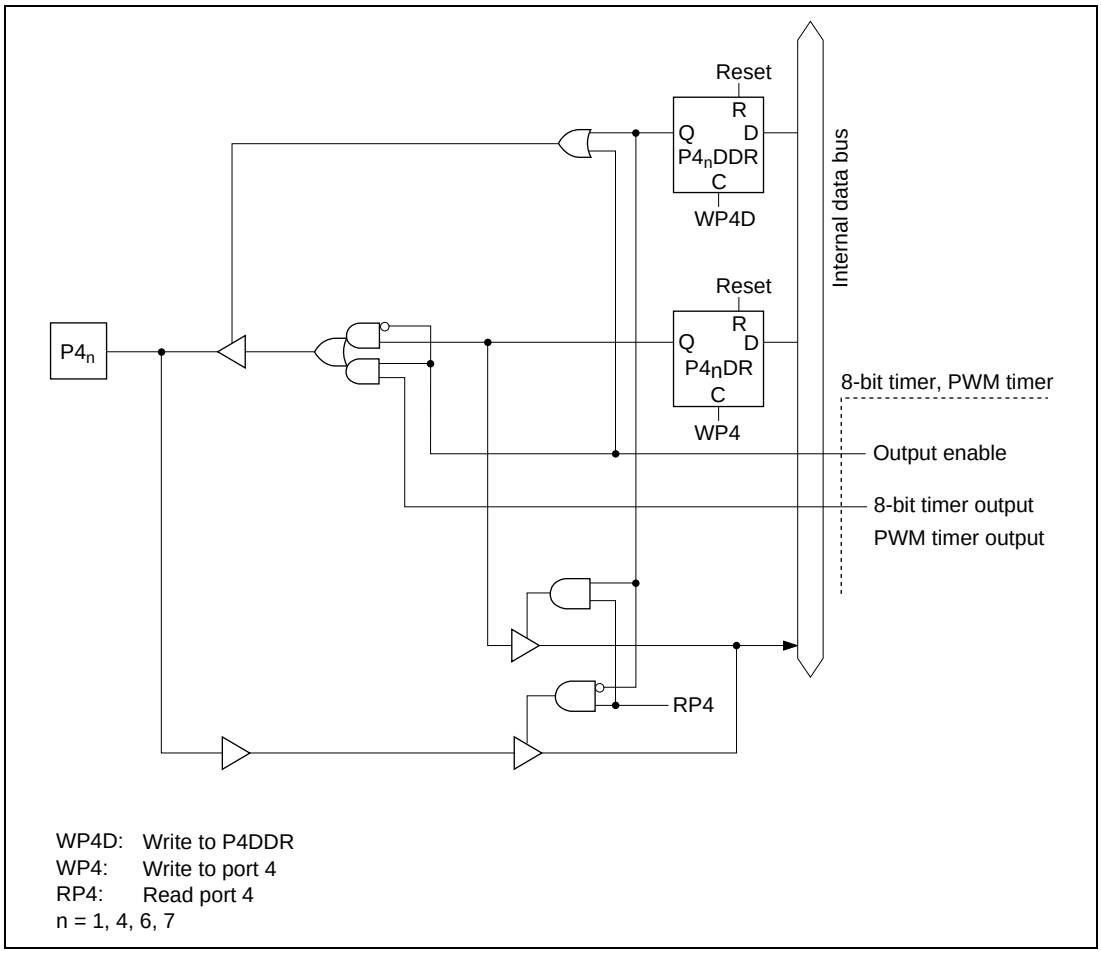


Figure C-4 (b) Port 4 Block Diagram (Pins $P4_1$, $P4_4$, $P4_6$, $P4_7$) [H8/3534]

C.5 Port 5 Block Diagrams

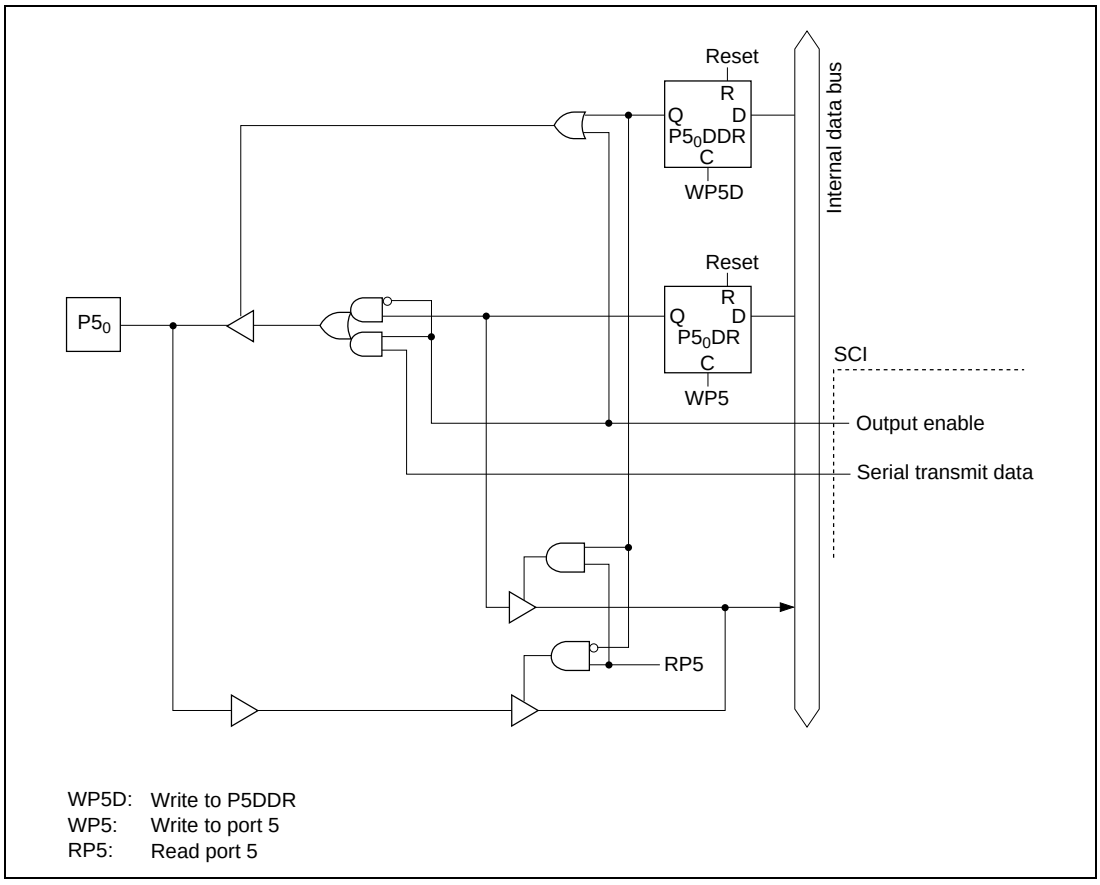


Figure C-5 (a) Port 5 Block Diagram (Pin $P5_0$)

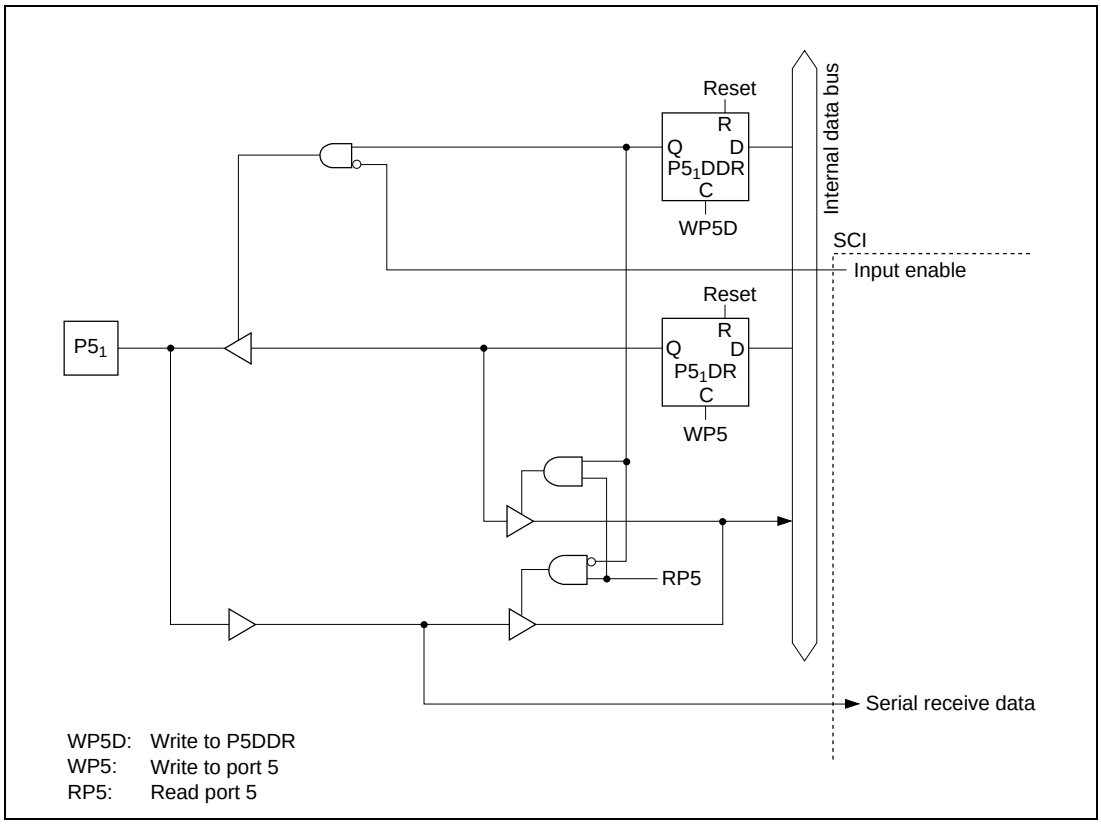


Figure C-5 (b) Port 5 Block Diagram (Pin $P5_1$)

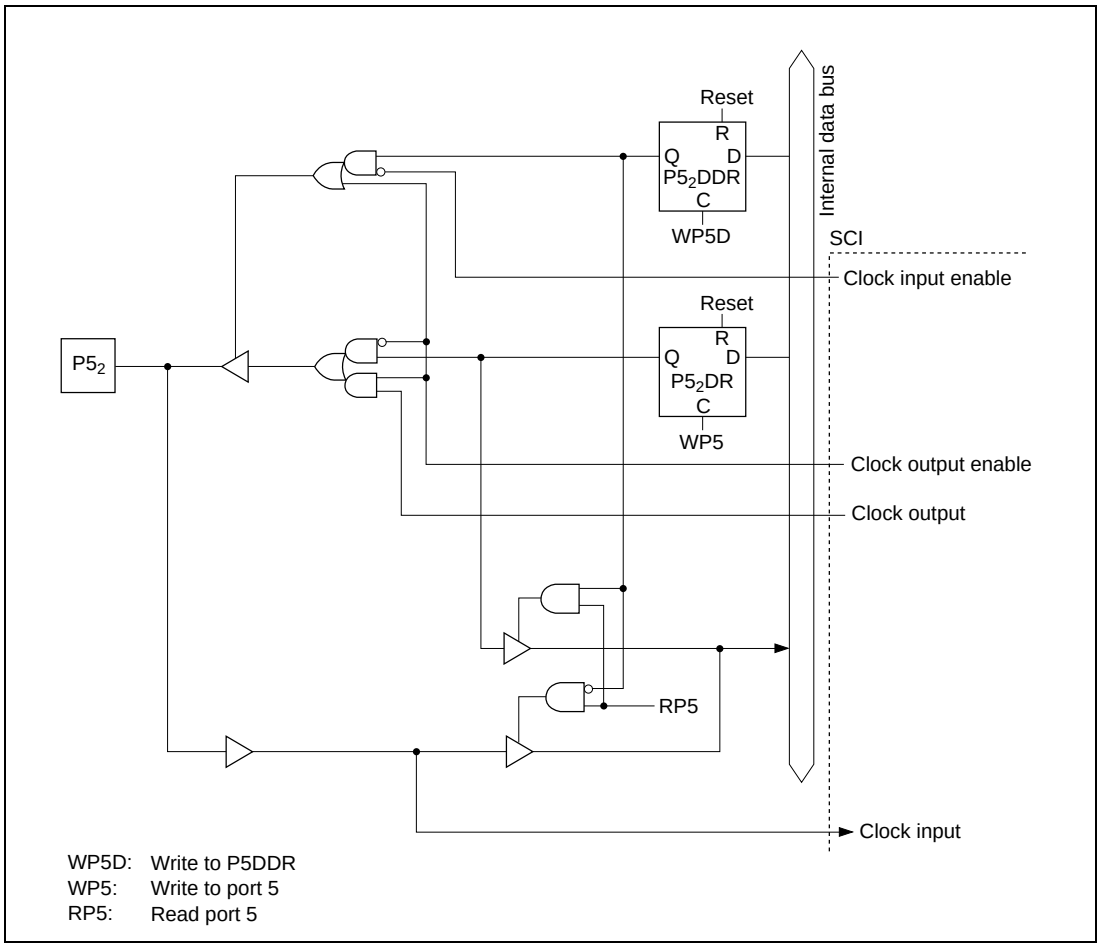


Figure C-5 (c) Port 5 Block Diagram (Pin $P5_2$)

C.6 Port 6 Block Diagrams

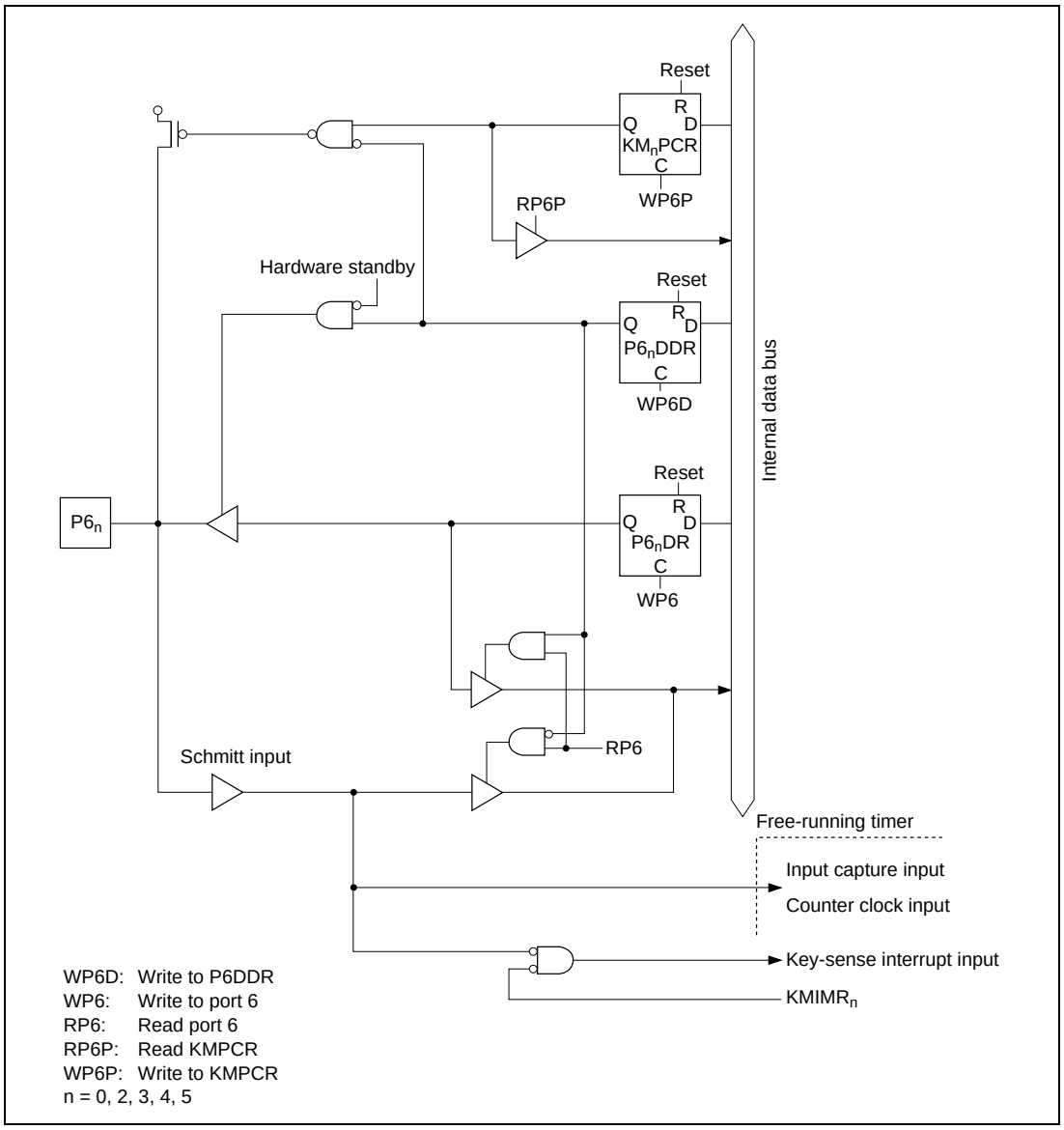


Figure C-6 (a) Port 6 Block Diagram (Pins $P6_0$, $P6_2$, $P6_3$, $P6_4$, $P6_5$) [H8/3534]

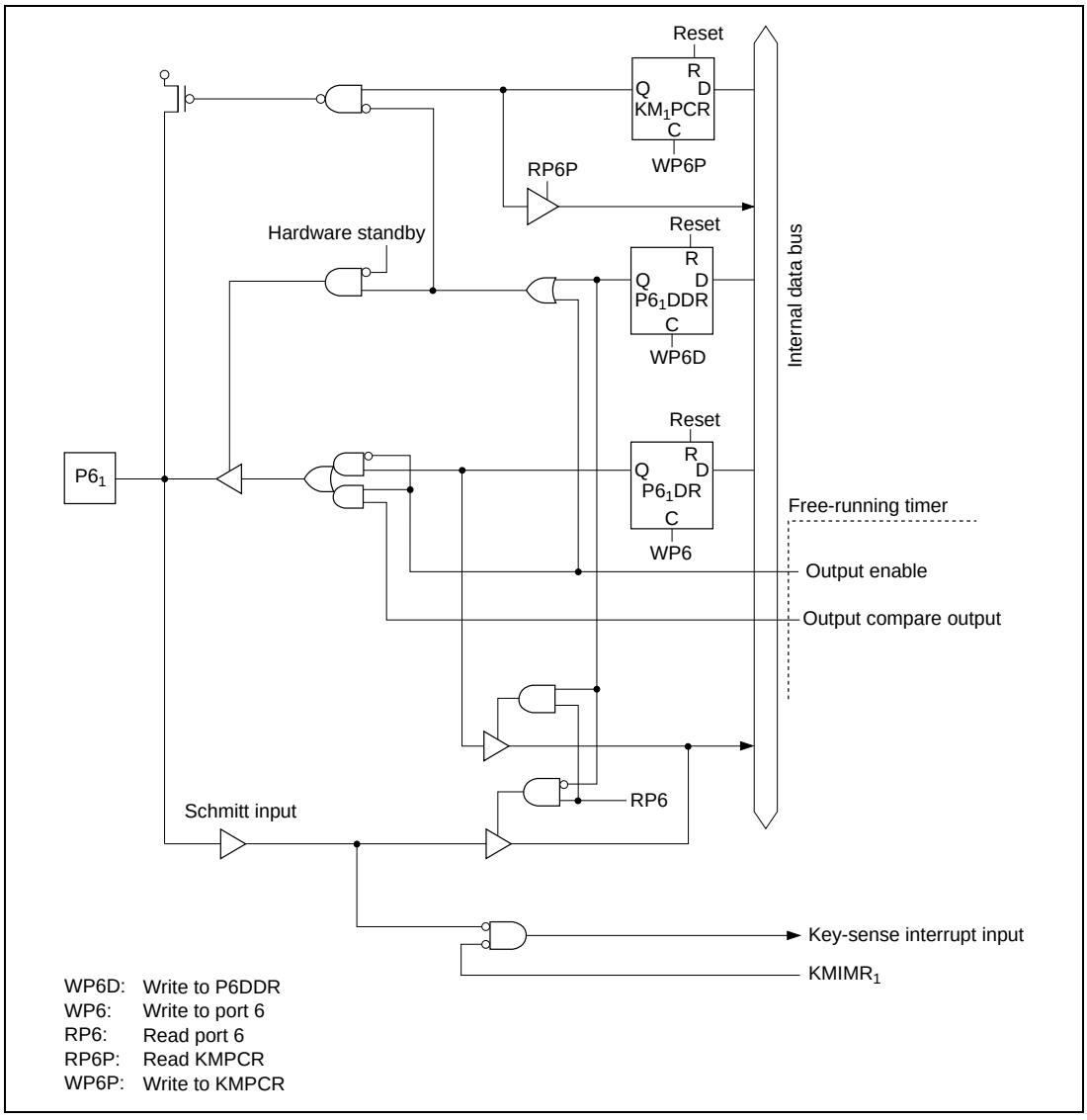


Figure C-6 (b) Port 6 Block Diagram (Pin $P6_1$) [H8/3534]

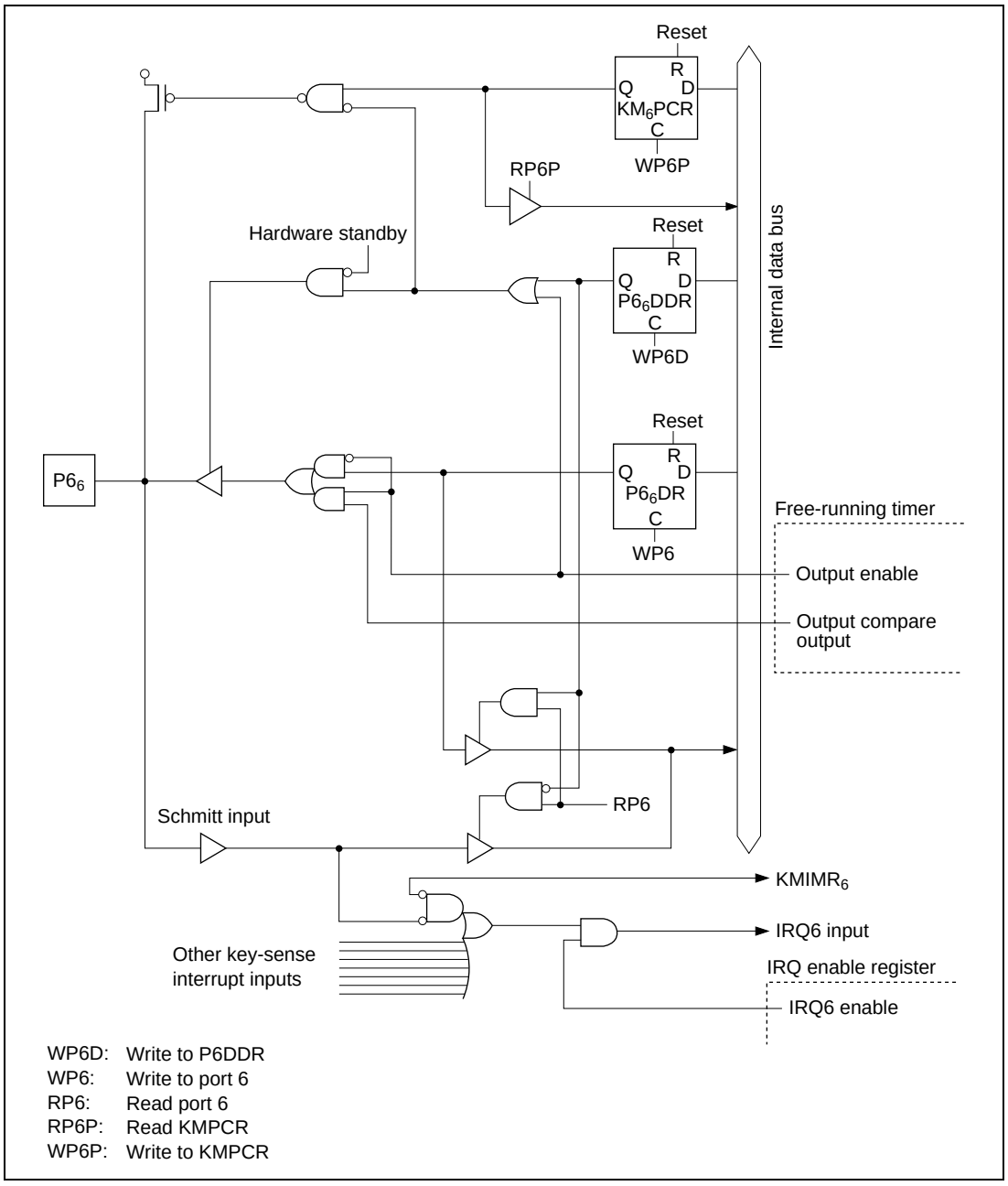


Figure C-6 (c) Port 6 Block Diagram (Pin $P6_6$) [H8/3534]

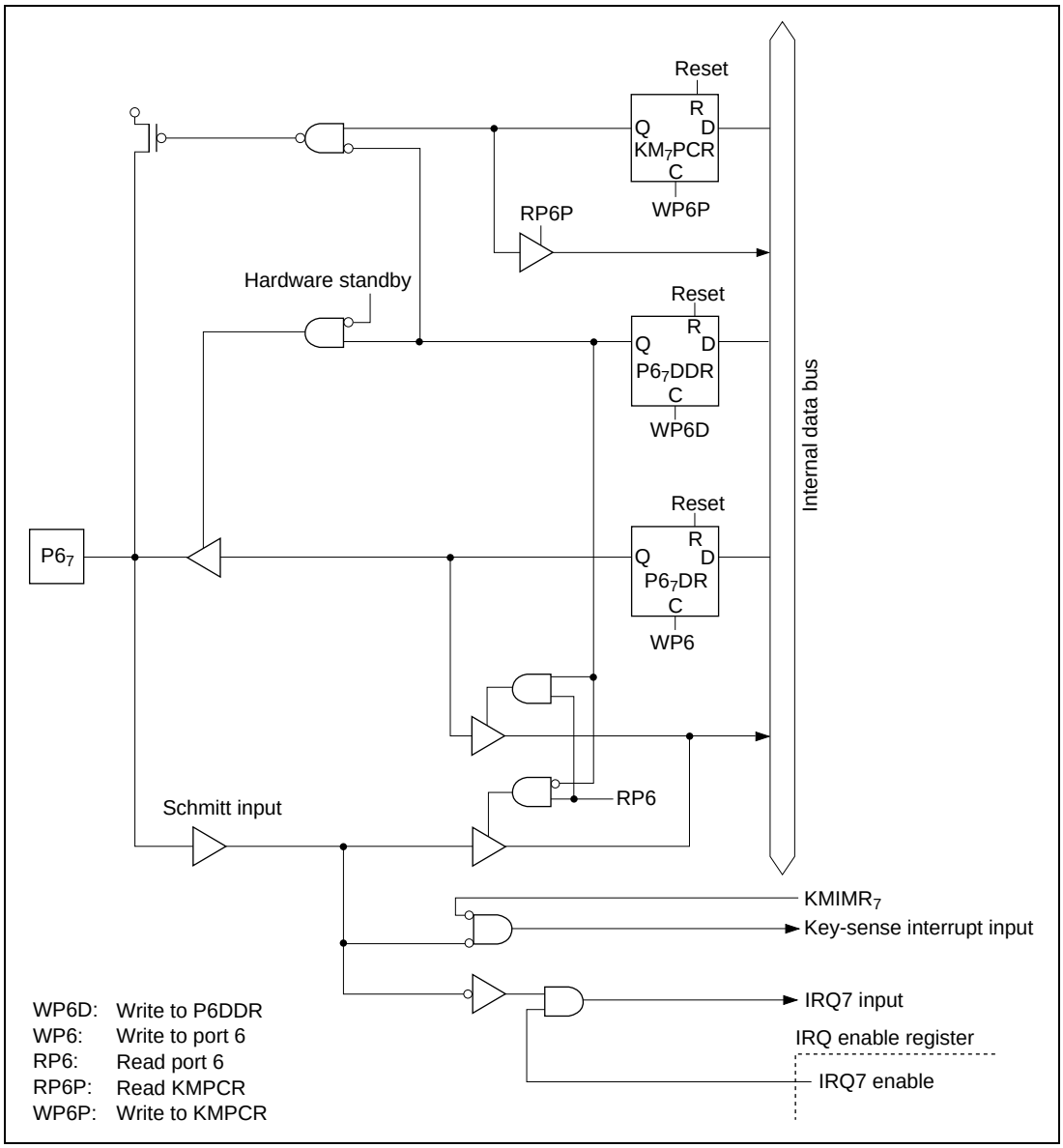


Figure C-6 (d) Port 6 Block Diagram (Pin $P6_7$) [H8/3534]

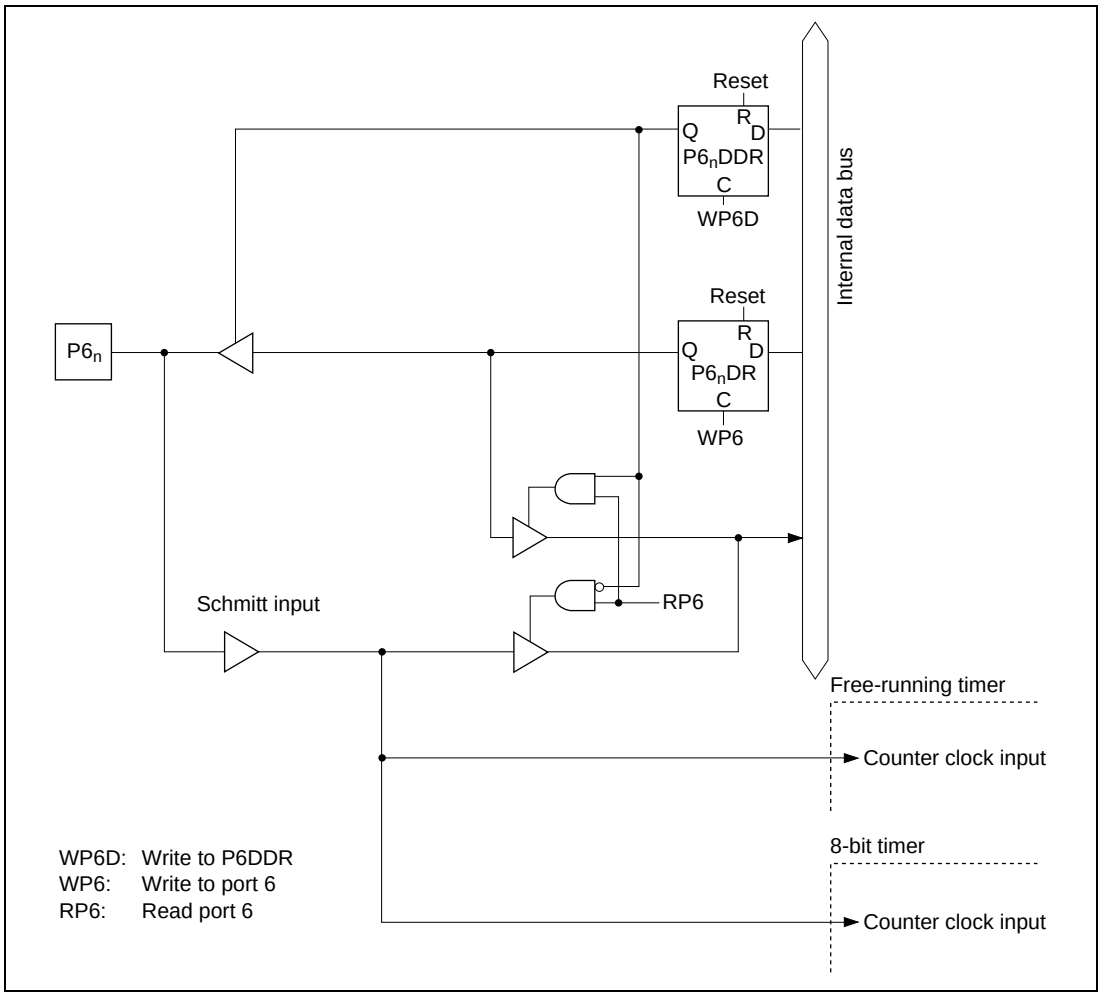


Figure C-6 (e) Port 6 Block Diagram (Pin $P6_0$) [H8/3522]

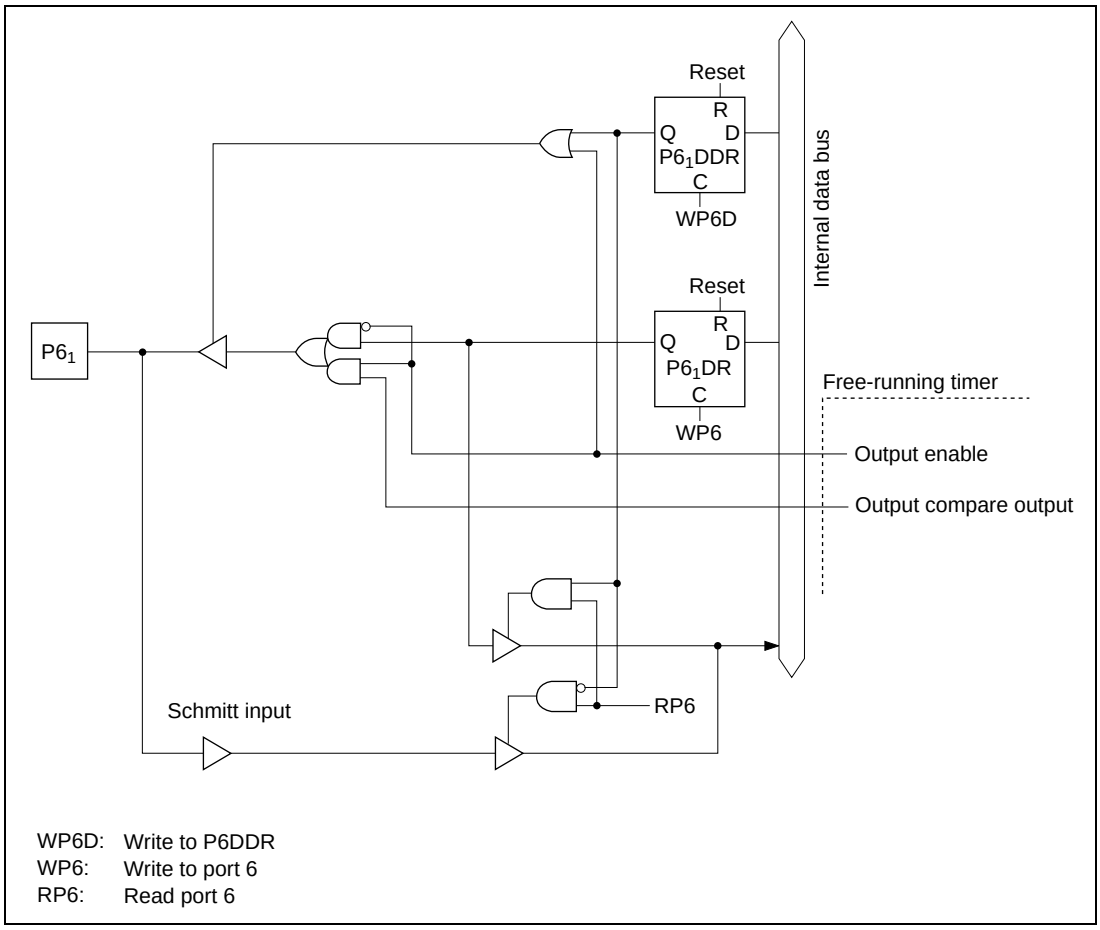


Figure C-6 (f) Port 6 Block Diagram (Pin P6$_1$) [H8/3522]

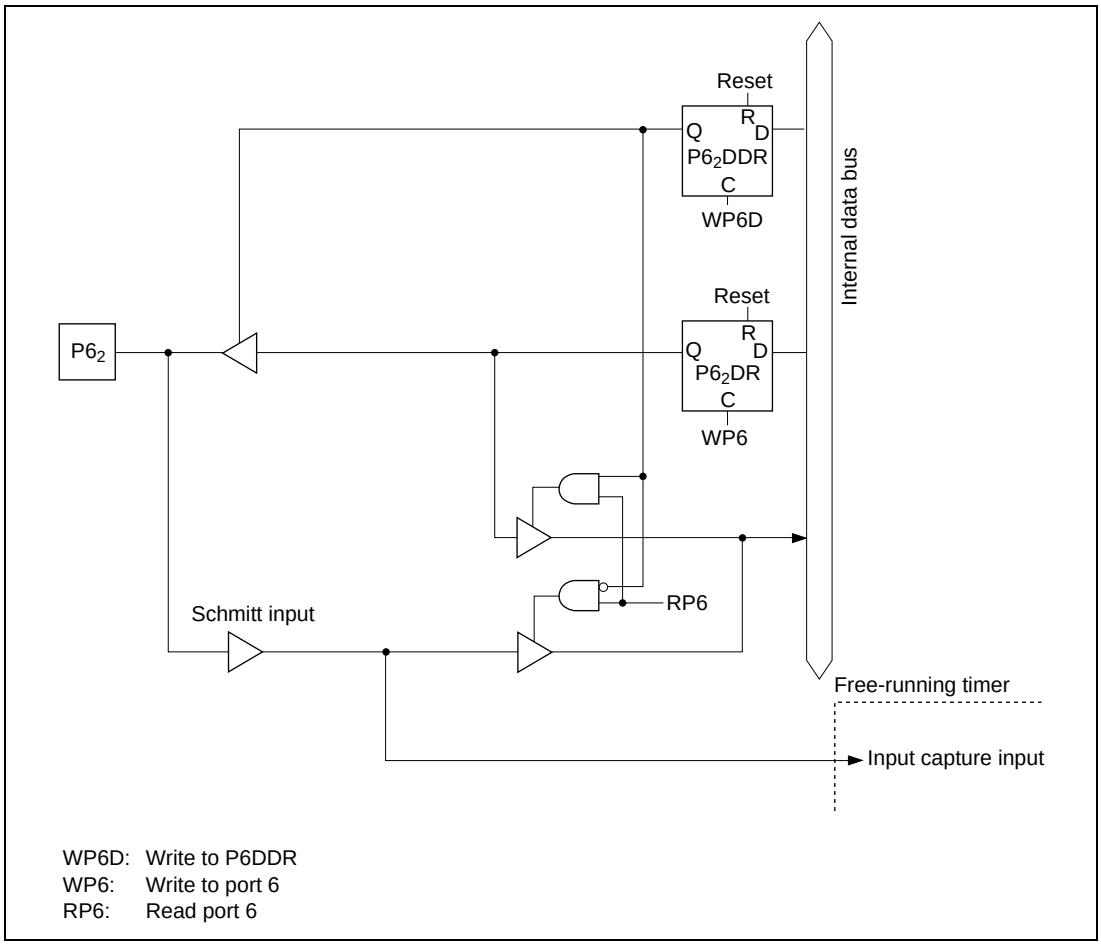


Figure C-6 (g) Port 6 Block Diagram (Pin $P6_2$) [H8/3522]

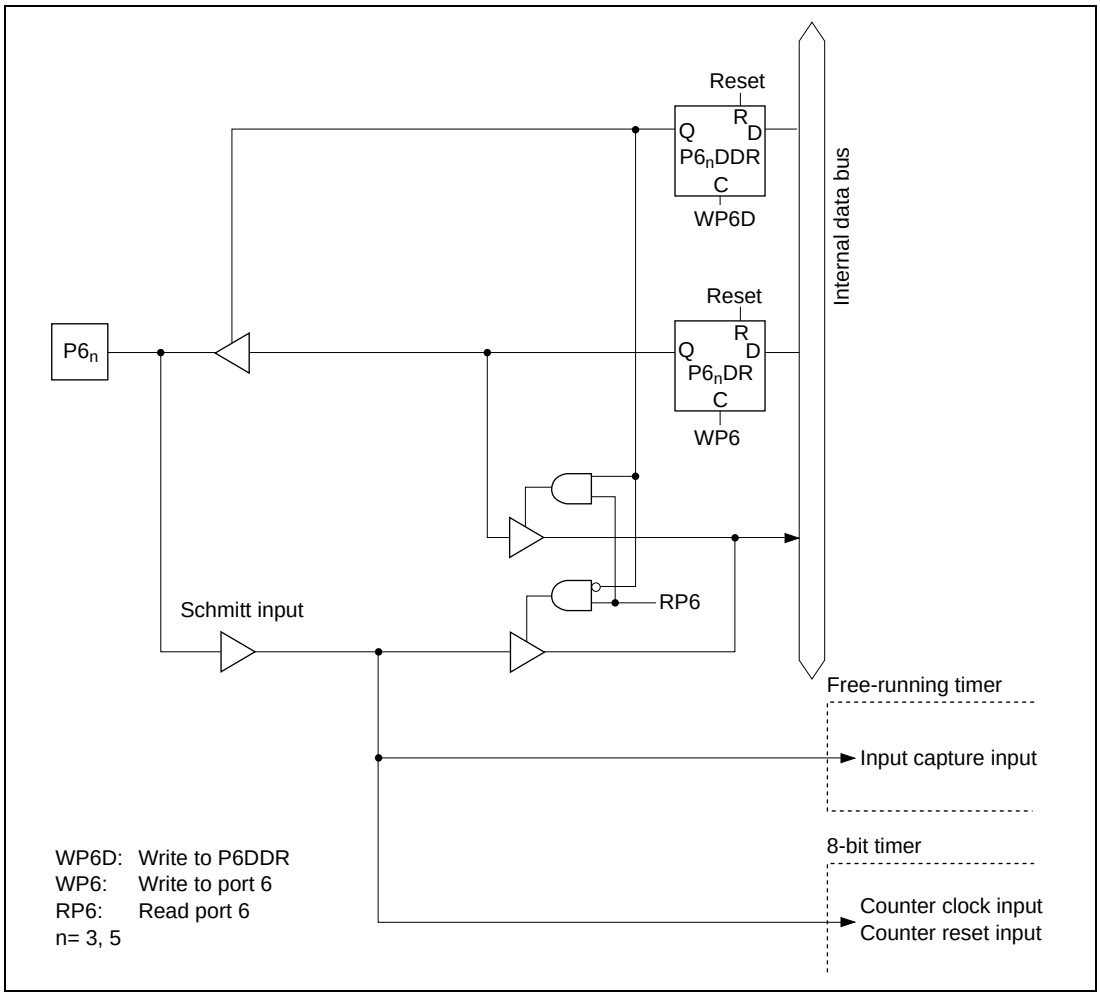


Figure C-6 (h) Port 6 Block Diagram (Pins $P6_3$, $P6_5$) [H8/3522]

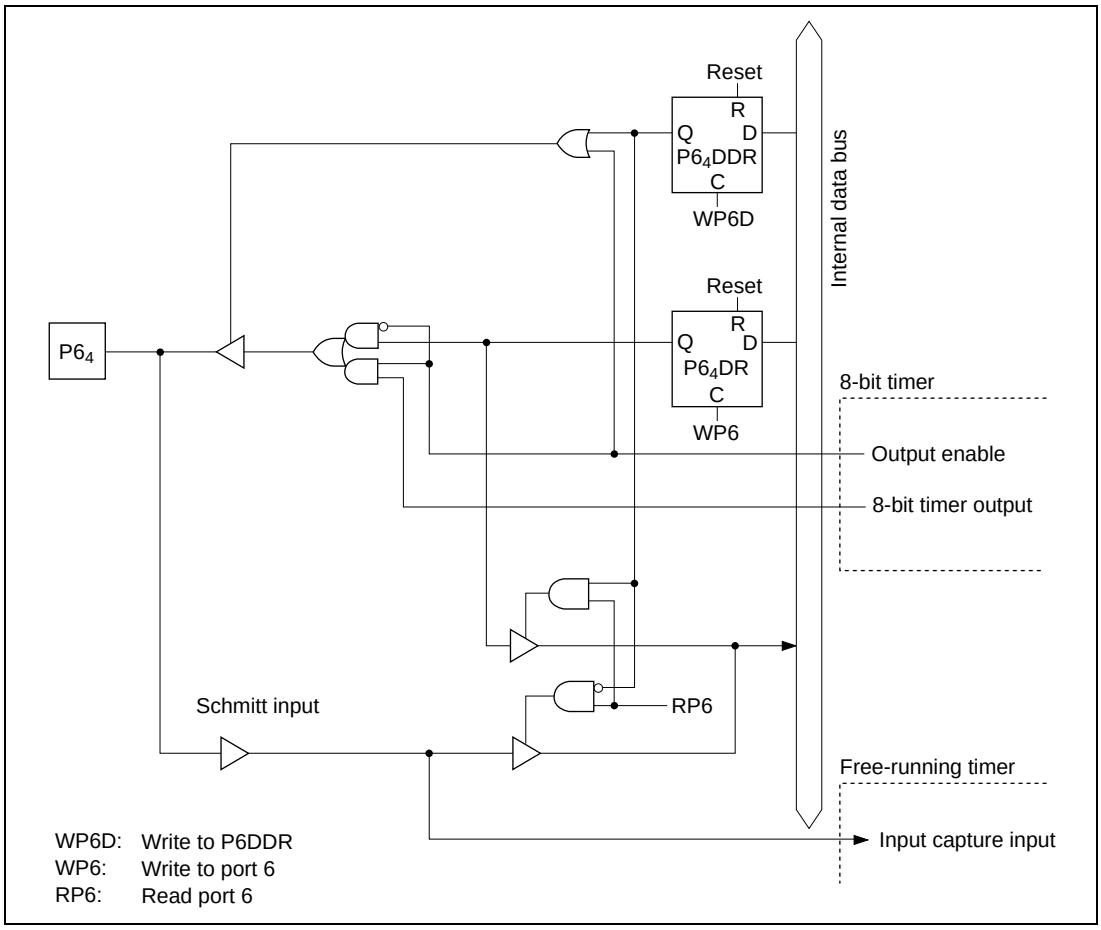


Figure C-6 (i) Port 6 Block Diagram (Pins P6$_4$) [H8/3522]

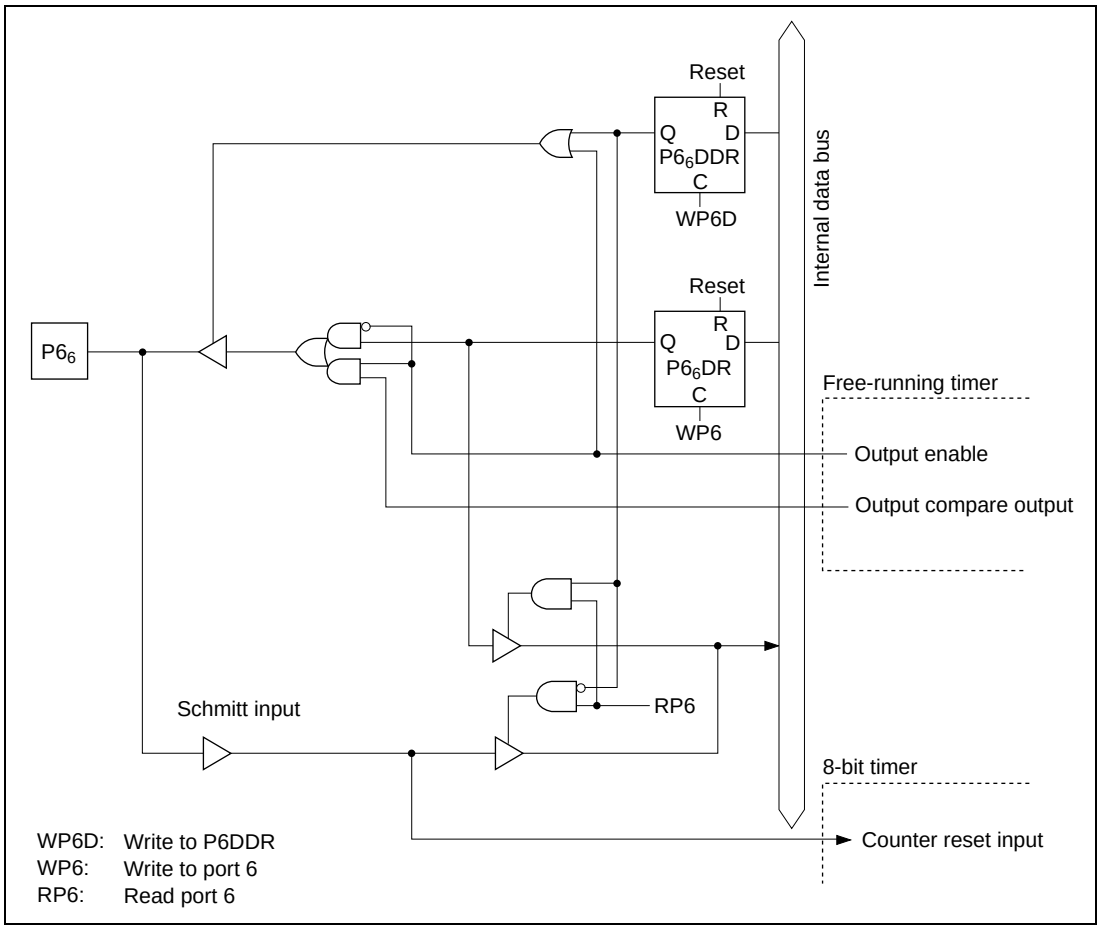


Figure C-6 (j) Port 6 Block Diagram (Pins $P6_6$) [H8/3522]

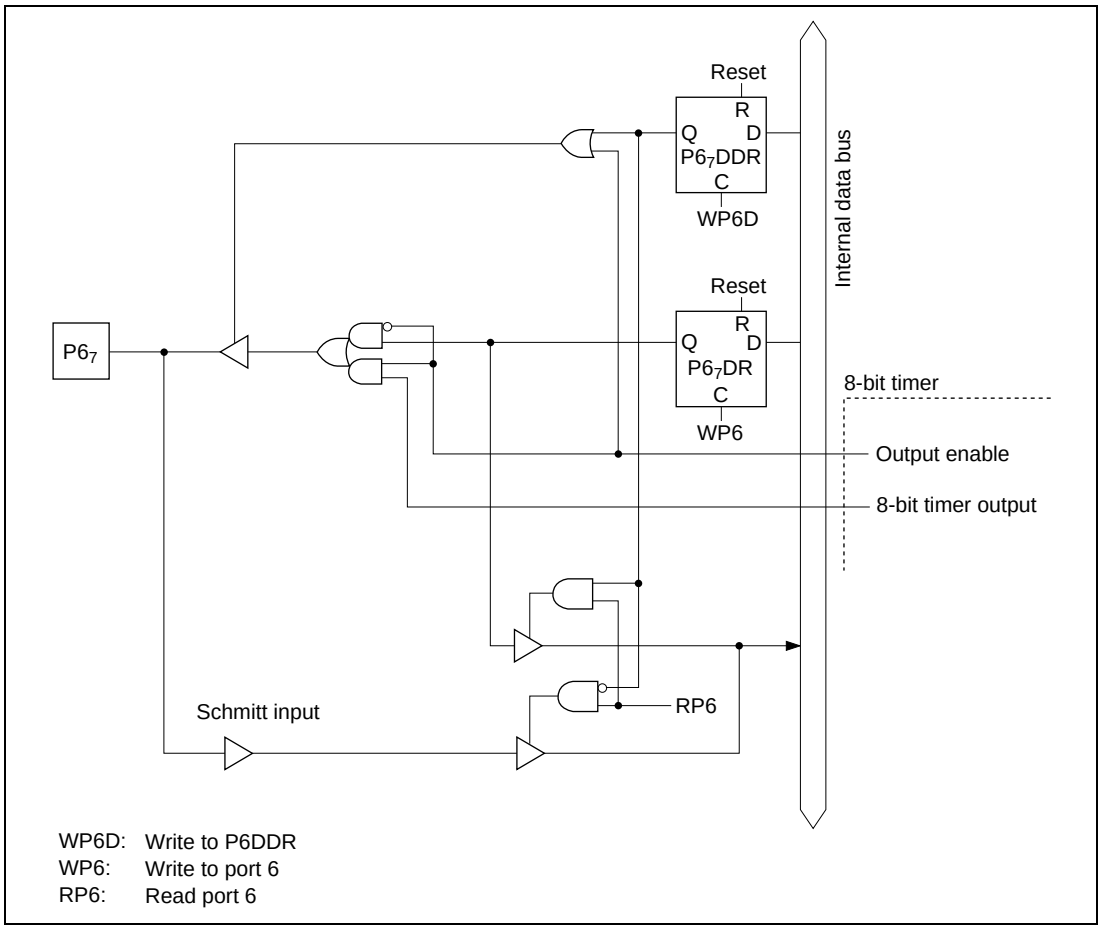


Figure C-6 (k) Port 6 Block Diagram (Pins $P6_7$) [H8/3522]

C.7 Port 7 Block Diagram

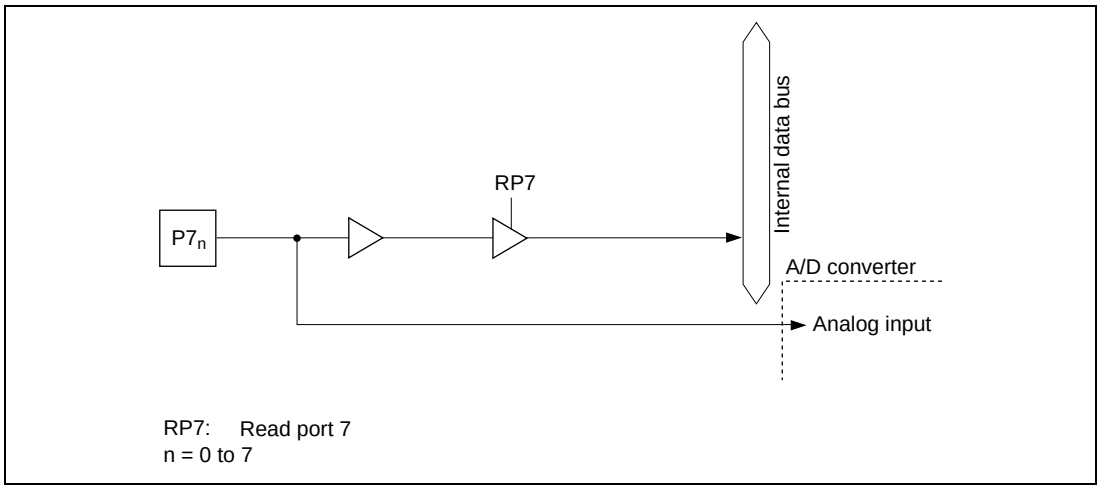


Figure C-7 Port 7 Block Diagram (Pins $P7_0$ to $P7_7$)

C.8 Port 8 Block Diagrams

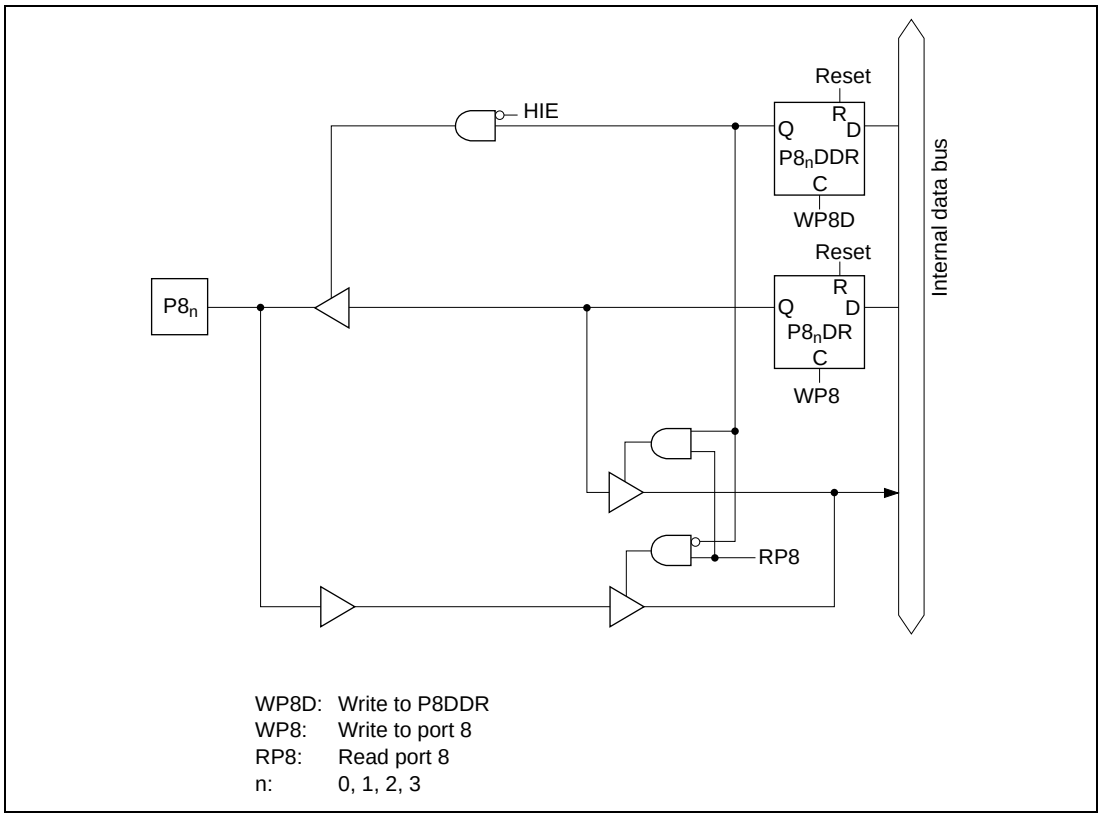


Figure C-8 (a) Port 8 Block Diagram (Pins $P8_0$ to $P8_3$) [H8/3534]

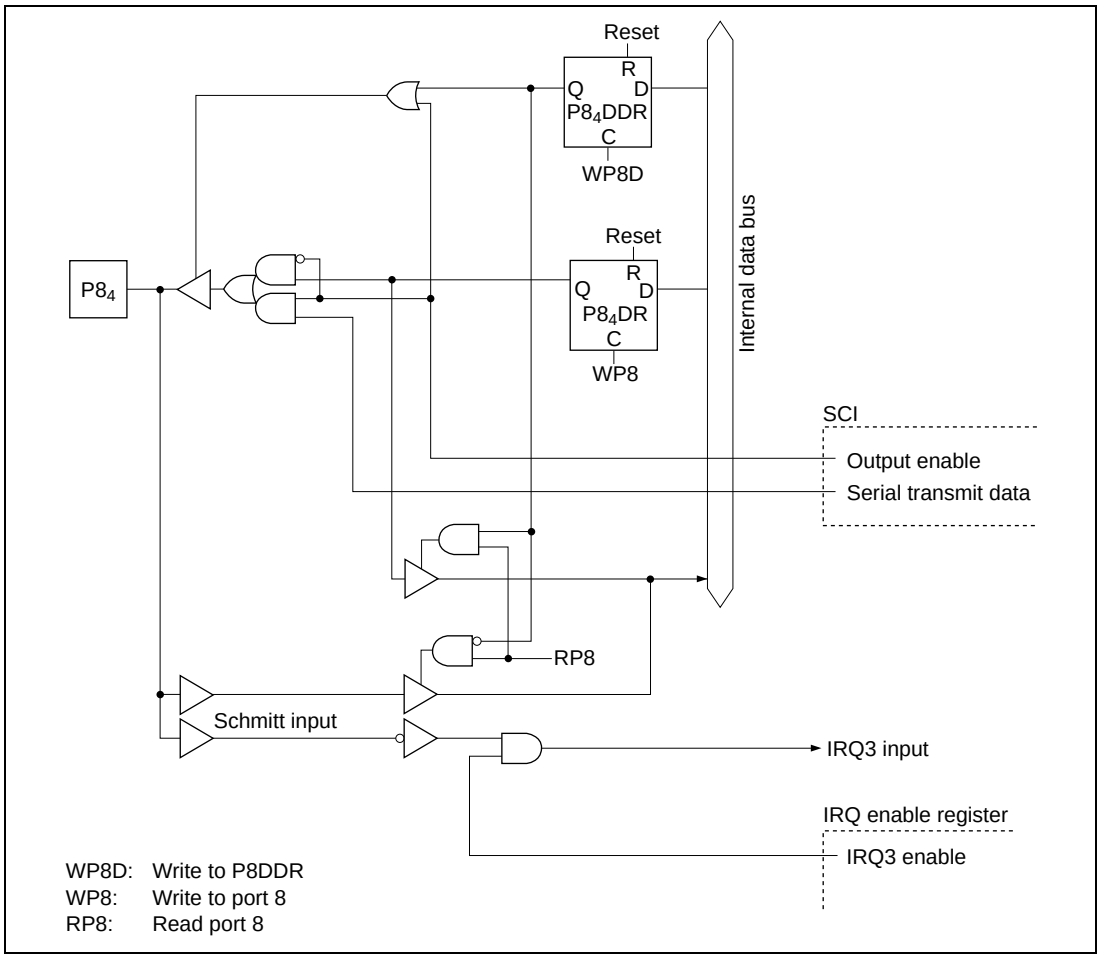


Figure C-8 (b) Port 8 Block Diagram (Pin P8$_4$) [H8/3534]

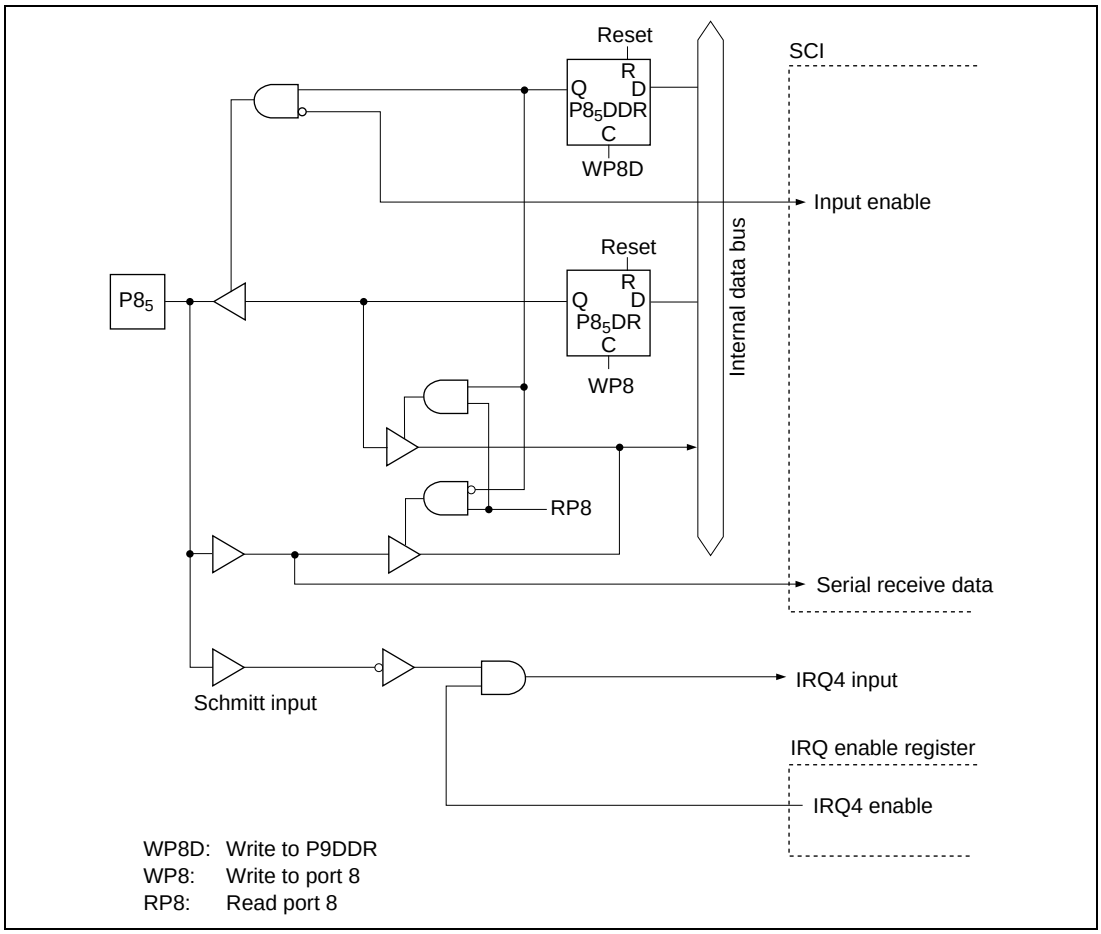


Figure C-8 (c) Port 8 Block Diagram (Pin P8$_5$) [H8/3534]

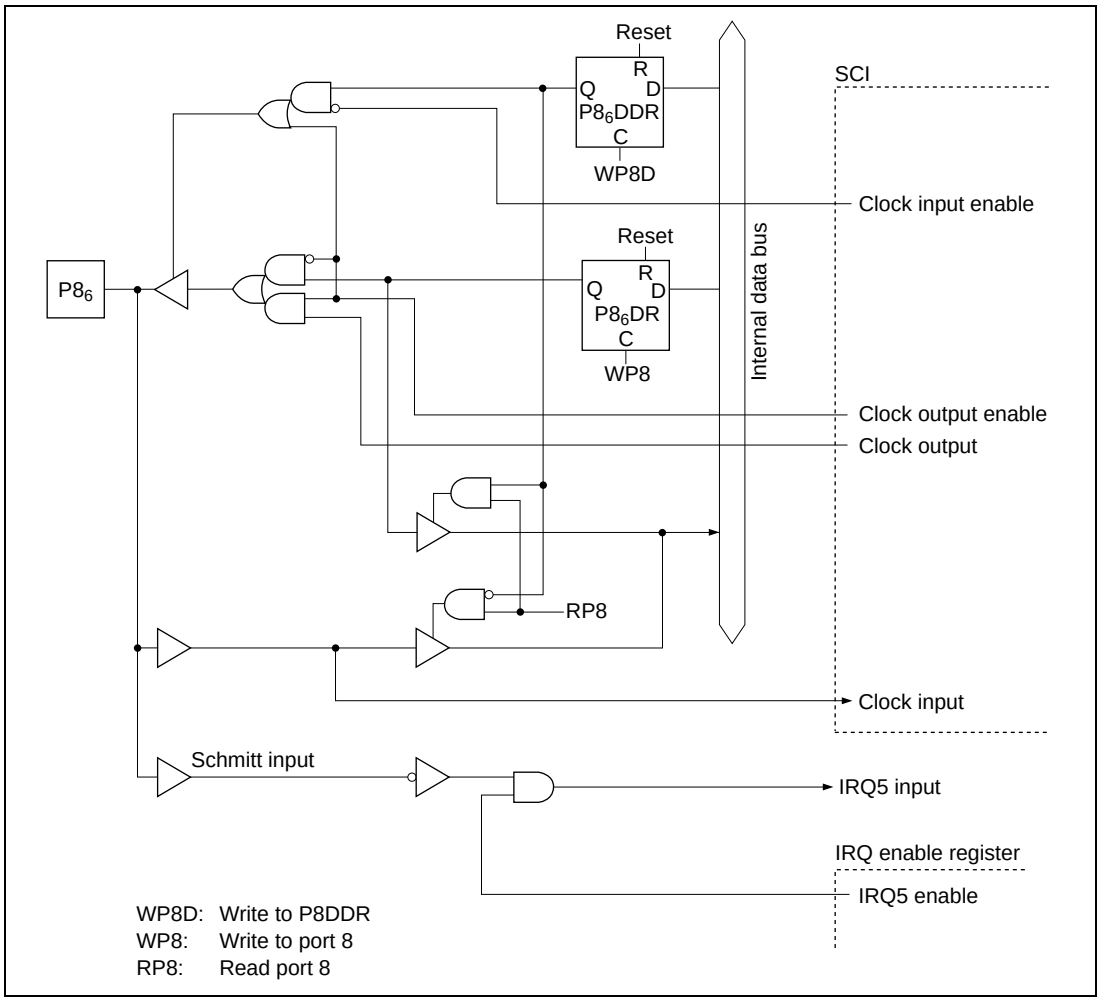


Figure C-8 (d) Port 8 Block Diagram (Pin $P8_6$) [H8/3534]

C.9 Port 9 Block Diagrams [H8/3534] and Port 4 Block Diagrams [H8/3522]

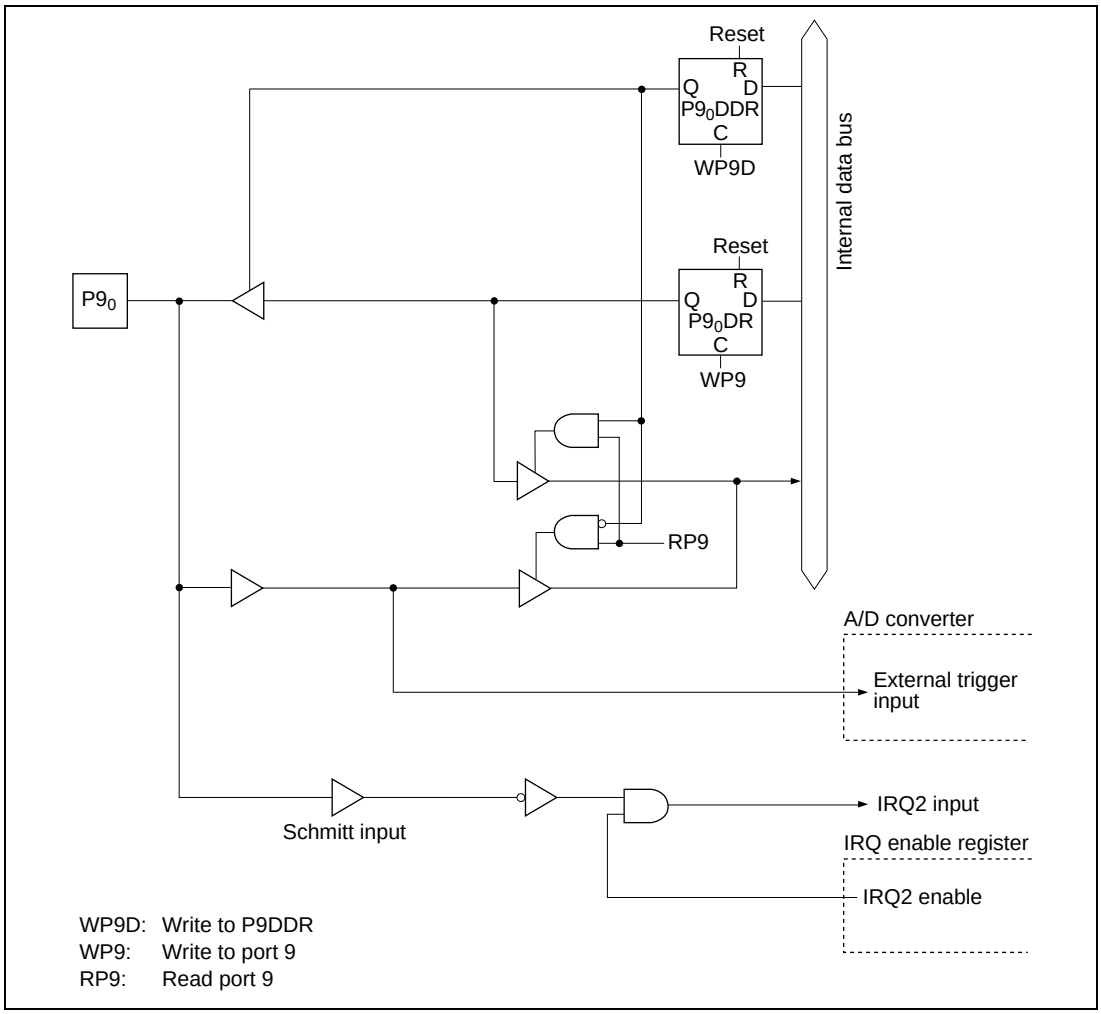


Figure C-9 (a) Port 9 Block Diagram (Pin P9$_0$) [H8/3534]
Port 4 Block Diagram (Pin P4$_0$) [H8/3522]

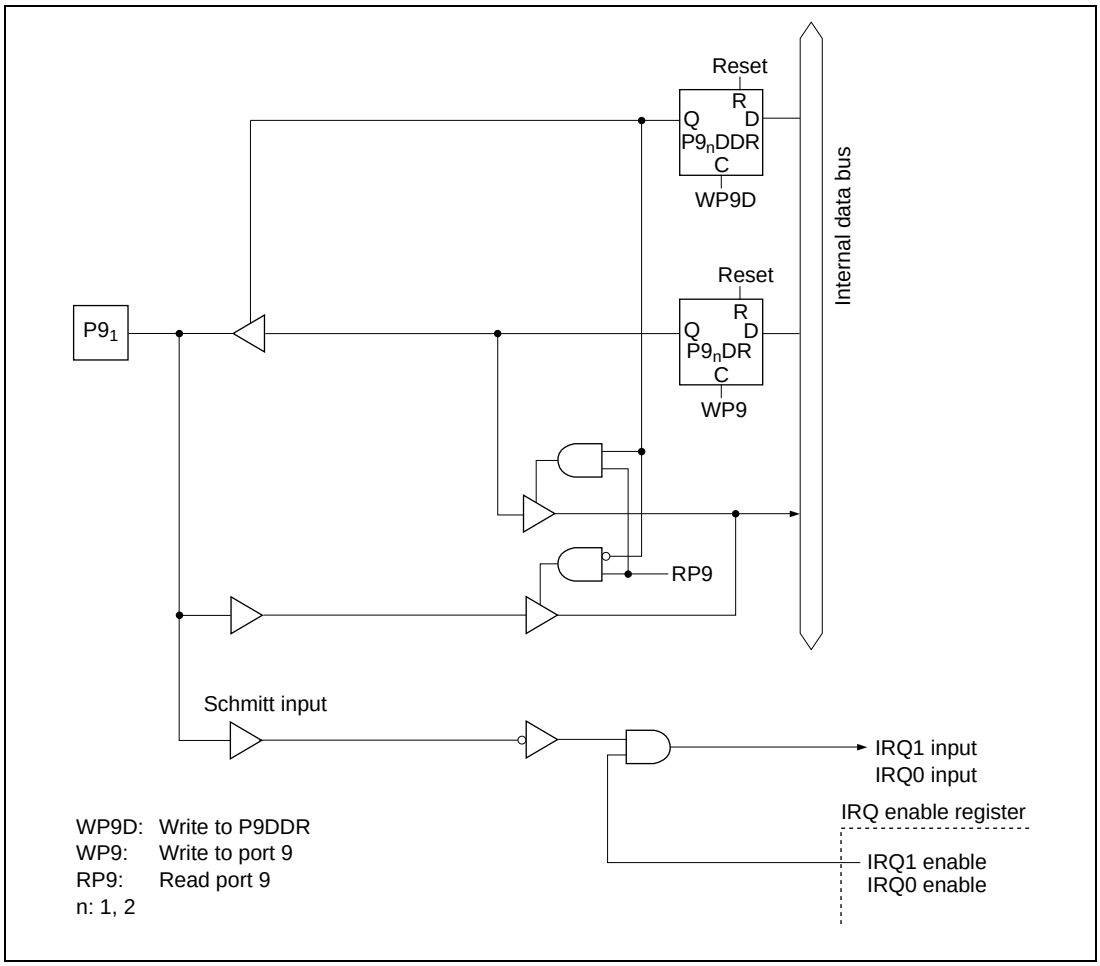


Figure C-9 (b) Port 9 Block Diagram (Pins $P9_1$, $P9_2$) [H8/3534]
Port 4 Block Diagram (Pins $P4_1$, $P4_2$) [H8/3522]

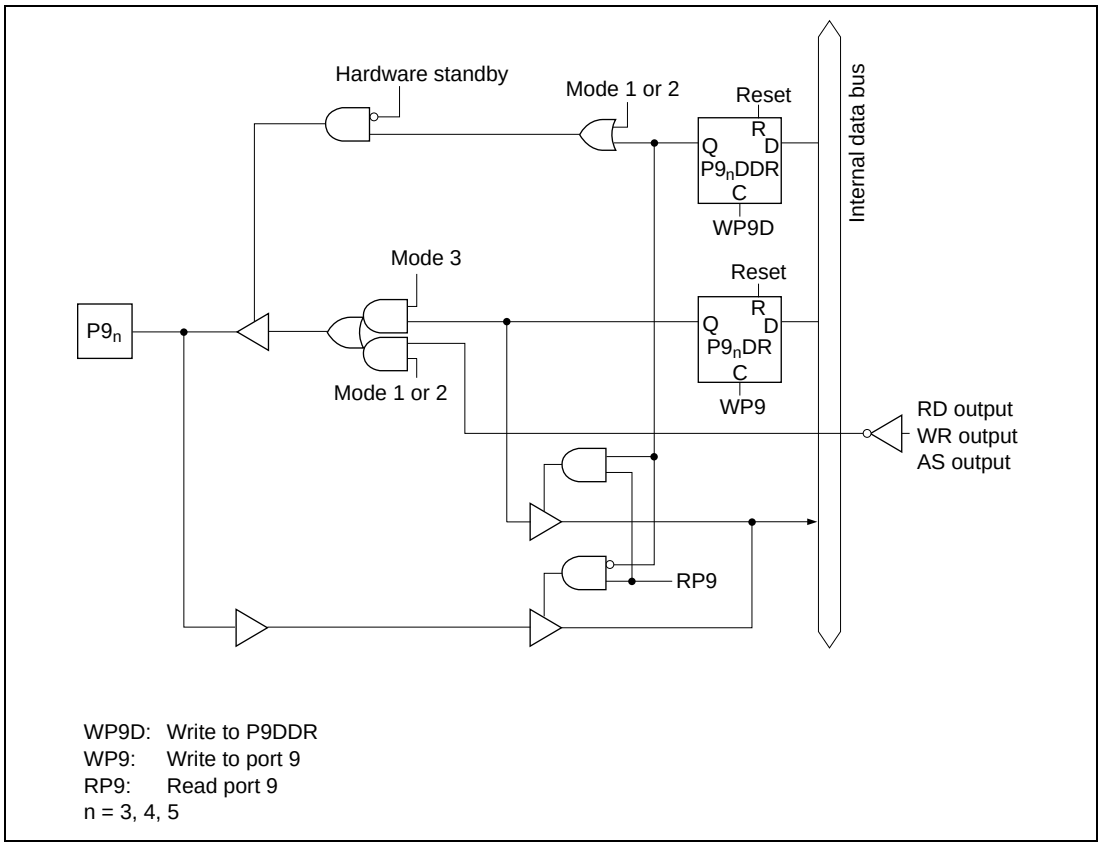


Figure C-9 (c) Port 9 Block Diagram (Pins $P9_3$, $P9_4$, $P9_5$) [H8/3534]
Port 4 Block Diagram (Pins $P4_3$, $P4_4$, $P4_5$) [H8/3522]

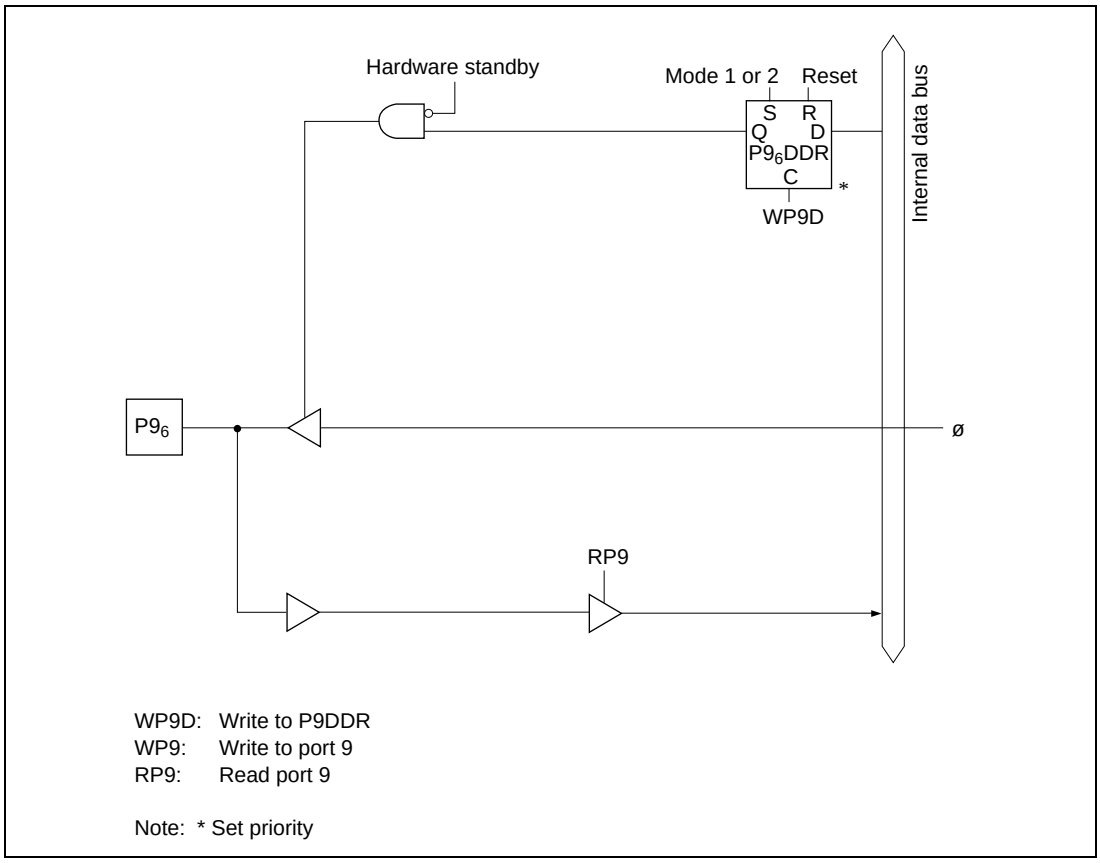
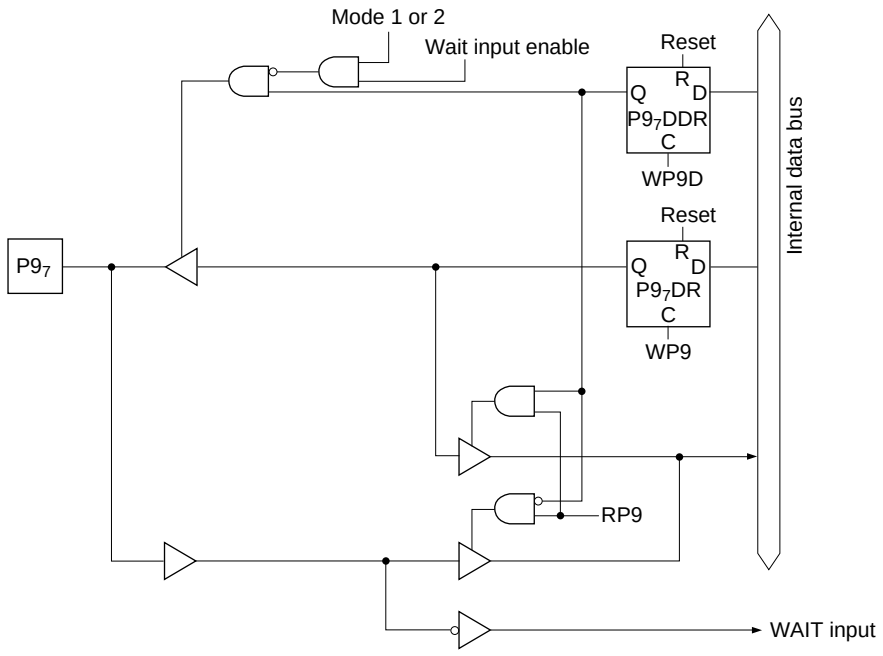


Figure C-9 (d) Port 9 Block Diagram (Pin $P9_6$) [H8/3534]
Port 4 Block Diagram (Pin $P4_6$) [H8/3522]



WP9D: Write to P9DDR
WP9: Write to port 9
RP9: Read port 9

Note: For a block diagram when the SDA pin function is selected, see section 13, I^2C Bus Interface.

Figure C-9 (e) Port 9 Block Diagram (Pin $P9_7$) [H8/3534]
Port 4 Block Diagram (Pin $P4_7$) [H8/3522]

Appendix D Port States in Each Mode

Table D-1 Port States

Pin Name	Mode	Reset	Hardware Standby	Software Standby	Sleep Mode	Normal Operation
$P1_7$ to $P1_0$	1	Low	3-state	Low	Prev. state	A_7 to A_0
A_7 to A_0	2	3-state		Low if DDR = 1, prev. state if DDR = 0	(Addr. output pins: last address accessed	Addr. output or input port
	3			Prev. state		I/O port
$P2_7$ to $P2_0$	1	Low	3-state	Low	Prev. state	A_{15} to A_8
A_{15} to A_8	2	3-state		Low if DDR = 1, prev. state if DDR = 0	(Addr. output pins: last address accessed)	Addr. output or input port
	3			Prev. state		I/O port
$P3_7$ to $P3_0$	1	3-state	3-state	3-state	3-state	D_7 to D_0
D_7 to D_0	2					
	3			Prev. state	Prev. state	I/O port
$P4_7$ to $P4_0$	1	3-state	3-state	Prev. state*	Prev. state	I/O port
[H8/3534]	2					
	3					
$P5_2$ to $P5_0$	1	3-state	3-state	Prev. state*	Prev. state	I/O port
	2					
	3					

Notes: 1. 3-state: High-impedance state

2. Prev. state: Previous state. Input ports are in the high-impedance state and output ports hold their previous level. (For ports 1 to 3 and port 6 (H8/3534 only), the MOS pull-up is on if DDR = 0 and PCR = 1.)

* On-chip supporting modules are initialized, so these pins revert to I/O ports according to the DDR and DR bits.

Table D-1 Port States (cont)

Pin Name	Mode	Reset	Hardware Standby	Software Standby	Sleep Mode	Normal Operation
$P6_7$ to $P6_0$	1	3-state	3-state	Prev. state*	Prev. state	I/O port
	2					
	3					
$P7_7$ to $P7_0$	1	3-state	3-state	3-state	3-state	Input port
	2					
	3					
$P8_6$ to $P8_0$ [H8/3534 only]	1	3-state	3-state	Prev. state*	Prev. state	I/O port
	2					
	3					
$P9_7$/$\overline{WAIT}$ [H8/3534]	1	3-state	3-state	3-state/ prev. state*	3-state/ prev. state	$\overline{WAIT}$ input or I/O port
	2					
$P4_7$/$\overline{WAIT}$ [H8/3522]	3			Prev. state*	Prev. state	I/O port
$P9_6$/ϕ [H8/3534]	1	Clock output	3-state	High	Clock	Clock
$P4_6$/ϕ [H8/3522]	2					
	3	3-state		High if DDR = 1, 3-state if DDR = 0	Clock output if DDR = 1, 3-state if DDR = 0	Clock output if DDR = 1, input port if DDR = 0
$P9_5$ to $P9_3$, [H8/3534]	1	High	3-state	High	High	$\overline{AS}$, $\overline{WR}$, $\overline{RD}$
	2					
$P4_5$ to $P4_3$ [H8/3522] $\overline{AS}$, $\overline{WR}$, $\overline{RD}$	3	3-state		Prev. state	Prev. state	I/O port
$P9_2$ to $P9_0$ [H8/3534]	1	3-state	3-state	Prev. state	Prev. state	I/O port
$P4_2$ to $P4_6$ [H8/3522]	2					
	3					

Notes: 1. 3-state: High-impedance state

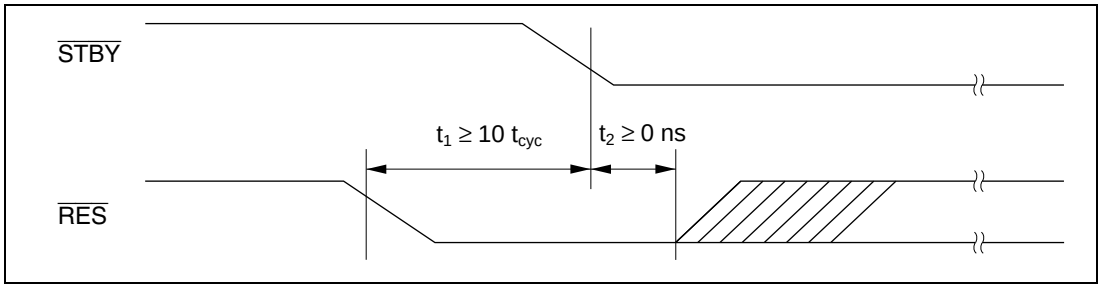
2. Prev. state: Previous state. Input ports are in the high-impedance state and output ports hold their previous level. (For ports 1 to 3 and port 6 (H8/3534 only), the MOS pull-up is on if DDR = 0 and PCR = 1.)

* On-chip supporting modules are initialized, so these pins revert to I/O ports according to the DDR and DR bits.

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

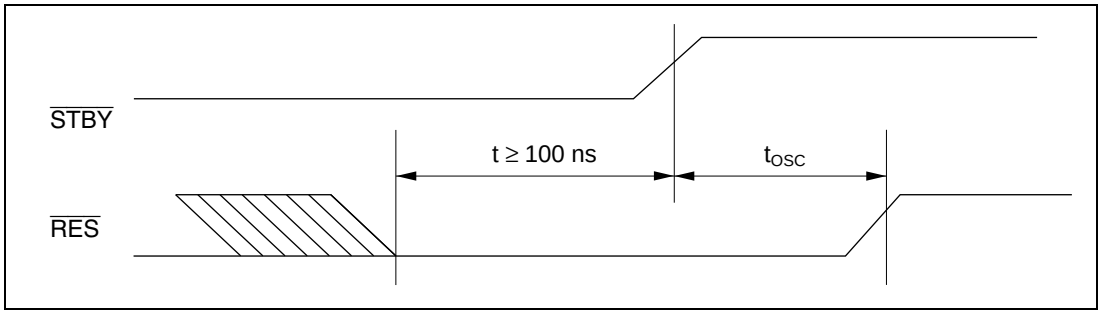
Timing of Transition to Hardware Standby Mode

(1) To retain RAM contents when the RAME bit in SYSCR is set to 1, drive the $\overline{\text{RES}}$ signal low 10 system clock cycles before the $\overline{\text{STBY}}$ signal goes low, as shown below. $\overline{\text{RES}}$ must remain low until $\overline{\text{STBY}}$ goes low (minimum delay from $\overline{\text{STBY}}$ low to $\overline{\text{RES}}$ high: 0 ns).



(2) When the RAME bit in SYSCR is cleared to 0 or when it is not necessary to retain RAM contents, $\overline{\text{RES}}$ does not have to be driven low as in (1).

Timing of Recovery From Hardware Standby Mode: Drive the $\overline{\text{RES}}$ signal low approximately 100 ns before $\overline{\text{STBY}}$ goes high.



Appendix F Product Code Lineup

Table F-1 H8/3534, H8/3522 Product Code Lineup

Product Type			Product Code	Mark Code	Package (Hitachi Package Code)
H8/3534	Mask ROM version	Standard product	HD6433534F	HD6433534(***)F	80-pin QFP (FP-80A)
H8/3522	Mask ROM version	Standard product	HD6433522F	HD6433522(***)F	64-pin QFP (FP-64A)
			HD6433522P	HD6433522(***)P	64-pin shrink DIP (DP-64S)

Note: (***) in mask ROM version codes is the ROM code

Appendix G Package Dimensions

Figure G-1 shows the dimensions of the FP-80A package. Figure G-2 shows the dimensions of the FP-64A package. Figure G-3 shows the dimensions of the DP-64S package.

Unit: mm

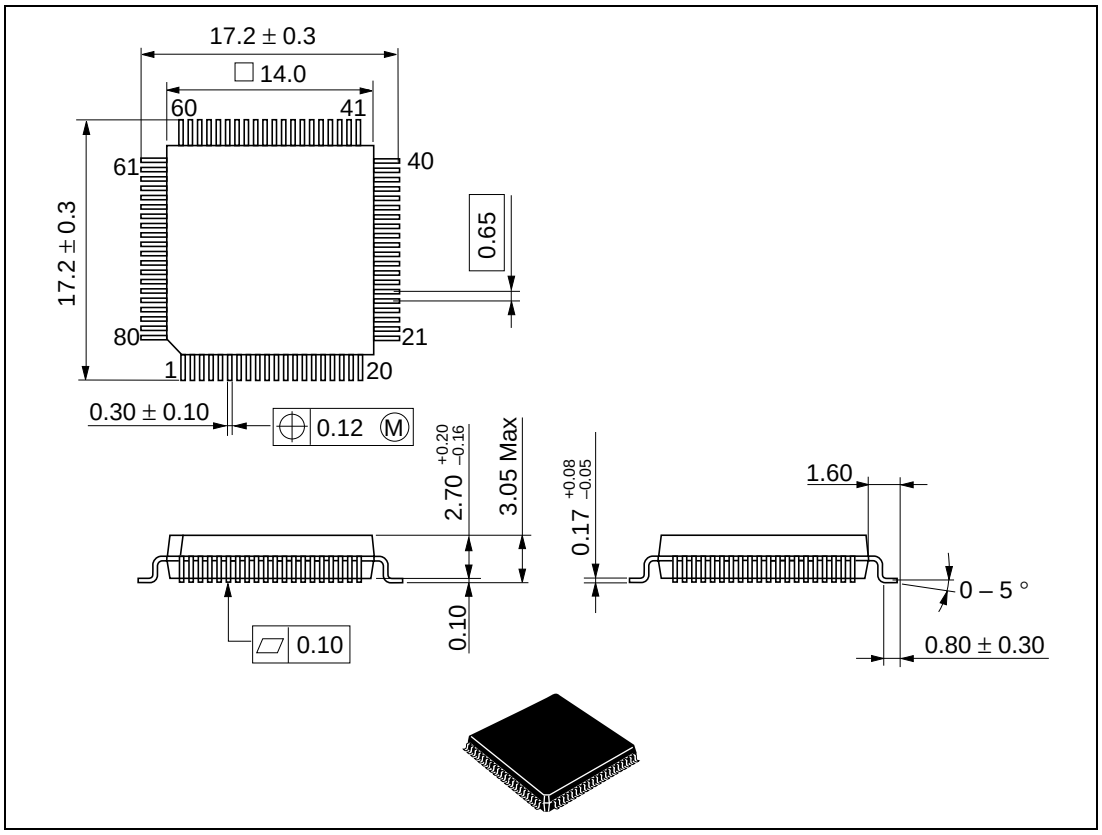


Figure G-1 Package Dimensions (FP-80A)

Unit: mm

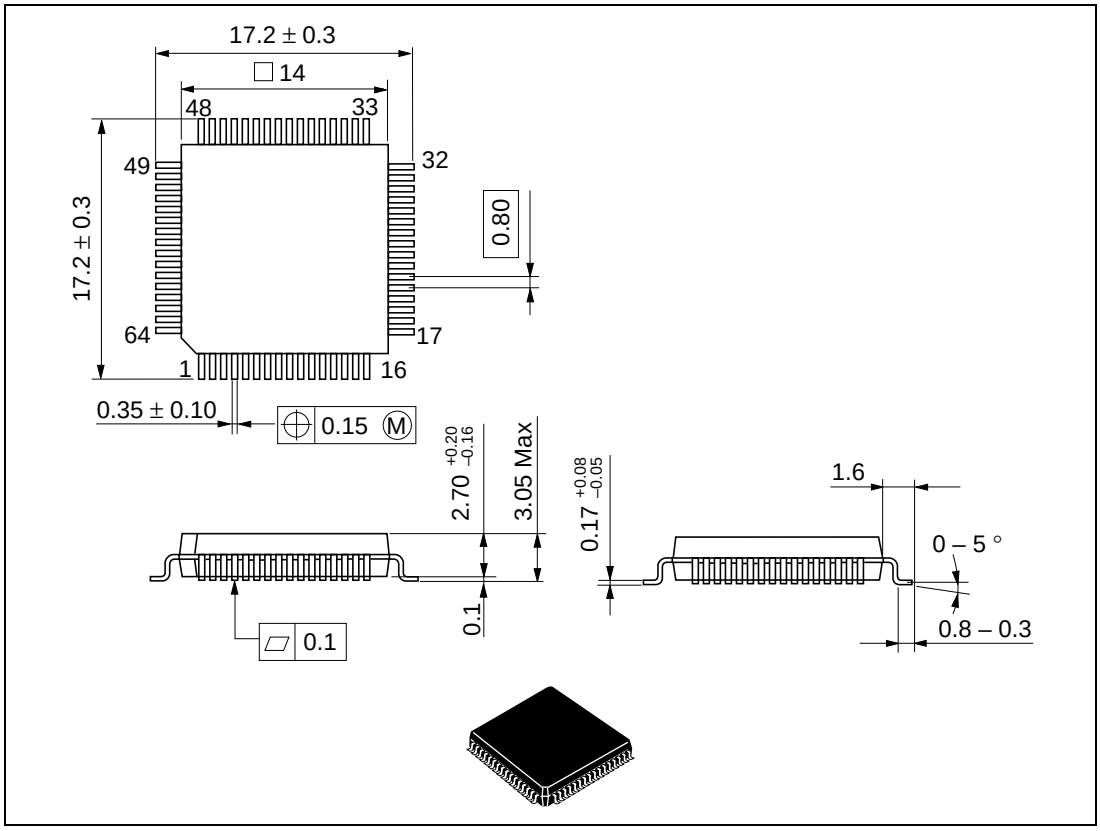


Figure G-2 Package Dimensions (FP-64A)

Unit: mm

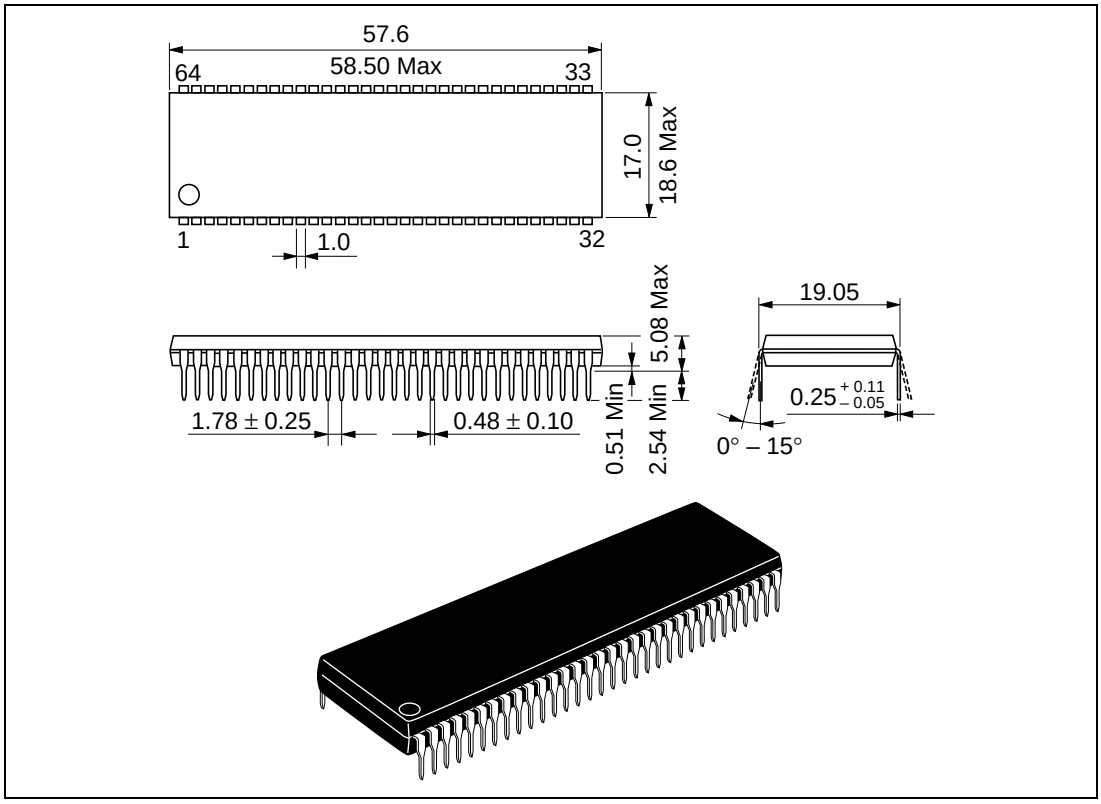


Figure G-3 Package Dimensions (DP-64S)