REJ09B0303-0300

Everywhere you imagine. RENESAS

The revision list can be viewed directly by clicking the title page.

The revision list summarizes the locations of revisions and additions. Details should always be checked by referring to the relevant text.

8

H8/3577 Group, H8/3567 Group
Hardware Manual

Hardware Manual

Renesas 8-Bit Single-Chip Microcomputer
H8 Family/H8/300 Series

H8/3577	HD6433577
	HD6473577
H8/3574	HD6433574
H8/3567	HD6433567
	HD6473567
H8/3564	HD6433564
H8/3567U	HD6433567U
	HD6473567U
H8/3564U	HD6433564U

Rev. 3.00
Revision Date: Mar 17, 2006

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.
The NC (not connected) pins are either not connected to any of the internal circuitry or are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

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

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined.
The states of internal circuits are undefined until full power is supplied throughout the chip and a low level is input on the reset pin. During the period where the states are undefined, the register settings and the output state of each pin are also undefined. Design your system so that it does not malfunction because of processing while it is in this undefined state. For those products which have a reset function, reset the LSI immediately after the power supply has been turned on.

4. Prohibition of Access to Undefined or Reserved Addresses

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

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Main Revisions for This Edition

   The list of revisions is a summary of points that have been revised or added to earlier versions. This does not include all of the revised contents. For details, see the actual locations in this manual.
5. Contents
6. Overview
7. Description of Functional Modules
   - CPU and System-Control Modules
   - On-Chip Peripheral Modules

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

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

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

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

Preface

The H8/3577 Group and H8/3567 Group comprise single-chip microcomputers built around the H8/300 CPU and equipped with on-chip supporting functions required for system configuration. Versions are available with PROM (ZTAT™) or mask ROM as on-chip ROM.

On-chip supporting functions include a16-bit free-running timer (FRT), 8-bit timer (TMR), watchdog timer (WDT), two PWM timers (PWM and PWMX), a serial communication interface (SCI), I^2C bus interface (IIC), A/D converter (ADC), and I/O ports.

The H8/3577 Group comprises 64-pin models with the above supporting functions on-chip. The H8/3567 Group comprises the 42-pin H8/3567 and H8/3564 with fewer PWM, ADC, and I/O port channels, and the 64-pin H8/3567U and H8/3564U with on-chip universal serial bus (USB) hubs and function.

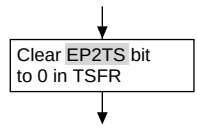
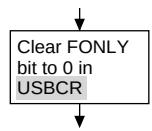
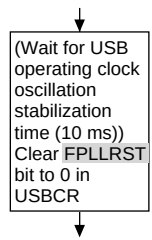
Use of the H8/3577 Group or H8/3567 Group enables compact, high-performance systems to be implemented easily. The comprehensive timer functions and their interconnectability (timer connection facility) make these groups ideal for applications such as PC monitor systems.

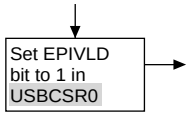
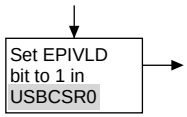
This manual describes the hardware of the H8/3577 Group and H8/3567 Group. Refer to the *H8/300 Series Programming Manual* for a detailed description of the instruction set.

Note: ZTAT (Zero Turn-Around Time) is a trademark of Renesas Technology Corp.

On-Chip Supporting Modules

Group	H8/3577 Group	H8/3567 Group
Product names	H8/3577, H8/3574	H8/3567, H8/3564, H8/3567U, H8/3564U
Universal serial bus (USB)	—	—/Available (H8/3567U, H8/3564U)
8-bit PWM timer (PWM)	×16	×8
14-bit PWM timer (PWMX)	×2	×2
16-bit free-running timer (FRT)	×1	×1
8-bit timer (TMR)	×4	×4
Timer connection	Available	Available
Watchdog timer (WDT)	×1	×1
Serial communication interface (SCI)	×1	×1
I^2C bus interface (IIC)	×2	×2
A/D converter	×8	×4

Item	Page	Revision (See Manual for Details)
All	—	• Notification of change in company name amended (Before) Hitachi, Ltd. → (After) Renesas Technology Corp. • Product naming convention amended (Before) H8/3577 Series → (After) H8/3577 Group (Before) H8/3567 Series → (After) H8/3567 Group
5.2.1 System Control Register (SYSCR)	84	Bit table amended Bit: ··· 4 ··· ··· INTM0 ··· Initial value: 0 Read/Write: R
7.3.5 Operation when OUT Token Is Received (Endpoints 0 and 2) Figure 7.3 (2) Operation when OUT Token Is Received (EP2-OUT: Initial FIFO Full) (cont)	155	Figure amended 
7.3.9 USB Module Startup Sequence Initial Operation Procedures:	164	Description amended 8. After DPLL operation stabilization time, HSRST bit is cleared to 0 by firmware
Figure 7.5 USB Hub Initial Operation Procedure	165	Figure amended 
Figure 7.6 USB Function Initial Operation Procedure	166	Figure amended 

Item	Page	Revision (See Manual for Details)
7.3.9 USB Module Startup Sequence Figure 7.6 USB Function Initial Operation Procedure (cont)	167	Figure amended 
Figure 7.9 USB Function Standalone Mode Upstream Disconnection/ Reconnection	172	Figure amended 
12.2.8 Timer Connection Register S (TCONRS)	296	Table amended

Bit 7	Accessible Registers							
TMRX/Y	H'FFF0	H'FFF1	H'FFF2	H'FFF3	H'FFF4	H'FFF5	H'FFF6	H'FFF7
0 (Initial value)	TCRX (TMRX)	TCSRX (TMRX)	TICRR (TMRX)	TICRF (TMRX)	TCNTX (TMRX)	TCORC (TMRX)	TCORAX (TMRX)	TCORBX (TMRX)
1	TCRY (TMRY)	TCSRY (TMRY)	TCORAY (TMRY)	TCORBY (TMRY)	TCNTY (TMRY)	TISR (TMRY)		

16.3.1 I^2C Bus Data Format Table 16.4 Description of I^2C Bus Data Format Symbols	447	Newly added		
B.1 Addresses	579	Table amended		

Address	Register Name	...	Module Name	Bus Width
H'FFE2	ADDRBH		A/D	8
H'FFE3	ADDRBL			
H'FFE4	ADDRCH			
H'FFE5	ADDRCL			
H'FFE6	ADDRDH			
H'FFE7	ADDRDL			

B.3 Functions UTESTR0, UTESTR1	587	Bit table added
UTESTR2	606	Bit table added

Item	Page	Revision (See Manual for Details)
Appendix G Package Dimensions Figure G.1 DP-64S Package Dimensions	703	Figure replaced
Figure G.2 FP-64A Package Dimensions	704	Figure replaced
Figure G.3 DP-42S Package Dimensions	705	Figure replaced
Figure G.4 FP-44A Package Dimensions	706	Figure replaced

Contents

Section 1 Overview

1.1 Overview

The H8/3577 Group and H8/3567 Group comprise single-chip microcomputers (MCUs) built around the H8/300 CPU and equipped with on-chip supporting functions required for system configuration.

On-chip supporting functions required for system configuration include ROM and RAM, a 16-bit free-running timer (FRT), 8-bit timer (TMR), watchdog timer (WDT), two PWM timers (PWM and PWMX), serial communication interface (SCI), I^2C bus interface (IIC), A/D converter (ADC), and I/O ports. The H8/3577 Group comprises 64-pin MCUs, and the H8/3567 Group 42-pin MCUs, but the H8/3567 Group also includes a 64-pin variation with on-chip universal serial bus (USB) hubs and function.

The on-chip ROM is either PROM (ZTAT) or mask ROM, with a capacity of 56 or 32 kbytes. There is only one operating mode: single-chip mode.

The features of the H8/3577 Group and H8/3567 Group are shown in table 1.1.

Table 1.1 Features

Item	Specifications
CPU	• General-register architecture — Sixteen 8-bit general registers (also usable as eight 16-bit registers) • High-speed operation suitable for realtime control — Maximum operating frequency: 20 MHz/5 V (HD6433564-10: 10 MHz/5 V) — High-speed arithmetic operations 8/16-bit register-register add/subtract: 0.1 μs (20-MHz operation) 8 × 8-bit register-register multiply: 0.7 μs (20-MHz operation) 16 ÷ 8-bit register-register divide: 0.7 μs (20-MHz operation) • Instruction set suitable for high-speed operation — 2-byte or 4-byte instruction length — Register-register basic operations — Memory-register data transfer by MOV instruction • Instructions with special features — Multiply instructions (8 bits × 8 bits) — Divide instructions (16 bits ÷ 8 bits) — Bit-accumulator instructions — Bit position specifiable by means of register indirect specification
16-bit free-running timer (FRT), 1 channel	• One 16-bit free-running counter (usable for external event counting) • Two output compare outputs • Four input capture inputs (with buffer operation capability)
8-bit timer (TMR), 2 channels (TMR0, TMR1)	Each channel has: • One 8-bit up-counter (usable for external event counting) • Two timer constant registers • The two channels can be connected

Item	Specifications
Timer connection and 8-bit timer (TMR), 2 channels (TMRX, TMRY)	Input/output and FRT, TMR1, TMRX, TMRY can be interconnected • Measurement of input signal or frequency-divided waveform pulse width and cycle (FRT, TMR_1) • Output of waveform obtained by modification of input signal edge (FRT, TMR_1) • Determination of input signal duty cycle (TMRX) • Output of waveform synchronized with input signal (FRT, TMRX, TMRY) • Automatic generation of cyclical waveform (FRT, TMRY)
Watchdog timer (WDT), 1 channel	• Watchdog timer or interval timer function selectable
8-bit PWM timer (PWM)	• Maximum of 16 (H8/3577 Group) or 8 (H8/3567 Group) outputs • Pulse duty cycle settable from 0 to 100% • Resolution: 1/256 • 1.25 MHz maximum carrier frequency (20-MHz operation)
14-bit PWM timer (PWMX)	• Maximum of 2 outputs • Resolution: 1/16384 • 312.5 kHz maximum carrier frequency (20-MHz operation)
Serial communi-cation interface (SCI), 1 channel (SCI0)	• Asynchronous mode or synchronous mode selectable • Multiprocessor communication function
A/D converter	• Resolution: 10 bits • Input: 8 channels (H8/3577 Group) 4 channels (H8/3567 Group) • High-speed conversion : 6.7 µs minimum conversion time (20-MHz operation) • Single or scan mode selectable • Sample-and-hold function • A/D conversion can be activated by external trigger or timer trigger
I/O ports	• Input/output pins: 43 (H8/3577 Group, H8/3567 Group models with on-chip USB) or 27 (H8/3567 Group) • Input-only pins: 8 (H8/3577 Group) or 4 (H8/3567 Group)

Item	Specifications
Memory	• PROM or mask ROM • High-speed static RAM

Product Code	ROM	RAM
H8/3577, H8/3567, H8/3567U	56 kbytes	2 kbytes
H8/3574, H8/3564, H8/3564U	32 kbytes	2 kbytes

Interrupt controller	• Four external interrupt pins (NMI, $\overline{IRQ}_0$ to $\overline{IRQ}_2$) • 26 internal interrupt sources (H8/3567U Group: 30 sources)
Power-down state	• Medium-speed mode • Sleep mode • Module stop mode • Software standby mode • Hardware standby mode
Clock pulse generator	• Built-in duty correction circuit
Packages	• 64-pin plastic DIP (DP-64S) • 64-pin plastic QFP (FP-64A) • 42-pin plastic DIP (DP-42S) • 44-pin plastic QFP (FP-44A)
I^2C bus interface (IIC), 2 channels	• Conforms to Philips I^2C bus interface standard • Single master mode/slave mode • Arbitration lost condition can be identified • Supports two slave addresses
Universal serial bus interface (USB) [H8/3567U, H8/3564U]	• Comprises five downstream hubs and one function (four sets of downstream pins) • Three-endpoint monitor device class function EP0: For USB control EP1, EP2: For monitor control • Supports 12 Mbps high-speed transfer mode • Built-in 12 MHz clock pulse generator and 4X multiplication circuit • Built-in bus driver/receiver (requires 3.3 V analog power supply)

Item	Specifications

Product lineup

Group	Product Code		ROM/RAM (Bytes)	Packages
	Mask ROM Version	ZTAT Version		
H8/3577	HD6433577	HD6473577	56 k/2 k	DP-64S, FP-64A
	HD6433574	—	32 k/2 k	
H8/3567	HD6433567	HD6473567	56 k/2 k	DP-42S, FP-44A
	HD6433564-20	—	32 k/2 k	
	HD6433564-10	—	32 k/2 k	DP-42S
	HD6433567U	HD6473567U	56 k/2 k	DP-64S, FP-64A
	HD6433564U	—	32 k/2 k	

1.2 Internal Block Diagrams

Figures 1.1 and 1.2 show internal block diagrams of the H8/3577 Group and H8/3567 Group.

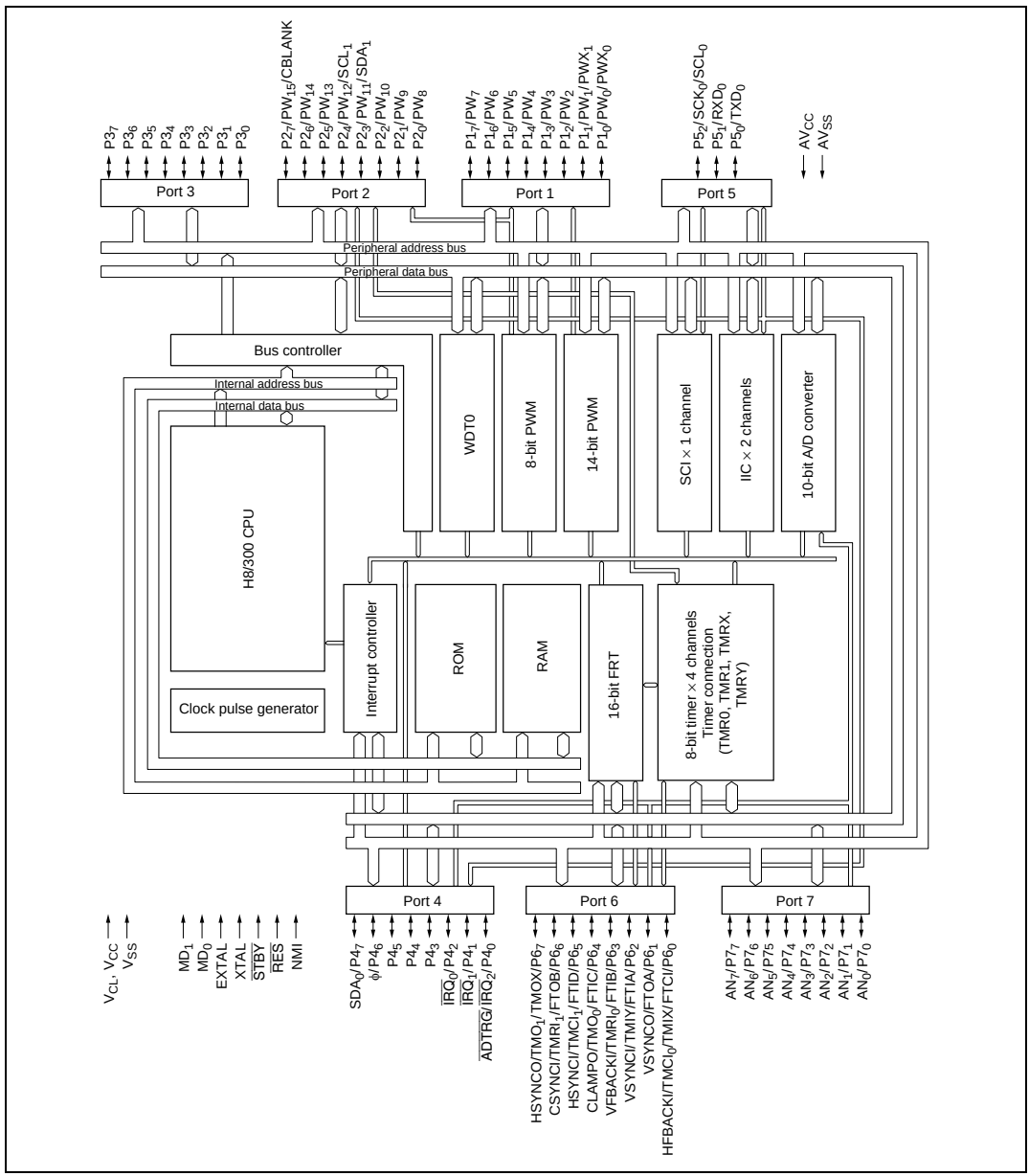


Figure 1.1 Internal Block Diagram of H8/3577 Group

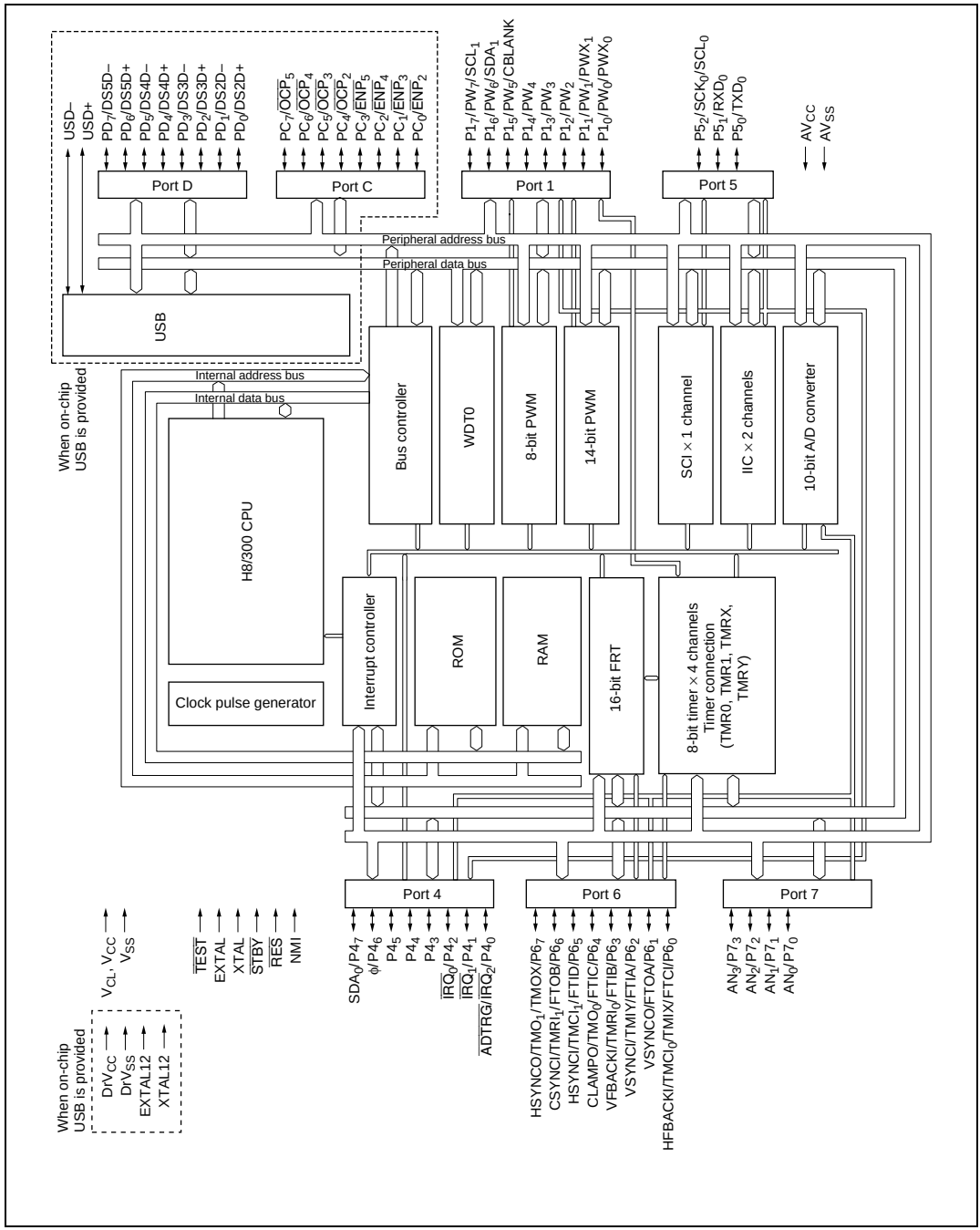


Figure 1.2 Internal Block Diagram of H8/3567 Group

1.3 Pin Arrangement and Functions

1.3.1 Pin Arrangement

The pin arrangements of the H8/3577 Group are shown in figures 1.3 and 1.4, and those of the H8/3567 Group in figures 1.5 to 1.8.

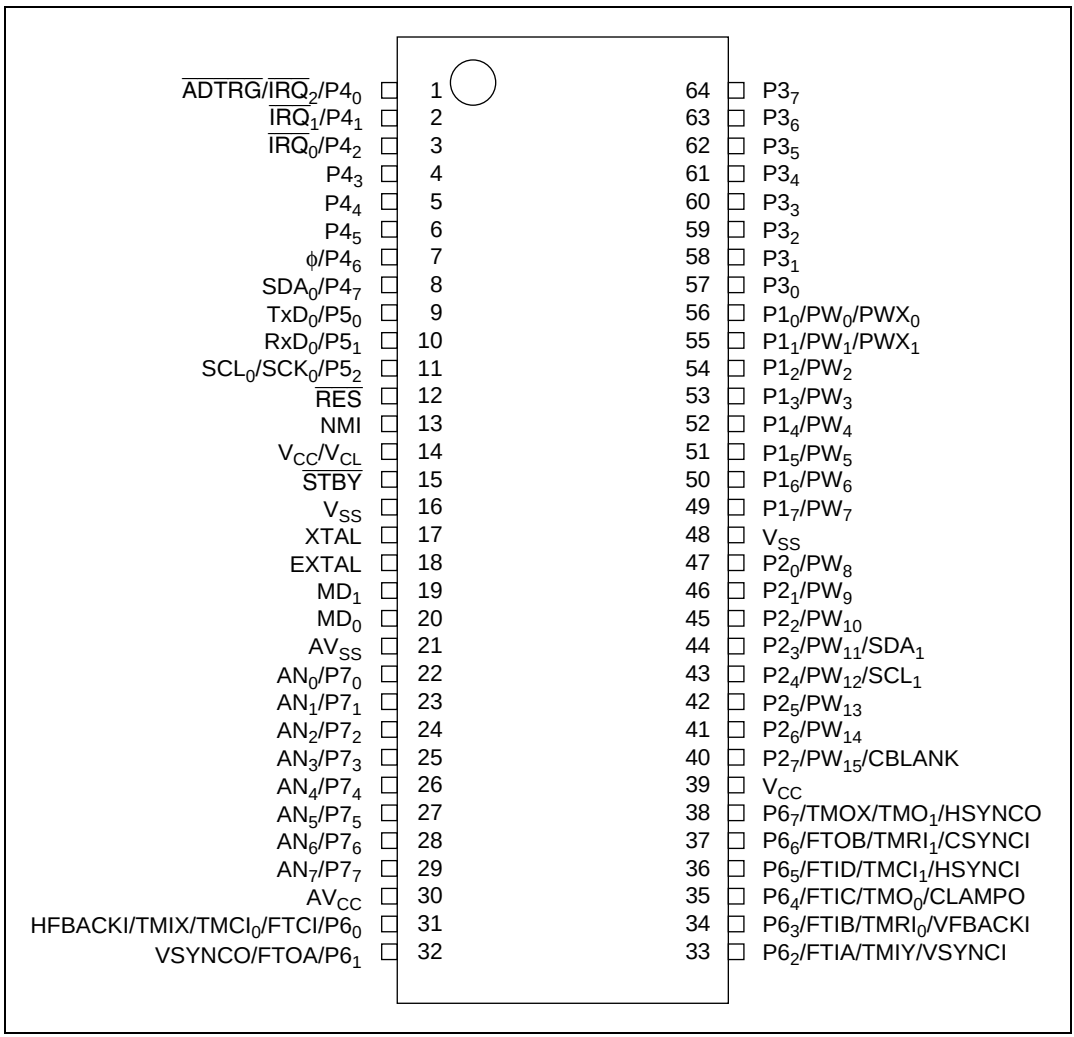


Figure 1.3 H8/3577 Group Pin Arrangement (DP-64S: Top View)

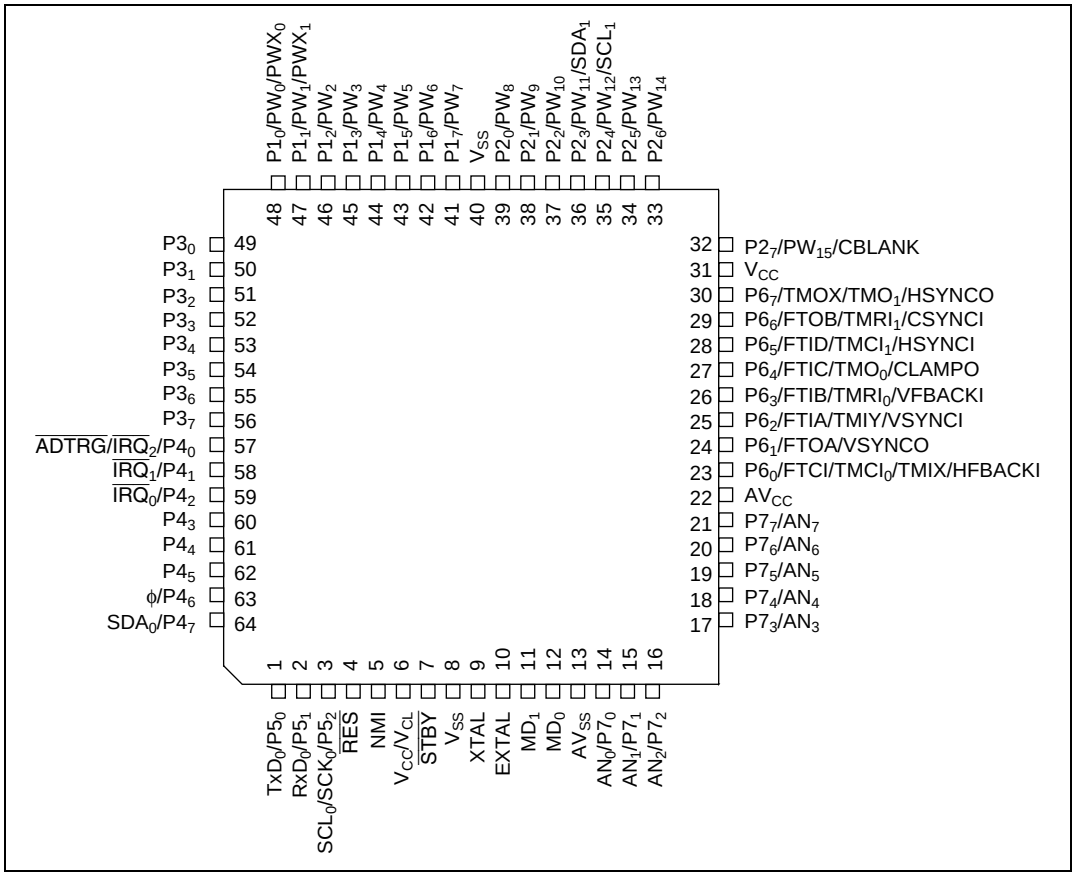


Figure 1.4 H8/3577 Group Pin Arrangement (FP-64A: Top View)

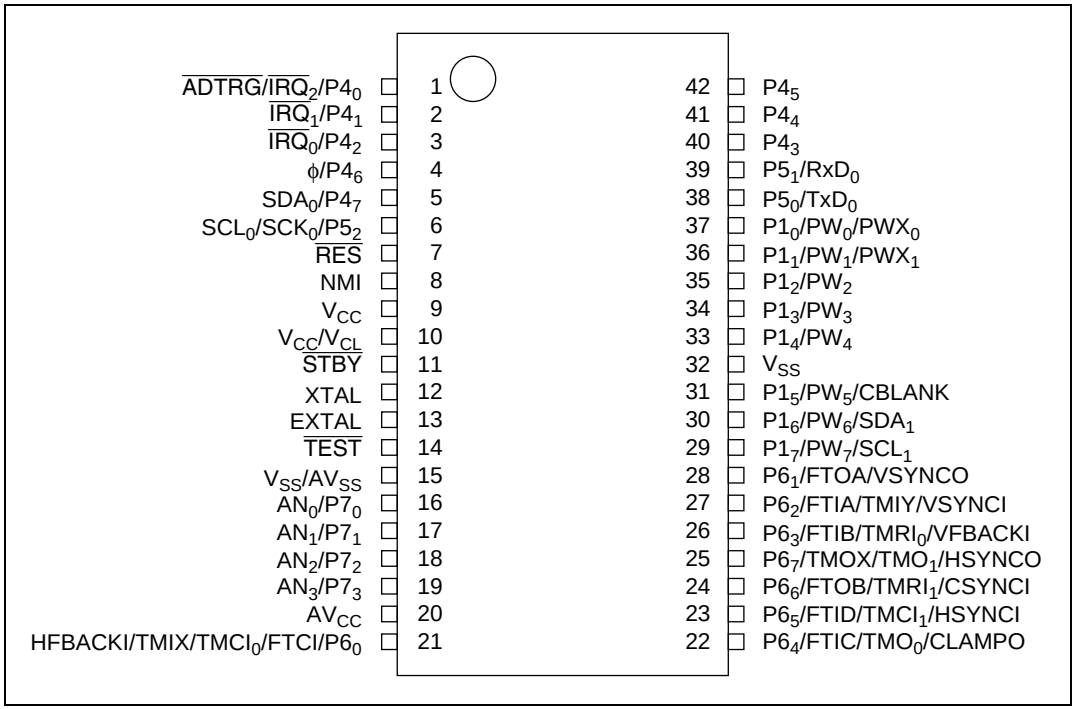


Figure 1.5 H8/3567 Group Pin Arrangement (No USB; DP-42S: Top View)

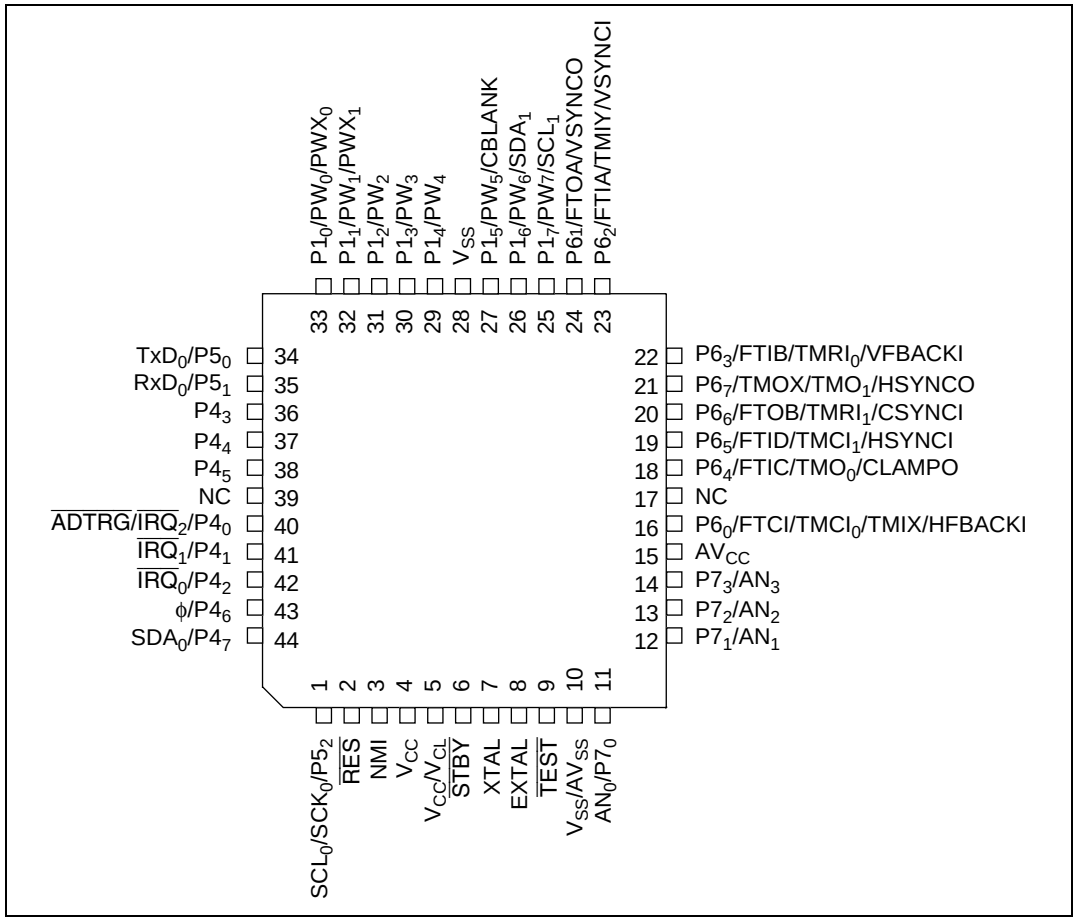


Figure 1.6 H8/3567 Group Pin Arrangement (No USB; FP-44A: Top View)

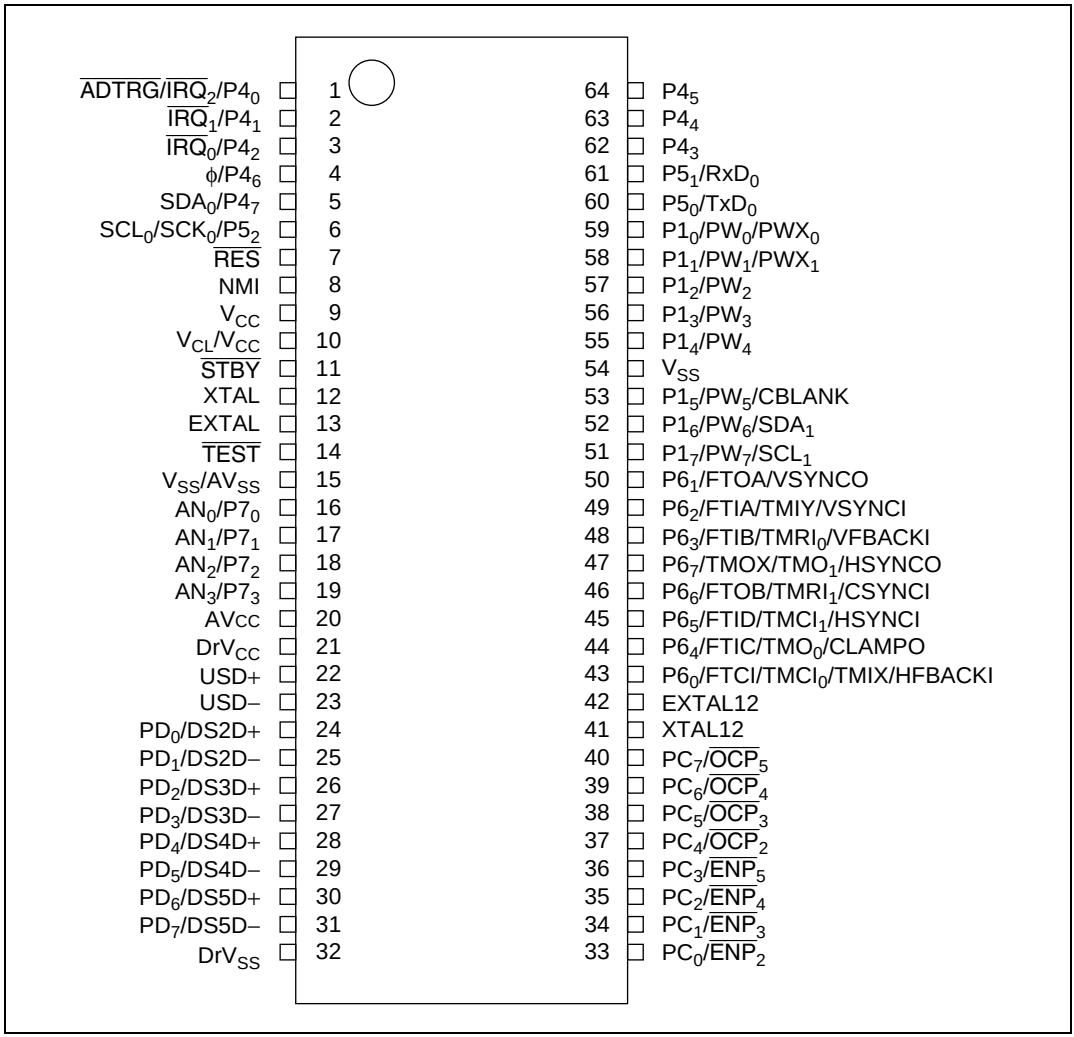


Figure 1.7 H8/3567 Group Pin Arrangement (USB On-Chip; DP-64S: Top View)

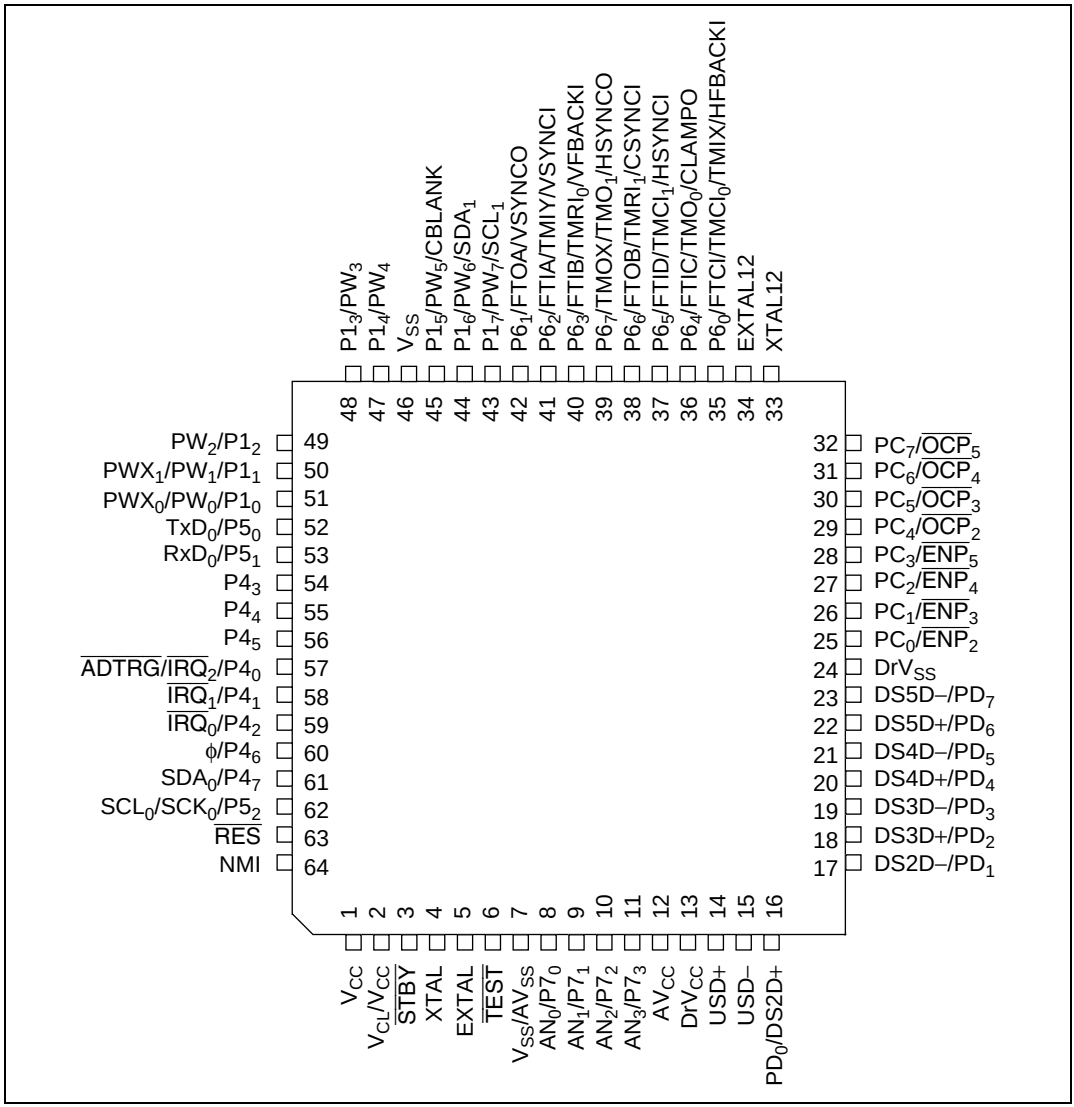


Figure 1.8 H8/3567 Group Pin Arrangement (USB On-Chip; FP-64A: Top View)

1.3.2 List of Pin Functions

H8/3577 Group pin functions are listed in table 1.2, and H8/3567 Group pin functions in tables 1.3 and 1.4.

Table 1.2 List of H8/3577 Group Pin Functions

Pin No.		Pin Name	
DP-64S	**FP-64A**	**Single-Chip Mode**	**PROM Writer Mode**
1	57	$P4_0/\overline{IRQ_2}/\overline{ADTRG}$	EA_{16}
2	58	$P4_1/\overline{IRQ_1}$	EA_{15}
3	59	$P4_2/\overline{IRQ_0}$	$\overline{PGM}$
4	60	$P4_3$	NC
5	61	$P4_4$	NC
6	62	$P4_5$	NC
7	63	$P4_6/\phi$	NC
8	64	$P4_7/SDA_0$	NC
9	1	$P5_0/TxD_0$	NC
10	2	$P5_1/RxD_0$	NC
11	3	$P5_2/SCK_0/SCL_0$	NC
12	4	$\overline{RES}$	V_{PP}
13	5	NMI	EA_9
14	6	V_{CL}, V_{CC} (ZTAT)	V_{CC}
15	7	$\overline{STBY}$	V_{SS}
16	8	V_{SS}	V_{SS}
17	9	XTAL	NC
18	10	EXTAL	NC
19	11	MD_1	V_{SS}
20	12	MD_0	V_{SS}
21	13	AV_{SS}	V_{SS}
22	14	$P7_0/AN_0$	NC
23	15	$P7_1/AN_1$	NC
24	16	$P7_2/AN_2$	NC
25	17	$P7_3/AN_3$	NC

Pin No.		Pin Name	
DP-64S	FP-64A	Single-Chip Mode	PROM Writer Mode
26	18	$P7_4/AN_4$	NC
27	19	$P7_5/AN_5$	NC
28	20	$P7_6/AN_6$	NC
29	21	$P7_7/AN_7$	NC
30	22	AV_{CC}	V_{CC}
31	23	$P6_0$/FTCI/$TMCI_0$/HFBACKI/TMIX	NC
32	24	$P6_1$/FTOA/VSYNCO	NC
33	25	$P6_2$/FTIA/VSYNCI/TMIY	NC
34	26	$P6_3$/FTIB/$TMRI_0$/VFBACKI	V_{CC}
35	27	$P6_4$/FTIC/TMO_0/CLAMPO	V_{CC}
36	28	$P6_5$/FTID/$TMCI_1$/HSYNCI	NC
37	29	$P6_6$/FTOB/$TMRI_1$/CSYNCI	NC
38	30	$P6_7$/TMO_1/TMOX/HSYNCO	NC
39	31	V_{CC}	V_{CC}
40	32	$P2_7/PW_{15}$/CBLANK	$\overline{CE}$
41	33	$P2_6/PW_{14}$	EA_{14}
42	34	$P2_5/PW_{13}$	EA_{13}
43	35	$P2_4/PW_{12}/SCL_1$	EA_{12}
44	36	$P2_3/PW_{11}/SDA_1$	EA_{11}
45	37	$P2_2/PW_{10}$	EA_{10}
46	38	$P2_1/PW_9$	$\overline{OE}$
47	39	$P2_0/PW_8$	EA_8
48	40	V_{SS}	V_{SS}
49	41	$P1_7/PW_7$	EA_7
50	42	$P1_6/PW_6$	EA_6
51	43	$P1_5/PW_5$	EA_5
52	44	$P1_4/PW_4$	EA_4
53	45	$P1_3/PW_3$	EA_3
54	46	$P1_2/PW_2$	EA_2
55	47	$P1_1/PW_1/PWX_1$	EA_1

Pin No.		Pin Name	
DP-64S	**FP-64A**	**Single-Chip Mode**	**PROM Writer Mode**
56	48	$P1_0/PW_0/PWX_0$	EA_0
57	49	$P3_0$	EO_0
58	50	$P3_1$	EO_1
59	51	$P3_2$	EO_2
60	52	$P3_3$	EO_3
61	53	$P3_4$	EO_4
62	54	$P3_5$	EO_5
63	55	$P3_6$	EO_6
64	56	$P3_7$	EO_7

Table 1.3 List of H8/3567 Group Pin Functions (No USB)

Pin No.		Pin Name	
DP-42S	**FP-44A**	**Single-Chip Mode**	**PROM Writer Mode**
1	40	$P4_0/\overline{IRQ_2}/\overline{ADTRG}$	EA_{16}
2	41	$P4_1/\overline{IRQ_1}$	$\overline{CE}$
3	42	$P4_2/\overline{IRQ_0}$	$\overline{PGM}$
4	43	$P4_6/\phi$	EA_{11}
5	44	$P4_7/SDA_0$	V_{CC}
6	1	$P5_2/SCK_0/SCL_0$	V_{CC}
7	2	$\overline{RES}$	V_{PP}
8	3	NMI	EA_9
9	4	V_{CC}	V_{CC}
10	5	V_{CL}, V_{CC} (ZTAT)	V_{CC}
11	6	$\overline{STBY}$	V_{SS}
12	7	XTAL	NC
13	8	EXTAL	NC
14	9	$\overline{TEST}$	V_{SS}
15	10	AV_{SS}/V_{SS}	V_{SS}
16	11	$P7_0/AN_0$	EA_{12}
17	12	$P7_1/AN_1$	EA_{13}
18	13	$P7_2/AN_2$	EA_{14}
19	14	$P7_3/AN_3$	EA_{15}
20	15	AV_{CC}	V_{CC}
21	16	$P6_0$/FTCI/$TMCI_0$/HFBACKI/TMIX	EO_0
—	17	NC	NC
22	18	$P6_4$/FTIC/TMO_0/CLAMPO	EO_4
23	19	$P6_5$/FTID/$TMCI_1$/HSYNCI	EO_5
24	20	$P6_6$/FTOB/$TMRI_1$/CSYNCI	EO_6
25	21	$P6_7$/TMO_1/TMOX/HSYNCO	EO_7
26	22	$P6_3$/FTIB/$TMRI_0$/VFBACKI	EO_3
27	23	$P6_2$/FTIA/VSYNCI/TMIY	EO_2
28	24	$P6_1$/FTOA/VSYNCO	EO_1

Pin No.		Pin Name	
DP-42S	**FP-44A**	**Single-Chip Mode**	**PROM Writer Mode**
29	25	$P1_7/PW_7/SCL_1$	EA_7
30	26	$P1_6/PW_6/SDA_1$	EA_6
31	27	$P1_5/PW_5$/CBLANK	EA_5
32	28	V_{SS}	V_{SS}
33	29	$P1_4/PW_4$	EA_4
34	30	$P1_3/PW_3$	EA_3
35	31	$P1_2/PW_2$	EA_2
36	32	$P1_1/PW_1/PWX_1$	EA_1
37	33	$P1_0/PW_0/PWX_0$	EA_0
38	34	$P5_0/TxD_0$	NC
39	35	$P5_1/RxD_0$	NC
40	36	$P4_3$	EA_8
41	37	$P4_4$	$\overline{OE}$
42	38	$P4_5$	EA_{10}
—	39	NC	NC

Table 1.4 List of H8/3567 Group Pin Functions (USB On-Chip)

Pin No.		Pin Name	
DP-64S	**FP-64A**	**Single-Chip Mode**	**PROM Writer Mode**
1	57	$P4_0/\overline{IRQ_2}/\overline{ADTRG}$	EA_{16}
2	58	$P4_1/\overline{IRQ_1}$	$\overline{CE}$
3	59	$P4_2/\overline{IRQ_0}$	$\overline{PGM}$
4	60	$P4_6/\phi$	EA_{11}
5	61	$P4_7/SDA_0$	V_{CC}
6	62	$P5_2/SCK_0/SCL_0$	V_{CC}
7	63	$\overline{RES}$	V_{PP}
8	64	NMI	EA_9
9	1	V_{CC}	V_{CC}
10	2	V_{CL}, V_{CC} (ZTAT)	V_{CC}
11	3	$\overline{STBY}$	V_{SS}
12	4	XTAL	NC
13	5	EXTAL	NC
14	6	$\overline{TEST}$	V_{SS}
15	7	AV_{SS}/V_{SS}	V_{SS}
16	8	$P7_0/AN_0$	EA_{12}
17	9	$P7_1/AN_1$	EA_{13}
18	10	$P7_2/AN_2$	EA_{14}
19	11	$P7_3/AN_3$	EA_{15}
20	12	AV_{CC}	V_{CC}
21	13	DrV_{CC}	V_{CC}
22	14	USD+	NC
23	15	USD–	NC
24	16	PD_0/DS2D+	NC
25	17	PD_1/DS2D–	NC
26	18	PD_2/DS3D+	NC
27	19	PD_3/DS3D–	NC
28	20	PD_4/DS4D+	NC
29	21	PD_5/DS4D–	NC

Pin No.		Pin Name	
DP-64S	**FP-64A**	**Single-Chip Mode**	**PROM Writer Mode**
30	22	PD_6/DS5D+	NC
31	23	PD_7/DS5D–	NC
32	24	DrV_{SS}	V_{SS}
33	25	$PC_0/\overline{ENP}_2$	NC
34	26	$PC_1/\overline{ENP}_3$	NC
35	27	$PC_2/\overline{ENP}_4$	NC
36	28	$PC_3/\overline{ENP}_5$	NC
37	29	$PC_4/\overline{OCP}_2$	NC
38	30	$PC_5/\overline{OCP}_3$	NC
39	31	$PC_6/\overline{OCP}_4$	NC
40	32	$PC_7/\overline{OCP}_5$	NC
41	33	XTAL12	NC
42	34	EXTAL12	NC
43	35	$P6_0$/FTCI/$TMCI_0$/HFBACKI/TMIX	EO_0
44	36	$P6_4$/FTIC/TMO_0/CLAMPO	EO_4
45	37	$P6_5$/FTID/$TMCI_1$/HSYNCI	EO_5
46	38	$P6_6$/FTOB/$TMRI_1$/CSYNCI	EO_6
47	39	$P6_7$/TMO_1/TMOX/HSYNCO	EO_7
48	40	$P6_3$/FTIB/TMRI0/VFBACKI	EO_3
49	41	$P6_2$/FTIA/VSYNCI/TMIY	EO_2
50	42	$P6_1$/FTOA/VSYNCO	EO_1
51	43	$P1_7/PW_7/SCL_1$	EA_7
52	44	$P1_6/PW_6/SDA_1$	EA_6
53	45	$P1_5/PW_5$/CBLANK	EA_5
54	46	V_{SS}	V_{SS}
55	47	$P1_4/PW_4$	EA_4
56	48	$P1_3/PW_3$	EA_3
57	49	$P1_2/PW_2$	EA_2
58	50	$P1_1/PW_1/PWX_1$	EA_1
59	51	$P1_0/PW_0/PWX_0$	EA_0

Pin No.		Pin Name	
DP-64S	**FP-64A**	**Single-Chip Mode**	**PROM Writer Mode**
60	52	$P5_0/TxD_0$	NC
61	53	$P5_1/RxD_0$	NC
62	54	$P4_3$	EA_8
63	55	$P4_4$	$\overline{OE}$
64	56	$P4_5$	EA_{10}

1.3.3 Pin Functions

Table 1.5 summarizes the functions of the H8/3577 Group and H8/3567 Group pins.

Table 1.5 Pin Functions

		Pin No.							
		H8/3577 Group		H8/3567 Group (No USB)		H8/3567 Group (USB On-Chip)			
Type	Symbol	DP-64S	FP-64A	DP-42S	FP-44A	DP-64S	FP-64A	I/O	Name and Function
Power	V_{CC}	39	31	9	4	9	1	Input	**Power:** For connection to the power supply (5 V).
	V_{CL}/V_{CC}	14	6	10	5	10	2	Input	**Internal step-up power:** For connection to an external capacitor. In the ZTAT version, connect this pin to the power supply (5 V).
	V_{SS}	16, 48	8, 40	32	28	15, 54	7, 46	Input	**Ground:** For connection to the power supply (0 V). Connect all V_{SS} pins to the system power supply (0 V).
Clock	XTAL	17	9	12	7	12	4	Input	For connection of a crystal resonator or external clock input.
	EXTAL	18	10	13	8	13	5	Input	For connection examples, see section 20, Clock Pulse Generator.
	ϕ	7	63	4	43	4	60	Output	**System clock:** Supplies the system clock to external devices.
Operating mode control	MD_1	19	11	—	—	—	—	Input	**Mode pins:** These pins set the operating mode. Connect all three pins—MD_1, MD_0, and $\overline{TEST}$—to the power supply (5 V).
	MD_0	20	12						
	$\overline{TEST}$	—	—	14	9	14	6		
System control	$\overline{RES}$	12	4	7	2	7	63	Input	**Reset input:** When this pin is driven low, the chip goes to the reset state.
	$\overline{STBY}$	15	7	11	6	11	3	Input	**Standby:** When this pin is driven low, a transition is made to hardware standby mode.

		Pin No.							
		H8/3577 Group		H8/3567 Group (No USB)		H8/3567 Group (USB On-Chip)			
Type	**Symbol**	**DP-64S**	**FP-64A**	**DP-42S**	**FP-44A**	**DP-64S**	**FP-64A**	**I/O**	**Name and Function**
Interrupts	NMI	13	5	8	3	8	64	Input	**Nonmaskable interrupt:** Requests a nonmaskable interrupt.
	$\overline{IRQ}_0$ to $\overline{IRQ}_2$	3 to 1	59 to 57	3 to 1	42 to 40	3 to 1	59 to 57	Input	**Interrupt request 0 to 2:** These pins request a maskable interrupt.
16-bit free-running timer (FRT)	FTCI	31	23	21	16	43	35	Input	**FRT counter clock input:** Pin that inputs an external clock signal to the free-running counter (FRC).
	FTOA	32	24	28	24	50	42	Output	**FRT output compare A output:** The output compare A output pin.
	FTOB	37	29	24	20	46	38	Output	**FRT output compare B output:** The output compare B output pin.
	FTIA	33	25	27	23	49	41	Input	**FRT input capture A input:** The input capture A input pin.
	FTIB	34	26	26	22	48	40	Input	**FRT input capture B input:** The input capture B input pin.
	FTIC	35	27	22	18	44	36	Input	**FRT input capture C input:** The input capture C input pin.
	FTID	36	28	23	19	45	37	Input	**FRT input capture D input:** The input capture D input pin.
8-bit timer (TMR_0, TMR_1, TMRX, TMRY)	TMO_0	35	27	22	18	44	36	Output	**Compare-match output:** Compare-match output pins for TMR0, TMR1, and TMRX.
	TMO_1	38	30	25	21	47	39		
	TMOX	38	30	25	21	47	39		
	$TMCI_0$	31	23	21	16	43	35	Input	**Counter external clock input:** Pins that input an external clock to the TMR0 and TMR1 counters.
	$TMCI_1$	36	28	23	19	45	37		
	$TMRI_0$	34	26	26	22	48	40	Input	**Counter external reset input:** TMR0 and TMR1 counter reset input pins.
	$TMRI_1$	37	29	24	20	46	38		

Type	Symbol	Pin No. H8/3577 Group DP-64S	H8/3577 Group FP-64A	H8/3567 Group (No USB) DP-42S	H8/3567 Group (No USB) FP-44A	H8/3567 Group (USB On-Chip) DP-64S	H8/3567 Group (USB On-Chip) FP-64A	I/O	Name and Function
8-bit timer (TMR0, TMR1, TMRX, TMRY)	TMIX	31	23	21	16	43	35	Input	**Counter external clock input/reset input:** Pins with a dual function of TMRX and TMRY counter clock input and reset input.
	TMIY	33	25	27	23	49	41		
Serial communication interface (SCI0)	TxD_0	9	1	38	34	60	52	Output	**Transmit data:** Data output pins.
	RxD_0	10	2	39	35	61	53	Input	**Receive data:** Data input pins.
	SCK_0	11	3	6	1	6	62	Input/ output	**Serial clock:** Clock input/output pins. The SCK_0 output type is NMOS push-pull.
A/D converter	AN_7 to AN_4	29 to 26	21 to 18	—	—	—	—	Input	**Analog 7 to 0:** Analog input pins.
	AN_3 to AN_0	25 to 22	17 to 14	19 to 16	14 to 11	19 to 16	11 to 8	Input	
	$\overline{ADTRG}$	1	57	1	40	1	57	Input	**A/D conversion external trigger input:** Pin for input of an external trigger to start A/D conversion.
	AV_{CC}	30	22	20	15	20	12	Input	**Analog power:** The A/D converter reference power supply pin. When the A/D converter is not used, connect this pin to the system power supply (+5 V).
	AV_{SS}	21	13	15	10	15	7	Input	**Analog ground:** The A/D converter ground pin. Connect this pin to the system power supply (0 V).

		Pin No.							
		H8/3577 Group		H8/3567 Group (No USB)		H8/3567 Group (USB On-Chip)			
Type	Symbol	DP-64S	FP-64A	DP-42S	FP-44A	DP-64S	FP-64A	I/O	Name and Function
PWM timer (PWM)	PW_{15} to PW_{8}	40 to 47	32 to 39	—	—	—	—	Output	**PWM timer output:** PWM timer pulse output pins.
	PW_{7} to PW_{0}	49 to 56	41 to 48	29 to 31	25 to 27	51 to 53	43 to 45		
				33 to 37	29 to 33	55 to 59	47 to 51		
14-bit PWM timer (PWMX)	PWX_{0}	56	48	37	33	59	51	Output	**PWMX timer output:** PWM D/A pulse output pins.
	PWX_{1}	55	47	36	32	58	50		
Timer connection	VSYNCI	33	25	27	23	49	41	Input	**Timer connection input:** Timer connection synchronization signal input pins.
	HSYNCI	36	28	23	19	45	37		
	CSYNCI	37	29	24	20	46	38		
	VFBACKI	34	26	26	22	48	40		
	HFBACKI	31	23	21	16	43	35		
	VSYNCO	32	24	28	24	50	42	Output	**Timer connection output:** Timer connection synchronization signal output pins.
	HSYNCO	38	30	25	21	47	39		
	CLAMPO	35	27	22	18	44	36		
	CBLANK	40	32	31	27	53	45		
I^2C bus interface (IIC)	SCL_{0}	11	3	6	1	6	62	Input/ Output	**I^2C clock input/output (channels 0 and 1):** I^2C clock input/output pins.
	SCL_{1}	43	35	29	25	51	43		These pins have a bus drive function.
									The SCL_{0} output type is NMOS open-drain.
	SDA_{0}	8	64	5	44	5	61	Input/ Output	**I^2C data input/output (channels 0 and 1):** I^2C data input/output pins.
	SDA_{1}	44	36	30	26	52	44		These pins have a bus drive function.
									The SDA_{0} output type is NMOS open-drain.

		Pin No.							
		H8/3577 Group		H8/3567 Group (No USB)		H8/3567 Group (USB On-Chip)			
Type	Symbol	DP-64S	FP-64A	DP-42S	FP-44A	DP-64S	FP-64A	I/O	Name and Function
Universal serial bus (USB)	USD+	—	—	—	—	22	14	Input/ Output	**Upstream data input/output:** USB upstream data input/ output pins.
	USD–					23	15		
	DS2D+	—	—	—	—	24	16	Input/ Output	**Upstream data input/output 2 to 5:** USB hub downstream data input/output pins.
	DS2D–					25	17		
	DS3D+					26	18		
	DS3D–					27	19		
	DS4D+					28	20		
	DS4D–					29	21		
	DS5D+					30	22		
	DS5D–					31	23		
	$\overline{ENP_2}$ to $\overline{ENP_5}$	—	—	—	—	33 to 36	25 to 28	Output	**Power supply control IC power output enable signal output:** Output pins to USB port power supply control IC enable input
	$\overline{OCP_2}$ to $\overline{OCP_5}$	—	—	—	—	37 to 40	29 to 32	Input	**Overcurrent detection signal input:** Input pins for overcurrent detection signal from USB port power supply control IC
	XTAL12	—	—	—	—	41	33	Input	**USB clock input:** For connection of a 12 MHz crystal resonator or external clock input. Quadrupled to 48 MHz inside the chip.
	EXTAL12	—	—	—	—	42	34	Input	
	DrV_{CC}	—	—	—	—	21	13	Input	**Bus driver power:** For connection of the bus driver/receiver power supply (3.3 V).
	DrV_{SS}	—	—	—	—	32	24	Input	**Bus driver ground:** For connection of the bus driver/receiver power supply (0 V).

		Pin No.							
		H8/3577 Group		H8/3567 Group (No USB)		H8/3567 Group (USB On-Chip)			
Type	Symbol	DP-64S	FP-64A	DP-42S	FP-44A	DP-64S	FP-64A	I/O	Name and Function
I/O ports	$P1_7$ to $P1_0$	49 to 56	41 to 48	29 to 31 33 to 37	25 to 27 29 to 33	51 to 53 55 to 59	43 to 45 47 to 51	Input/ Output	**Port 1:** Eight input/output pins. The direction of each pin can be selected in the port 1 data direction register (P1DDR).
	$P2_7$ to $P2_0$	40 to 47	32 to 39	—	—	—	—	Input/ Output	**Port 2:** Eight input/output pins. The direction of each pin can be selected in the port 2 data direction register (P2DDR).
	$P3_7$ to $P3_0$	64 to 57	56 to 49	—	—	—	—	Input/ Output	**Port 3:** Eight input/output pins. The direction of each pin can be selected in the port 3 data direction register (P3DDR).
	$P4_7$ to $P4_0$	8 to 1	64 to 57	5, 4 42 to 40 3 to 1	44, 43 38 to 36 42 to 40	5, 4 64 to 62 3 to 1	61, 60 56 to 54 59 to 57	Input/ Output	**Port 4:** Eight input/output pins. The direction of each pin (except $P4_6$) can be selected in the port 4 data direction register (P4DDR). $P4_7$ is an NMOS push-pull output.
	$P5_2$ to $P5_0$	11 to 9	3 to 1	6 39 38	1 35 34	6 61 60	62 53 52	Input/ Output	**Port 5:** Three input/output pins. The direction of each pin can be selected in the port 5 data direction register (P5DDR). $P5_2$ is an NMOS push-pull output.
	$P6_7$ to $P6_0$	38 to 31	30 to 23	25 to 22 26 to 28 21	21 to 18 22 to 24 16	47 to 44 48 to 50 43	39 to 36 40 to 42 35	Input/ Output	**Port 6:** Eight input/output pins. The direction of each pin can be selected in the port 6 data direction register (P6DDR).
	$P7_7$ to $P7_4$	29 to 26	21 to 18	—	—	—	—	Input	**Port 7:** Eight (H8/3577 Group) or four (H8/3567 Group) input pins.
	$P7_3$ to $P7_0$	25 to 22	17 to 14	19 to 16	14 to 11	19 to 16	11 to 8		

Type	Symbol	Pin No. H8/3577 Group DP-64S	Pin No. H8/3577 Group FP-64A	Pin No. H8/3567 Group (No USB) DP-42S	Pin No. H8/3567 Group (No USB) FP-44A	Pin No. H8/3567 Group (USB On-Chip) DP-64S	Pin No. H8/3567 Group (USB On-Chip) FP-64A	I/O	Name and Function
I/O ports	PC_7 to PC_0	—	—	—	—	40 to 33	32 to 25	Input/ Output	**Port C:** Eight input/output pins. The direction of each pin can be selected in the port C data direction register (PCDDR).
	PD_7 to PD_0	—	—	—	—	31 to 24	23 to 16	Input/ Output	**Port D:** Eight input/output pins. The direction of each pin can be selected in the port D data direction register (PDDDR). These pins are driven by DrV_{CC} (3.3 V).

Section 2 CPU

2.1 Overview

The H8/300 CPU has sixteen 8-bit general registers, which can also be paired as eight 16-bit registers. Its concise instruction set is designed for high-speed operation.

2.1.1 Features

Features of the H8/300 CPU are listed below.

- General-register architecture
  Sixteen 8-bit general registers, also usable as eight 16-bit general registers
- Instruction set with 55 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+/@–Rn)
  — Absolute address (@aa:8/@aa:16)
  — Immediate (#xx:8/#xx:16)
  — Program-counter relative (@(d:8, PC))
  — Memory indirect (@@aa:8)
- 64-kbyte address space
- High-speed operation
  — All frequently used instructions are executed in two to four states
  — High-speed arithmetic and logic operations
    8- or 16-bit register-register add or subtract: 0.1 µs (operating at ϕ = 20 MHz)
    8 × 8-bit multiply: 0.7 µs (operating at ϕ = 20 MHz)
    16 ÷ 8-bit divide: 0.7 µs (operating at ϕ = 20 MHz)
- Low-power operation modes
  SLEEP instruction for transfer to low-power operation

2.1.2 Address Space

The H8/300 CPU supports an address space of up to 64 kbytes for storing program code and data.

See section 3.3, Address Map, for details of the memory map.

2.1.3 Register Configuration

Figure 2.1 shows the register structure of the H8/300 CPU. There are two groups of registers: the general registers and control registers.

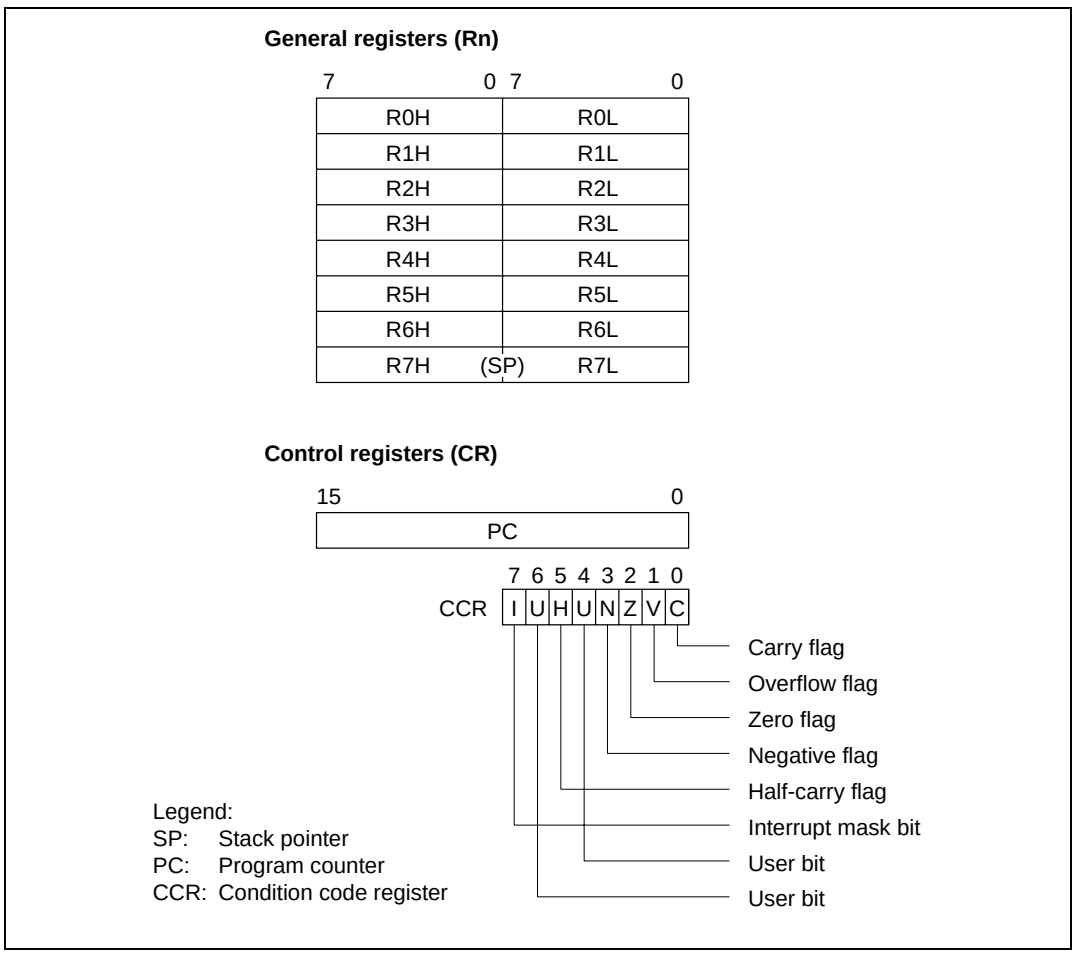


Figure 2.1 CPU Registers

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

When used as address registers, the general registers are accessed as 16-bit registers (R0 to R7).

R7 also functions as the stack pointer (SP), used implicitly by hardware in exception processing and subroutine calls. When it functions as the stack pointer, as indicated in figure 2.2, SP (R7) points to the top of the stack.

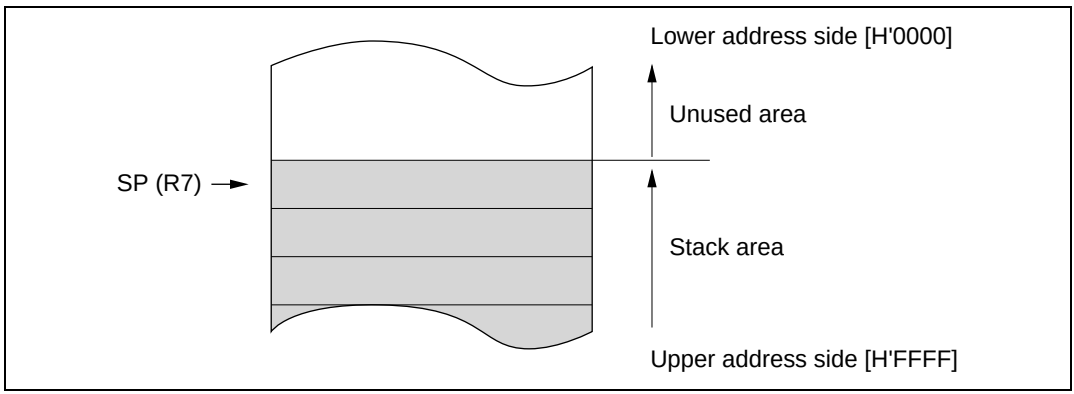


Figure 2.2 Stack Pointer

2.2.2 Control Registers

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

Program Counter (PC)

This 16-bit register indicates the address of the next instruction the CPU will execute. All instructions are fetched 16 bits (1 word) at a time, so the least significant bit of the PC is ignored (always regarded as 0).

Condition Code Register (CCR)

This 8-bit register contains internal status information, including the interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags. These bits can be read and written by software (using the LDC, STC, ANDC, ORC, and XORC instructions). The N, Z, V, and C flags are used as branching conditions for conditional branching (Bcc) instructions.

Bit 7—Interrupt Mask Bit (I): When this bit is set to 1, interrupts are masked. This bit is set to 1 automatically at the start of exception handling. The interrupt mask bit may be read and written by software. For further details, see section 5, Interrupt Controller.

Bit 6—User Bit (U): Can be used freely by the user.

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

The H flag is used implicitly by the DAA and DAS instructions.

When the ADD.W, SUB.W, or CMP.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and is cleared to 0 otherwise.

Bit 4—User Bit (U): Can be used freely by the user.

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

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

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

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

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

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

Some instructions leave some or all of the flag bits unchanged.

Refer to the *H8/300 Series Programming Manual* for the action of each instruction on the flag bits.

2.2.3 Initial Register Values

In reset exception handling, the program counter (PC) is initialized by a vector address (H'0000) load, and the I bit 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 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.

1-bit data is handled by bit manipulation instructions, and is accessed by being specified as bit n (n = 0, 1, 2, ... 7) in the operand data (byte).

Byte data is handled by all arithmetic and logic instructions except ADDS and SUBS. Word data is handled by the MOV.W, ADD.W, SUB.W, CMP.W, ADDS, SUBS, MULXU (8 bits × 8 bits), and DIVXU (16 bits ÷ 8 bits) instructions.

With the DAA and DAS decimal adjustment instructions, byte data is handled as two 4-bit BCD data units.

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.

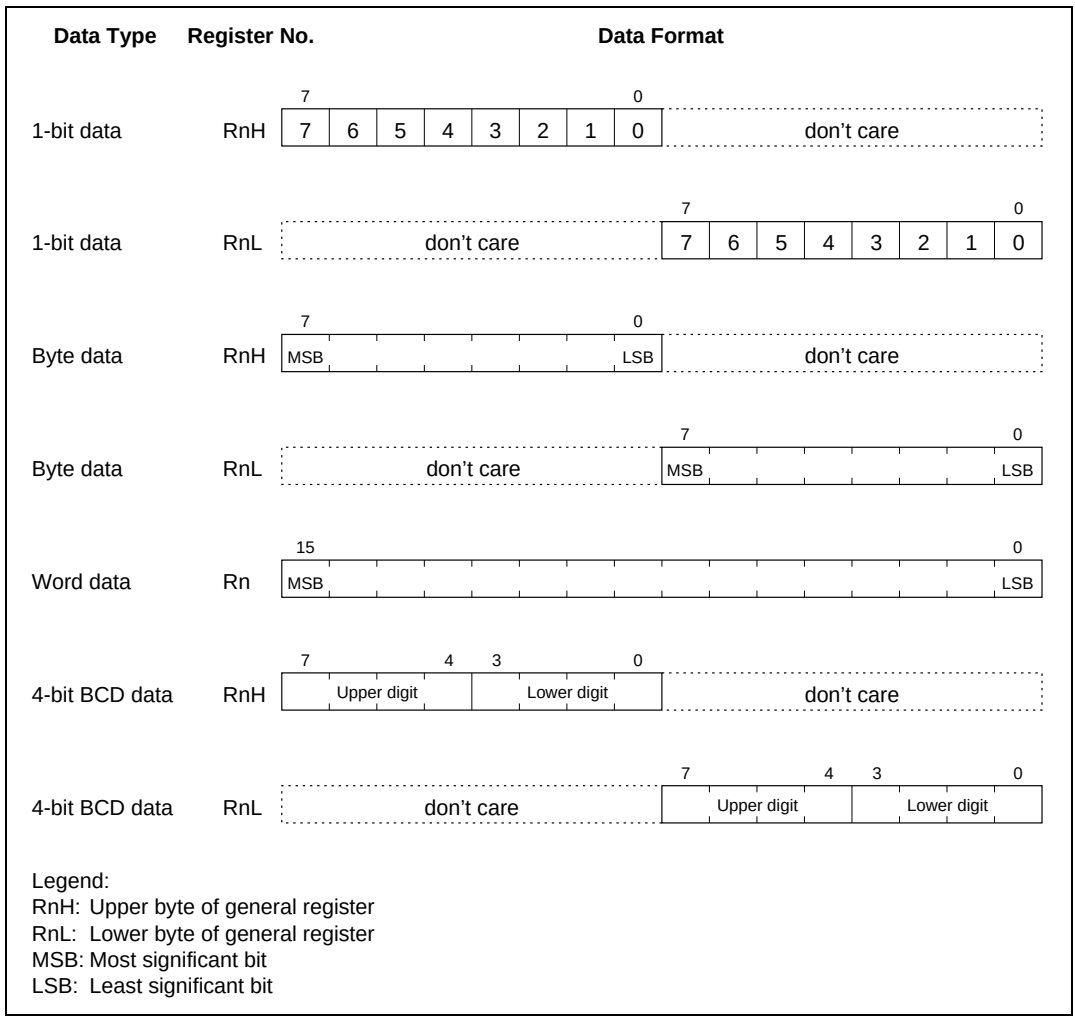


Figure 2.3 General Register Data Formats

2.3.2 Memory Data Formats

Figure 2.4 indicates the data formats in memory. For access by the H8/300L CPU, word data stored in memory must always begin at an even address. When word data beginning at an odd address is accessed, the least significant bit is regarded as 0, and the word data beginning at the preceding address is accessed. The same applies to instruction codes.

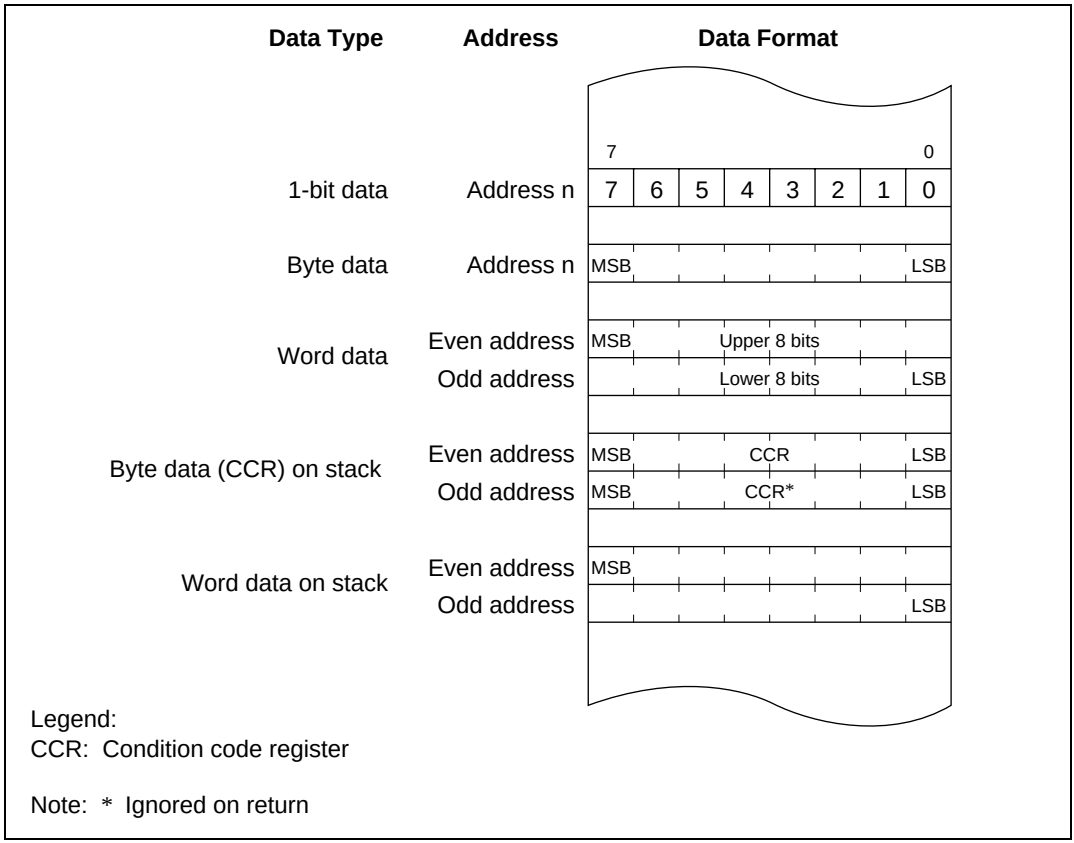


Figure 2.4 Memory Data Formats

When the stack is accessed using R7 as an address register, word access should always be performed. The CCR is stored as word data with the same value in the upper 8 bits and the lower 8 bits. On return, the lower 8 bits are ignored.

2.4 Addressing Modes

2.4.1 Addressing Modes

The H8/300 CPU supports the eight addressing modes listed in table 2.1. Each instruction uses a subset of these addressing modes.

Table 2.1 Addressing Modes

No.	Address Modes	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.

   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 in memory.

3. **Register Indirect with Displacement—@(d:16, Rn):** The instruction has a second word (bytes 3 and 4) containing a displacement which is added to the contents of the specified general register to obtain the operand address in memory.

   This mode is used only in MOV instructions. 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.

     The register field of the instruction specifies a 16-bit general register containing the address of the operand. After the operand is accessed, the register is incremented by 1 for MOV.B or 2 for MOV.W, and the result of the addition is stored in the register. 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.

     The register field of the instruction specifies a 16-bit general register which is decremented by 1 or 2 to obtain the address of the operand in memory. The register retains the decremented value. The size of the decrement is 1 for MOV.B or 2 for MOV.W. For MOV.W, the original contents of the register must be even.

5. **Absolute Address—@aa:8 or @aa:16:** The instruction specifies the absolute address of the operand in memory.

   The absolute address may be 8 bits long (@aa:8) or 16 bits long (@aa:16). The MOV.B and bit manipulation instructions can use 8-bit absolute addresses. The MOV.B, MOV.W, JMP, and JSR instructions can use 16-bit absolute addresses.

   For an 8-bit absolute address, the upper 8 bits are assumed to be 1 (H'FF). The address range is H'FF00 to H'FFFF (65280 to 65535).

6. **Immediate—#xx:8 or #xx:16:** The second byte (#xx:8) or the third and fourth bytes (#xx:16) of the instruction code are used directly as the operand. Only MOV.W instructions can be used with #xx:16.

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

7. **Program-Counter Relative—@(d:8, PC):** This mode is used in the Bcc and BSR instructions. An 8-bit displacement in byte 2 of the instruction code is sign-extended to 16 bits and added to the program counter contents to generate a branch destination address, and the PC contents to be added are the start address of the next instruction, so that the possible branching range is –126 to +128 bytes (–63 to +64 words) from the branch instruction. The displacement should be an even number.

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. This specifies an operand in memory, and a branch is performed with the contents of this operand as the branch address.

   The upper 8 bits of the absolute address are assumed to be 0 (H'00), so the address range is from H'0000 to H'00FF (0 to 255). Note that with the H8/300 Series, the lower end of the address area is also used as a vector area. See section 4, Exception Handling, for details on the vector area.

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 2.3.2, Memory Data Formats, for further information.

2.4.2 Effective Address Calculation

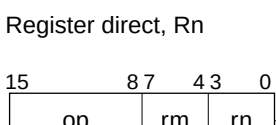
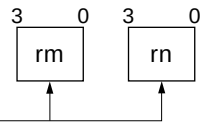
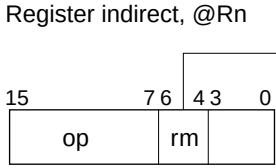
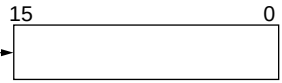
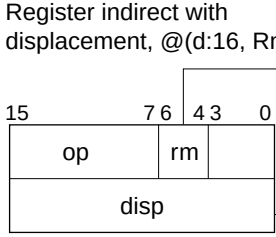
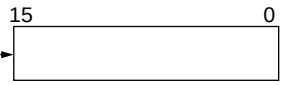
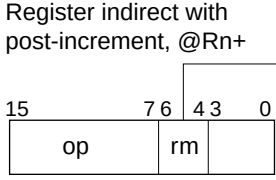
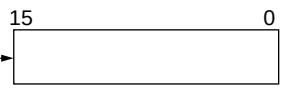
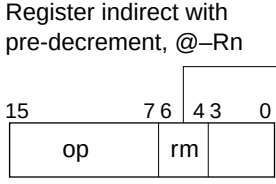
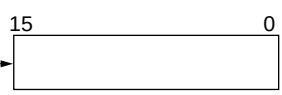
Table 2.2 shows how effective addresses are calculated in each of the addressing modes.

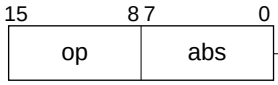
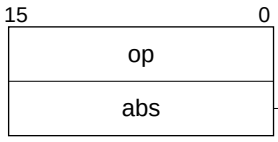
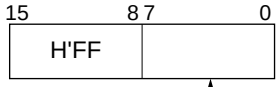
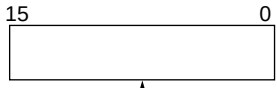
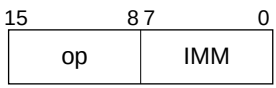
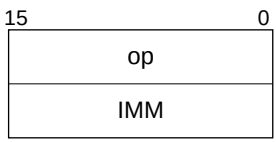
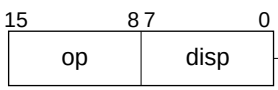
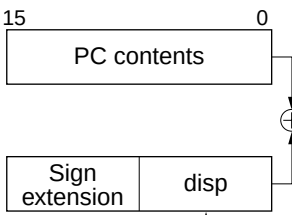
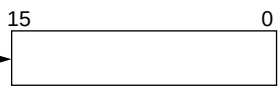
Arithmetic and logic instructions use register direct addressing (1). The ADD.B, ADDX, SUBX, CMP.B, AND, OR, and XOR instructions can also use immediate addressing (6).

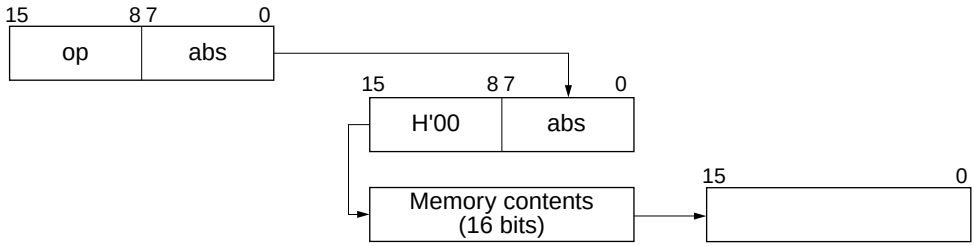
Data transfer instructions can use all addressing modes except program-counter relative (7) and memory indirect (8).

Bit manipulation instructions use register direct (1), register indirect (2), or 8-bit absolute addressing (5) to specify a byte operand, and 3-bit immediate addressing (6) to specify a bit position in that byte. The BSET, BCLR, BNOT, and BTST instructions can also use register direct addressing (1) to specify the bit position.

Table 2.2 Effective Address Calculation

No.	Addressing Mode and Instruction Format	Effective Address Calculation Method	Effective Address (EA)
1	Register direct, Rn		 Operand is contents of registers indicated by rm/rn
2	Register indirect, @Rn		
3	Register indirect with displacement, @(d:16, Rn)		
4	Register indirect with post-increment, @Rn+		
	Register indirect with pre-decrement, @–Rn		

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

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

Legend:

rm, rn: Register field
op: Operation field
disp: Displacement
IMM: Immediate data
abs: Absolute address

2.5 Instruction Set

The H8/300 Series can use a total of 55 instructions, which are grouped by function in table 2.3.

Table 2.3 Instruction Set

Function	Instructions	Number
Data transfer	MOV, PUSH*1, POP*1	1
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 (Cannot be used in the H8/3577 Group and H8/3567 Group)	EEPMOV	1
		Total: 55

Notes: 1. PUSH Rn is equivalent to MOV.W Rn, @–SP.
POP Rn is equivalent to MOV.W @SP+, Rn. The same applies to machine language.
2. Bcc is a conditional branch instruction.

Tables 2.4 to 2.11 show the function of each instruction. The notation used is defined next.

Notation

Rd	General register (destination)
Rs	General register (source)
Rn	General register
(EAd), <Ead>	Destination operand
(EAs), <Eas>	Source operand
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
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	AND logical
∨	OR logical
⊕	Exclusive OR logical
→	Move
~	Logical negation (logical complement)
:3	3-bit length
:8	8-bit length
:16	16-bit length
( ), < >	Contents of operand indicated by effective address

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:16, @–Rn, and @Rn+ addressing modes are available for word data. The @aa:8 addressing mode is available for byte data only. The @–R7 and @R7+ modes require a word-size specification.
POP	W	@SP+ → Rn Pops a general register from the stack. Equivalent to MOV.W @SP+, Rn.
PUSH	W	Rn → @–SP Pushes general register onto the stack. Equivalent to MOV.W Rn, @–SP.

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

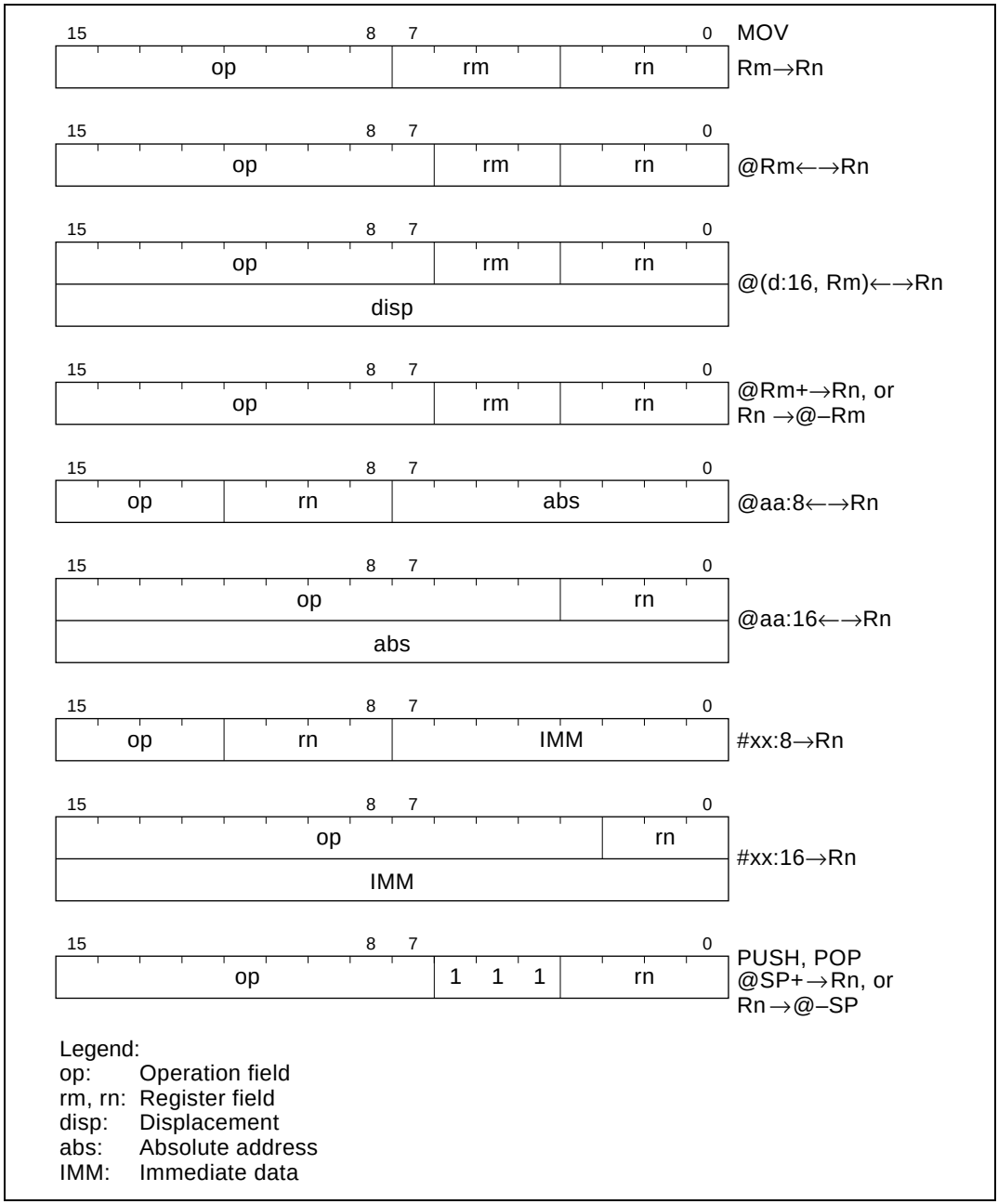


Figure 2.5 Data Transfer Instruction Codes

2.5.2 Arithmetic Operations

Table 2.5 describes the arithmetic instructions.

Table 2.5 Arithmetic Instructions

Instruction	Size*	Function
ADD	B/W	Rd ± Rs → Rd, Rd + #IMM → Rd
SUB		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	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd
SUBX		Performs addition or subtraction with carry on data in two general registers, or addition or subtraction with carry on immediate data and data in a general register.
INC	B	Rd ± 1 → Rd
DEC		Increments or decrements a general register
ADDS	W	Rd ± 1 → Rd, Rd ± 2 → Rd
SUBS		Adds or subtracts 1 or 2 to or from a general register
DAA	B	Rd decimal adjust → Rd
DAS		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, and indicates the result in the CCR. 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

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

2.5.3 Logic Operations

Table 2.6 describes the four instructions that perform logic operations.

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

Notes: * Size: Operand size
B: Byte

2.5.4 Shift Operations

Table 2.7 describes the eight shift instructions.

Table 2.7 Shift Instructions

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

Notes: * Size: Operand size
B: Byte

Figure 2.6 shows the instruction code format of arithmetic, logic, and shift instructions.

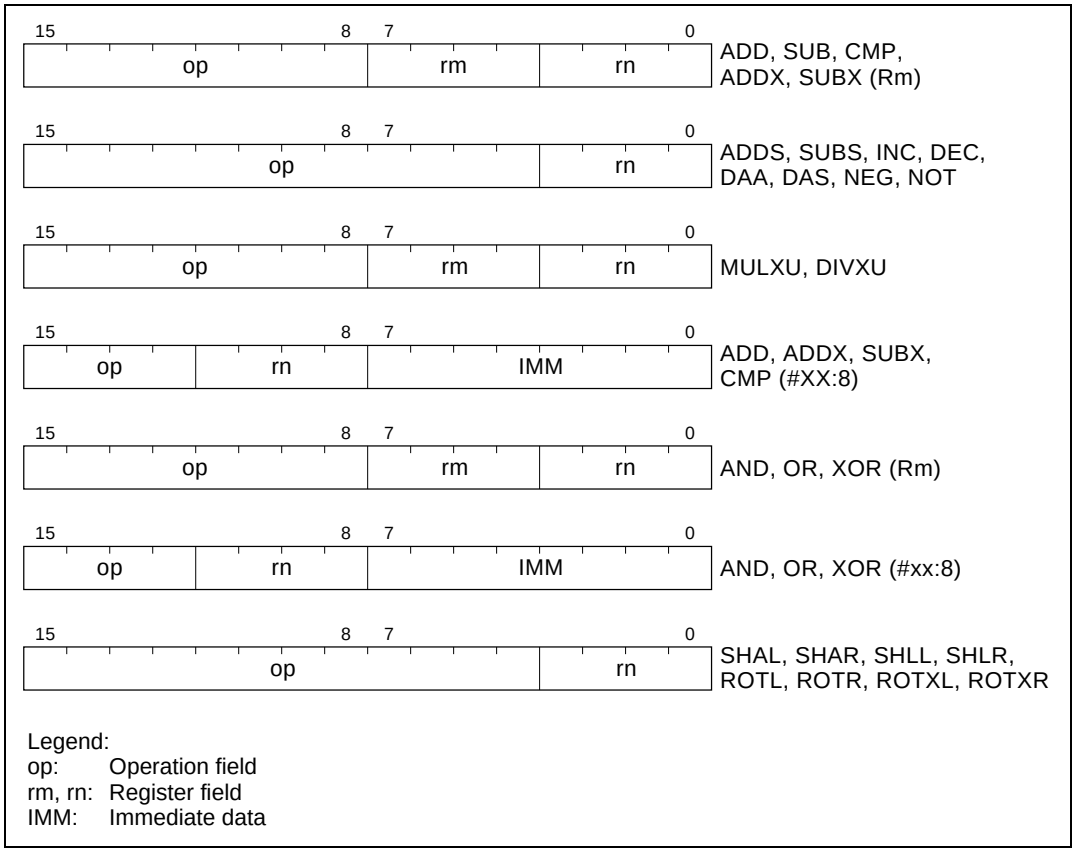


Figure 2.6 Arithmetic, Logic, and Shift Instruction Codes

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 number is specified by 3-bit immediate data or the lower three bits of a general register.
BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in a general register or memory to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BNOT	B	~ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BTST	B	~ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the C flag with a specified bit in a general register or memory, and stores the result in the C flag.
BIAND	B	C ∧ [~ (<bit-No.> of <EAd>)] → C ANDs the C flag with the inverse of a specified bit in a general register or memory, and stores the result in the C flag. 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, and stores the result in the C flag.
BIOR	B	C ∨ [~ (<bit-No.> of <EAd>)] → C ORs the C flag with the inverse of a specified bit in a general register or memory, and stores the result in the C flag. The bit number is specified by 3-bit immediate data.

Notes: * Size: Operand size
B: Byte

Instruction	Size*	Function
BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C XORs the C flag with a specified bit in a general register or memory, and stores the result in the C flag.
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, and stores the result in the C flag. 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	B	~ (<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	B	~ 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.

Notes: * Size: Operand size
B: Byte

Certain precautions are required in bit manipulation. See 2.8.1, Notes on Bit Manipulation, for details.

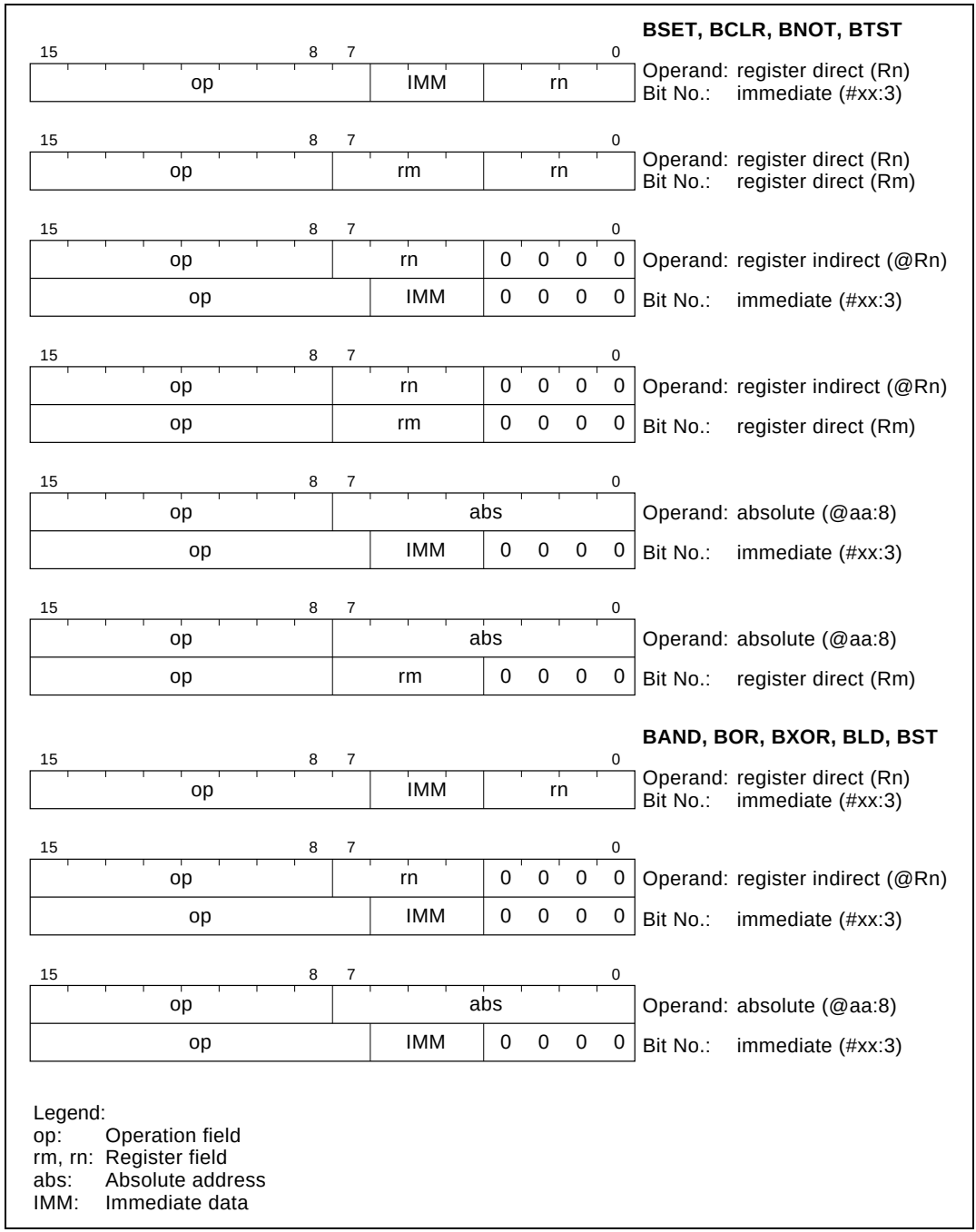


Figure 2.7 Bit Manipulation Instruction Codes

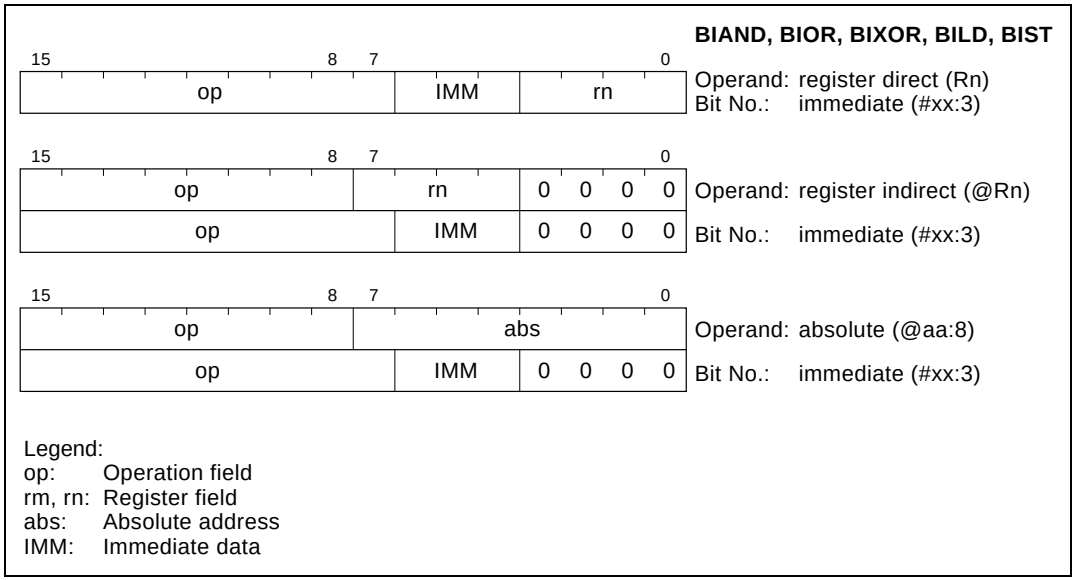


Figure 2.7 Bit Manipulation Instruction Codes (cont)

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 to the designated address if condition cc is true. The branching conditions are given below.

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

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

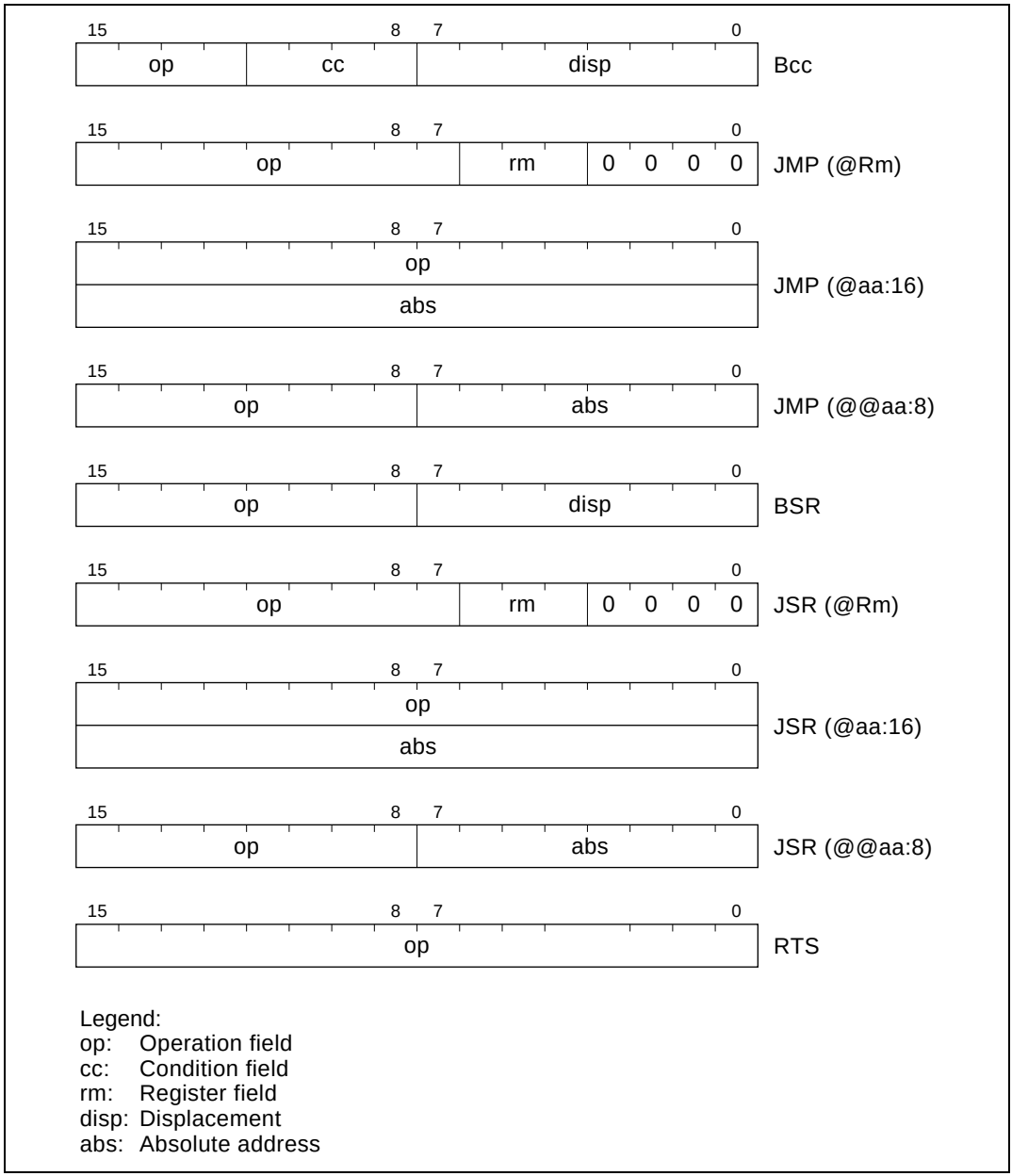


Figure 2.8 Branching Instruction Codes

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 from active mode to a power-down mode. See section 21, Power-Down State, for details.
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

Notes: * Size: Operand size
B: Byte

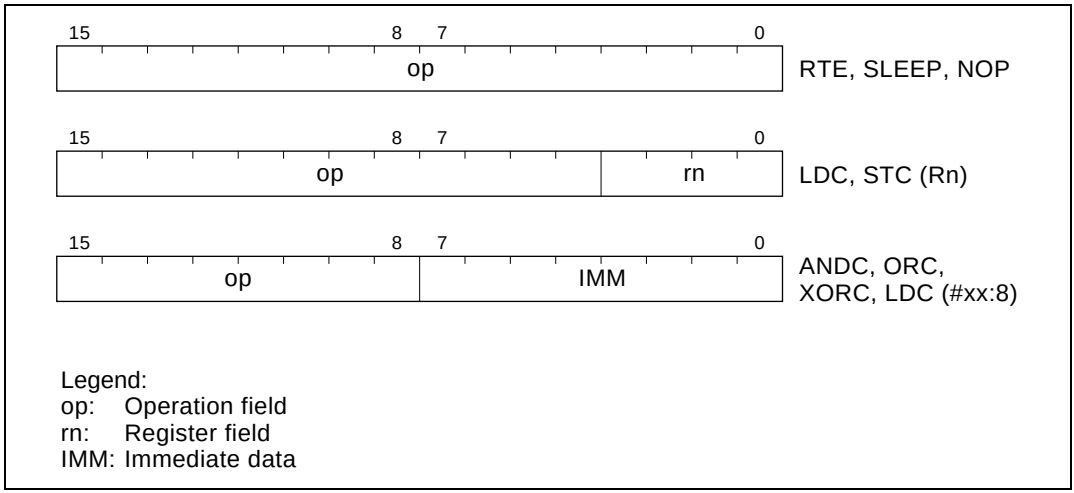


Figure 2.9 System Control Instruction Codes

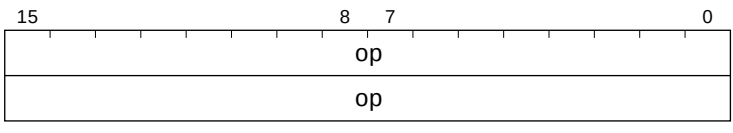
2.5.8 Block Data Transfer Instruction

Table 2.11 describes the block data transfer instruction. Figure 2.10 shows its object code format.

Table 2.11 Block Data Transfer Instruction

Instruction	Size	Function
EEPMOV (Cannot be used in the H8/3577 Group and H8/3567 Group)	—	If R4L ≠ 0 then repeat @R5+ → @R6+ R4L – 1 → R4L until R4L = 0 else next; Block transfer instruction. Transfers the number of data bytes specified by R4L from locations starting at the address indicated by R5 to locations starting at the address indicated by R6. After the transfer, the next instruction is executed.

Certain precautions are required in using the EEPMOV instruction. See 2.8.2, Notes on Use of the EEPMOV Instruction, for details.



Legend:
op: Operation field

Figure 2.10 Block Data Transfer Instruction Code

2.6 Basic Operational Timing

CPU operation is synchronized by a system clock (ϕ). The period from a rising edge of ϕ to the next rising edge is called one state. A bus cycle consists of two states or three states. The cycle differs depending on whether access is to on-chip memory or to on-chip peripheral modules.

2.6.1 Access to On-Chip Memory (RAM, ROM)

Access to on-chip memory takes place in two states. The data bus width is 16 bits, allowing access in byte or word size. Figure 2.11 shows the on-chip memory access cycle.

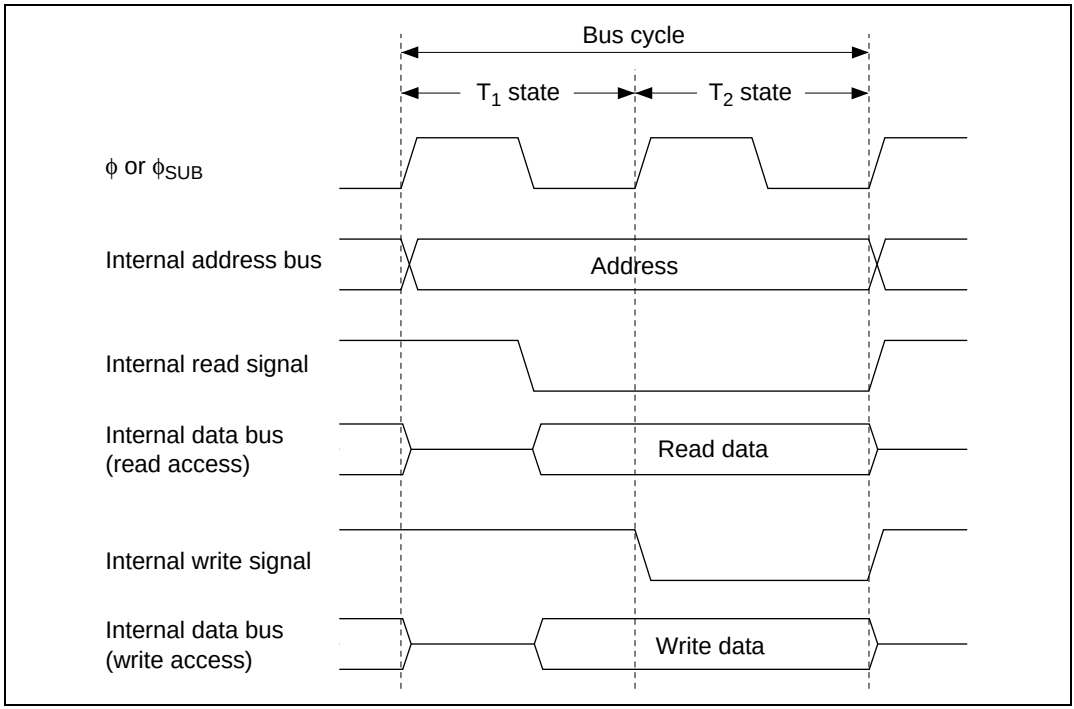


Figure 2.11 On-Chip Memory Access Cycle

2.6.2 Access to On-Chip Peripheral Modules

On-chip peripheral modules are accessed in three states. The data bus width is either 8 or 16 bits, so access in both byte and word size is supported.

There are two categories of on-chip peripheral modules: 8-bit and 16-bit. To access word data from an 8-bit module, two instructions must be used. The upper byte is accessed first, followed by the lower byte. Accessing word data from a 16-bit module requires only one instruction.

There are two types of registers: byte and word. The word register refers to registers were, as with a 16-bit counter, attempting to access the two bytes separately will cause problems. For word registers containing 8-bit modules, a circuit with a temporary register is available to allow normal access to the upper byte first, followed by the lower byte. Note that word registers containing only 16-bit modules do not have such a circuit. Therefore, only word access may be used with such registers.

Figure 2.12 shows the access timing for on-chip peripheral modules.

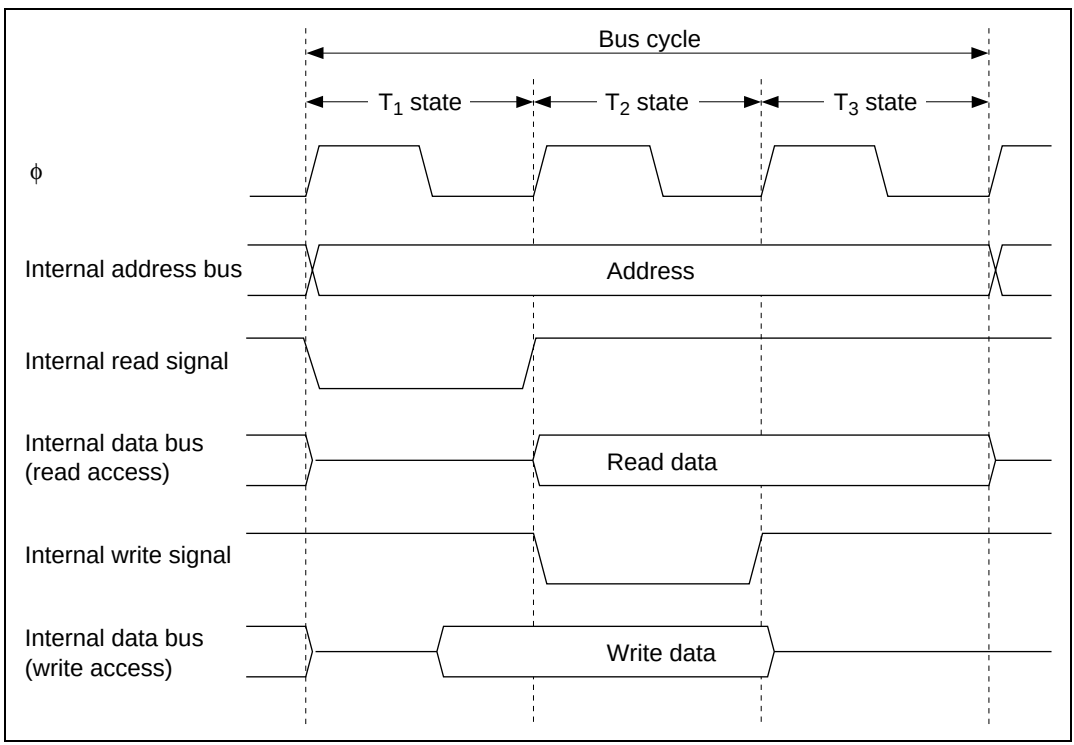


Figure 2.12 On-Chip Peripheral Module Access Cycle

2.7 CPU States

2.7.1 Overview

There are four CPU states: the reset state, program execution state, program halt state, and exception-handling state. The program execution state includes active (high-speed or medium-speed) mode. In the program halt state there are a sleep (high-speed or medium-speed) mode and standby mode. These states are shown in figure 2.13. Figure 2.14 shows the state transitions.

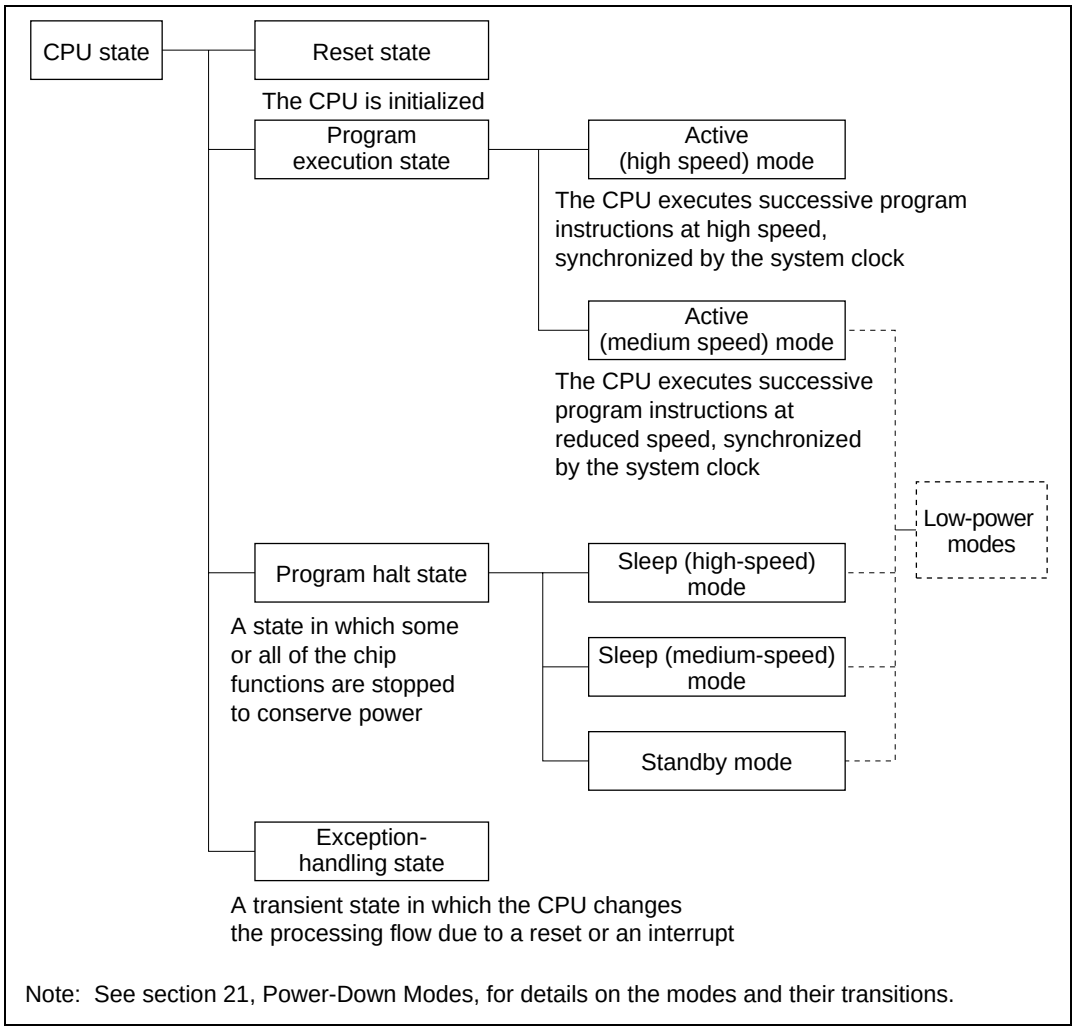


Figure 2.13 CPU Operation States

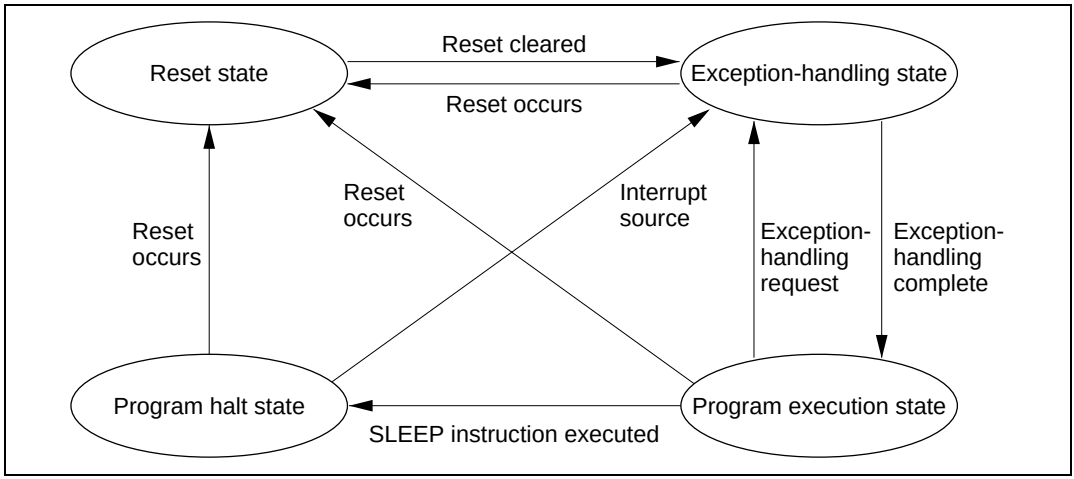


Figure 2.14 State Transitions

2.7.2 Reset State

The CPU is initialized in the reset state.

2.7.3 Program Execution State

In the program execution state the CPU executes program instructions in sequence.

There are two active modes (high-speed and medium-speed) when the CPU is in the program execution state.

2.7.4 Program Halt State

In the program halt state there are three modes: two sleep modes (high speed and medium speed) and standby mode. See section 21, Power-Down Modes for details on these modes.

2.7.5 Exception-Handling State

The exception-handling state is a transient state occurring when exception handling is started by a reset or interrupt and the CPU changes its normal processing flow. In exception handling caused by an interrupt, SP (R7) is referenced and the PC and CCR values are saved on the stack.

For details on interrupt handling, see section 4, Exception Handling.

2.8 Application Notes

2.8.1 Notes on Bit Manipulation

The BSET, BCLR, BNOT, BST, and BIST instructions read one byte of data, modify the data, then write the data byte again. Special care is required when using these instructions in cases where two registers are assigned to the same address, in the case of registers that include write-only bits, and when the instruction accesses an I/O port.

Order of Operation		Operation
1	Read	Read byte data at the designated address
2	Modify	Modify a designated bit in the read data
3	Write	Write the altered byte data to the designated address

As in the examples above, $P1_7$ and $P1_6$ are input pins, with a low-level signal input at $P1_7$ and a high-level signal at $P1_6$. The remaining pins, $P1_5$ to $P1_0$, are output pins that output low-level signals. In this example, the BCLR instruction is used to change pin $P1_0$ to an input port.

[A: Prior to executing BCLR]

	$P1_7$	**$P1_6$**	**$P1_5$**	**$P1_4$**	**$P1_3$**	**$P1_2$**	**$P1_1$**	**$P1_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
DDR	0	0	1	1	1	1	1	1
DR	1	0	0	0	0	0	0	0

[B: BCLR instruction executed]

```
BCLR   #0  ,  P1DDR
```

The BCLR instruction is executed designating DDR.

[C: After executing BCLR]

	$P1_7$	**$P1_6$**	**$P1_5$**	**$P1_4$**	**$P1_3$**	**$P1_2$**	**$P1_1$**	**$P1_0$**
Input/output	Output	Output	Output	Output	Output	Output	Output	Input
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
DDR	1	1	1	1	1	1	1	0
DR	1	0	0	0	0	0	0	0

[D: Explanation of how BCLR operates]

When the BCLR instruction is executed, first the CPU reads P1DDR. Since P1DDR is a write-only register, the CPU reads an undefined value. In this example, the DDR value is H'FF, but the data read by the CPU is undefined; it is taken to be H'FF.

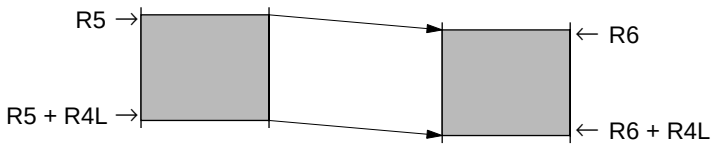
Next, the CPU clears bit 0 in the read data to 0, changing the data to H'FE.

Finally, this value (H'FE) is written to DDR and BCLR instruction execution ends.

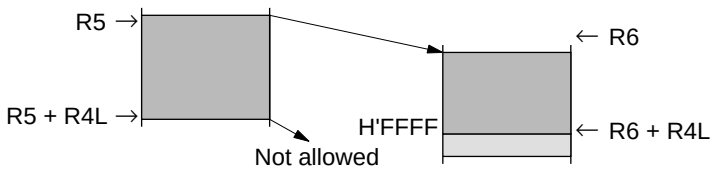
As a result of this operation, bit 0 in DDR becomes 0, making $P1_0$ an input port. However, bits 7 and 6 in DDR change to 1, so that $P1_7$ and $P1_6$ change from input pins to output pins.

2.8.2 Notes on Use of the EEPMOV Instruction (Cannot Be Used in the H8/3577 Group and H8/3567 Group)

- 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.



- 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.



Section 3 MCU Operating Modes

3.1 Overview

3.1.1 Operating Mode Selection

The H8/3577 Group and H8/3567 Group operate in the single-chip mode. The operating mode is specified by the setting of the mode pins (MD_1 to MD_0 or $\overline{TEST}$).

Table 3.1 lists the MCU operating modes.

Table 3.1 MCU Operating Mode Selection

- H8/3577 Group

MCU Operating Mode	MD_1	MD_0	Description
Mode 0	0	0	—
Mode 1	0	1	—
Mode 2	1	0	—
Mode 3	1	1	Single-chip mode

- H8/3567 Group

MCU Operating Mode	$\overline{TEST}$	Description
Mode 0	0	—
Mode 3	1	Single-chip mode

The H8/3577 Group and H8/3567 Group support the use of mode 3 only. Therefore, the mode pins must be set for mode 3 as indicated above.

3.1.2 Register Configuration

The H8/3577 Group and H8/3567 Group have a mode control register (MDCR) that indicates the inputs at the mode pins (MD_1 and MD_0 or $\overline{TEST}$), a system control register (SYSCR) that controls the operation of the MCU, and a serial timer control register (STCR) that controls the operation of the supporting modules. Table 3.2 summarizes these registers.

Table 3.2 MCU Registers

Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R	H'03	H'FFC5
System control register	SYSCR	R/W	H'09	H'FFC4
Serial timer control register	STCR	R/W	H'00	H'FFC3

Note: * Lower 16 bits of the address.

3.2 Register Descriptions

3.2.1 Mode Control Register (MDCR)

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

Note: * Determined by pins MD_1 and MD_0 or $\overline{TEST}$ pin.

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

Bit 7—Expanded Mode Enable (EXPE): This bit should not be set to 1.

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

Bits 1 and 0—Mode Select 1 and 0 (MDS1, MDS0): These bits indicate the input levels at pins MD_1, MD_0, and $\overline{TEST}$ (the current operating mode). Bits MDS1 and MDS0 correspond to MD_1 and MD_0 (H8/3577 Group). Alternately, bits MDS1 and MDS0 both correspond to the $\overline{TEST}$ pin (H8/3567 Group). MDS1 and MDS0 are read-only bits—they cannot be written to. The mode pin (MD_1, MD_0, and $\overline{TEST}$) input levels are latched into these bits when MDCR is read.

3.2.2 System Control Register (SYSCR)

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

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

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

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

Bit 7—Chip Select 2 Enable (CS2E): This bit should not be set to 1.

Bit 6—IOS Enable (IOSE): This bit should not be set to 1.

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

Bit 3

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

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

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

Bit 1

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

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

Bit 0

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

3.2.3 Serial Timer Control Register (STCR)

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

STCR is an 8-bit readable/writable register that controls register access, the IIC operating mode, selects the TCNT input clock and controls USB. For details of functions other than register access control, see the descriptions of the relevant modules. If a module controlled by STCR is not used, do not write 1 to the corresponding bit.

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

Bit 7—Reserved: Do not write 1 to this bit.

Bits 6 and 5—I^{2}C Control (IICX1, IICX0): These bits control the operation of the I^{2}C bus interface. For details, see section 16, I^{2}C Bus Interface.

Bit 4—I^{2}C Master Enable (IICE): Controls CPU access to the I^{2}C bus interface data registers and control registers (ICCR, ICSR, ICDR/SARX, and ICMR/SAR), the PWMX data registers and

control registers (DADRAH/DACR, DADRAL, DADRBH/DACNTH, and DADRBL/DACNTL), and the SCI control registers (SMR, BRR, and SCMR).

Bit 4

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

Bit 3—Reserved: Do not write 1 to this bit.

Bit 2—USB enable (USBE): This bit controls CPU access to the USB data register and control register.

Bit 2

USBE	Description
0	Prohibition of the above register access (Initial value)
1	Permission of the above register access

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

3.3 Address Map

Address maps are shown in figure 3.1 and figure 3.2.

The on-chip ROM capacity is 56 kbytes (H8/3577, H8/3567, H8/3567U) or 32 kbytes (H8/3574, H8/3564, H8/3564U). Do not try access to reserved areas and the addresses where no memory and no I/O register exists.

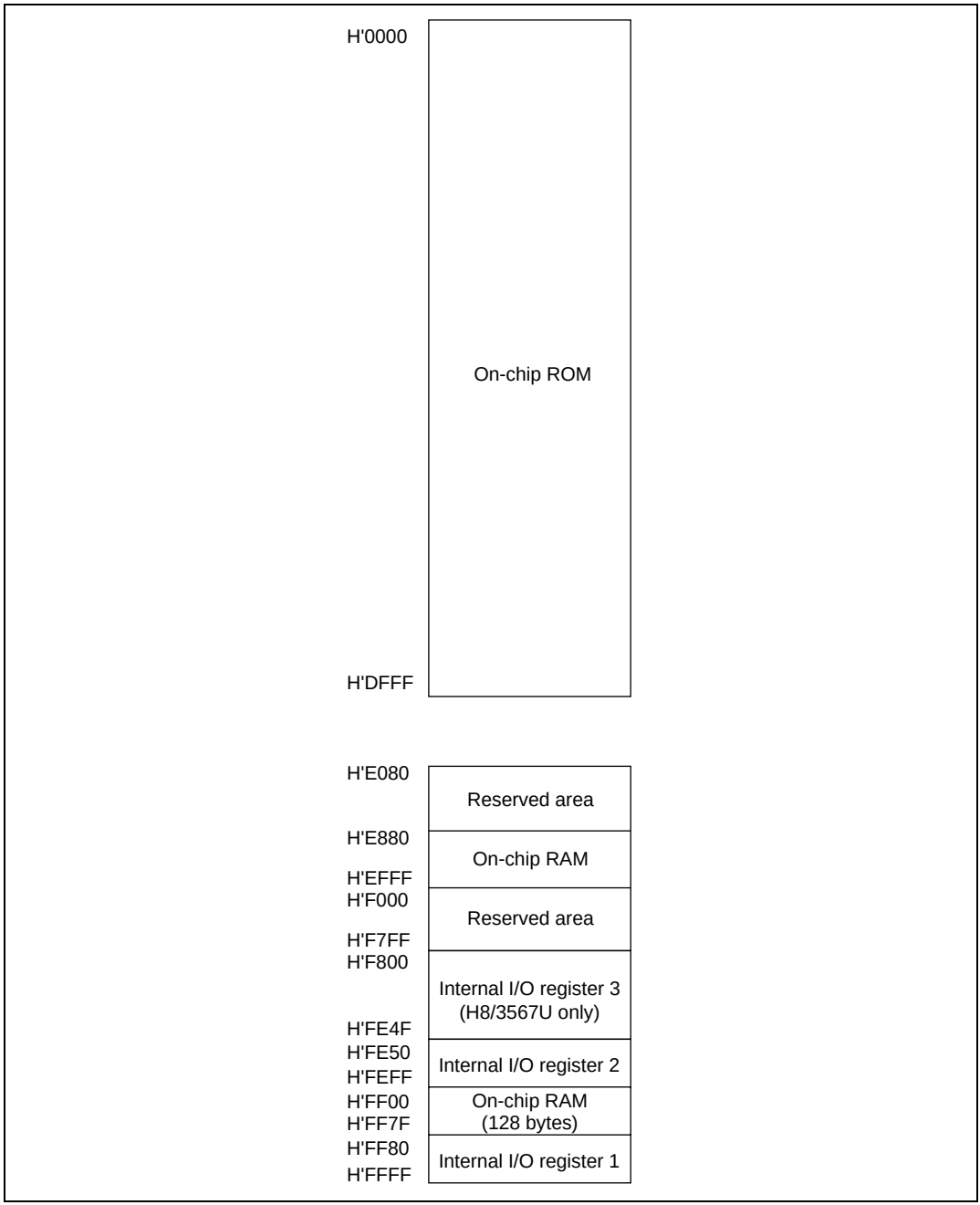


Figure 3.1 H8/3577, H8/3567, and H8/3567U Address Map

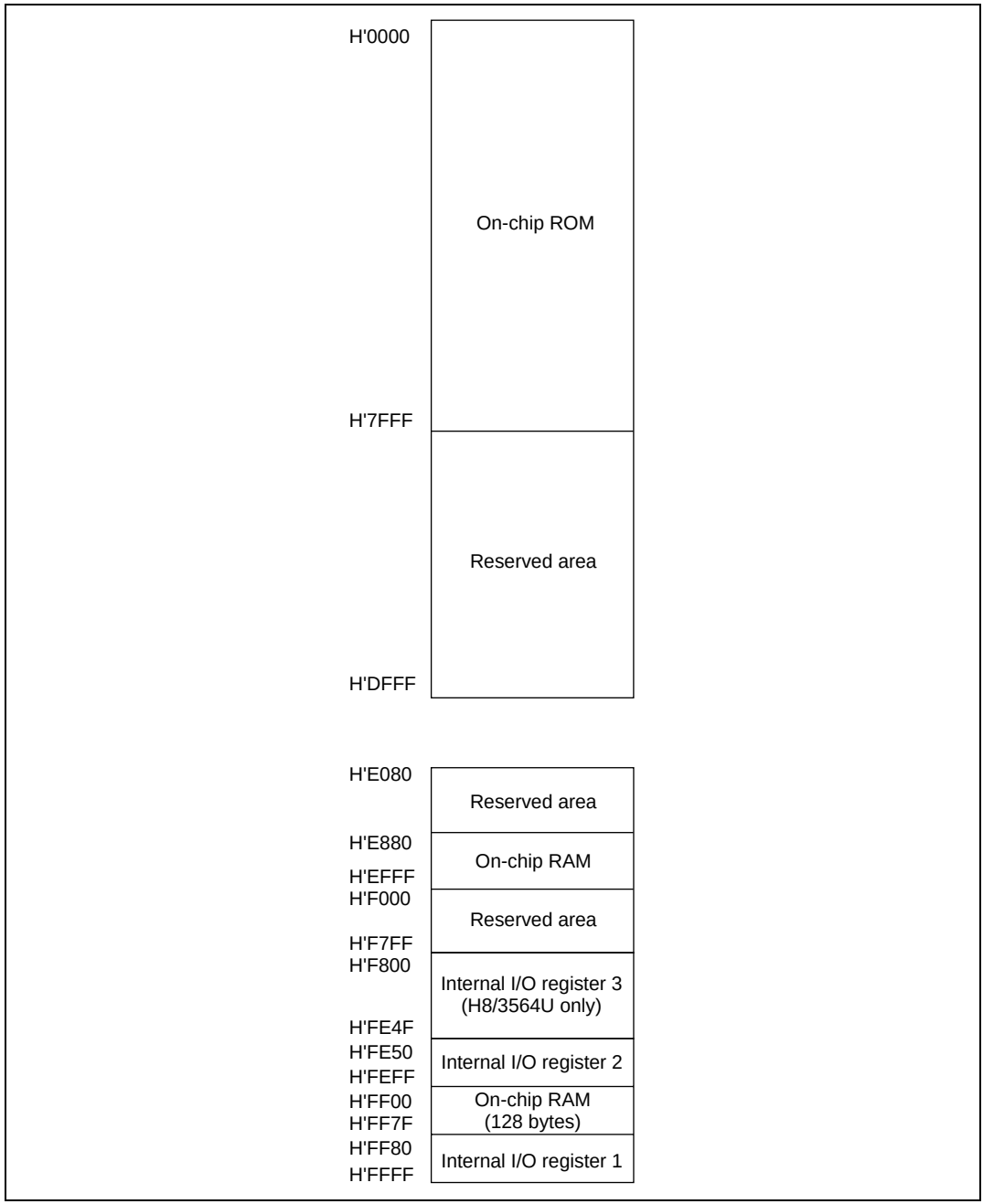


Figure 3.2 H8/3574, H8/3564, and H8/3564U Address Map

Section 4 Exception Handling

4.1 Overview

4.1.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, or interrupt. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High ↑ Low	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued.*

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

4.1.2 Exception Handling Operation

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

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

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

4.1.3 Exception Sources and Vector Table

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

Table 4.2 lists the exception sources and their vector addresses.

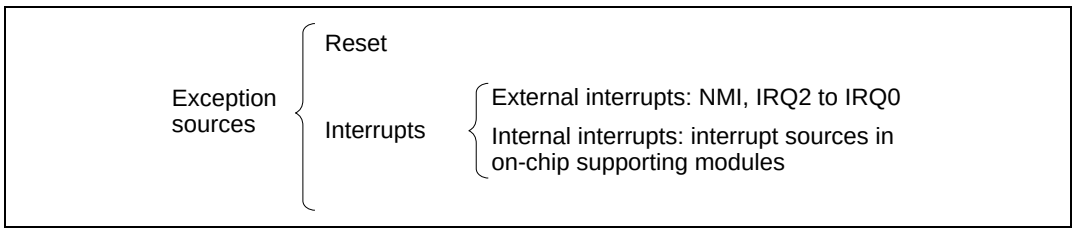


Figure 4.1 Exception Sources

Table 4.2 Exception Vector Table

Exception Source		Vector Number	Vector Address*
Reset		0	H'0000 to H'0001
Reserved for system use		1	H'0002 to H'0003
		2	H'0004 to H'0005
		3	H'0006 to H'0007
External interrupt	NMI	4	H'0008 to H'0009
	IRQ0	5	H'000A to H'000B
	IRQ1	6	H'000C to H'000D
	IRQ2	7	H'000E to H'000F
Reserved		8	H'0010 to H'0011
		9	H'0012 to H'0013
		10	H'0014 to H'0015
		11	H'0016 to H'0017
		12	H'0018 to H'0019
Internal interrupt*		13	H'001A to H'001B
		⋮	⋮
		53	H'006A to H'006B

Note: * For details on internal interrupt vectors, see section 5.3.3, Interrupt Exception Vector Table.

4.2 Reset

4.2.1 Overview

A reset has the highest exception priority.

When the $\overline{\text{RES}}$ pin goes low, all processing halts and the MCU enters the reset state. A reset initializes the internal state of the CPU and the registers of on-chip supporting modules.

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

MCUs can also be reset by overflow of the watchdog timer. For details, see section 14, Watchdog Timer.

4.2.2 Reset Sequence

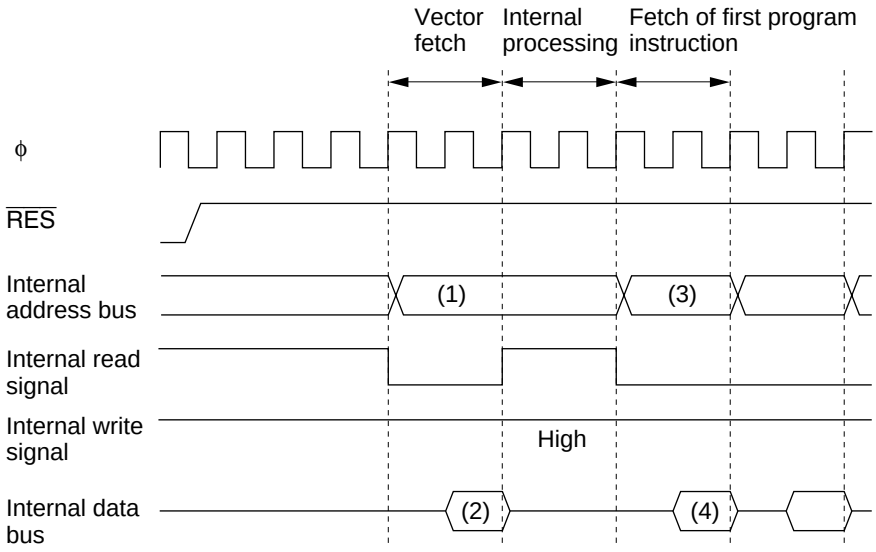
The MCU enters the reset state when the $\overline{\text{RES}}$ pin goes low.

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

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

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

Figure 4.2 shows an example of the reset sequence.



(1) Reset exception vector address ((1) = H'0000)
(2) Start address (contents of reset exception vector address)
(3) Start address ((3) = (2))
(4) First program instruction

Figure 4.2 Reset Sequence

4.2.3 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.W #xx:16, SP).

4.3 Interrupts

Interrupt exception handling can be requested by four external sources (NMI and IRQ2 to IRQ0), and internal sources in the on-chip supporting modules. Figure 4.3 shows the interrupt sources and the number of interrupts of each type.

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

NMI is the highest-priority interrupt. Interrupts are controlled by the interrupt controller.

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

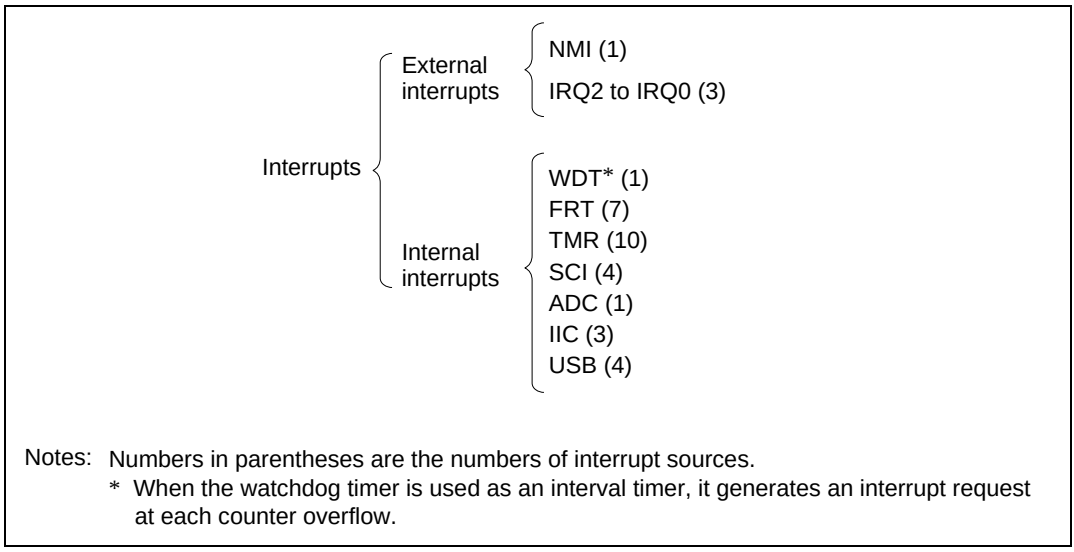
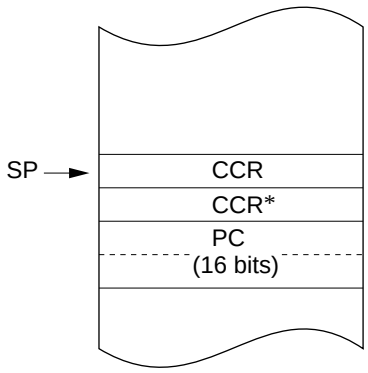


Figure 4.3 Interrupt Sources and Number of Interrupts

4.4 Stack Status after Exception Handling

Figure 4.4 shows the stack after completion of interrupt exception handling.



Interrupt control mode 0

Note: * Ignored on return.

Figure 4.4 Stack Status after Exception Handling

4.5 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 and POP (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 4.5 shows an example of damage caused when the stack pointer contains an odd address.

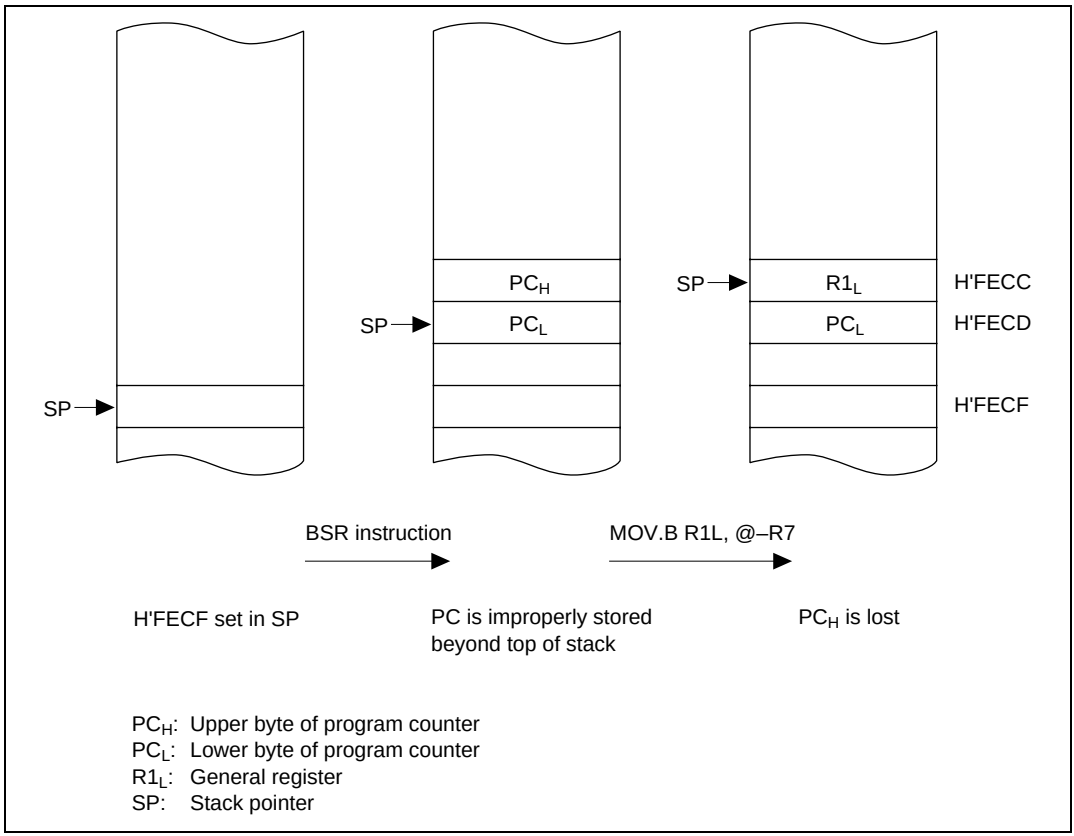


Figure 4.5 Example of Damage Caused by Setting an Odd Address in R7

Section 5 Interrupt Controller

5.1 Overview

5.1.1 Features

The MCUs control interrupts by means of an interrupt controller. The interrupt controller has the following features:

- Independent vector addresses
  - All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Four external interrupt pins
  - NMI is the highest-priority interrupt, and is accepted at all times. A rising or falling edge at the NMI pin can be selected for the NMI interrupt.
  - Falling edge, rising edge, or both edge detection, or level sensing, at pins $\overline{IRQ}_2$ to $\overline{IRQ}_0$ can be selected for interrupts IRQ2 to IRQ0.

5.1.2 Block Diagram

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

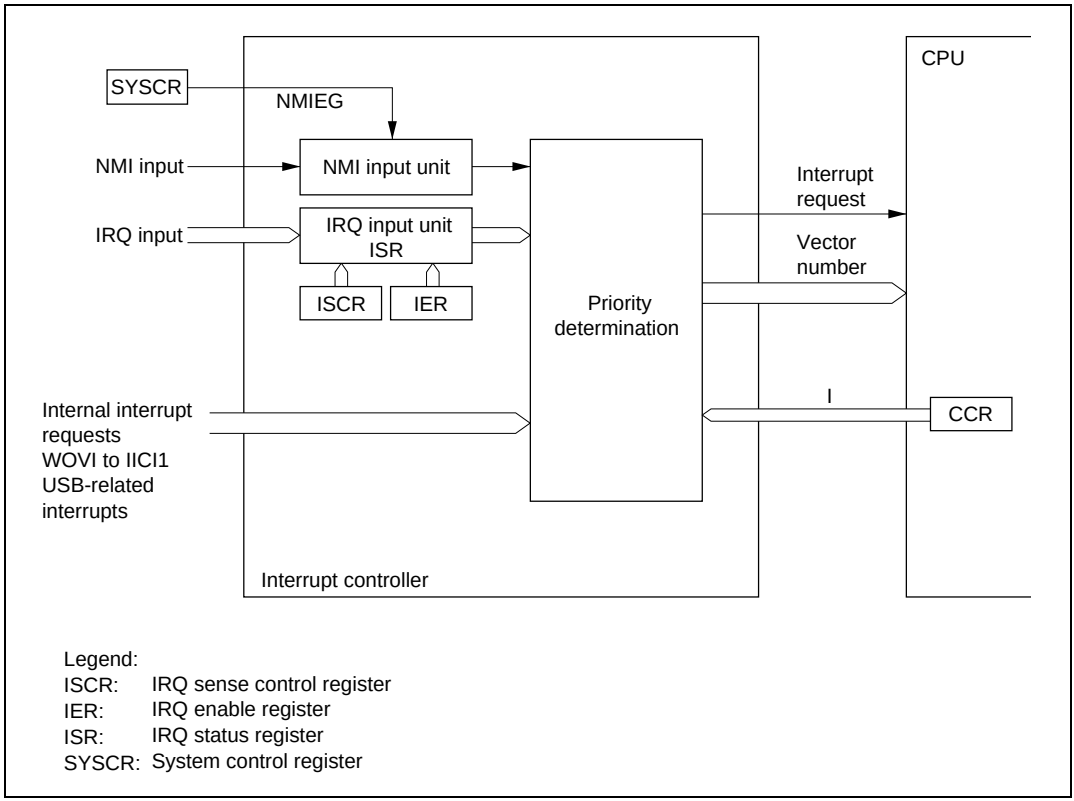


Figure 5.1 Block Diagram of Interrupt Controller

5.1.3 Pin Configuration

Table 5.1 summarizes the pins of the interrupt controller.

Table 5.1 Interrupt Controller Pins

Name	Symbol	I/O	Function
Nonmaskable interrupt	NMI	Input	Nonmaskable external interrupt; rising or falling edge can be selected
External interrupt requests 2 to 0	$\overline{IRQ_2}$ to $\overline{IRQ_0}$	Input	Maskable external interrupts; rising, falling, or both edges, or level sensing, can be selected

5.1.4 Register Configuration

Table 5.2 summarizes the registers of the interrupt controller.

Table 5.2 Interrupt Controller Registers

Name	Abbreviation	R/W	Initial Value	Address
System control register	SYSCR	R/W	H'09	H'FFC4
IRQ sense control register H	ISCRH	R/W	H'00	H'FEEC
IRQ sense control register L	ISCRL	R/W	H'00	H'FEED
IRQ enable register	IER	R/W	H'F8	H'FFC2
IRQ status register	ISR	R/(W)*	H'00	H'FEEB

Note: * Only 0 can be written, for flag clearing.

5.2 Register Descriptions

5.2.1 System Control Register (SYSCR)

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

SYSCR is an 8-bit readable/writable register, bit 2 of which selects the detected edge for NMI.

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

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

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): The INTM1 and 0 bits must not be set to 1.

Bit 5 INTM1	Bit 4 INTM0	Interrupt Control Mode	Description	
0	0	0	Interrupts are controlled by I bit	(Initial value)
	1	1	Cannot be used in H8/3577 Group and H8/3567 Group	
1	0	2	Cannot be used in H8/3577 Group and H8/3567 Group	
	1	3	Cannot be used in H8/3577 Group and H8/3567 Group	

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

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

5.2.2 IRQ Enable Register (IER)

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

IER is a register that controls enabling and disabling of interrupt requests IRQ2 to IRQ0.

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

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

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

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

Note: n = 2 to 0

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

- ISCRH

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

- ISCRL

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

ISCRH and ISCRL are 8-bit readable/writable registers that select rising edge, falling edge, or both edge detection, or level sensing, for the input at pins $\overline{IRQ}_2$ to $\overline{IRQ}_0$.

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

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

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

ISCRL Bits 5 to 0

IRQ2SCB to IRQ0SCB	IRQ2SCA to IRQ0SCA	Description
0	0	Interrupt request generated at $\overline{IRQ}_2$ to $\overline{IRQ}_0$ input low level (Initial value)
	1	Interrupt request generated at falling edge of $\overline{IRQ}_2$ to $\overline{IRQ}_0$ input
1	0	Interrupt request generated at rising edge of $\overline{IRQ}_2$ to $\overline{IRQ}_0$ input
	1	Interrupt request generated at both falling and rising edges of $\overline{IRQ}_2$ to $\overline{IRQ}_0$ input

5.2.4 IRQ Status Register (ISR)

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

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

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

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

Bits 7 to 3—Reserved

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

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

Note: n = 2 to 0

5.3 Interrupt Sources

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

5.3.1 External Interrupts

There are four external interrupt sources: NMI, and $\overline{IRQ_2}$ to $\overline{IRQ_0}$. NMI, and IRQ2 to IRQ0 can be used to restore the H8/3577 Group and H8/3567 Group chip from software standby mode.

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

The vector number for NMI interrupt exception handling is 4.

IRQ2 to IRQ0 Interrupts: Interrupts IRQ2 to IRQ0 are requested by an input signal at pins $\overline{IRQ_2}$ to $\overline{IRQ_0}$. Interrupts IRQ2 to IRQ0 have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins $\overline{IRQ_2}$ to $\overline{IRQ_0}$.
- Enabling or disabling of interrupt requests IRQ2 to IRQ0 can be selected with IER.
- The status of interrupt requests IRQ2 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

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

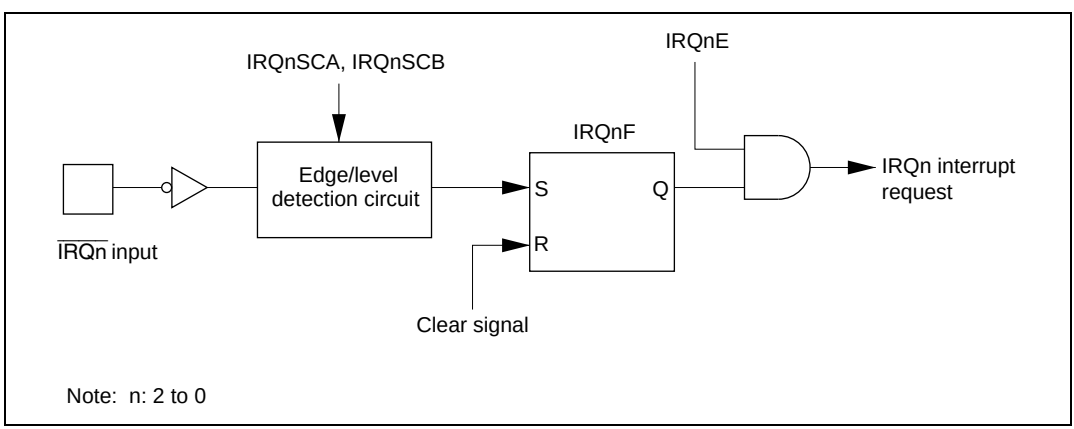


Figure 5.2 Block Diagram of Interrupts IRQ2 to IRQ0

Figure 5.3 shows the timing of IRQnF setting.

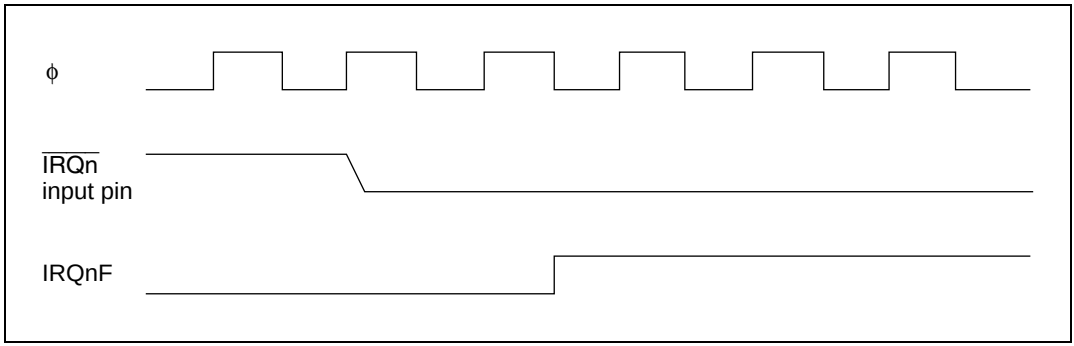


Figure 5.3 Timing of IRQnF Setting

The vector numbers for IRQ2 to IRQ0 interrupt exception handling are 7 to 5.

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

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

5.3.2 Internal Interrupts

There are 26 sources (30 sources in the version with an on-chip USB) for internal interrupts from on-chip supporting modules.

For each on-chip supporting module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. If any one of these is set to 1, an interrupt request is issued to the interrupt controller.

5.3.3 Interrupt Exception Vector Table

Table 5.3 shows interrupt exception handling sources, vector addresses, and interrupt priorities. For default priorities, the lower the vector number, the higher the priority.

Priorities within a module are fixed as shown in table 5.3.

Table 5.3 Interrupt Exception Handling Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	Priority
NMI	External pin	4	H'0008	High
IRQ0		5	H'000A	
IRQ1		6	H'000C	
IRQ2		7	H'000E	
Reserved	—	8 to 12	H'0010 to H'0018	
WOVI0 (interval timer)	Watchdog timer 0	13	H'001A	
ADI (A/D conversion end)	A/D	14	H'001C	
ICIA (input capture A)	Free-running timer	15	H'001E	
ICIB (input capture B)		16	H'0020	
ICIC (input capture C)		17	H'0022	
ICID (input capture D)		18	H'0024	
OCIA (output compare A)		19	H'0026	
OCIB (output compare B)		20	H'0028	
FOVI (overflow)		21	H'002A	
CMIA0 (compare-match A)	8-bit timer channel 0	22	H'002C	
CMIB0 (compare-match B)		23	H'002E	
OVI0 (overflow)		24	H'0030	
CMIA1 (compare-match A)	8-bit timer channel 1	25	H'0032	
CMIB1 (compare-match B)		26	H'0034	
OVI1 (overflow)		27	H'0036	Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	Priority
CMIAY (compare-match A)	8-bit timer channels Y, X	28	H'0038	High
CMIBY (compare-match B)		29	H'003A	
OVIY (overflow)		30	H'003C	
ICIX (input capture X)		31	H'003E	
Reserved	—	32 to 35	H'0040 to H'0046	
ERI0 (receive error 0)	SCI channel 0	36	H'0048	
RXI0 (reception completed 0)		37	H'004A	
TXI0 (transmit data empty 0)		38	H'004C	
TEI0 (transmission end 0)		39	H'004E	
Reserved	—	40 to 43	H'0050 to H'0056	
IICI0 (1-byte transmission/ reception completed)	IIC channel 0	44	H'0058	
DDCSWI (format switch)		45	H'005A	
IICI1 (1-byte transmission/ reception completed)	IIC channel 1	46	H'005C	
Reserved	—	47 to 49	H'005E to H'0062	
USBIA	USB	50	H'0064	
USBIB		51	H'0066	
USBIC		52	H'0068	
USBID		53	H'006A	Low

5.4 Interrupt Operation

5.4.1 Interrupt Operation

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

Table 5.4 shows the interrupt control modes.

Table 5.4 Interrupt Control Modes

	SYSCR			
Interrupt Control Mode	**INTM1**	**INTM0**	**Interrupt Mask Bits**	**Description**
0	0	0	I	Interrupt mask control is performed by the I bit

Figure 5.4 shows a block diagram of the priority decision circuit.

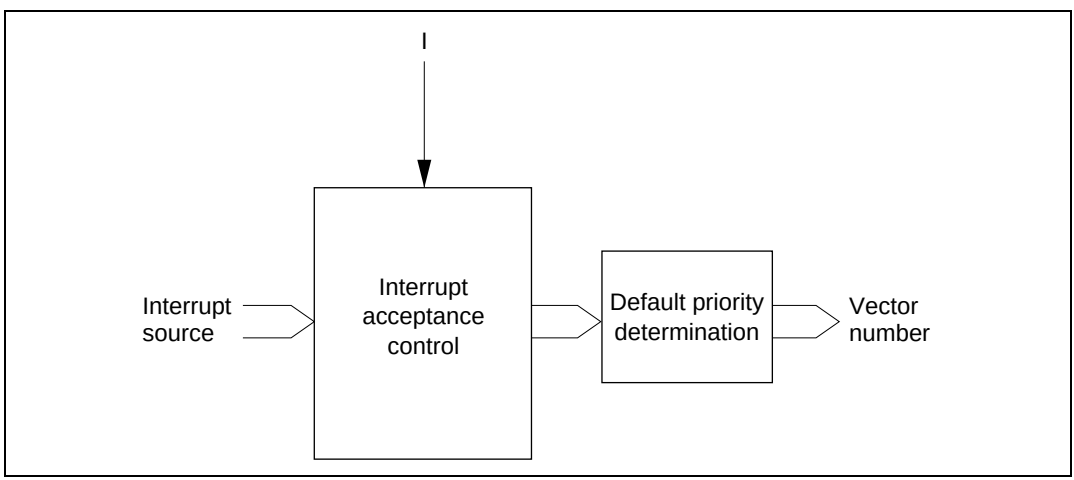


Figure 5.4 Block Diagram of Interrupt Control Operation

Interrupt Acceptance Control: In interrupt control mode 0, interrupt acceptance control is performed by means of the I bit in CCR.

Table 5.5 shows the interrupts selected in each interrupt control mode.

Table 5.5 Interrupts Selected in Each Interrupt Control Mode

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

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

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

Table 5.6 shows operations and control signal functions in each interrupt control mode.

Table 5.6 Operations and Control Signal Functions in Each Interrupt Control Mode

Interrupt Control Mode	Setting		Interrupt Acceptance Control		Determination
	INTM1	INTM0		I	
0	0	0	O	IM	O

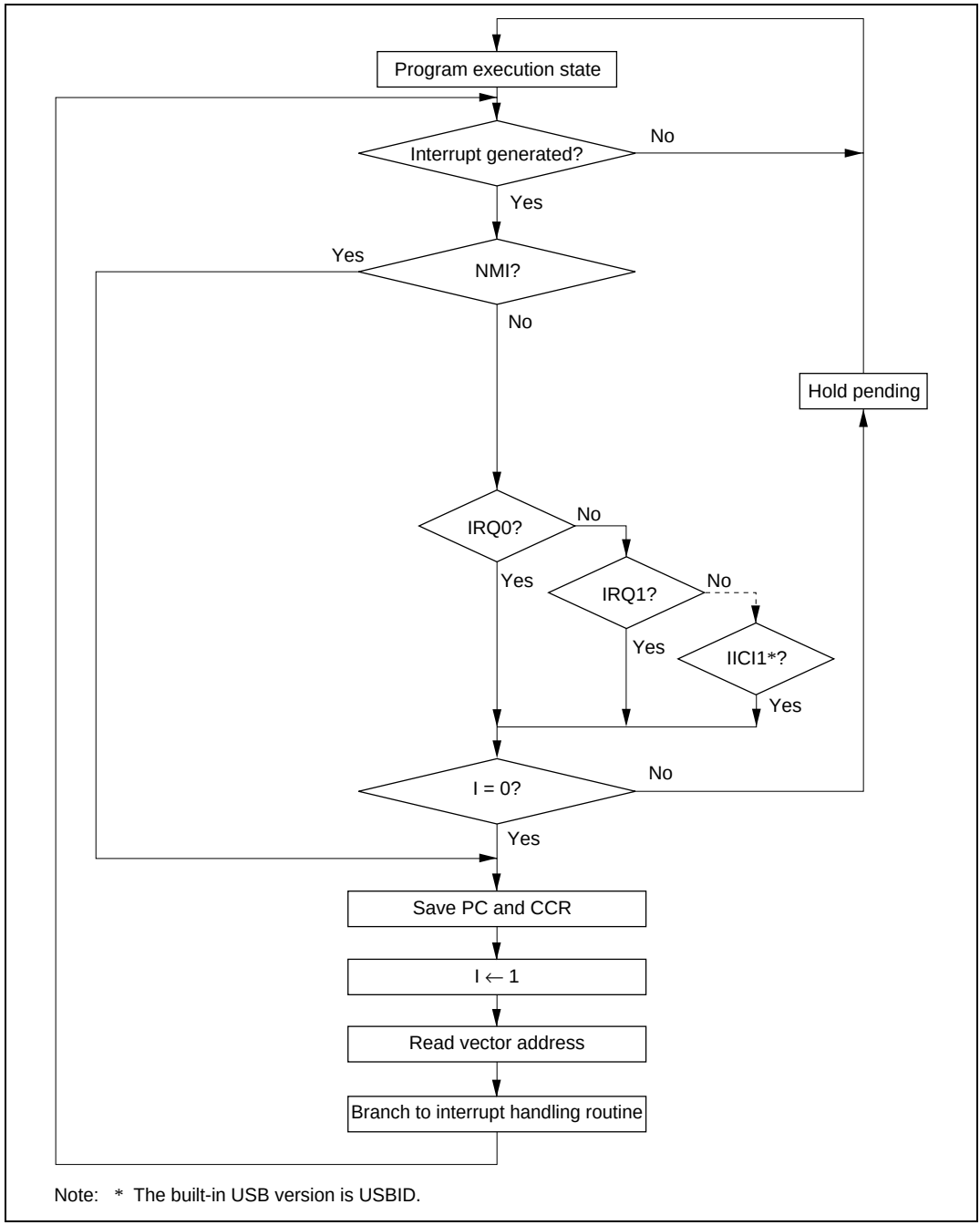
Legend:
O: Interrupt operation control performed
IM: Used as interrupt mask bit

5.4.2 Interrupt Control Mode 0

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

Figure 5.5 shows a flowchart of the interrupt acceptance operation in this case.

1. If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.
2. If a number of interrupt requests are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.3 is selected.
3. The I bit is then referenced. If the I bit is cleared to 0, the interrupt request is accepted. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending.
4. When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.
5. The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.
6. Next, the I bit in CCR is set to 1. This disables all interrupts except NMI.
7. A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.



Note: * The built-in USB version is USBID.

Figure 5.5 Flowchart of Procedure Up to Interrupt Acceptance

5.4.3 Interrupt Exception Handling Sequence

Figure 5.6 shows the interrupt exception handling sequence.

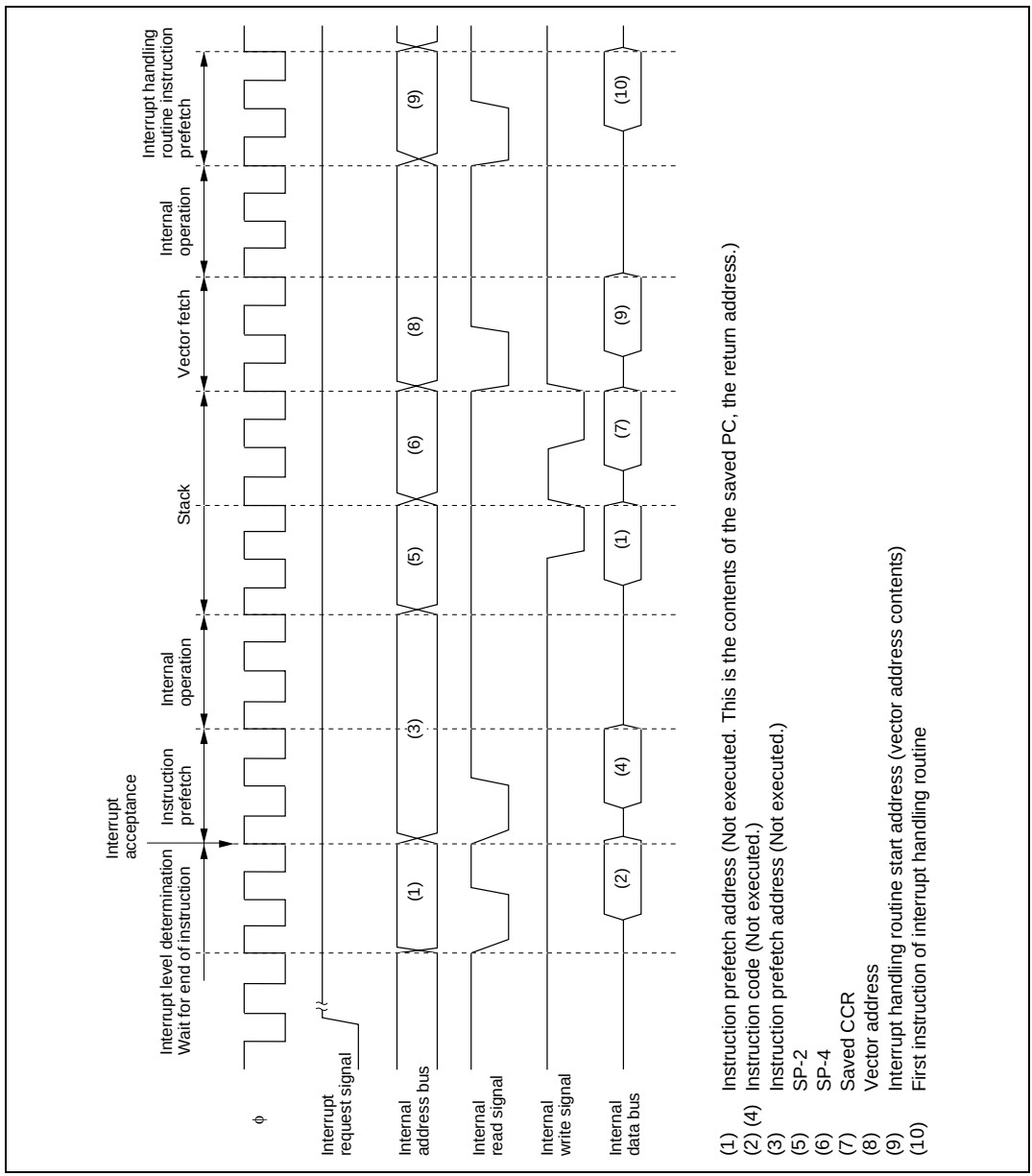


Figure 5.6 Interrupt Exception Handling

5.4.4 Interrupt Response Times

Table 5.7 shows interrupt response times—the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine.

Table 5.7 Interrupt Response Times

		Number of States
No.	**Item**	**Normal Mode**
1	Interrupt priority determination*1	3
2	Number of wait states until executing instruction ends*2	1 to 13
3	PC, CCR stack save	4
4	Vector fetch	2
5	Instruction fetch*3	4
6	Internal processing*4	4
Total		18 to 30

Notes: 1. Two states in case of internal interrupt.
2. Refers to MULXS and DIVXS instructions. Except EEPMOV instruction.
3. Prefetch after interrupt acceptance and interrupt handling routine prefetch.
4. Internal processing after interrupt acceptance and internal processing after vector fetch.

5.5 Usage Notes

5.5.1 Contention between Interrupt Generation and Disabling

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

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

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

Figure 5.7 shows an example in which the CMIEA bit in 8-bit timer register TCR is cleared to 0.

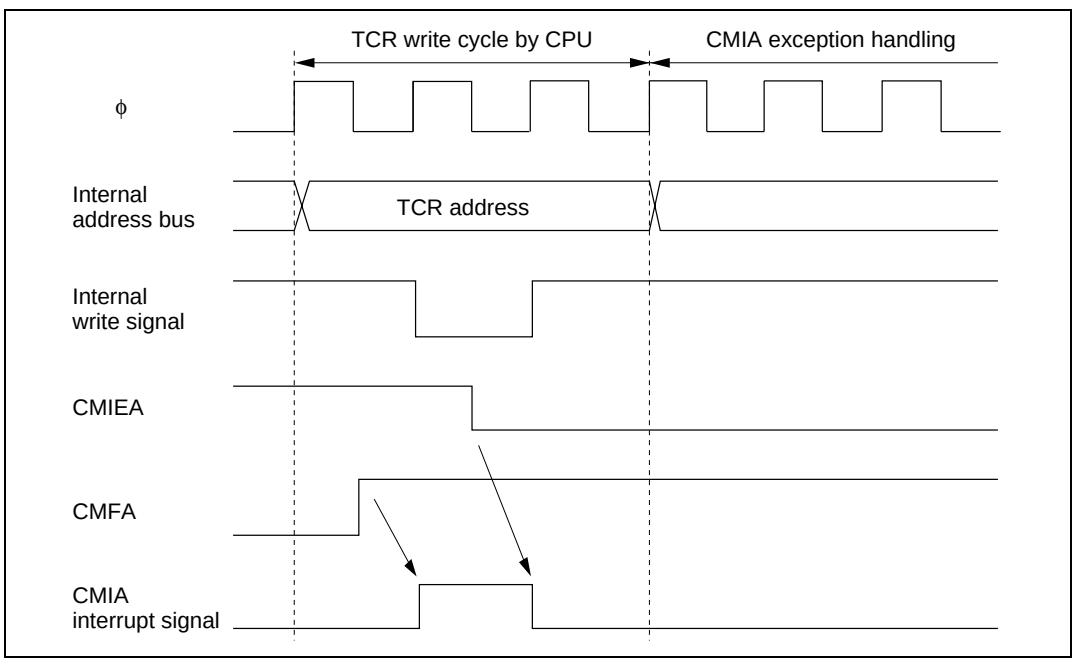


Figure 5.7 Contention between Interrupt Generation and Disabling

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

5.5.2 Instructions that Disable Interrupts

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

5.5.3 Interrupts during Execution of EEPMOV Instruction

With the EEPMOV instruction, an interrupt request (including NMI) issued during the transfer is not accepted until the move is completed. The EEPMOV instruction cannot be used in the H8/3577 Group and H8/3567 Group.

Section 6 Bus Controller

6.1 Overview

As the H8/3577 Group and H8/3567 Group do not have external expansion functions, they do not incorporate a bus controller function.

However, from the viewpoint of maintaining software compatibility with similar products, care must be taken not to set inappropriate values in the bus controller related control registers.

6.2 Register Descriptions

6.2.1 Bus Control Register (BCR)

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

Bits 7 and 6—Idle Cycle Insert 1 and 0 (ICIS1, ICIS0): Do not write 0 to these bits.

Bit 5—Burst ROM Enable (BRSTRM): Do not write 1 to this bit.

Bit 4—Burst Cycle Select 1 (BRSTS1): Do not write 0 to this bit.

Bit 3—Burst Cycle Select 0 (BRSTS0): Do not write 1 to this bit.

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

Bits 1 and 0—IOS Select 1 and 0 (IOS1, IOS0): Do not write 0 to these bits.

6.2.2 Wait State Control Register (WSCR)

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

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

Bit 5—Bus Width Control (ABW): Do not write 0 to this bit.

Bit 4—Access State Control (AST): Do not write 0 to this bit.

Bits 3 and 2—Wait Mode Select 1 and 0 (WMS1, WMS0): Do not write 1 to these bits.

Bits 1 and 0—Wait Count 1 and 0 (WC1, WC0): Do not write 0 to these bits.

Section 7 Universal Serial Bus Interface (USB)

It is built in the H8/3567U and H8/3564U Group and not in the H8/3577, H8/3574, H8/3567 and H8/3564 Group.

7.1 Overview

The H8/3567U and H8/3564U have an on-chip universal serial bus (USB) comprising hubs and a function. The universal serial bus is an interface for personal computer peripherals whose standardization is being promoted by a core group of companies, including Intel Corporation.

The USB is provided with a number of device classes to handle the great variety of personal computer peripheral devices. The USB in the H8/3567U and H8/3564U are targeted at the hub device class and HID (Human Interface Device) class (mainly a monitor device class).

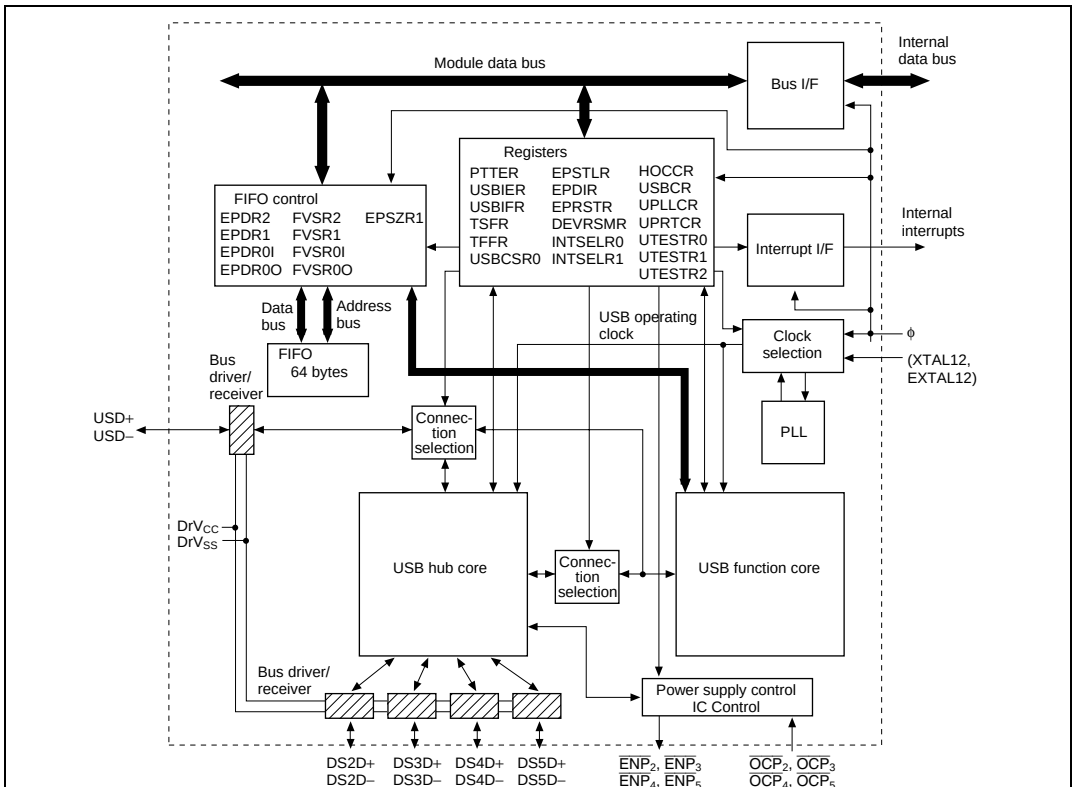
7.1.1 Features

- Compound device conforming to USB standard*
  - Apart from initial settings and power-down mode settings, USB hubs decode and execute hub class commands automatically, independently of CPU operations
  - USB function decodes and executes standard commands
    Device class commands are decoded and executed by the CPU (firmware creation required)
- Five downstream hubs and one function
  - One down stream is connected internally to the USB function
  - Internal downstream disconnection function
    (Only power-down mode USB hubs operable)
  - Four sets of downstream external pins
  - Automatic control of downstream port external power supply control IC (individual port control)
- Three-endpoint monitor device class function
  - EP0: USB control endpoint (dedicated to control transfer)
  - EP1, EP2: Monitor control endpoints (dedicated to interrupt transfer)
  - EP0I, EP0O, and EP2 can use a maximum 16-byte FIFO (maximum packet size of 8 bytes), and EP1 can use a maximum 32-byte FIFO (maximum packet size of 16 bytes)
- Supports 12 Mbps high-speed transfer mode
- Built-in 12 MHz clock pulse generator and frequency division/multiplication circuit
- Built-in bus driver/receiver
  - Driven by DrV_{SS}/DrV_{CC} (3.3 V)

Note: * The USB function conforms to USB Standard 1.1 and the USB hub to USB Standard 1.0.

7.1.2 Block Diagram

Figure 7.1 shows a block diagram of the USB.



Legend:

EPDR2:	Endpoint data register 2	EPRSTR:	Endpoint reset register	DS4D+:	Downstream 4 data + pin
EPDR1:	Endpoint data register 1	DEVRSMR:	Device resume register	DS4D–:	Downstream 4 data – pin
EPDR0I:	Endpoint data register 0I	INTSELR0:	Interrupt source select register 0	DS5D+:	Downstream 5 data + pin
EPDR0O:	Endpoint data register 0O	INTSELR1:	Interrupt source select register 1	DS5D–:	Downstream 5 data – pin
FVSR2:	FIFO valid size register 2	HOCCR:	Hub overcurrent control register	XTAL12:	USB clock oscillator pin
FVSR1:	FIFO valid size register 1	USBCR:	USB control register	EXTAL12:	USB clock oscillator pin
FVSR0I:	FIFO valid size register 0I	UPLLCR:	USB PLL control register	DrV_{CC}:	Bus driver power supply pin
FVSR0O:	FIFO valid size register 0O	UPRTCR:	USB port control register	DrV_{SS}:	Bus driver ground pin
EPSZR1:	Endpoint size register 1	UTESTR0:	USB test register 0	$\overline{OCP}_2$:	Overcurrent detection pin 2
PTTER:	Packet transmit enable register	UTESTR1:	USB test register 1	$\overline{OCP}_3$:	Overcurrent detection pin 3
USBIER:	USB interrupt enable register	UTESTR2:	USB test register 2	$\overline{OCP}_4$:	Overcurrent detection pin 4
USBIFR:	USB interrupt flag register	USD+:	Upstream data + pin	$\overline{OCP}_5$:	Overcurrent detection pin 5
TSFR:	Transfer success flag register	USD–:	Upstream data – pin	$\overline{ENP}_2$:	Power supply output enable pin 2
TFFR:	Transfer fail flag register	DS2D+:	Downstream 2 data + pin	$\overline{ENP}_3$:	Power supply output enable pin 3
USBCSR0:	USB control/status register 0	DS2D–:	Downstream 2 data – pin	$\overline{ENP}_4$:	Power supply output enable pin 4
EPSTLR:	Endpoint stall register	DS3D+:	Downstream 3 data + pin	$\overline{ENP}_5$:	Power supply output enable pin 5
EPDIR:	Endpoint direction register	DS3D–:	Downstream 3 data – pin		

Figure 7.1 Block Diagram of USB

7.1.3 Pin Configuration

Table 7.1 shows the pins used by the USB.

Table 7.1 USB Pins

Name	Abbre-viation	I/O	Function
Upstream data + pin	USD+	Input/output	USB hub/function data input/output
Upstream data – pin	USD–	Input/output	
Downstream 2 data + pin	DS2D+	Input/output	USB hub repeater input/output (port 2)
Downstream 2 data – pin	DS2D–	Input/output	
Downstream 3 data + pin	DS3D+	Input/output	USB hub repeater input/output (port 3)
Downstream 3 data – pin	DS3D–	Input/output	
Downstream 4 data + pin	DS4D+	Input/output	USB hub repeater input/output (port 4)
Downstream 4 data – pin	DS4D–	Input/output	
Downstream 5 data + pin	DS5D+	Input/output	USB hub repeater input/output (port 5)
Downstream 5 data – pin	DS5D–	Input/output	
Overcurrent detection pins 2 to 5	$\overline{OCP}_2$ to $\overline{OCP}_5$	Input	Power supply control IC overcurrent detection signal input
Power supply output enable control pins 2 to 5	$\overline{ENP}_2$ to $\overline{ENP}_5$	Output	Power supply control IC power output enable signal output
USB clock oscillator pin	XTAL12	Input	12 MHz crystal oscillation
USB clock oscillator pin	EXTAL12	Input	
Bus Driver power supply pin	DrV_{CC}	Input	Bus driver/receiver, port D power supply
Bus Driver ground pin	DrV_{SS}	Input	Bus driver/receiver, port D ground

7.1.4 Register Configuration

The USB register configuration is shown in table 7.2. Registers relating to USB hub initialization and status display are USBCR, USBCSR0, HOCCR, and UPLLCR, as well as some bits in the test registers; the other registers relate to the USB function.

When USBCR, USBCSR0, HOCCR, and UPLLCR are all in the initial state, the USB module is completely disabled, and ports C and D function as I/O ports.

When accessing a USB register, the USBE bit in STCR must be set to 1.

Table 7.2 USB Registers

Name	Abbreviation	R/W	Initial Value	Address
Endpoint data register 2	EPDR2	R or W*1	H'00	H'FDE1
FIFO valid size register 2	FVSR2	R	H'0010	H'FDE2
Endpoint size register 1	EPSZR1	R/W	H'44	H'FDE4
Endpoint data register 1	EPDR1	W	H'00	H'FDE5
FIFO valid size register 1	FVSR1	R	H'0010	H'FDE6
Endpoint data register 0O	EPDR0O	R	H'00	H'FDE9
FIFO valid size register 0O	FVSR0O	R	H'0000	H'FDEA
Endpoint data register 0I	EPDR0I	W	H'00	H'FDED
FIFO valid size register 0I	FVSR0I	R	H'0010	H'FDEE
Packet transmit enable register	PTTER	R/(W)*2	H'00	H'FDF0
USB interrupt enable register	USBIER	R/W	H'00	H'FDF1
USB interrupt flag register	USBIFR	R/(W)*3	H'00	H'FDF2
Transfer success flag register	TSFR	R/(W)*3	H'00	H'FDF3
Transfer fail flag register	TFFR	R/(W)*3	H'00	H'FDF4
USB control/status register 0	USBCSR0	R/W	H'00	H'FDF5
Endpoint stall register	EPSTLR	R/W	H'00	H'FDF6
Endpoint direction register	EPDIR	R/W	H'FC	H'FDF7
Endpoint reset register	EPRSTR	R/(W)*2	H'00	H'FDF8
Device resume register	DEVRSMR	R/(W)*2	H'00	H'FDF9
Interrupt source select register 0	INTSELR0	R/W	H'00	H'FDFA
Interrupt source select register 1	INTSELR1	R/W	H'00	H'FDFB
Hub overcurrent control register	HOCCR	R/W	H'00	H'FDFC
USB control register	USBCR	R/W	H'7F	H'FDFD
USB PLL control register	UPLLCR	R/W	H'01	H'FDFE
USB port control register	UPRTCR	R/W	H'00	H'FDC0
USB test register 0	UTESTR0	R/W	H'00	H'FDC1
USB test register 1	UTESTR1	R/W	H'00	H'FDC2
USB test register 2	UTESTR2	R/W	H'FF	H'FDFF
Other test registers	—	—	—	H'FDC3 to H'FDE0

Name	Abbreviation	R/W	Initial Value	Address
Serial timer control register	STCR	R/W	H'00	H'FFC3
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

Notes: 1. Write-only or read-only depending on the transfer direction set in the endpoint direction register.
2. Only 1 can be written.
3. Only 0 can be written after reading 1 to clear the flags.

7.2 Register Descriptions

In the USB protocol, the host transmits a token to initiate a single data transfer (a transaction). A transaction consists of a token packet, data packet, and handshake packet. The token packet contains the address endpoint of the transfer target device and the transfer type, the data packet contains data, and the handshake packet contains information relating to transfer setup/non-setup.

In data transfer from the host to a slave, the host transmits an OUT token or SETUP token, followed by data (an OUT or SETUP transaction). In data transfer from a slave to the host, the host transmits an IN token and waits for data from the slave (an IN transaction). In the following descriptions, these host-based IN and OUT operations may be referred to as “input” and “output.” Also, items relating to host input transfer may be designated “IN” (IN transaction, IN-FIFO, EP0in, etc.), while items relating to host output transfer are designated “OUT” (OUT transaction, OUT-FIFO, EP0out, etc.).

Where an explicit expression such as “transmitted by the host” or “received by the host” is not used, the terms “transmission” and “reception” refer to transmission and reception from the standpoint of the USB module and slave CPU.

7.2.1 USB Data FIFO

The FIFO, together with EPDR, functions as an intermediary role in data transfer between the H8 CPU (slave) and the USB function. The USB function uses the FIFO to execute data transfer to and from the USB host (host).

The H8/3567U and H8/3564U have an on-chip 64-byte FIFO. This FIFO is divided into four 16-byte FIFOs, used for endpoint 0 host input transfer and host output transfer (control transfer), endpoint 1 host input transfer (interrupt transfer), and endpoint 2 host input transfer or host output transfer. If endpoint 2 is not used, a 32-byte length can be selected for the endpoint 1 FIFO. The maximum data packet size is set at half the number of FIFO bytes.

In host input transfer, all the data to be transmitted from the slave is written to the FIFO before slave transmission is started. In host output transfer, the slave reads all the data from the FIFO after host output transfer is completed.

7.2.2 Endpoint Size Register 1 (EPSZR1)

Bit	7	6	5	4	3	2	1	0
	EP1SZ3	EP1SZ2	EP1SZ1	EP1SZ0	EP2SZ3	EP2SZ2	EP2SZ1	EP2SZ0
Initial value	0	1	0	0	0	1	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EPSZR1 specifies the number of FIFO bytes used for each USB function endpoint 1 and 2 host input transfer/host output transfer. The number of bytes in the endpoint 0 FIFO is fixed at 16. Both host input (EP0in) and host output (EP0out) can be selected for endpoint 0, host input for endpoint 1, and host input and host output for endpoint 2.

With the H8/3567U and H8/3564U, when endpoints 1 and 2 are both used, set a 16-byte size for the respective FIFOs. When only endpoint 1 is used, set a 16- or 32-byte size. If the 32-byte size is selected, set 0 as the endpoint 2 FIFO size.

EPSZR1 is initialized to H'44 by a system reset or a function soft reset.

EPSZR1	Bits 7 to 4	EP1 FIFO size
EPSZR1	Bits 3 to 0	EP2 FIFO size

Bit 7 Bit 3	**Bit 6 Bit 2**	**Bit 5 Bit 1**	**Bit 4 Bit 0**	
SZ3	**SZ2**	**SZ1**	**SZ0**	**Operating Mode**
0	0	0	0	FIFO size = 0 bytes (settable for EP2 only)
			1	Setting prohibited
		1	0	Setting prohibited
			1	Setting prohibited
	1	0	0	FIFO size = 16 bytes (Initial value)
			1	FIFO size = 32 bytes (settable for EP1 only)
		1	0	Setting prohibited
			1	Setting prohibited
1	—	—	—	Setting prohibited

7.2.3 Endpoint Data Registers 0I, 0O, 1, 2 (EPDR0I, EPDR0O, EPDR1, EPDR2)

Bit		7	6	5	4	3	2	1	0
		D7	D6	D5	D4	D3	D2	D1	D0
Initial value		0	0	0	0	0	0	0	0
Read/ Write	EPDR0I	W	W	W	W	W	W	W	W
	EPDR0O	R	R	R	R	R	R	R	R
	EPDR1	W	W	W	W	W	W	W	W
	EPDR2	R or W*	R or W*	R or W*	R or W*	R or W*	R or W*	R or W*	R or W*

Note: * Write-only or read-only depending on the transfer direction set in the endpoint direction register.

The EPDR registers play an intermediary role in data transfer between the CPU and FIFO for each host input transfer/host output transfer involving the respective USB function endpoints. EPDR0I and EPDR1 are used for host input transfer, and so are write-only registers; if read, the contents of the read data are not guaranteed. EPDR0O is used for host output transfer, and so is a read-only register; it cannot be written to.

For EPDR2, the endpoint transfer direction is determined by the endpoint direction register. EPDR2 is a write-only register when designated for host input transfer, and a read-only register when designated for host output transfer. If EPDR2 is read when functioning as a write-only register, the contents of the read data are not guaranteed. When EPDR2 is functioning as a read-only register, it cannot be written to.

Data written to EPDR0I, EPDR1, or EPDR2 (when a write-only register) is stored in the FIFO, and is made valid by setting the EPTE bit in the packet transmit enable register (PTTER). Valid data is transferred to the USB function, and transferred to the host, in accordance with a USB function request.

Data transferred from the host is stored in the FIFO by the USB function, and becomes valid when all the data packet bytes have been received and an ACK handshake is transmitted. When EPDR0O or EPDR2 (when a read-only register) is read, the contents are stored in the FIFO, and when the data is valid it is read in the order in which it was transferred.

The EPDR registers are initialized to H'00 by a system reset or a function soft reset.

7.2.4 FIFO Valid Size Registers 0I, 0O, 1, 2 (FVSR0I, FVSR0O, FVSR1, FVSR2)

	FVSR0IH, FVSR0OH, FVSR1H, FVSR2H								FVSR0IL, FVSR0OL, FVSR1L, FVSR2L							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	N9	N8	N7	N6	N5	N4	N3	N2	N1	N0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0/1*	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * The initial value of bit N4 is 0 in FVSR0O, and 1 in the other FVSR registers.

The FVSR registers indicate the number of valid data bytes in the FIFO for each host input/host output involving the respective USB function endpoints. In host input transfer, the FVSR register indicates the number of bytes that the slave CPU can write to the FIFO (the FIFO size minus the number of bytes written to the FIFO by the slave CPU but not read (transmitted) by the USB function). In host output transfer, the FVSR register indicates the number of bytes received and written to the FIFO by the USB function but not read by the slave CPU.

In host input transfer, the FVSR value is decremented by the number of bytes written when the slave CPU writes to EPDR and sets the EPTE bit in PTTER, and is incremented by the number of bytes read when the USB function reads the FIFO and receives an ACK handshake from the host.

In host output transfer, the FVSR value is incremented by the number of bytes written when the USB function writes to the FIFO and transmits an ACK handshake, and is decremented by 1 each time the slave CPU reads EPDR.

If a transfer error occurs, data retransfer may be necessary. In this case, the FVSR value is not changed and the FIFO for the relevant channel is rewound.

In the USB protocol, for each endpoint DATA0 and DATA1 packets are transmitted and received alternately when data transfer is performed. This toggling between DATA0 and DATA1 also serves as an indicator of whether or not data transfer has been performed normally. If DATA0/DATA1 toggling is not performed normally in host output transfer, the USB function will abort processing of that transaction and the FVSR value will not change.

Since the FVSR registers are 2-byte registers and the H8's FIFOs are 16 or 32 bytes in length, the FIFO status can be indicated in the lower byte alone. Only the lower byte of the FVSR registers should be read.

The upper byte of the FVSR registers cannot be accessed directly. When the lower byte is read, the upper byte is transferred to a temporary register, and when the upper byte is read, the contents of this temporary register are read. When a word read is used on an FVSR register, the operation is

automatically divided into two byte accesses, with the upper byte read first, followed by the lower byte. Caution is required in this case, since the upper byte value that is read is the value at the point when the lower byte was read previously.

FVSR0I and FVSR1 are automatically initialized to H'0010 and H'0000, respectively, when a SETUP token is received.

The FVSR registers are initialized by a system reset or a function soft reset. The initial value depends on the transfer direction and FIFO size determined by EPDIR and EPSZR.

7.2.5 Endpoint Direction Register (EPDIR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2DIR	EP1DIR	—	—
Initial value	1	1	1	1	1	1	0	0
Read/Write	R	R	R	R	R/W	R/W	R	R

EPDIR controls the data transfer direction for USB function endpoints other than endpoint 0.

With the H8/3567U and H8/3564U, EP1 should be designated for host input transfer and EP2 for host input transfer or host output transfer.

EPDIR is initialized to H'FC by a system reset or a function soft reset.

Bit 3—Endpoint 2 Data Transfer Direction Control Flag (EP2DIR): Switches the endpoint 2 data transfer direction.

Bit 3

EP2DIR	Description	
0	Endpoint 2 is designated for host output transfer	
1	Endpoint 2 is designated for host input transfer	(Initial value)

Bit 2—Endpoint 1 Data Transfer Direction Control Flag (EP1DIR): Switches the endpoint 1 data transfer direction. This bit must not be cleared to 0.

Bit 2

EP1DIR	Description	
0	Setting prohibited	
1	Endpoint 1 is designated for host input transfer	(Initial value)

7.2.6 Packet Transmit Enable Register (PTTER)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2TE	EP1TE	EP0ITE	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R

Note: * Only 1 can be written.

PTTER contains control bits (EPTE) that control the FIFO valid size registers for USB function host input transfer.

In the USB protocol, communication is carried out using packets. The minimum unit of data transfer is a transaction, and a transaction is made up of a token packet, data packet, and handshake packet.

In host input transfer, the USB function receives an IN token (packet). If operation has not stalled, in response to this token the USB function must transmit a data packet or, if there is no data, a NAK handshake.

When EPTE is set to 1 after the data to be transmitted by the USB function has been written to the FIFO by the slave CPU, the FVSR contents are updated. This enables transmission of the data written to the FIFO. This EPTE-bit data transmission control prevents data transmission from being done while the slave CPU is writing data to the FIFO. The EPTE can only be written with 1, and are always read as 0.

Bit 3—Endpoint 2 Packet Transmit Enable (EP2TE): Updates endpoint 2 FVSR2 when the EP2DIR bit is set to 1.

Bit 3

EP2TE	Description	
0	Normal read value	(Initial value)
(1)	[1 write] Endpoint 2 IN-FIFO FVSR2 is updated	

Bit 2—Endpoint 1 Packet Transmit Enable (EP1TE): Updates endpoint 1 FVSR1.

Bit 2

EP1TE	Description	
0	Normal read value	(Initial value)
(1)	[1 write] Endpoint 1 IN-FIFO FVSR1 is updated	

Bit 1—Endpoint 0I Packet Transmit Enable (EP0ITE): Updates endpoint 0 FVSR0I.

Bit 1

EP0ITE	Description	
0	Normal read value	(Initial value)
(1)	[1 write] Endpoint 0 IN-FIFO FVSR0I is updated	

7.2.7 USB Interrupt Enable Register (USBIER)

Bit	7	6	5	4	3	2	1	0
	—	—	BRSTE	SOFE	SPNDE	TFE	TSE	SETUPE
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W

USBIER contains enable bits that enable interrupts from the USB function to the slave CPU.

USBIER is initialized to H'00 by a system reset or a function soft reset.

Bit 5—Bus Reset Interrupt Enable (BRSTE): Enables or disables bus request interrupts to the internal CPU.

Bit 5

BRSTE	Description	
0	USB function bus request interrupts disabled	(Initial value)
1	USB function bus request interrupts enabled	

Bit 4—SOF Interrupt Enable (SOFE): Enables or disables SOF (Start of Frame) interrupts to the internal CPU.

Bit 4

SOFE	Description	
0	USB function SOF interrupts disabled	(Initial value)
1	USB function SOF interrupts enabled	

Bit 3—Suspend Interrupt Enable (SPNDE): Enables or disables suspend OUT interrupts and suspend IN interrupts to the internal CPU.

Bit 3

SPNDE	Description	
0	USB function suspend OUT interrupts and suspend IN interrupts disable	(Initial value)
1	USB function suspend OUT interrupts and suspend IN interrupts enabled	

Bit 2—Transfer Failed Interrupt Enable (TFE): Enables or disables transfer failed interrupts to the internal CPU.

Bit 2

TFE	Description	
0	USB function transfer failed interrupts disabled	(Initial value)
1	USB function transfer failed interrupts enabled	

Bit 1—Transfer Successful Interrupt Enable (TSE): Enables or disables transfer successful interrupts to the internal CPU.

Bit 1

TSE	Description	
0	USB function transfer successful interrupts disabled	(Initial value)
1	USB function transfer successful interrupts enabled	

Bit 0—Setup Interrupt Enable (SETUPE): Enables or disables setup interrupts to the internal CPU.

Bit 0

SETUPE	Description	
0	USB function setup interrupts disabled	(Initial value)
1	USB function setup interrupts enabled	

7.2.8 USB Interrupt Flag Register (USBIFR)

Bit	7	6	5	4	3	2	1	0
	TS	TF	—	BRSTF	SOFF	SPNDOF	SPNDIF	SETUPF
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, after reading 1, to clear the flag.

USBIFR contains interrupt flags that generate interrupts from the USB function to the slave CPU.

The USB module has four interrupt sources (USBIA, USBIB, USBIC, and USBID). USBIA is a dedicated setup interrupt. A single transfer successful interrupt or transfer failed interrupt can be assigned to USBIB and USBIC. All other interrupts (all transfer successful interrupts and transfer failed interrupts, bus reset interrupts, SOF interrupts, and suspend OUT and suspend IN interrupts) are assigned to USBID.

USBIFR is initialized to H'00 by a system reset or a function soft reset.

Bit 7—Transfer Successful Interrupt Status (TS): Status flag that indicates that transfer has ended normally at a USB function endpoint.

When the TSE bit is 1, USBID interrupt request is sent to the slave CPU, but if a setting has been made for the source that set TS to 1 to request USBIB or USBIC interrupt, has priority for processing in accordance with the priority order in the slave CPU's interrupt controller (INTC).

TS is a read-only flag.

Bit 7

TS	Description	
0	All bits in transfer success flag register (TSFR) are 0	(Initial value)
1	At least one bit in transfer success flag register (TSFR) is 1	

Bit 6—Transfer Failed Interrupt Status (TF): Status flag that indicates that transfer has ended abnormally at a USB function endpoint.

When the TFE bit is 1, USBID interrupt request is sent to the slave CPU, but if a setting has been made for the source that set TF to 1 to request USBIB or USBIC interrupt, has priority for processing in accordance with the priority order in the slave CPU's interrupt controller (INTC).

TF is a read-only flag.

Bit 6

TF	Description	
0	All bits in transfer fail flag register (TFFR) are 0	(Initial value)
1	At least one bit in transfer fail flag register (TFFR) is 1	

Bit 4—Bus Reset Interrupt Flag (BRSTF): Status flag that indicates that the USB function has detected a bus reset from upstream.

When the BRSTE bit is 1, USBID interrupt request is sent to the slave CPU.

Bit 4

BRSTF	Description	
0	[Clearing condition]	(Initial value)
	When 0 is written to BRSTF after reading BRSTF = 1	
1	[Setting condition]	
	When USB function detects a bus reset from upstream	

Bit 3—SOF Interrupt Flag (SOFF): Status flag that indicates that the USB function has detected SOF (Start of Frame).

When the SOFE bit is 1, USBID interrupt request is sent to the slave CPU.

Bit 3

SOFF	Description	
0	[Clearing condition]	(Initial value)
	When 0 is written to SOFF after reading SOFF = 1	
1	[Setting condition]	
	When USB function detects SOF (Start of Frame)	

Bit 2—Suspend OUT Interrupt Flag (SPNDOF): Status flag that indicates that the USB function has detected a change in the bus status, and has switched from the suspend state to the normal state.

When the SPNDE bit is 1, USBID interrupt request is sent to the slave CPU.

Bit 2

SPNDOF	Description	
0	[Clearing condition]	(Initial value)
	When 0 is written to SPNDOF after reading SPNDOF = 1	
1	[Setting condition]	
	When USB function switches from suspend state to normal state	

Bit 1—Suspend IN Interrupt Flag (SPNDIF): Status flag that indicates that the USB function has detected a bus idle state lasting longer that the specified time, and has switched from the normal state to the suspend state.

When the SPNDE bit is 1, USBID interrupt request is sent to the slave CPU.

Bit 1

SPNDIF	Description
0	[Clearing condition] (Initial value) When 0 is written to SPNDIF after reading SPNDIF = 1
1	[Setting condition] When USB function switches from normal state to suspend state

Bit 0—Setup Interrupt Flag (SETUPF): Status flag that indicates that USB function endpoint 0 has received a SETUP token.

When the SETUPE bit is 1, USBIA interrupt request is sent to the slave CPU.

Bit 0

SETUPF	Description
0	[Clearing condition] (Initial value) When 0 is written to SETUPF after reading SETUPF = 1
1	[Setting condition] When USB function endpoint 0 receives SETUP token

7.2.9 Transfer Success Flag Register (TSFR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2TS	EP1TS	EP0ITS	EP0OTS
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, after reading 1, to clear the flag.

TSFR contains status flags (EPTS flags) that indicate that a USB function endpoint host input/host output transaction has ended normally. The condition for a normal end of a transaction is reception of an ACK handshake in host input transfer, or transmission of an ACK handshake in host output transfer.

When at least one EPTS flag is set to 1, the TS flag in USBIFR is also set at the same time. The TS flag generates an interrupt to the slave CPU. The EPTS flags must be cleared to 0 in the interrupt handling routine. When all the EPTS flags are cleared, the TS flag is automatically cleared to 0. Only 0 can be written to the EPTS flags, after first reading 1.

When the USB function receives a SETUP token, the EP0ITS and EP0OTS flags are automatically cleared to 0.

TSFR is initialized to H'00 by a system reset or a function soft reset.

Bit 3—Endpoint 2 Transfer Success Flag (EP2TS): Indicates that an endpoint 2 host input transfer or host output transfer has ended normally.

Bit 3

EP2TS	Description
0	Endpoint 2 is in transfer standby state (Initial value) [Clearing condition] When 0 is written to EP2TS after reading EP2TS = 1
1	Endpoint 2 host input transfer (IN transaction) or host output transfer (OUT transaction) has ended normally [Setting conditions] • ACK handshake established after IN token reception and data transfer (ACK reception) • ACK handshake established after OUT token reception and data transfer (ACK transmission)

Bit 2—Endpoint 1 Transfer Success Flag (EP1TS): Indicates that an endpoint 1 host input transfer has ended normally.

Bit 2

EP1TS	Description
0	Endpoint 1 is in transfer standby state (Initial value) [Clearing condition] When 0 is written to EP1TS after reading EP1TS = 1
1	Endpoint 1 host input transfer (IN transaction) has ended normally [Setting condition] ACK handshake established after IN token reception and data transfer (ACK reception)

Bit 1—Endpoint 0 Host Input Transfer Success Flag (EP0ITS): Indicates that an endpoint 0 host input transfer has ended normally.

Bit 1

EP0ITS	Description
0	Endpoint 0 is in host input transfer standby state (Initial value) [Clearing conditions] • When 0 is written to EP0ITS after reading EP0ITS = 1 • When endpoint 0 receives a SETUP token
1	Endpoint 0 host input transfer (IN transaction) has ended normally [Setting condition] ACK handshake established after IN token reception and data transfer (ACK reception)

Bit 0—Endpoint 0 Host Output Transfer Success Flag (EP0OTS): Indicates that an endpoint 0 host output transfer has ended normally.

Host output transfers to endpoint 0 include OUT transactions and SETUP transactions. These operations are the same in terms of data transfer, but differ as regards flag handling.

Most commands transferred in SETUP transactions are processed within the USB function, in which case the EP0OTS flag is not set and the EP0OTF flag is.

In the case of a command that cannot be processed within the USB function, the EP0OTS flag is set.

Bit 0

EP0OTS	Description
0	Endpoint 0 is in host output transfer standby state (Initial value) [Clearing conditions] • When 0 is written to EP0OTS after reading EP0OTS = 1 • When endpoint 0 receives a SETUP token
1	Endpoint 0 host output transfer (OUT transaction or SETUP transaction) has ended normally [Setting conditions] • ACK handshake established after OUT token reception and data transfer (ACK transmission) • When command received after SETUP token reception requires processing by the slave CPU

7.2.10 Transfer Fail Flag Register (TFFR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2TF	EP1TF	EP0ITF	EP0OTF
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, after reading 1, to clear the flag.

TFFR contains status flags (EPTF flags) that indicate that a USB function endpoint host input/host output transaction has not ended normally. The condition for an abnormal end of a transaction is NAK handshake reception, or NAK handshake transmission when there is no transfer data (FVSR = FIFO size (FIFO empty)), in host input transfer, or, in host output transfer, NAK handshake transmission due to a FIFO full condition, etc., or any of various communication errors (DATA0/DATA1 toggle error, bit stuffing error, bit count error, CRC error, transfer of a number of bytes exceeding MaxPktSz, etc.) during data transfer.

When at least one EPTF flag is set to 1, the TF flag in USBIFR is also set at the same time. The TF flag generates an interrupt to the slave CPU. The EPTF flags must be cleared to 0 in the interrupt handling routine. When all the EPTF flags are cleared, the TF flag is automatically cleared to 0. Only 0 can be written to the EPTF flags, after first reading 1.

When the USB function receives a SETUP token, the EP0ITF and EP0OTF flags are automatically cleared to 0.

TFFR is initialized to H'00 by a system reset or a function soft reset.

Bit 3—Endpoint 2 Transfer Fail Flag (EP2TF): Indicates that an endpoint 2 host input transfer or host output transfer has not ended normally.

Bit 3

EP2TF	Description
0	Endpoint 2 is in transfer standby state (Initial value) [Clearing condition] When 0 is written to EP2TF after reading EP2TF = 1
1	Endpoint 2 host input transfer (IN transaction) or host output transfer (OUT transaction) has ended abnormally [Setting conditions] • ACK handshake not established after IN token reception and data transfer • Data transfer not possible due to FIFO empty condition after IN token reception • Data transfer not possible due to FIFO full condition after OUT token reception (NAK transmission) • Data transfer errors after OUT token reception

Bit 2—Endpoint 1 Transfer Fail Flag (EP1TF): Indicates that an endpoint 1 host input transfer has not ended normally.

Bit 2

EP1TF	Description
0	Endpoint 1 is in transfer standby state (Initial value) [Clearing condition] When 0 is written to EP1TF after reading EP1TF = 1
1	Endpoint 1 host input transfer (IN transaction) has ended abnormally [Setting conditions] • ACK handshake not established after IN token reception and data transfer • Data transfer not possible due to FIFO empty condition after IN token reception (NAK transmission)

Bit 1—Endpoint 0 Host Input Transfer Fail Flag (EP0ITF): Indicates that an endpoint 0 host input transfer has not ended normally.

Bit 1

EP0ITF	Description
0	Endpoint 0 is in host input transfer standby state (Initial value) [Clearing conditions] • When 0 is written to EP0ITF after reading EP0ITF = 1 • When endpoint 0 receives a SETUP token
1	Endpoint 0 host input transfer (IN transaction) has ended abnormally [Setting conditions] • ACK handshake not established after IN token reception and data transfer • Data transfer not possible due to FIFO empty condition after IN token reception (NAK transmission)

Bit 0—Endpoint 0 Host Output Transfer Fail Flag (EP0OTF): Indicates that an endpoint 0 host output transfer has not ended normally.

Host output transfers to endpoint 0 include OUT transactions and SETUP transactions. These operations are the same in terms of data transfer, but differ as regards flag handling.

Most commands transferred in SETUP transactions are processed within the USB function, in which case the EP0OTS flag is not set and the EP0OTF flag is.

In the case of a command that cannot be processed within the USB function, the EP0OTS flag is set.

Bit 0

EP0OTF	Description
0	Endpoint 0 is in host output transfer standby state (Initial value) [Clearing conditions] • When 0 is written to EP0OTF after reading EP0OTF = 1 • When endpoint 0 receives a SETUP token
1	Endpoint 0 host output transfer (OUT transaction or SETUP transaction) has ended abnormally [Setting conditions] • Transfer not possible due to FIFO full condition after OUT token reception (NAK transmission) • Data transfer not possible because EP0OTC = 0 after OUT token reception (NAK transmission) • Communication error after OUT token reception • When command received after SETUP token reception can be processed within the USB function

7.2.11 USB Control/Status Register 0 (USBCSR0)

Bit	7	6	5	4	3	2	1	0
	DP5CNCT	DP4CNCT	DP3CNCT	DP2CNCT	EP0STOP	EPIVLD	EP0OTC	CKSTOP
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W

USBCSR0 contains flags that indicate the USB hubs' downstream port connection status, and bits that control the operation of the USB function.

USBCSR0 is initialized to H'00 by a system reset, and bits 3 to 0 are also cleared to 0 by a function soft reset.

Bits 7 to 4—Downstream Port Connect 5 to 2 (DP5CNCT, DP4CNCT, DP3CNCT, DP2CNCT): Read-only status flags that indicate the connection status of the USB hubs' external downstream ports.

Bits 7 to 4

DP5CNCT to DP2CNCT	Description	
0	Cable is not connected to downstream port [Clearing conditions] • System reset • Downstream port disconnect • USB hub upstream port disconnect (Total downstream disconnect by software in reconnect process)	(Initial value)
1	Cable is connected to downstream port, and power is being supplied [Setting condition] Downstream port connect	

Bit 3—Endpoint 0 Stop (EP0STOP): Bit that protects the contents of the USB function endpoint 0 FIFO. Setting EP0STOP to 1 enables the data transferred to the EP0 OUT-FIFO by a SETUP transaction to be protected.

Bit 3

EP0STOP	Description	
0	EP0 OUT-FIFO, IN-FIFO operational [Clearing conditions] • System reset • Function soft reset	(Initial value)
1	EP0 OUT-FIFO reading stopped • FVSR0O contents are not changed by an EPDR0O read EP0 IN-FIFO writing and transfer stopped • FIFO contents are not changed by an EPDR0I write • FVSR0I contents are not changed by setting EP0ITE	

Bit 2—Endpoint Information Valid (EPIVLD): This bit makes the USB function block operational.

Part of the process that makes the USB function block operational includes an endpoint information setting. After a system reset or function soft reset, the USB function block does not have any endpoint information. Endpoint information for the USB function in the H8/3567U and H8/3564U (see section 7.3.9, USB Module Startup Sequence) can be set by sequential writes to EPDR0I. When all the data has been written, the written endpoint information is made valid by setting the EPIVLD bit to 1. Writing 0 to the EPIVLD bit has no effect.

Bit 2

EPIVLD	Description	
0	Endpoint information (EPINFO) has not been set [Clearing conditions] • System reset • Function soft reset	(Initial value)
1	Endpoint information (EPINFO) has been set	

Bit 1—Endpoint 0O Transfer Control (EP0OTC): Controls USB function endpoint 0 control transfer. Clearing EP0OTC to 0 disables writes to the EP0 OUT-FIFO. A change of data transfer direction within a control transfer can be reported by means of the transfer fail interrupt caused by this action. In control transfer, a command is received in the SETUP transaction (command stage), then data transfer is performed in an OUT or IN transaction (data stage), and finally a transfer equivalent to a handshake is carried out in an IN or OUT transaction (status stage).

When a SETUP token is received, EP0OTC is set to 1, FVSR is initialized, and command data can be received. On completion of command data reception, EP0OTC is cleared to 0 and the contents of the EP0O-FIFO are protected. If the command cannot be processed automatically by the USB function core, the EP0OTS flag is set and the slave CPU must decode the command.

If command decoding shows that an OUT transaction will follow as the data stage, the slave CPU must set EP0OTC to 1 in preparation for an OUT transaction. If the command stage is followed by an IN transaction data stage, the slave CPU leaves EP0OTC cleared to 0. When the host CPU begins an OUT transaction as the status stage, the EP0OTF flag is set and a transfer fail interrupt is generated, enabling the slave CPU to recognize the end of the data stage. In response to this interrupt, the slave CPU sets EP0OTC to 1 and receives retransferred status stage data.

Bit 1

EP0OTC	Description
0	EP0 OUT-FIFO writing stopped (Initial value) • Subsequent writes to EP0 OUT-FIFO are invalid [Clearing conditions] • System reset • Function soft reset • Command data reception in SETUP transaction (EP0OTS flag setting)
1	EP0 OUT-FIFO operational [Setting conditions] • SETUP token reception • When 1 is written to EP0OTC after reading EP0OTC = 0

Bit 0—Clock Stop (CKSTOP): Controls the USB function operating clock. When the USB function is placed in the suspend state due to a bus idle condition, this bit should be set to 1 after the necessary processing is completed. The clock supply to the USB function is then stopped, reducing power consumption.

When the CKSTOP bit is set to 1, writes to USB module registers are invalid. If these registers are read, the contents of the read data are not guaranteed, but there are no read-related status changes (such as decrementing of FVSR).

If a bus idle condition of the specified duration or longer is detected, the suspend IN interrupt flag is set, and when a change in the bus status is subsequently detected the suspend OUT interrupt flag is set. When the suspend OUT interrupt flag is set, the CKSTOP bit is simultaneously cleared to 0.

Bit 0

CKSTOP	Description
0	Clock is supplied to USB function (Initial value) [Clearing conditions] • System reset • Function soft reset • Suspend OUT interrupt flag setting
1	Clock supply to USB function is stopped [Setting condition] When 1 is written to CKSTOP after reading CKSTOP = 0 in the function suspend state

7.2.12 Endpoint Stall Register (EPSTLR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2STL	EP1STL	—	EP0STL
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/W	R/W	R	R/W

EPSTLR contains bits (EPSTL) that place the USB function endpoints in the stall state.

When an EPSTL bit is set to 1, the corresponding endpoint sends a STALL handshake in reply to the start of a transaction through reception of a token from the host.

When the USB function receives a SETUP token, the EP0STL bit is automatically cleared to 0.

EPSTLR is initialized to H'00 by a system reset or a function soft reset.

Bit 3—Endpoint 2 Stall (EP2STL): Places endpoint 2 in the stall state.

Bit 3

EP2STL	Description
0	Endpoint 2 is operational (Initial value)
1	Endpoint 2 is in stall state

Bit 2—Endpoint 1 Stall (EP1STL): Places endpoint 1 in the stall state.

Bit 2	
EP1STL	**Description**
0	Endpoint 1 is operational (Initial value)
1	Endpoint 1 is in stall state

Bit 0—Endpoint 0 Stall (EP0STL): Places endpoint 0 in the stall state. Writing 0 to the EP0STL bit has no effect.

Bit 0	
EP0STL	**Description**
0	Endpoint 0 is operational (Initial value) [Clearing condition] When endpoint 0 receives a SETUP token
1	Endpoint 0 is in stall state [Setting condition] When 1 is written to EP0STL after reading EP0STL = 0

7.2.13 Endpoint Reset Register (EPRSTR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2RST	EP1RST	EP0IRST	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R

Note: * Only 1 can be written.

EPRSTR contains control bits (EPRST) that reset the pointer of the FIFO for a USB function endpoint host input transfer.

When an EPRST bit is set to 1, the corresponding FIFO valid size register (FVSR) is initialized.

The EPRST bits can only be written with 1, and are always read as 0.

Bit 3—Endpoint 2 Reset (EP2RST): Initializes the endpoint 2 FIFO.

Bit 3

EP2RST	Description	
0	Normal read value	(Initial value)
(1)	[1 write]	
	EP2DIR = 0: FVSR2 is initialized to H'0000	
	EP2DIR = 1: FVSR2 is initialized to H'0010	

Bit 2—Endpoint 1 Reset (EP1RST): Initializes the endpoint 1 FIFO.

Bit 2

EP1RST	Description	
0	Normal read value	(Initial value)
(1)	[1 write]	
	EP1 FIFO size = 16 bytes: FVSR1 is initialized to H'0010	
	EP1 FIFO size = 32 bytes: FVSR1 is initialized to H'0020	

Bit 1—Endpoint 0I Reset (EP0IRST): Initializes the endpoint 0I FIFO.

Bit 1

EP0IRST	Description	
0	Normal read value	(Initial value)
(1)	[1 write]	
	FVSR0I is initialized to H'0010	

7.2.14 Device Resume Register (DEVRSMR)

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

Note: * Only 1 can be written.

DEVRSMR contains a bit (DVR) that control remote wakeup of the USB function suspend state. When 1 is written to the DVR bit, the suspend state is cleared.

The DVR bit can only be written with 1, and is always read as 0.

1 can be written to the DVR bit even if the CKSTOP bit is set to 1 in USBCSR0.

Bit 0—Device Resume (DVR): Clears the suspend state.

Bit 0

DVR	Description	
0	Normal read value	(Initial value)
(1)	[1 write] Suspend state is cleared (remote wakeup)	

7.2.15 Interrupt Source Select Register 0 (INTSELR0)

Bit	7	6	5	4	3	2	1	0
	TSELB	EPIBS2	EPIBS1	EPIBS0	TSELC	EPICS2	EPICS1	EPICS0
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

INTSELR0 contains bits that select the USB function USBIB and USBIC interrupt sources.

INTSELR0 is initialized to H'00 by a system reset or a function soft reset.

Bit 7—Transfer Select B (TSELB): Together with bits EPIBS2 to EPIBS0, selects USBIB interrupt source.

Bit 7

TSELB	Description
0	USBIB interrupt is requested by a TS interrupt; the endpoint constituting the TS interrupt source is specified by bits EPIBS2 to EPIBS0 (Initial value)
1	USBIB interrupt is requested by a TF interrupt; the endpoint constituting the TF interrupt source is specified by bits EPIBS2 to EPIBS0

Bits 6 to 4—Interrupt B Endpoint Select 2 to 0 (EPIBS2 to EPIBS0): Together with the TSELB bit, these bits select USBIB interrupt source.

Bit 6	Bit 5	Bit 4		
EPIBS2	EPIBS1	EPIBS0	Description	
0	0	0	Endpoint not selected	(Initial value)
		1	Endpoint 1 selected	
	1	0	Endpoint 2 selected	
		1	Setting prohibited	
1	—	—	Setting prohibited	

Bit 3—Transfer Select C (TSELC): Together with bits EPICS2 to EPICS0, selects USBIC interrupt source.

Bit 3

TSELC	Description
0	USBIC interrupt is requested by a TS interrupt; the endpoint constituting the TS interrupt source is specified by bits EPICS2 to EPICS0 (Initial value)
1	USBIC interrupt is requested by a TF interrupt; the endpoint constituting the TF interrupt source is specified by bits EPICS2 to EPICS0

Bits 2 to 0—Interrupt C Endpoint Select 2 to 0 (EPICS2 to EPICS0): Together with the TSELC bit, these bits select USBIC interrupt source.

Bit 2	Bit 1	Bit 0		
EPICS2	**EPICS1**	**EPICS0**	**Description**	
0	0	0	Endpoint not selected	(Initial value)
		1	Endpoint 1 selected	
	1	0	Endpoint 2 selected	
		1	Setting prohibited	
1	—	—	Setting prohibited	

7.2.16 Interrupt Source Select Register 1 (INTSELR1)

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

Register INTSELR1 is not used in this model.

Do not write 1 to the bits in INTSELR1.

7.2.17 Hub Overcurrent Control Register (HOCCR)

Bit	7	6	5	4	3	2	1	0
	—	—	PCSP	OCDSP	HOC5E	HOC4E	HOC3E	HOC2E
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W

The USB hub downstream ports are connected to the USB connector as data (D+/D–). The power supply (VBUS) connected to the USB connector is generated by connecting a power supply control IC externally.

HOCCR contains bits that control the power supply control IC control input/output.

HOCCR is initialized to H'00 by a system reset.

Bit 5—Power Supply Enable Control Polarity (PCSP): This bit is set according to the polarity of the power supply control IC output enable inputs.

The power supply control IC output enable inputs are connected to H8 pins $\overline{ENP}_5$ to $\overline{ENP}_2$.

Bit 5

PCSP	Description	
0	Power supply control IC requires low-level input for enabling	(Initial value)
1	Power supply control IC requires high-level input for enabling	

Bit 4—Overcurrent Detection Polarity (OCDSP): This bit is set according to the polarity of the power supply control IC overcurrent detection outputs.

The power supply control IC overcurrent detection outputs are connected to H8 pins $\overline{OCP}_5$ to $\overline{OCP}_2$.

Bit 4

OCDSP	Description	
0	Power supply control IC outputs low level in case of overcurrent detection	(Initial value)
1	Power supply control IC outputs high level in case of overcurrent detection	

Bits 3 to 0—Overcurrent Detection Control Enable 5 to 2 (HOC5E to HOC2E): These pins select whether or not power supply control IC control is performed for each USB hub downstream port.

If any of the four downstream ports are not used, the corresponding D+/D- pins should be pulled down as specified. Leave the corresponding HOCE bit cleared to 0, disabling the corresponding output enable pin and overcurrent detection pin. Disabled pins can be used as general port pins (port C).

Bit 3

HOC5E	Description	
0	Pins $\overline{ENP}_5$ and $\overline{OCP}_5$ are general ports (PC_7, PC_3)	(Initial value)
1	Pins $\overline{ENP}_5$ and $\overline{OCP}_5$ have output enable and overcurrent detection functions	

Bit 2

HOC4E	Description	
0	Pins $\overline{ENP}_4$ and $\overline{OCP}_4$ are general ports (PC_6, PC_2)	(Initial value)
1	Pins $\overline{ENP}_4$ and $\overline{OCP}_4$ have output enable and overcurrent detection functions	

Bit 1

HOC3E	Description	
0	Pins $\overline{ENP}_3$ and $\overline{OCP}_3$ are general ports (PC_5, PC_1)	(Initial value)
1	Pins $\overline{ENP}_3$ and $\overline{OCP}_3$ have output enable and overcurrent detection functions	

Bit 0

HOC2E	Description	
0	Pins $\overline{ENP}_2$ and $\overline{OCP}_2$ are general ports (PC_4, PC_0)	(Initial value)
1	Pins $\overline{ENP}_2$ and $\overline{OCP}_2$ have output enable and overcurrent detection functions	

7.2.18 USB Control Register (USBCR)

Bit	7	6	5	4	3	2	1	0
	FADSEL	FONLY	FNCSTP	UIFRST	HPLLRST	HSRST	FPLLRST	FSRST
Initial value	0	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

USBCR contains bits (FADSEL, FONLY, FNCSTP) that control USB function and USB hub internal connection, and reset control bits for sequential enabling of the operation of each part according to the USB module start-up sequence.

USBCR is initialized to H'7F by a system reset [in an H8/3567U and H8/3564U reset (by $\overline{RES}$ input or the watchdog timer), and in hardware standby mode]. It is not initialized in software standby mode.

Bit 7—USB Function I/O Analog/Digital Select (FADSEL): Selects the USB function data input/output method when the FONLY bit is set to 1 so that the USB hub block is disabled and only the USB function block operates.

Bit 7

FADSEL	Description
0	USD+ and USD– pins are used for USB function block data input/output (Initial value)
1	USB function block data input/output is implemented by multiplexing Philips transceiver/receiver (PDIUSB11A) compatible control input/output with port C pins

Port C	Philips PDIUSB11A		
PC_7	Input	VP	Differential input (+)
PC_6	Input	VM	Differential input (–)
PC_5	Input	RCV	Data input
PC_4	Output	VPO	Differential output (+)
PC_3	Output	VMO	Differential output (–)
PC_2	Output	$\overline{OE}$	Output enable
PC_1	Output	SUSPEND	Suspend setting
PC_0	Output	SPEED	Speed setting High level fixed output for 12 Mbps specification

Bit 6—USB Function Select (FONLY): Selects enabling/disabling of the USB hub block. When the USB hub block is enabled, the USB function block is connected internally to USB hub downstream port 1. When the USB hub block is disabled, the USB function block is directly connected to the upstream port, and the USB operating clock selected/divided/multiplied in accordance with UPLLCR settings is not supplied to the USB hub block.

Bit 6

FONLY	Description
0	USB function block is connected internally to USB hub downstream port 1 USB hub block is enabled
1	USB function block is directly connected to upstream port USB hub block is disabled (Initial value)

Bit 5—USB Function Stop/Suspend (FNCSTP): With the H8/3567U and H8/3564U, it is possible to disconnect the USB function block from the USB hub block's downstream port 1, and set a power-down state in which the USB operating clock supply is halted. Register accesses by the CPU are still possible in this state.

The FNCSTP bit is used when disconnecting the USB function block and switching the microcomputer block to power-down mode when the system's power supply is cut, or when reconnecting the USB function block when recovering from power-down mode or in the event of a power-on reset.

When the FNCSTP bit is set to 1, the USB operating clock selected/divided/multiplied in accordance with UPLLCR settings is not supplied to the USB function block.

Bit 5

FNCSTP	Description
0	For USB function block, USB hub downstream port 1 internal connection is set to connected state
1	For USB function block, USB hub downstream port 1 internal connection is set to disconnected state, and power-down state is set (Initial value)

Bit 4—USB Interface Soft reset (UIFRST): Resets the EPSZR1, USBIER, EPDIR, INTSELR0, and INTSELR1 registers. When UIFRST is set to 1, the EPSZR1, USBIER, EPDIR, INTSELR0, and INTSELR1 registers are initialized.

Bit 4

UIFRST	Description
0	EPSZR1, USBIER, EPDIR, INTSELR0, and INTSELR1 are placed in operational state
1	EPSZR1, USBIER, EPDIR, INTSELR0, and INTSELR1 are placed in reset state (Initial value)

Bit 3—Hub Block PLL Soft reset (HPLLRST): Resets the USB bus clock circuit (DPLL) in the hub.

When HPLLRST is set to 1, the DPLL circuit in the hub is reset, and bus clock synchronous operation halts. HPLLRST is cleared to 0 after PLL operation stabilizes.

Bit 3

HPLLRST	Description	
0	Hub DPLL is placed in operational state	
1	Hub DPLL is placed in reset state	(Initial value)

Bit 2—Hub Block Internal State Soft reset (HSRST): Resets the internal state of the USB hub block.

When HSRST is set to 1, the internal state of the USB hub block, excluding the internal USB bus clock circuit (DPLL), is initialized. HSRST is cleared to 0 after DPLL operation stabilizes.

Bit 2

HSRST	Description	
0	Internal state of USB hub block is set to operational state	
1	Internal state of USB hub block is set to reset state (excluding DPLL)	(Initial value)

Bit 1—Function Block PLL Soft reset (FPLLRST): Resets the USB bus clock circuit (DPLL) in the function.

When FPLLRST is set to 1, the DPLL circuit in the function is reset, and bus clock synchronous operation halts. FPLLRST is cleared to 0 after PLL operation stabilizes.

Bit 1

FPLLRST	Description	
0	Function DPLL is placed in operational state	
1	Function DPLL is placed in reset state	(Initial value)

Bit 0—Function Block Internal State Reset (FSRST): Resets the internal state of the USB function block.

When FSRST is set to 1, the internal state of the USB function block, excluding the internal bus clock circuit (DPLL), is initialized. FSRST is cleared to 0 after DPLL operation stabilizes.

The state in which FSRST = 1 and UIFRST = 1 is called a function soft reset.

Bit 0

FSRST	Description
0	Internal state of USB function block is set to operational state
1	Internal state of USB function block is set to reset state (excluding DPLL) (Initial value)

7.2.19 USB PLL Control Register (UPLLCR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	CKSEL2	CKSEL1	CKSEL0	PFSEL1	PFSEL0
Initial value	0	0	0	0	0	0	0	1
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W

UPLLCR contains bits that control the method of generating the USB function and USB hub operating clock.

UPLLCR is initialized to H'01 by a system reset [in an H8/3567U and H8/3564U reset (by $\overline{\text{RES}}$ input or the watchdog timer), and in hardware standby mode]. It is not initialized in software standby mode.

Bits 4 to 2—Clock Source Select 2 to 0 (CKSEL2 to CKSEL0): These bits select the source of the clock supplied to the USB operating clock generator (PLL).

CKSEL0 selects either the USB clock pulse generator (XTAL12) or the system clock pulse generator (XTATL) as as the clock source. When selected as a clock source, the USB clock pulse generator starts operating. It operates with CKSEL2=1, CKSEL0=1.

When CKSEL2 = 1 and CKSEL1 = 1, the PLL operates.

When CKSEL1 is cleared to 0, a clock is not input to the PLL, and PLL operation halts. The 48 MHz signal from the USB clock pulse generator can be input directly as the USB operating clock.

When CKSEL2 is cleared to 0, a clock is not input to the PLL, and PLL operation halts.

Bit 4	Bit 3	Bit 2		
CKSEL2	**CKSEL1**	**CKSEL0**	**Description**	
0	0	0	PLL operation halted, clock input halted	(Initial value)
	—	—	PLL operation halted, clock input halted	
1	0	0	Setting prohibited	
		1	PLL operation halted USB clock pulse generator (XTAL12: 48 MHz) used directly instead of PLL output	
	1	0	PLL operates with system clock pulse generator (XTAL) as clock source	
		1	PLL operates with USB clock pulse generator (XTAL12) as clock source	

Bits 1 and 0—PLL Frequency Select 1 and 0 (PFSEL1, PFSEL0): These bits select the frequency of the clock supplied to the USB operating clock pulse generator (PLL).

The PLL generates the 48 MHz USB operating clock using the frequency selected with these bits as the clock source frequency.

Bit 1	Bit 0		
PFSEL1	**PFSEL0**	**Description**	
0	0	PLL input clock is 8 MHz	
	1	PLL input clock is 12 MHz	(Initial value)
1	0	PLL input clock is 16 MHz	
	1	PLL input clock is 20 MHz	

7.2.20 USB Port Control Register (UPRTCR)

Bit	7	6	5	4	3	2	1	0
	—	—	DSPSEL2	DSPSEL1	DSPSEL0	PCNMD2	PCNMD1	PCNMD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W

UPRTCR is a test register. Its initial settings should not be changed.

UPRTCR is initialized to H'00 by a system reset (reset of this LSI by a $\overline{\text{RES}}$ input or by the watchdog timer, and in hardware standby mode). It is not initialized in software standby mode.

Bits 5 to 3—Downstream Port Select 2 to 0 (DSPSEL2 to DSPSEL0): These bits select the downstream port to be tested.

Bit 5	Bit 4	Bit 3		
DSPSEL2	**DSPSEL1**	**DSPSEL0**	**Description**	
0	0	0	Downstream port 2 selected	(Initial value)
		1	Downstream port 3 selected	
	1	0	Downstream port 4 selected	
		1	Downstream port 5 selected	
1	—	—	Downstream port 1 selected	

Bits 2 to 0—Port Connection Mode Select 2 to 0 (PCNMD2 to PCNMD0): These bits set ports C and D to the normal operating mode or a test operating mode. The PCNMD bits must be set to B'000.

Bit 2	Bit 1	Bit 0		
PCNMD2	**PCNMD1**	**PCNMD0**	**Description**	
0	0	0	User mode	(Initial value)
		1	Digital upstream mode	
	1	0	Digital downstream mode	
		1	Digital upstream/downstream mode	
1	0	0	Upstream transceiver/receiver monitor mode	
		1	Downstream transceiver/receiver monitor mode	
	1	—	Reserved	

7.2.21 USB Test Registers 2, 1, 0 (UTESTR2, UTESTR1, UTESTR0)

UTESTR2, UTESTR1, and UTESTR0 are test registers. Their initial settings should not be changed.

UTESTR1 and UTESTR0 are initialized to H'00 by a system reset [in an H8/3567U or H8/3564U reset (by $\overline{\text{RES}}$ input or the watchdog timer), and in standby mode]. They are not initialized in software standby mode.

UTESTR2 is initialized to H'FF by a system reset [in an H8/3567U or H8/3564U reset (by $\overline{\text{RES}}$ input or the watchdog timer), and in standby mode]. It is not initialized in software standby mode.

UTESTR0

Bit	7	6	5	4	3	2	1	0
	TEST15	TEST14	TEST13	TEST12	TEST11	TEST10	TEST9	TEST8
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

UTESTR1

Bit	7	6	5	4	3	2	1	0
	TEST7	TEST6	TEST5	TEST4	TEST3	TEST2	TEST1	TEST0
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

UTESTR2

Bit	7	6	5	4	3	2	1	0
	TESTA	TESTB	TESTC	TESTD	TESTE	TESTF	TESTG	TESTH
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

7.2.22 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers that perform module stop mode control.

When the MSTP1 bit is set to 1, the USB module stops operating and enters module stop mode at the end of the bus cycle. However, when USB clocks (XTAL12, EXTAL12) are selected as USB operating clocks, the USB module does not stop operating. For details, see section 21.5, Module Stop Mode.

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

MSTPCRL Bit 1—Module Stop (MSTP1): Specifies module stop mode for the USB module.

MSTPCRL Bit 1	
MSTP1	**Description**
0	USB module stop mode cleared
1	USB module stop mode set (Initial value)

7.2.23 Serial Timer Control Register (STCR)

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

STCR is an 8-bit readable/writable register that controls register access, the IIC operating mode, selects the TCNT input clock and controls USB. For details of functions other than register access control, see the descriptions of the relevant modules. If a module controlled by STCR is not used, do not write 1 to the corresponding bit.

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

Bit 7—Reserved: Do not write 1 to this bit.

Bits 6 and 5—I²C Control (IICX1, IICX0): These bits control the operation of the I²C bus interface. For details, see section 16, I²C Bus Interface.

Bit 4—I²C Master Enable (IICE): Controls CPU access to the I²C bus interface data registers and control registers (ICCR, ICSR, ICDR/SARX, and ICMR/SAR), the PWMX data registers and control registers (DADRAH/DACR, DADRAL, DADRBH/DACNTH, and DADRBL/DACNTL), and the SCI control registers (SMR, BRR, and SCMR).

Bit 4

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

Bit 3—Reserved: Do not write 1 to this bit.

Bit 2—USB enable (USBE): This bit controls CPU access to the USB data register and control register.

Bit 2

USBE	Description
0	Prohibition of the above register access (Initial value)
1	Permission of the above register access

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

7.3 Operation

USB is an interface for peripherals of the personal computers standardized by Intel and others and the standard is defined by the USB Specification. Operation of the USB hubs and USB function in this model is based on the definitions of the USB Specification. This section gives only a brief overview of the USB bus specifications, and focuses on operations by the slave CPU.

7.3.1 USB Compound Device Configuration

A USB compound device is a USB device incorporating USB hubs and a USB function. The H8/3567U and H8/3564U incorporate a compound device with a configuration in which the USB function is internally connected to one downstream port of a USB hub with five downstream ports.

With a USB compound device, it is usual for the USB function to be constantly connected to the USB hub. With the H8/3567U and H8/3564U, however, the internally connected USB function is not constantly connected to the USB hub. After release from an H8 reset, the USB function can be connected or disconnected under program control. Therefore, the device is not identified as a compound device in the hub descriptor wHub Characteristics.

There are two power feed modes for a USB device: bus feed and self-feed. The H8/3567 Group use the self-feed method.

With the H8/3567U and H8/3564U a setting can be made to disconnect the USB function block and operate the USB hub block alone. In this case, it is possible to place the slave CPU in software standby mode, and operate it in power-down mode.

7.3.2 Functions of USB Hub Block

The USB hub block implements the functions described in section 11 of the USB Specification.

There are five downstream ports; downstream port 1 can be connected to the USB function block internally, while downstream ports 2 to 5 are connected to external pins. Downstream ports 2 to 5 have their respective overcurrent detection pins ($\overline{OCP}_2$ to $\overline{OCP}_5$) and power supply output enable pins ($\overline{ENP}_2$ to $\overline{ENP}_5$), making it is possible to control enabling/disabling of the power supply control IC connected to the VBUS, and report overcurrent detection to the host, on an individual port basis.

As exchanges with the USB host are all executed automatically within the USB hub, USB hub block exchanges with the slave CPU are limited to the following cases:

1. USB module reset or operation halt
   a. Slave CPU system reset
      (Internal reset by $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ input, or WDT0)
   b. Module stop condition initiated by slave CPU
      (USB module stopped by means of MSTPCR)
   c. USB module reset by means of HPLLRST or HSRST bit in USBCR

2. Downstream port overcurrent detection and power supply output enable control
   a. Overcurrent detection and power supply output enable for individual ports by bits HOC5E to HOC2E in HOCCR.
      (When a downstream port itself is not used, DSD+/DSD- pins require pullup/pulldown as specified.)

7.3.3 Functions of USB Function

The USB function block has three endpoints.

By using a combination of endpoint 2 enabling/disabling and IN/OUT mode with endpoint 1 MaxPacketSize, the three alternates shown below can be selected for the USB function block. Twice the MaxPacketSize value is set for the number of FIFO bytes.

As the command that selects the alternate is a USB standard command, it is not possible to notify the slave CPU of the alternate selected. It is therefore necessary to ensure that the selected alternate is the same for the H8 firmware and the host CPU device driver.

			Endpoint 0		Endpoint 1		Endpoint 2	
Configuration	**Interface**	**Alternate**	**IN/OUT**	**FIFO**	**IN/OUT**	**FIFO**	**IN/OUT**	**FIFO**
1	0	0	IN/OUT	16 bytes each	IN	16 bytes	IN	16 bytes
		1	IN/OUT	16 bytes each	IN	16 bytes	OUT	16 bytes
		2	IN/OUT	16 bytes each	IN	32 bytes	None	None

The USB function supports control transfer by means of endpoint 0 and input transfer by means of endpoints 1 and 2.

A control transfer consists of a number of transactions. The command transmitted from the host in the SETUP transaction is first decoded by the USB function core.

When a SETUP token is received, FVSR is initialized and EP0OTC is set to 1, and command reception is enabled. If the received command is a USB standard command (other than GetDescriptor or SetDescriptor), the EP0OTF flag is set and the slave CPU is notified of the fact that a USB standard command has been received. In this case, the remaining transactions in the control transfer are processed within the USB function without intervention by the slave CPU.

If the received command is a GetDescriptor or SetDescriptor command, or a command specific to a device class, the EP0OTS flag is set. The slave CPU must read the command from the FIFO, then decode and execute it. The remaining transactions in the control transfer must also be processed by the slave CPU using the FIFO, etc.

Input transfers consist of individual IN or OUT transactions. These must all be processed by the slave CPU using the FIFO, etc.

When processing by the slave CPU is necessary as described above, the communication processing load is shared between the USB function and the slave CPU. The roles of the USB function and the slave CPU, and the flag and bits used in the interface, are shown in table 7.3.

Table 7.3 Role Sharing between USB Function and Slave CPU

Item/Description		Operating Hardware	Related Registers/ Flags/Bits
1	D+/D– signal analog ↔ digital conversion	Port block	—
		USB function core	
2	Serial ↔ parallel conversion/bit stuffing	USB function core	SOFF
	PID determination/addition, CRC determination/addition		
3	Token packet determination/notifying slave CPU of SETUP	USB function core	SETUPF
4	Handshake packet determination/generation	USB function core	
	DAT0/1 PID toggling, FIFO rewinding, ACK/NAK detection/return		FVSR, EPTE
	ACK handshake detection and slave CPU notification/ACK handshake return		TS, EPTS
	Data error detection and slave CPU notification/NAK handshake return		TF, EPTF
	STALL handshake return		EPSTL
5	Data packet reception/regeneration/transfer to slave CPU	USB function core	FIFO
6	USB command decoding and execution	USB function core	FIFO
		Slave CPU	

Processing of electrical signals on the USB bus line and processing of signal bit streams is performed by the bus driver/receiver in the port block and the USB function core block. The token/acknowledgment type and data bytes are extracted and, conversely, acknowledgment and data bytes are converted to bit stream electrical signals (items 1 and 2).

When a SETUP token is received, if a GetDescriptor or SetDescriptor command, or a command specific to a device class, is received, the EP0OTS flag is set and the slave CPU is notified (item 3). The command itself is transferred using the FIFO, and must be decoded and executed by the slave CPU (item 6). The remaining transactions in the control transfer must also be processed by the slave CPU using the FIFO, etc. (items 4 and 5).

Reception of an IN or OUT token in control transfer or interrupt transfer is not reported to the CPU, and the operation continues with data transfer. In the case of an IN transaction, the transmit data is prepared in the FIFO beforehand, and if the EPTE bit is set transmission is started, or if not, a NAK handshake is performed. When an IN transaction ends, normal or abnormal termination of the transfer is confirmed by means of the host handshake, and is reported to the slave CPU by means of TS/TF/EPTS/EPTF. In the case of an OUT transaction, an ACK handshake is performed when all the data has been received in the FIFO, or a NAK handshake if it was not possible to receive all the data. With both IN transactions and OUT transactions, a STALL handshake is performed if the endpoint is placed in the stall state by means of EPSTL.

7.3.4 Operation when SETUP Token Is Received (Endpoint 0)

The group of transactions initiated when the host issues a SETUP token is called a control transfer. A control transfer consists of three stages: setup, data, and status. Control transfers are of two kinds: control write transfers and control read transfers. The type of transfer (read or write) and the number of transfer bytes in the data stage are determined by the 8-byte command transferred OUT in the setup stage.

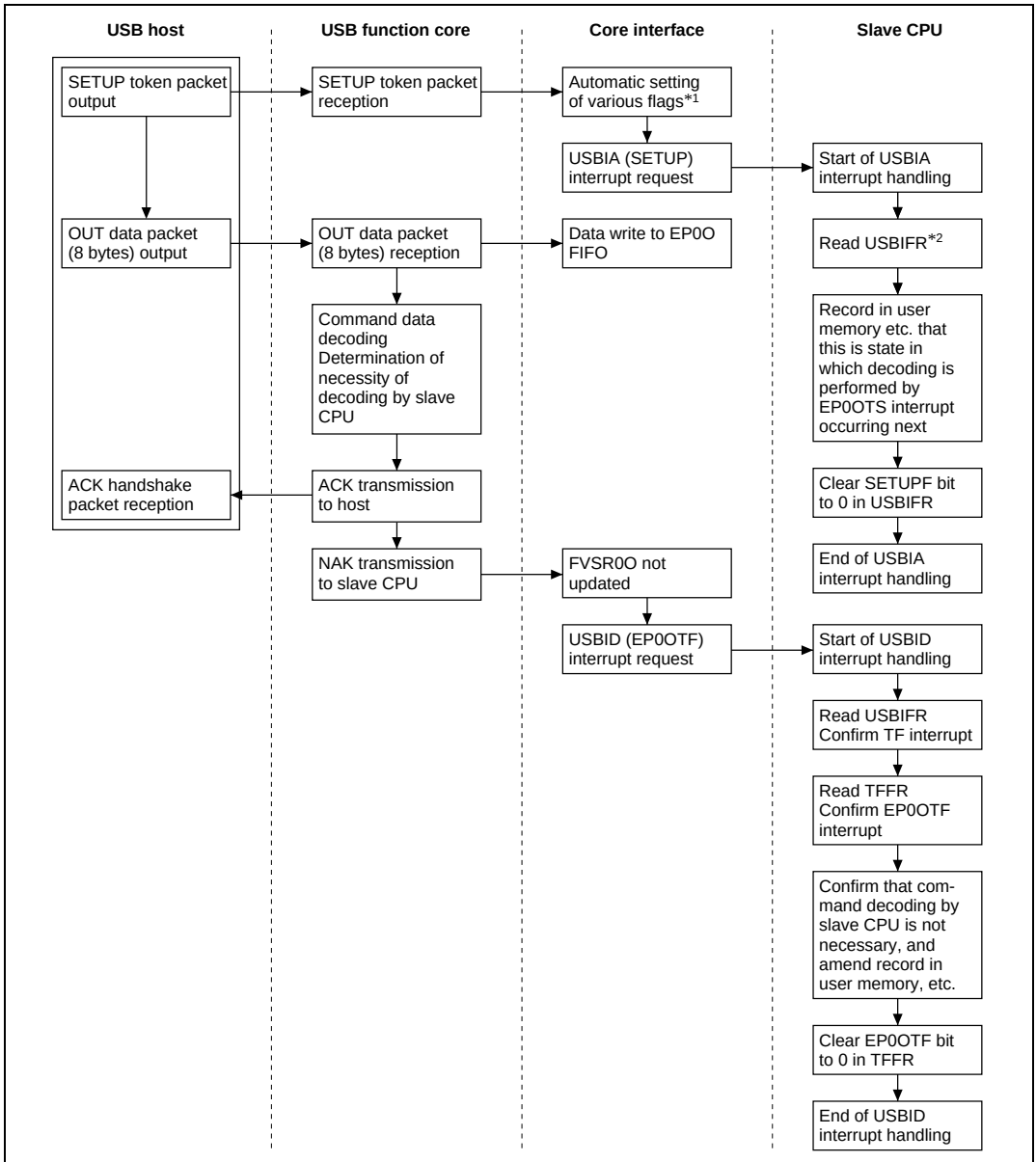
The setup stage consists of a setup transaction, the data stage may have no transaction or one or more data transactions, and the status stage consists of a single data transaction. The packets contained in each transaction are shown in the table 7.4.

Table 7.4 Packets in Each Transaction

Stage		Token Phase	Data Phase	Handshake Phase[*1]
Setup stage		SETUP token packet	OUT data packet (8 bytes) (host → slave)	ACK handshake packet (slave → host)
Control write transfer	Data stage	OUT token packet	OUT data packet (host → slave)	ACK/NAK/STALL handshake packet (slave → host)
	Status stage	IN token packet	IN data packet (0 bytes)[*2] (slave → host)	ACK handshake packet (host → slave)
			NAK/STALL handshake packet (slave → host)	—
Control read transfer	Data stage	IN token packet	IN data packet (slave → host)	ACK handshake packet (host → slave)
			NAK/STALL handshake packet (slave → host)	—
	Status stage	OUT token packet	OUT data packet (host → slave)	ACK/NAK/STALL handshake packet (slave → host)
No data stage	Status stage	IN token packet	IN data packet (0 bytes)[*2] (slave → host)	ACK handshake packet (host → slave)
			NAK/STALL handshake packet (slave → host)	—

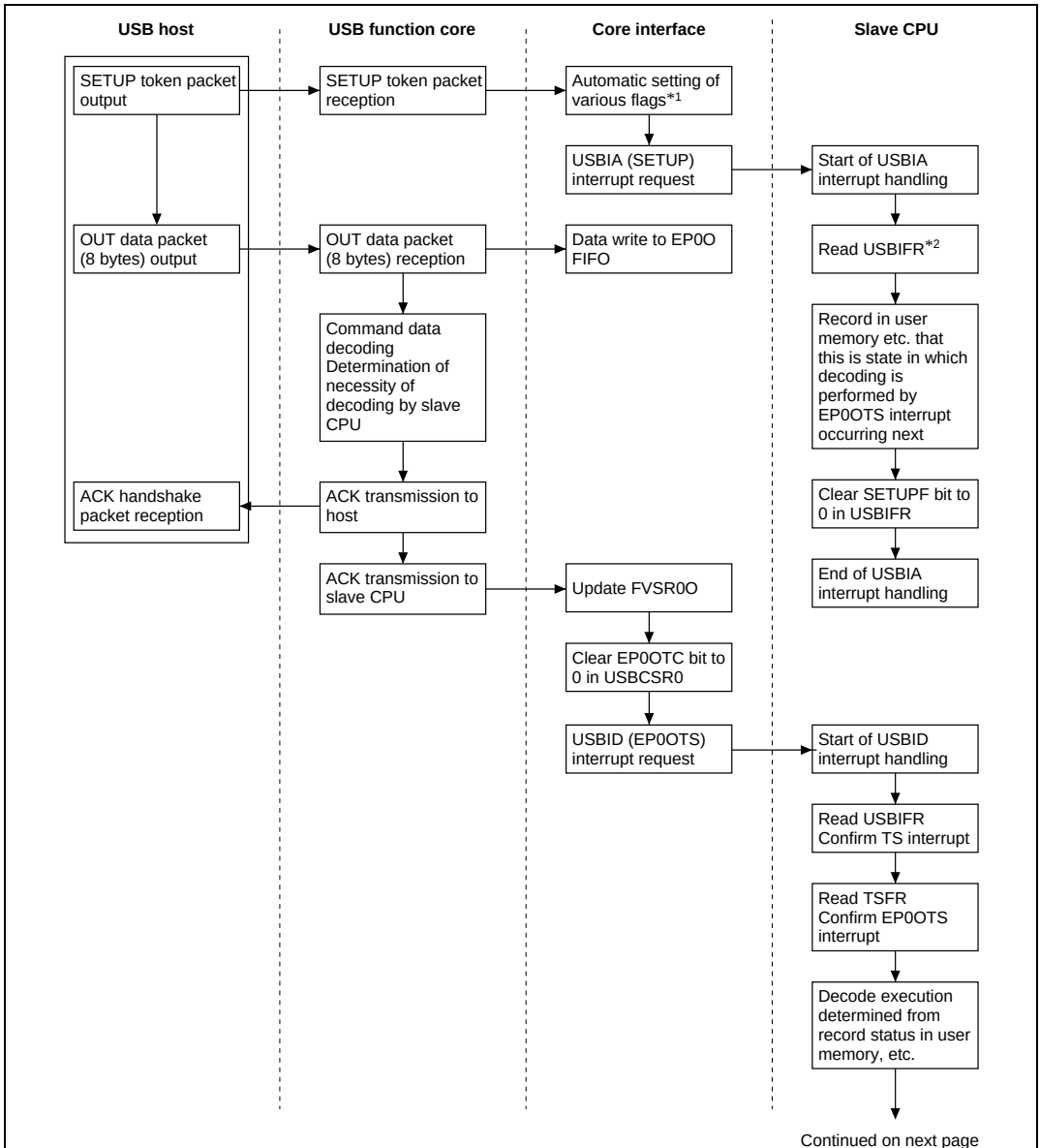
Notes: 1. This phase is present only if a data packet transfer was executed in the data phase.
2. When all the data in the FIFO has been transferred and the FIFO is empty, the EPTE bit is cleared to 0. If an IN transaction is then started, a NAK handshake is returned.
A 0-byte data packet is transferred by setting the EPTE bit to 1 when the FIFO is empty.

Figure 7.2 shows the operation of the USB function core and the H8 firmware when the USB function receives a SETUP token (setup transaction). For other cases, see section 7.3.5, Operation when OUT Token Is Received, and section 7.3.6, Operation when IN Token Is Received.



Notes: 1. Bit EP0OTC set to 1 in USBCSR0, FVSR0I and FVSR0O initialized, bits EP0ITS and EP0OTS cleared to 0 in TSFR, bits EP0ITF and EP0OTF cleared to 0 in TFFR, bit EP0STL cleared to 0 in EPSTLR.
2. As the USBIA interrupt is assigned only to the SETUP interrupt, there is no need for processing to determine the interrupt source.

Figure 7.2 (1) Operation when SETUP Token Is Received (Decoding by Slave CPU Not Required)



Notes: 1. Bit EP0OTC set to 1 in USBCSR0, FVSR0I and FVSR0O initialized, bits EP0ITS and EP0OTS cleared to 0 in TSFR, bits EP0ITF and EP0OTF cleared to 0 in TFFR, bit EP0STL cleared to 0 in EPSTLR.
2. As the USBIA interrupt is assigned only to the SETUP interrupt, there is no need for processing to determine the interrupt source.

Figure 7.2 (2) Operation when SETUP Token Is Received (Decoding by Slave CPU Required)

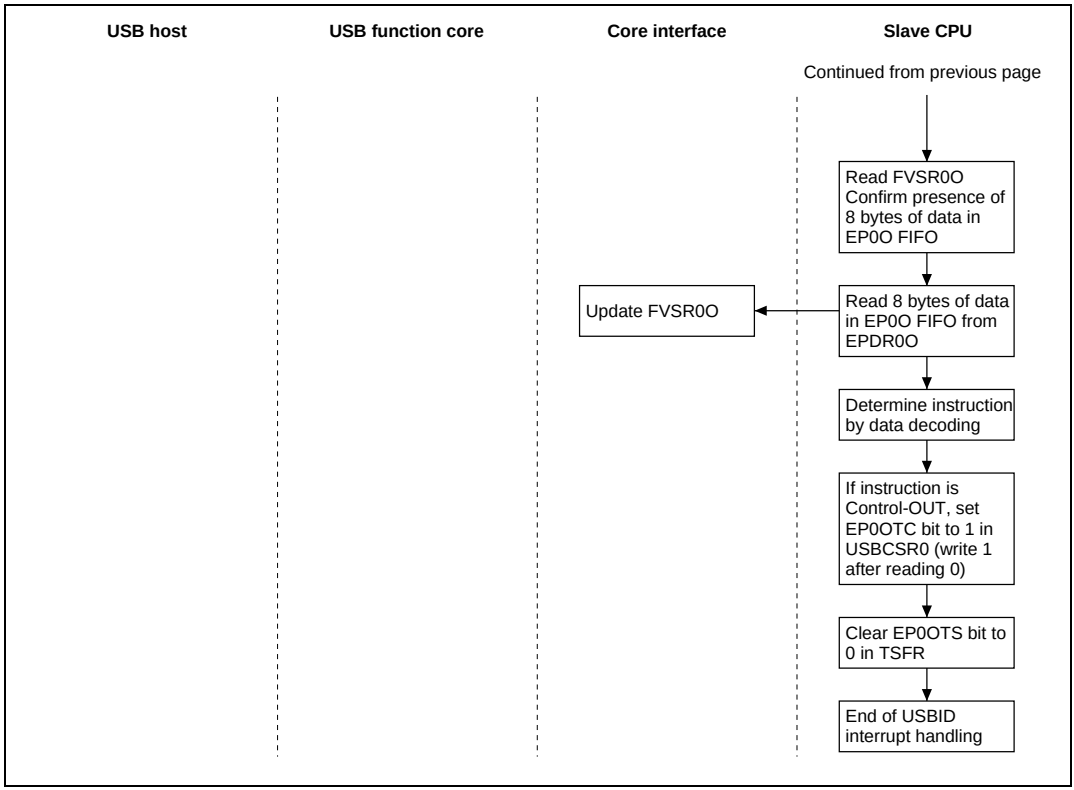
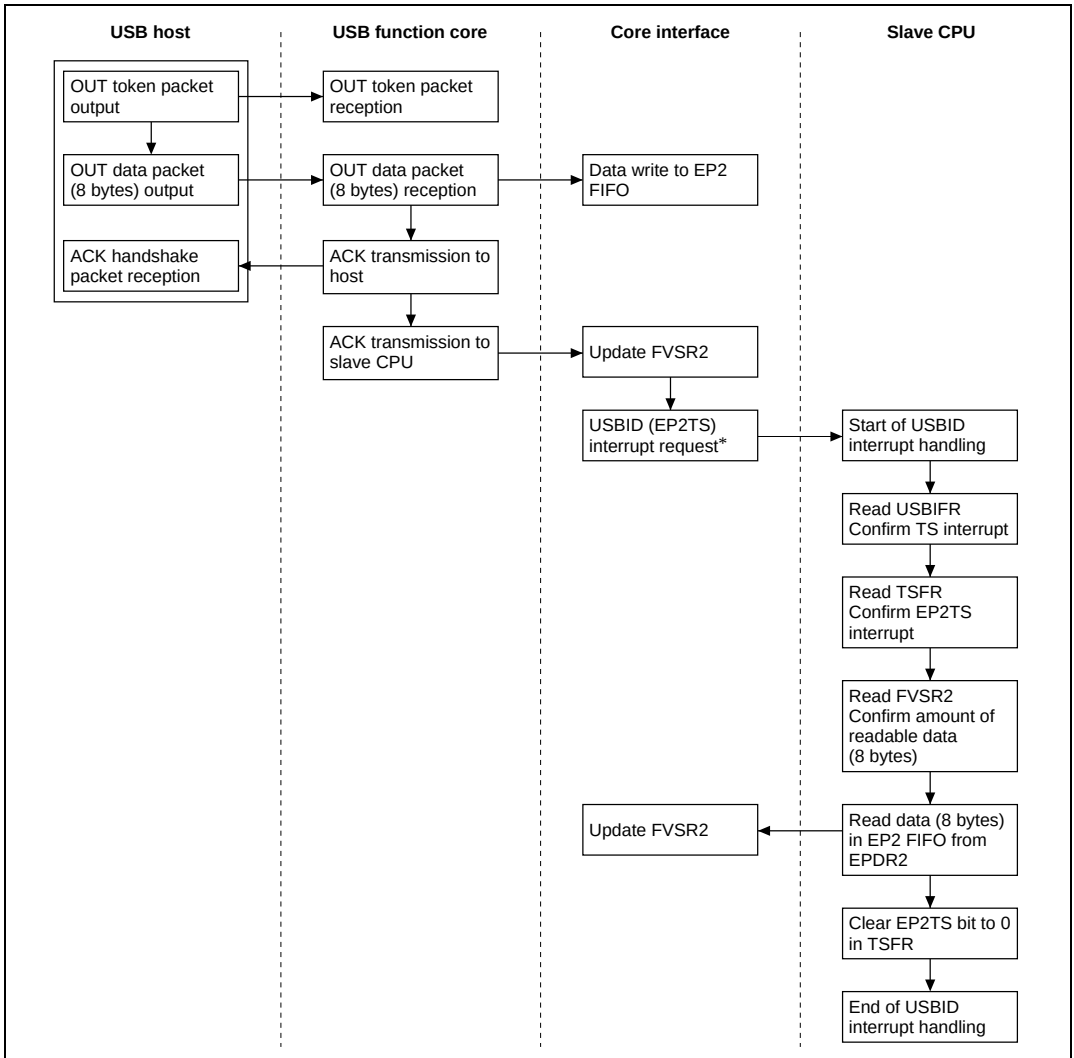


Figure 7.2 (2) Operation when SETUP Token Is Received (Decoding by Slave CPU Required) (cont)

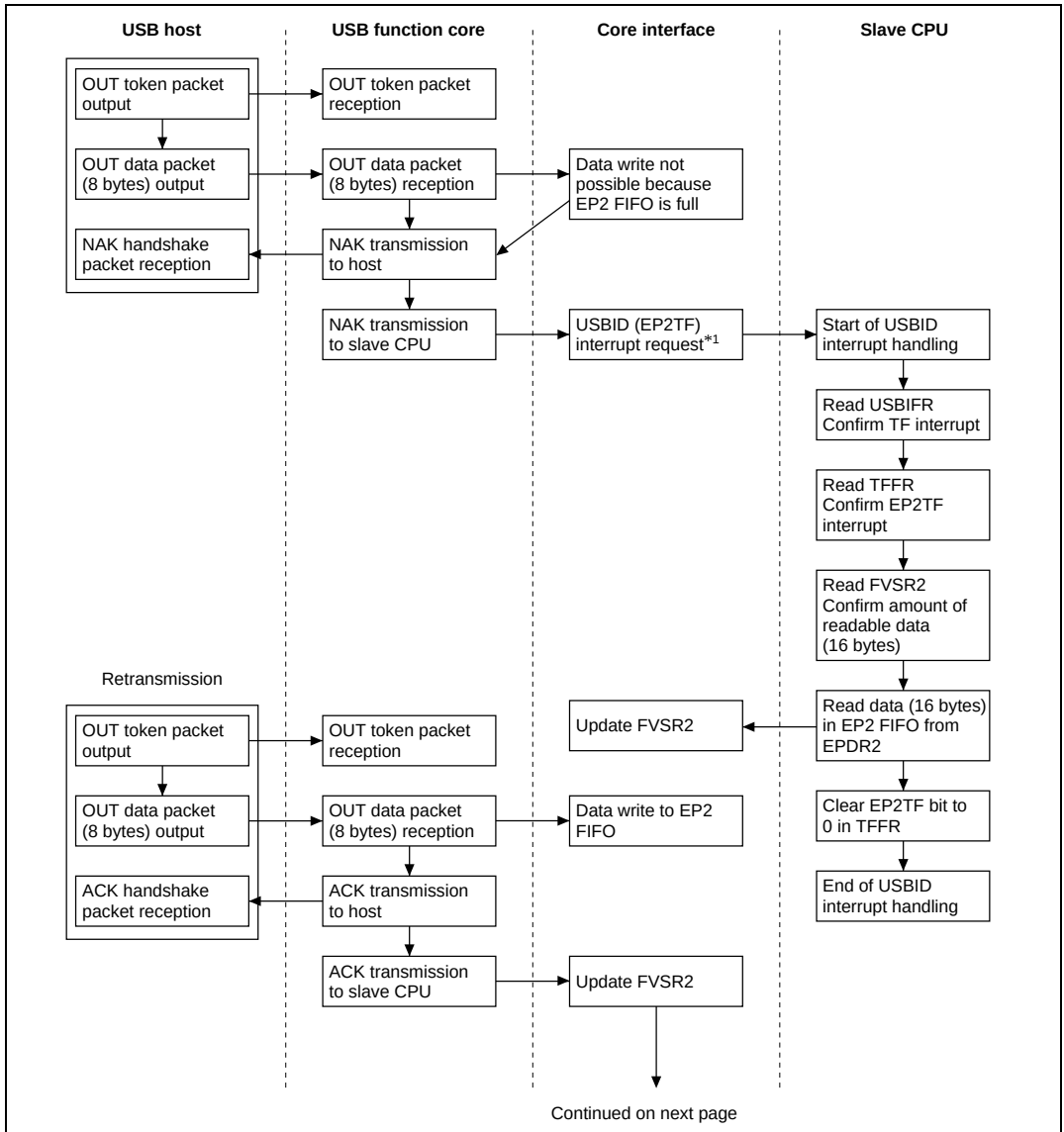
7.3.5 Operation when OUT Token Is Received (Endpoints 0 and 2)

Figure 7.3 shows the operation of the USB function core and the H8 firmware when the USB function receives an OUT token (OUT transaction). OUT transactions are used in the data stage and status stage of a control transfer, and in an input transfer.



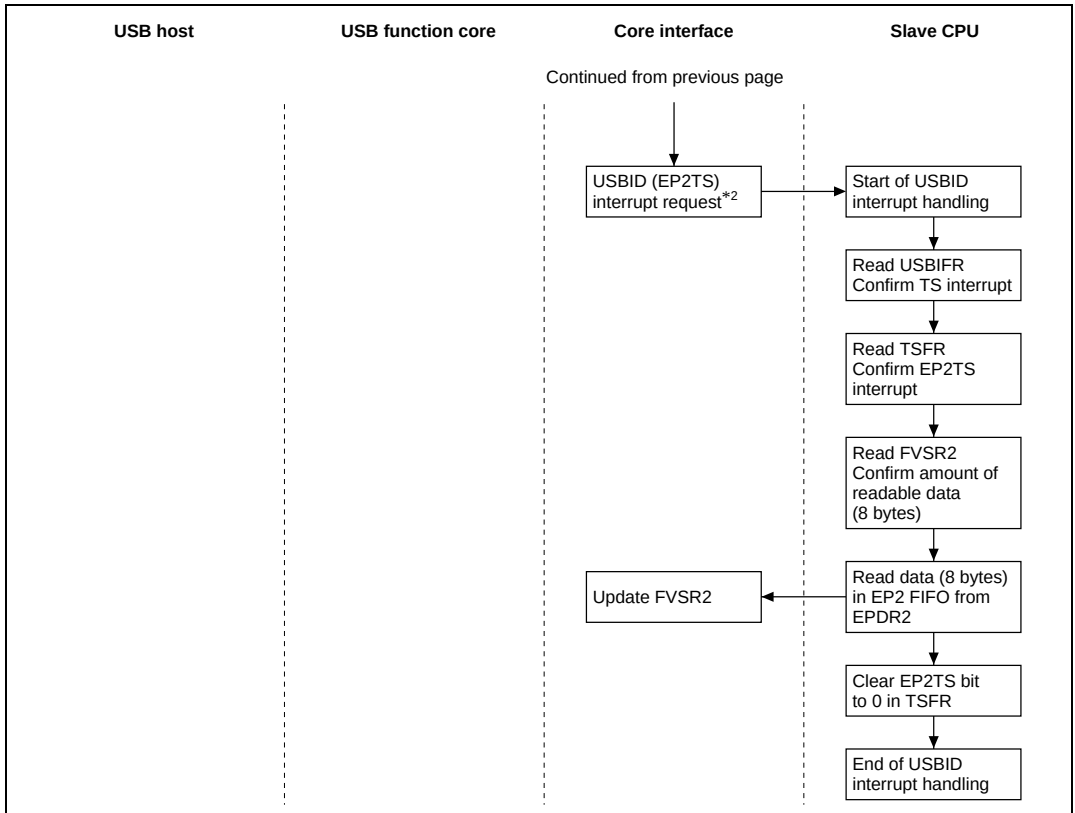
Note: * When the EP2TS interrupt is set for USBIB or USBIC by the INTSELR0 setting, that interrupt request is generated. When an USBIB or USBIC interrupt is generated, there is no need for processing to determine the interrupt source. (A register read is necessary in order to write 0 after reading 1.)

Figure 7.3 (1) Operation when OUT Token Is Received (EP2-OUT: Initial FIFO Empty)



Note: 1. When the EP2TF interrupt is set for USBIB or USBIC by the INTSELR0 setting, that interrupt request is generated. When an USBIB or USBIC interrupt is generated, there is no need for processing to determine the interrupt source. (A register read is necessary in order to write 0 after reading 1.)

Figure 7.3 (2) Operation when OUT Token Is Received (EP2-OUT: Initial FIFO Full)

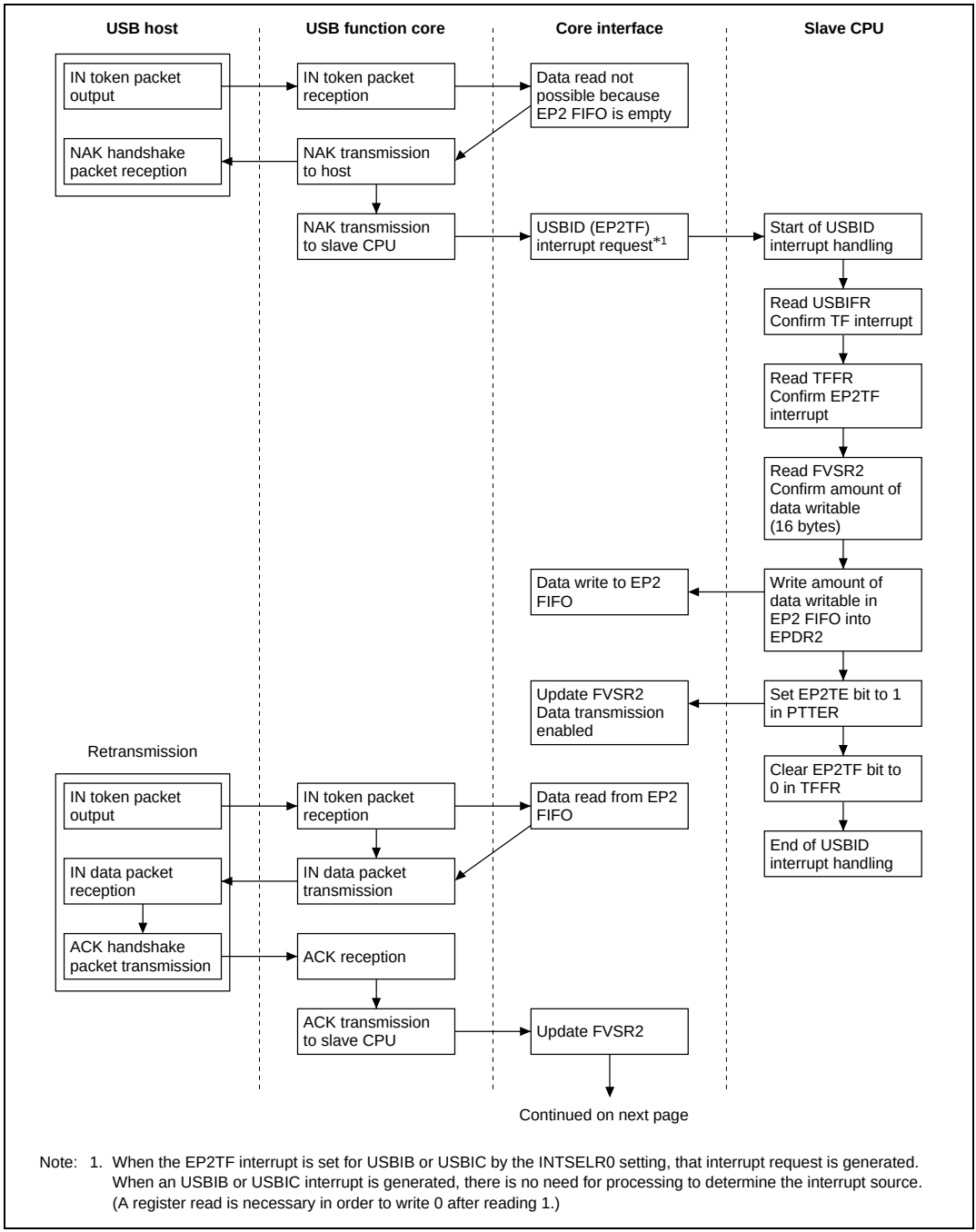


Note: 2. When the EP2TS interrupt is set for USBIB or USBIC by the INTSELR0 setting, that interrupt request is generated. When an USBIB or USBIC interrupt is generated, there is no need for processing to determine the interrupt source. (A register read is necessary in order to write 0 after reading 1.)

Figure 7.3 (2) Operation when OUT Token Is Received (EP2-OUT: Initial FIFO Full) (cont)

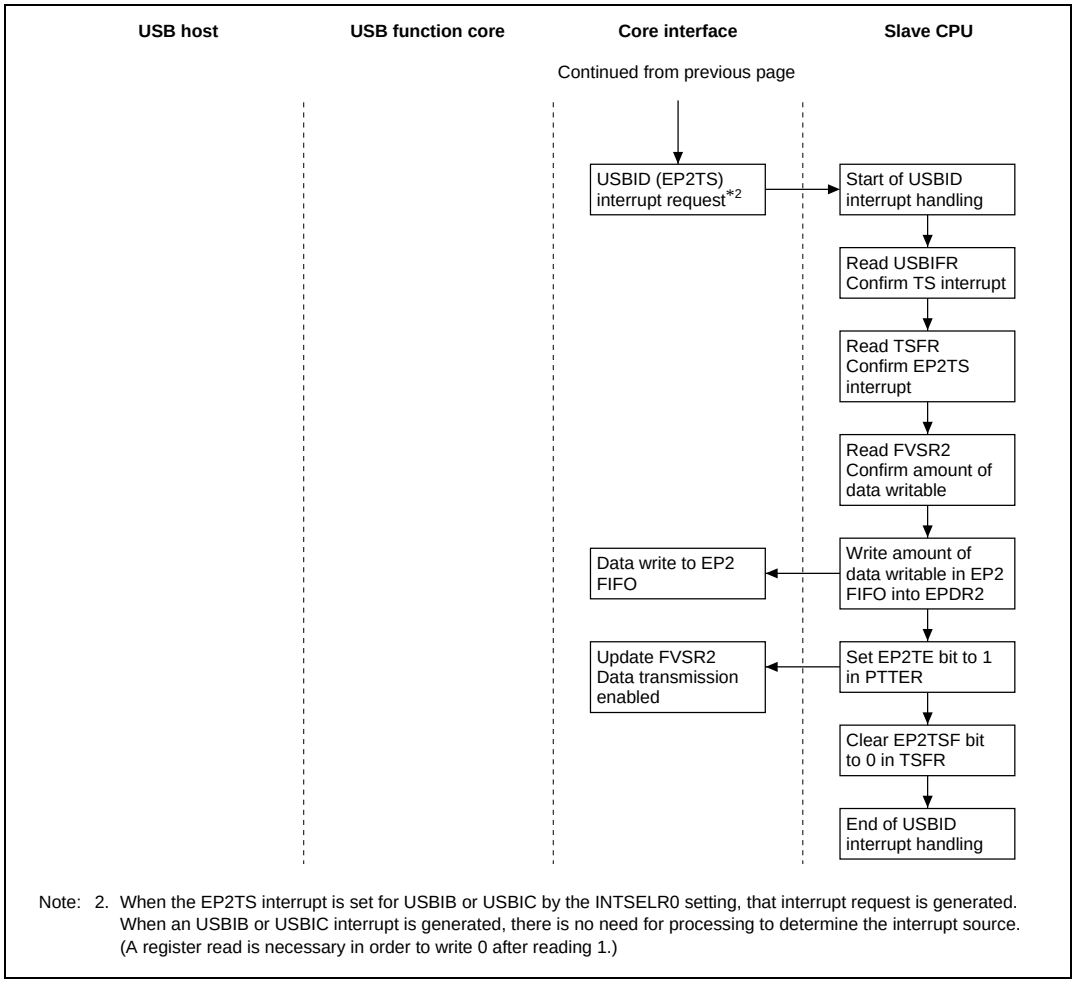
7.3.6 Operation when IN Token Is Received (Endpoints 0, 1, and 2)

Figure 7.4 shows the operation of the USB function core and the H8 firmware when the USB function receives an IN token (IN transaction). IN transactions are used in the data stage and status stage of a control transfer, and in an input transfer.



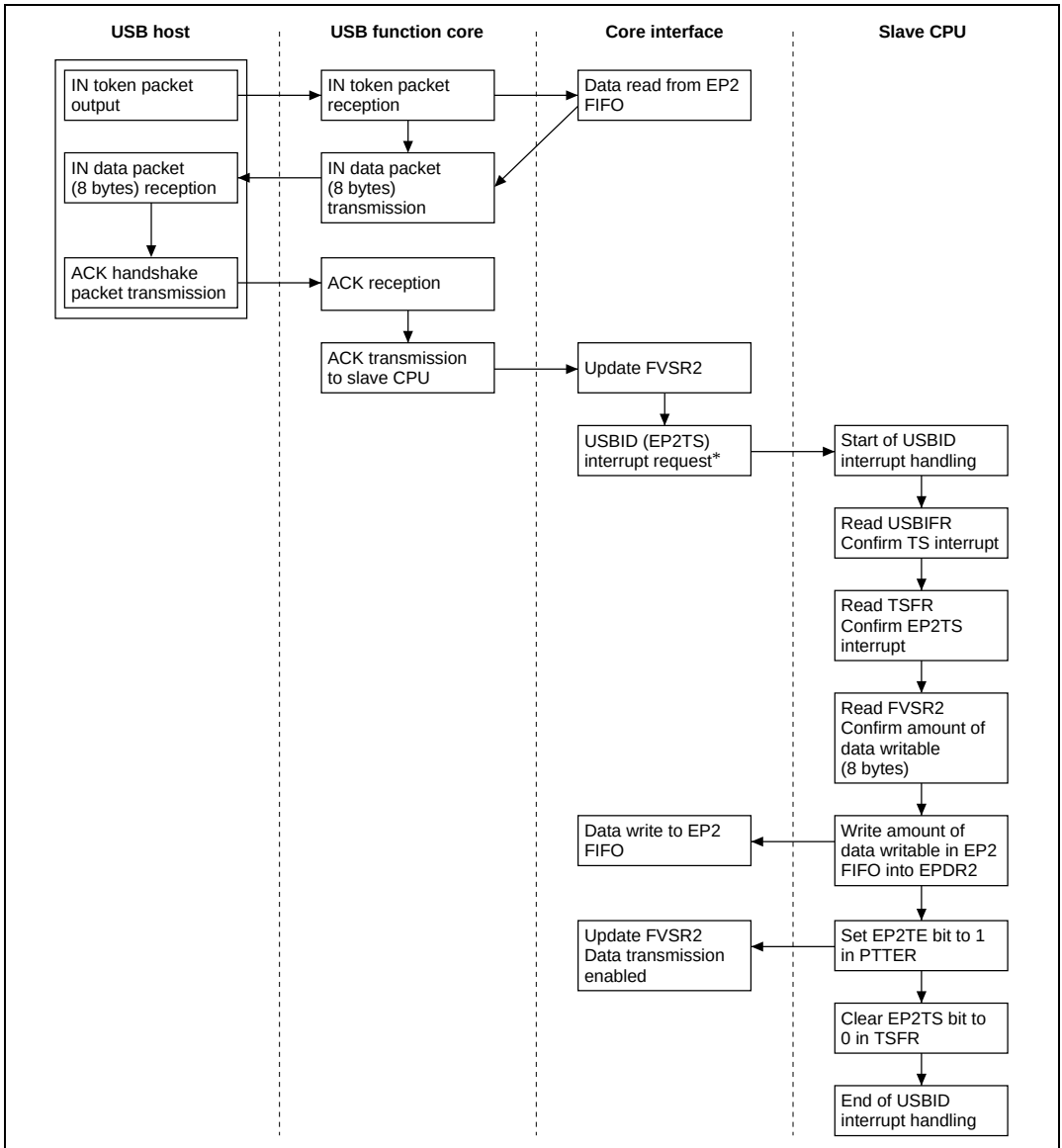
Note: 1. When the EP2TF interrupt is set for USBIB or USBIC by the INTSELR0 setting, that interrupt request is generated. When an USBIB or USBIC interrupt is generated, there is no need for processing to determine the interrupt source. (A register read is necessary in order to write 0 after reading 1.)

Figure 7.4 (1) Operation when IN Token Is Received (EP2-IN: Initial FIFO Empty)



Note: 2. When the EP2TS interrupt is set for USBIB or USBIC by the INTSELR0 setting, that interrupt request is generated. When an USBIB or USBIC interrupt is generated, there is no need for processing to determine the interrupt source. (A register read is necessary in order to write 0 after reading 1.)

Figure 7.4 (1) Operation when IN Token Is Received (EP2-IN: Initial FIFO Empty) (cont)



Note: * When the EP2TS interrupt is set for USBIB or USBIC by the INTSELR0 setting, that interrupt request is generated. When an USBIB or USBIC interrupt is generated, there is no need for processing to determine the interrupt source. (A register read is necessary in order to write 0 after reading 1.)

Figure 7.4 (2) Operation when IN Token Is Received (EP2-IN: Initial FIFO Full)

7.3.7 Suspend/Resume Operations

If the USB data line is idle for a period longer than that stipulated in the USB Specification, the H8/3567 Group's USB hubs and USB function automatically enter the suspend state.

The suspend state is automatically cleared (i.e. operation is resumed) when the upstream side (host) restarts data transmission, but operation can also be forcibly resumed by the USB function (remote wakeup).

Changes in the suspend/resume state can be ascertained by means of the SPNDIF and SPNDOF flags. Remote wakeup is executed by setting the DVR bit.

7.3.8 USB Module Reset and Operation-Halted States

A reset or operation-halted state can be set for the USB module by means of a number of control bits. For information on sequential setting of these bits when starting up the USB module, see section 7.3.9, USB Module Startup Sequence.

There are several kinds of USB module reset and operation-halted state, as listed below. In the hardware standby and reset, the entire USB module is initialized. In the descriptions of individual bits in the register descriptions, this initialization condition is not indicated, and only "(Initial value)" is shown.

1. Hardware standby state
2. Reset state
3. Module stop state
4. Software standby state
5. USB function stop state
6. USB function only state
7. USB bus reset state
8. USB suspend state

Hardware Standby State: When the H8/3567 Group's $\overline{\text{STBY}}$ pin is driven low, the chip enters the hardware standby state. In the hardware standby state, all the H8/3567 Group's initializable registers and internal states are initialized, and all H8/3567 Group pins go to the high-impedance state. XTAL-EXTAL system clock oscillation and XTAL12-EXTAL12 USB clock oscillation both halt.

Reset State: When the H8/3567 Group's $\overline{\text{RES}}$ pin is driven low, the chip enters the reset state. In the reset state, all the H8/3567 Group's initializable registers and internal states are initialized, and all H8/3567 Group pins go to the input state. XTAL-EXTAL system clock oscillation is enabled.

Module Stop State: When bit 1 of MSTPCR is set to 1, the USB module enters the module stop state. In the module stop state, supply of system clock to the USB module is stopped. However, when USB clocks (XTAL12, EXTAL12) are selected as USB operating clocks, the USB module does not stop the operation. When setting the USB module stop state, return the value of UPLLCR to the initial state. Also, it is recommended to return the value of USBCR to the initial state to prepare for cancellation of the module stop state. As bit 1 of MSTPCR is initialized to 1 by a transition to hardware standby mode or a reset, the USB module is in the module stop state after reset release.

Software Standby State: When a SLEEP instruction is executed after setting the SSBY bit to 1 in SBYCR, the chip enters the software standby state. In the software standby state the USB module does not enter the reset or operation-halted state. However, since the USB function cannot fulfill its role when the slave CPU halts due to a transition to the software standby state, operation of the USB function must be halted before the software standby state setting is made. Set the FNCSTP bit to 1 in USBCR to disconnect the USB function from the bus (see USB Function Stop State below).

In the software standby state, XTAL-EXTAL system clock oscillation halts. If the system clock has been set as the USB operating clock by means of the CKSEL bits in UPLLCR, the USB hubs cannot operate, either, since the clock is halted. If the USB clock (XTAL12-EXTAL12) has been set as the USB operating clock, the hub block alone can operate.

USB Function Stop State: When the FNCSTP bit is set to 1 in USBCR, the USB function stop state is entered. In the USB function stop state, the USB function is disconnected from the bus. If the FONLY bit has been cleared to 0 in USBCR, internal connection between the USB function and USB hub is also cut. If the FONLY bit has been set to 1, the USB function is connected to the upstream port USD+/USD– pins. If the FNCSTP bit and FSRST bits are both set to 1, the USD+/USD– pins go to the high-impedance state.

The USB operating clock supply to the USB function block is halted.

Clearing the USB function stop state requires execution of the USB function block related sequence described in section 7.3.9, USB Module Startup Sequence. When setting the USB function stop state, it is recommended that the UIFRST, FPLLRST, and FSRST bits be set to 1 in USBCR in preparation for reduced current dissipation and release. As a result, the following registers are initialized.

Registers	UIFRST/FSRS	Notes
EPDR2, EPDR1, EPDR0O, EPDR0I	FSRST	
FVSR2, FVSR1, FVSR0O, FVSR0I	FSRST	
EPSZR1	UIFRST	
USBIER	UIFRST	
USBIFR, TSFR, TFFR	FSRST	
USBCSR0	FSRST	Bits 3 to 0 only
EPSTLR	FSRST	
EPDIR	UIFRST	
INTSELR0, INTSELR1	UIFRST	

USB Function Only State: When the FONLY bit is set to 1 in USBCR, the USB function only state is entered. In the USB function stop state, the USB function is connected to the upstream port, and the USB operating clock supply to the USB hub block is halted.

It is recommended that USB hub block operation be halted by setting the HSRST bit to 1. This will place the downstream ports in the high-impedance state and enable port D, which also has a downstream port function, to operate as a general I/O port. HOCCR should be initialized to H'00.

USB Bus Reset State: When a new device is connected to the USB bus, or when error recovery is executed, the USD+/USD– pin signals go to the bus reset state for a given period.

In the USB function, the bus reset interrupt flag is set to 1 when a USB bus reset is detected. A bus reset initializes the USB hub internal state to the default state. Control registers that select the USB function internal state and USB function operating state are not initialized by a USB bus reset. These registers must be initialized by setting the FSRST bit to 1.

Registers initialized by the UIFRST bit are not initialized by a USB bus reset.

USB Suspend State: If the USB bus remains idle for longer than a certain time, the USB hub block and USB function block enter the suspend state. In the suspend state, some operating clocks are halted internally and current dissipation is reduced.

When the USB function enters the suspend state, or when the suspend state is cleared by a change in the USD+/USD– pin signals, the suspend IN interrupt flag or suspend OUT interrupt flag, respectively, is set to 1.

The remote wakeup from the suspend state can be executed by write 1 to the DVR bit in DEVRSMR.

7.3.9 USB Module Startup Sequence

Component Elements: The USB module has a number of component elements requiring startup in a fixed sequence by firmware (an H8 program) to ensure normal operation and correct recognition by the USB host.

The USB components that need to be considered are as follows:

a. USB clock pulse generator (12 MHz), USB operating clock generation PLL (48 MHz)
b. USB bus clock synchronization DPLL (12 MHz)
c. EPINFO—Endpoint configuration information
d. Slave CPU, core interface
e. USB hub core, USB function core

a. USB clock pulse generator (12 MHz), USB operating clock generation PLL (48 MHz)

   The USB clock pulse generator is connected to XTAL12-EXTAL12 and generates a 12 MHz USB clock. The USB operating clock PLL, multiplies the clock input from the USB clock pulse generator or system clock pulse generator to give a 48 MHz clock. The input clock frequency must be 8, 12, 16, or 20 MHz.

   As USB clock pulse generator oscillation has not started when a system reset is released, an oscillation stabilization period 10 ms that includes the USB operating clock PLL must be provided by firmware. Oscillation is started when XTAL12-EXTAL12 is set as the USB clock source with the CKSEL bits in UPLLCR. The PLL multiplication factor is selected with the PFSEL bits in UPLLCR. While waiting for oscillation to stabilize 10 ms, the UIFRST, HPLLRST, HSRST, FPLLRST, and FSRST bits are set to 1 in USBCR, placing the USB bus clock synchronization DPLL, USB hub core, USB function core, etc., in the reset state.

b. USB bus clock synchronization DPLL (12 MHz)

   USB data transfer is performed at a maximum rate of 12 Mbps. The bit data sampling timing can be controlled by adjusting the phase during reception of the synchronization pattern that precedes a packet, using the 48 MHz USB operating clock. This mechanism is called the USB bus clock synchronization DPLL.

   A USB bus clock synchronization DPLL operation stabilization period must be provided by firmware. While waiting for operation to stabilize, the HSRST and FSRST bits are set to 1 in USBCR, placing the USB hub core, USB function core, etc., in the reset state.

   An operation stabilization period of at least ten 48 MHz clock cycles is recommended.

c. EPINFO—Endpoint configuration information

The USB function core block can handle both bulk transfer and isochronous transfer, but for reasons related to the CPU interface specifications and the data transfer capability of the CPU itself, the H8 handles only control transfer and interrupt transfer processing.

Information comprising settings for the number of endpoints, supported transfer types, maximum packet byte length, etc. (EPINFO) is written to the USB function block by firmware each time the USB function is initialized. In the H8/3567U and H8/3564U, three alternates are provided, and EPINFO is written for all three. However, since firmware has no way of knowing which alternate the host has selected, the module will not operate normally if the choice of alternate is changed during operation. The same alternate must be designated in the host driver software and the slave firmware.

Table 7.5 shows the endpoint configuration information to be written to the USB function block. Write all 65 one-byte values to EPDR01 in the order A1, A2, ... A5, B1, B2, ... M4, M5.

Table 7.5 Endpoint Configuration Information

	1	2	3	4	5
A	H'00	H'00	H'11	H'00	H'00
B	H'14	H'38	H'10	H'00	H'01
C	H'24	H'38	H'10	H'00	H'02
D	H'14	H'78	H'10	H'00	H'01
E	H'24	H'70	H'10	H'00	H'02
F	H'14	H'B8	H'20	H'00	H'01
G	H'35	H'20	H'10	H'00	H'03
H	H'45	H'20	H'10	H'00	H'04
I	H'55	H'20	H'10	H'00	H'05
J	H'65	H'20	H'10	H'00	H'06
K	H'36	H'20	H'10	H'00	H'03
L	H'46	H'20	H'10	H'00	H'04
M	H'56	H'20	H'10	H'00	H'05

d. Slave CPU, core interface

These are the basic parts that execute firmware. The slave CPU begins operating immediately after reset release, whereas core interface access is enabled when the module stop state is cleared.

e. USB hub core, USB function core

These are the central parts of the USB interface. Implementation of the USB bus interface is made possible by normal operation of component elements a to d.

Initial Operation Procedures: The initial operation procedures for the USB hubs and USB function are shown in figures 7.5 and 7.6.

When the USB module is used as a compound device, these two initial operation procedures must be executed, first for the USB hubs, then for the USB function. Clear the UIFRST bit to 0 before executing the USB function block procedure.

The compound device initial operation procedure is summarized below.

1. H8 is in power-off or hardware standby state
2. Power-on, $\overline{\text{STBY}}$ pin high-level application, etc., is performed, and finally high level is applied to $\overline{\text{RES}}$ pin and H8 starts operating
3. USBE bit in STCR is set to 1 by firmware
4. USB module is released from module stop state by firmware
5. FONLY bit is cleared to 0 by firmware
6. HOCCR and PLLCR are set by firmware; wait for USB operating clock oscillation to stabilize
7. After elapse of 10 ms oscillation stabilization time, HPLLRST bit is cleared to 0 by firmware
8. After DPLL operation stabilization time, HSRST bit is cleared to 0 by firmware
   a. USB host (upstream port) performs USB hub block bus reset
   b. USB host performs USB hub block configuration
   c. → Start of USB hub block operation
9. UIFRST bit is cleared to 0 and USB function related registers are set by firmware
10. FNCSTP bit is cleared to 0 and USB function is connected to USB hub by firmware
11. FPLLRST bit is cleared to 0 by firmware
12. After DPLL operation stabilization time, FSRST bit is cleared to 0 by firmware
13. EPINFO is written to USB function core by firmware, and finally EPIVLD bit is set to 1
14. Wait for bus reset interrupt
    a. USB host (USB hub block) performs USB function block bus reset
    b. USB host performs USB function block configuration
    c. → Start of USB function block operation

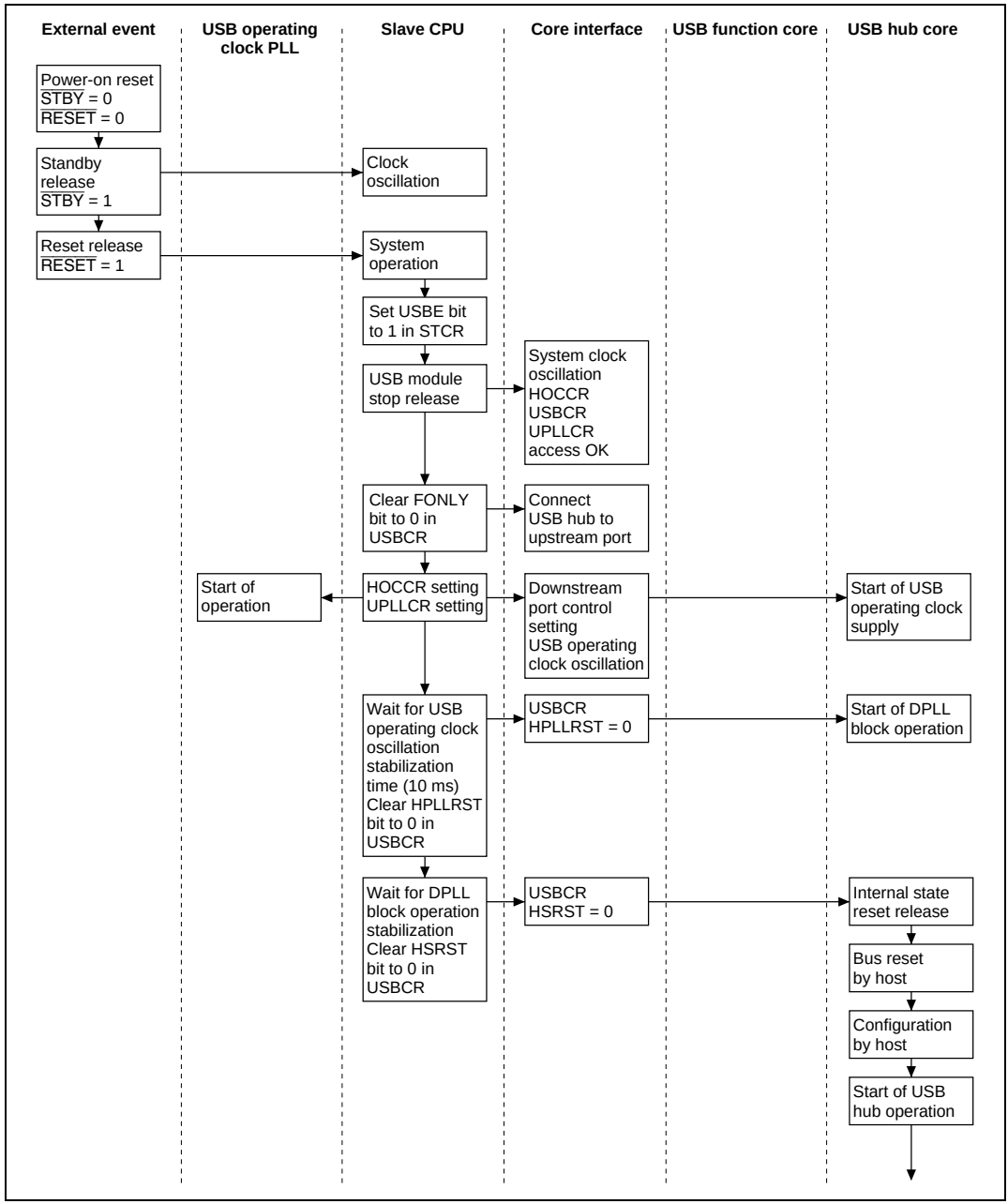


Figure 7.5 USB Hub Initial Operation Procedure

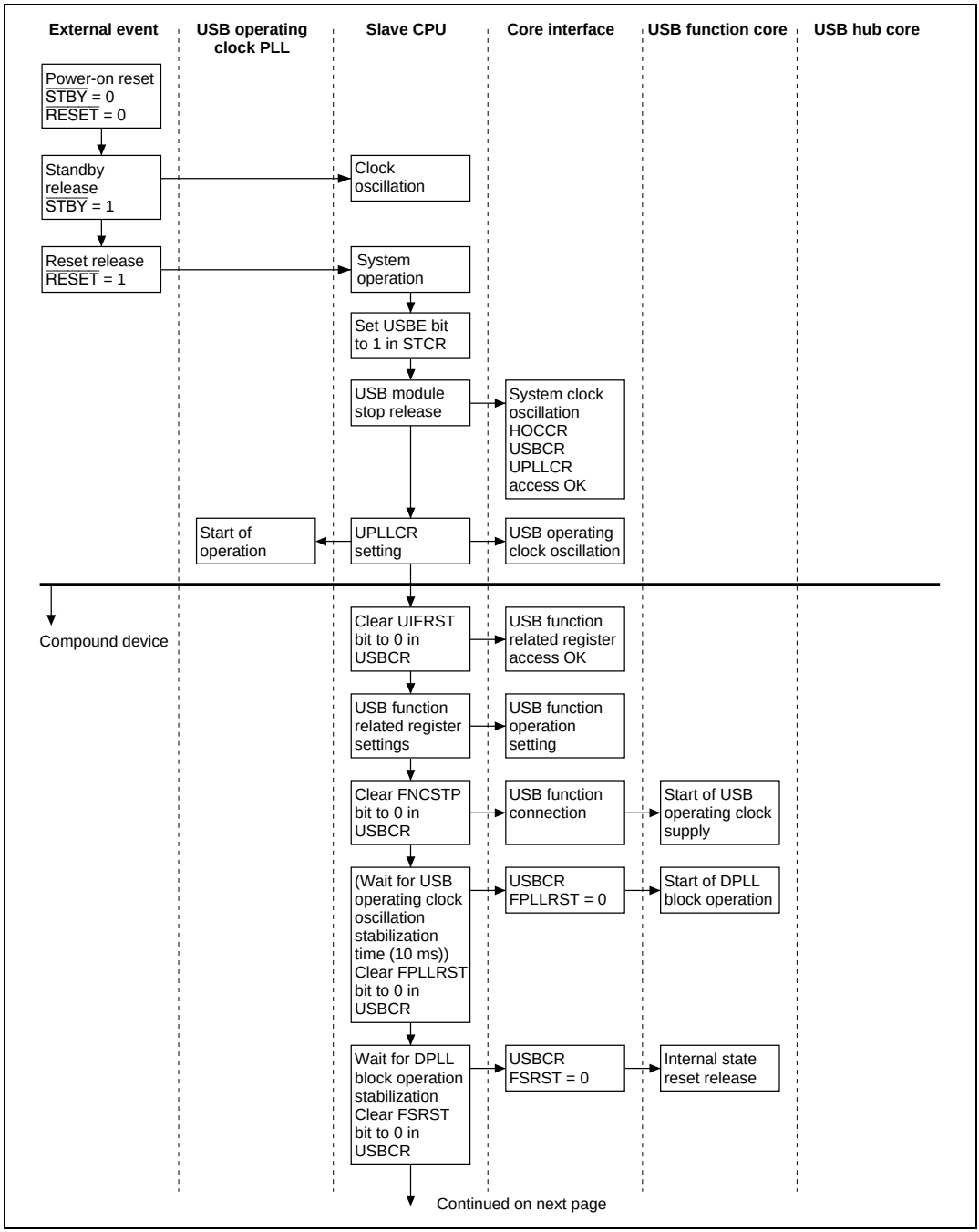


Figure 7.6 USB Function Initial Operation Procedure

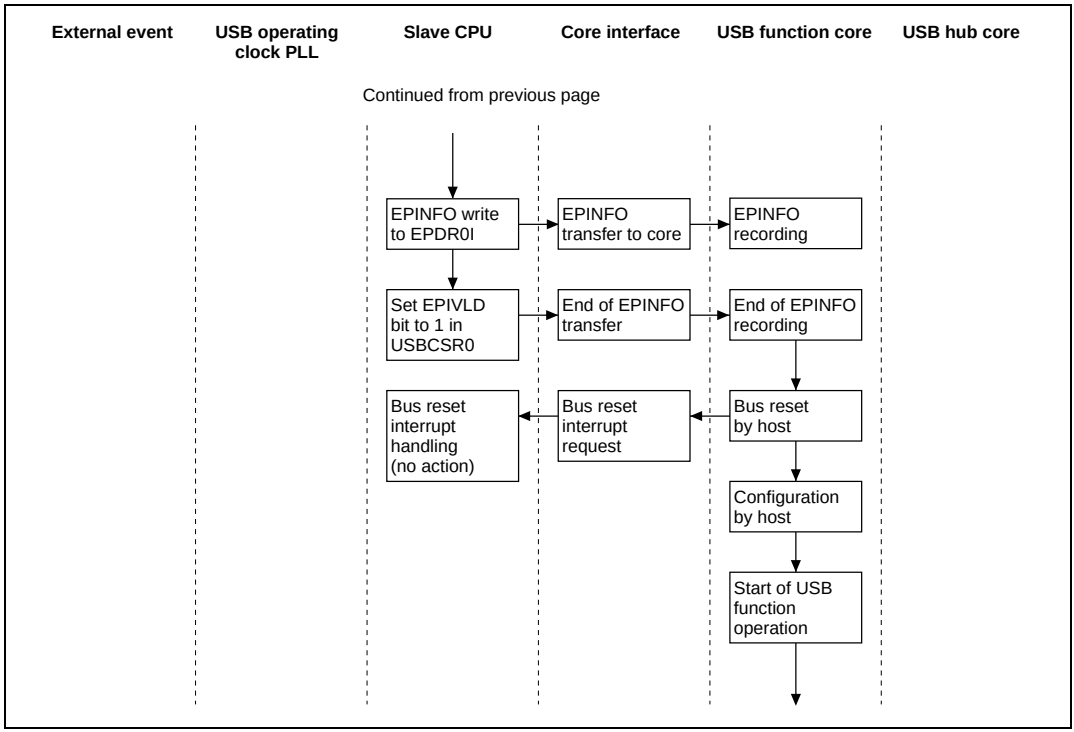


Figure 7.6 USB Function Initial Operation Procedure (cont)

Disconnection/Reconnection Procedures: The initial operation procedures for USB hub/USB function disconnection and reconnection are shown in figures 7.7 to 7.10. There are three kinds of USB function disconnection: compound device hub block upstream disconnection, upstream disconnection in USB function standalone mode, and compound device function block disconnection by firmware.

In the case of upstream disconnection, the USB bus continues in the idle state, and so the suspend state is entered. In order to detect reconnection, some method independent of the USB protocol is needed, such as detecting VBUS connection by means of an interrupt. Trigger events (such as cutoff of the system power supply) whereby the USB function block is disconnected by firmware also require detection by means of a separate interrupt, etc.

When USB hub upstream disconnection occurs in the compound device state, the USB function block enters the suspend state. When upstream reconnection is detected by means of an external interrupt, etc., initialization of both the USB hub block and USB function block is performed by firmware.

The compound device upstream port disconnection/reconnection procedure is as follows:

1. Upstream port is disconnected
2. USB hub block and USB function block enter suspend state, and suspend IN interrupt is generated in USB function block
3. Upstream port is reconnected
4. Upstream port reconnection is detected by means of external interrupt, etc.
5. HSRST and FSRST bits are set to 1 by firmware
6. Step 8 in initial operation procedure is executed

7 onward: Operations from step 12 onward in initial operation procedure are executed

The compound device USB function block disconnection/reconnection procedure is as follows:

1. State requiring disconnection of USB function is detected
2. Bits FNCSTP, FPLLRST, and FSRST are set to 1

If necessary, software standby mode is set

3. Detection of event enabling reconnection of USB function

Software standby mode is exited

4. If necessary, USB function control registers are re-set
5. FNCSTP bit is cleared to 0
6. FPLLRST bit is cleared to 0
7. FSRST bit is cleared to 0

8 onward: Operations from step 13 onward in initial operation procedure are executed

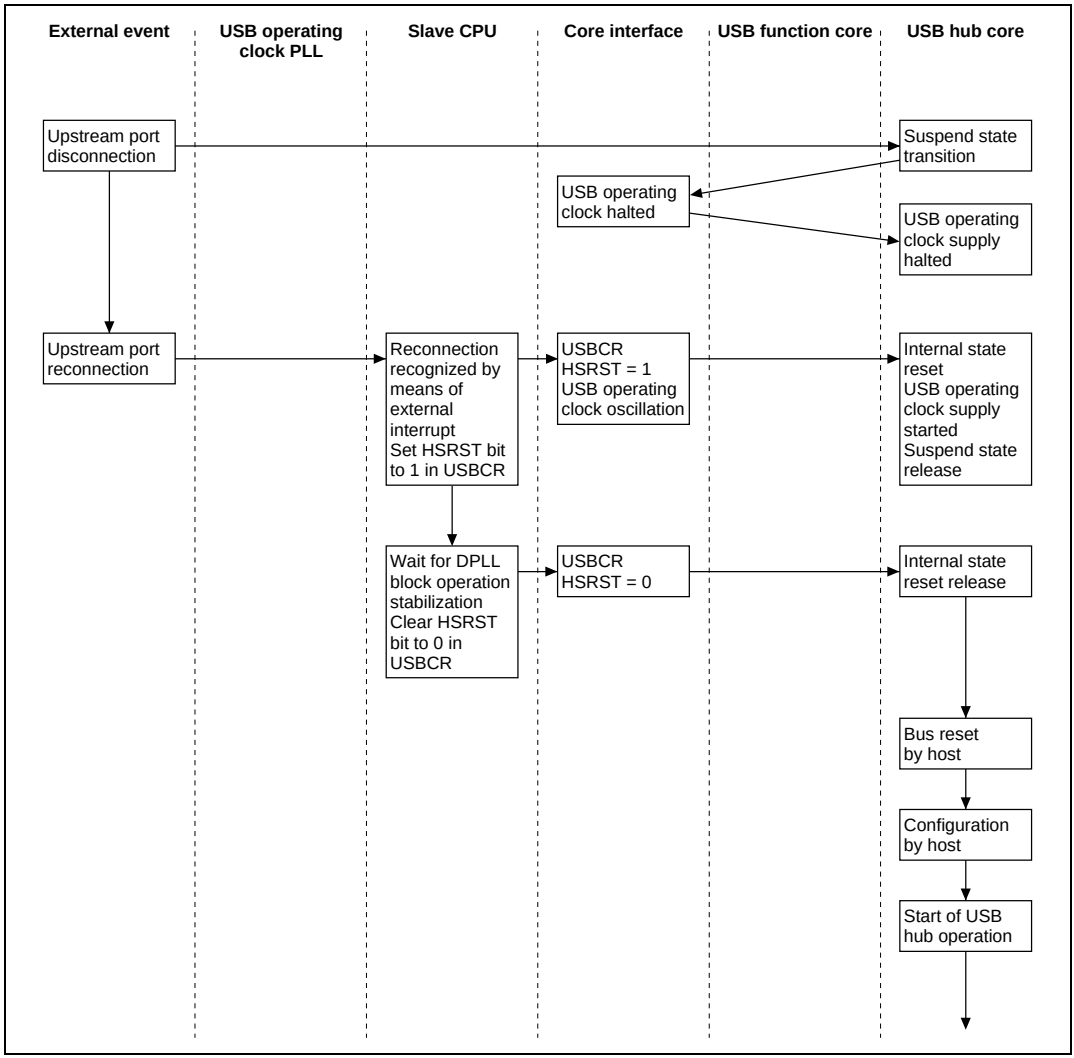


Figure 7.7 USB Hub Block Upstream Disconnection/Reconnection

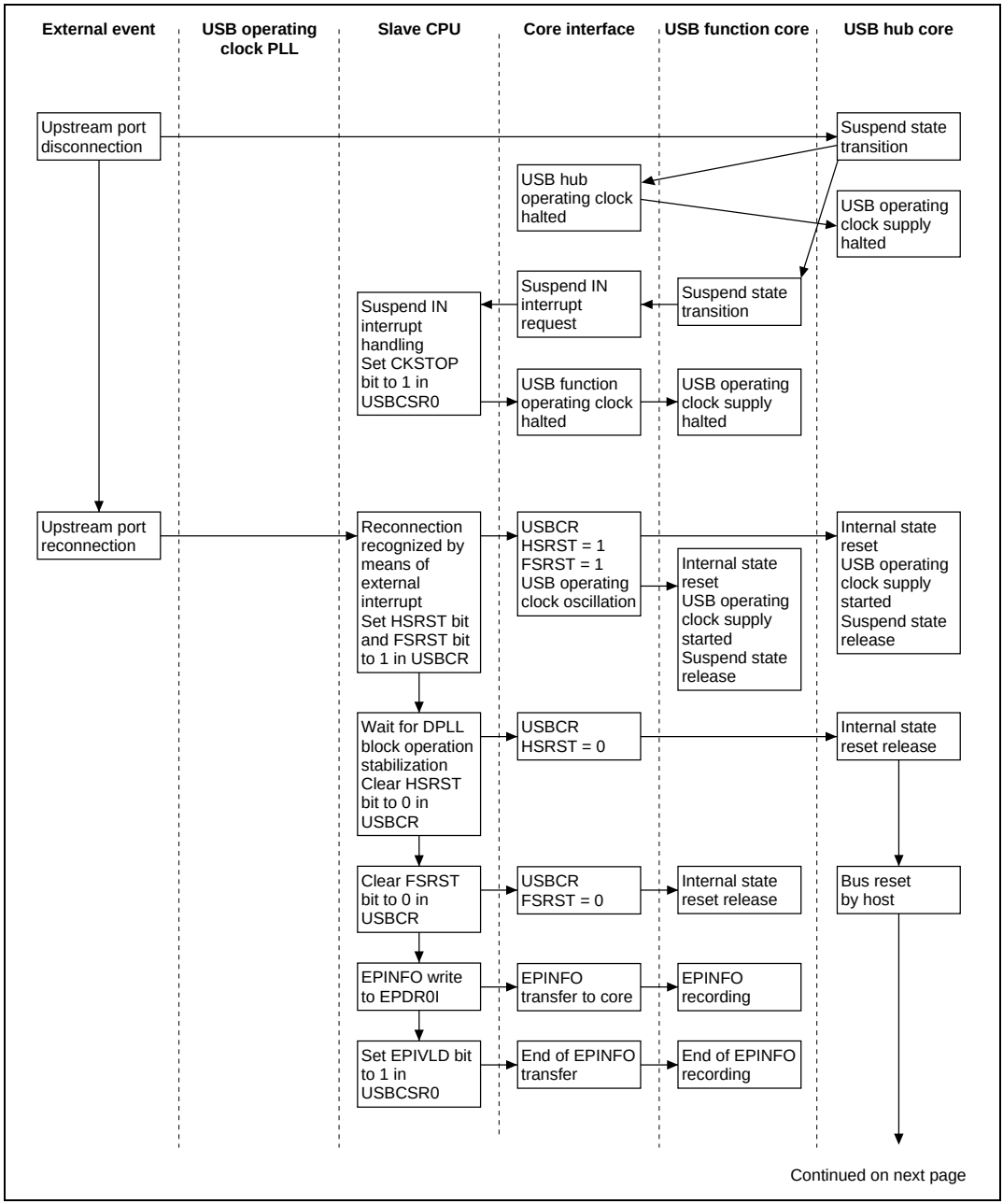


Figure 7.8 USB Compound Device Upstream Disconnection/Reconnection

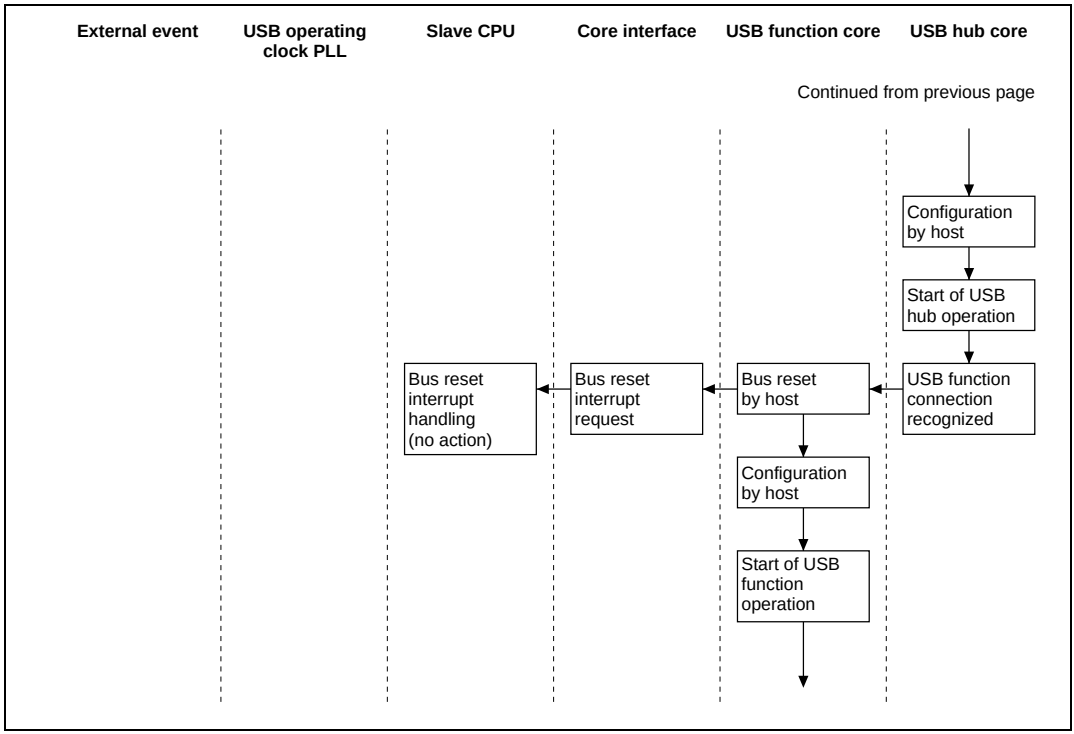


Figure 7.8 USB Compound Device Upstream Disconnection/Reconnection (cont)

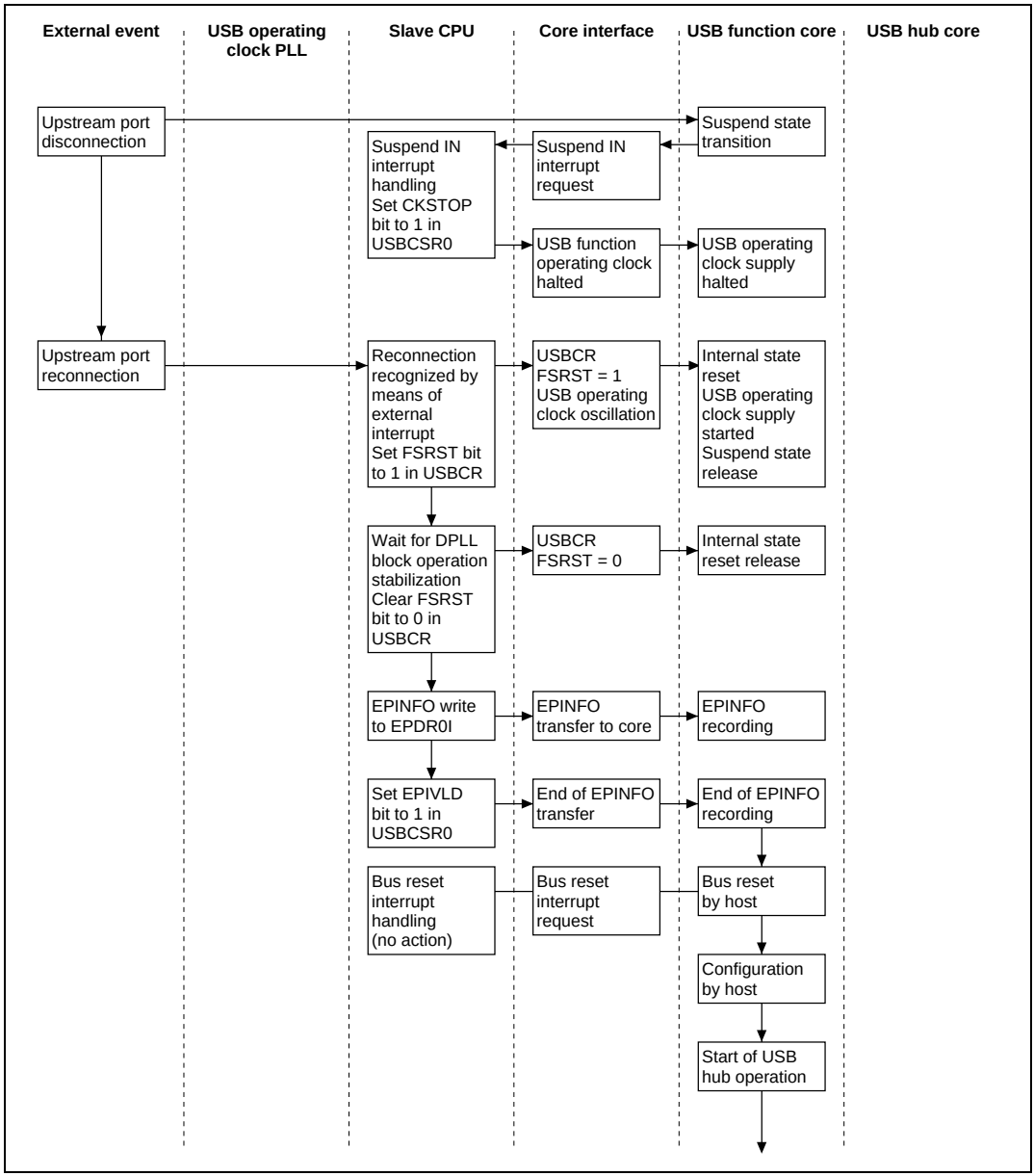


Figure 7.9 USB Function Standalone Mode Upstream Disconnection/Reconnection

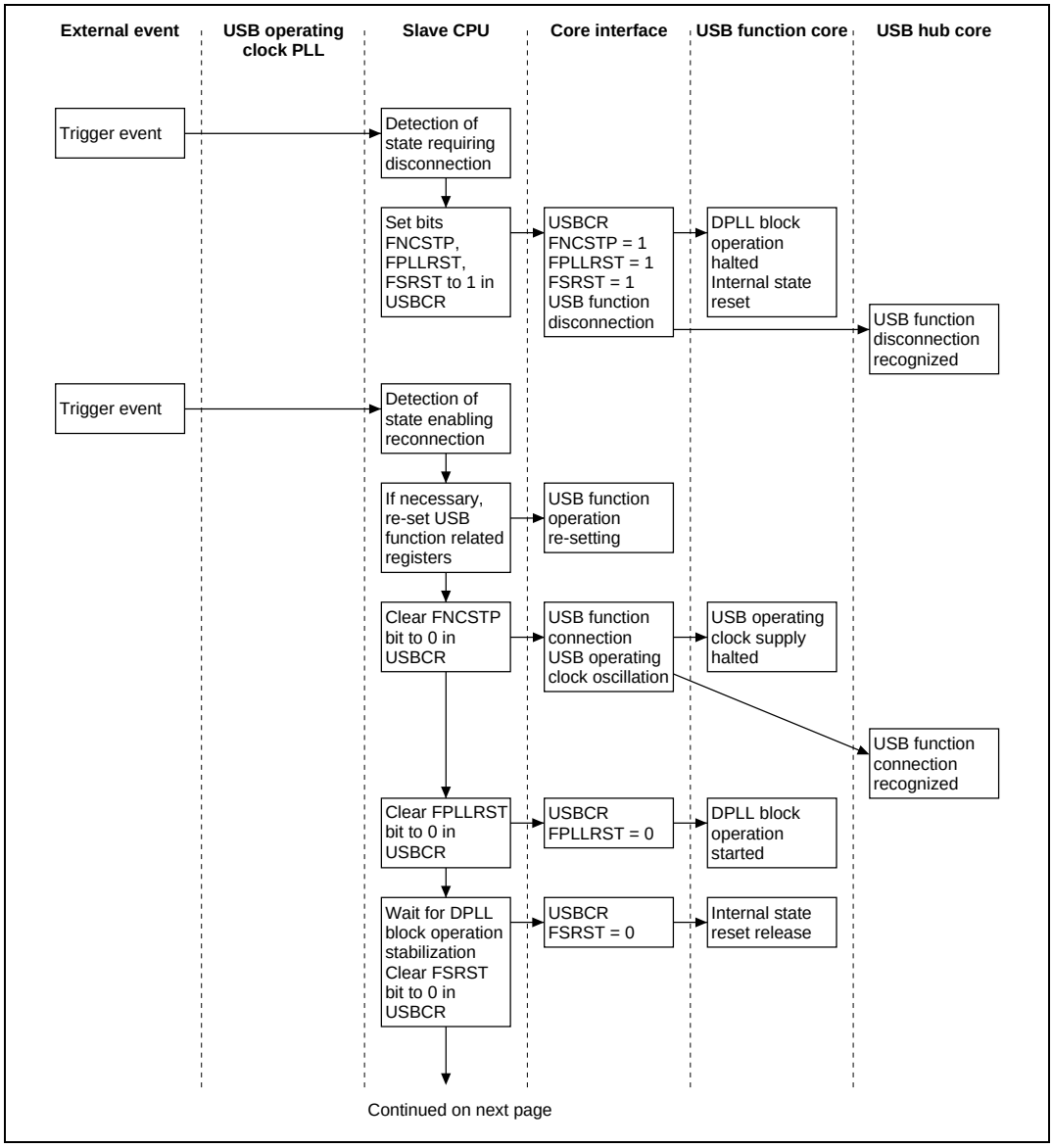


Figure 7.10 USB Function Block Disconnection/Reconnection

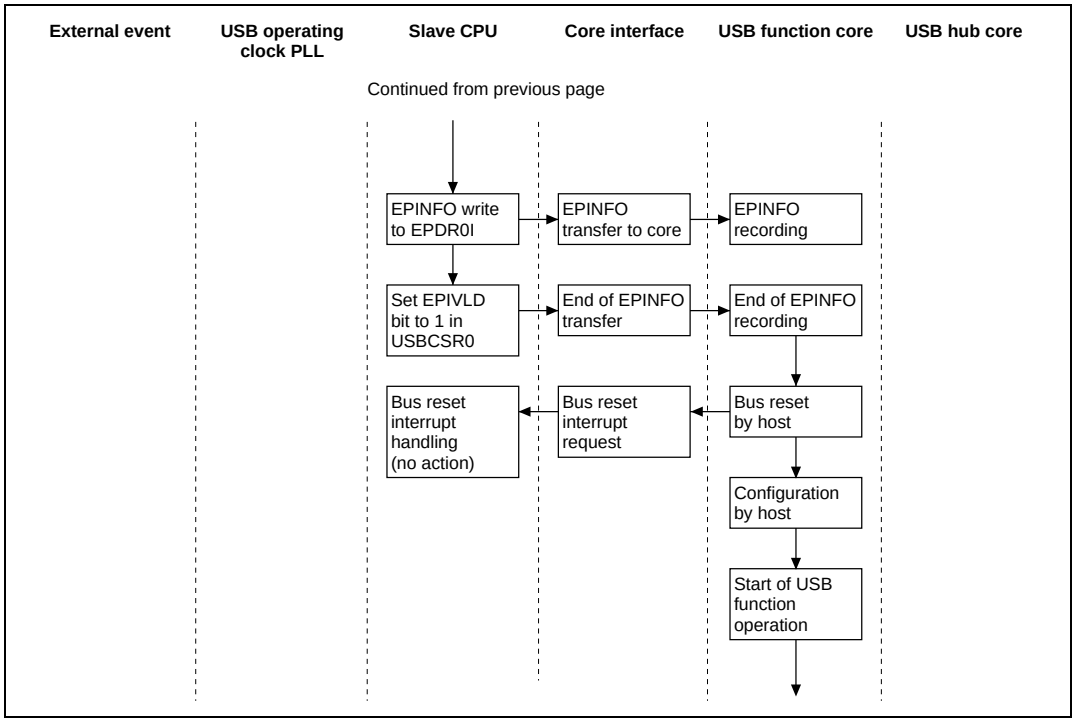


Figure 7.10 USB Function Block Disconnection/Reconnection (cont)

7.3.10 USB Module Slave CPU Interrupts

The USB module has four slave CPU interrupt sources: USBIA, USBIB, USBIC and USBID. Table 7.6 shows the interrupt sources and their priority order. The interrupt sources are the USBIFR and TSFR/TFFR interrupt flags. For each interrupt, the interrupt flag can be enabled or disabled by means of the corresponding interrupt enable bit in USBIER. In the USBID interrupt handling routine, USBIFR and TSFR/TFFR must be read to determine the interrupt source before processing is carried out.

Table 7.6 USB Interrupt Sources

Interrupt Source	Description	Priority
USBIA	Interrupt initiated by SETUP	High
USBIB	Interrupt initiated by EPTS or EPTF of endpoint specified by INTSELR0	
USBIC	Interrupt initiated by EPTS or EPTF of endpoint specified by INTSELR0	
USBID	Interrupt initiated by SOF, SPND, BRST, TS, or TF	Low

Section 8 I/O Ports

8.1 Overview

The H8/3577 Group has six input/output ports (ports 1 to 6), and one input-only port (port 7). The H8/3567 Group has four input/output ports (ports 1, 4, 5, and 6), and one input-only port. H8/3567 Group models with an on-chip USB have additional USB pins plus two input/output ports (ports C and D) for controlling the USB power supply circuit.

Table 8.1 summarizes the port functions. The pins of each port also have other functions.

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

H8/3577 Group ports 1 to 3 have a built-in MOS input pull-up function, and use DDR and a MOS input pull-up control register (PCR) to control the on/off status of the MOS input pull-ups.

Ports 1 to 6 can drive one TTL load and a 30 pF capacitive load. All the input/output ports can drive a Darlington transistor pair when in output mode.

The output type of pin $P5_2$ in port 5 and pin $P4_7$ in port 4 is NMOS push-pull.

Port C has the same load drive capacity as ports 1 to 6.

Port D also has a USB hub downstream input/output function, and operates on the USB power supply (3.3 V).

Table 8.1 H8/3577 Group and H8/3567 Group Port Functions

Port	Summary	Pins	Description
Port 1	• 8-bit I/O port • Built-in MOS input pull-up (H8/3577 Group only)	$P1_7/PW_7$ (/SCL_1) $P1_6/PW_6$ (/SDA_1) $P1_5/PW_5$ (/CBLANK) $P1_4/PW_4$ $P1_3/PW_3$ $P1_2/PW_2$ $P1_1/PW_1/PWX_1$ $P1_0/PW_0/PWX_0$	I/O port also functioning as PWM timer output pins (PW_7 to PW_0, PWX_1, PWX_0) (both H8/3577 and H8/3567 Group) Additional functions: timer connection output pin (CBLANK) and I^2C bus interface 1 I/O pins (SCL_1, SDA_1) (H8/3567 Group only)
Port 2	• 8-bit I/O port H8/3577 Group: Present H8/3567 Group: Absent • Built-in MOS input pull-up (H8/3577 Group only)	$P2_7/PW_{15}$/CBLANK $P2_6/PW_{14}$ $P2_5/PW_{13}$ $P2_4/PW_{12}/SCL_1$ $P2_3/PW_{11}/SDA_1$ $P2_2/PW_{10}$ $P2_1/PW_9$ $P2_0/PW_8$	I/O port also functioning as PWM timer output pins (PW_{15} to PW_8), or timer connection output pin (CBLANK) and I^2C bus interface 1 I/O pins (SCL_1, SDA_1)
Port 3	• 8-bit I/O port H8/3577 Group: Present H8/3567 Group: Absent • Built-in MOS input pull-up (H8/3577 Group only)	$P3_7$ to $P3_0$	I/O port
Port 4	• 8-bit I/O port	$P4_7/SDA_0$	I/O port also functioning as I^2C bus interface 0 I/O pin (SDA0)
		$P4_6/\phi$	When DDR = 0 (after reset): Input port When DDR = 1: ϕ output pin
		$P4_5$ to $P4_3$	I/O ports
		$P4_2/\overline{IRQ}_0$ $P4_1/\overline{IRQ}_1$	I/O ports also functioning as external interrupt input pins ($\overline{IRQ}_0$, $\overline{IRQ}_1$)
		$P4_0/\overline{IRQ}_2/\overline{ADTRG}$	I/O port also functioning as external interrupt input pin ($\overline{IRQ}_2$) and A/D converter external trigger input pin ($\overline{ADTRG}$)

Port	Summary	Pins	Description
Port 5	• 3-bit I/O port	$P5_2/SCK_0/SCL_0$ $P5_1/RxD_0$ $P5_0/TxD_0$	I/O port also functioning as SCI0 I/O pins (TxD_0, RxD_0, SCK_0) and I^2C bus interface 0 I/O pin (SCL_0)
Port 6	• 8-bit I/O port	$P6_7/TMOX/TMO_1/HSYNCO$ $P6_6/FTOB/TMRI_1/CSYNCI$ $P6_5/FTID/TMCI_1/HSYNCI$ $P6_4/FTIC/TMO_0/CLAMPO$ $P6_3/FTIB/TMRI_0/VFBACKI$ $P6_2/FTIA/TMIY/VSYNCI$ $P6_1/FTOA/VSYNCO$ $P6_0/FTCI/TMIX/TMCI_0/$ HFBACKI	I/O port also functioning as FRT I/O pins (FTCI, FTOA, FTIA, FTIB, FTIC, FTID, FTOB), 8-bit timer 0 and 1 I/O pins ($TMCI_0$, $TMRI_0$, TMO_0, $TMCI_1$, $TMRI_1$, TMO_1), 8-bit timer X and Y I/O pins (TMOX, TMIX, TMIY), and timer connection I/O pins (HSYNCO, CSYNCI, HSYNCI, CLAMPO, VFBACKI, VSYNCI, VSYNCO, HFBACKI)
Port 7	• 8-bit input port (H8/3577 Group) • 4-bit input port (H8/3567 Group)	$P7_7$ to $P7_4/AN_7$ to AN_4 $P7_3$ to $P7_0/AN_3$ to AN_0	I/O port also functioning as A/D converter analog inputs (AN_7 to AN_0)
Port C	• 8-bit I/O port (H8/3567 Group version with on-chip USB only)	PC_7 to $PC_4/\overline{OCP}_5$ to $\overline{OCP}_2$ PC_3 to $PC_0/\overline{ENP}_5$ to $\overline{ENP}_2$	I/O port also functioning as external power supply circuit overcurrent detection signal input pins ($\overline{OCP}_5$ to $\overline{OCP}_2$) and power output enable signal output pins ($\overline{ENP}_5$ to $\overline{ENP}_2$)
Port D	• 8-bit I/O port (H8/3567 Group version with on-chip USB only) Power supply: DrVCC (3.3 V)	PD_7/DS5D–, PD6/DS5D+, PD_5/DS4D–, PD4/DS4D+, PD_3/DS3D–, PD2/DS3D+, PD_1/DS2D–, PD0/DS2D+	I/O port also functioning as USB downstream I/O pins

8.2 Port 1

8.2.1 Overview

Port 1 is an 8-bit I/O port. Port 1 is also used for 8-bit PWM output (PW_7 to PW_0), 14-bit PWM output (PWX_1, PWX_0), timer connection output (CBLANK) [H8/3567 Group only], and IIC1 input/output (SCL_1, SDA_1) [H8/3567 Group only].

In the H8/3577 Group, port 1 has a built-in MOS input pull-up function that can be controlled by software.

Figure 8.1 shows the port 1 pin configuration.

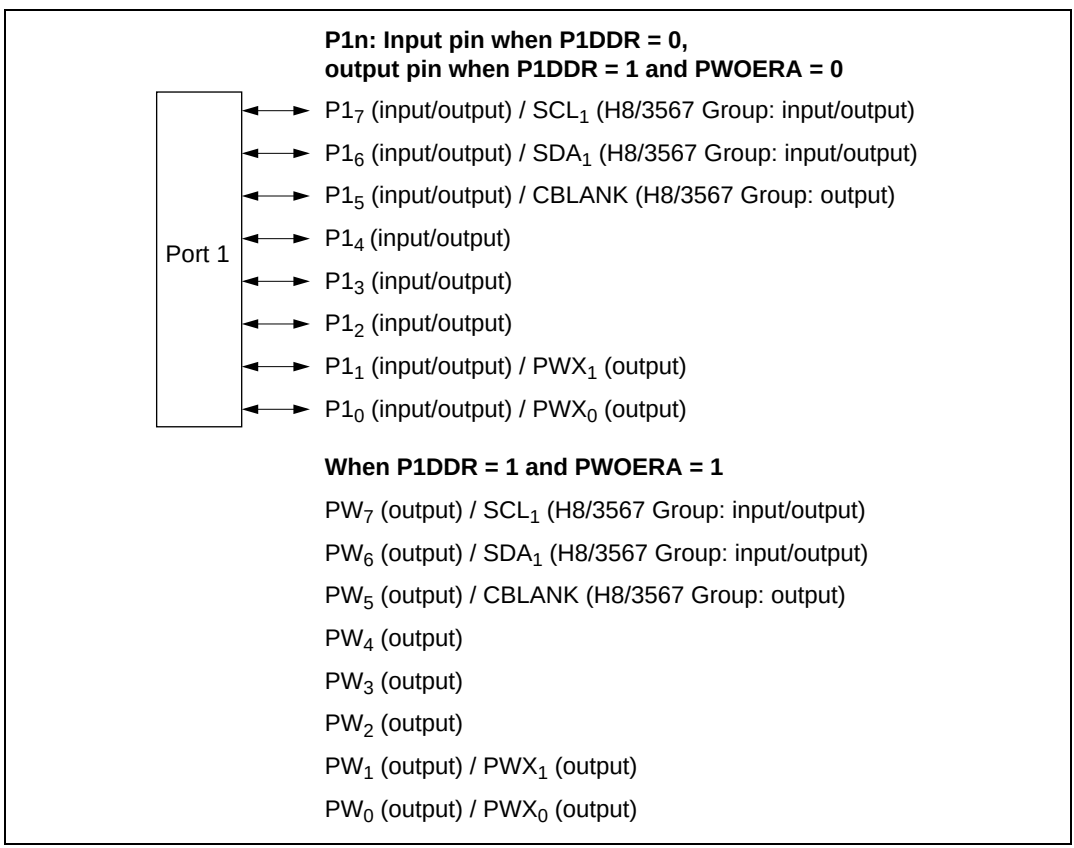


Figure 8.1 Port 1 Pin Functions

8.2.2 Register Configuration

Table 8.2 shows the port 1 register configuration.

Table 8.2 Port 1 Registers

Name	Abbreviation	R/W	Initial Value	Address
Port 1 data direction register	P1DDR	W	H'00	H'FFB0
Port 1 data register	P1DR	R/W	H'00	H'FFB2
Port 1 MOS pull-up control register* [H8/3577 Group only]	P1PCR	R/W	H'00	H'FFAC

Note: * P1PCR cannot be read or written to in the H8/3567 Group. A read will return an undefined value.

Port 1 Data Direction Register (P1DDR)

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

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

P1DDR is initialized to H'00 by a reset and in hardware standby mode. It retains its previous state in software standby mode.

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

$P1_0$ and $P1_1$ can be used for PWMX output regardless of the P1DDR settings.

In the H8/3567 Group, $P1_7$, $P1_6$, and $P1_5$ can be used for supporting function output or input/output regardless of the P1DDR settings.

Port 1 Data Register (P1DR)

Bit	7	6	5	4	3	2	1	0
	$P1_7DR$	$P1_6DR$	$P1_5DR$	$P1_4DR$	$P1_3DR$	$P1_2DR$	$P1_1DR$	$P1_0DR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1DR is an 8-bit readable/writable register that stores output data for the port 1 pins ($P1_7$ to $P1_0$). If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read directly regardless of the actual pin states. If a port 1 read is performed while P1DDR bits are cleared to 0, the pin states are read.

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

Port 1 MOS Pull-Up Control Register (P1PCR)

Bit	7	6	5	4	3	2	1	0
	$P1_7PCR$	$P1_6PCR$	$P1_5PCR$	$P1_4PCR$	$P1_3PCR$	$P1_2PCR$	$P1_1PCR$	$P1_0PCR$
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

P1PCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port 1 on a bit-by-bit basis.

When a P1DDR bit is cleared to 0 (input port setting), setting the corresponding P1PCR bit to 1 turns on the MOS input pull-up for that pin.

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

8.2.3 Pin Functions

Port 1 is used for PWM output or as an I/O port, with input or output specifiable individually for each pin. Setting a P1DDR bit to 1 makes the corresponding port 1 pin a PWM output or output port, while clearing the bit to 0 makes the pin an input port. $P1_0$ and $P1_1$ can be used for PWMX output regardless of the P1DDR settings.

In the H8/3567 Group, $P1_7$, $P1_6$, and $P1_5$ also function as IIC1 I/O pins (SCL_1, SDA_1) and the timer connection output pin (CBLANK). $P1_7$, $P1_6$, and $P1_5$ can be used for supporting function input/output regardless of the P1DDR settings.

Port 1 pin functions are shown in table 8.3.

Table 8.3 Port 1 Pin Functions

Pin	Pin Functions and Selection Method
$P1_7$/PW_7 ($/SCL_1$)	The pin function is selected as shown below by a combination of bit ICE in ICCR of IIC1 (H8/3567 Group), bit OE7 in PWOERA, and bit $P1_7$DDR.

ICE	0			1
$P1_7$DDR	0	1		—
PWOERA: OE7	—	0	1	—
Pin function	$P1_7$ input	$P1_7$ output	PW_7 output	SCL_1 I/O

$P1_6$/PW_6 ($/SDA_1$)	The pin function is selected as shown below by a combination of bit ICE in ICCR of IIC1 (H8/3567 Group), bit OE6 in PWOERA, and bit $P1_6$DDR.

ICE	0			1
$P1_6$DDR	0	1		—
PWOERA: OE6	—	0	1	—
Pin function	$P1_6$ input	$P1_6$ output	PW_6 output	SDA_1 I/O

$P1_5$/PW_5 (/CBLANK)	The pin function is selected as shown below by a combination of bit CBE in timer connection TCONR0 (H8/3567 Group), bit OE5 in PWOERA, and bit $P1_5$DDR.

CBE	0			1
$P1_5$DDR	0	1		—
PWOERA: OE5	—	0	1	—
Pin function	$P1_5$ input	$P1_5$ output	PW_5 output	CBLANK output

Pin	Pin Functions and Selection Method			
$P1_4/PW_4$	$P1_4$DDR	0	1	
	PWOERA: OE4	0	0	1
	Pin function	$P1_4$ input	$P1_4$ output	PW_4 output
$P1_3/PW_3$	$P1_3$DDR	0	1	
	PWOERA: OE3	0	0	1
	Pin function	$P1_3$ input	$P1_3$ output	PW_3 output
$P1_2/PW_2$	$P1_2$DDR	0	1	
	PWOERA: OE2	0	0	1
	Pin function	$P1_2$ input	$P1_2$ output	PW_2 output

$P1_1/PW_1/PWX_1$

The pin function is selected as shown below by a combination of bit OEB in DACR of PWMX, bit OE1 in PWOERA, and bit $P1_1$DDR.

DACR: OEB	0			1
$P1_1$DDR	0	1		—
PWOERA: OE1	—	0	1	—
Pin function	$P1_1$ input	$P1_1$ output	PW_1 output	PWX_1 output

$P1_0/PW_0/PWX_0$

The pin function is selected as shown below by a combination of bit OEA in DACR of PWMX, bit OE0 in PWOERA, and bit $P1_0$DDR.

DACR: OEA	0			1
$P1_0$DDR	0	1		—
PWOERA: OE0	—	0	1	—
Pin function	$P1_0$ input	$P1_0$ output	PW_0 output	PWX_0 output

8.2.4 MOS Input Pull-Up Function

In the H8/3577 Group, port 1 has a built-in MOS input pull-up function that can be controlled by software.

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

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

Table 8.4 summarizes the MOS input pull-up states.

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

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Off	Off	On/Off	On/Off

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

8.3 Port 2 [H8/3577 Group Only]

8.3.1 Overview

Port 2 is an 8-bit I/O port. Port 2 is also used for 8-bit PWM output (PW_{15} to PW_8), timer connection output (CBLANK), and IIC1 input/output (SCL_1, SDA_1).

Port 2 is provided in the H8/3577 Group, but not in the H8/3567 Group. Therefore the H8/3567 Group does not have the port 2 I/O pin functions or eight 8-bit PWM output pin (PW_{15} to PW_8) functions, and provides the timer connection output pin (CBLANK) function and IIC1 I/O pin (SCL_1, SDA_1) functions in port 1.

Port 2 has a built-in MOS input pull-up function that can be controlled by software.

Figure 8.2 shows the port 2 pin configuration.

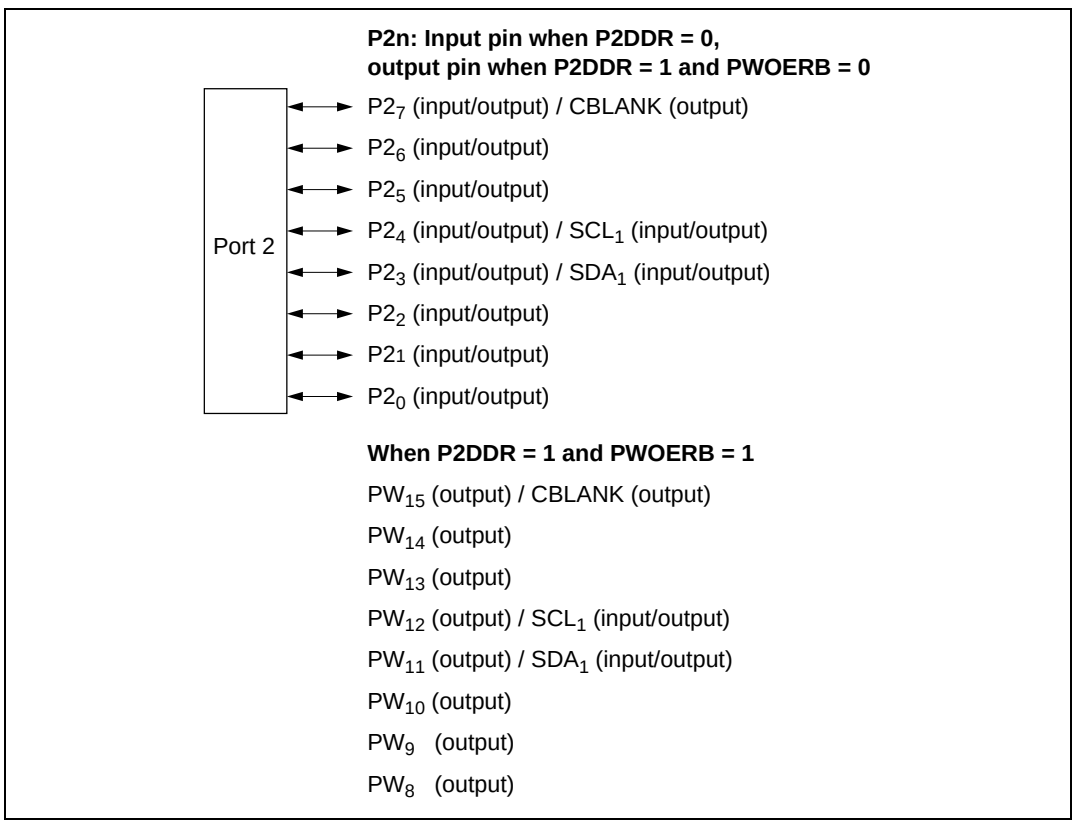


Figure 8.2 Port 2 Pin Functions

8.3.2 Register Configuration

Table 8.5 shows the port 2 register configuration.

Table 8.5 Port 2 Registers

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

Port 2 Data Direction Register (P2DDR)

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

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

P2DDR is initialized to H'00 by a reset and in hardware standby mode. It retains its previous state in software standby mode.

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

$P2_3$, $P2_4$, and $P2_7$ can be used for supporting function output regardless of the P2DDR settings.

Port 2 Data Register (P2DR)

Bit	7	6	5	4	3	2	1	0
	$P2_7DR$	$P2_6DR$	$P2_5DR$	$P2_4DR$	$P2_3DR$	$P2_2DR$	$P2_1DR$	$P2_0DR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2DR is an 8-bit readable/writable register that stores output data for the port 2 pins ($P2_7$ to $P2_0$). If a port 2 read is performed while P2DDR bits are set to 1, the P2DR values are read directly regardless of the actual pin states. If a port 2 read is performed while P2DDR bits are cleared to 0, the pin states are read.

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

Port 2 MOS Pull-Up Control Register (P2PCR)

Bit	7	6	5	4	3	2	1	0
	$P2_7PCR$	$P2_6PCR$	$P2_5PCR$	$P2_4PCR$	$P2_3PCR$	$P2_2PCR$	$P2_1PCR$	$P2_0PCR$
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

P2PCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port 2 on a bit-by-bit basis.

When a P2DDR bit is cleared to 0 (input port setting), setting the corresponding P2PCR bit to 1 turns on the MOS input pull-up for that pin.

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

8.3.3 Pin Functions

Port 2 is used for PWM output, timer connection output (CBLANK), and IIC1 input/output (SCL1, SDA1), or as an I/O port, with input or output specifiable individually for each pin. Setting a P2DDR bit to 1 makes the corresponding port 2 pin a PWM output or output port, while clearing the bit to 0 makes the pin an input port. $P2_3$, $P2_4$, and $P2_7$ can be used for supporting function output regardless of the P2DDR settings.

Port 2 pin functions are shown in table 8.6.

Table 8.6 Port 2 Pin Functions

Pin | **Pin Functions and Selection Method**

$P2_7/PW_{15}$/ CBLANK

The pin function is selected as shown below by a combination of bit CBE in timer connection TCONR0, bit OE15 in PWOERB, and bit $P2_7$DDR.

CBE	0			1
$P2_7$DDR	0	1		—
PWOERB: OE15	—	0	1	—
Pin function	$P2_7$ input	$P2_7$ output	PW_{15} output	CBLANK output

$P2_6/PW_{14}$

$P2_6$DDR	0	1	
PWOERB: OE14	0	0	1
Pin function	$P2_6$ input	$P2_6$ output	PW_{14} output

$P2_5/PW_{13}$

$P2_5$DDR	0	1	
PWOERB: OE13	0	0	1
Pin function	$P2_5$ input	$P2_5$ output	PW_{13} output

$P2_4/PW_{12}$/ SCL_1

The pin function is selected as shown below by a combination of bit ICE in ICCR of IIC1, bit OE12 in PWOERB, and bit $P2_4$DDR.

ICE	0			1
$P2_4$DDR	0	1		—
PWOERB: OE12	—	0	1	—
Pin function	$P2_4$ input	$P2_4$ output	PW_{12} output	SCL_1 I/O

Pin	Pin Functions and Selection Method

$P2_3/PW_{11}/SDA_1$

The pin function is selected as shown below by a combination of bit ICE in ICCR of IIC1, bit OE11 in PWOERB, and bit $P2_3$DDR.

ICE	0			1
$P2_3$DDR	0	1		—
PWOERB: OE11	—	0	1	—
Pin function	$P2_3$ input	$P2_3$ output	PW_{11} output	SDA_1 I/O

$P2_2/PW_{10}$

$P2_2$DDR	0	1	
PWOERB: OE10	0	0	1
Pin function	$P2_2$ input	$P2_2$ output	PW_{10} output

$P2_1/PW_9$

$P2_1$DDR	0	1	
PWOERB: OE9	0	0	1
Pin function	$P2_1$ input	$P2_1$ output	PW_9 output

$P2_0/PW_8$

$P2_0$DDR	0	1	
PWOERB: OE8	0	0	1
Pin function	$P2_0$ input	$P2_0$ output	PW_8 output

8.3.4 MOS Input Pull-Up Function

Port 2 has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be specified as on or off for individual bits.

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

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

Table 8.7 summarizes the MOS input pull-up states.

Table 8.7 MOS Input Pull-Up States (Port 2)

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Off	Off	On/Off	On/Off

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

8.4 Port 3 [H8/3577 Group Only]

8.4.1 Overview

Port 3 is an 8-bit I/O port.

Port 3 is provided in the H8/3577 Group, but not in the H8/3567 Group.

Port 3 has a built-in MOS input pull-up function that can be controlled by software.

Figure 8.3 shows the port 3 pin configuration.

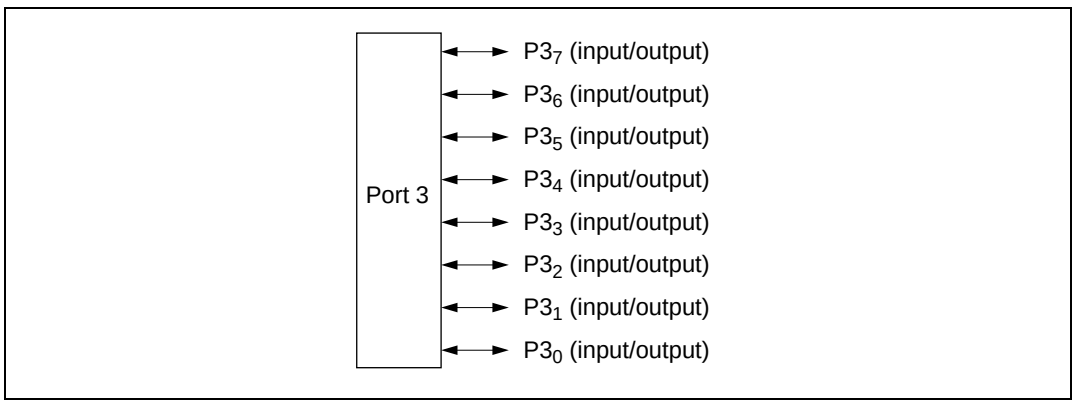


Figure 8.3 Port 3 Pin Functions

8.4.2 Register Configuration

Table 8.8 shows the port 3 register configuration.

Table 8.8 Port 3 Registers

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

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

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

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

Port 3 Data Register (P3DR)

Bit	7	6	5	4	3	2	1	0
	$P3_7DR$	$P3_6DR$	$P3_5DR$	$P3_4DR$	$P3_3DR$	$P3_2DR$	$P3_1DR$	$P3_0DR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3DR is an 8-bit readable/writable register that stores output data for the port 3 pins ($P3_7$ to $P3_0$). If a port 3 read is performed while P3DDR bits are set to 1, the P3DR values are read directly regardless of the actual pin states. If a port 3 read is performed while P3DDR bits are cleared to 0, the pin states are read.

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

Port 3 MOS Pull-Up Control Register (P3PCR)

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

P3PCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port 3 on a bit-by-bit basis.

When a P3DDR bit is cleared to 0 (input port setting), setting the corresponding P3PCR bit to 1 turns on the MOS input pull-up for that pin.

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

8.4.3 Pin Functions

Port 3 is used as an I/O port, with input or output specifiable individually for each pin. Setting a P3DDR bit to 1 makes the corresponding port 3 pin an output port, while clearing the bit to 0 makes the pin an input port.

8.4.4 MOS Input Pull-Up Function

Port 3 has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be specified as on or off for individual bits.

When a P3DDR bit is cleared to 0, setting the corresponding P3PCR bit to 1 turns on the MOS input pull-up for that pin.

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

Table 8.9 summarizes the MOS input pull-up states.

Table 8.9 MOS Input Pull-Up States (Port 3)

Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Off	Off	On/Off	On/Off

Legend:
Off: MOS input pull-up is always off.
On/Off: On when P3DDR = 0 and P3PCR = 1; otherwise off.

8.5 Port 4

8.5.1 Overview

Port 4 is an 8-bit I/O port. Port 4 is also used for external interrupt input ($\overline{IRQ}_0$ to $\overline{IRQ}_2$), A/D converter input ($\overline{ADTRG}$), IIC0 input/output (SDA0), and system clock (ϕ) output. The output type of $P4_7$ is NMOS push-pull. The output type of SDA0 is NMOS open-drain with direct bus drive capability.

Figure 8.4 shows the port 4 pin configuration.

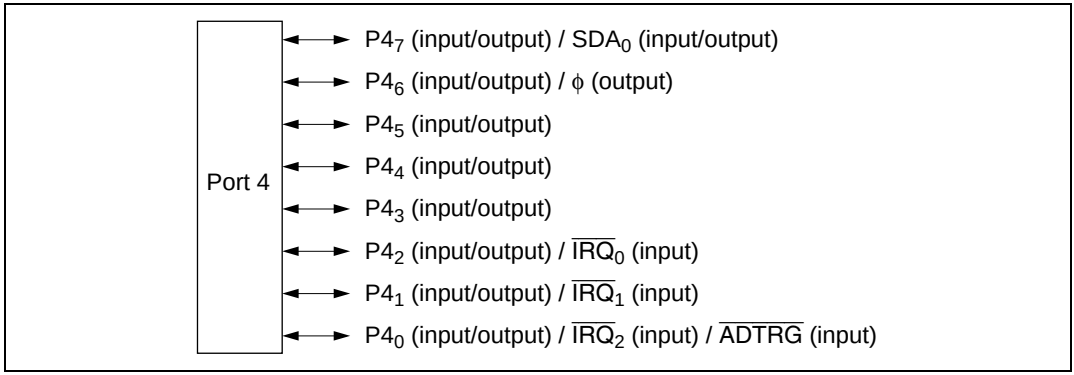


Figure 8.4 Port 4 Pin Functions

8.5.2 Register Configuration

Table 8.10 shows the port 4 register configuration.

Table 8.10 Port 4 Registers

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

P4DDR is initialized to H'00 by a reset and in hardware standby mode. It retains its previous state in software standby mode.

When P4DDR bits are set to 1, pin $P4_6$ functions as the ϕ output pin, and pins $P4_7$ and $P4_5$ to $P4_0$ function as output ports. Clearing a P4DDR bit to 0 makes the corresponding pin an input port.

Port 4 Data Register (P4DR)

Bit	7	6	5	4	3	2	1	0
	$P4_7DR$	$P4_6DR$	$P4_5DR$	$P4_4DR$	$P4_3DR$	$P4_2DR$	$P4_1DR$	$P4_0DR$
Initial value	0	—*	0	0	0	0	0	0
Read/Write	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the state of pin $P4_6$.

P4DR is an 8-bit readable/writable register that stores output data for the port 4 pins ($P4_7$ to $P4_0$). Except for $P4_6$, if a port 4 read is performed while P4DDR bits are set to 1, the P4DR values are read directly regardless of the actual pin states. If a port 4 read is performed while P4DDR bits are cleared to 0, the pin states are read.

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

8.5.3 Pin Functions

Port 4 pins are also used for external interrupt input ($\overline{IRQ_0}$ to $\overline{IRQ_2}$), A/D converter input ($\overline{ADTRG}$), IIC0 input/output (SDA_0), and system clock (ϕ) output.

Port 4 pin functions are shown in table 8.11.

Table 8.11 Port 4 Pin Functions

Pin	Pin Functions and Selection Method
$P4_7/SDA_0$	The pin function is selected as shown below by a combination of bit ICE in ICCR of IIC0 and bit $P4_7DDR$.

ICE	0		1
$P4_7DDR$	0	1	—
Pin function	$P4_7$ input	$P4_7$ output	SDA_0 I/O

When this pin is designated as the $P4_7$ output pin, it is an NMOS push-pull output. The output type of SDA0 is NMOS open-drain with direct bus drive capability.

$P4_6/\phi$			
	$P4_6DDR$	0	1
	Pin function	$P4_6$ input	ϕ output

$P4_5$			
	$P4_5DDR$	0	1
	Pin function	$P4_5$ input	$P4_5$ output

$P4_4$			
	$P4_4DDR$	0	1
	Pin function	$P4_4$ input	$P4_4$ output

$P4_3$			
	$P4_3DDR$	0	1
	Pin function	$P4_3$ input	$P4_3$ output

$P4_2/\overline{IRQ_0}$			
	$P4_2DDR$	0	1
	Pin function	$P4_2$ input	$P4_2$ output
		$\overline{IRQ_0}$ input	

When bit IRQ0E is set to 1 in IER, this pin is used as the $\overline{IRQ_0}$ input pin.

Pin	Pin Functions and Selection Method		
$P4_1/\overline{IRQ}_1$	$P4_1$DDR	0	1
	Pin function	$P4_1$ input	$P4_1$ output
		$\overline{IRQ}_1$ input	
	When bit IRQ1E is set to 1 in IER, this pin is used as the $\overline{IRQ}_1$ input pin.		
$P4_0/\overline{IRQ}_2/\overline{ADTRG}$	$P4_0$DDR	0	1
	Pin function	$P4_0$ input	$P4_0$ output
		$\overline{IRQ}_2$ input, $\overline{ADTRG}$ input	
	When bit IRQ2E is set to 1 in IER, this pin is used as the $\overline{IRQ}_2$ input pin. When bits TRGS1 and TRGS0 are both set to 1 in the A/D converter's ADCR register, this pin is used as the $\overline{ADTRG}$ input pin.		

8.6 Port 5

8.6.1 Overview

Port 5 is a 3-bit I/O port. Port 5 is also used for SCI0 input/output (TxD_0, RxD_0, SCK_0) and IIC0 input/output (SCL_0). The output type of $P5_2$ and SCK_0 is NMOS push-pull. The output type of SCL_0 is NMOS open-drain.

Figure 8.5 shows the port 5 pin configuration.

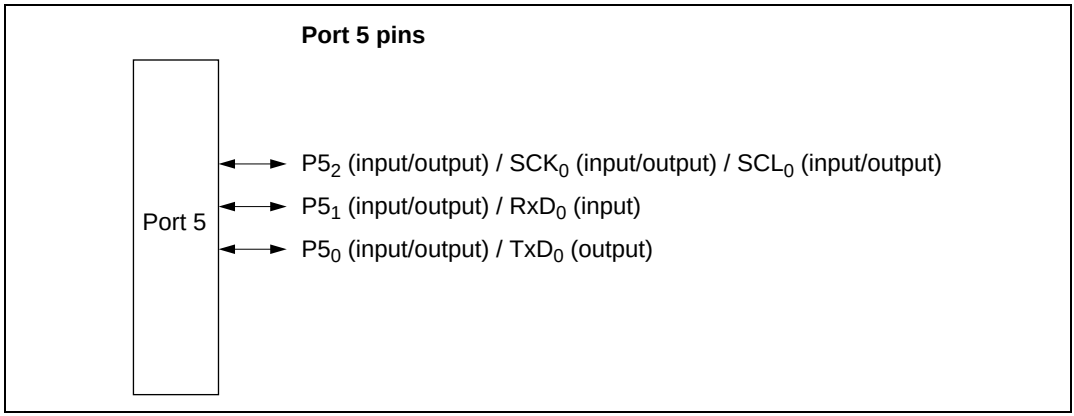


Figure 8.5 Port 5 Pin Functions

8.6.2 Register Configuration

Table 8.12 shows the port 5 register configuration.

Table 8.12 Port 5 Registers

Name	Abbreviation	R/W	Initial Value	Address
Port 5 data direction register	P5DDR	W	H'F8	H'FFB8
Port 5 data register	P5DR	R/W	H'F8	H'FFBA

Port 5 Data Direction Register (P5DDR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	$P5_2DDR$	$P5_1DDR$	$P5_0DDR$
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	W	W	W

P5DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 5. P5DDR cannot be read; if it is, an undefined value will be returned. Bits 7 to 3 are reserved.

Setting a P5DDR bit to 1 makes the corresponding port 5 pin an output port, while clearing the bit to 0 makes the pin an input port.

P5DDR is initialized to H'F8 by a reset and in hardware standby mode. It retains its previous state in software standby mode. As SCI0 is initialized, the pin states are determined by IIC0's ICCR, P5DDR, and P5DR specifications.

Port 5 Data Register (P5DR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	$P5_2DR$	$P5_1DR$	$P5_0DR$
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

P5DR is an 8-bit readable/writable register that stores output data for the port 5 pins ($P5_2$ to $P5_0$). If a port 5 read is performed while P5DDR bits are set to 1, the P5DR values are read directly regardless of the actual pin states. If a port 5 read is performed while P5DDR bits are cleared to 0, the pin states are read.

Bits 7 to 3 are reserved; they cannot be modified and are always read as 1.

P5DR is initialized to H'F8 by a reset and in hardware standby mode. It retains its previous state in software standby mode.

8.6.3 Pin Functions

Port 5 pins are also used for SCI0 input/output (TxD_0, RxD_0, SCK_0) and IIC0 input/output (SCL_0).

Port 5 pin functions are shown in table 8.13.

Table 8.13 Port 5 Pin Functions

Pin	Pin Functions and Selection Method
$P5_2/SCK_0/SCL_0$	The pin function is selected as shown below by a combination of bit C/$\overline{A}$ in SMR of SCI0, bits CKE0 and CKE1 in SCR, bit ICE in ICCR of IIC0, and bit $P5_2$DDR.

ICE	0				1
CKE1	0			1	0
C/$\overline{A}$	0		1	—	0
CKE0	0	1	—	—	0
$P5_2$DDR	0	1	—	—	—
Pin function	$P5_2$ input	$P5_2$ output	SCK_0 output	SCK_0 output	SCK_0 input	SCL_0 I/O

When this pin is used as the SCL_0 I/O pin, bits CKE1 and CKE0 in SCR of SCI0 and bit C/$\overline{A}$ in SMR must all be cleared to 0. The output type of SCL_0 is NMOS open-drain with direct bus drive capability.

When this pin is designated as the $P5_2$ output pin or SCK_0 output pin, it is an NMOS push-pull output.

$P5_1/RxD_0$ — The pin function is selected as shown below by a combination of bit RE in SCR of SCI0 and bit $P5_1$DDR.

RE	0		1
$P5_1$DDR	0	1	—
Pin function	$P5_1$ input	$P5_1$ output	RxD_0 input

$P5_0/TxD_0$ — The pin function is selected as shown below by a combination of bit TE in SCR of SCI0 and bit $P5_0$DDR.

TE	0		1
$P5_0$DDR	0	1	—
Pin function	$P5_0$ input	$P5_0$ output	TxD_0 output

8.7 Port 6

8.7.1 Overview

Port 6 is an 8-bit I/O port. It is also used for 16-bit free-running timer (FRT) input/output (FTOA, FTOB, FTIA to FTID, FTCI), timer 0 and 1 (TMR_0, TMR_1) input/output ($TMCI_0$, $TMRI_0$, TMO_0, $TMCI_1$, $TMRI_1$, TMO_1), timer X (TMRX) input/output (TMOX, TMIX), timer Y (TMRY) input (TMIY), and timer connection input/output (CSYNCI, HSYNCI, HSYNCO, HFBACKI, VSYNCI, VSYNCO, VFBACKI, CLAMPO).

Figure 8.6 shows the port 6 pin configuration.

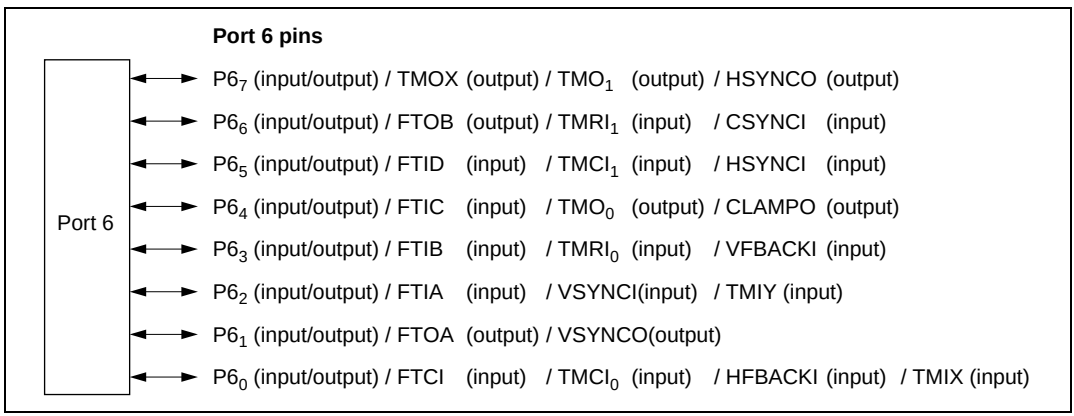


Figure 8.6 Port 6 Pin Functions

8.7.2 Register Configuration

Table 8.14 shows the port 6 register configuration.

Table 8.14 Port 6 Registers

Name	Abbreviation	R/W	Initial Value	Address
Port 6 data direction register	P6DDR	W	H'00	H'FFB9
Port 6 data register	P6DR	R/W	H'00	H'FFBB

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 write-only register, the individual bits of which specify input or output for the pins of port 6. P6DDR cannot be read; if it is, an undefined value will be returned.

Setting a P6DDR bit to 1 makes the corresponding port 6 pin an output port, while clearing the bit to 0 makes the pin an input port.

P6DDR is initialized to H'00 by a reset and in hardware standby mode. It retains its previous state in software standby mode.

Port 6 Data Register (P6DR)

Bit	7	6	5	4	3	2	1	0
	$P6_7DR$	$P6_6DR$	$P6_5DR$	$P6_4DR$	$P6_3DR$	$P6_2DR$	$P6_1DR$	$P6_0DR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P6DR is an 8-bit readable/writable register that stores output data for the port 6 pins ($P6_7$ to $P6_0$). If a port 6 read is performed while P6DDR bits are set to 1, the P6DR values are read directly regardless of the actual pin states. If a port 6 read is performed while P6DDR bits are cleared to 0, the pin states are read.

P6DR is initialized to H'00 by a reset and in hardware standby mode. It retains its previous state in software standby mode.

8.7.3 Pin Functions

Port 6 pins are also used for 16-bit free-running timer (FRT) input/output (FTOA, FTOB, FTIA to FTID, FTCI), timer 0 and 1 (TMR_0, TMR_1) input/output ($TMCI_0$, $TMRI_0$, TMO_0, $TMCI_1$, $TMRI_1$, TMO_1), timer X (TMRX) input/output (TMOX, TMIX), timer Y (TMRY) input (TMIY), and timer connection input/output (CSYNCI, HSYNCI, HSYNCO, HFBACKI, VSYNCI, VSYNCO, VFBACKI, CLAMPO.

Port 6 pin functions are shown in table 8.15.

Table 8.15 Port 6 Pin Functions

Pin	Pin Functions and Selection Method
$P6_7$/TMO_1/ TMOX/ HSYNCO	The pin function is selected as shown below by a combination of bits OS3 to OS0 in TCSR of TMR1 and TMRX, bit HOE of timer connection TCONRO, and bit $P6_7$DDR.

HOE	0				1
TMRX: OS3–0	All 0			Not all 0	—
TMR1: OS3–0	All 0		Not all 0	—	—
$P6_7$DDR	0	1	—	—	—
Pin function	$P6_7$ input	$P6_7$ output	TMO_1 output	TMOX output	HSYNCO output

$P6_6$/FTOB/ $TMRI_1$/CSYNCI	The pin function is selected as shown below by a combination of bit OEB in TOCR of FRT and bit $P6_6$DDR.

OEB	0		1
$P6_6$DDR	0	1	—
Pin function	$P6_6$ input	$P6_6$ output	FTOB output
	$TMRI_1$ input, CSYNCI input		

When bits CCLR1 and CCLR0 are both set to 1 in TCR of TMR1, this pin is used as the $TMRI_1$ input pin.

$P6_5$/FTID/ $TMCI_1$/HSYNCI	$P6_5$DDR	0	1
	Pin function	$P6_5$ input	$P6_5$ output
		FTID input, $TMCI_1$ input, HSYNCI input	

When an external clock is selected with bits CKS2 to CKS0 in TCR of TMR_1, this pin is used as the $TMCI_1$ input pin.

$P6_4$/FTIC/ TMO_0/ CLAMPO	The pin function is selected as shown below by a combination of bits OS3 to OS0 in TCSR of TMR0, bit CLOE of timer connection TCONRO, and bit $P6_4$DDR.

CLOE	0			1
OS3–0	All 0		Not all 0	All 0
$P6_4$DDR	0	1	—	—
Pin function	$P6_4$ input	$P6_4$ output	TMO_0 output	CLAMPO output
	FTIC input			

When this pin is used as the CLAMPO pin, bits OS3 to OS0 in TCSR of TMR_0 must be cleared to 0.

Pin	Pin Functions and Selection Method

$P6_3$/FTIB/ TMRI0/VFBACKI

$P6_3$DDR	0	1
Pin function	$P6_3$ input	$P6_3$ output
	FTIB input, TMRI0 input, VFBACKI input	

When bits CCLR1 and CCLR0 are both set to 1 in TCR of TMR0, this pin is used as the $TMRI_0$ input pin.

$P6_2$/FTIA/ VSYNCI/TMIY

$P6_2$DDR	0	1
Pin function	$P6_2$ input	$P6_2$ output
	FTIA input, VSYNCI input, TMIY input	

$P6_1$/FTOA/ VSYNCO

The pin function is selected as shown below by a combination of bit OEA in TOCR of FRT, bit VOE of timer connection TCONRO, and bit $P6_1$DDR.

VOE	0			1
OEA	0		1	0
$P6_1$DDR	0	1	—	—
Pin function	$P6_1$ input	$P6_1$ output	$FTOA_0$ output	VSYNCO output

When this pin is used as the VSYNCO pin, bit OEA in TOCR of FRT must be cleared to 0.

$P6_0$/FTCI/$TMCI_0$/ HFBACKI/TMIX

$P6_0$DDR	0	1
Pin function	$P6_0$ input	$P6_0$ output
	FTCI input, $TMCI_0$ input, HFBACKI input, TMIX input	

When an external clock is selected with bits CKS1 and CKS0 in TCR of FRT, this pin is used as the FTCI input pin.

When an external clock is selected with bits CKS2 to CKS0 in TCR of TMR0, this pin is used as the $TMCI_0$ input pin.

8.8 Port 7

8.8.1 Overview

Port 7 is an 8-bit input port. Port 7 is also used for A/D converter analog input (AN_7 to AN_0).

Bits 7 to 4 of port 7 are provided in the H8/3577 Group, but not in the H8/3567 Group. Therefore the H8/3567 Group does not have the input pin functions or four A/D converter analog input pin (AN_7 to AN_4) functions corresponding to bits 7 to 4 of port 7.

Figure 8.7 shows the port 7 pin configuration.

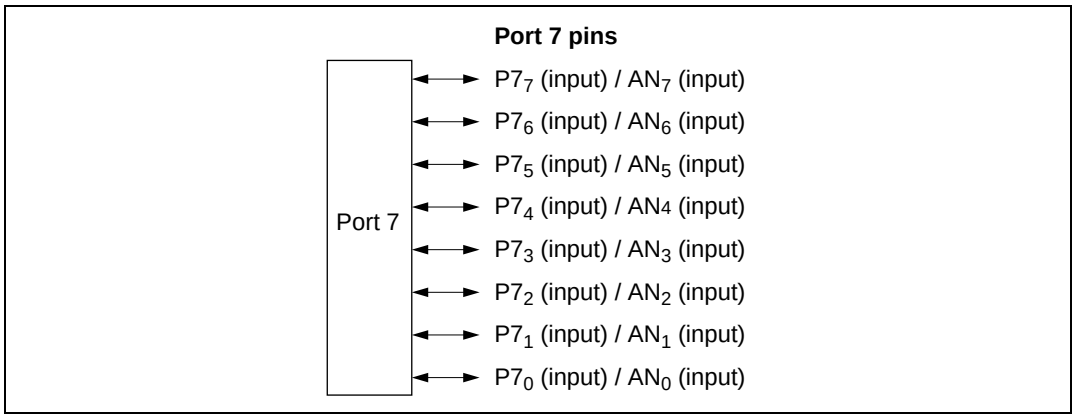


Figure 8.7 Port 7 Pin Functions

8.8.2 Register Configuration

Table 8.16 shows the port 7 register configuration. As port 7 is an input port, it has no data direction register or data register.

Table 8.16 Port 7 Registers

Name	Abbreviation	R/W	Initial Value	Address
Port 7 input data register	P7PIN	R	Undefined	H'FFBE

Port 7 Input Data Register (P7PIN)

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

Note: * Determined by the state of pins $P7_7$ to $P7_0$.

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

In the H8/3567 Group, reading bits 7 to 4 will return an undefined value.

8.8.3 Pin Functions

Port 7 pins are also used for A/D converter analog input (AN_7 to AN_0).

In the H8/3567 Group, the port 7 pins ($P7_0$ to $P7_3$) are also used for A/D converter analog input (AN_0 to AN_3).

8.9 Port C [H8/3567 Group Version with On-Chip USB Only]

8.9.1 Overview

Port C is an 8-bit I/O port.

Port C is provided only in the H8/3567 Group version with an on-chip USB.

Port C is also used for input/output to control the USB hub downstream port power supply IC.

Figure 8.8 shows the port C pin configuration.

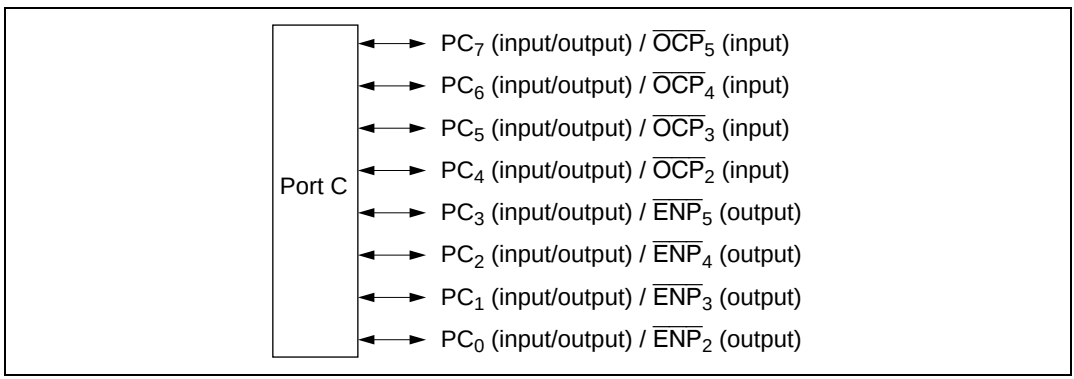


Figure 8.8 Port C Pin Functions

8.9.2 Register Configuration

Table 8.17 shows the port C register configuration.

Table 8.17 Port C Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port C data direction register	PCDDR	W	H'00	H'FE4E
Port C output data register	PCODR	R/W	H'00	H'FE4C
Port C input data register	PCPIN	R	Undefined	H'FE4E

Note: * PCPIN and PCDDR have the same address.

Port C Data Direction Register (PCDDR)

Bit	7	6	5	4	3	2	1	0
	PC_7DDR	PC_6DDR	PC_5DDR	PC_4DDR	PC_3DDR	PC_2DDR	PC_1DDR	PC_0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

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

Setting a PCDDR bit to 1 makes the corresponding port C pin an output port, while clearing the bit to 0 makes the pin an input port.

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

Port C Data Output Register (PCODR)

Bit	7	6	5	4	3	2	1	0
	PC_7ODR	PC_6ODR	PC_5ODR	PC_4ODR	PC_3ODR	PC_2ODR	PC_1ODR	PC_0ODR
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

PCODR is an 8-bit readable/writable register that stores output data for the port C pins (PC_7 to PC_0). PCODR can be read and written to at all times, regardless of the contents of PCDDR.

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

Port C Input Data Register (PCPIN)

Bit	7	6	5	4	3	2	1	0
	PC_7PIN	PC_6PIN	PC_5PIN	PC_4PIN	PC_3PIN	PC_2PIN	PC_1PIN	PC_0PIN
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

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

When a PCPIN read is performed, the pin states are always read.

PCPIN and PCDDR have the same address. When a write is performed, data is written to PCDDR and the port C setting changes.

8.9.3 Pin Functions

Port C pins PC_7 to PC_4 are also used as input pins ($\overline{OCP}_5$ to $\overline{OCP}_2$) that receive overcurrent detection signals (overcurrent signals) output from the USB hub downstream port power supply IC. Port C pins PC_3 to PC_0 are also used as output pins ($\overline{ENP}_5$ to $\overline{ENP}_2$) for power supply output enable signals (enable signals) input to the USB hub downstream port power supply IC. The power supply IC control pin function can be enabled or disabled for each $\overline{OCP}$/$\overline{ENP}$ pair by means of bits 3 to 0 (HOC5E to HOC2E) in the USB's HOCCR register.

When the power supply IC control pin function is disabled, port C is used as an I/O port, with input or output specifiable individually for each pin. Setting a PCDDR bit to 1 makes the corresponding port C pin an output port, while clearing the bit to 0 makes the pin an input port.

8.10 Port D [H8/3567 Group Version with On-Chip USB Only]

8.10.1 Overview

Port D is an 8-bit I/O port.

Port D is provided only in the H8/3567 Group version with an on-chip USB.

Port D is also used for USB hub downstream data input/output.

Port D input/output characteristics are prescribed by the USB bus driver/receiver power supply (DrVCC) voltage.

Figure 8.9 shows the port D pin configuration.

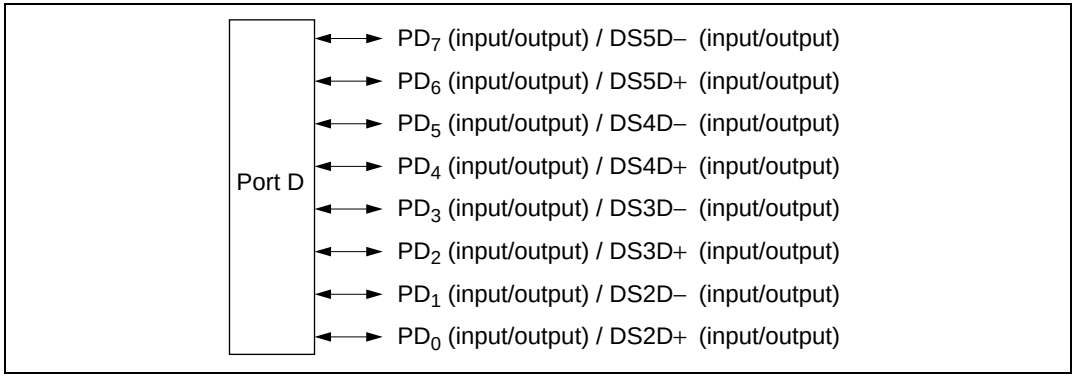


Figure 8.9 Port D Pin Functions

8.10.2 Register Configuration

Table 8.18 shows the port D register configuration.

Table 8.18 Port D Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port D data direction register	PDDDR	W	H'00	H'FE4F
Port D output data register	PDODR	R/W	H'00	H'FE4D
Port D input data register	PDPIN	R	Undefined	H'FE4F

Note: * PDPIN and PDDDR have the same address.

Port D Data Direction Register (PDDDR)

Bit	7	6	5	4	3	2	1	0
	PD_7DDR	PD_6DDR	PD_5DDR	PD_4DDR	PD_3DDR	PD_2DDR	PD_1DDR	PD_0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

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

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

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

Port D Data Output Register (PDODR)

Bit	7	6	5	4	3	2	1	0
	PD_7ODR	PD_6ODR	PD_5ODR	PD_4ODR	PD_3ODR	PD_2ODR	PD_1ODR	PD_0ODR
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

PDODR is an 8-bit readable/writable register that stores output data for the port D pins (PD_7 to PD_0). PDODR can be read and written to at all times, regardless of the contents of PDDDR.

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

Port D Input Data Register (PDPIN)

Bit	7	6	5	4	3	2	1	0
	PD_7PIN	PD_6PIN	PD_5PIN	PD_4PIN	PD_3PIN	PD_2PIN	PD_1PIN	PD_0PIN
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

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

When a PDPIN read is performed, the pin states are always read.

PDPIN and PDDDR have the same address. When a write is performed, data is written to PDDDR and the port D setting changes.

8.10.3 Pin Functions

Port D pins are also used for USB hub downstream data input/output.

When the FONLY bit is cleared to 1 in the USBCR register, port D is used as an I/O port, with input or output specifiable individually for each pin. Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, while clearing the bit to 0 makes the pin an input port. When the FONLY bit is cleared to 0, port D is used for USB hub downstream data input/output.

The USB provided in the H8/3567 Group has a built-in bus driver/receiver, and port D operates on the bus driver/receiver power supply (DrV_{CC}) regardless of the setting of the FONLY bit. Therefore, Port D input/output characteristics are prescribed by the DrV_{CC} voltage.

Section 9 8-Bit PWM Timers

9.1 Overview

The H8/3577 Group and H8/3567 Group have an on-chip PWM (pulse width modulation) timer, with sixteen (H8/3577 Group) or eight (H8/3567 Group) outputs. Sixteen output waveforms are generated from a common time base, enabling PWM output with a high carrier frequency to be produced using pulse division. The PWM timer module has sixteen 8-bit PWM data registers (PWDRs), and an output pulse with a duty cycle of 0 to 100% can be obtained as specified by PWDR and the port data register (P1DR or P2DR).

9.1.1 Features

The PWM timer module has the following features.

- Operable at a maximum carrier frequency of 1.25 MHz using pulse division (at 20 MHz operation)
- Duty cycles from 0 to 100% with 1/256 resolution (100% duty realized by port output)
- Direct or inverted PWM output, and PWM output enable/disable control

9.1.2 Block Diagram

Figure 9.1 shows a block diagram of the PWM timer module.

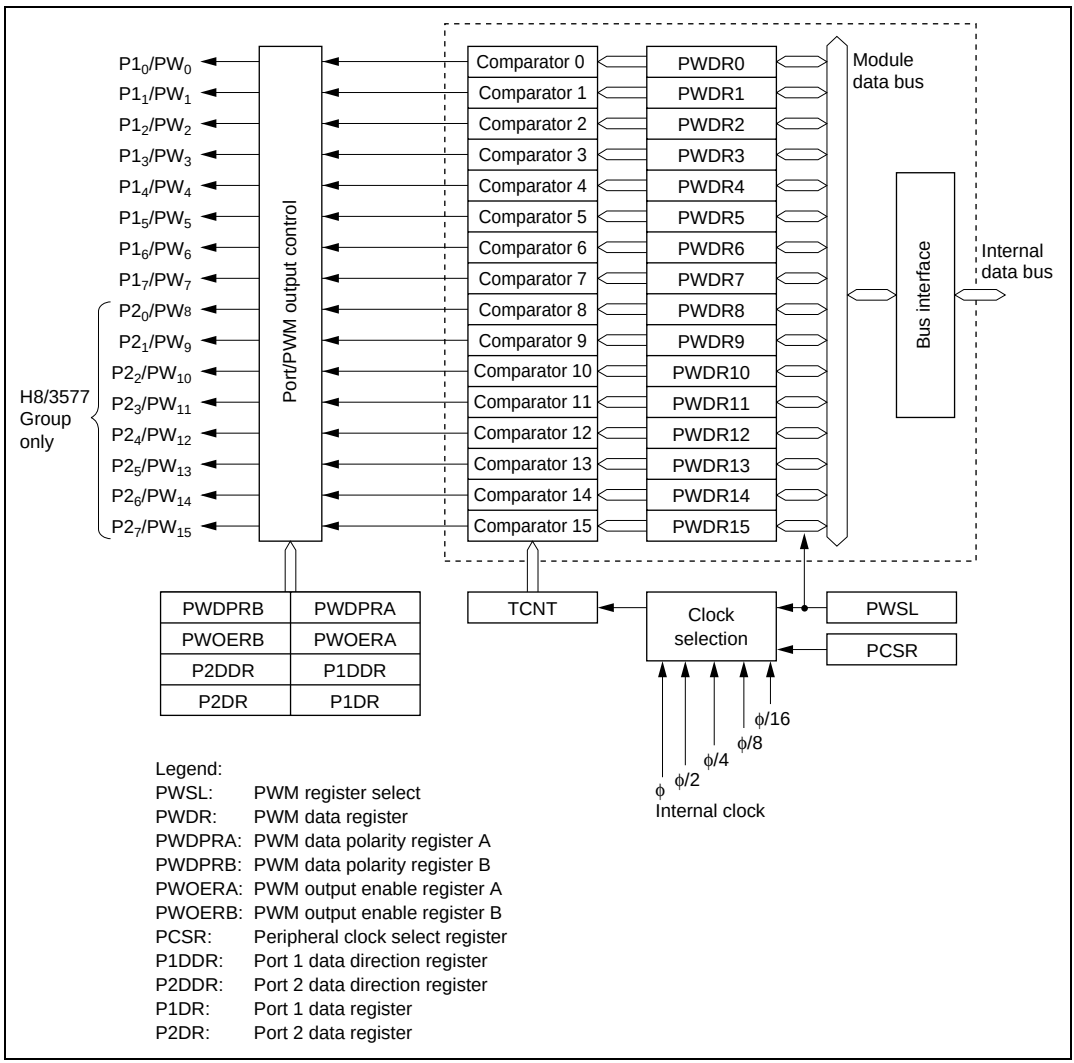


Figure 9.1 Block Diagram of PWM Timer Module

9.1.3 Pin Configuration

Table 9.1 shows the PWM output pin.

Table 9.1 Pin Configuration

Name	Abbreviation	I/O	Function
PWM output pin 0 to 7	PW_0 to PW_7	Output	PWM timer pulse output 0 to 7
PWM output pin 8 to 15	PW_8 to PW_{15}	Output	PWM timer pulse output 8 to 15 (H8/3577 Group only)

9.1.4 Register Configuration

Table 9.2 lists the registers of the PWM timer module.

Table 9.2 PWM Timer Module Registers

Name	Abbreviation	R/W	Initial Value	Address
PWM register select	PWSL	R/W	H'20	H'FFD6
PWM data registers 0 to 15	PWDR0 to PWDR15	R/W	H'00	H'FFD7
PWM data polarity register A	PWDPRA	R/W	H'00	H'FFD5
PWM data polarity register B	PWDPRB	R/W	H'00	H'FFD4
PWM output enable register A	PWOERA	R/W	H'00	H'FFD3
PWM output enable register B	PWOERB	R/W	H'00	H'FFD2
Port 1 data direction register	P1DDR	W	H'00	H'FFB0
Port 2 data direction register	P2DDR	W	H'00	H'FFB1
Port 1 data register	P1DR	R/W	H'00	H'FFB2
Port 2 data register	P2DR	R/W	H'00	H'FFB3
Peripheral clock select register	PCSR	R/W	H'00	H'FF82
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

9.2 Register Descriptions

9.2.1 PWM Register Select (PWSL)

Bit	7	6	5	4	3	2	1	0
	PWCKE	PWCKS	—	—	RS3	RS2	RS1	RS0
Initial value	0	0	1	0	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

PWSL is an 8-bit readable/writable register used to select the PWM timer input clock and the PWM data register.

PWSL is initialized to H'20 by a reset, and in the standby modes, and module stop mode.

Bits 7 and 6—PWM Clock Enable, PWM Clock Select (PWCKE, PWCKS): These bits, together with bits PWCKA and PWCKB in PCSR, select the internal clock input to TCNT in the PWM timer.

PWSL		PCSR			
Bit 7	**Bit 6**	**Bit 2**	**Bit 1**		
PWCKE	**PWCKS**	**PWCKB**	**PWCKA**	**Description**	
0	—	—	—	Clock input is disabled	(Initial value)
1	0	—	—	ϕ (system clock) is selected	
	1	0	0	$\phi/2$ is selected	
			1	$\phi/4$ is selected	
		1	0	$\phi/8$ is selected	
			1	$\phi/16$ is selected	

The PWM resolution, PWM conversion period, and carrier frequency depend on the selected internal clock, and can be found from the following equations.

Resolution (minimum pulse width) = 1/internal clock frequency
PWM conversion period = resolution × 256
Carrier frequency = 16/PWM conversion period

Thus, with a 20 MHz system clock (ϕ), the resolution, PWM conversion period, and carrier frequency are as shown below.

Table 9.3 Resolution, PWM Conversion Period, and Carrier Frequency when φ = 20 MHz

Internal Clock Frequency	Resolution	PWM Conversion Period	Carrier Frequency
φ	50 ns	12.8 µs	1250 kHz
φ/2	100 ns	25.6 µs	625 kHz
φ/4	200 ns	51.2 µs	312.5 kHz
φ/8	400 ns	102.4 µs	156.3 kHz
φ/16	800 ns	204.8 µs	78.1 kHz

Bit 5—Reserved: This bit is always read as 1 and cannot be modified.

Bit 4—Reserved: This bit is always read as 0 and cannot be modified.

Bits 3 to 0—Register Select (RS3 to RS0): These bits select the PWM data register.

Bit 3	Bit 2	Bit 1	Bit 0	
RS3	RS2	RS1	RS0	Register Selection
0	0	0	0	PWDR0 selected
			1	PWDR1 selected
		1	0	PWDR2 selected
			1	PWDR3 selected
	1	0	0	PWDR4 selected
			1	PWDR5 selected
		1	0	PWDR6 selected
			1	PWDR7 selected
1	0	0	0	PWDR8 selected
			1	PWDR9 selected
		1	0	PWDR10 selected
			1	PWDR11 selected
	1	0	0	PWDR12 selected
			1	PWDR13 selected
		1	0	PWDR14 selected
			1	PWDR15 selected

9.2.2 PWM Data Registers (PWDR0 to PWDR15)

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

Each PWDR is an 8-bit readable/writable register that specifies the duty cycle of the basic pulse to be output, and the number of additional pulses. The value set in PWDR corresponds to a 0 or 1 ratio in the conversion period. The upper 4 bits specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16. The lower 4 bits specify how many extra pulses are to be added within the conversion period comprising 16 basic pulses. Thus, a specification of 0/256 to 255/256 is possible for 0/1 ratios within the conversion period. For 256/256 (100%) output, port output should be used.

PWDR is initialized to H'00 by a reset, and in the standby modes, and module stop mode.

9.2.3 PWM Data Polarity Registers A and B (PWDPRA and PWDPRB)

PWDPRA

Bit	7	6	5	4	3	2	1	0
	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWDPRB

Bit	7	6	5	4	3	2	1	0
	OS15	OS14	OS13	OS12	OS11	OS10	OS9	OS8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Each PWDPR is an 8-bit readable/writable register that controls the polarity of the PWM output. Bits OS0 to OS15 correspond to outputs PW_0 to PW_{15}.

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

OS	Description	
0	PWM direct output (PWDR value corresponds to high width of output)	(Initial value)
1	PWM inverted output (PWDR value corresponds to low width of output)	

9.2.4 PWM Output Enable Registers A and B (PWOERA and PWOERB)

PWOERA

Bit	7	6	5	4	3	2	1	0
	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWOERB

Bit	7	6	5	4	3	2	1	0
	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Each PWOER is an 8-bit readable/writable register that switches between PWM output and port output. Bits OE15 to OE0 correspond to outputs PW_{15} to PW_0. To set a pin in the output state, a setting in the port direction register is also necessary. Bits $P1_7DDR$ to $P1_0DDR$ correspond to outputs PW_7 to PW_0, and bits $P2_7DDR$ to $P2_0DDR$ correspond to outputs PW_{15} to PW_8.

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

DDR	OE	Description	
0	0	Port input	(Initial value)
	1	Port input	
1	0	Port output or PWM 256/256 output	
	1	PWM output (0 to 255/256 output)	

9.2.5 Peripheral Clock Select Register (PCSR)

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

PCSR is an 8-bit readable/writable register that selects the PWM timer input clock.

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

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

Bits 2 and 1—PWM Clock Select (PWCKB, PWCKA): Together with bits PWCKE and PWCKS in PWSL, these bits select the internal clock input to TCNT in the PWM timer. For details, see section 9.2.1, PWM Register Select (PWSL).

Bit 0—Reserved: Do not set this bit to 1.

9.2.6 Port 1 Data Direction Register (P1DDR)

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

P1DDR is an 8-bit write-only register that specifies the input/output direction and PWM output for each pin of port 1 on a bit-by-bit basis.

Port 1 pins are multiplexed with pins PW_0 to PW_7. The bit corresponding to a pin to be used for PWM output should be set to 1.

For details on P1DDR, see section 8.2, Port 1.

9.2.7 Port 2 Data Direction Register (P2DDR)

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

P2DDR is an 8-bit write-only register that specifies the input/output direction and PWM output for each pin of port J on a bit-by-bit basis.

Port 2 pins are multiplexed with pins PW_8 to PW_{15}. The bit corresponding to a pin to be used for PWM output should be set to 1.

For details on P2DDR, see section 8.3, Port 2.

9.2.8 Port 1 Data Register (P1DR)

Bit	7	6	5	4	3	2	1	0
	$P1_7DR$	$P1_6DR$	$P1_5DR$	$P1_4DR$	$P1_3DR$	$P1_2DR$	$P1_1DR$	$P1_0DR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1DR is an 8-bit readable/writable register used to fix PWM output at 1 (when OS = 0) or 0 (when OS = 1).

For details on P1DR, see section 8.2, Port 1.

9.2.9 Port 2 Data Register (P2DR)

Bit	7	6	5	4	3	2	1	0
	$P2_7DR$	$P2_6DR$	$P2_5DR$	$P2_4DR$	$P2_3DR$	$P2_2DR$	$P2_1DR$	$P2_0DR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2DR is an 8-bit readable/writable register used to fix PWM output at 1 (when OS = 0) or 0 (when OS = 1).

For details on P2DR, see section 8.3, Port 2.

9.2.10 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers, and is used to perform module stop mode control.

When the MSTP11 bit is set to 1, 8-bit PWM timer operation is halted and a transition is made to module stop mode. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 3—Module Stop (MSTP11): Specifies PWM module stop mode.

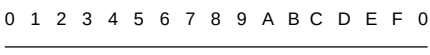
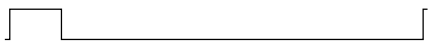
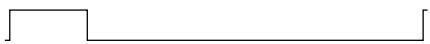
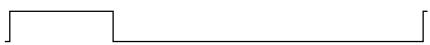
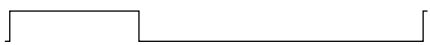
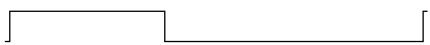
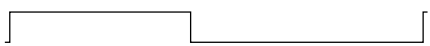
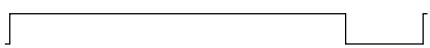
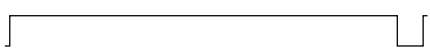
MSTPCRH Bit 3	
MSTP11	**Description**
0	PWM module stop mode is cleared
1	PWM module stop mode is set (Initial value)

9.3 Operation

9.3.1 Correspondence between PWM Data Register Contents and Output Waveform

The upper 4 bits of PWDR specify the duty cycle of the basic pulse as 0/16 to 15/16 with a resolution of 1/16, as shown in table 9.4.

Table 9.4 Duty Cycle of Basic Pulse

Upper 6 Bits	Basic Pulse Waveform (Internal)
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

The lower 4 bits of PWDR specify the position of pulses added to the 16 basic pulses, as shown in table 9.5. An additional pulse consists of a high period (when OS = 0) with a width equal to the resolution, added before the rising edge of a basic pulse. When the upper 4 bits of PWDR are 0000, there is no rising edge of the basic pulse, but the timing for adding pulses is the same.

Table 9.5 Position of Pulses Added to Basic Pulses

	Basic Pulse No.														
Lower 4 Bits	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0000																
0001																Yes
0010								Yes								Yes
0011								Yes				Yes				Yes
0100				Yes				Yes				Yes				Yes
0101				Yes				Yes				Yes		Yes		Yes
0110				Yes		Yes		Yes				Yes		Yes		Yes
0111				Yes		Yes		Yes		Yes		Yes		Yes		Yes
1000		Yes		Yes		Yes		Yes		Yes		Yes		Yes		Yes
1001		Yes		Yes		Yes		Yes		Yes		Yes		Yes	Yes	Yes
1010		Yes		Yes		Yes	Yes	Yes		Yes		Yes		Yes	Yes	Yes
1011		Yes		Yes		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes
1100		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes
1101		Yes	Yes	Yes		Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
1110		Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
1111		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

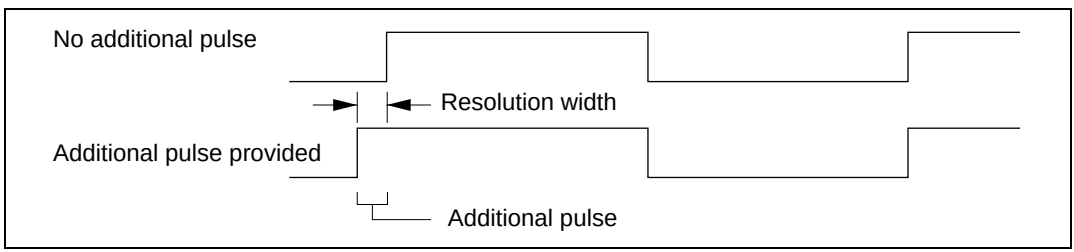


Figure 9.2 Example of Additional Pulse Timing (When Upper 4 Bits of PWDR = 1000)

Section 10 14-Bit PWM Timer

10.1 Overview

The H8/3577 Group and H8/3567 Group have an on-chip 14-bit PWM (pulse width modulator) with two output channels.

Each channel can be connected to an external low-pass filter to operate as a 14-bit D/A converter.

Both channels share the same counter (DACNT) and control register (DACR).

10.1.1 Features

The features of the 14-bit PWM D/A are listed below.

- The pulse is subdivided into multiple base cycles to reduce ripple.
- Two resolution settings and two base cycle settings are available

  The resolution can be set equal to one or two system clock cycles. The base cycle can be set equal to $T \times 64$ or $T \times 256$, where T is the resolution.
- Four operating rates

  The two resolution settings and two base cycle settings combine to give a selection of four operating rates.

10.1.2 Block Diagram

Figure 10.1 shows a block diagram of the PWM D/A module.

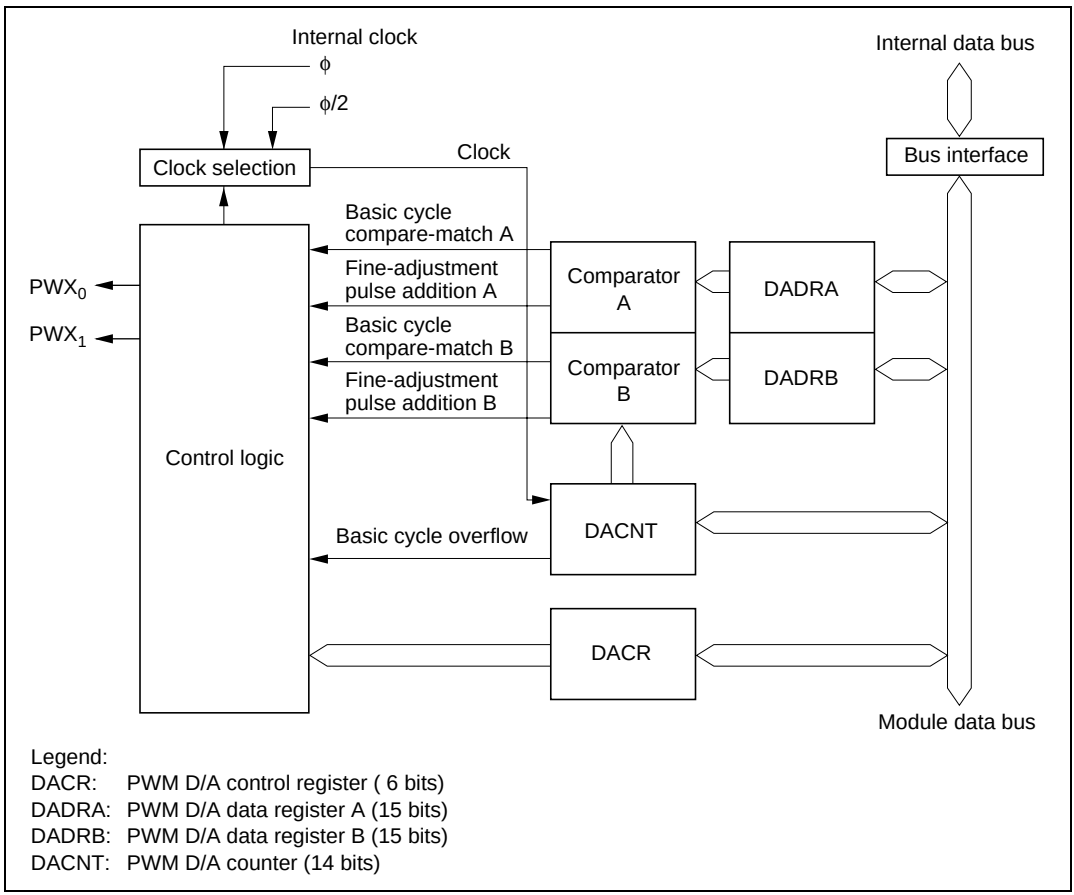


Figure 10.1 PWM D/A Block Diagram

10.1.3 Pin Configuration

Table 10.1 lists the pins used by the PWM D/A module.

Table 10.1 Input and Output Pins

Channel	Name	Abbr.	I/O	Function
A	PWM output pin 0	PWX_0	Output	PWM output, channel A
B	PWM output pin 1	PWX_1	Output	PWM output, channel B

10.1.4 Register Configuration

Table 10.2 lists the registers of the PWM D/A module.

Table 10.2 Register Configuration

Name	Abbreviation	R/W	Initial value	Address
PWM D/A control register	DACR	R/W	H'30	H'FFA0*
PWM D/A data register A high	DADRAH	R/W	H'FF	H'FFA0*
PWM D/A data register A low	DADRAL	R/W	H'FF	H'FFA1*
PWM D/A data register B high	DADRBH	R/W	H'FF	H'FFA6*
PWM D/A data register B low	DADRBL	R/W	H'FF	H'FFA7*
PWM D/A counter high	DACNTH	R/W	H'00	H'FFA6*
PWM D/A counter low	DACNTL	R/W	H'03	H'FFA7*
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

Note: * The registers of the 14-bit PWM timer are assigned to the same addresses as other registers. Selection of each register is performed by the IICE bit of the serial timer control register (STCR). Also, the same addresses are shared by DADRAH and DACR, and by DADRB and DACNT. Switching is performed by the REGS bit in DACNT or DADRB.

10.2 Register Descriptions

10.2.1 PWM D/A Counter (DACNT)

	DACNTH								DACNTL							
Bit (CPU)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit (Counter)	7	6	5	4	3	2	1	0	8	9	10	11	12	13	—	—
															—	REGS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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

DACNT is a 14-bit readable/writable up-counter that increments on an input clock pulse. The input clock is selected by the clock select bit (CKS) in DACR. The CPU can read and write the DACNT value, but since DACNT is a 16-bit register, data transfers between it and the CPU are performed using a temporary register (TEMP). See section 10.3, Bus Master Interface, for details.

DACNT functions as the time base for both PWM D/A channels. When a channel operates with 14-bit precision, it uses all DACNT bits. When a channel operates with 12-bit precision, it uses the lower 12 (counter) bits and ignores the upper two (counter) bits.

DACNT is initialized to H'0003 by a reset, in the standby modes, and module stop mode, and by the PWME bit.

Bit 1 of DACNTL (CPU) is not used, and is always read as 1.

DACNTL Bit 0—Register Select (REGS): DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed. The REGS bit can be accessed regardless of whether DADRB or DACNT is selected.

Bit 0

REGS	Description	
0	DADRA and DADRB can be accessed	
1	DACR and DACNT can be accessed	(Initial value)

10.2.2 D/A Data Registers A and B (DADRA and DADRB)

	DADRH								DADRL							
Bit (CPU)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit (Data)	13	12	11	10	9	8	7	6	5	4	3	2	1	0	—	—
DADRA	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—
DADRB	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

There are two 16-bit readable/writable D/A data registers: DADRA and DADRB. DADRA corresponds to PWM D/A channel A, and DADRB to PWM D/A channel B. The CPU can read and write the PWM D/A data register values, but since DADRA and DADRB are 16-bit registers, data transfers between them and the CPU are performed using a temporary register (TEMP). See section 10.3, Bus Master Interface, for details.

The least significant (CPU) bit of DADRA is not used and is always read as 1.

DADR is initialized to H'FFFF by a reset, and in the standby modes, and module stop mode.

Bits 15 to 2—PWM D/A Data 13 to 0 (DA13 to DA0): The digital value to be converted to an analog value is set in the upper 14 bits of the PWM D/A data register.

In each base cycle, the DACNT value is continually compared with these upper 14 bits to determine the duty cycle of the output waveform, and to decide whether to output a fine-adjustment pulse equal in width to the resolution. To enable this operation, the data register must be set within a range that depends on the carrier frequency select bit (CFS). If the DADR value is outside this range, the PWM output is held constant.

A channel can be operated with 12-bit precision by keeping the two lowest data bits (DA0 and DA1) cleared to 0 and writing the data to be converted in the upper 12 bits. The two lowest data bits correspond to the two highest counter (DACNT) bits.

Bit 1—Carrier Frequency Select (CFS)

Bit 1		
CFS	**Description**	
0	Base cycle = resolution (T) × 64 DADR range = H'0401 to H'FFFD	
1	Base cycle = resolution (T) × 256 DADR range = H'0103 to H'FFFF	(Initial value)

DADRA Bit 0—Reserved: This bit cannot be modified and is always read as 1.

DADRB Bit 0—Register Select (REGS): DADRA and DACR, and DADRB and DACNT, are located at the same addresses. The REGS bit specifies which registers can be accessed. The REGS bit can be accessed regardless of whether DADRB or DACNT is selected.

Bit 0		
REGS	**Description**	
0	DADRA and DADRB can be accessed	
1	DACR and DACNT can be accessed	(Initial value)

10.2.3 PWM D/A Control Register (DACR)

Bit	7	6	5	4	3	2	1	0
	TEST	PWME	—	—	OEB	OEA	OS	CKS
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

DACR is an 8-bit readable/writable register that selects test mode, enables the PWM outputs, and selects the output phase and operating speed.

DACR is initialized to H'30 by a reset, and in the standby modes, and module stop mode.

Bit 7—Test Mode (TEST): Selects test mode, which is used in testing the chip. Normally this bit should be cleared to 0.

Bit 7

TEST	Description	
0	PWM (D/A) in user state: normal operation	(Initial value)
1	PWM (D/A) in test state: correct conversion results unobtainable	

Bit 6—PWM Enable (PWME): Starts or stops the PWM D/A counter (DACNT).

Bit 6

PWME	Description	
0	DACNT operates as a 14-bit up-counter	(Initial value)
1	DACNT halts at H'0003	

Bits 5 and 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Output Enable B (OEB): Enables or disables output on PWM D/A channel B.

Bit 3

OEB	Description	
0	PWM (D/A) channel B output (at the PWX_1 pin) is disabled	(Initial value)
1	PWM (D/A) channel B output (at the PWX_1 pin) is enabled	

Bit 2—Output Enable A (OEA): Enables or disables output on PWM D/A channel A.

Bit 2

OEA	Description	
0	PWM (D/A) channel A output (at the PWX_0 pin) is disabled	(Initial value)
1	PWM (D/A) channel A output (at the PWX_0 pin) is enabled	

Bit 1—Output Select (OS): Selects the phase of the PWM D/A output.

Bit 1

OS	Description	
0	Direct PWM output	(Initial value)
1	Inverted PWM output	

Bit 0—Clock Select (CKS): Selects the PWM D/A resolution. If the system clock (ϕ) frequency is 10 MHz, resolutions of 100 ns and 200 ns can be selected.

Bit 0

CKS	Description	
0	Operates at resolution (T) = system clock cycle time (t_{cyc})	(Initial value)
1	Operates at resolution (T) = system clock cycle time (t_{cyc}) × 2	

10.2.4 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers, and is used to perform module stop mode control.

When the MSTP11 bit is set to 1, 14-bit PWM timer operation is halted and a transition is made to module stop mode. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 3—Module Stop (MSTP11): Specifies PWMX module stop mode.

MSTPCRH Bit 3

MSTP11	Description	
0	PWMX module stop mode is cleared	
1	PWMX module stop mode is set	(Initial value)

10.3 Bus Master Interface

DACNT, DADRA, and DADRB are 16-bit registers. The data bus linking the bus master and the on-chip supporting modules, however, is only 8 bits wide. When the bus master accesses these registers, it therefore uses an 8-bit temporary register (TEMP).

These registers are written and read as follows (taking the example of the CPU interface).

- Write

  When the upper byte is written, the upper-byte write data is stored in TEMP. Next, when the lower byte is written, the lower-byte write data and TEMP value are combined, and the combined 16-bit value is written in the register.

- Read

  When the upper byte is read, the upper-byte value is transferred to the CPU and the lower-byte value is transferred to TEMP. Next, when the lower byte is read, the lower-byte value in TEMP is transferred to the CPU.

These registers should always be accessed 16 bits at a time (by word access or two consecutive byte accesses), and the upper byte should always be accessed before the lower byte. Correct data will not be transferred if only the upper byte or only the lower byte is accessed.

Figure 10.2 shows the data flow for access to DACNT. The other registers are accessed similarly.

Example 1: Write to DACNT

```
MOV.W R0, @DACNT      ; Write R0 contents to DACNT
```

Example 2: Read DADRA

```
MOV.W @DADRA, R0      ; Copy contents of DADRA to R0
```

Table 10.3 Read and Write Access Methods for 16-Bit Registers

	Read		Write	
Register Name	**Word**	**Byte**	**Word**	**Byte**
DADRA and DADRB	Yes	Yes	Yes	×
DACNT	Yes	×	Yes	×

Notes: Yes: Permitted type of access. Word access includes successive byte accesses to the upper byte (first) and lower byte (second).
×: This type of access may give incorrect results.

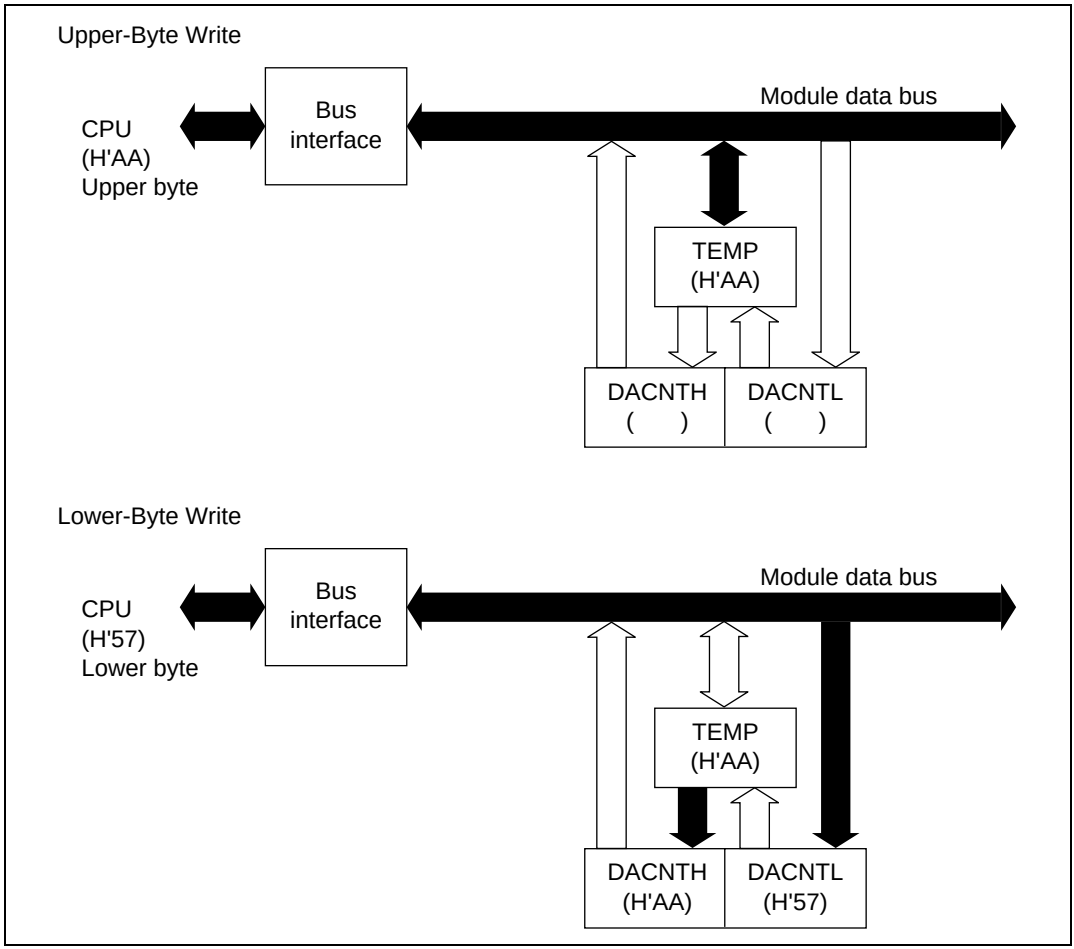


Figure 10.2 (a) Access to DACNT (CPU Writes H'AA57 to DACNT)

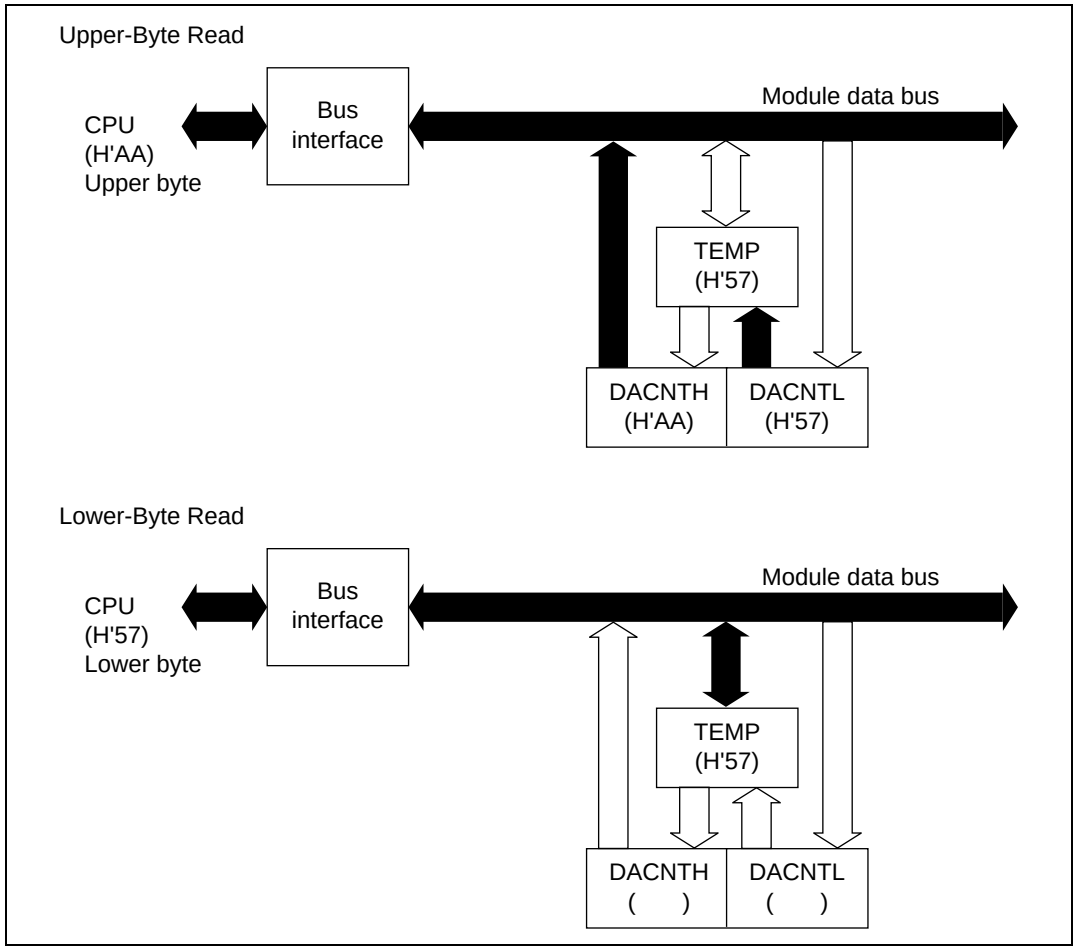


Figure 10.2 (b) Access to DACNT (CPU Reads H'AA57 from DACNT)

10.4 Operation

A PWM waveform like the one shown in figure 10.3 is output from the PWMX pin. When OS = 0, the value in DADR corresponds to the total width (T_L) of the low (0) pulses output in one conversion cycle (256 pulses when CFS = 0, 64 pulses when CFS = 1). When OS = 1, the output waveform is inverted and the DADR value corresponds to the total width (T_H) of the high (1) output pulses. Figure 10.4 shows the types of waveform output available.

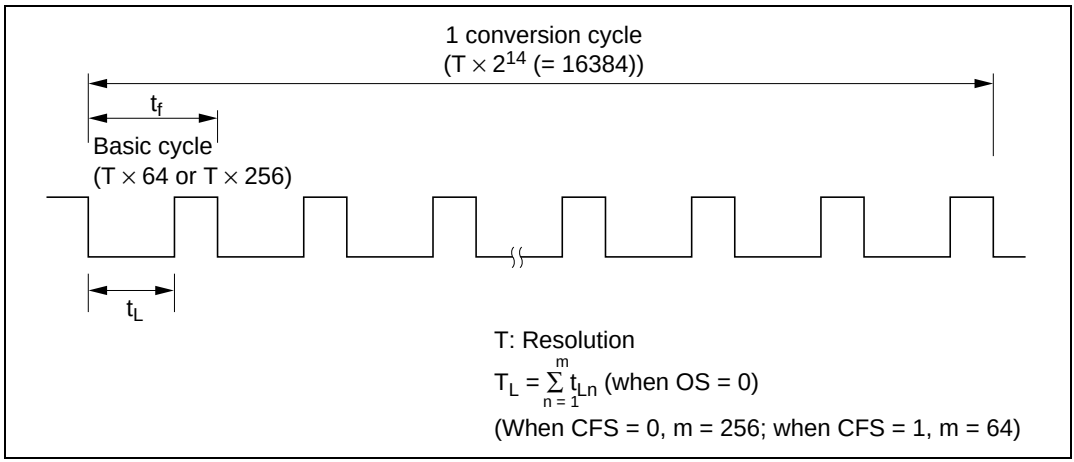


Figure 10.3 PWM D/A Operation

Table 10.4 summarizes the relationships of the CKS, CFS, and OS bit settings to the resolution, base cycle, and conversion cycle. The PWM output remains flat unless DADR contains at least a certain minimum value. Table 10.4 indicates the range of DADR settings that give an output waveform like the one in figure 10.3, and lists the conversion cycle length when low-order DADR bits are kept cleared to 0, reducing the conversion precision to 12 bits or 10 bits.

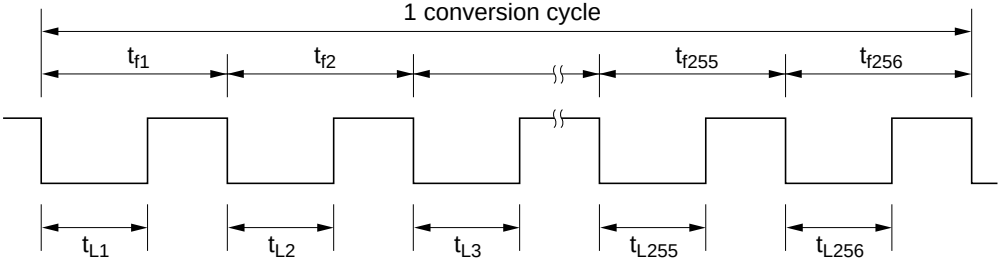
Table 10.4 Settings and Operation (Examples when ϕ = 10 MHz)

CKS	Resolution T (µs)	CFS	Base Cycle (µs)	Conversion Cycle (µs)	T_L (if OS = 0) T_H (if OS = 1)	Fixed DADR Bits					Conversion Cycle* (µs)
						Precision (Bits)	Bit Data				
							3	2	1	0	
0	0.1	0	6.4	1638.4	1. Always low (or high) (DADR = H'0001 to H'03FD)	14					1638.4
					2. (Data value) × T (DADR = H'0401 to H'FFFD)	12			0	0	409.6
						10	0	0	0	0	102.4
		1	25.6	1638.4	1. Always low (or high) (DADR = H'0003 to H'00FF)	14					1638.4
					2. (Data value) × T (DADR = H'0103 to H'FFFF)	12			0	0	409.6
						10	0	0	0	0	102.4
1	0.2	0	12.8	3276.8	1. Always low (or high) (DADR = H'0001 to H'03FD)	14					3276.8
					2. (Data value) × T (DADR = H'0401 to H'FFFD)	12			0	0	819.2
						10	0	0	0	0	204.8
		1	51.2	3276.8	1. Always low (or high) (DADR = H'0003 to H'00FF)	14					3276.8
					2. (Data value) × T (DADR = H'0103 to H'FFFF)	12			0	0	819.2
						10	0	0	0	0	204.8

Note: * This column indicates the conversion cycle when specific DADR bits are fixed.

1. OS = 0 (DADR corresponds to T_L)

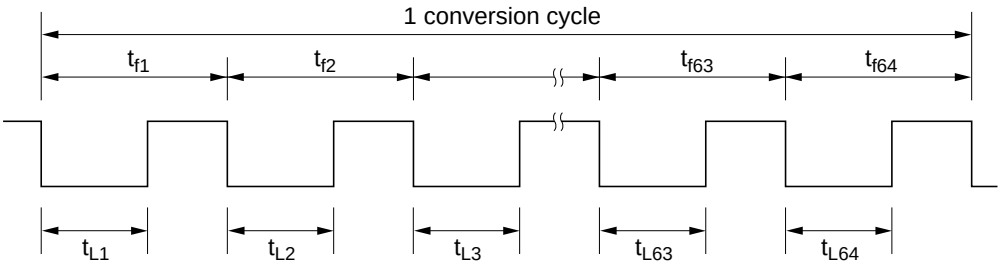
   a. CFS = 0 [base cycle = resolution (T) × 64]



$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f255} = t_{f256} = T \times 64$
$t_{L1} + t_{L2} + t_{L3} + \cdots + t_{L255} + t_{L256} = T_L$

Figure 10.4 (1) Output Waveform

   b. CFS = 1 [base cycle = resolution (T) × 256]



$t_{f1} = t_{f2} = t_{f3} = \cdots = t_{f63} = t_{f64} = T \times 256$
$t_{L1} + t_{L2} + t_{L3} + \cdots + t_{L63} + t_{L64} = T_L$

Figure 10.4 (2) Output Waveform

2. OS = 1 (DADR corresponds to T_H)

   a. CFS = 0 [base cycle = resolution (T) × 64]

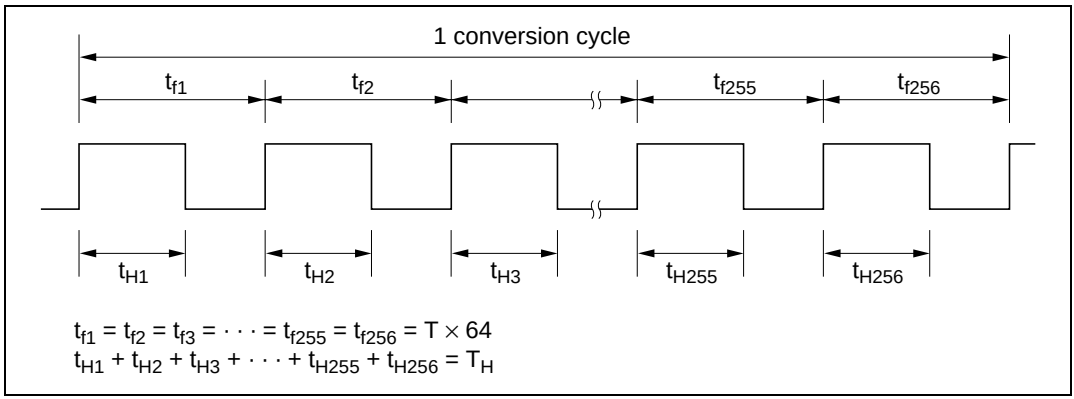


Figure 10.4 (3) Output Waveform

   b. CFS = 1 [base cycle = resolution (T) × 256]

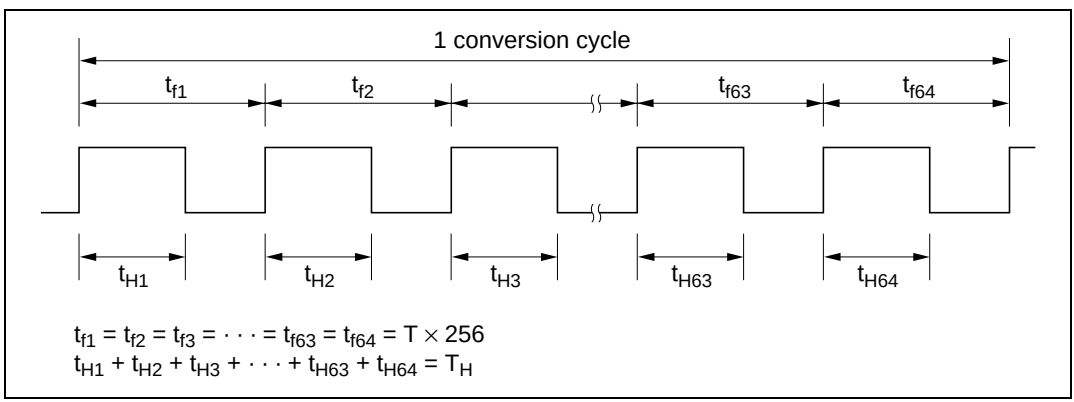


Figure 10.4 (4) Output Waveform

Section 11 16-Bit Free-Running Timer

11.1 Overview

The H8/3577 Group and H8/3567 Group have a single-channel on-chip 16-bit free-running timer (FRT). Applications of the FRT module include rectangular-wave output (up to two independent waveforms), input pulse width measurement, and measurement of external clock periods.

11.1.1 Features

The features of the free-running timer module are listed below.

- Selection of four clock sources
  - The free-running counter can be driven by an internal clock source (φ/2, φ/8, or φ/32), or an external clock input (enabling use as an external event counter).
- Two independent comparators
  - Each comparator can generate an independent waveform.
- Four input capture channels
  - The current count can be captured on the rising or falling edge (selectable) of an input signal.
  - The four input capture registers can be used separately, or in a buffer mode.
- Counter can be cleared under program control
  - The free-running counters can be cleared on compare-match A.
- Seven independent interrupts
  - Two compare-match interrupts, four input capture interrupts, and one overflow interrupt can be requested independently.
- Special functions provided by automatic addition function
  - The contents of OCRAR and OCRAF can be added to the contents of OCRA automatically, enabling a periodic waveform to be generated without software intervention.
  - The contents of ICRD can be added automatically to the contents of OCRDM × 2, enabling input capture operations in this interval to be restricted.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of the free-running timer.

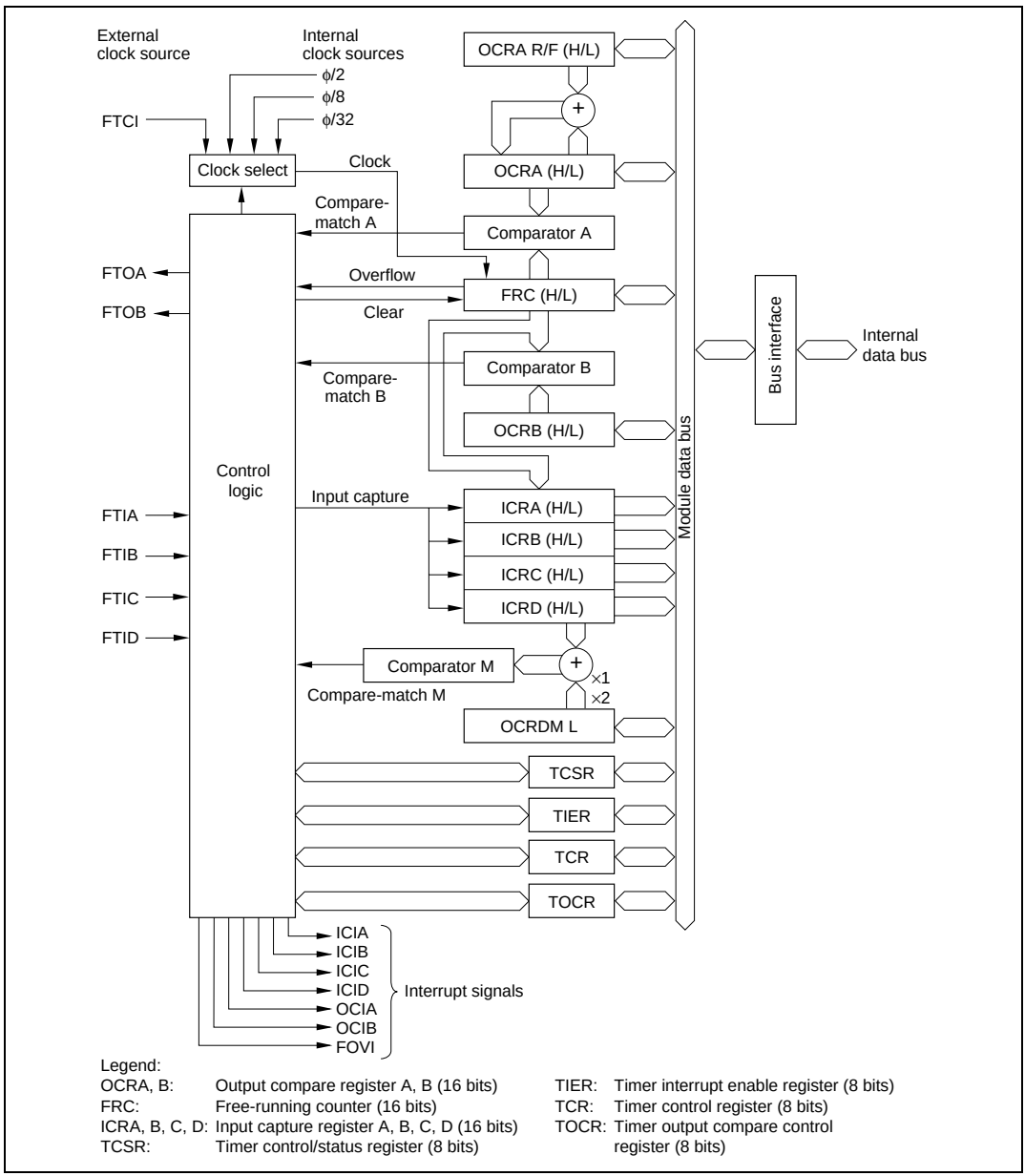


Figure 11.1 Block Diagram of 16-Bit Free-Running Timer

11.1.3 Input and Output Pins

Table 11.1 lists the input and output pins of the free-running timer module.

Table 11.1 Input and Output Pins of Free-Running Timer Module

Name	Abbreviation	I/O	Function
Counter clock input	FTCI	Input	FRC counter clock input
Output compare A	FTOA	Output	Output compare A output
Output compare B	FTOB	Output	Output compare B output
Input capture A	FTIA	Input	Input capture A input
Input capture B	FTIB	Input	Input capture B input
Input capture C	FTIC	Input	Input capture C input
Input capture D	FTID	Input	Input capture D input

11.1.4 Register Configuration

Table 11.2 lists the registers of the free-running timer module.

Table 11.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address
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	FRC	R/W	H'0000	H'FF92
Output compare register A	OCRA	R/W	H'FFFF	H'FF94*2
Output compare register B	OCRB	R/W	H'FFFF	H'FF94*2
Timer control register	TCR	R/W	H'00	H'FF96
Timer output compare control register	TOCR	R/W	H'00	H'FF97
Input capture register A	ICRA	R	H'0000	H'FF98*3
Input capture register B	ICRB	R	H'0000	H'FF9A*3
Input capture register C	ICRC	R	H'0000	H'FF9C*3
Input capture register D	ICRD	R	H'0000	H'FF9E
Output compare register AR	OCRAR	R/W	H'FFFF	H'FF98*3
Output compare register AF	OCRAF	R/W	H'FFFF	H'FF9A*3
Output compare register DM	OCRDM	R/W	H'0000	H'FF9C*3
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

Notes: 1. Bits 7 to 1 are read-only; only 0 can be written to clear the flags.
Bit 0 is readable/writable.
2. OCRA and OCRB share the same address. Access is controlled by the OCRS bit in TOCR.
3. ICRA, ICRB, and ICRC share the same addresses with OCRAR, OCRAF, and OCRDM. Access is controlled by the ICRS bit in TOCR.

11.2 Register Descriptions

11.2.1 Free-Running Counter (FRC)

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

FRC is a 16-bit readable/writable up-counter that increments on an internal pulse generated from a clock source. The clock source is selected by bits CKS1 and CKS0 in TCR.

FRC can also be cleared by compare-match A.

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

FRC is initialized to H'0000 by a reset and in hardware standby mode.

11.2.2 Output Compare Registers A and B (OCRA, OCRB)

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

OCRA and OCRB are 16-bit readable/writable registers, the contents of which are continually compared with the value in the FRC. When a match is detected, the corresponding output compare flags (OCFA or OCFB) is set in TCSR.

In addition, if the output enable bit (OEA or OEB) in TOCR is set to 1, when OCR and FRC values match, the logic level selected by the output level bit (OLVLA or OLVLB) in TOCR is output at the output compare pin (FTOA or FTOB). Following a reset, the FTOA and FTOB output levels are 0 until the first compare-match.

OCR is initialized to H'FFFF by a reset and in hardware standby mode.

11.2.3 Input Capture Registers A to D (ICRA to ICRD)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

There are four input capture registers, A to D, each of which is a 16-bit read-only register.

When the rising or falling edge of the signal at an input capture input pin (FTIA to FTID) is detected, the current FRC value is copied to the corresponding input capture register (ICRA to ICRD). At the same time, the corresponding input capture flag (ICFA to ICFD) in TCSR is set to 1. The input capture edge is selected by the input edge select bits (IEDGA to IEDGD) in TCR.

ICRC and ICRD can be used as ICRA and ICRB buffer registers, respectively, and made to perform buffer operations, by means of buffer enable bits A and B (BUFEA, BUFEB) in TCR.

Figure 11.2 shows the connections when ICRC is specified as the ICRA buffer register (BUFEA = 1). When ICRC is used as the ICRA buffer, both rising and falling edges can be specified as transitions of the external input signal by setting IEDGA ≠ IEDGC. When IEDGA = IEDGC, either the rising or falling edge is designated. See table 11.3.

Note: The FRC contents are transferred to the input capture register regardless of the value of the input capture flag (ICF).

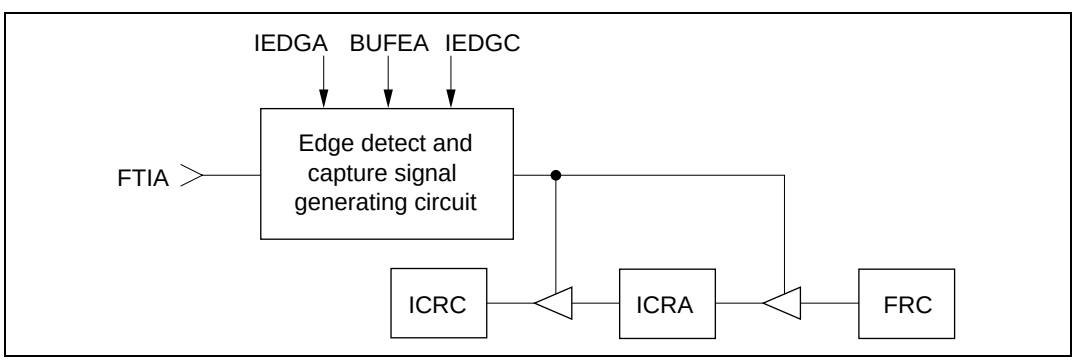


Figure 11.2 Input Capture Buffering (Example)

Table 11.3 Buffered Input Capture Edge Selection (Example)

IEDGA	IEDGC	Description
0	0	Captured on falling edge of input capture A (FTIA) (Initial value)
	1	Captured on both rising and falling edges of input capture A (FTIA)
1	0	
	1	Captured on rising edge of input capture A (FTIA)

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

ICR is initialized to H'0000 by a reset and in hardware standby mode.

11.2.4 Output Compare Registers AR and AF (OCRAR, OCRAF)

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

OCRAR and OCRAF are 16-bit readable/writable registers.

When the OCRAMS bit in TOCR is set to 1, the operation of OCRA is changed to include the use of OCRAR and OCRAF. The contents of OCRAR and OCRAF are automatically added alternately to OCRA, and the result is written to OCRA. The write operation is performed on the occurrence of compare-match A. In the 1st compare-match A after setting the OCRAMS bit to 1, OCRAF is added.

The operation due to compare-match A varies according to whether the compare-match follows addition of OCRAR or OCRAF. The value of the OLVLA bit in TOCR is ignored, and 1 is output on a compare-match A following addition of OCRAF, while 0 is output on a compare-match A following addition of OCRAR.

When using the OCRA automatic addition function, do not select internal clock ϕ/2 as the FRC counter input clock together with a set value of H'0001 or less for OCRAR (or OCRAF).

OCRAR and OCRAF are initialized to H'FFFF by a reset and in hardware standby mode.

11.2.5 Output Compare Register DM (OCRDM)

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

OCRDM is a 16-bit readable/writable register in which the upper 8 bits are fixed at H'00.

When the ICRDMS bit in TOCR is set to 1 and the contents of OCRDM are other than H'0000, the operation of ICRD is changed to include the use of OCRDM. The point at which input capture D occurs is taken as the start of a mask interval. Next, twice the contents of OCRDM is added to the contents of ICRD, and the result is compared with the FRC value. The point at which the values match is taken as the end of the mask interval. New input capture D events are disabled during the mask interval.

A mask interval is not generated when the ICRDMS bit is set to 1 and the contents of OCRDM are H'0000.

OCRDM is initialized to H'0000 by a reset and in hardware standby mode.

11.2.6 Timer Interrupt Enable Register (TIER)

Bit	7	6	5	4	3	2	1	0
	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

TIER is an 8-bit readable/writable register that enables and disables interrupts.

TIER is initialized to H'01 by a reset and in hardware standby mode.

Bit 7—Input Capture Interrupt A Enable (ICIAE): Selects whether to request input capture interrupt A (ICIA) when input capture flag A (ICFA) in TCSR is set to 1.

Bit 7

ICIAE	Description	
0	Input capture interrupt request A (ICIA) is disabled	(Initial value)
1	Input capture interrupt request A (ICIA) is enabled	

Bit 6—Input Capture Interrupt B Enable (ICIBE): Selects whether to request input capture interrupt B (ICIB) when input capture flag B (ICFB) in TCSR is set to 1.

Bit 6

ICIBE	Description	
0	Input capture interrupt request B (ICIB) is disabled	(Initial value)
1	Input capture interrupt request B (ICIB) is enabled	

Bit 5—Input Capture Interrupt C Enable (ICICE): Selects whether to request input capture interrupt C (ICIC) when input capture flag C (ICFC) in TCSR is set to 1.

Bit 5

ICICE	Description	
0	Input capture interrupt request C (ICIC) is disabled	(Initial value)
1	Input capture interrupt request C (ICIC) is enabled	

Bit 4—Input Capture Interrupt D Enable (ICIDE): Selects whether to request input capture interrupt D (ICID) when input capture flag D (ICFD) in TCSR is set to 1.

Bit 4

ICIDE	Description	
0	Input capture interrupt request D (ICID) is disabled	(Initial value)
1	Input capture interrupt request D (ICID) is enabled	

Bit 3—Output Compare Interrupt A Enable (OCIAE): Selects whether to request output compare interrupt A (OCIA) when output compare flag A (OCFA) in TCSR is set to 1.

Bit 3

OCIAE	Description	
0	Output compare interrupt request A (OCIA) is disabled	(Initial value)
1	Output compare interrupt request A (OCIA) is enabled	

Bit 2—Output Compare Interrupt B Enable (OCIBE): Selects whether to request output compare interrupt B (OCIB) when output compare flag B (OCFB) in TCSR is set to 1.

Bit 2

OCIBE	Description	
0	Output compare interrupt request B (OCIB) is disabled	(Initial value)
1	Output compare interrupt request B (OCIB) is enabled	

Bit 1—Timer Overflow Interrupt Enable (OVIE): Selects whether to request a free-running timer overflow interrupt (FOVI) when the timer overflow flag (OVF) in TCSR is set to 1.

Bit 1

OVIE	Description	
0	Timer overflow interrupt request (FOVI) is disabled	(Initial value)
1	Timer overflow interrupt request (FOVI) is enabled	

Bit 0—Reserved: This bit cannot be modified and is always read as 1.

11.2.7 Timer Control/Status Register (TCSR)

Bit	7	6	5	4	3	2	1	0
	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

Note: * Only 0 can be written in bits 7 to 1 to clear these flags.

TCSR is an 8-bit register used for counter clear selection and control of interrupt request signals.

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

Timing is described in section 11.3, Operation.

Bit 7—Input Capture Flag A (ICFA): This status flag indicates that the FRC value has been transferred to ICRA by means of an input capture signal. When BUFEA = 1, ICFA indicates that the old ICRA value has been moved into ICRC and the new FRC value has been transferred to ICRA.

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

Bit 7

ICFA	Description	
0	[Clearing condition]	(Initial value)
	Read ICFA when ICFA = 1, then write 0 in ICFA	
1	[Setting condition]	
	When an input capture signal causes the FRC value to be transferred to ICRA	

Bit 6—Input Capture Flag B (ICFB): This status flag indicates that the FRC value has been transferred to ICRB by means of an input capture signal. When BUFEB = 1, ICFB indicates that the old ICRB value has been moved into ICRD and the new FRC value has been transferred to ICRB.

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

Bit 6

ICFB	Description	
0	[Clearing condition]	(Initial value)
	Read ICFB when ICFB = 1, then write 0 in ICFB	
1	[Setting condition]	
	When an input capture signal causes the FRC value to be transferred to ICRB	

Bit 5—Input Capture Flag C (ICFC): This status flag indicates that the FRC value has been transferred to ICRC by means of an input capture signal. When BUFEA = 1, on occurrence of the signal transition in FTIC (input capture signal) specified by the IEDGC bit, ICFC is set but data is not transferred to ICRC. Therefore, in buffer operation, ICFC can be used as an external interrupt signal (by setting the ICICE bit to 1).

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

Bit 5

ICFC	Description	
0	[Clearing condition]	(Initial value)
	Read ICFC when ICFC = 1, then write 0 in ICFC	
1	[Setting condition]	
	When an input capture signal is received	

Bit 4—Input Capture Flag D (ICFD): This status flag indicates that the FRC value has been transferred to ICRD by means of an input capture signal. When BUFEB = 1, on occurrence of the signal transition in FTID (input capture signal) specified by the IEDGD bit, ICFD is set but data is not transferred to ICRD. Therefore, in buffer operation, ICFD can be used as an external interrupt by setting the ICIDE bit to 1.

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

Bit 4

ICFD	Description	
0	[Clearing condition]	(Initial value)
	Read ICFD when ICFD = 1, then write 0 in ICFD	
1	[Setting condition]	
	When an input capture signal is received	

Bit 3—Output Compare Flag A (OCFA): This status flag indicates that the FRC value matches the OCRA value. This flag must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 3

OCFA	Description	
0	[Clearing condition]	(Initial value)
	Read OCFA when OCFA = 1, then write 0 in OCFA	
1	[Setting condition]	
	When FRC = OCRA	

Bit 2—Output Compare Flag B (OCFB): This status flag indicates that the FRC value matches the OCRB value. This flag must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 2

OCFB	Description	
0	[Clearing condition]	(Initial value)
	Read OCFB when OCFB = 1, then write 0 in OCFB	
1	[Setting condition]	
	When FRC = OCRB	

Bit 1—Timer Overflow Flag (OVF): This status flag indicates that the FRC has overflowed (changed from H'FFFF to H'0000). This flag must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 1

OVF	Description	
0	[Clearing condition]	(Initial value)
	Read OVF when OVF = 1, then write 0 in OVF	
1	[Setting condition]	
	When FRC changes from H'FFFF to H'0000	

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

Bit 0

CCLRA	Description	
0	FRC clearing is disabled	(Initial value)
1	FRC is cleared at compare-match A	

11.2.8 Timer Control Register (TCR)

Bit	7	6	5	4	3	2	1	0
	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 hardware standby mode

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

Bit 7

IEDGA	Description	
0	Capture on the falling edge of FTIA	(Initial value)
1	Capture on the rising edge of FTIA	

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

Bit 6

IEDGB	Description	
0	Capture on the falling edge of FTIB	(Initial value)
1	Capture on the rising edge of FTIB	

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

Bit 5

IEDGC	Description	
0	Capture on the falling edge of FTIC	(Initial value)
1	Capture on the rising edge of FTIC	

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

Bit 4

IEDGD	Description	
0	Capture on the falling edge of FTID	(Initial value)
1	Capture on the rising edge of FTID	

Bit 3—Buffer Enable A (BUFEA): Selects whether ICRC is to be used as a buffer register for ICRA.

Bit 3

BUFEA	Description	
0	ICRC is not used as a buffer register for input capture A	(Initial value)
1	ICRC is used as a buffer register for input capture A	

Bit 2—Buffer Enable B (BUFEB): Selects whether ICRD is to be used as a buffer register for ICRB.

Bit 2		
BUFEB	**Description**	
0	ICRD is not used as a buffer register for input capture B	(Initial value)
1	ICRD is used as a buffer register for input capture B	

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

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

11.2.9 Timer Output Compare Control Register (TOCR)

Bit	7	6	5	4	3	2	1	0
	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TOCR is an 8-bit readable/writable register that enables output from the output compare pins, selects the output levels, switches access between output compare registers A and B, controls the ICRD and OCRA operating mode, and switches access to input capture registers A, B, and C.

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

Bit 7—Input Capture D Mode Select (ICRDMS): Specifies whether ICRD is used in the normal operating mode or in the operating mode using OCRDM.

Bit 7

ICRDMS	Description	
0	The normal operating mode is specified for ICRD	(Initial value)
1	The operating mode using OCRDM is specified for ICRD	

Bit 6—Output Compare A Mode Select (OCRAMS): Specifies whether OCRA is used in the normal operating mode or in the operating mode using OCRAR and OCRAF.

Bit 6

OCRAMS	Description	
0	The normal operating mode is specified for OCRA	(Initial value)
1	The operating mode using OCRAR and OCRAF is specified for OCRA	

Bit 5—Input Capture Register Select (ICRS): The same addresses are shared by ICRA and OCRAR, by ICRB and OCRAF, and by ICRC and OCRDM. The ICRS bit determines which registers are selected when the shared addresses are read or written to. The operation of ICRA, ICRB, and ICRC is not affected.

Bit 5

ICRS	Description	
0	The ICRA, ICRB, and ICRC registers are selected	(Initial value)
1	The OCRAR, OCRAF, and OCRDM registers are selected	

Bit 4—Output Compare Register Select (OCRS): OCRA and OCRB share the same address. When this address is accessed, the OCRS bit selects which register is accessed. This bit does not affect the operation of OCRA or OCRB.

Bit 4

OCRS	Description	
0	The OCRA register is selected	(Initial value)
1	The OCRB register is selected	

Bit 3—Output Enable A (OEA): Enables or disables output of the output compare A signal (FTOA).

Bit 3		
OEA	**Description**	
0	Output compare A output is disabled	(Initial value)
1	Output compare A output is enabled	

Bit 2—Output Enable B (OEB): Enables or disables output of the output compare B signal (FTOB).

Bit 2		
OEB	**Description**	
0	Output compare B output is disabled	(Initial value)
1	Output compare B output is enabled	

Bit 1—Output Level A (OLVLA): Selects the logic level to be output at the FTOA pin in response to compare-match A (signal indicating a match between the FRC and OCRA values). When the OCRAMS bit is 1, this bit is ignored.

Bit 1		
OLVLA	**Description**	
0	0 output at compare-match A	(Initial value)
1	1 output at compare-match A	

Bit 0—Output Level B (OLVLB): Selects the logic level to be output at the FTOB pin in response to compare-match B (signal indicating a match between the FRC and OCRB values).

Bit 0		
OLVLB	**Description**	
0	0 output at compare-match B	(Initial value)
1	1 output at compare-match B	

11.2.10 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

When the MSTP13 bit is set to 1, FRT operation is stopped at the end of the bus cycle, and module stop mode is entered. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 5—Module Stop (MSTP13): Specifies the FRT module stop mode.

Bit 5

MSTPCRH	Description	
0	FRT module stop mode is cleared	
1	FRT module stop mode is set	(Initial value)

11.3 Operation

11.3.1 FRC Increment Timing

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

Internal Clock: Any of three internal clocks (φ/2, φ/8, or φ/32) created by division of the system clock (φ) can be selected by making the appropriate setting in bits CKS1 and CKS0 in TCR. Figure 11.3 shows the increment timing.

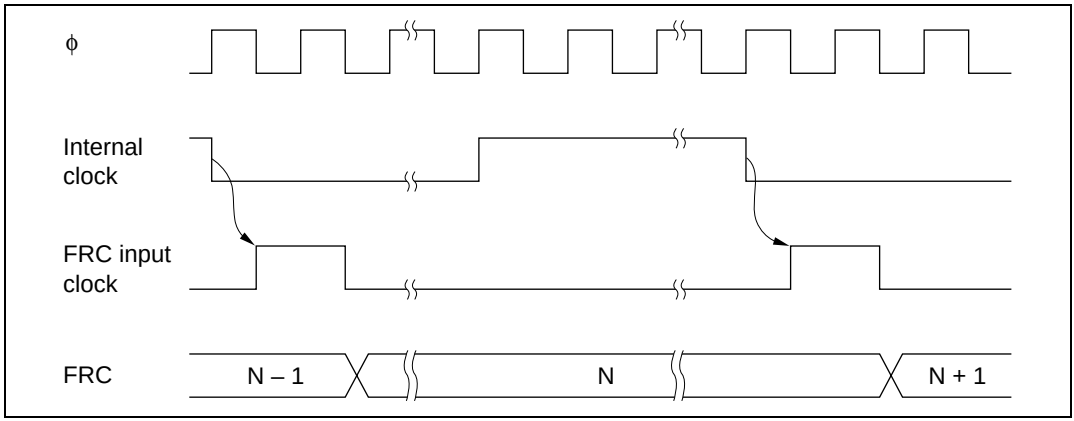


Figure 11.3 Increment Timing with Internal Clock Source

External Clock: If external clock input is selected by bits CKS1 and CKS0 in TCR, FRC increments on the rising edge of the external clock signal.

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

Figure 11.4 shows the increment timing.

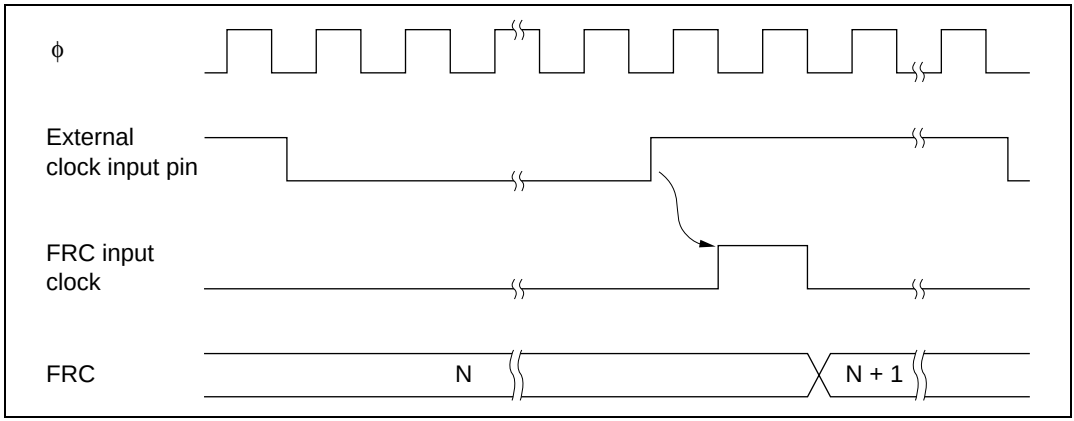


Figure 11.4 Increment Timing with External Clock Source

11.3.2 Output Compare Output Timing

When a compare-match occurs, the logic level selected by the output level bit (OLVLA or OLVLB) in TOCR is output at the output compare pin (FTOA or FTOB). Figure 11.5 shows the timing of this operation for compare-match A.

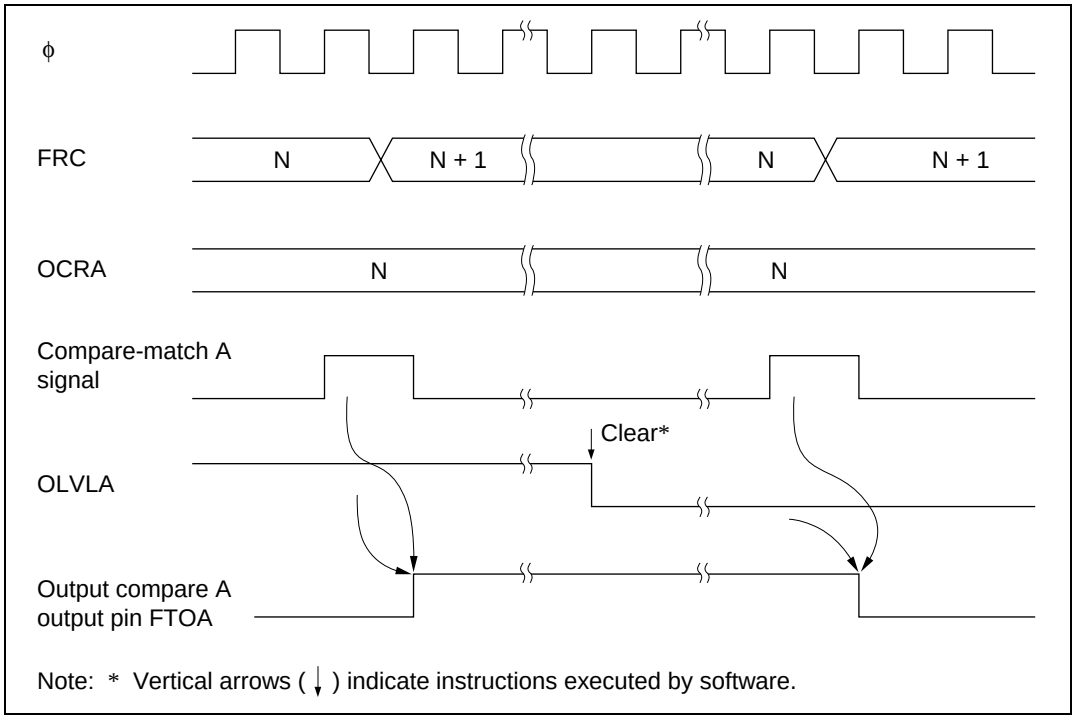


Figure 11.5 Timing of Output Compare A Output

11.3.3 FRC Clear Timing

FRC can be cleared when compare-match A occurs. Figure 11.6 shows the timing of this operation.

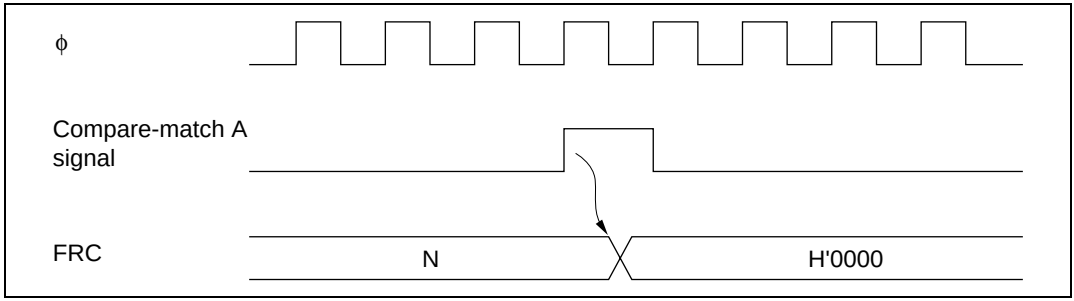


Figure 11.6 Clearing of FRC by Compare-Match A

11.3.4 Input Capture Input Timing

Input Capture Input Timing: An internal input capture signal is generated from the rising or falling edge of the signal at the input capture pin, as selected by the corresponding IEDGA to IEDGD bit in TCR. Figure 11.7 shows the usual input capture timing when the rising edge is selected (IEDGA to IEDGD = 1).

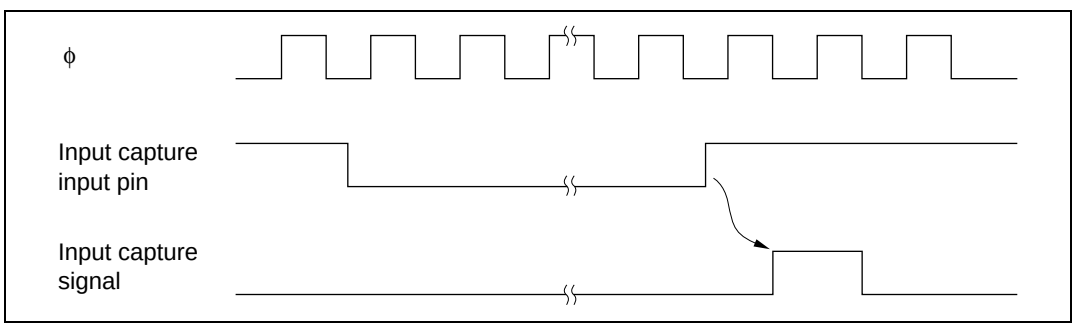


Figure 11.7 Input Capture Signal Timing (Usual Case)

If the upper byte of ICRA to ICRAD is being read when the corresponding input capture signal arrives, the internal input capture signal is delayed by one system clock (ϕ) period. Figure 11.8 shows the timing for this case.

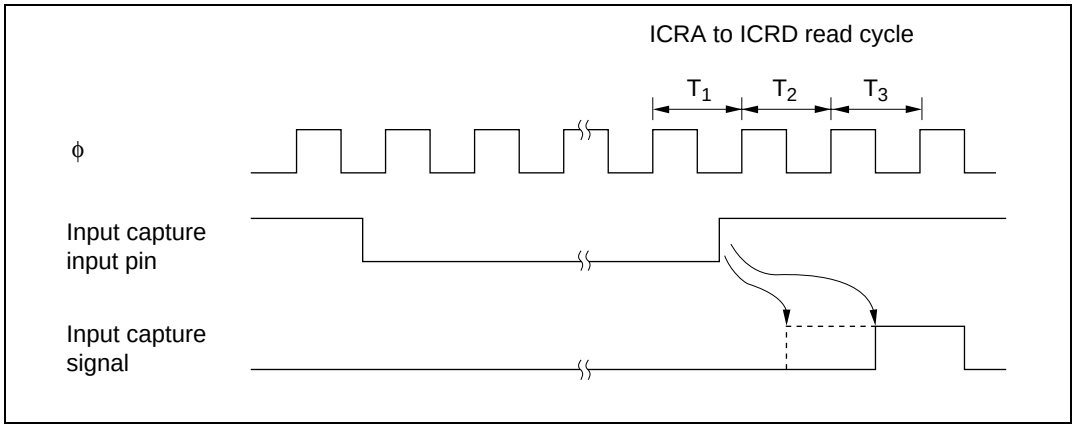


Figure 11.8 Input Capture Signal Timing (Input Capture Input when ICRA to ICRD Is Read)

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

Figure 11.9 shows how input capture operates when ICRA and ICRC are used in buffer mode (BUFEA = 1) and IEDGA and IEDGC are set to different values (IEDGA = 0 and IEDGC = 1, or IEDG A = 1 and IEDGC = 0), so that input capture is performed on both the rising and falling edges of FTIA.

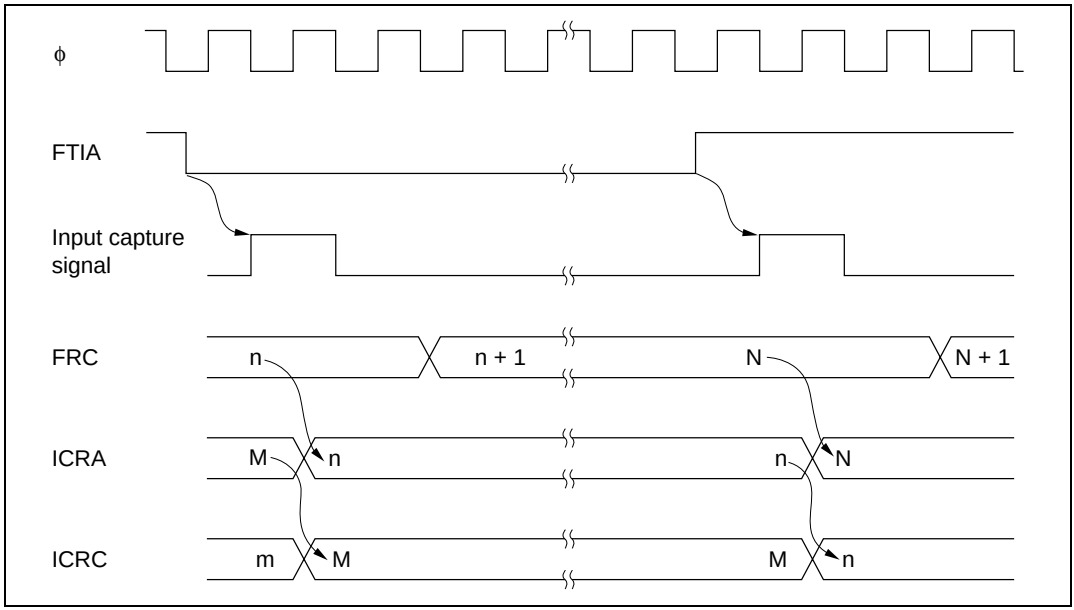


Figure 11.9 Buffered Input Capture Timing (Usual Case)

When ICRC or ICRD is used as a buffer register, its input capture flag is set by the selected transition of its input capture signal. For example, if ICRC is used to buffer ICRA, when the edge transition selected by the IEDGC bit occurs on the FTIC input capture line, ICFC will be set, and if the ICIEC bit is set, an interrupt will be requested. The FRC value will not be transferred to ICRC, however.

In buffered input capture, if the upper byte of either of the two registers to which data will be transferred (ICRA and ICRC, or ICRB and ICRD) is being read when the input signal arrives, input capture is delayed by one system clock (ϕ) period. Figure 11.10 shows the timing when BUFEA = 1.

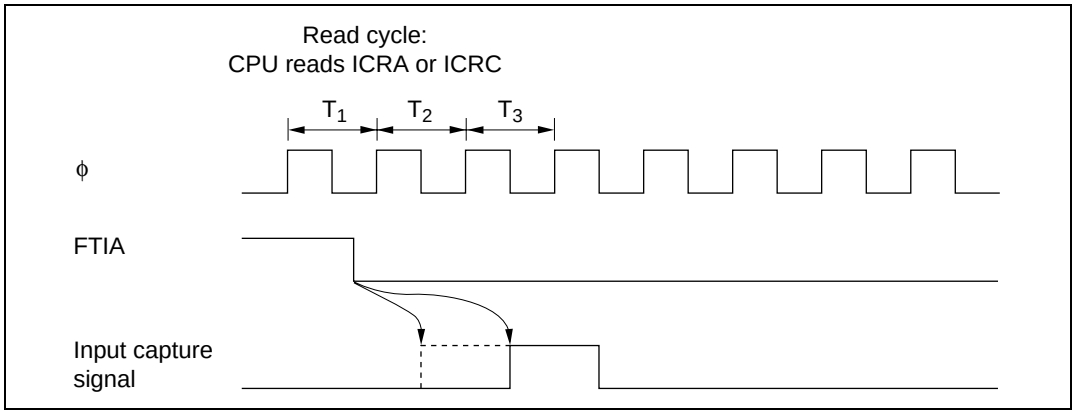


Figure 11.10 Buffered Input Capture Timing (Input Capture Input when ICRA or ICRC Is Read)

11.3.5 Timing of Input Capture Flag (ICFA to ICFD) Setting

The input capture flag ICFA to ICFD is set to 1 by the internal input capture signal. The FRC value is simultaneously transferred to the corresponding input capture register (ICRx). Figure 11.11 shows the timing of this operation.

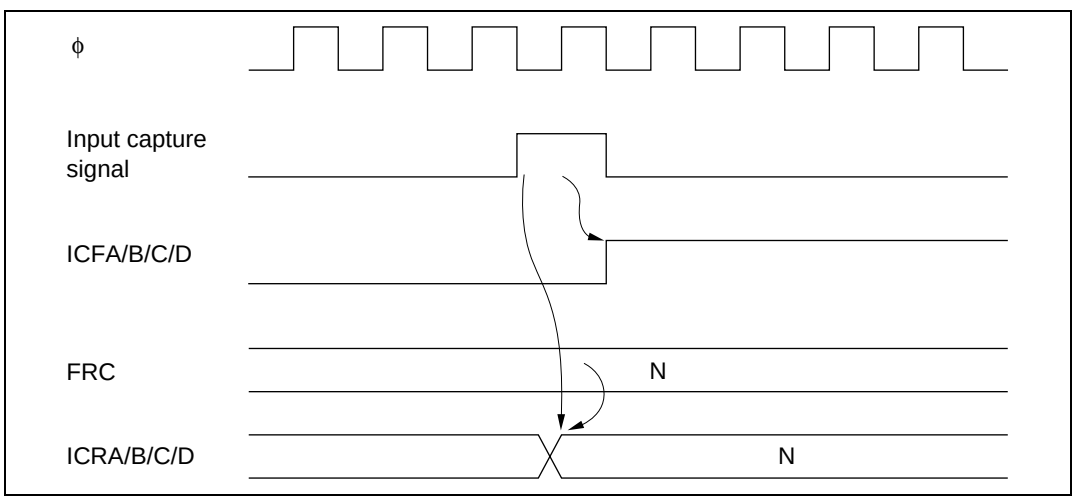


Figure 11.11 Setting of Input Capture Flag (ICFA to ICFD)

11.3.6 Setting of Output Compare Flags A and B (OCFA, OCFB)

The output compare flags are set to 1 by an internal compare-match signal generated when the FRC value matches the OCRA or OCRB value. This compare-match signal is generated at the last state in which the two values match, just before FRC increments to a new value.

Accordingly, when the FRC and OCR values match, the compare-match signal is not generated until the next period of the clock source. Figure 11.12 shows the timing of the setting of OCFA and OCFB.

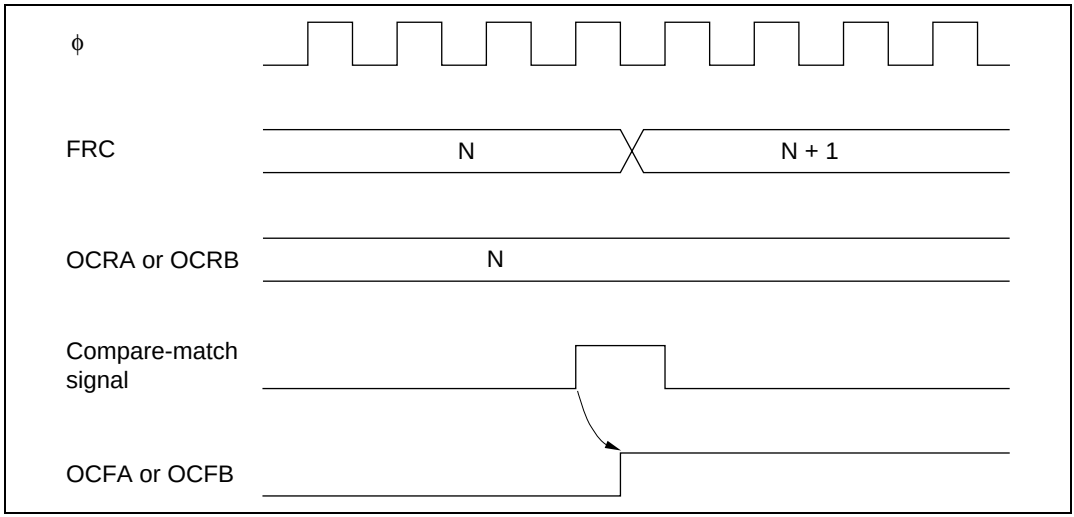


Figure 11.12 Setting of Output Compare Flag (OCFA, OCFB)

11.3.7 Setting of FRC Overflow Flag (OVF)

The FRC overflow flag (OVF) is set to 1 when FRC overflows (changes from H'FFFF to H'0000). Figure 11.13 shows the timing of this operation.

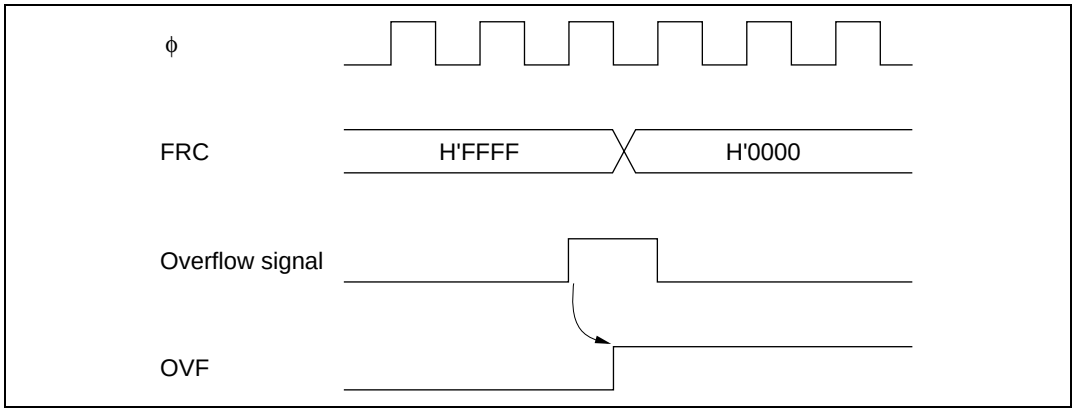


Figure 11.13 Setting of Overflow Flag (OVF)

11.3.8 Automatic Addition of OCRA and OCRAR/OCRAF

When the OCRAMS bit in TOCR is set to 1, the contents of OCRAR and OCRAF are automatically added to OCRA alternately, and when an OCRA compare-match occurs a write to OCRA is performed. The OCRA write timing is shown in figure 11.14.

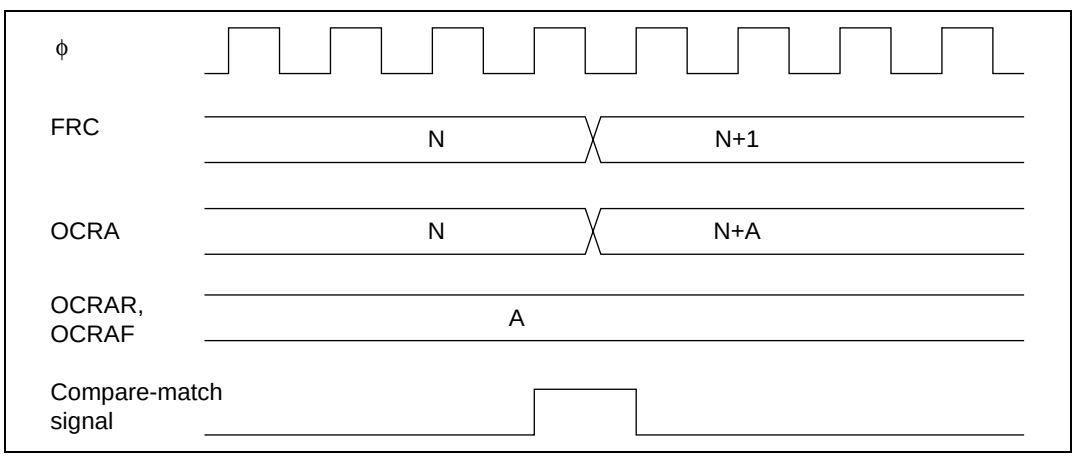


Figure 11.14 OCRA Automatic Addition Timing

11.3.9 ICRD and OCRDM Mask Signal Generation

When the ICRDMS bit in TOCR is set to 1 and the contents of OCRDM are other than H'0000, a signal that masks the ICRD input capture function is generated.

The mask signal is set by the input capture signal. The mask signal setting timing is shown in figure 11.15.

The mask signal is cleared by the sum of the ICRD contents and twice the OCRDM contents, and an FRC compare-match. The mask signal clearing timing is shown in figure 11.16.

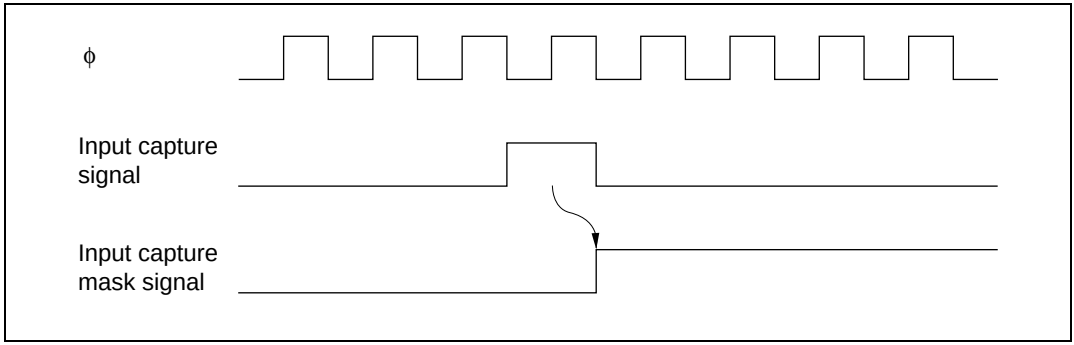


Figure 11.15 Input Capture Mask Signal Setting Timing

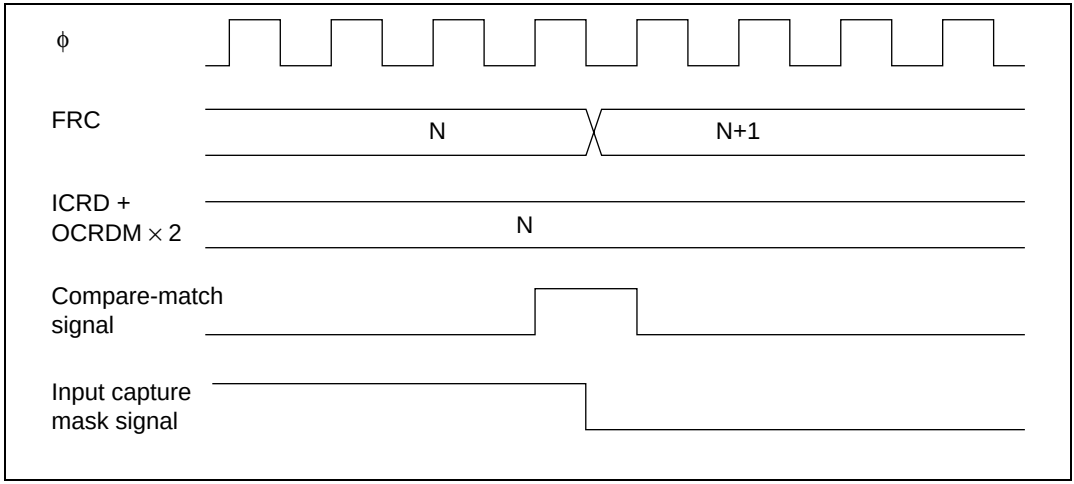


Figure 11.16 Input Capture Mask Signal Clearing Timing

11.4 Interrupts

The free-running timer can request seven interrupts (three types): input capture A to D (ICIA, ICIB, ICIC, ICID), output compare A and B (OCIA and OCIB), and overflow (FOVI). Each interrupt can be enabled or disabled by an enable bit in TIER. Independent signals are sent to the interrupt controller for each interrupt. Table 11.4 lists information about these interrupts.

Table 11.4 Free-Running Timer Interrupts

Interrupt	Description	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

11.5 Sample Application

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

- The CCLRA bit in TCSR is set to 1.
- Each time a compare-match interrupt occurs, software inverts the corresponding output level bit in TOCR (OLVLA or OLVLB).

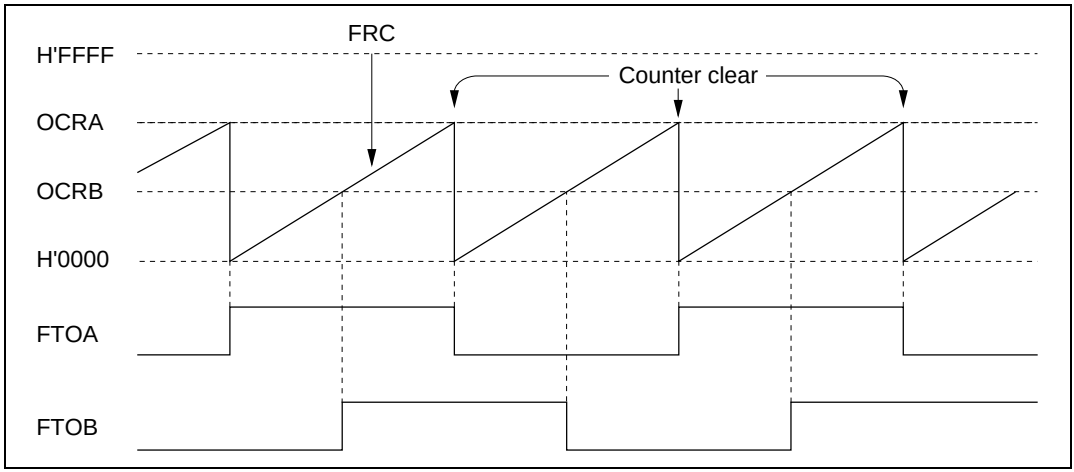


Figure 11.17 Pulse Output (Example)

11.6 Usage Notes

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

Contention between FRC Write and Clear: If an internal counter clear signal is generated during the state after an FRC write cycle, the clear signal takes priority and the write is not performed.

Figure 11.18 shows this type of contention.

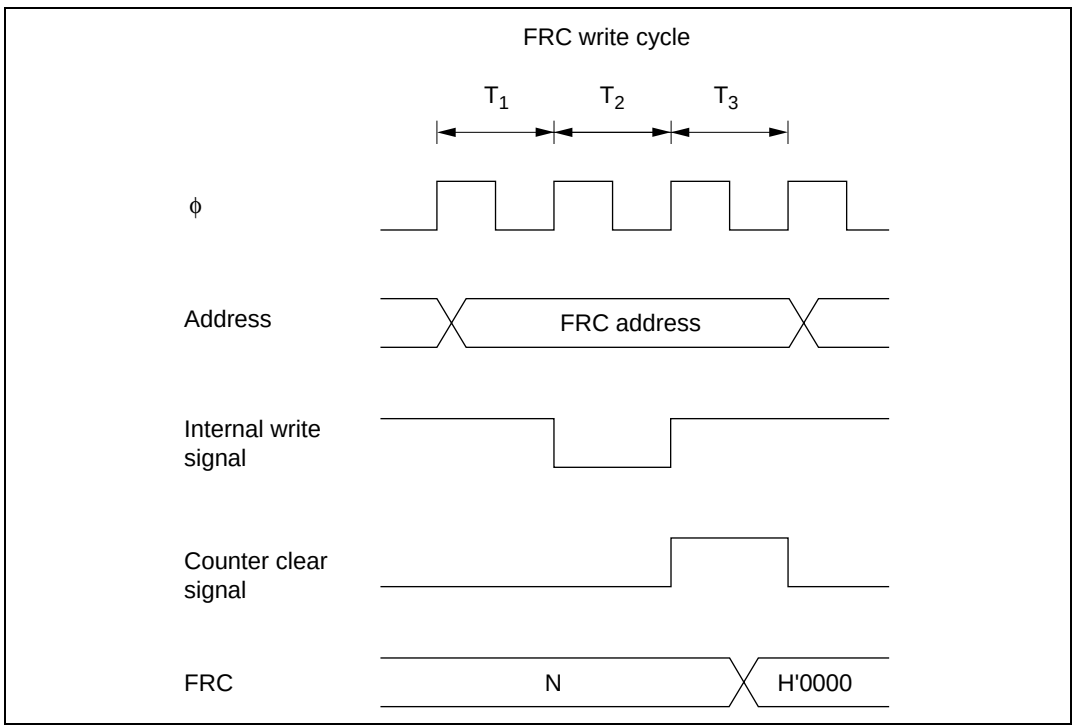


Figure 11.18 FRC Write-Clear Contention

Contention between FRC Write and Increment: Even if an increment pulse is generated in the T_3 state during FRC write cycle, it is not incremented and the count write takes priority.

Figure 11.19 shows this type of contention.

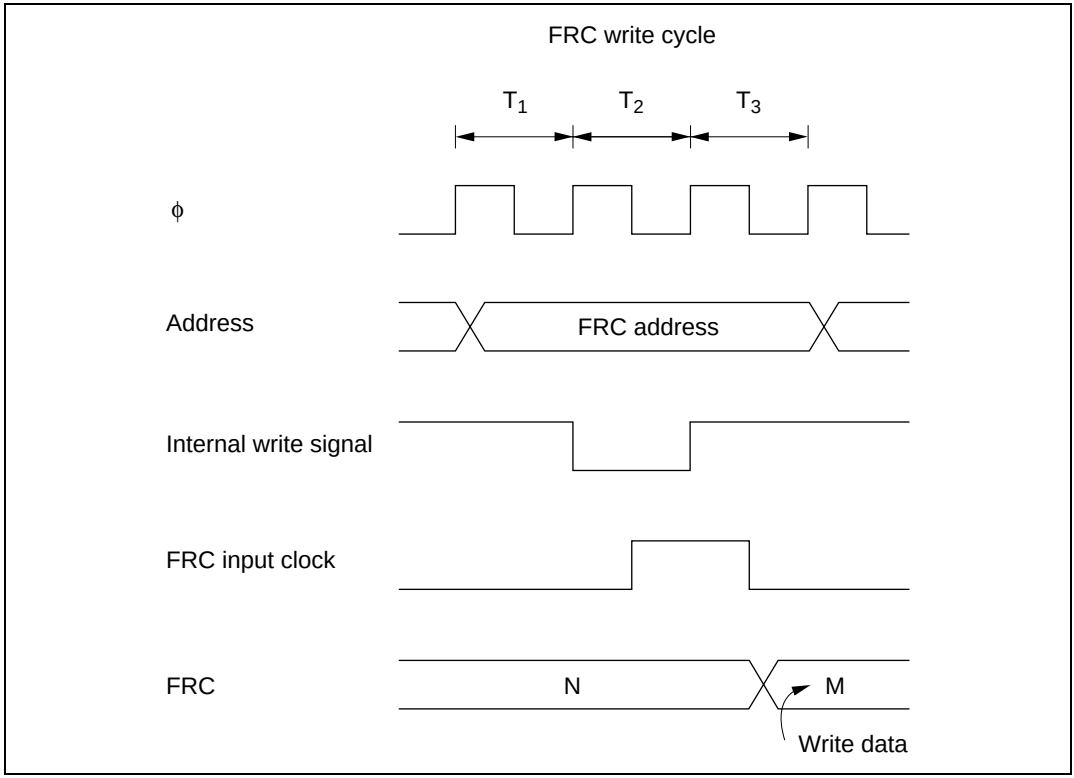


Figure 11.19 FRC Write-Increment Contention

Contention between OCR Write and Compare-Match: If a compare-match occurs in the T_3 state during the OCRA or OCRB write cycle, the OCR write takes priority and the compare-match signal is inhibited.

Figure 11.20 shows this type of contention.

When the automatic addition of OCRAR/OCRAF to OCRA is selected and a compare-match occurs in the T_3 state during the OCRA, OCRAR or OCRAF write cycle, the OCRA, OCRAR or OCRAF write takes priority and the compare-match signal is inhibited. Consequently, the result of automatic addition is not written.

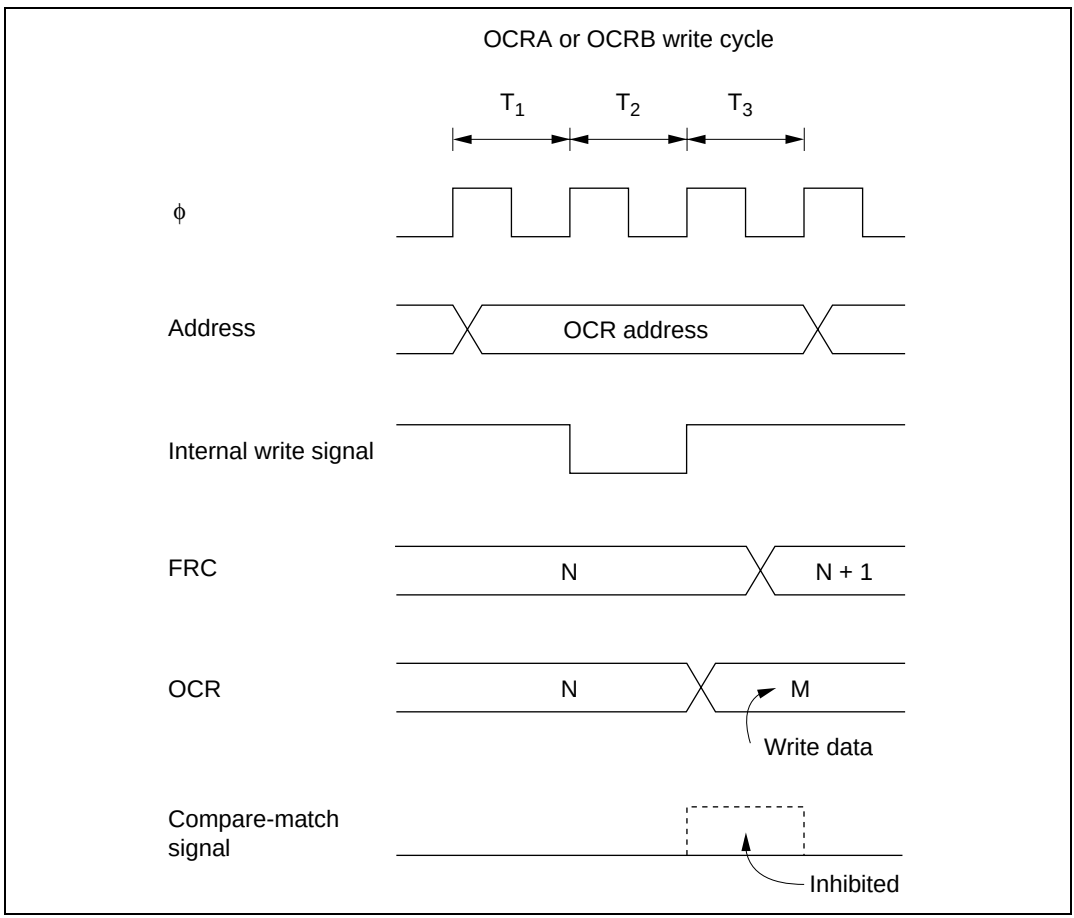


Figure 11.20 Contention between OCR Write and Compare-Match (When Not Using the Function of Automatic Addition)

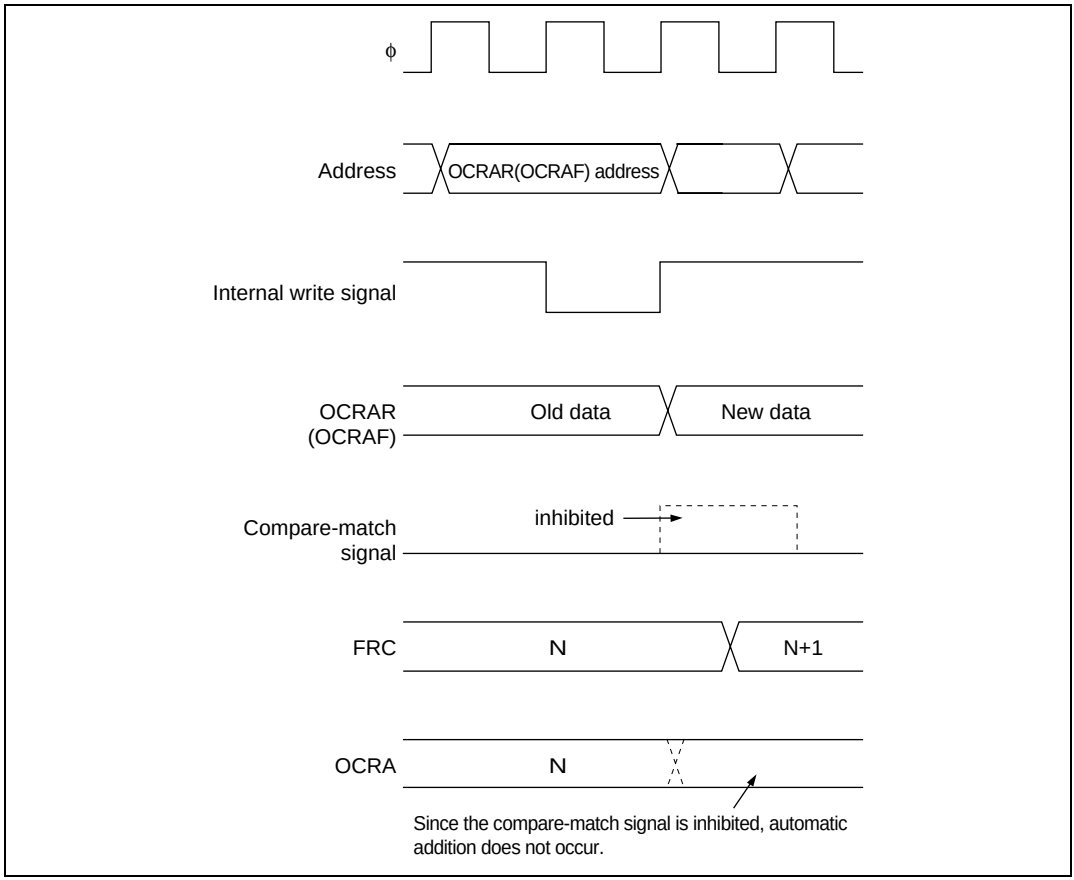


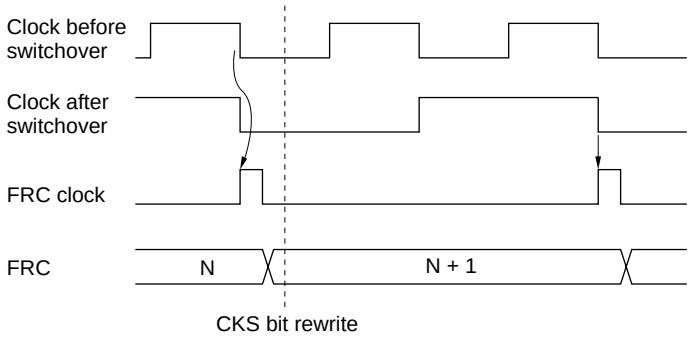
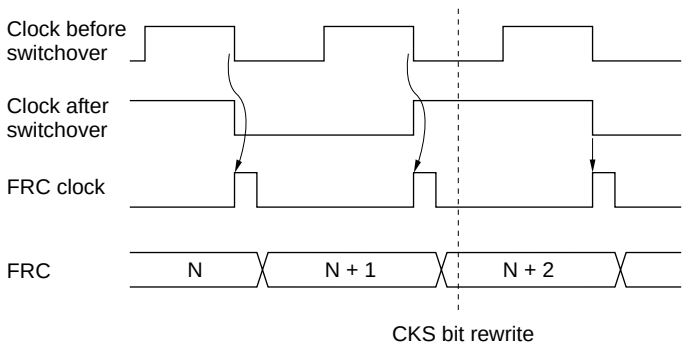
Figure 11.21 Contention between OCRAR/OCRAF Write and Compare-Match (When Using Automatic Addition)

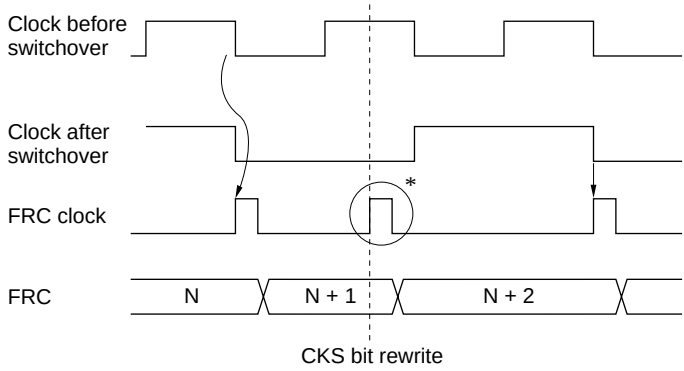
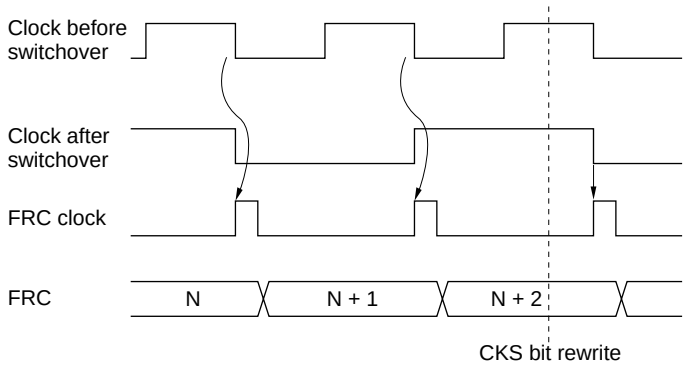
Switching of Internal Clock and FRC Operation: When the internal clock is changed, the changeover may cause FRC to increment. This depends on the time at which the clock select bits (CKS1 and CKS0) are rewritten, as shown in table 11.5.

When an internal clock is used, the FRC clock is generated on detection of the falling edge of the internal clock scaled from the system clock (ϕ). If the clock is changed when the old source is high and the new source is low, as in case no. 3 in table 11.5, the changeover is regarded as a falling edge that triggers the FRC increment clock pulse.

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

Table 11.5 Switching of Internal Clock and FRC Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	FRC Operation
1	Switching from low to low	
2	Switching from low to high	

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	FRC Operation
3	Switching from high to low	
4	Switching from high to high	

Note: * Generated on the assumption that the switchover is a falling edge; FRC is incremented.

Section 12 8-Bit Timers

12.1 Overview

The H8/3577 Group and H8/3567 Group have an on-chip 8-bit timer module with two channels (TMR_0 and TMR_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-matches. The 8-bit timer module can be used as a multifunction timer in a variety of applications, such as generation of a rectangular-wave output with an arbitrary duty cycle.

The H8/3577 Group and H8/3567 Group also have two similar 8-bit timer channels (TMRX and TMRY) that can be used in a connected configuration using the timer connection function. TMRX and TMRY have greater input/output and interrupt function related restrictions than TMR_0 and TMR_1.

12.1.1 Features

- Selection of clock sources
  - TMR_0, TMR_1: The counter input clock can be selected from six internal clocks and an external clock (enabling use as an external event counter).
  - TMRX, TMRY: The counter input clock can be selected from three internal clocks and an external clock (enabling use as an external event counter).
- Selection of three ways to clear the counters
  - The counters can be cleared on compare-match A or B, or by an external reset signal.
- Timer output controlled by two compare-match signals
  - The timer output signal in each channel is controlled by two independent compare-match signals, enabling the timer to be used for various applications, such as the generation of pulse output or PWM output with an arbitrary duty cycle.

    (Note: TMRY does not have a timer output pin.)
- Cascading of the two channels (TMR_0, TMR_1)
  - Operation as a 16-bit timer can be performed using channel 0 as the upper half and channel 1 as the lower half (16-bit count mode).
  - Channel 1 can be used to count channel 0 compare-match occurrences (compare-match count mode).
- Multiple interrupt sources for each channel
  - TMR_0, TMR_1, TMRY: Two compare-match interrupts and one overflow interrupt can be requested independently.
  - TMRX: One input capture source is available.

12.1.2 Block Diagram

Figure 12.1 shows a block diagram of the 8-bit timer module (TMR_0 and TMR_1).

TMRX and TMRY have a similar configuration, but cannot be cascaded. TMRX also has an input capture function. For details, see section 13, Timer Connection.

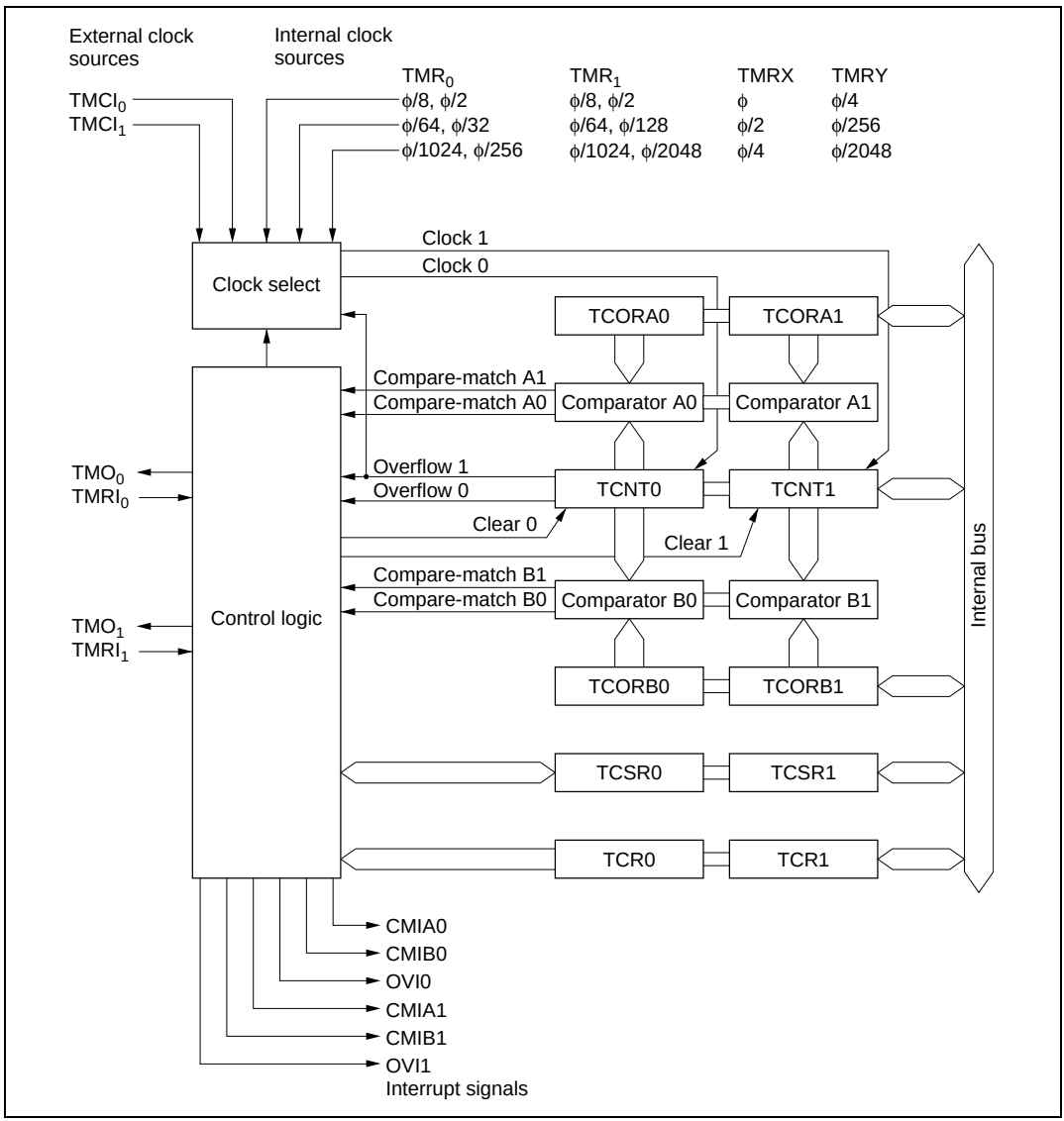


Figure 12.1 Block Diagram of 8-Bit Timer Module

12.1.3 Pin Configuration

Table 12.1 summarizes the input and output pins of the 8-bit timer module.

Table 12.1 8-Bit Timer Input and Output Pins

Channel	Name	Symbol*	I/O	Function
0	Timer output	TMO_0	Output	Output controlled by compare-match
	Timer clock input	$TMCI_0$	Input	External clock input for the counter
	Timer reset input	$TMRI_0$	Input	External reset input for the counter
1	Timer output	TMO_1	Output	Output controlled by compare-match
	Timer clock input	$TMCI_1$	Input	External clock input for the counter
	Timer reset input	$TMRI_1$	Input	External reset input for the counter
X	Timer output	TMOX	Output	Output controlled by compare-match
	Timer clock/ reset input	HFBACKI/TMIX (TMCIX/TMRIX)	Input	External clock/reset input for the counter
Y	Timer clock/reset input	VSYNCI/TMIY (TMCIY/TMRIY)	Input	External clock/reset input for the counter

Note: * The abbreviations TMO, TMCI, and TMRI are used in the text, omitting the channel number.

Channel X and Y I/O pins have the same internal configuration as channels 0 and 1, and therefore the same abbreviations are used.

12.1.4 Register Configuration

Table 12.2 summarizes the registers of the 8-bit timer module.

Table 12.2 8-Bit Timer Registers

Channel	Name	Abbreviation*2	R/W	Initial value	Address
0	Timer control register 0	TCR0	R/W	H'00	H'FFC8
	Timer control/status register 0	TCSR0	R/(W)*1	H'00	H'FFCA
	Time constant register A0	TCORA0	R/W	H'FF	H'FFCC
	Time constant register B0	TCORB0	R/W	H'FF	H'FFCE
	Time counter 0	TCNT0	R/W	H'00	H'FFD0
1	Timer control register 1	TCR1	R/W	H'00	H'FFC9
	Timer control/status register 1	TCSR1	R/(W)*1	H'10	H'FFCB
	Time constant register A1	TCORA1	R/W	H'FF	H'FFCD
	Time constant register B1	TCORB1	R/W	H'FF	H'FFCF
	Timer counter 1	TCNT1	R/W	H'00	H'FFD1
Common	Serial timer control register	STCR	R/W	H'00	H'FFC3
	Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
		MSTPCRL	R/W	H'FF	H'FF87
	Timer connection register S	TCONRS	R/W	H'00	H'FFFE
X	Timer control register X	TCRX	R/W	H'00	H'FFF0
	Timer control/status register X	TCSRX	R/(W)*1	H'00	H'FFF1
	Time constant register AX	TCORAX	R/W	H'FF	H'FFF6
	Time constant register BX	TCORBX	R/W	H'FF	H'FFF7
	Timer counter X	TCNTX	R/W	H'00	H'FFF4
	Time constant register C	TCORC	R/W	H'FF	H'FFF5
	Input capture register R	TICRR	R	H'00	H'FFF2
	Input capture register F	TICRF	R	H'00	H'FFF3
Y	Timer control register Y	TCRY	R/W	H'00	H'FFF0
	Timer control/status register Y	TCSRY	R/(W)*1	H'00	H'FFF1
	Time constant register AY	TCORAY	R/W	H'FF	H'FFF2
	Time constant register BY	TCORBY	R/W	H'FF	H'FFF3
	Timer counter Y	TCNTY	R/W	H'00	H'FFF4
	Timer input select register	TISR	R/W	H'FE	H'FFF5

Notes: 1. Only 0 can be written in bits 7 to 5, to clear these flags.
2. The abbreviations TCR, TCSR, TCORA, TCORB, and TCNT are used in the text, omitting the channel designation (0, 1, X, or Y).

Each pair of registers for channel 0 and channel 1 comprises a 16-bit register with the upper 8 bits for channel 0 and the lower 8 bits for channel 1, so they can be accessed together by word access. (Access is not divided into two 8-bit accesses.)

Certain of the channel X and channel Y registers are assigned to the same address. The TMRX/Y bit in TCONRS determines which register is accessed.

12.2 Register Descriptions

12.2.1 Timer Counter (TCNT)

	TCNT0								TCNT1							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNTX, TCNTY

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

Each TCNT is an 8-bit readable/writable up-counter.

TCNT0 and TCNT1 comprise a single 16-bit register, so they can be accessed together by word access.

TCNT increments on pulses generated from an internal or external clock source. This clock source is selected by clock select bits CKS2 to CKS0 in TCR.

TCNT can be cleared by an external reset input signal or compare-match signal. Counter clear bits CCLR1 and CCLR0 in TCR select the method of clearing.

When TCNT overflows from H'FF to H'00, the overflow flag (OVF) in TCSR is set to 1.

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

12.2.2 Time Constant Register A (TCORA)

	TCORA0								TCORA1							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCORAX, TCORAY

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

TCORA is an 8-bit readable/writable register.

TCORA0 and TCORA1 comprise a single 16-bit register, so they can be accessed together by word access.

TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag A (CMFA) in TCSR is set. Note, however, that comparison is disabled during the T2 state of a TCORA write cycle.

The timer output can be freely controlled by these compare-match signals and the settings of output select bits OS1 and OS0 in TCSR.

TCORA is initialized to H'FF by a reset and in hardware standby mode.

12.2.3 Time Constant Register B (TCORB)

	TCORB0								TCORB1							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCORBX, TCORBY

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

TCORB is an 8-bit readable/writable register. TCORB0 and TCORB1 comprise a single 16-bit register, so they can be accessed together by word access.

TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare-match flag B (CMFB) in TCSR is set. Note, however, that comparison is disabled during the T2 state of a TCORB write cycle.

The timer output can be freely controlled by these compare-match signals and the settings of output select bits OS3 and OS2 in TCSR.

TCORB is initialized to H'FF by a reset and in hardware standby mode.

12.2.4 Timer Control Register (TCR)

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCR is an 8-bit readable/writable register that selects the clock source and the time at which TCNT is cleared, and enables interrupts.

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

For details of the timing, see section 12.3, Operation.

Bit 7—Compare-Match Interrupt Enable B (CMIEB): Selects whether the CMFB interrupt request (CMIB) is enabled or disabled when the CMFB flag in TCSR is set to 1.

Note that a CMIB interrupt is not requested by TMRX, regardless of the CMIEB value.

Bit 7

CMIEB	Description	
0	CMFB interrupt request (CMIB) is disabled	(Initial value)
1	CMFB interrupt request (CMIB) is enabled	

Bit 6—Compare-Match Interrupt Enable A (CMIEA): Selects whether the CMFA interrupt request (CMIA) is enabled or disabled when the CMFA flag in TCSR is set to 1.

Note that a CMIA interrupt is not requested by TMRX, regardless of the CMIEA value.

Bit 6

CMIEA	Description	
0	CMFA interrupt request (CMIA) is disabled	(Initial value)
1	CMFA interrupt request (CMIA) is enabled	

Bit 5—Timer Overflow Interrupt Enable (OVIE): Selects whether the OVF interrupt request (OVI) is enabled or disabled when the OVF flag in TCSR is set to 1.

Note that an OVI interrupt is not requested by TMRX, regardless of the OVIE value.

Bit 5		
OVIE	**Description**	
0	OVF interrupt request (OVI) is disabled	(Initial value)
1	OVF interrupt request (OVI) is enabled	

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

Bit 4	Bit 3		
CCLR1	**CCLR0**	**Description**	
0	0	Clearing is disabled	(Initial value)
	1	Cleared on compare-match A	
1	0	Cleared on compare-match B	
	1	Cleared on rising edge of external reset input	

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select whether the clock input to TCNT is an internal or external clock.

The input clock can be selected from either six or three clocks, all divided from the system clock (φ). The falling edge of the selected internal clock triggers the count.

When use of an external clock is selected, three types of count can be selected: at the rising edge, the falling edge, and both rising and falling edges.

Some functions differ between channel 0 and channel 1, because of the cascading function.

	TCR			STCR		
	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
Channel	CKS2	CKS1	CKS0	ICKS1	ICKS0	Description
0	0	0	0	—	—	Clock input disabled (Initial value)
	0	0	1	—	0	φ/8 internal clock source, counted on the falling edge
	0	0	1	—	1	φ/2 internal clock source, counted on the falling edge
	0	1	0	—	0	φ/64 internal clock source, counted on the falling edge
	0	1	0	—	1	φ/32 internal clock source, counted on the falling edge
	0	1	1	—	0	φ/1024 internal clock source, counted on the falling edge
	0	1	1	—	1	φ/256 internal clock source, counted on the falling edge
	1	0	0	—	—	Counted on TCNT1 overflow signal*
1	0	0	0	—	—	Clock input disabled (Initial value)
	0	0	1	0	—	φ/8 internal clock source, counted on the falling edge
	0	0	1	1	—	φ/2 internal clock source, counted on the falling edge
	0	1	0	0	—	φ/64 internal clock source, counted on the falling edge
	0	1	0	1	—	φ/128 internal clock source, counted on the falling edge
	0	1	1	0	—	φ/1024 internal clock source, counted on the falling edge
	0	1	1	1	—	φ/2048 internal clock source, counted on the falling edge
	1	0	0	—	—	Counted on TCNT0 compare-match A*

Note: * If the count input of channel 0 is the TCNT1 overflow signal and that of channel 1 is the TCNT0 compare-match signal, no incrementing clock will be generated. Do not use this setting.

	TCR			STCR		
	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
Channel	CKS2	CKS1	CKS0	ICKS1	ICKS0	Description
X	0	0	0	—	—	Clock input disabled (Initial value)
	0	0	1	—	—	Counted on ϕ internal clock source
	0	1	0	—	—	ϕ/2 internal clock source, counted on the falling edge
	0	1	1	—	—	ϕ/4 internal clock source, counted on the falling edge
	1	0	0	—	—	Clock input disabled
Y	0	0	0	—	—	Clock input disabled (Initial value)
	0	0	1	—	—	ϕ/4 internal clock source, counted on the falling edge
	0	1	0	—	—	ϕ/256 internal clock source, counted on the falling edge
	0	1	1	—	—	ϕ/2048 internal clock source, counted on the falling edge
	1	0	0	—	—	Clock input disabled
Common	1	0	1	—	—	External clock source, counted at rising edge
	1	1	0	—	—	External clock source, counted at falling edge
	1	1	1	—	—	External clock source, counted at both rising and falling edges

12.2.5 Timer Control/Status Register (TCSR)

TCSR0

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

TCSR1

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/W

TCSRX

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W

TCSRY

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written in bits 7 to 5, and in bit 4 in TCSRX, to clear these flags.

TCSR is an 8-bit register that indicates compare-match and overflow statuses (and input capture status in TMRX only), and controls compare-match output.

TCSR0, TCSRX, and TCSRY are initialized to H'00, and TCSR1 is initialized to H'10, by a reset and in hardware standby mode.

Bit 7—Compare-Match Flag B (CMFB): Status flag indicating whether the values of TCNT and TCORB match.

Bit 7

CMFB	Description	
0	[Clearing condition]	(Initial value)
	Read CMFB when CMFB = 1, then write 0 in CMFB	
1	[Setting condition]	
	When TCNT = TCORB	

Bit 6—Compare-match Flag A (CMFA): Status flag indicating whether the values of TCNT and TCORA match.

Bit 6

CMFA	Description	
0	[Clearing condition]	(Initial value)
	Read CMFA when CMFA = 1, then write 0 in CMFA	
1	[Setting condition]	
	When TCNT = TCORA	

Bit 5 —Timer Overflow Flag (OVF): Status flag indicating that TCNT has overflowed (changed from H'FF to H'00).

Bit 5

OVF	Description	
0	[Clearing condition]	(Initial value)
	Read OVF when OVF = 1, then write 0 in OVF	
1	[Setting condition]	
	When TCNT overflows from H'FF to H'00	

TCSR0 Bit 4—A/D Trigger Enable (ADTE): Enables or disables A/D converter start requests by compare-match A.

TCSR0 Bit 4

ADTE	Description	
0	A/D converter start requests by compare-match A are disabled	(Initial value)
1	A/D converter start requests by compare-match A are enabled	

TCSR1 Bit 4—Reserved: This bit cannot be modified and is always read as 1.

TCSRX Bit 4—Input Capture Flag (ICF): Status flag that indicates detection of a rising edge followed by a falling edge in the external reset signal after the ICST bit in TCONRI has been set to 1.

TCSRX Bit 4

ICF	Description	
0	[Clearing condition] Read ICF when ICF = 1, then write 0 in ICF	(Initial value)
1	[Setting condition] When a rising edge followed by a falling edge is detected in the external reset signal after the ICST bit in TCONRI has been set to 1	

TCSRY Bit 4—Input Capture Interrupt Enable (ICIE): Selects enabling or disabling of the interrupt request by ICF (ICIX) when the ICF bit in TCSRX is set to 1.

TCSRY Bit 4

ICIE	Description	
0	Interrupt request by ICF (ICIX) is disabled	(Initial value)
1	Interrupt request by ICF (ICIX) is enabled	

Bits 3 to 0—Output Select 3 to 0 (OS3 to OS0): These bits specify how the timer output level is to be changed by a compare-match of TCOR and TCNT.

OS3 and OS2 select the effect of compare-match B on the output level, OS1 and OS0 select the effect of compare-match A on the output level, and both of them can be controlled independently.

Note, however, that priorities are set such that: trigger output > 1 output > 0 output. If compare-matches occur simultaneously, the output changes according to the compare-match with the higher priority.

Timer output is disabled when bits OS3 to OS0 are all 0.

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

Bit 3	Bit 2		
OS3	**OS2**	**Description**	
0	0	No change when compare-match B occurs	(Initial value)
	1	0 is output when compare-match B occurs	
1	0	1 is output when compare-match B occurs	
	1	Output is inverted when compare-match B occurs (toggle output)	

Bit 1	Bit 0		
OS1	**OS0**	**Description**	
0	0	No change when compare-match A occurs	(Initial value)
	1	0 is output when compare-match A occurs	
1	0	1 is output when compare-match A occurs	
	1	Output is inverted when compare-match A occurs (toggle output)	

12.2.6 Serial Timer Control Register (STCR)

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

STCR is an 8-bit readable/writable register that controls register access, the IIC operating mode (when the on-chip IIC option is included), and on-chip flash memory (in F-ZTAT versions), and also selects the TCNT input clock.

For details on functions not related to the 8-bit timers, see section 3.2.3, Serial Timer Control Register (STCR), and the descriptions of the relevant modules. If a module controlled by STCR is not used, do not write 1 to the corresponding bit.

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

Bit 7—Reserved: Do not write 1 to this bit.

Bits 6 to 4—I²C Control (IICX1, IICX0, IICE): These bits control the operation of the I²C bus interface when the IIC option is included on-chip. See section 16, I²C Bus Interface, for details.

Bit 3—Reserved: This bit must not be set to 1.

Bit 2—USB Enable (USBE): This bit controls CPU access to the USB data register and control register.

Bit 2

USBE	Description	
0	Prohibition of the above register access	(initial value)
1	Permission of the above register access	

Bits 1 and 0—Internal Clock Select 1 and 0 (ICKS1, ICKS0): These bits, together with bits CKS2 to CKS0 in TCR, select the clock to be input to TCNT. For details, see section 12.2.4, Timer Control Register.

12.2.7 System Control Register (SYSCR)

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

Only bit 1 is described here. For details on functions not related to the 8-bit timers, see sections 3.2.2 and 5.2.1, System Control Register (SYSCR), and the descriptions of the relevant modules.

Bit 1—Host Interface Enable (HIE): Controls CPU access to 8-bit timer (channel X and Y) data registers and control registers, and timer connection control registers.

Bit 1

HIE	Description
0	CPU access to 8-bit timer (channel X and Y) data registers and control registers, and timer connection control registers, is enabled (Initial value)
1	CPU access to 8-bit timer (channel X and Y) data registers and control registers, and timer connection control registers, is disabled

12.2.8 Timer Connection Register S (TCONRS)

Bit	7	6	5	4	3	2	1	0
	TMRX/Y	ISGENE	HOMOD1	HOMOD0	VOMOD1	VOMOD0	CLMOD1	CLMOD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCONRS is an 8-bit readable/writable register that controls access to the TMRX and TMRY registers and timer connection operation.

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

Bit 7—TMRX/TMRY Access Select (TMRX/Y): The TMRX and TMRY registers can only be accessed when the HIE bit in SYSCR is cleared to 0. In the H8/3577 Group and H8/3567 Group, some of the TMRX registers and the TMRY registers are assigned to the same memory space addresses (H'FFF0 to H'FFF5), and the TMRX/Y bit determines which registers are accessed.

Bit 7	Accessible Registers							
TMRX/Y	**H'FFF0**	**H'FFF1**	**H'FFF2**	**H'FFF3**	**H'FFF4**	**H'FFF5**	**H'FFF6**	**H'FFF7**
0 (Initial value)	TCRX (TMRX)	TCSRX (TMRX)	TICRR (TMRX)	TICRF (TMRX)	TCNTX (TMRX)	TCORC (TMRX)	TCORAX (TMRX)	TCORBX (TMRX)
1	TCRY (TMRY)	TCSRY (TMRY)	TCORAY (TMRY)	TCORBY (TMRY)	TCNTY (TMRY)	TISR (TMRY)		

12.2.9 Input Capture Register (TICR) [TMRX Additional Function]

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	—	—	—

TICR is an 8-bit internal register to which the contents of TCNT are transferred on the falling edge of external reset input. The CPU cannot read or write to TICR directly.

The TICR function is used in timer connection. For details, see section 13, Timer Connection.

12.2.10 Time Constant Register C (TCORC) [TMRX Additional Function]

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

TCORC is an 8-bit readable/writable register. The sum of the contents of TCORC and TICR is continually compared with the value in TCNT. When a match is detected, a compare-match C signal is generated. Note, however, that comparison is disabled during the T2 state of a TCORC write cycle and a TICR input capture cycle.

TCORC is initialized to H'FF by a reset and in hardware standby mode.

The TCORC function is used in timer connection. For details, see section 13, Timer Connection.

12.2.11 Input Capture Registers R and F (TICRR, TICRF) [TMRX Additional Functions]

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TICRR and TICRF are 8-bit read-only registers. When the ICST bit in TCONRI is set to 1, TICRR and TICRF capture the contents of TCNT successively on the rise and fall of the external reset input. When one capture operation ends, the ICST bit is cleared to 0.

TICRR and TICRF are each initialized to H'00 by a reset and in hardware standby mode.

The TICRR and TICRF functions are used in timer connection. For details, see section 13, Timer Connection.

12.2.12 Timer Input Select Register (TISR) [TMRY Additional Function]

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

TISR is an 8-bit readable/writable register that selects the external clock/reset signal source for the counter.

TISR is initialized to H'FE by a reset and in hardware standby mode.

Bits 7 to 1—Reserved: Do not write 0.

Bit 0—Input Select (IS): Selects the internal synchronization signal (IVG signal) or the timer clock/reset input pin (TMIY (TMCIY/TMRIY)) as the external clock/reset signal source for the counter.

Bit 0

IS	Description	
0	IVG signal is selected	(Initial value)
1	TMIY (TMCIY/TMRIY) is selected	

12.2.13 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers, and is used to perform module stop mode control.

When the MSTP12 bit or MSTP8 bit is set to 1, at the end of the bus cycle 8-bit timer operation is halted on channels 0 and 1 or channels X and Y, respectively, and a transition is made to module stop mode. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 4—Module Stop (MSTP12): Specifies 8-bit timer (channel 0/1) module stop mode.

MSTPCRH Bit 4	
MSTP12	**Description**
0	8-bit timer (channel 0/1) module stop mode is cleared
1	8-bit timer (channel 0/1) module stop mode is set (Initial value)

MSTPCRH Bit 0—Module Stop (MSTP8): Specifies 8-bit timer (channel X/Y) and timer connection module stop mode.

MSTPCRH Bit 0	
MSTP8	**Description**
0	8-bit timer (channel X/Y) and timer connection module stop mode is cleared
1	8-bit timer (channel X/Y) and timer connection module stop mode is set (Initial value)

12.3 Operation

12.3.1 TCNT Incrementation Timing

TCNT is incremented by input clock pulses (either internal or external).

Internal Clock: An internal clock created by dividing the system clock (ϕ) can be selected by setting bits CKS2 to CKS0 in TCR. Figure 12.2 shows the count timing.

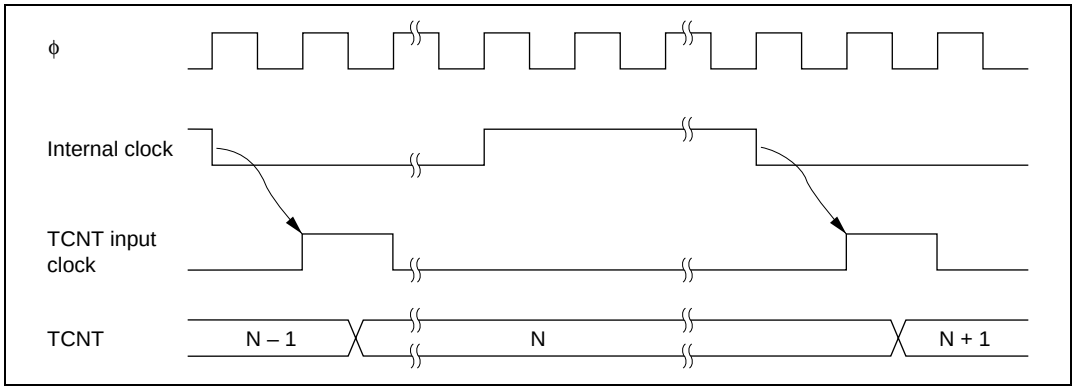


Figure 12.2 Count Timing for Internal Clock Input

External Clock: Three incrementation methods can be selected by setting bits CKS2 to CKS0 in TCR: at the rising edge, the falling edge, and both rising and falling edges.

Note that the external clock pulse width must be at least 1.5 states for incrementation at a single edge, and at least 2.5 states for incrementation at both edges. The counter will not increment correctly if the pulse width is less than these values.

Figure 12.3 shows the timing of incrementation at both edges of an external clock signal.

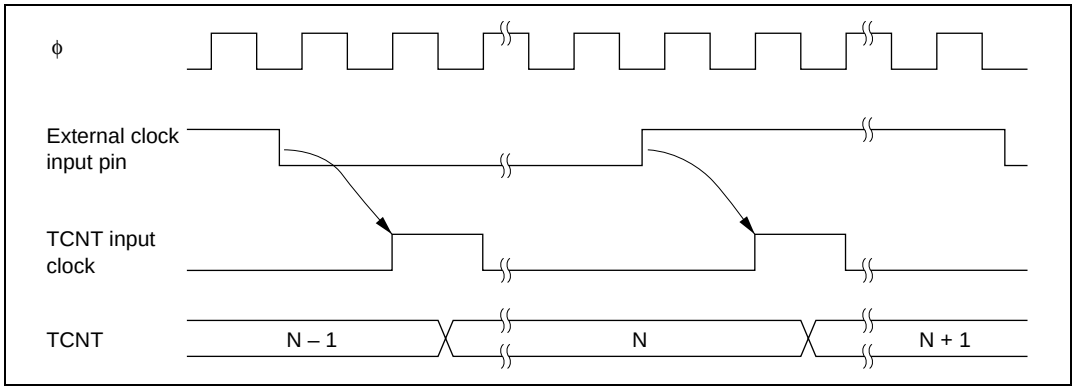


Figure 12.3 Count Timing for External Clock Input

12.3.2 Compare-Match Timing

Setting of Compare-Match Flags A and B (CMFA, CMFB): The CMFA and CMFB flags in TCSR are set to 1 by a compare-match signal generated when the TCOR and TCNT values match. The compare-match signal is generated at the last state in which the match is true, just before the timer counter is updated.

Therefore, when TCOR and TCNT match, the compare-match signal is not generated until the next incrementation clock input. Figure 12.4 shows this timing.

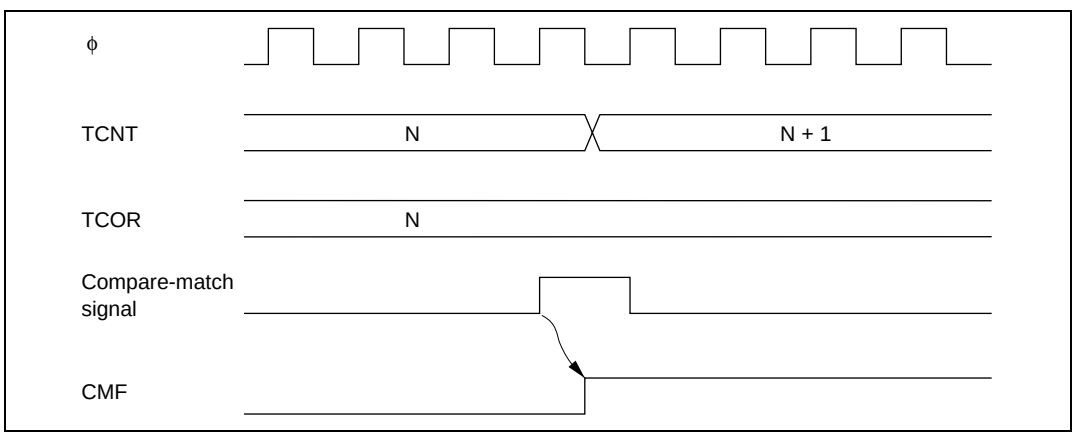


Figure 12.4 Timing of CMF Setting

Timer Output Timing: When compare-match A or B occurs, the timer output changes as specified by the output select bits (OS3 to OS0) in TCSR. Depending on these bits, the output can remain the same, be set to 0, be set to 1, or toggle.

Figure 12.5 shows the timing when the output is set to toggle at compare-match A.

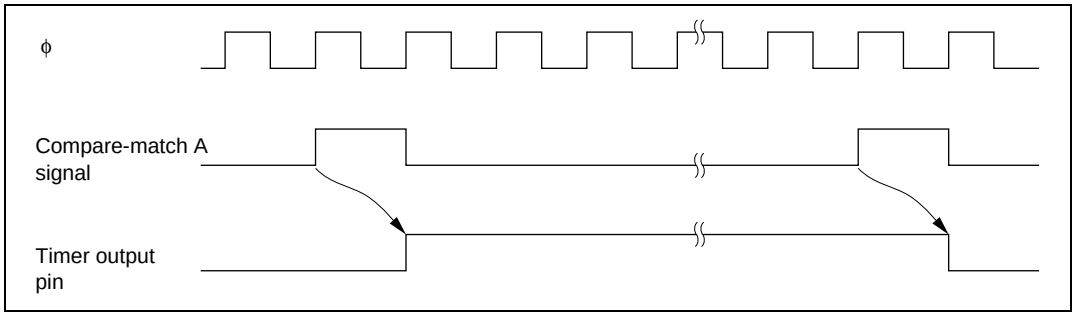


Figure 12.5 Timing of Timer Output

Timing of Compare-Match Clear: TCNT is cleared when compare-match A or B occurs, depending on the setting of the CCLR1 and CCLR0 bits in TCR. Figure 12.6 shows the timing of this operation.

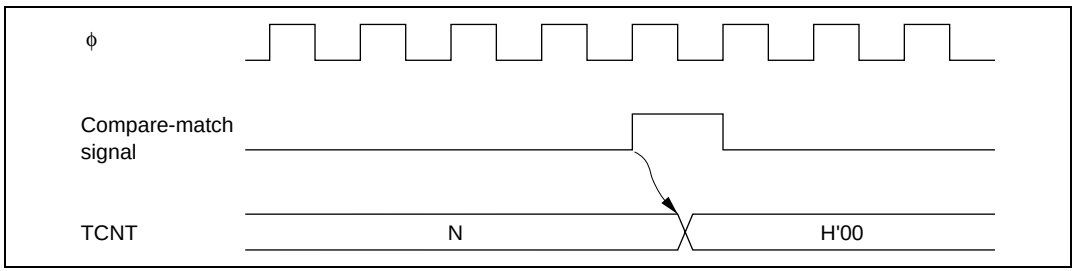


Figure 12.6 Timing of Compare-Match Clear

12.3.3 TCNT External Reset Timing

TCNT is cleared at the rising edge of an external reset input, depending on the settings of the CCLR1 and CCLR0 bits in TCR. The width of the clearing pulse must be at least 1.5 states. Figure 12.7 shows the timing of this operation.

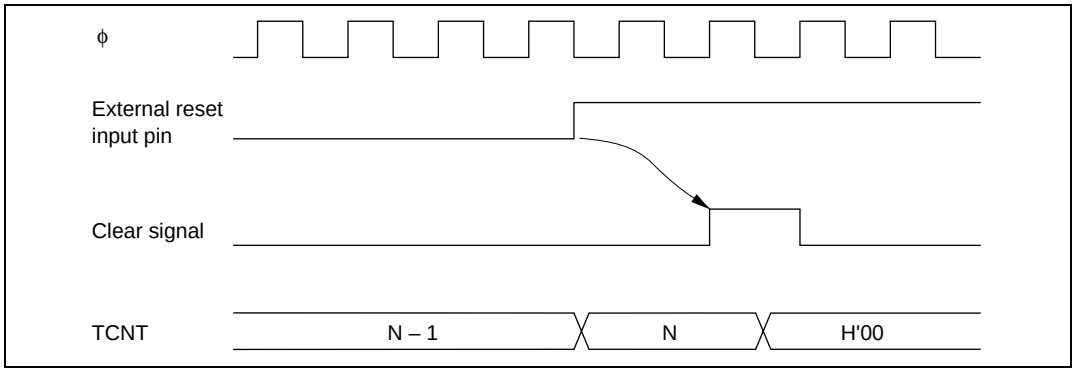


Figure 12.7 Timing of Clearing by External Reset Input

12.3.4 Timing of Overflow Flag (OVF) Setting

OVF in TCSR is set to 1 when the timer count overflows (changes from H'FF to H'00). Figure 12.8 shows the timing of this operation.

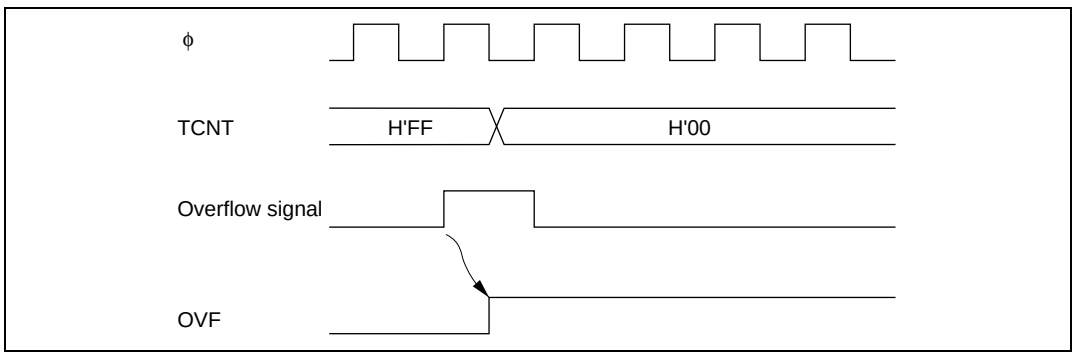


Figure 12.8 Timing of OVF Setting

12.3.5 Operation with Cascaded Connection

If bits CKS2 to CKS0 in either TCR0 or TCR1 are set to B'100, the 8-bit timers of the two channels are cascaded. With this configuration, a single 16-bit timer can be used (16-bit timer mode) or compare-matches of 8-bit channel 0 can be counted by the timer of channel 1 (compare-match count mode). In this case, the timer operates as described below.

16-Bit Count Mode: When bits CKS2 to CKS0 in TCR0 are set to B'100, the timer functions as a single 16-bit timer with channel 0 occupying the upper 8 bits and channel 1 occupying the lower 8 bits.

- Setting of compare-match flags
  - The CMF flag in TCSR0 is set to 1 when a 16-bit compare-match occurs.
  - The CMF flag in TCSR1 is set to 1 when a lower 8-bit compare-match occurs.
- Counter clear specification
  - If the CCLR1 and CCLR0 bits in TCR0 have been set for counter clear at compare-match, the 16-bit counter (TCNT0 and TCNT1 together) is cleared when a 16-bit compare-match occurs. The 16-bit counter (TCNT0 and TCNT1 together) is cleared even if counter clear by the TMRI0 pin has also been set.
  - The settings of the CCLR1 and CCLR0 bits in TCR1 are ignored. The lower 8 bits cannot be cleared independently.
- Pin output
  - Control of output from the TMO0 pin by bits OS3 to OS0 in TCSR0 is in accordance with the 16-bit compare-match conditions.
  - Control of output from the TMO1 pin by bits OS3 to OS0 in TCSR1 is in accordance with the lower 8-bit compare-match conditions.

Compare-Match Count Mode: When bits CKS2 to CKS0 in TCR1 are B'100, TCNT1 counts compare-match A's for channel 0.

Channels 0 and 1 are controlled independently. Conditions such as setting of the CMF flag, generation of interrupts, output from the TMO pin, and counter clearing are in accordance with the settings for each channel.

Usage Note: If the 16-bit count mode and compare-match count mode are set simultaneously, the input clock pulses for TCNT0 and TCNT1 are not generated and thus the counters will stop operating. Simultaneous setting of these two modes should therefore be avoided.

12.4 Interrupt Sources

The TMR_0, TMR_1, and TMRY 8-bit timers can generate three types of interrupt: compare-match A and B (CMIA and CMIB), and overflow (OVI). TMRX can generate only an ICIX interrupt. An interrupt is requested when the corresponding interrupt enable bit is set in TCR or TCSR. Independent signals are sent to the interrupt controller for each interrupt.

An overview of 8-bit timer interrupt sources is given in tables 12.3 to 12.5.

Table 12.3 TMR_0 and TMR_1 8-Bit Timer Interrupt Sources

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

Table 12.4 TMRX 8-Bit Timer Interrupt Source

Interrupt source	Description
ICIX	Requested by ICF

Table 12.5 TMRY 8-Bit Timer Interrupt Sources

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

12.5 8-Bit Timer Application Example

In the example below, the 8-bit timer is used to generate a pulse output with a selected duty cycle, as shown in figure 12.9. The control bits are set as follows:

- In TCR, CCLR1 is cleared to 0 and CCLR0 is set to 1 so that the timer counter is cleared by a TCORA compare-match.
- In TCSR, bits OS3 to OS0 are set to B'0110, causing 1 output at a TCORA compare-match and 0 output at a TCORB compare-match.

With these settings, the 8-bit timer provides output of pulses at a rate determined by TCORA with a pulse width determined by TCORB. No software intervention is required.

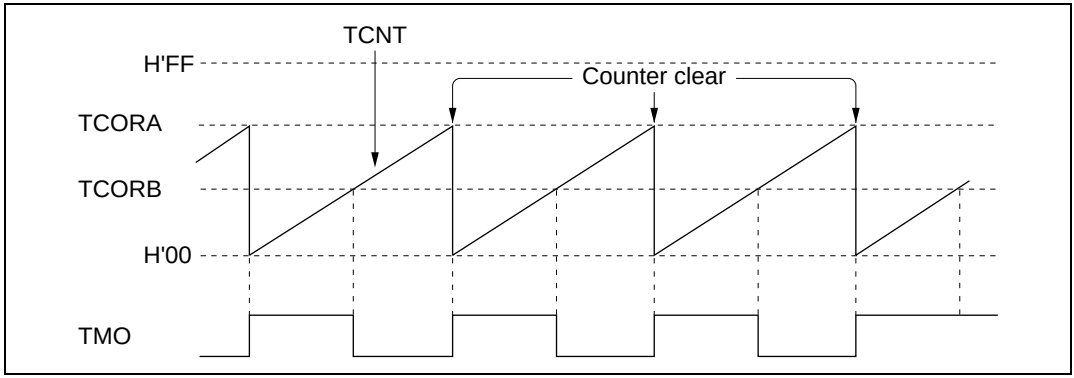


Figure 12.9 Pulse Output (Example)

12.6 Usage Notes

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

12.6.1 Contention between TCNT Write and Clear

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the clear takes priority, so that the counter is cleared and the write is not performed. Figure 12.10 shows this operation.

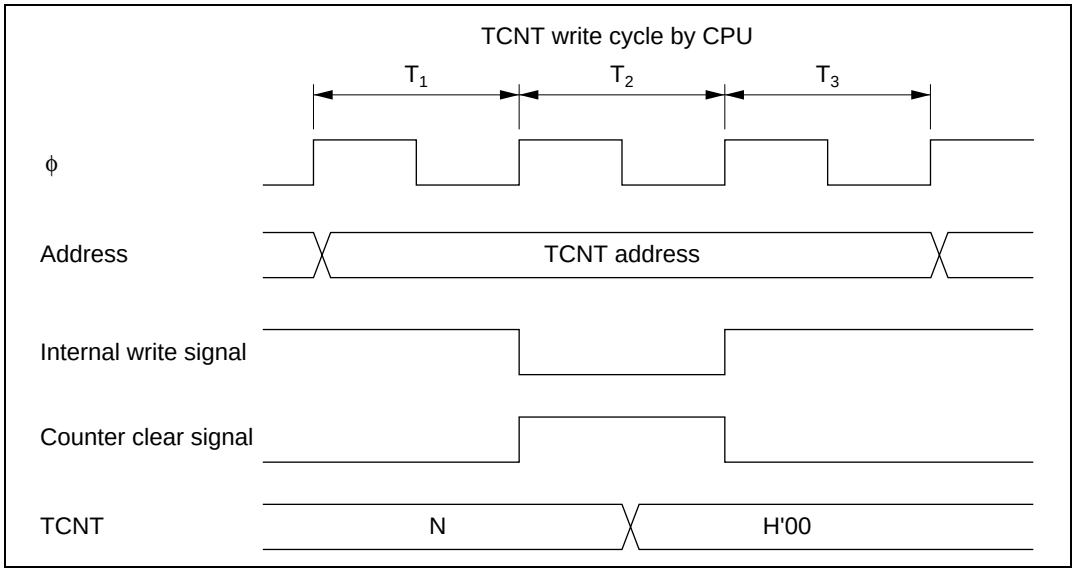


Figure 12.10 Contention between TCNT Write and Clear

12.6.2 Contention between TCNT Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the counter is not incremented. Figure 12.11 shows this operation.

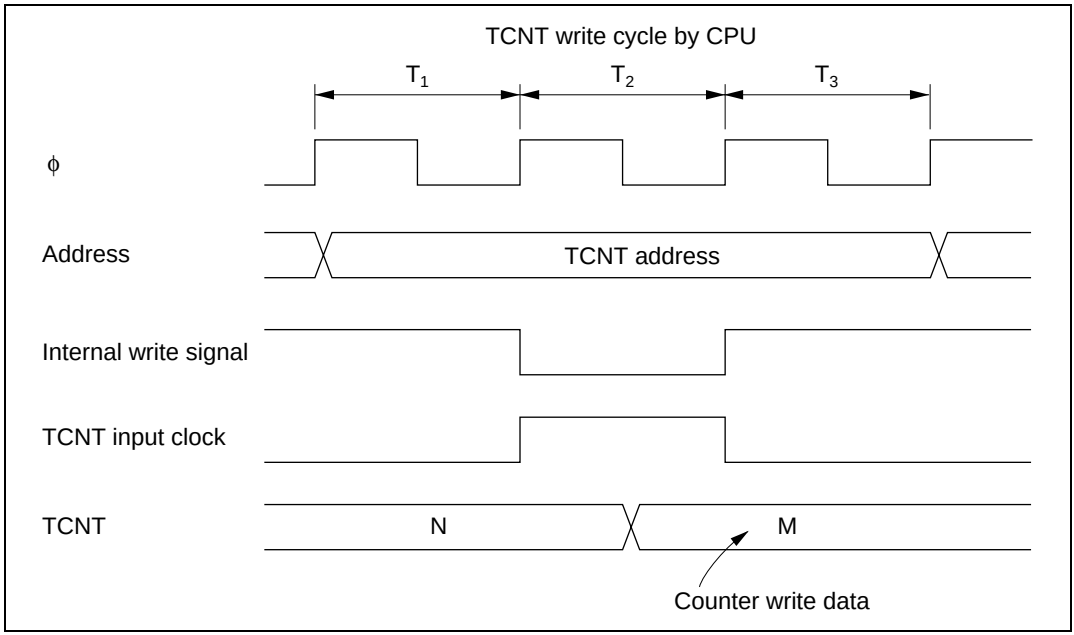


Figure 12.11 Contention between TCNT Write and Increment

12.6.3 Contention between TCOR Write and Compare-Match

During the T2 state of a TCOR write cycle, the TCOR write has priority even if a compare-match occurs and the compare-match signal is disabled. Figure 12.12 shows this operation.

With TMRX, an ICR input capture contends with a compare-match in the same way as with a write to TCORC. In this case, the input capture has priority and the compare-match signal is inhibited.

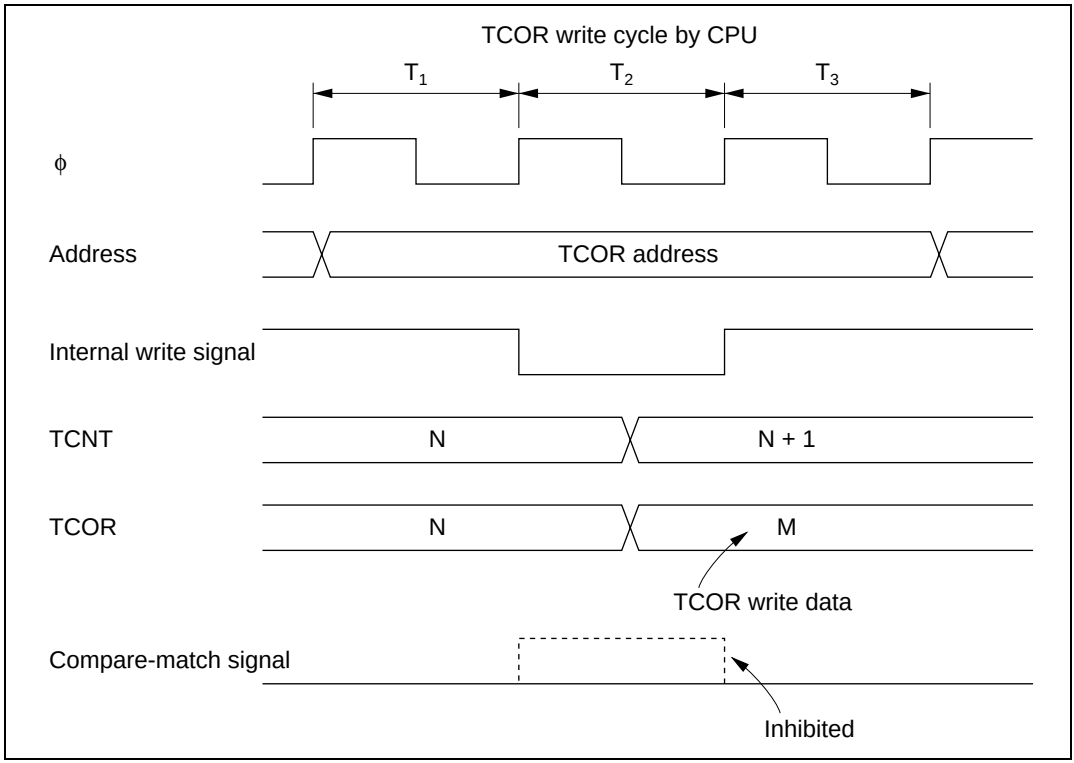


Figure 12.12 Contention between TCOR Write and Compare-Match

12.6.4 Contention between Compare-Matches A and B

If compare-matches A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output states set for compare-match A and compare-match B, as shown in table 12.6.

Table 12.6 Timer Output Priorities

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	
No change	Low

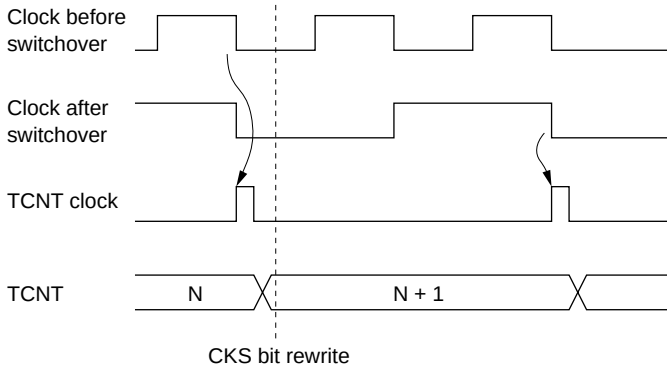
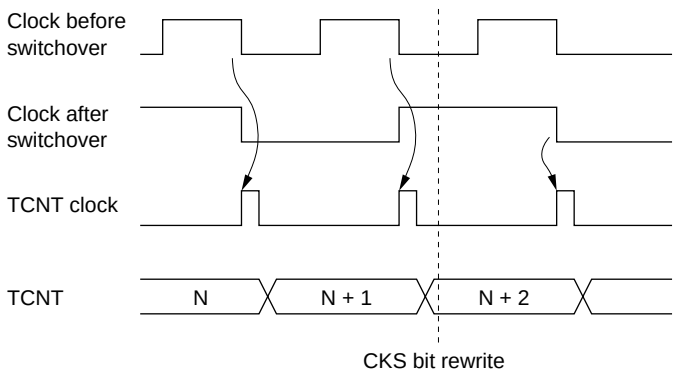
12.6.5 Switching of Internal Clocks and TCNT Operation

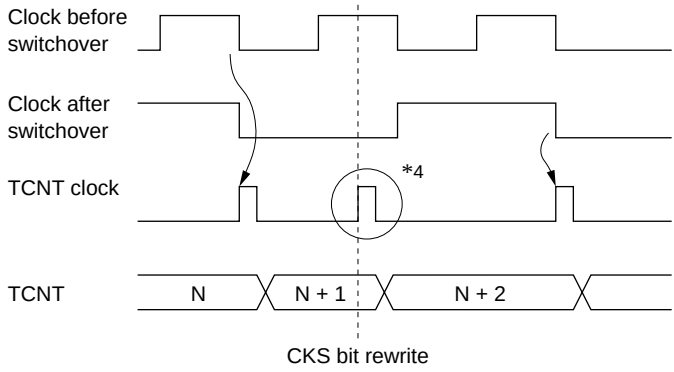
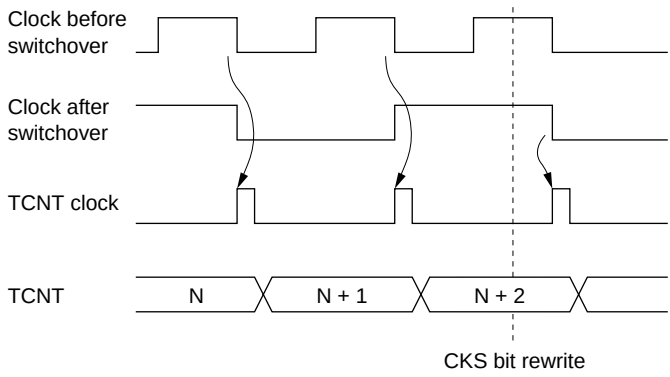
TCNT may increment erroneously when the internal clock is switched over. Table 12.7 shows the relationship between the timing at which the internal clock is switched (by writing to the CKS1 and CKS0 bits) and the TCNT operation

When the TCNT clock is generated from an internal clock, the falling edge of the internal clock pulse is detected. If clock switching causes a change from high to low level, as shown in no. 3 in table 12.7, a TCNT clock pulse is generated on the assumption that the switchover is a falling edge. This increments TCNT.

Erroneous incrementation can also happen when switching between internal and external clocks.

Table 12.7 Switching of Internal Clock and TCNT Operation

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
1	Switching from low to low*1	
2	Switching from low to high*2	

No.	Timing of Switchover by Means of CKS1 and CKS0 Bits	TCNT Clock Operation
3	Switching from high to low*3	
4	Switching from high to high	

Notes: 1. Includes switching from low to stop, and from stop to low.
2. Includes switching from stop to high.
3. Includes switching from high to stop.
4. Generated on the assumption that the switchover is a falling edge; TCNT is incremented.

Section 13 Timer Connection

13.1 Overview

The H8/3577 Group and H8/3567 Group allow interconnection between a combination of input signals, the single free-running timer (FRT) channel, and the three 8-bit timer channels (TMR_1, TMRX, and TMRY). This capability can be used to implement complex functions such as PWM decoding and clamp waveform output. All the timers are initially set for independent operation.

13.1.1 Features

The features of the timer connection facility are as follows.

- Five input pins and four output pins, all of which can be designated for phase inversion. Positive logic is assumed for all signals used within the timer connection facility.
- An edge-detection circuit is connected to the input pins, simplifying signal input detection.
- TMRX can be used for PWM input signal decoding.
- TMRX can be used for clamp waveform generation.
- An external clock signal divided by TMR_1 can be used as the FRT capture input signal.
- An internal synchronization signal can be generated using the FRT and TMRY.
- A signal generated/modified using an input signal and timer connection can be selected and output.

13.1.2 Block Diagram

Figure 13.1 shows a block diagram of the timer connection facility.

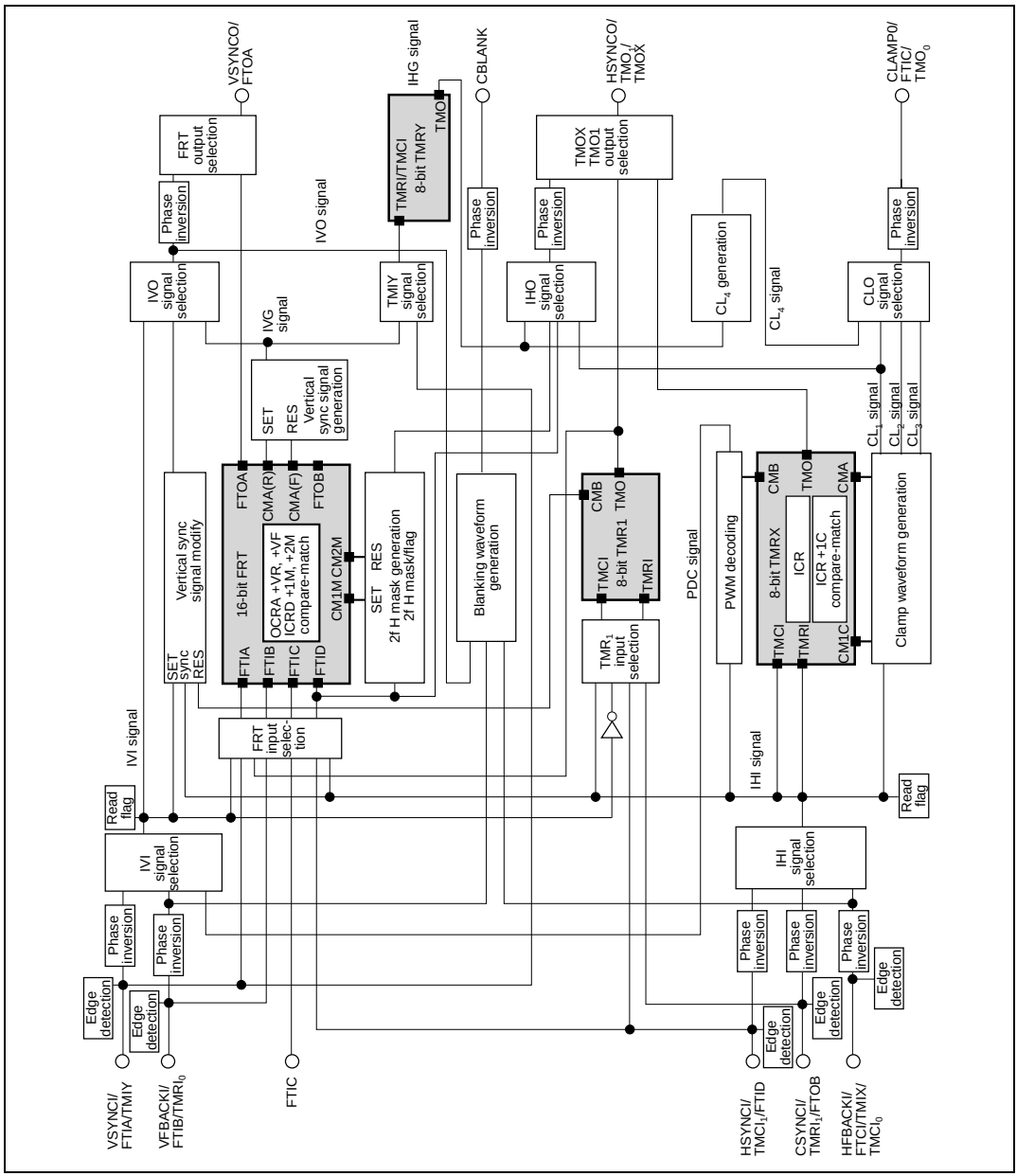


Figure 13.1 Block Diagram of Timer Connection Facility

13.1.3 Input and Output Pins

Table 13.1 lists the timer connection input and output pins.

Table 13.1 Timer Connection Input and Output Pins

Name	Abbreviation	Input/ Output	Function
Vertical synchronization signal input pin	VSYNCI	Input	Vertical synchronization signal input pin or FTIA input pin/TMIY input pin
Horizontal synchronization signal input pin	HSYNCI	Input	Horizontal synchronization signal input pin or FTID input pin/$TMCI_1$ input pin
Composite synchronization signal input pin	CSYNCI	Input	Composite synchronization signal input pin or $TMRI_1$ input pin/FTOB output pin
Spare vertical synchronization signal input pin	VFBACKI	Input	Spare vertical synchronization signal input pin or FTIB input pin/$TMRI_0$ input pin
Spare horizontal synchronization signal input pin	HFBACKI	Input	Spare horizontal synchronization signal input pin or FTCI input pin/$TMCI_0$ input pin/TMIX input pin
Vertical synchronization signal output pin	VSYNCO	Output	Vertical synchronization signal output pin or FTOA output pin
Horizontal synchronization signal output pin	HSYNCO	Output	Horizontal synchronization signal output pin or TMO_1 output pin/TMOX output pin
Clamp waveform output pin	CLAMPO	Output	Clamp waveform output pin or TMO_0 output pin/FTIC input pin
Blanking waveform output pin	CBLANK	Output	Blanking waveform output pin

13.1.4 Register Configuration

Table 13.2 lists the timer connection registers. Timer connection registers can only be accessed when the HIE bit in SYSCR is 0.

Table 13.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address
Timer connection register I	TCONRI	R/W	H'00	H'FFFC
Timer connection register O	TCONRO	R/W	H'00	H'FFFD
Timer connection register S	TCONRS	R/W	H'00	H'FFFE
Edge sense register	SEDGR	R/(W)*1	H'00*2	H'FFFF
Module stop control register	MSTPRH	R/W	H'3F	H'FF86
	MSTPRL	R/W	H'FF	H'FF87

Notes: 1. Bits 7 to 2: Only 0 can be written to clear the flags.
2. Bits 1 and 0: Undefined (reflect the pin states).

13.2 Register Descriptions

13.2.1 Timer Connection Register I (TCONRI)

Bit	7	6	5	4	3	2	1	0
	SIMOD1	SIMOD0	SCONE	ICST	HFINV	VFINV	HIINV	VIINV
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCONRI is an 8-bit readable/writable register that controls connection between timers, the signal source for synchronization signal input, phase inversion, etc.

TCONR1 is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 and 6—Input Synchronization Mode Select 1 and 0 (SIMOD1, SIMOD0): These bits select the signal source of the IHI and IVI signals.

Bit 7	Bit 6	Description		
SIMOD1	**SIMOD0**	**Mode**	**IHI Signal**	**IVI Signal**
0	0	No signal (Initial value)	HFBACKI input	VFBACKI input
	1	S-on-G mode	CSYNCI input	PDC input
1	0	Composite mode	HSYNCI input	PDC input
	1	Separate mode	HSYNCI input	VSYNCI input

Bit 5—Synchronization Signal Connection Enable (SCONE): Selects the signal source of the FRT FTI input and the TMR1 TMCI1/TMRI1 input.

Bit 5	Description						
SCONE	**Mode**	**FTIA**	**FTIB**	**FTIC**	**FTID**	**$TMCI_1$**	**$TMRI_1$**
0	Normal connection (Initial value)	FTIA input	FTIB input	FTIC input	FTID input	$TMCI_1$ input	$TMRI_1$ input
1	Synchronization signal connection mode	IVI signal	TMO_1 signal	VFBACKI input	IHI signal	IHI signal	IVI inverse signal

Bit 4—Input Capture Start Bit (ICST): The TMRX external reset input (TMRIX) is connected to the IHI signal. TMRX has input capture registers (TICR, TICRR, and TICRF). TICRR and TICRF can measure the width of a short pulse by means of a single capture operation under the control of the ICST bit. When a rising edge followed by a falling edge is detected on TMRIX after the ICST bit is set to 1, the contents of TCNT at those points are captured into TICRR and TICRF, respectively, and the ICST bit is cleared to 0.

Bit 4	
ICST	**Description**
0	The TICRR and TICRF input capture functions are stopped (Initial value) [Clearing condition] When a rising edge followed by a falling edge is detected on TMRIX
1	The TICRR and TICRF input capture functions are operating (Waiting for detection of a rising edge followed by a falling edge on TMRIX) [Setting condition] When 1 is written in ICST after reading ICST = 0

Bits 3 to 0—Input Synchronization Signal Inversion (HFINV, VFINV, HIINV, VIINV): These bits select inversion of the input phase of the spare horizontal synchronization signal (HFBACKI), the spare vertical synchronization signal (VFBACKI), the horizontal synchronization signal and composite synchronization signal (HSYNCI, CSYNCI), and the vertical synchronization signal (VSYNCI).

Bit 3

HFINV	Description	
0	The HFBACKI pin state is used directly as the HFBACKI input	(Initial value)
1	The HFBACKI pin state is inverted before use as the HFBACKI input	

Bit 2

VFINV	Description	
0	The VFBACKI pin state is used directly as the VFBACKI input	(Initial value)
1	The VFBACKI pin state is inverted before use as the VFBACKI input	

Bit 1

HIINV	Description	
0	The HSYNCI and CSYNCI pin states are used directly as the HSYNCI and CSYNCI inputs	(Initial value)
1	The HSYNCI and CSYNCI pin states are inverted before use as the HSYNCI and CSYNCI inputs	

Bit 0

VIINV	Description	
0	The VSYNCI pin state is used directly as the VSYNCI input	(Initial value)
1	The VSYNCI pin state is inverted before use as the VSYNCI input	

13.2.2 Timer Connection Register O (TCONRO)

Bit	7	6	5	4	3	2	1	0
	HOE	VOE	CLOE	CBOE	HOINV	VOINV	CLOINV	CBOINV
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCONRO is an 8-bit readable/writable register that controls output signal output, phase inversion, etc.

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

Bits 7 to 4—Output Enable (HOE, VOE, CLOE, CBOE): These bits control enabling/disabling of horizontal synchronization signal (HSYNCO), vertical synchronization signal (VSYNCO), clamp waveform (CLAMPO), and blanking waveform (CBLANK) output. When output is disabled, the state of the relevant pin is determined by the port DR and DDR, FRT, TMR, and PWM settings.

Output enabling/disabling control does not affect the port, FRT, or TMR input functions, but some FRT and TMR input signal sources are determined by the SCONE bit in TCONRI.

Bit 7

HOE	Description	
0	The $P6_7$/TMO_1/TMOX/HSYNCO pin functions as the $P6_7$/TMO_1/TMOX pin	(Initial value)
1	The $P6_7$/TMO_1/TMOX/HSYNCO pin functions as the HSYNCO pin	

Bit 6

VOE	Description	
0	The $P6_1$/FTOA/VSYNCO pin functions as the $P6_1$/FTOA pin	(Initial value)
1	The $P6_1$/FTOA/VSYNCO pin functions as the VSYNCO pin	

Bit 5

CLOE	Description	
0	The $P6_4$/FTIC/TMO_0/CLAMPO pin functions as the $P6_4$/FTIC/TMO_0 pin	(Initial value)
1	The $P6_4$/FTIC/TMO_0/CLAMPO pin functions as the CLAMPO pin	

Bit 4

CBOE	Description	
0	[H8/3577 Group] $P2_7$/PW_{15}/CBLANK pin functions as the $P2_7$/PW_{15} pin	
	[H8/3567 Group] $P1_5$/PW_5/CBLANK pin functions as the $P1_5$/PW_5 pin	(Initial value)
1	[H8/3577 Group] $P2_7$/PW_{15}/CBLANK pin functions as the CBLANK pin	
	[H8/3567 Group] $P1_5$/PW_5/CBLANK pin functions as the CBLANK pin	

Bits 3 to 0—Output Synchronization Signal Inversion (HOINV, VOINV, CLOINV, CBOINV): These bits select inversion of the output phase of the horizontal synchronization signal (HSYNCO), the vertical synchronization signal (VSYNCO), the clamp waveform (CLAMPO), and the blank waveform (CBLANK).

Bit 3

HOINV	Description	
0	The IHO signal is used directly as the HSYNCO output	(Initial value)
1	The IHO signal is inverted before use as the HSYNCO output	

Bit 2

VOINV	Description	
0	The IVO signal is used directly as the VSYNCO output	(Initial value)
1	The IVO signal is inverted before use as the VSYNCO output	

Bit 1

CLOINV	Description	
0	The CLO signal (CL_1, CL_2, CL_3, or CL_4 signal) is used directly as the CLAMPO output	(Initial value)
1	The CLO signal (CL_1, CL_2, CL_3, or CL_4 signal) is inverted before use as the CLAMPO output	

Bit 0

CBOINV	Description	
0	The CBLANK signal is used directly as the CBLANK output	(Initial value)
1	The CBLANK signal is inverted before use as the CBLANK output	

13.2.3 Timer Connection Register S (TCONRS)

Bit	7	6	5	4	3	2	1	0
	TMRX/Y	ISGENE	HOMOD1	HOMOD0	VOMOD1	VOMOD0	CLMOD1	CLMOD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCONRS is an 8-bit readable/writable register that selects 8-bit timer TMRX/TMRY access and the synchronization signal output signal source and generation method.

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

Bit 7—TMRX/TMRY Access Select (TMRX/Y): The TMRX and TMRY registers can only be accessed when the HIE bit in SYSCR is cleared to 0. In the H8/3577 Group and H8/3567 Group, some of the TMRX registers and the TMRY registers are assigned to the same memory space addresses (H'FFF0 to H'FFF5), and the TMRX/Y bit determines which registers are accessed.

Bit 7	
TMRX/Y	**Description**
0	The TMRX registers are accessed at addresses H'FFF0 to H'FFF5 (Initial value)
1	The TMRY registers are accessed at addresses H'FFF0 to H'FFF5

Bit 6—Internal Synchronization Signal Select (ISGENE): Selects internal synchronization signals (IHG, IVG, and CL_4 signals) as the signal sources for the IHO, IVO, and CLO signals.

Bits 5 and 4—Horizontal Synchronization Output Mode Select 1 and 0 (HOMOD1, HOMOD0): These bits select the signal source and generation method for the IHO signal.

Bit 6	Bit 5	Bit 4	
ISGENE	**VOMOD1**	**VOMOD0**	**Description**
0	0	0	The IHI signal (without 2fH modification) is selected (Initial value)
		1	The IHI signal (with 2fH modification) is selected
	1	0	The CL_1 signal is selected
		1	
1	0	0	The IHG signal is selected
		1	
	1	0	
		1	

Bits 3 and 2—Vertical Synchronization Output Mode Select 1 and 0 (VOMOD1, VOMOD0): These bits select the signal source and generation method for the IVO signal.

Bit 6	Bit 3	Bit 2		
ISGENE	**VOMOD1**	**VOMOD0**	**Description**	
0	0	0	The IVI signal (without fall modification or IHI synchronization) is selected	(Initial value)
		1	The IVI signal (without fall modification, with IHI synchronization) is selected	
	1	0	The IVI signal (with fall modification, without IHI synchronization) is selected	
		1	The IVI signal (with fall modification and IHI synchronization) is selected	
1	0	0	The IVG signal is selected	
		1		
	1	0		
		1		

Bits 1 and 0—Clamp Waveform Mode Select 1 and 0 (CLMOD1, CLMOD0): These bits select the signal source for the CLO signal (clamp waveform).

Bit 6	Bit 1	Bit 0		
ISGENE	**CLMOD1**	**CLMOD2**	**Description**	
0	0	0	The CL_1 signal is selected	(Initial value)
		1	The CL_2 signal is selected	
	1	0	The CL_3 signal is selected	
		1		
1	0	0	The CL_4 signal is selected	
		1		
	1	0		
		1		

13.2.4 Edge Sense Register (SEDGR)

Bit	7	6	5	4	3	2	1	0
	VEDG	HEDG	CEDG	HFEDG	VFEDG	PREQF	IHI	IVI
Initial value	0	0	0	0	0	0	—*2	—*2
Read/Write	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R	R

Notes: 1. Only 0 can be written, to clear the flags.
2. The initial value is undefined since it depends on the pin states.

SEDGR is an 8-bit readable/writable register used to detect a rising edge on the timer connection input pins and the occurrence of 2fH modification, and to determine the phase of the IVI and IHI signals.

The upper 6 bits of SEDGR are initialized to 0 by a reset and in hardware standby mode. The initial value of the lower 2 bits is undefined, since it depends on the pin states.

Bit 7—VSYNCI Edge (VEDG): Detects a rising edge on the VSYNCI pin.

Bit 7

VEDG	Description	
0	[Clearing condition] When 0 is written in VEDG after reading VEDG = 1	(Initial value)
1	[Setting condition] When a rising edge is detected on the VSYNCI pin	

Bit 6—HSYNCI Edge (HEDG): Detects a rising edge on the HSYNCI pin.

Bit 6

HEDG	Description	
0	[Clearing condition] When 0 is written in HEDG after reading HEDG = 1	(Initial value)
1	[Setting condition] When a rising edge is detected on the HSYNCI pin	

Bit 5—CSYNCI Edge (CEDG): Detects a rising edge on the CSYNCI pin.

Bit 5

CEDG	Description	
0	[Clearing condition]	(Initial value)
	When 0 is written in CEDG after reading CEDG = 1	
1	[Setting condition]	
	When a rising edge is detected on the CSYNCI pin	

Bit 4—HFBACKI Edge (HFEDG): Detects a rising edge on the HFBACKI pin.

Bit 4

HFEDG	Description	
0	[Clearing condition]	(Initial value)
	When 0 is written in HFEDG after reading HFEDG = 1	
1	[Setting condition]	
	When a rising edge is detected on the HFBACKI pin	

Bit 3—VFBACKI Edge (VFEDG): Detects a rising edge on the VFBACKI pin.

Bit 3

VFEDG	Description	
0	[Clearing condition]	(Initial value)
	When 0 is written in VFEDG after reading VFEDG = 1	
1	[Setting condition]	
	When a rising edge is detected on the VFBACKI pin	

Bit 2—Pre-Equalization Flag (PREQF): Detects the occurrence of an IHI signal 2fH modification condition. The generation of a falling/rising edge in the IHI signal during a mask interval is expressed as the occurrence of a 2fH modification condition. For details, see section 13.3.4, IHI Signal 2fH Modification.

Bit 2

PREQF	Description
0	[Clearing condition] (Initial value) When 0 is written in PREQF after reading PREQF = 1
1	[Setting condition] When an IHI signal 2fH modification condition is detected

Bit 1—IHI Signal Level (IHI): Indicates the current level of the IHI signal. Signal source and phase inversion selection for the IHI signal depends on the contents of TCONRI. Read this bit to determine whether the input signal is positive or negative, then maintain the IHI signal at positive phase by modifying TCONRI.

Bit 1

IHI	Description
0	The IHI signal is low
1	The IHI signal is high

Bit 0—IVI Signal Level (IVI): Indicates the current level of the IVI signal. Signal source and phase inversion selection for the IVI signal depends on the contents of TCONRI. Read this bit to determine whether the input signal is positive or negative, then maintain the IVI signal at positive phase by modifying TCONRI.

Bit 0

IVI	Description
0	The IVI signal is low
1	The IVI signal is high

13.2.5 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR, comprising two 8-bit readable/writable registers, performs module stop mode control. When the MSTP13, MSTP12, and MSTP8 bits are set to 1, the 16-bit free-running timer, 8-bit timer channels 0 and 1 and channels X and Y, and timer connection, respectively, halt and enter module stop mode at the end of the bus cycle. See section 21.5, Module Stop Mode, for details.

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

MSTPCRH Bit 5—Module Stop (MSTP13): Specifies FRT module stop mode.

MSTPCRH Bit 5	
MSTP13	**Description**
0	FRT module stop mode is cleared
1	FRT module stop mode is set (Initial value)

MSTPCRH Bit 4—Module Stop (MSTP12): Specifies 8-bit timer channel 0 and 1 module stop mode.

MSTPCRH Bit 4	
MSTP12	**Description**
0	8-bit timer channel 0 and 1 module stop mode is cleared
1	8-bit timer channel 0 and 1 module stop mode is set (Initial value)

MSTPCRH Bit 0—Module Stop (MSTP8): Specifies 8-bit timer channel X and Y and timer connection module stop mode.

MSTPCRH Bit 0	
MSTP8	**Description**
0	8-bit timer channel X and Y and timer connection module stop mode is cleared
1	8-bit timer channel X and Y and timer connection module stop mode is set (Initial value)

13.3 Operation

13.3.1 PWM Decoding (PDC Signal Generation)

The timer connection facility and TMRX can be used to decode a PWM signal in which 0 and 1 are represented by the pulse width. To do this, a signal in which a rising edge is generated at regular intervals must be selected as the IHI signal.

The timer counter (TCNT) in TMRX is set to count the internal clock pulses and to be cleared on the rising edge of the external reset signal (IHI signal). The value to be used as the threshold for deciding the pulse width is written in TCORB. The PWM decoder contains a delay latch which uses the IHI signal as data and compare-match signal B (CMB) as a clock, and the state of the IHI signal (the result of the pulse width decision) at the compare-match signal B timing after TCNT is reset by the rise of the IHI signal is output as the PDC signal. The pulse width setting using TICRR and TICRF of TMRX can be used to determine the pulse width decision threshold. Examples of TCR and TCORB in TMRX settings are shown in tables 13.3 and 13.4, and the timing chart is shown in figure 13.2.

Table 13.3 Examples of TCR Settings

Bit(s)	Abbreviation	Contents	Description
7	CMIEB	0	Interrupts due to compare-match and overflow are disabled
6	CMIEA	0	
5	OVIE	0	
4 and 3	CCLR1, CCLR0	11	TCNT is cleared by the rising edge of the external reset signal (IHI signal)
2 to 0	CKS2 to CKS0	001	Incremented on internal clock: ϕ

Table 13.4 Examples of TCORB (Pulse Width Threshold) Settings

	ϕ:10 MHz	ϕ: 12 MHz	ϕ: 16 MHz	ϕ: 20 MHz
H'07	0.8 µs	0.67 µs	0.5 µs	0.4 µs
H'0F	1.6 µs	1.33 µs	1 µs	0.8 µs
H'1F	3.2 µs	2.67 µs	2 µs	1.6 µs
H'3F	6.4 µs	5.33 µs	4 µs	3.2 µs
H'7F	12.8 µs	10.67 µs	8 µs	6.4 µs

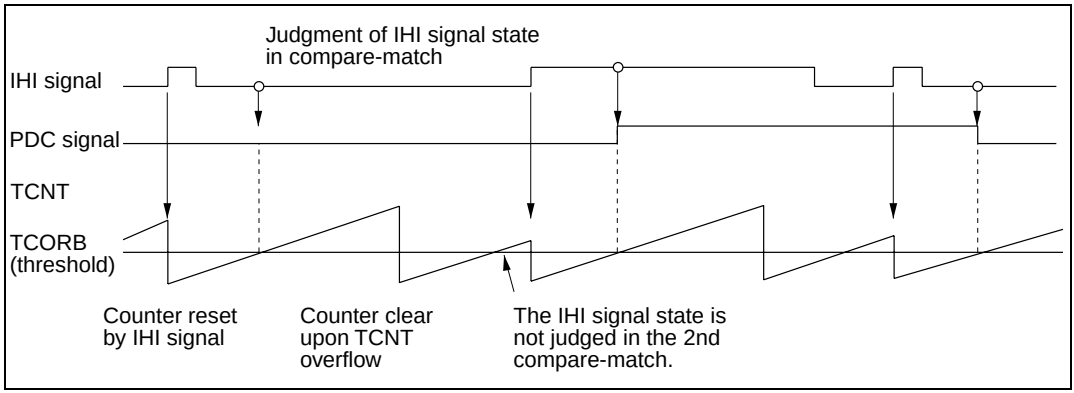


Figure 13.2 Timing Chart for PWM Decoding

13.3.2 Clamp Waveform Generation (CL_1/CL_2/CL_3 Signal Generation)

The timer connection facility and TMRX can be used to generate signals with different duty cycles and rising/falling edges (clamp waveforms) in synchronization with the input signal (IHI signal). Three clamp waveforms can be generated: the CL_1, CL_2, and CL_3 signals. In addition, the CL_4 signal can be generated using TMRY.

The CL_1 signal rises simultaneously with the rise of the IHI signal, and when the CL_1 signal is high, the CL_2 signal rises simultaneously with the fall of the IHI signal. The fall of both the CL_1 and the CL_2 signal can be specified by TCORA.

The rise of the CL_3 signal can be specified as simultaneous with the sampling of the fall of the IHI signal using the system clock, and the fall of the CL_3 signal can be specified by TCORC. The CL_3 signal falls at the rise of the IHI signal.

TCNT in TMRX is set to count internal clock pulses and to be cleared on the rising edge of the external reset signal (IHI signal).

The value to be used as the CL_1 signal pulse width is written in TCORA. Write a value of H'02 or more in TCORA when internal clock ϕ is selected as the TMRX counter clock, and a value or H'01 or more when $\phi/2$ is selected. When internal clock ϕ is selected, the CL_1 signal pulse width is (TCORA set value + 3 ± 0.5). When the CL_2 signal is used, the setting must be made so that this pulse width is greater than the IHI signal pulse width.

The value to be used as the CL_3 signal pulse width is written in TCORC. The TICR register in TMRX captures the value of TCNT at the inverse of the external reset signal edge (in this case, the falling edge of the IHI signal). The timing of the fall of the CL_3 signal is determined by the sum of

the contents of TICR and TCORC. Caution is required if the rising edge of the IHI signal precedes the fall timing set by the contents of TCORC, since the IHI signal will cause the CL_3 signal to fall.

Examples of TMRX TCR settings are the same as those in table 13.3. The clamp waveform timing charts are shown in figures 13.3 and 13.4.

Since the rise of the CL_1 and CL_2 signals is synchronized with the edge of the IHI signal, and their fall is synchronized with the system clock, the pulse width variation is equivalent to the resolution of the system clock.

Both the rise and the fall of the CL_3 signal are synchronized with the system clock and the pulse width is fixed, but there is a variation in the phase relationship with the IHI signal equivalent to the resolution of the system clock.

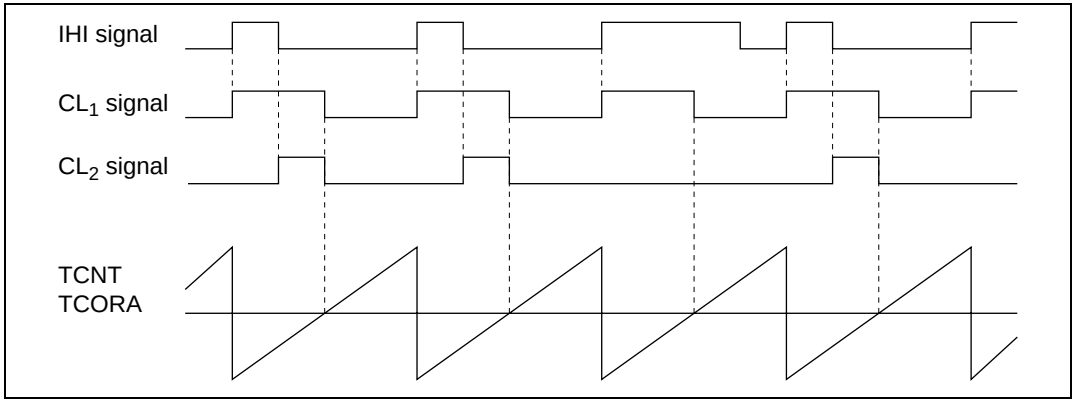


Figure 13.3 Timing Chart for Clamp Waveform Generation (CL_1 and CL_2 Signals)

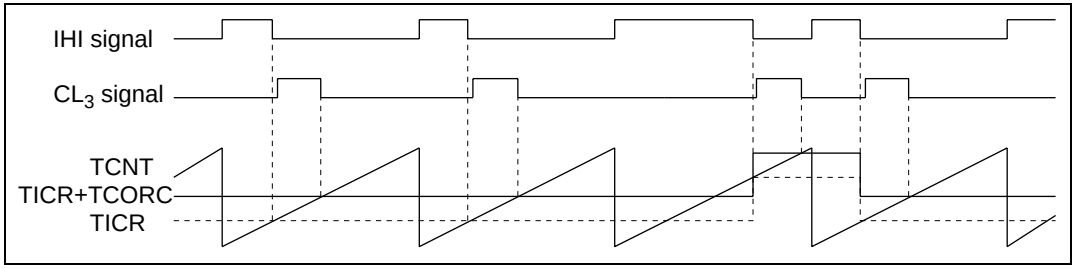


Figure 13.4 Timing Chart for Clamp Waveform Generation (CL_3 Signal)

13.3.3 Measurement of 8-Bit Timer Divided Waveform Period

The timer connection facility, TMR_1, and the free-running timer (FRT) can be used to measure the period of an IHI signal divided waveform. Since TMR_1 can be cleared by a rising edge of the IVI signal, the rise and fall of the IHI signal divided waveform can be virtually synchronized with the IVI signal. This enables period measurement to be carried out efficiently.

To measure the period of an IHI signal divided waveform, TCNT in TMR_1 is set to count the external clock (IHI signal) pulses and to be cleared on the rising edge of the external reset signal (IVI signal). The value to be used as the division factor is written in TCORA, and the TMO output method is specified by the OS bits in TCSR. Examples of TMR_1 TCR and TCSR settings are shown in table 13.5, and the timing chart for measurement of the IVI signal and IHI signal divided waveform periods is shown in figure 13.5. The period of the IHI signal divided waveform is given by (ICRD(3) – ICRD(2)) × the resolution.

Table 13.5 Examples of TCR and TCSR Settings

Register	Bit(s)	Abbreviation	Contents	Description
TCR in TMR_1	7	CMIEB	0	Interrupts due to compare-match and overflow are disabled
	6	CMIEA	0	
	5	OVIE	0	
	4, 3	CCLR1, CCLR0	11	TCNT is cleared by the rising edge of the external reset signal (IVI signal)
	2 to 0	CKS2 to CKS0	101	TCNT is incremented on the rising edge of the external clock (IHI signal)
TCSR in TMR_1	3 to 0	OS3 to OS0	0011	Not changed by compare-match B; output inverted by compare-match A (toggle output): division by 512
			1001	or when TCORB < TCORA, 1 output on compare-match B, and 0 output on compare-match A: division by 256
TCR in FRT	6	IEDGB	0/1	0: FRC value is transferred to ICRB on falling edge of input capture input B (IHI divided signal waveform) 1: FRC value is transferred to ICRB on rising edge of input capture input B (IHI divided signal waveform)
	1, 0	CKS1, CKS0	01	FRC is incremented on internal clock: $\phi/8$
TCSR in FRT	0	CCLRA	0	FRC clearing is disabled

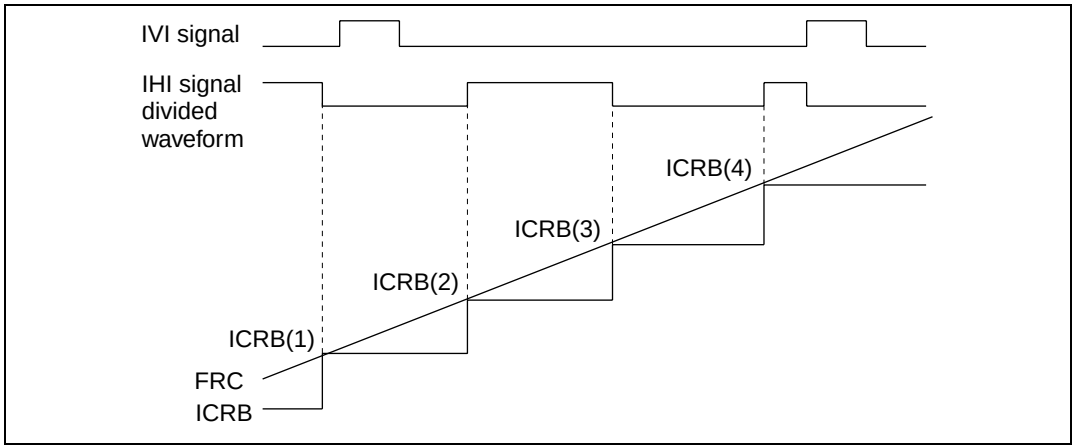


Figure 13.5 Timing Chart for Measurement of IVI Signal and IHI Signal Divided Waveform Periods

13.3.4 IHI Signal and 2fH Modification

By using the timer connection FRT, even if there is a part of the IHI signal with twice the frequency, this can be eliminated. In order for this function to operate properly, the duty cycle of the IHI signal must be approximately 30% or less, or approximately 70% or above.

The 8-bit OCRDM contents or twice the OCRDM contents can be added automatically to the data captured in ICRD in the FRT, and compare-matches generated at these points. The interval between the two compare-matches is called a mask interval. A value equivalent to approximately 1/3 the IHI signal period is written in OCRDM. ICRD is set so that capture is performed on the rise of the IHI signal.

Since the IHI signal supplied to the IHO signal selection circuit is normally set on the rise of the IHI signal and reset on the fall, its waveform is the same as that of the original IHI signal. When 2fH modification is selected, IHI signal edge detection is disabled during mask intervals. Capture is also disabled during these intervals.

Examples of FRT TCR settings are shown in table 13.6, and the 2fH modification timing chart is shown in figure 13.6.

Table 13.6 Examples of TCR, TCSR, TCOR, and OCRDM Settings

Register	Bit(s)	Abbreviation	Contents	Description
TCR in FRT	4	IEDGD	1	FRC value is transferred to ICRD on the rising edge of input capture input D (IHI signal)
	1, 0	CKS1, CKS0	01	FRC is incremented on internal clock: $\phi/8$
TCSR in FRT	0	CCLRA	0	FRC clearing is disabled
TCOR in FRT	7	ICRDMS	1	ICRD is set to the operating mode in which OCRDM is used
OCRDM in FRT	7 to 0	OCRDM7 to OCRDM0	H'01 to H'FF	Specifies the period during which ICRD operation is masked

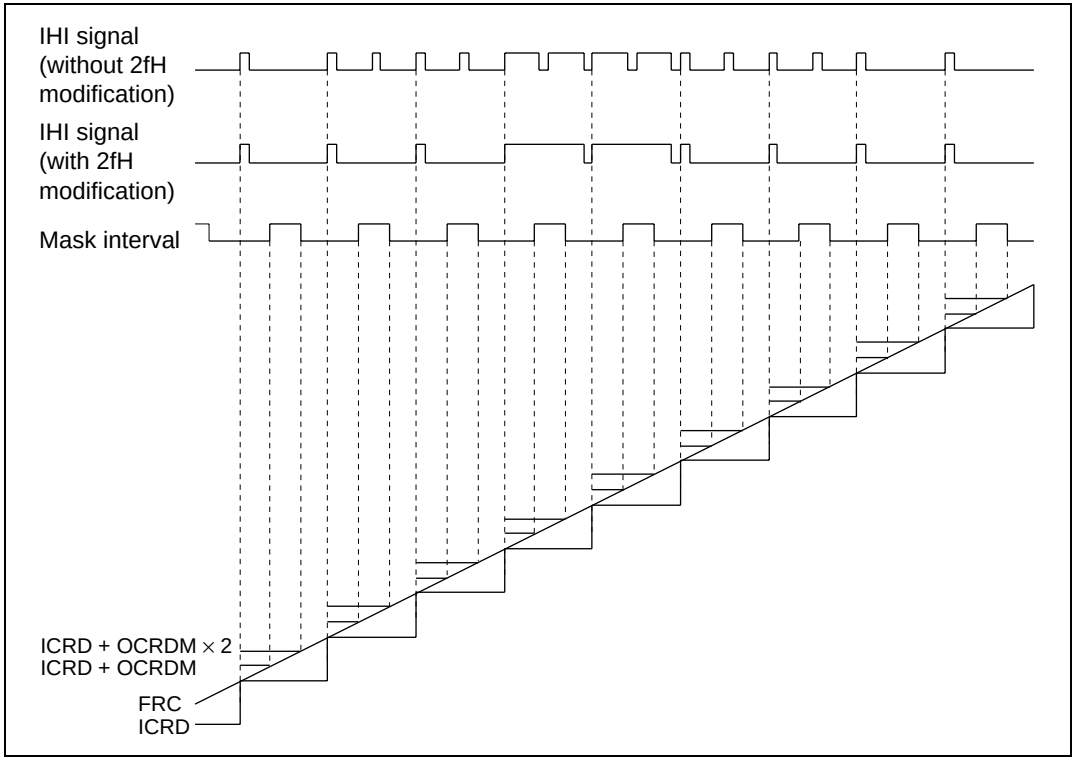


Figure 13.6 2fH Modification Timing Chart

13.3.5 IVI Signal Fall Modification and IHI Synchronization

By using the timer connection TMR_1, the fall of the IVI signal can be shifted backward by the specified number of IHI signal waveforms. Also, the fall of the IVI signal can be synchronized with the rise of the IHI signal.

To perform 8-bit timer divided waveform period measurement, TCNT in TMR_1 is set to count external clock (IHI signal) pulses, and to be cleared on the rising edge of the external reset signal (inverse of the IVI signal). The number of IHI signal pulses until the fall of the IVI signal is written in TCORB.

Since the IVI signal supplied to the IVO signal selection circuit is normally set on the rise of the IVI signal and reset on the fall, its waveform is the same as that of the original IVI signal. When fall modification is selected, a reset is performed on a TMR_1 TCORB compare-match.

The fall of the waveform generated in this way can be synchronized with the rise of the IHI signal, regardless of whether or not fall modification is selected.

Examples of TMR_1 TCORB, TCR, and TCSR settings are shown in table 13.7, and the fall modification/IHI synchronization timing chart is shown in figure 13.7.

Table 13.7 Examples of TCORB, TCR, and TCSR Settings

Register	Bit(s)	Abbreviation	Contents	Description
TCR in TMR_1	7	CMIEB	0	Interrupts due to compare-match and overflow are disabled
	6	CMIEA	0	
	5	OVIE	0	
	4, 3	CCLR1, CCLR0	11	TCNT is cleared by the rising edge of the external reset signal (inverse of the IVI signal)
	2 to 0	CKS2 to CKS0	101	TCNT is incremented on the rising edge of the external clock (IHI signal)
TCSR in TMR_1	3 to 0	OS3 to OS0	0011	Not changed by compare-match B; output inverted by compare-match A (toggle output)
			1001	or when TCORB < TCORA, 1 output on compare-match B, 0 output on compare-match A
TOCRB in TMR_1			H'03 (example)	Compare-match on the 4th (example) rise of the IHI signal after the rise of the inverse of the IVI signal

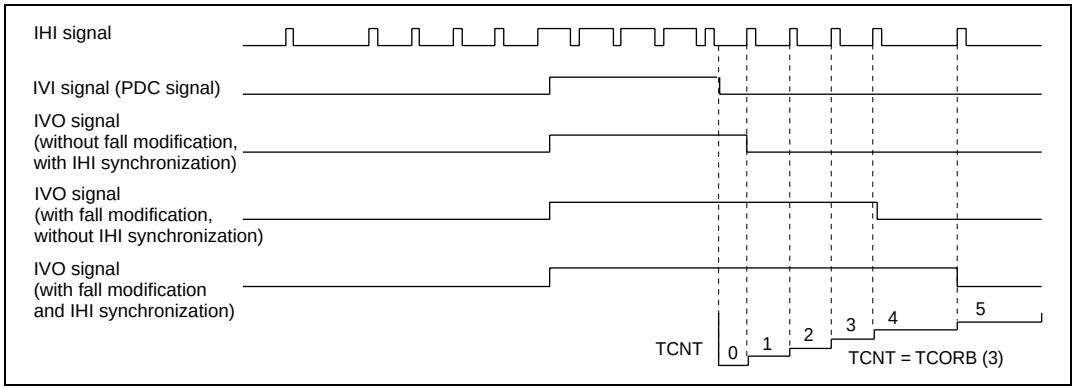


Figure 13.7 Fall Modification/IHI Synchronization Timing Chart

13.3.6 Internal Synchronization Signal Generation (IHG/IVG/CL_4 Signal Generation)

By using the timer connection FRT and TMRY, it is possible to automatically generate internal signals (IHG and IVG signals) corresponding to the IHI and IVI signals. As the IHG signal is synchronized with the rise of the IVG signal, the IHG signal period must be made a divisor of the IVG signal period in order to keep it constant. In addition, the CL_4 signal can be generated in synchronization with the IHG signal.

The contents of OCRA in the FRT are updated by the automatic addition of the contents of OCRAR or OCRAF, alternately, each time a compare-match occurs. A value corresponding to the 0 interval of the IVG signal is written in OCRAR, and a value corresponding to the 1 interval of the IVG signal is written in OCRAF. The IVG signal is set by a compare-match after an OCRAR addition, and reset by a compare-match after an OCRAF addition.

The IHG signal is the TMRY 8-bit timer output. TMRY is set to count internal clock pulses, and to be cleared on TCORA compare-match, to fix the period and set the timer output. TCORB is set so as to reset the timer output. The IVG signal is connected as the TMRY reset input (TMRI), and the rise of the IVG signal can be treated in the same way as a TCORA compare-match.

The CL_4 signal is a waveform that rises within one system clock period after the fall of the IHG signal, and has a 1 interval of 6 system clock periods.

Examples of settings of TCORA, TCORB, TCR, and TCSR in TMRY, and OCRAR, OCRAF, and TCR in the FRT, are shown in table 13.8, and the IHG signal/IVG signal timing chart is shown in figure 13.8.

Table 13.8 Examples of OCRAR, OCRAF, TOCR, TCORA, TCORB, TCR, and TCSR Settings

Register	Bit(s)	Abbreviation	Contents	Description	
TCR in TMRY	7	CMIEB	0	Interrupts due to compare-match and overflow are disabled	
	6	CMIEA	0		
	5	OVIE	0		
	4, 3	CCLR1, CCLR0	01	TCNT is cleared by compare-match A	
	2 to 0	CKS2 to CKS0	001	TCNT is incremented on internal clock: $\phi/4$	
TCSR in TMRY	3 to 0	OS3 to OS0	0110	0 output on compare-match B 1 output on compare-match A	
TOCRA in TMRY			H'3F (example)	IHG signal period = $\phi \times 256$	
TOCRB in TMRY			H'03 (example)	IHG signal 1 interval = $\phi \times 16$	
TCR in FRT	1, 0	CKS1, CKS0	01	FRC is incremented on internal clock: $\phi/8$	
OCRAR in FRT			H'7FEF (example)	IVG signal 0 interval = $\phi \times 262016$	IVG signal period = $\phi \times 262144$ (1024 times IHG signal)
OCRAF in FRT			H'000F (example)	IVG signal 1 interval = $\phi \times 128$	
TOCR in FRT	6	OCRAMS	1	OCRA is set to the operating mode in which OCRAR and OCRAF are used	

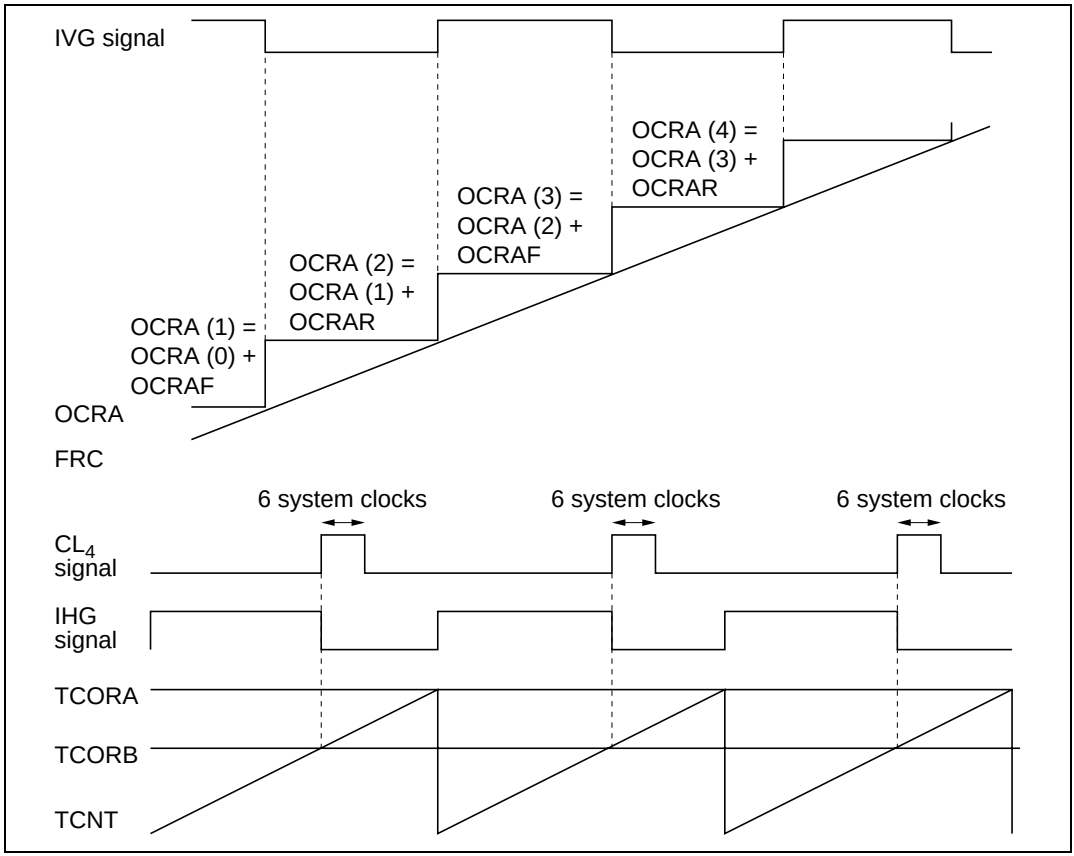


Figure 13.8 IVG Signal/IHG Signal/CL_4 Signal Timing Chart

13.3.7 HSYNCO Output

With the HSYNCO output, the meaning of the signal source to be selected and use or non-use of modification varies according to the IHI signal source and the waveform required by external circuitry. The meaning of the HSYNCO output in each mode is shown in table 13.9.

Table 13.9 Meaning of HSYNCO Output in Each Mode

Mode	IHI Signal	IHO Signal	Meaning of IHO Signal
No signal	HFBACKI input	IHI signal (without 2fH modification)	HFBACKI input is output directly
		IHI signal (with 2fH modification)	Meaningless unless there is a double-frequency part in the HFBACKI input
		CL_1 signal	HFBACKI input 1 interval is changed before output
		IHG signal	Internal synchronization signal is output
S-on-G mode	CSYNCI input	IHI signal (without 2fH modification)	CSYNCI input (composite synchronization signal) is output directly
		IHI signal (with 2fH modification)	Double-frequency part of CSYNCI input (composite synchronization signal) is eliminated before output
		CL_1 signal	CSYNCI input (composite synchronization signal) horizontal synchronization signal part is separated before output
		IHG signal	Internal synchronization signal is output
Composite mode	HSYNCI input	IHI signal (without 2fH modification)	HSYNCI input (composite synchronization signal) is output directly
		IHI signal (with 2fH modification)	Double-frequency part of HSYNCI input (composite synchronization signal) is eliminated before output
		CL_1 signal	HSYNCI input (composite synchronization signal) horizontal synchronization signal part is separated before output
		IHG signal	Internal synchronization signal is output
Separate mode	HSYNCI input	IHI signal (without 2fH modification)	HSYNCI input (horizontal synchronization signal) is output directly
		IHI signal (with 2fH modification)	Meaningless unless there is a double-frequency part in the HSYNCI input (horizontal synchronization signal)
		CL_1 signal	HSYNCI input (horizontal synchronization signal) 1 interval is changed before output
		IHG signal	Internal synchronization signal is output

13.3.8 VSYNCO Output

With the VSYNCO output, the meaning of the signal source to be selected and use or non-use of modification varies according to the IVI signal source and the waveform required by external circuitry. The meaning of the VSYNCO output in each mode is shown in table 13.10.

Table 13.10 Meaning of VSYNCO Output in Each Mode

Mode	IVI Signal	IVO Signal	Meaning of IVO Signal
No signal	VFBACKI input	IVI signal (without fall modification or IHI synchronization)	VFBACKI input is output directly
		IVI signal (without fall modification, with IHI synchronization)	Meaningless unless VFBACKI input is synchronized with HFBACKI input
		IVI signal (with fall modification, without IHI synchronization)	VFBACKI input fall is modified before output
		IVI signal (with fall modification and IHI synchronization)	VFBACKI input fall is modified and signal is synchronized with HFBACKI input before output
		IVG signal	Internal synchronization signal is output
S-on-G mode or composite mode	PDC signal	IVI signal (without fall modification or IHI synchronization)	CSYNCI/HSYNCI input (composite synchronization signal) vertical synchronization signal part is separated before output
		IVI signal (without fall modification, with IHI synchronization)	CSYNCI/HSYNCI input (composite synchronization signal) vertical synchronization signal part is separated, and signal is synchronized with CSYNCI/HSYNCI input before output
		IVI signal (with fall modification, without IHI synchronization)	CSYNCI/HSYNCI input (composite synchronization signal) vertical synchronization signal part is separated, and fall is modified before output
		IVI signal (with fall modification and IHI synchronization)	CSYNCI/HSYNCI input (composite synchronization signal) vertical synchronization signal part is separated, fall is modified, and signal is synchronized with CSYNCI/HSYNCI input before output
		IVG signal	Internal synchronization signal is output

Mode	IVI Signal	IVO Signal	Meaning of IVO Signal
Separate mode	VSYNCI input	IVI signal (without fall modification or IHI synchronization)	VSYNCI input (vertical synchronization signal) is output directly
		IVI signal (without fall modification, with IHI synchronization)	Meaningless unless VSYNCI input (vertical synchronization signal) is synchronized with HSYNCI input (horizontal synchronization signal)
		IVI signal (with fall modification, without IHI synchronization)	VSYNCI input (vertical synchronization signal) fall is modified before output
		IVI signal (with fall modification and IHI synchronization)	VSYNCI input (vertical synchronization signal) fall is modified and signal is synchronized with HSYNCI input (horizontal synchronization signal) before output
		IVG signal	Internal synchronization signal is output

13.3.9 CBLANK Output

Using the signals generated/selected with timer connection, it is possible to generate a waveform based on the composite synchronization signal (blanking waveform).

One kind of blanking waveform is generated by combining HFBACKI and VFBACKI inputs, with the phase polarity made positive by means of bits HFINV and VFINV in TCONRI, with the IVO signal.

The composition logic is shown in figure 13.9.

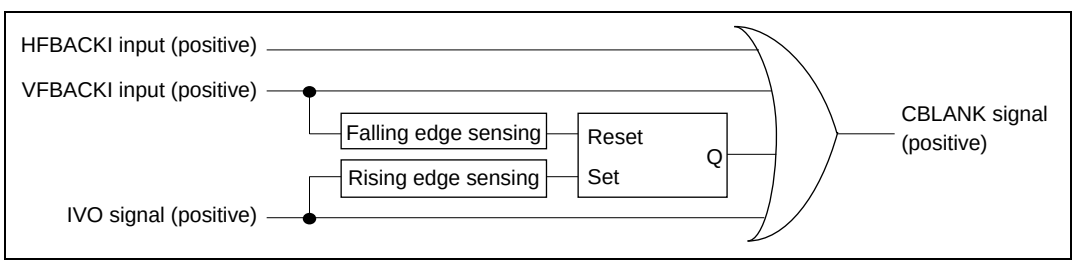


Figure 13.9 CBLANK Output Waveform Generation

Section 14 Watchdog Timer (WDT)

14.1 Overview

The H8/3577 Group and H8/3567 Group have an on-chip watchdog timer (WDT0). The WDT outputs an overflow signal if a system crash prevents the CPU from writing to the timer counter, allowing it to overflow. At the same time, the WDT can also generate an internal reset signal or internal NMI interrupt signal.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer mode, an interval timer interrupt is generated each time the counter overflows.

14.1.1 Features

- Switchable between watchdog timer mode and interval timer mode
  - WOVI interrupt generation in interval timer mode
- Internal reset or internal interrupt generated when the timer counter overflows
  - Choice of internal reset or NMI interrupt generation in watchdog timer mode
- Choice of 8 counter input clocks
  - Maximum WDT interval: system clock period × 131072 × 256

14.1.2 Block Diagram

Figure 14.1 shows block diagram of WDT0.

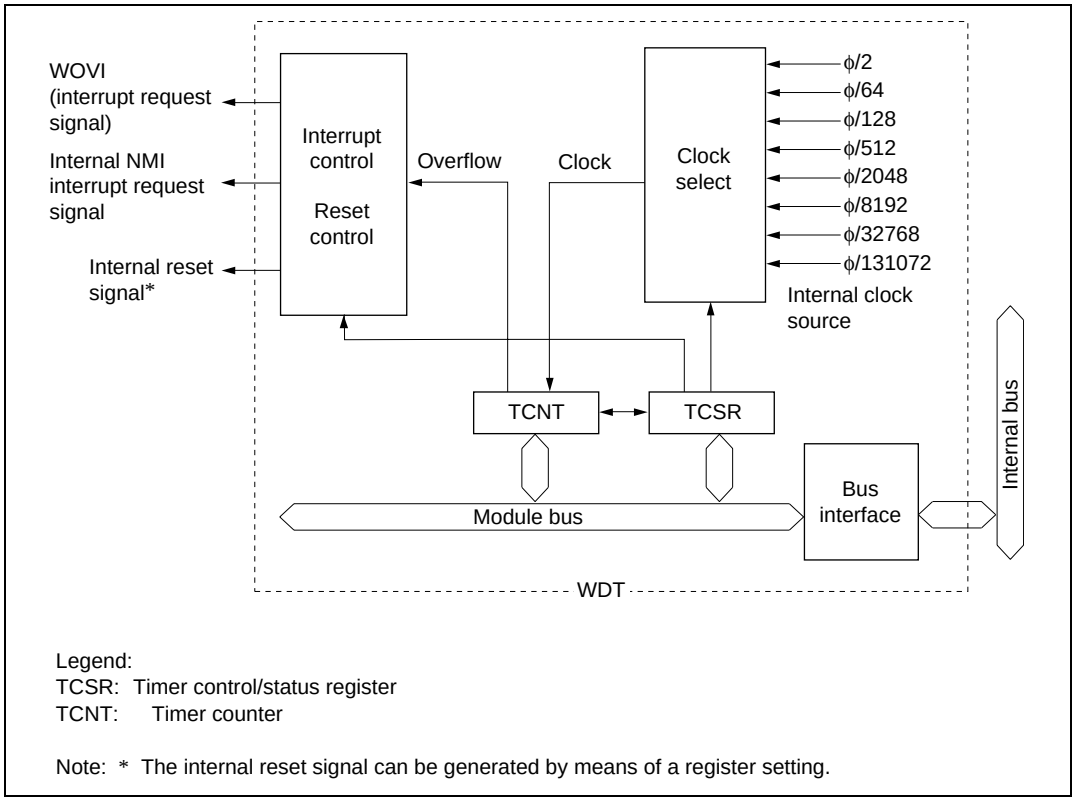


Figure 14.1 Block Diagram of WDT0

14.1.3 Register Configuration

The WDT has four registers, as summarized in table 14.1. These registers control clock selection, WDT mode switching, the reset signal, etc.

Table 14.1 WDT Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address	
					Write*1	Read
0	Timer control/status register 0	TCSR0	R/(W)*2	H'00	H'FFA8	H'FFA8
	Timer counter 0	TCNT0	R/W	H'00	H'FFA8	H'FFA9
Common	System control register	SYSCR	R/W	H'09	H'FFC4	H'FFC4

Notes: 1. For details of write operations, see section 14.2.4, Notes on Register Access.
2. Only 0 can be written in bit 7, to clear the flag.

14.2 Register Descriptions

14.2.1 Timer Counter (TCNT)

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

TCNT is an 8-bit readable/writable* up-counter.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from the internal clock source selected by bits CKS2 to CKS0 in TCSR. When the TCNT value overflows (changes from H'FF to H'00), the OVF flag in TCSR is set to 1.

TCNT is initialized to H'00 by a reset, in hardware standby mode, or when the TME bit is cleared to 0. It is not initialized in software standby mode.

Note: * The method of writing to TCNT is more complicated than for most other registers, to prevent accidental overwriting. For details see section 14.2.4, Notes on Register Access.

14.2.2 Timer Control/Status Register (TCSR0)

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	RSTS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

TCSR is an 8-bit readable/writable* register. Its functions include selecting the clock source to be input to TCNT, and the timer mode.

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

Note: * The method of writing to TCSR is more complicated than for most other registers, to prevent accidental overwriting. For details see section 14.2.4, Notes on Register Access.

Bit 7—Overflow Flag (OVF): A status flag that indicates that TCNT has overflowed from H'FF to H'00.

Bit 7

OVF	Description
0	[Clearing conditions] • Write 0 in the TME bit (Initial value) • Read TCSR when OVF = 1*, then write 0 in OVF
1	[Setting condition] When TCNT overflows (changes from H'FF to H'00) (When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.)

Note: * When the interval timer interrupt is disabled and OVF is polled, reading OVF while set to 1 should be performed at least twice.

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether the WDT is used as a watchdog timer or interval timer. If used as an interval timer, the WDT generates an interval timer interrupt request (WOVI) when TCNT overflows. If used as a watchdog timer, the WDT generates a reset or NMI interrupt when TCNT overflows.

Bit 6

WT/$\overline{IT}$	Description	
0	Interval timer: Sends the CPU an interval timer interrupt request (WOVI) when TCNT overflows	(Initial value)
1	Watchdog timer: Generates a reset or NMI interrupt when TCNT overflows	

Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted.

Bit 5

TME	Description	
0	TCNT is initialized to H'00 and halted	(Initial value)
1	TCNT counts	

TCSR0 Bit 4—Reset Select (RSTS): Reserved. This bit should not be set to 1.

Bit 3—Reset or NMI (RST/$\overline{NMI}$): Specifies whether an internal reset or NMI interrupt is requested on TCNT overflow in watchdog timer mode.

Bit 3

RST/$\overline{NMI}$	Description	
0	An NMI interrupt is requested	(Initial value)
1	An internal reset is requested	

Bits 2 to 0—Clock Select 2 to 0 (CKS2 to CKS0): These bits select the clock to be input to TCNT from internal clocks obtained by dividing the system clock.

Bit 2	Bit 1	Bit 0	Description	
CKS2	**CKS1**	**CKS0**	**Clock**	**Overflow Period* (when ϕ = 20 MHz)**
0	0	0	ϕ/2 (Initial value)	25.6 µs
		1	ϕ/64	819.2 µs
	1	0	ϕ/128	1.6 ms
		1	ϕ/512	6.6 ms
1	0	0	ϕ/2048	26.2 ms
		1	ϕ/8192	104.9 ms
	1	0	ϕ/32768	419.4 ms
		1	ϕ/131072	1.68 s

Note: * The overflow period is the time from when TCNT starts counting up from H'00 until overflow occurs.

14.2.3 System Control Register (SYSCR)

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

Only bit 3 is described here. For details on functions not related to the watchdog timer, see sections 3.2.2 and 5.2.1, System Control Register (SYSCR), and the descriptions of the relevant modules.

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

Bit 3		
XRST	**Description**	
0	Reset is generated by watchdog timer overflow	
1	Reset is generated by external reset input	(Initial value)

14.2.4 Notes on Register Access

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

Writing to TCNT and TCSR: These registers must be written to by a word transfer instruction. They cannot be written to with byte transfer instructions.

Figure 14.2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. For a write to TCNT, the upper byte of the written word must contain H'5A and the lower byte must contain the write data. For a write to TCSR, the upper byte of the written word must contain H'A5 and the lower byte must contain the write data. This transfers the write data from the lower byte to TCNT or TCSR.

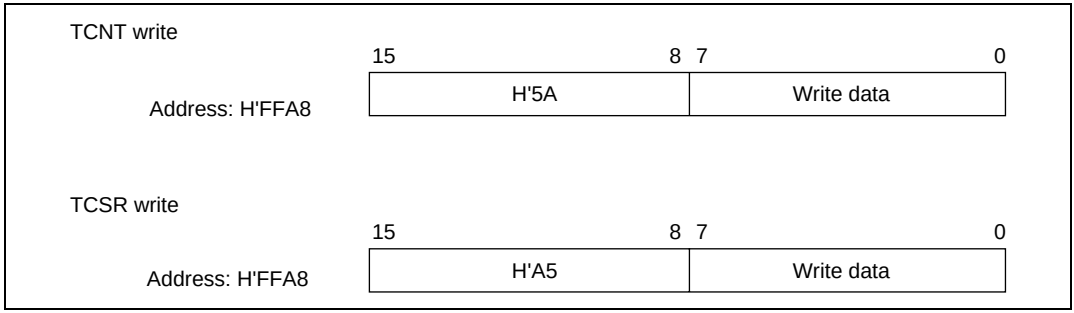


Figure 14.2 Format of Data Written to TCNT and TCSR

Reading TCNT and TCSR: These registers are read in the same way as other registers. The read addresses are H'FFA8 for TCSR, and H'FFA9 for TCNT.

14.3 Operation

14.3.1 Watchdog Timer Operation

To use the WDT as a watchdog timer, set the WT/$\overline{\text{IT}}$ and TME bits in TCSR to 1. Software must prevent TCNT overflows by rewriting the TCNT value (normally by writing H'00) before overflow occurs. This ensures that TCNT does not overflow while the system is operating normally. If TCNT overflows without being rewritten because of a system crash or other error, an internal reset or NMI interrupt request is generated.

When the RST/$\overline{\text{NMI}}$ bit is set to 1, the chip is reset for 518 system clock periods (518 ϕ) by a counter overflow. This is illustrated in figure 14.3.

An internal reset request from the watchdog timer and reset input from the $\overline{\text{RES}}$ pin are handled via the same vector. The reset source can be identified from the value of the XRST bit in SYSCR.

If a reset caused by an input signal from the $\overline{\text{RES}}$ pin and a reset caused by WDT overflow occur simultaneously, the $\overline{\text{RES}}$ pin reset has priority, and the XRST bit in SYSCR is set to 1.

An NMI interrupt request from the watchdog timer and an interrupt request from the NMI pin are handled via the same vector. Simultaneous handling of a watchdog timer NMI interrupt request and an NMI pin interrupt request must therefore be avoided.

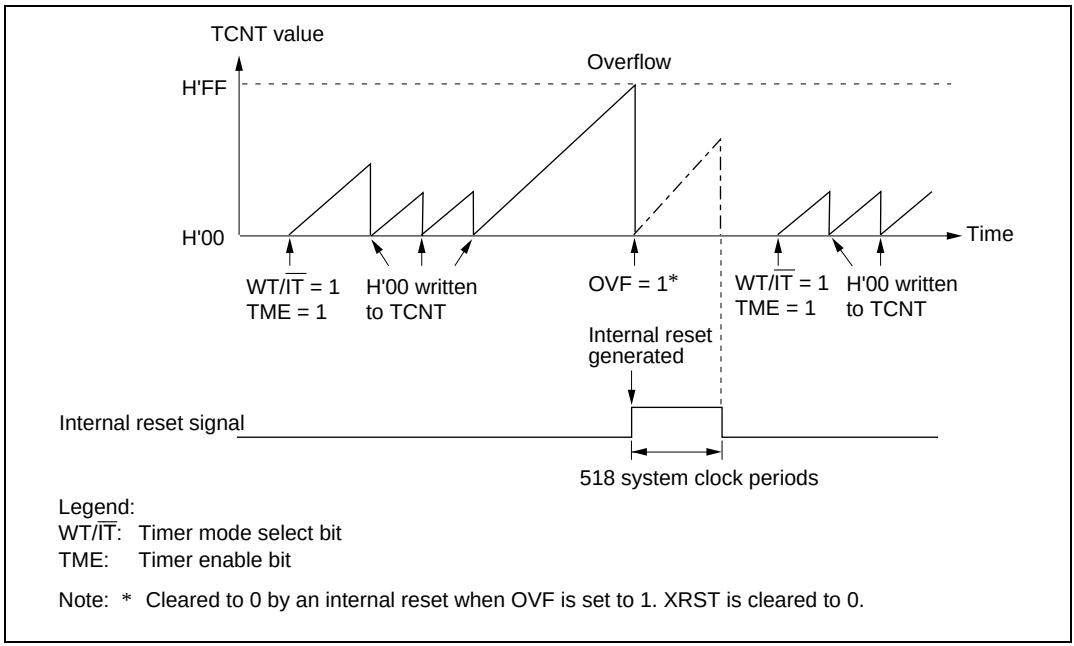


Figure 14.3 Operation in Watchdog Timer Mode

14.3.2 Interval Timer Operation

To use the WDT as an interval timer, clear the WT/$\overline{\text{IT}}$ bit in TCSR to 0 and set the TME bit to 1. An interval timer interrupt (WOVI) is generated each time TCNT overflows, provided that the WDT is operating as an interval timer, as shown in figure 14.4. This function can be used to generate interrupt requests at regular intervals.

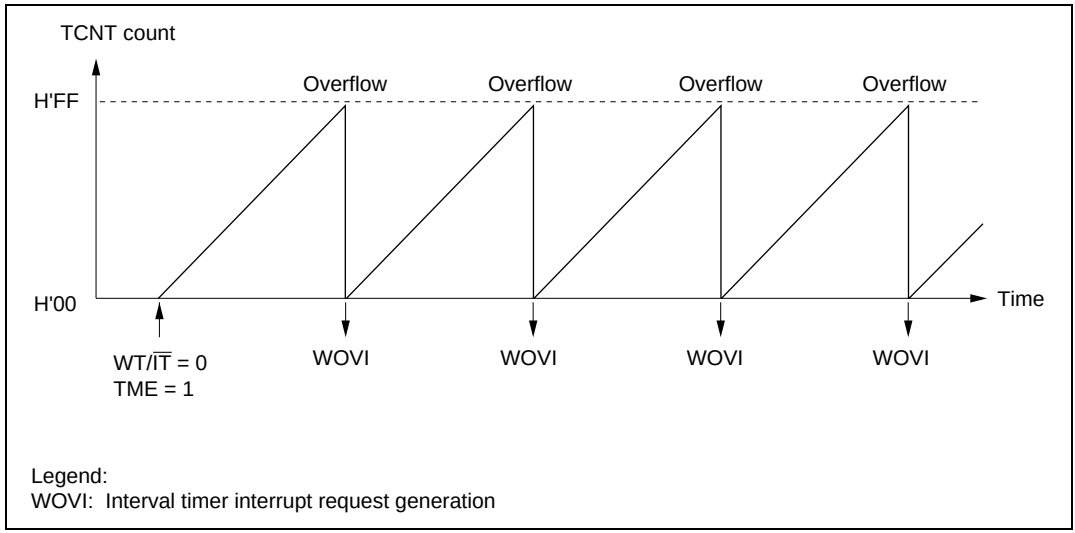


Figure 14.4 Operation in Interval Timer Mode

14.3.3 Timing of Setting of Overflow Flag (OVF)

The OVF bit in TCSR is set to 1 if TCNT overflows during interval timer operation. At the same time, an interval timer interrupt (WOVI) is requested. This timing is shown in figure 14.5.

If NMI request generation is selected in watchdog timer mode, when TCNT overflows the OVF bit in TCSR is set to 1 and at the same time an NMI interrupt is requested.

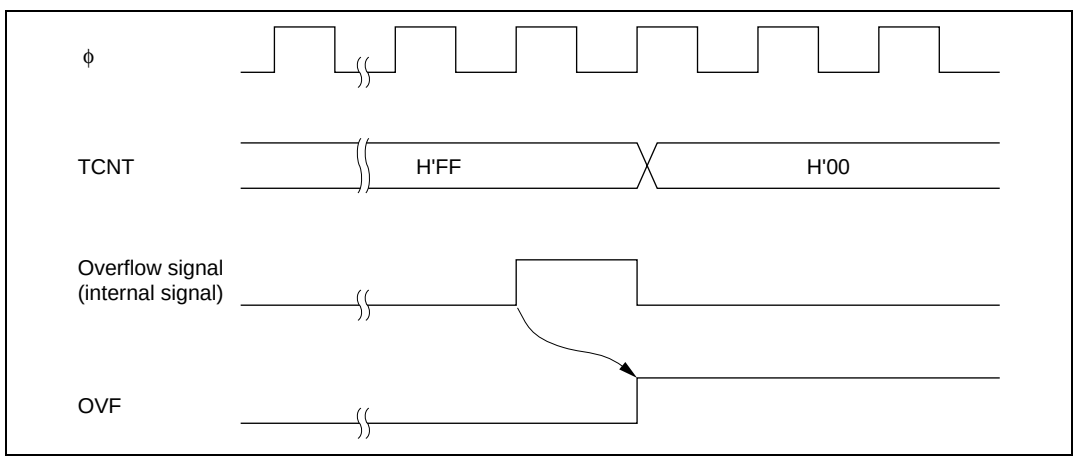


Figure 14.5 Timing of OVF Setting

14.4 Interrupts

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine. When NMI interrupt request generation is selected in watchdog timer mode, an overflow generates an NMI interrupt request.

14.5 Usage Notes

14.5.1 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 14.6 shows this operation.

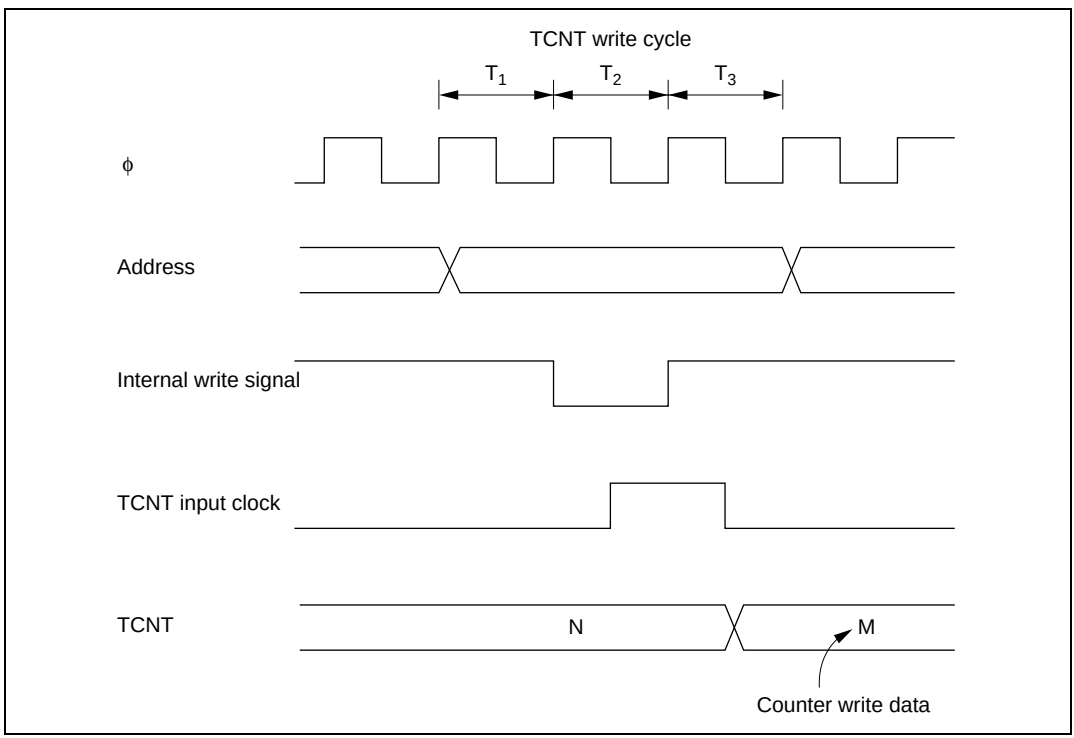


Figure 14.6 Contention between TCNT Write and Increment

14.5.2 Changing Value of CKS2 to CKS0

If bits CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits CKS2 to CKS0.

14.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, or vice versa, while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

Section 15 Serial Communication Interface (SCI)

15.1 Overview

The H8/3577 Group and H8/3567 Group are equipped with a single-channel serial communication interface (SCI). The SCI can handle both asynchronous and clocked synchronous serial communication. A function is also provided for serial communication between processors (multiprocessor communication function).

15.1.1 Features

SCI features are listed below.

- Choice of asynchronous or synchronous serial communication mode

  Asynchronous mode

  — Serial data communication is executed using an asynchronous system in which synchronization is achieved character by character

  Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA)

  — A multiprocessor communication function is provided that enables serial data communication with a number of processors

  — Choice of 12 serial data transfer formats

Data length:	7 or 8 bits
Stop bit length:	1 or 2 bits
Parity:	Even, odd, or none
Multiprocessor bit:	1 or 0

  — Receive error detection: Parity, overrun, and framing errors

  — Break detection: Break can be detected by reading the RxD pin level directly in case of a framing error

  Synchronous mode

  — Serial data communication is synchronized with a clock

  Serial data communication can be carried out with other chips that have a synchronous communication function

  — One serial data transfer format

Data length:	8 bits

  — Receive error detection: Overrun errors detected

- Full-duplex communication capability
  - The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously
  - Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data
- LSB-first or MSB-first transfer can be selected
  - This selection can be made regardless of the communication mode (with the exception of 7-bit data transfer in asynchronous mode)*

  Note: * LSB-first transfer is used in the examples in this section.

- Built-in baud rate generator allows any bit rate to be selected
- Choice of serial clock source: internal clock from baud rate generator or external clock from SCK pin
- Four interrupt sources
  - Four interrupt sources (transmit-data-empty, transmit-end, receive-data-full, and receive error) that can issue requests independently

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the SCI.

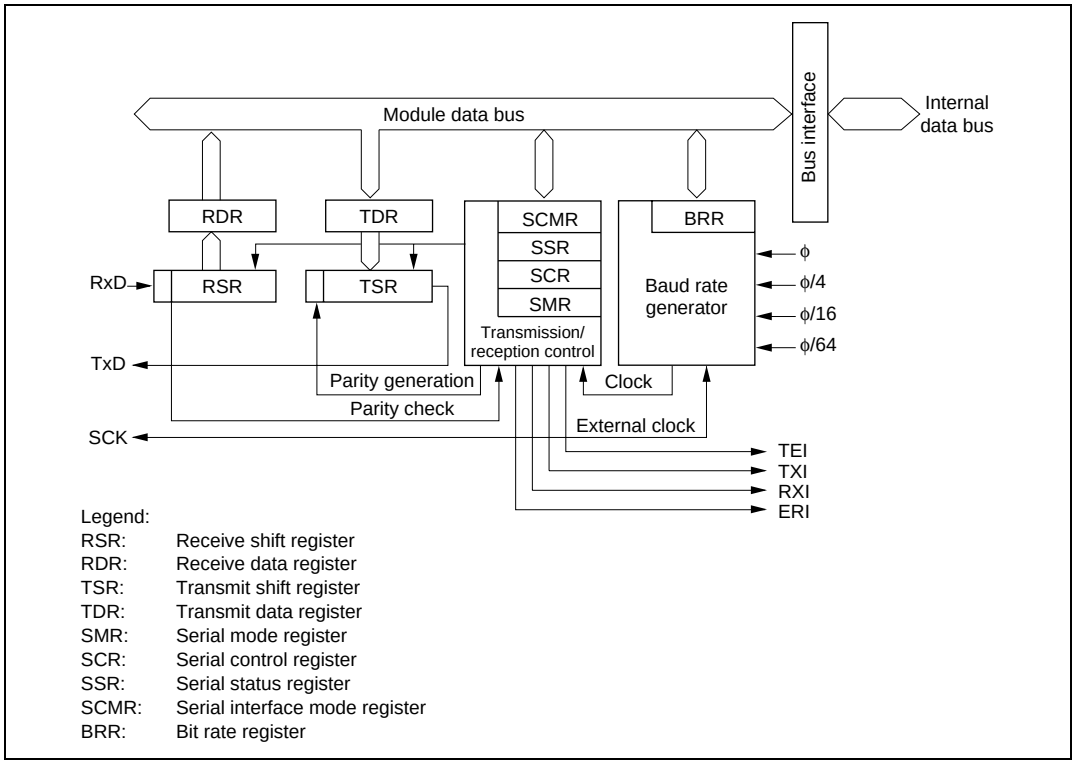


Figure 15.1 Block Diagram of SCI

15.1.3 Pin Configuration

Table 15.1 shows the serial pins used by the SCI.

Table 15.1 SCI Pins

Channel	Pin Name	Symbol	I/O	Function
0	Serial clock pin 0	SCK_0	I/O	SCI0 clock input/output
	Receive data pin 0	RxD_0	Input	SCI0 receive data input
	Transmit data pin 0	TxD_0	Output	SCI0 transmit data output

Note: The abbreviations SCK, RxD, and TxD are used in the text, omitting the channel number.

15.1.4 Register Configuration

The SCI has the internal registers shown in table 15.2. These registers are used to specify asynchronous mode or synchronous mode, the data format, and the bit rate, and to control the transmitter/receiver.

Table 15.2 SCI Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address
0	Serial mode register 0	SMR0	R/W	H'00	H'FFD8*2
	Bit rate register 0	BRR0	R/W	H'FF	H'FFD9*2
	Serial control register 0	SCR0	R/W	H'00	H'FFDA
	Transmit data register 0	TDR0	R/W	H'FF	H'FFDB
	Serial status register 0	SSR0	R/(W)*1	H'84	H'FFDC
	Receive data register 0	RDR0	R	H'00	H'FFDD
	Serial interface mode register 0	SCMR0	R/W	H'F2	H'FFDE*3
Common	Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
		MSTPCRL	R/W	H'FF	H'FF87

Notes: 1. Only 0 can be written, to clear flags.
2. Some serial communication interface registers are assigned to the same addresses as other registers. In this case, register selection is performed by the IICE bit in the serial timer control register (STCR).

15.2 Register Descriptions

15.2.1 Receive Shift Register (RSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

RSR is a register used to receive serial data.

The SCI sets serial data input from the RxD pin in RSR in the order received, starting with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to RDR automatically.

RSR cannot be directly read or written to by the CPU.

15.2.2 Receive Data Register (RDR)

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 is a register that stores received serial data.

When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR where it is stored, and completes the receive operation. After this, RSR is receive-enabled.

Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed.

RDR is a read-only register, and cannot be written to by the CPU.

RDR is initialized to H'00 by a reset, and in standby mode, and module stop mode.

15.2.3 Transmit Shift Register (TSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

TSR is a register used to transmit serial data.

To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from TDR to TSR, and transmission started, automatically. However, data transfer from TDR to TSR is not performed if the TDRE bit in SSR is set to 1.

TSR cannot be directly read or written to by the CPU.

15.2.4 Transmit Data Register (TDR)

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

TDR is an 8-bit register that stores data for serial transmission.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts serial transmission. Continuous serial transmission can be carried out by writing the next transmit data to TDR during serial transmission of the data in TSR.

TDR can be read or written to by the CPU at all times.

TDR is initialized to H'FF by a reset, and in standby mode, and module stop mode.

15.2.5 Serial Mode Register (SMR)

Bit	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SMR is an 8-bit register used to set the SCI's serial transfer format and select the baud rate generator clock source.

SMR can be read or written to by the CPU at all times.

SMR is initialized to H'00 by a reset, and in standby mode, and module stop mode.

Bit 7—Communication Mode (C/$\overline{A}$): Selects asynchronous mode or synchronous mode as the SCI operating mode.

Bit 7

C/$\overline{A}$	Description	
0	Asynchronous mode	(Initial value)
1	Synchronous mode	

Bit 6—Character Length (CHR): Selects 7 or 8 bits as the data length in asynchronous mode. In synchronous mode, a fixed data length of 8 bits is used regardless of the CHR setting.

Bit 6

CHR	Description	
0	8-bit data	(Initial value)
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and LSB-first/MSB-first selection is not available.

Bit 5—Parity Enable (PE): In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking in reception. In synchronous mode, or when a multiprocessor format is used, parity bit addition and checking is not performed, regardless of the PE bit setting.

Bit 5

PE	Description	
0	Parity bit addition and checking disabled	(Initial value)
1	Parity bit addition and checking enabled*	

Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

Bit 4—Parity Mode (O/$\overline{E}$): Selects either even or odd parity for use in parity addition and checking.

The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking, in asynchronous mode. The O/$\overline{E}$ bit setting is invalid in synchronous mode, when parity bit addition and checking is disabled in asynchronous mode, and when a multiprocessor format is used.

Bit 4

O/$\overline{E}$	Description	
0	Even parity*1	(Initial value)
1	Odd parity*2	

Notes: 1. When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is even.
2. When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.

Bit 3—Stop Bit Length (STOP): Selects 1 or 2 bits as the stop bit length in asynchronous mode. The STOP bit setting is only valid in asynchronous mode. If synchronous mode is set the STOP bit setting is invalid since stop bits are not added.

Bit 3

STOP	Description	
0	1 stop bit*1	(Initial value)
1	2 stop bits*2	

Notes: 1. In transmission, a single 1 bit (stop bit) is added to the end of a transmit character before it is sent.
2. In transmission, two 1 bits (stop bits) are added to the end of a transmit character before it is sent.

In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next transmit character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, the PE bit and O/$\overline{E}$ bit parity settings are invalid. The MP bit setting is only valid in asynchronous mode; it is invalid in synchronous mode.

For details of the multiprocessor communication function, see section 15.3.3, Multiprocessor Communication Function.

Bit 2		
MP	**Description**	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source for the baud rate generator. The clock source can be selected from ϕ, ϕ/4, ϕ/16, and ϕ/64, according to the setting of bits CKS1 and CKS0.

For the relation between the clock source, the bit rate register setting, and the baud rate, see section 15.2.8, Bit Rate Register.

Bit 1	Bit 0		
CKS1	**CKS0**	**Description**	
0	0	ϕ clock	(Initial value)
	1	ϕ/4 clock	
1	0	ϕ/16 clock	
	1	ϕ/64 clock	

15.2.6 Serial Control Register (SCR)

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR is a register that performs enabling or disabling of SCI transfer operations, serial clock output in asynchronous mode, and interrupt requests, and selection of the serial clock source.

SCR can be read or written to by the CPU at all times.

SCR is initialized to H'00 by a reset, and in standby mode, and module stop mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit-data-empty interrupt (TXI) request generation when serial transmit data is transferred from TDR to TSR and the TDRE flag in SSR is set to 1.

Bit 7

TIE	Description	
0	Transmit-data-empty interrupt (TXI) request disabled*	(Initial value)
1	Transmit-data-empty interrupt (TXI) request enabled	

Note: * TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request generation when serial receive data is transferred from RSR to RDR and the RDRF flag in SSR is set to 1.

Bit 6

RIE	Description	
0	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request disabled*	(Initial value)
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled	

Note: * RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF, FER, PER, or ORER flag, then clearing the flag to 0, or clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of serial transmission by the SCI.

Bit 5

TE	Description	
0	Transmission disabled*1	(Initial value)
1	Transmission enabled*2	

Notes: 1. The TDRE flag in SSR is fixed at 1.

2. In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0.

   SMR setting must be performed to decide the transmission format before setting the TE bit to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of serial reception by the SCI.

Bit 4

RE	Description	
0	Reception disabled*1	(Initial value)
1	Reception enabled*2	

Notes: 1. Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.

2. Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in synchronous mode.

   SMR setting must be performed to decide the reception format before setting the RE bit to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE bit setting is only valid in asynchronous mode when receiving with the MP bit in SMR set to 1.

The MPIE bit setting is invalid in synchronous mode or when the MP bit is cleared to 0.

Bit 3

MPIE	Description	
0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received	(Initial value)
1	Multiprocessor interrupts enabled* Receive interrupt (RXI) requests, receive-error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.	

Note: * When receive data including MPB = 0 is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and ORER flags in SSR, is not performed. When receive data with MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting is enabled.

Bit 2—Transmit End Interrupt Enable (TEIE): Enables or disables transmit-end interrupt (TEI) request generation if there is no valid transmit data in TDR when the MSB is transmitted.

Bit 2

TEIE	Description	
0	Transmit-end interrupt (TEI) request disabled*	(Initial value)
1	Transmit-end interrupt (TEI) request enabled*	

Note: * TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the SCI clock source and enable or disable clock output from the SCK pin. The combination of the CKE1 and CKE0 bits determines whether the SCK pin functions as an I/O port, the serial clock output pin, or the serial clock input pin.

The setting of the CKE0 bit, however, is only valid for internal clock operation (CKE1 = 0) in asynchronous mode. The CKE0 bit setting is invalid in synchronous mode, and in the case of external clock operation (CKE1 = 1). The setting of bits CKE1 and CKE0 must be carried out before the SCI's operating mode is determined using SMR.

For details of clock source selection, see table 15.9 in section 15.3, Operation.

Bit 1	Bit 0		
CKE1	CKE0	Description	
0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port*1
		Synchronous mode	Internal clock/SCK pin functions as serial clock output*1
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*2
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*3
		Synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*3
		Synchronous mode	External clock/SCK pin functions as serial clock input

Notes: 1. Initial value
2. Outputs a clock of the same frequency as the bit rate.
3. Inputs a clock with a frequency 16 times the bit rate.

15.2.7 Serial Status Register (SSR)

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

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

SSR is an 8-bit register containing status flags that indicate the operating status of the SCI, and multiprocessor bits.

SSR can be read or written to by the CPU at all times. However, 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER. Also note that in order to clear these flags they must be read as 1 beforehand. The TEND flag and MPB flag are read-only flags and cannot be modified.

SSR is initialized to H'84 by a reset, and in standby mode, and module stop mode.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that data has been transferred from TDR to TSR and the next serial data can be written to TDR.

Bit 7 TDRE	Description
0	[Clearing condition] When 0 is written in TDRE after reading TDRE = 1
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Bit 6—Receive Data Register Full (RDRF): Indicates that the received data is stored in RDR.

Bit 6

RDRF	Description
0	[Clearing condition] (Initial value) When 0 is written in RDRF after reading RDRF = 1
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Note: RDR and the RDRF flag are not affected and retain their previous values when an error is detected during reception or when the RE bit in SCR is cleared to 0.
If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that an overrun error occurred during reception, causing abnormal termination.

Bit 5

ORER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written in ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1*2

Notes: 1. The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
2. The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In synchronous mode, serial transmission cannot be continued, either.

Bit 4—Framing Error (FER): Indicates that a framing error occurred during reception in asynchronous mode, causing abnormal termination.

Bit 4

FER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written in FER after reading FER = 1
1	[Setting condition] When the SCI checks the stop bit at the end of the receive data when reception ends, and the stop bit is 0*2

Notes: 1. The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

2. In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the FER flag is set to 1. In synchronous mode, serial transmission cannot be continued, either.

Bit 3—Parity Error (PER): Indicates that a parity error occurred during reception using parity addition in asynchronous mode, causing abnormal termination.

Bit 3

PER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written in PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR*2

Notes: 1. The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

2. If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In synchronous mode, serial transmission cannot be continued, either.

Bit 2—Transmit End (TEND): Indicates that there is no valid data in TDR when the last bit of the transmit character is sent, and transmission has been ended.

The TEND flag is read-only and cannot be modified.

Bit 2

TEND	Description
0	[Clearing condition] When 0 is written in TDRE after reading TDRE = 1
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Bit 1—Multiprocessor Bit (MPB): When reception is performed using a multiprocessor format in asynchronous mode, MPB stores the multiprocessor bit in the receive data.

MPB is a read-only bit, and cannot be modified.

Bit 1

MPB	Description
0	[Clearing condition] (Initial value)* When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Note: * Retains its previous state when the RE bit in SCR is cleared to 0 with multiprocessor format.

Bit 0—Multiprocessor Bit Transfer (MPBT): When transmission is performed using a multiprocessor format in asynchronous mode, MPBT stores the multiprocessor bit to be added to the transmit data.

The MPBT bit setting is invalid when a multiprocessor format is not used, when not transmitting, and in synchronous mode.

Bit 0

MPBT	Description
0	Data with a 0 multiprocessor bit is transmitted (Initial value)
1	Data with a 1 multiprocessor bit is transmitted

15.2.8 Bit Rate Register (BRR)

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

BRR is an 8-bit register that sets the serial transfer bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SMR.

BRR can be read or written to by the CPU at all times.

BRR is initialized to H'FF by a reset, and in standby mode, and module stop mode.

As baud rate generator control is performed independently for each channel, different values can be set for each channel.

Table 15.3 shows sample BRR settings in asynchronous mode, and table 15.4 shows sample BRR settings in synchronous mode.

Table 15.3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	Operating Frequency ϕ (MHz)											
	ϕ = 2 MHz			ϕ = 2.097152 MHz			ϕ = 2.4576 MHz			ϕ = 3 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	148	–0.04	1	174	–0.26	1	212	0.03
150	1	103	0.16	1	108	0.21	1	127	0.00	1	155	0.16
300	0	207	0.16	0	217	0.21	0	255	0.00	1	77	0.16
600	0	103	0.16	0	108	0.21	0	127	0.00	0	155	0.16
1200	0	51	0.16	0	54	–0.70	0	63	0.00	0	77	0.16
2400	0	25	0.16	0	26	1.14	0	31	0.00	0	38	0.16
4800	0	12	0.16	0	13	–2.48	0	15	0.00	0	19	–2.34
9600	—	—	—	0	6	–2.48	0	7	0.00	0	9	–2.34
19200	—	—	—	—	—	—	0	3	0.00	0	4	–2.34
31250	0	1	0.00	—	—	—	—	—	—	0	2	0.00
38400	—	—	—	—	—	—	0	1	0.00	—	—	—

	Operating Frequency ϕ (MHz)											
	ϕ = 3.6864 MHz			ϕ = 4 MHz			ϕ = 4.9152 MHz			ϕ = 5 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	64	0.70	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	191	0.00	1	207	0.16	1	255	0.00	2	64	0.16
300	1	95	0.00	1	103	0.16	1	127	0.00	1	129	0.16
600	0	191	0.00	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	95	0.00	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	47	0.00	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	23	0.00	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	11	0.00	0	12	0.16	0	15	0.00	0	15	1.73
19200	0	5	0.00	—	—	—	0	7	0.00	0	7	1.73
31250	—	—	—	0	3	0.00	0	4	–1.70	0	4	0.00
38400	0	2	0.00	—	—	—	0	3	0.00	0	3	1.73

	Operating Frequency φ (MHz)											
	φ = 6 MHz			φ = 6.144 MHz			φ = 7.3728 MHz			φ = 8 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	—	—	—	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	—	—	—

	Operating Frequency φ (MHz)											
	φ = 9.8304 MHz			φ = 10 MHz			φ = 12 MHz			φ = 12.288 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

	Operating Frequency ϕ (MHz)											
	ϕ = 14 MHz			ϕ = 14.7456 MHz			ϕ = 16 MHz			ϕ = 17.2032 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	75	0.48
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16	1.20
38400	—	—	—	0	11	0.00	0	12	0.16	0	13	0.00

	Operating Frequency ϕ (MHz)								
	ϕ = 18 MHz			ϕ = 19.6608 MHz			ϕ = 20 MHz		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	79	–0.12	3	86	0.31	3	88	–0.25
150	2	233	0.16	2	255	0.00	3	64	0.16
300	2	116	0.16	2	127	0.00	2	129	0.16
600	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	17	0.00	0	19	–1.70	0	19	0.00
38400	0	14	–2.34	0	15	0.00	0	15	1.73

Table 15.4 BRR Settings for Various Bit Rates (Synchronous Mode)

Bit Rate	Operating Frequency φ (MHz)											
	φ = 2 MHz		φ = 4 MHz		φ = 8 MHz		φ = 10 MHz		φ = 16 MHz		φ = 20 MHz	
(bits/s)	n	N	n	N	n	N	n	N	n	N	n	N
110	3	70	—	—								
250	2	124	2	249	3	124	—	—	3	249		
500	1	249	2	124	2	249	—	—	3	124	—	—
1 k	1	124	1	249	2	124	—	—	2	249	—	—
2.5 k	0	199	1	99	1	199	1	249	2	99	2	124
5 k	0	99	0	199	1	99	1	124	1	199	1	249
10 k	0	49	0	99	0	199	0	249	1	99	1	124
25 k	0	19	0	39	0	79	0	99	0	159	0	199
50 k	0	9	0	19	0	39	0	49	0	79	0	99
100 k	0	4	0	9	0	19	0	24	0	39	0	49
250 k	0	1	0	3	0	7	0	9	0	15	0	19
500 k	0	0*	0	1	0	3	0	4	0	7	0	9
1 M			0	0*	0	1			0	3	0	4
2.5 M							0	0*			0	1
5 M											0	0*

Legend:

Blank: Cannot be set.

—: Can be set, but there will be a degree of error.

*: Continuous transfer is not possible.

Note: As far as possible, the setting should be made so that the error is no more than 1%.

The BRR setting is found from the following equations.

Asynchronous mode:

$$N = \frac{\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Synchronous mode:

$$N = \frac{\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bits/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ϕ: Operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ϕ	0	0
1	ϕ/4	0	1
2	ϕ/16	1	0
3	ϕ/64	1	1

The bit rate error in asynchronous mode is found from the following equation:

$$\text{Error (\%)} = \left\{ \frac{\phi \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 15.5 shows the maximum bit rate for each frequency in asynchronous mode. Tables 15.6 and 15.7 show the maximum bit rates with external clock input.

Table 15.5 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ϕ (MHz)	Maximum Bit Rate (bits/s)	n	N
2	62500	0	0
2.097152	65536	0	0
2.4576	76800	0	0
3	93750	0	0
3.6864	115200	0	0
4	125000	0	0
4.9152	153600	0	0
5	156250	0	0
6	187500	0	0
6.144	192000	0	0
7.3728	230400	0	0
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0

Table 15.6 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ϕ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.5000	31250
2.097152	0.5243	32768
2.4576	0.6144	38400
3	0.7500	46875
3.6864	0.9216	57600
4	1.0000	62500
4.9152	1.2288	76800
5	1.2500	78125
6	1.5000	93750
6.144	1.5360	96000
7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500

Table 15.7 Maximum Bit Rate with External Clock Input (Synchronous Mode)

φ (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bits/s)
2	0.3333	333333.3
4	0.6667	666666.7
6	1.0000	1000000.0
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3

15.2.9 Serial Interface Mode Register (SCMR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	SMIF
Initial value	1	1	1	1	0	0	1	0
Read/Write	—	—	—	—	R/W	R/W	—	R/W

SCMR is an 8-bit readable/writable register used to select SCI functions.

SCMR is initialized to H'F2 by a reset, and in standby mode, and module stop mode.

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bit 3—Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

Bit 3

SDIR	Description	
0	TDR contents are transmitted LSB-first	(Initial value)
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

Bit 2—Data Invert (SINV): Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit(s): parity bit inversion requires inversion of the O/$\overline{E}$ bit in SMR.

Bit 2

SINV	Description	
0	TDR contents are transmitted without modification	(Initial value)
	Receive data is stored in RDR without modification	
1	TDR contents are inverted before being transmitted	
	Receive data is stored in RDR in inverted form	

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

Bit 0—Serial Communication Interface Mode Select (SMIF): Reserved bit. 1 should not be written in this bit.

Bit 0

SMIF	Description	
0	Normal SCI mode	(Initial value)
1	Reserved mode	

15.2.10 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP 15	MSTP 14	MSTP 13	MSTP 12	MSTP 11	MSTP 10	MSTP 9	MSTP 8	MSTP 7	MSTP 6	MSTP 5	MSTP 4	MSTP 3	MSTP 2	MSTP 1	MSTP 0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR, comprising two 8-bit readable/writable registers, performs module stop mode control. When bits MSTP7 is set to 1, SCI0 operation, respectively, stops at the end of the bus cycle and a transition is made to module stop mode. For details, see section 21.5., Module Stop Mode.

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

Bit 7—Module Stop (MSTP7): Specifies the SCI0 module stop mode.

Bit 7

MSTP7	Description	
0	SCI0 module stop mode is cleared	
1	SCI0 module stop mode is set	(Initial value)

15.3 Operation

15.3.1 Overview

The SCI can carry out serial communication in two modes: asynchronous mode in which synchronization is achieved character by character, and synchronous mode in which synchronization is achieved with clock pulses.

Selection of asynchronous or synchronous mode and the transmission format is made using SMR as shown in table 15.8. The SCI clock is determined by a combination of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 15.9.

Asynchronous Mode

- Data length: Choice of 7 or 8 bits
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bits (the combination of these parameters determines the transfer format and character length)
- Detection of framing, parity, and overrun errors, and breaks, during reception
- Choice of internal or external clock as SCI clock source
  - When internal clock is selected:

    The SCI operates on the baud rate generator clock and a clock with the same frequency as the bit rate can be output
  - When external clock is selected:

    A clock with a frequency of 16 times the bit rate must be input (the built-in baud rate generator is not used)

Synchronous Mode

- Transfer format: Fixed 8-bit data
- Detection of overrun errors during reception
- Choice of internal or external clock as SCI clock source
  - When internal clock is selected:

    The SCI operates on the baud rate generator clock and a serial clock is output off-chip
  - When external clock is selected:

    The built-in baud rate generator is not used, and the SCI operates on the input serial clock

Table 15.8 SMR Settings and Serial Transfer Format Selection

SMR Settings						SCI Transfer Format			
Bit 7	Bit 6	Bit 2	Bit 5	Bit 3		Data Length	Multi-processor Bit	Parity Bit	Stop Bit Length
C/$\overline{A}$	CHR	MP	PE	STOP	Mode				
0	0	0	0	0	Asynchronous mode	8-bit data	No	No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	1		0	0		7-bit data		No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Yes	No	1 bit
			—	1					2 bits
	1		—	0		7-bit data			1 bit
			—	1					2 bits
1	—	—	—	—	Synchronous mode	8-bit data	No		None

Table 15.9 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Setting			SCI Transfer Clock	
Bit 7	Bit 1	Bit 0		Clock Source	SCK Pin Function
C/$\overline{A}$	CKE1	CKE0	Mode		
0	0	0	Asynchronous mode	Internal	SCI does not use SCK pin
		1			Outputs clock with same frequency as bit rate
	1	0		External	Inputs clock with frequency of 16 times the bit rate
		1			
1	0	0	Synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

15.3.2 Operation in Asynchronous Mode

In asynchronous mode, characters are sent or received, each preceded by a start bit indicating the start of communication and followed by one or two stop bits indicating the end of communication. Serial communication is thus carried out with synchronization established on a character-by-character basis.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 15.2 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication.

One serial communication character consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally one or two stop bits (high level).

In asynchronous mode, the SCI performs synchronization at the falling edge of the start bit in reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16 times the length of one bit, so that the transfer data is latched at the center of each bit.

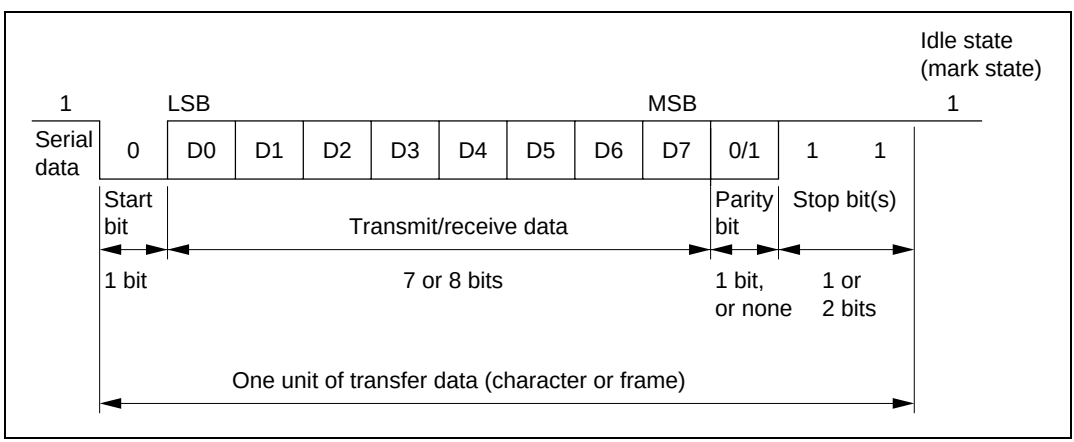


Figure 15.2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

Data Transfer Format

Table 15.10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected by settings in SMR.

Table 15.10 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer Format and Frame Length											
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	S	8-bit data								STOP		
0	0	0	1	S	8-bit data								STOP	STOP	
0	1	0	0	S	8-bit data								P	STOP	
0	1	0	1	S	8-bit data								P	STOP	STOP
1	0	0	0	S	7-bit data							STOP			
1	0	0	1	S	7-bit data							STOP	STOP		
1	1	0	0	S	7-bit data							P	STOP		
1	1	0	1	S	7-bit data							P	STOP	STOP	
0	—	1	0	S	8-bit data								MPB	STOP	
0	—	1	1	S	8-bit data								MPB	STOP	STOP
1	—	1	0	S	7-bit data							MPB	STOP		
1	—	1	1	S	7-bit data							MPB	STOP	STOP	

Legend:
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Clock

Either an internal clock generated by the built-in baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 15.9.

When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is at the center of each transmit data bit, as shown in figure 15.3.

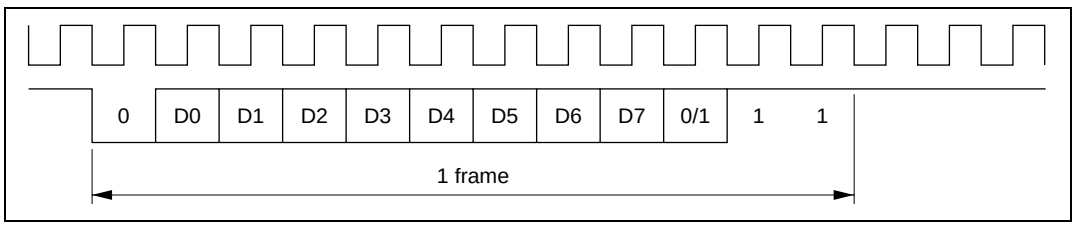


Figure 15.3 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

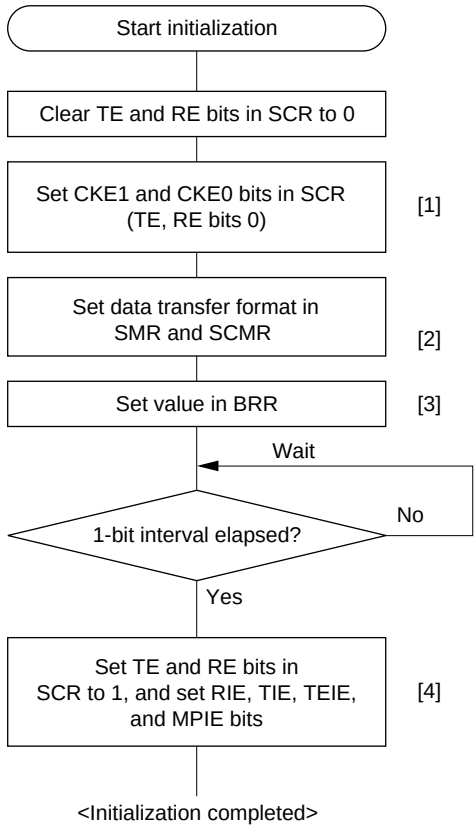
Data Transfer Operations

SCI Initialization (Asynchronous Mode): Before transmitting and receiving data, first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

When an external clock is used the clock should not be stopped during operation, including initialization, since operation is uncertain.

Figure 15.4 shows a sample SCI initialization flowchart.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. This is not necessary if an external clock is used.

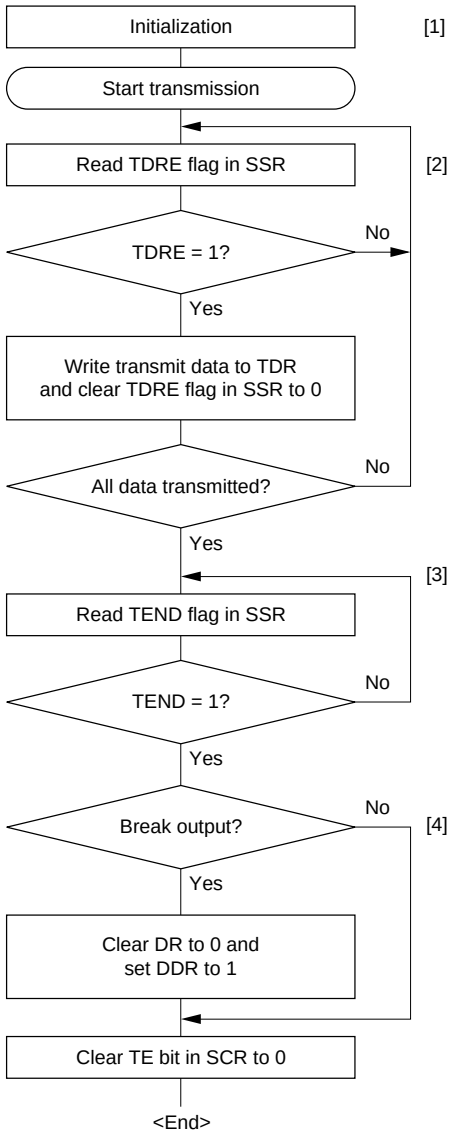
[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.

Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 15.4 Sample SCI Initialization Flowchart

Serial Data Transmission (Asynchronous Mode): Figure 15.5 shows a sample flowchart for serial transmission.

The following procedure should be used for serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, one frame of 1s is output and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 15.5 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission.

   If the TIE bit is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.

   The serial transmit data is sent from the TxD pin in the following order.

   a. Start bit:

      One 0-bit is output.

   b. Transmit data:

      8-bit or 7-bit data is output in LSB-first order.

   c. Parity bit or multiprocessor bit:

      One parity bit (even or odd parity), or one multiprocessor bit is output.

      A format in which neither a parity bit nor a multiprocessor bit is output can also be selected.

   d. Stop bit(s):

      One or two 1-bits (stop bits) are output.

   e. Mark state:

      1 is output continuously until the start bit that starts the next transmission is sent.

3. The SCI checks the TDRE flag at the timing for sending the stop bit.

   If the TDRE flag is cleared to 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 15.6 shows an example of the operation for transmission in asynchronous mode.

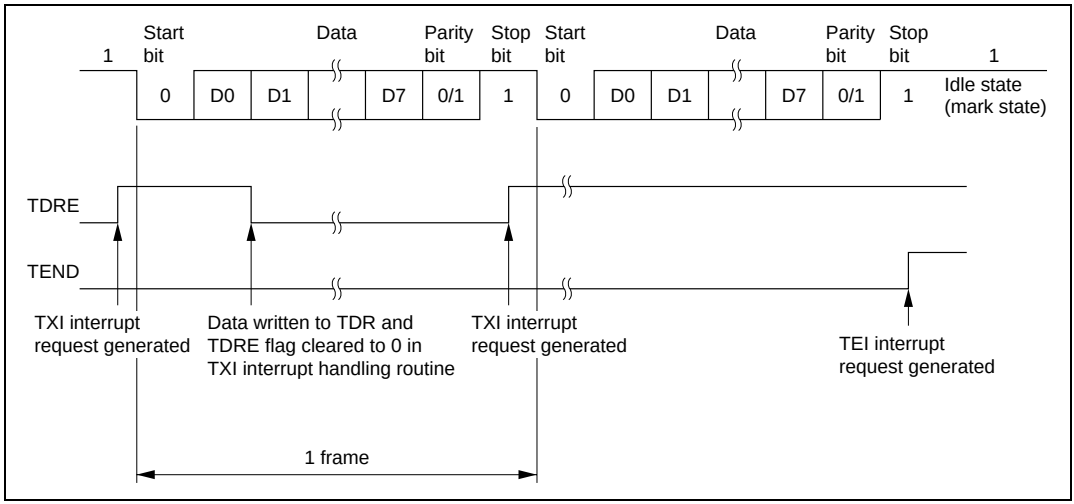


Figure 15.6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

Serial Data Reception (Asynchronous Mode): Figure 15.7 shows a sample flowchart for serial reception.

The following procedure should be used for serial data reception.

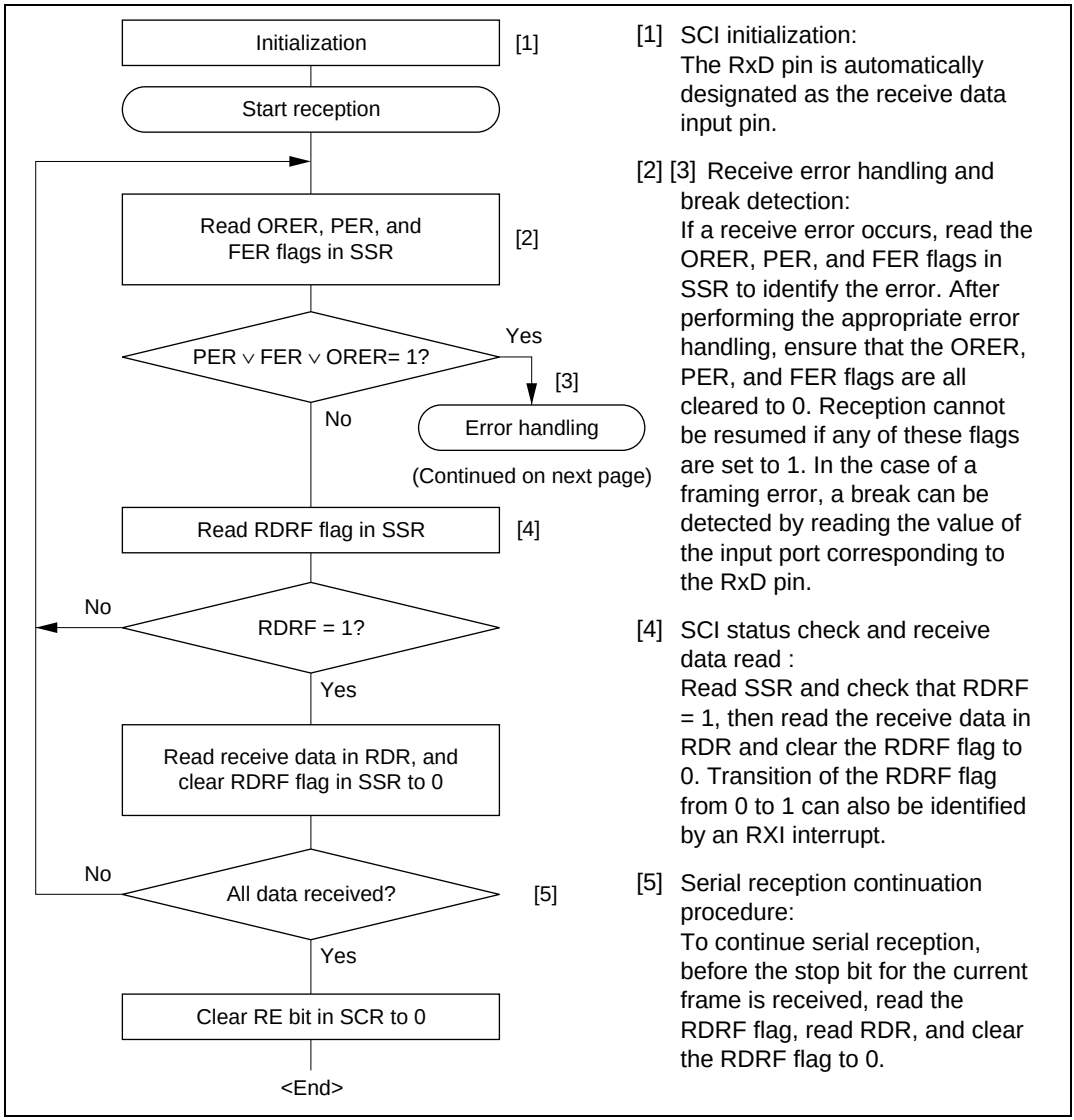


Figure 15.7 Sample Serial Reception Data Flowchart

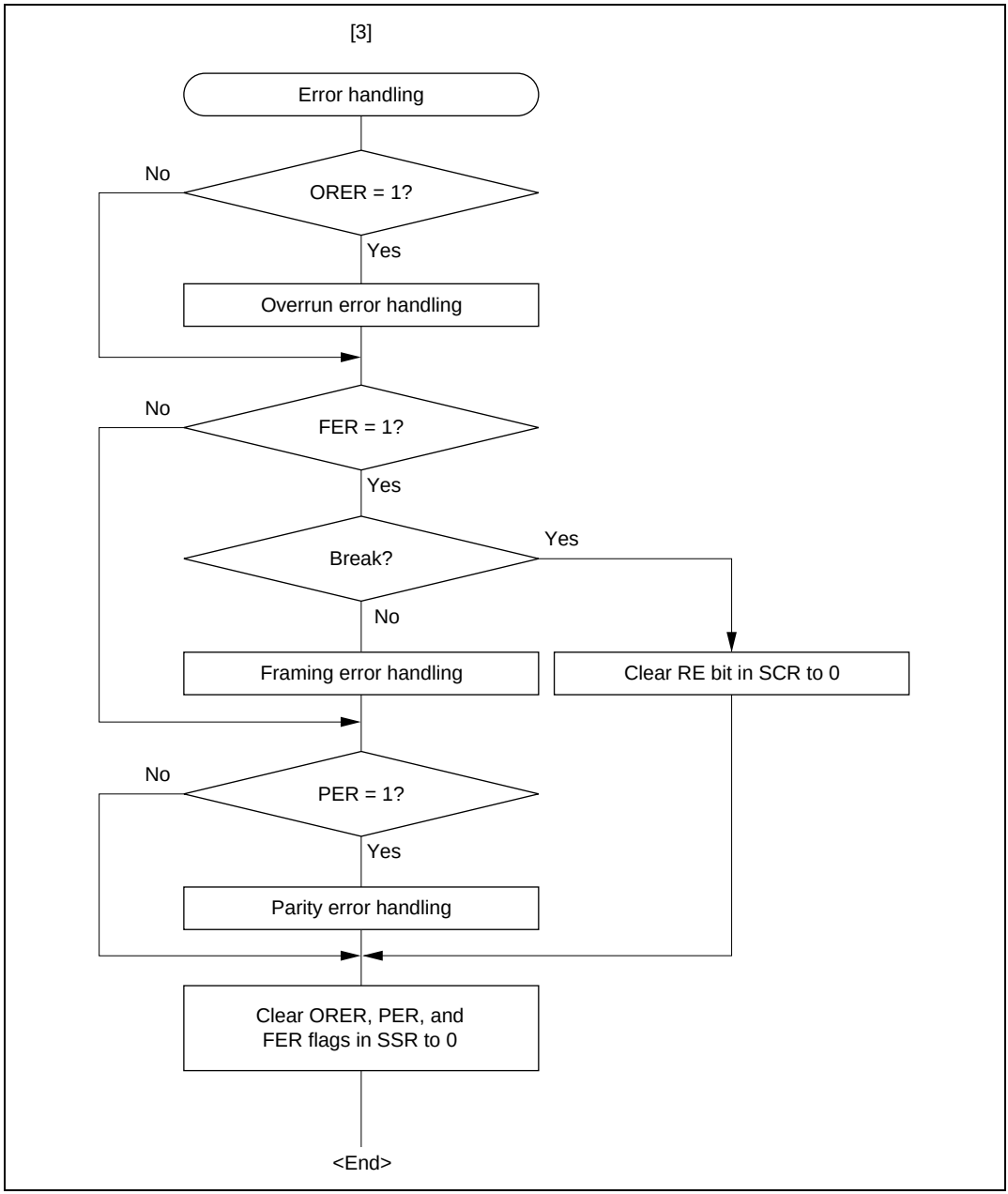


Figure 15.7 Sample Serial Reception Data Flowchart (cont)

In serial reception, the SCI operates as described below.

1. The SCI monitors the transmission line, and if a 0 stop bit is detected, performs internal synchronization and starts reception.

2. The received data is stored in RSR in LSB-to-MSB order.

3. The parity bit and stop bit are received.

   After receiving these bits, the SCI carries out the following checks.

   a. Parity check:

      The SCI checks whether the number of 1 bits in the receive data agrees with the parity (even or odd) set in the O/$\overline{E}$ bit in SMR.

   b. Stop bit check:

      The SCI checks whether the stop bit is 1.

      If there are two stop bits, only the first is checked.

   c. Status check:

      The SCI checks whether the RDRF flag is 0, indicating that the receive data can be transferred from RSR to RDR.

   If all the above checks are passed, the RDRF flag is set to 1, and the receive data is stored in RDR.

   If a receive error* is detected in the error check, the operation is as shown in table 15.11.

Note: * Subsequent receive operations cannot be performed when a receive error has occurred. Also note that the RDRF flag is not set to 1 in reception, and so the error flags must be cleared to 0.

4. If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated.

   Also, if the RIE bit in SCR is set to 1 when the ORER, PER, or FER flag changes to 1, a receive-error interrupt (ERI) request is generated.

Table 15.11 Receive Errors and Conditions for Occurrence

Receive Error	Abbreviation	Occurrence Condition	Data Transfer
Overrun error	ORER	When the next data reception is completed while the RDRF flag in SSR is set to 1	Receive data is not transferred from RSR to RDR
Framing error	FER	When the stop bit is 0	Receive data is transferred from RSR to RDR
Parity error	PER	When the received data differs from the parity (even or odd) set in SMR	Receive data is transferred from RSR to RDR

Figure 15.8 shows an example of the operation for reception in asynchronous mode.

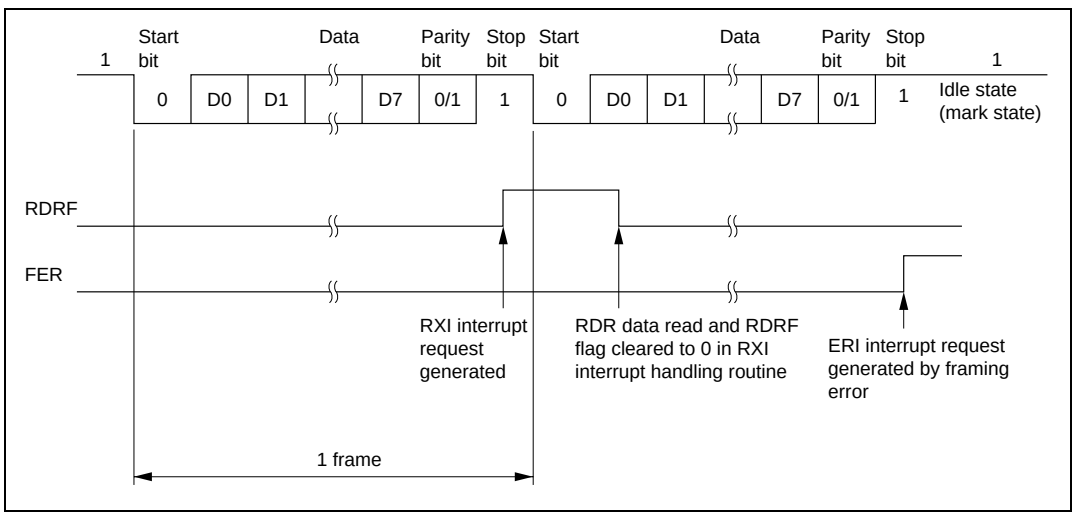


Figure 15.8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

15.3.3 Multiprocessor Communication Function

The multiprocessor communication function performs serial communication using a multiprocessor format, in which a multiprocessor bit is added to the transfer data, in asynchronous mode. Use of this function enables data transfer to be performed among a number of processors sharing transmission lines.

When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code.

The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station, and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle.

The transmitting station first sends the ID of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added.

The receiving station skips the data until data with a 1 multiprocessor bit is sent.

When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip the data until data with a 1 multiprocessor bit is again received. In this way, data communication is carried out among a number of processors.

Figure 15.9 shows an example of inter-processor communication using a multiprocessor format.

Data Transfer Format

There are four data transfer formats.

When a multiprocessor format is specified, the parity bit specification is invalid.

For details, see table 15.10.

Clock

See the section on asynchronous mode.

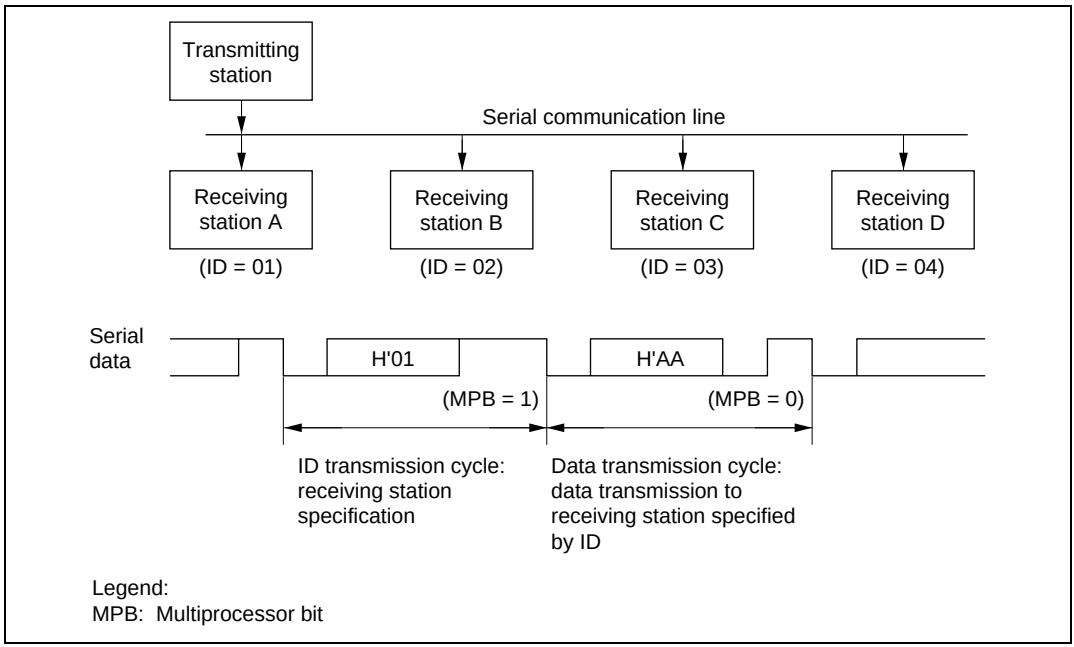
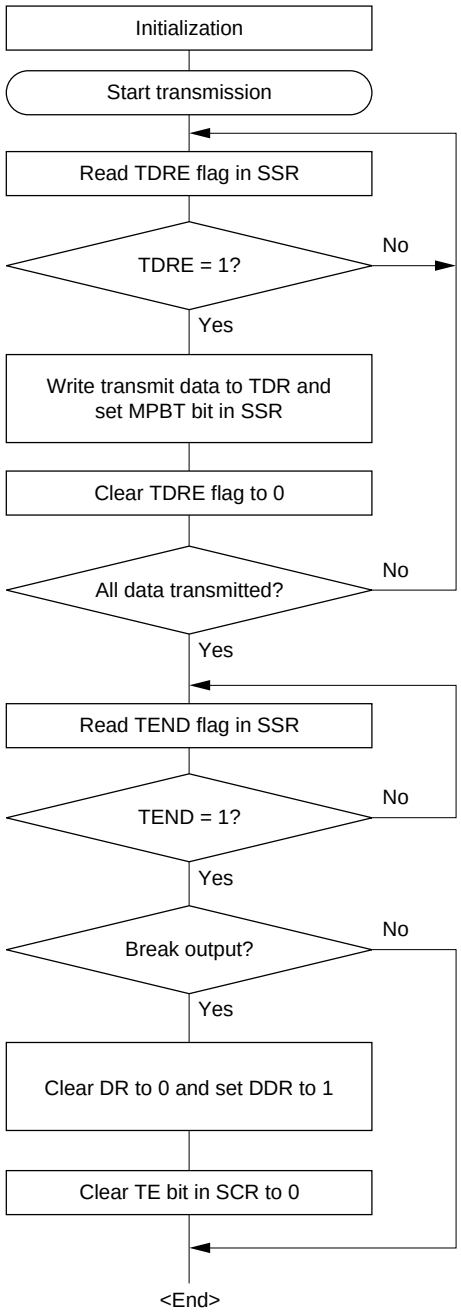


Figure 15.9 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

Data Transfer Operations

Multiprocessor Serial Data Transmission: Figure 15.10 shows a sample flowchart for multiprocessor serial data transmission.

The following procedure should be used for multiprocessor serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, one frame of 1s is output and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1.
Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the port DDR to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 15.10 Sample Multiprocessor Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission.

   If the TIE bit is set to 1 at this time, a transmit-data-empty interrupt (TXI) is generated.

   The serial transmit data is sent from the TxD pin in the following order.

   a. Start bit:

      One 0-bit is output.

   b. Transmit data:

      8-bit or 7-bit data is output in LSB-first order.

   c. Multiprocessor bit

      One multiprocessor bit (MPBT value) is output.

   d. Stop bit(s):

      One or two 1-bits (stop bits) are output.

   e. Mark state:

      1 is output continuously until the start bit that starts the next transmission is sent.

3. The SCI checks the TDRE flag at the timing for sending the stop bit.

   If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

   If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is set to 1 at this time, a transmit-end interrupt (TEI) request is generated.

Figure 15.11 shows an example of SCI operation for transmission using a multiprocessor format.

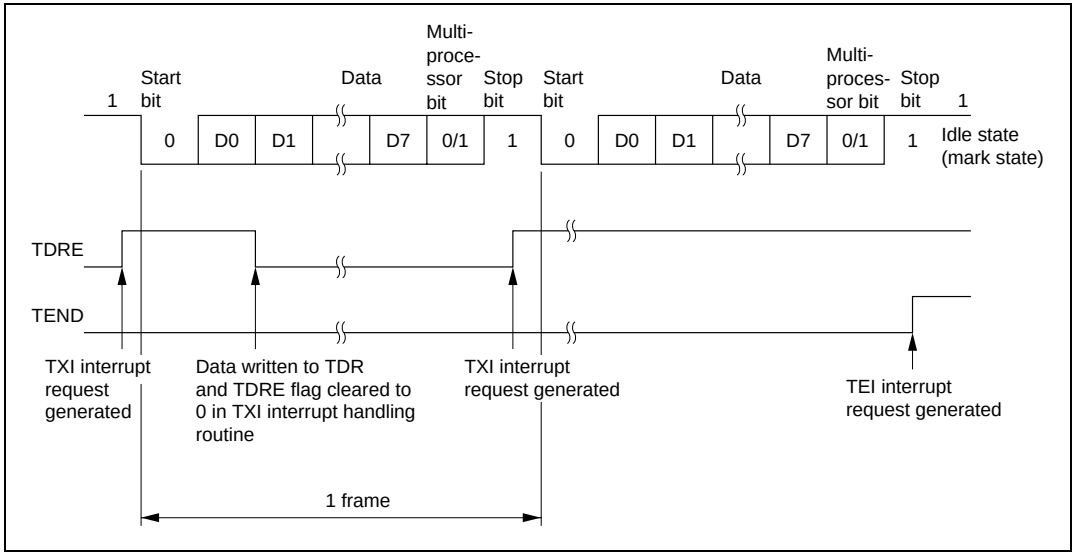
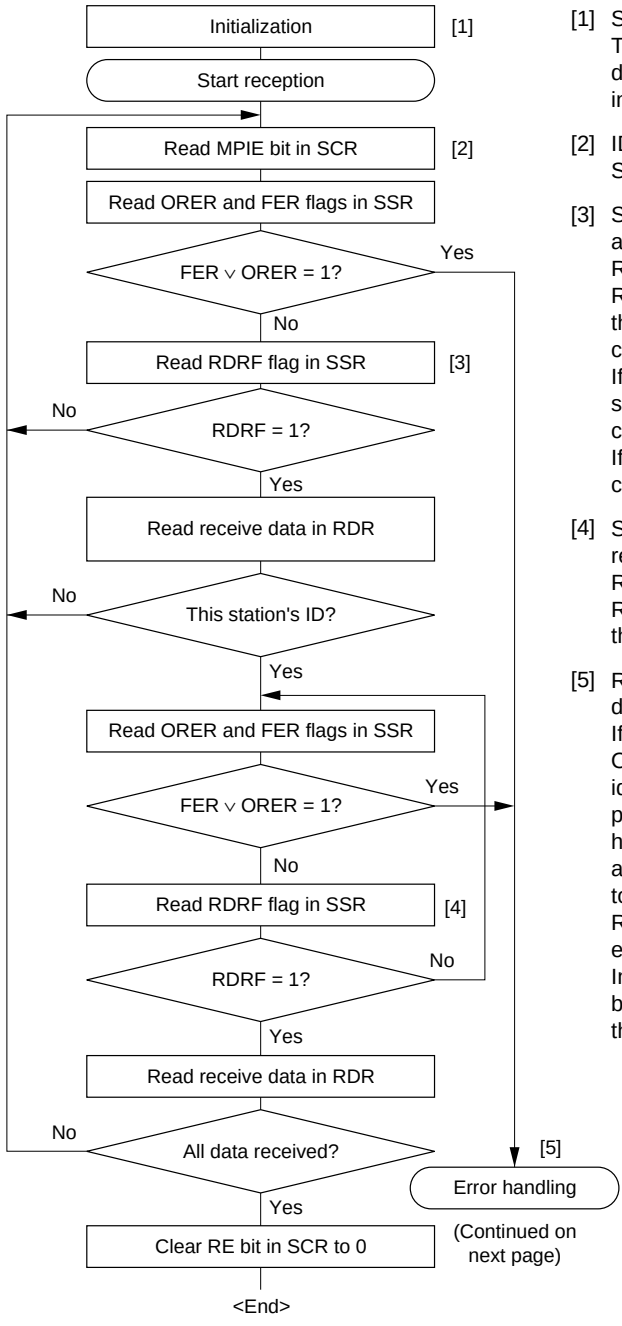


Figure 15.11 Example of SCI Operation in Transmission (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

Multiprocessor Serial Data Reception: Figure 15.12 shows a sample flowchart for multiprocessor serial reception.

The following procedure should be used for multiprocessor serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1.

[3] SCI status check, ID reception and comparison:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error handling and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error handling, ensure that the ORER and FER flags are both cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 15.12 Sample Multiprocessor Serial Reception Flowchart

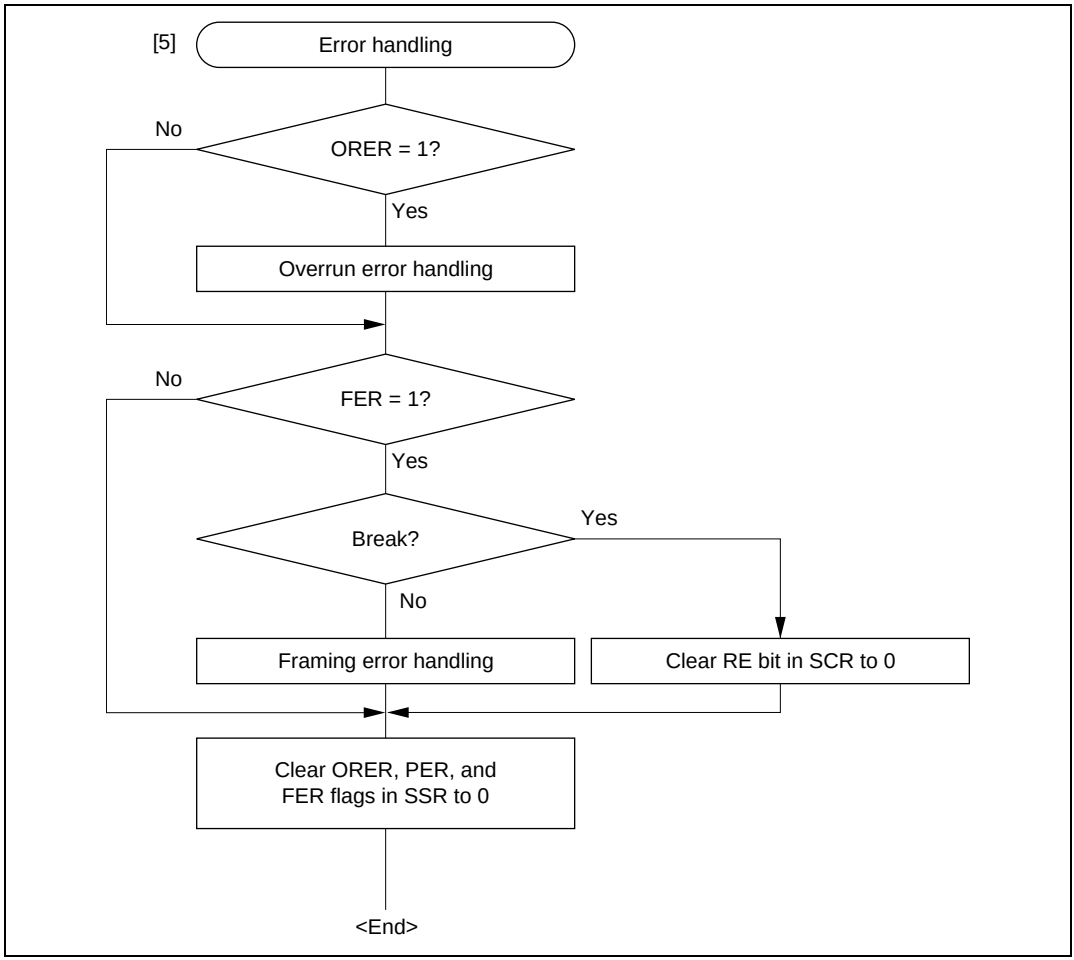


Figure 15.12 Sample Multiprocessor Serial Reception Flowchart (cont)

Figure 15.13 shows an example of SCI operation for multiprocessor format reception.

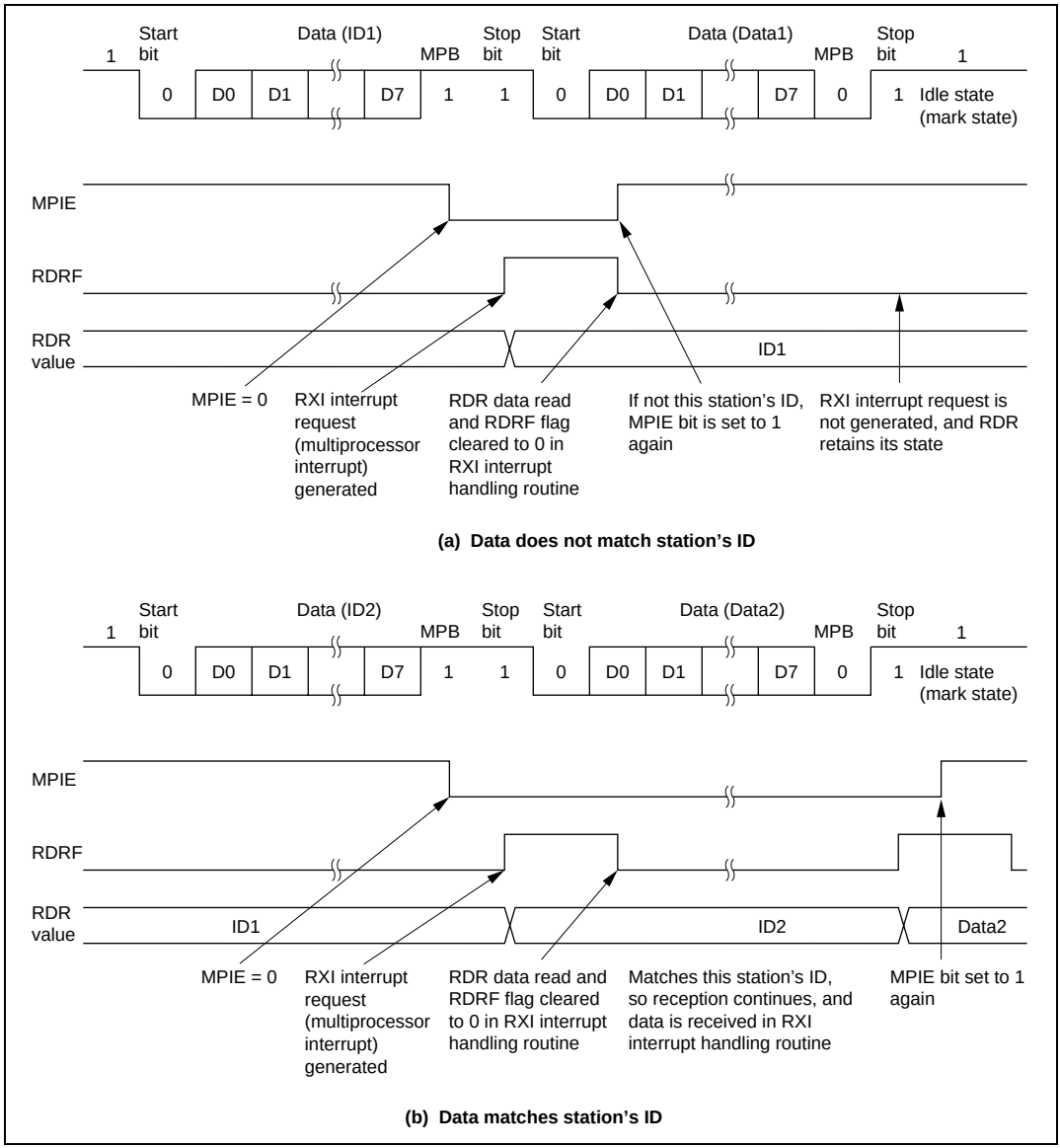


Figure 15.13 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

15.3.4 Operation in Synchronous Mode

In synchronous mode, data is transmitted or received in synchronization with clock pulses, making it suitable for high-speed serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 15.14 shows the general format for synchronous serial communication.

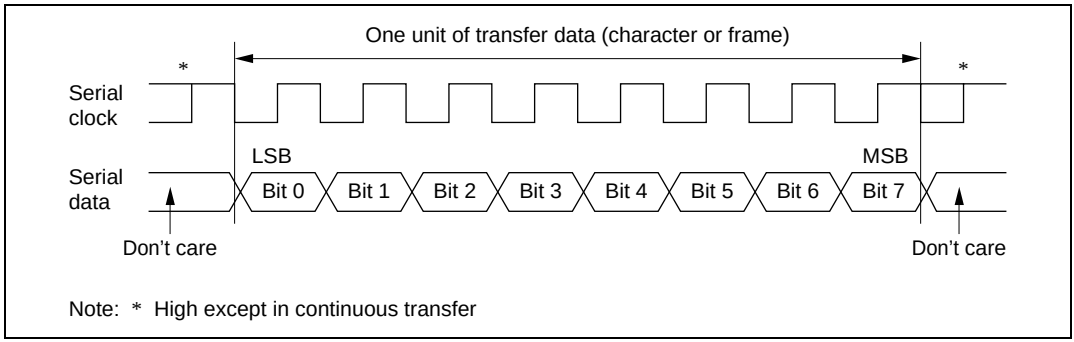


Figure 15.14 Data Format in Synchronous Communication

In synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. Data is guaranteed valid at the rising edge of the serial clock.

In synchronous serial communication, one character consists of data output starting with the LSB and ending with the MSB. After the MSB is output, the transmission line holds the MSB state.

In synchronous mode, the SCI receives data in synchronization with the rising edge of the serial clock.

Data Transfer Format

A fixed 8-bit data format is used.

No parity or multiprocessor bits are added.

Clock

Either an internal clock generated by the built-in baud rate generator or an external serial clock input at the SCK pin can be selected, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details on SCI clock source selection, see table 15.9.

When the SCI is operated on an internal clock, the serial clock is output from the SCK pin.

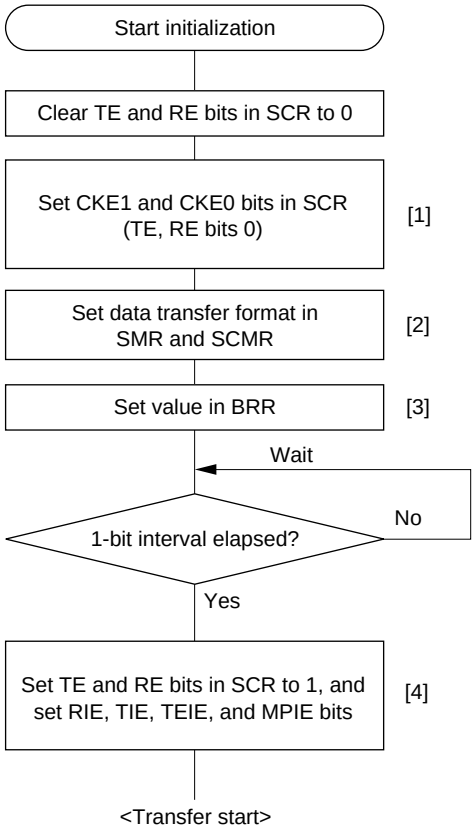
Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high. When only receive operations are performed, however, the serial clock is output until an overrun error occurs or the RE bit is cleared to 0. To perform receive operations in units of one character, select an external clock as the clock source.

Data Transfer Operations

SCI Initialization (Synchronous Mode): Before transmitting and receiving data, first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the settings of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

Figure 15.15 shows a sample SCI initialization flowchart.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, TE and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. This is not necessary if an external clock is used.

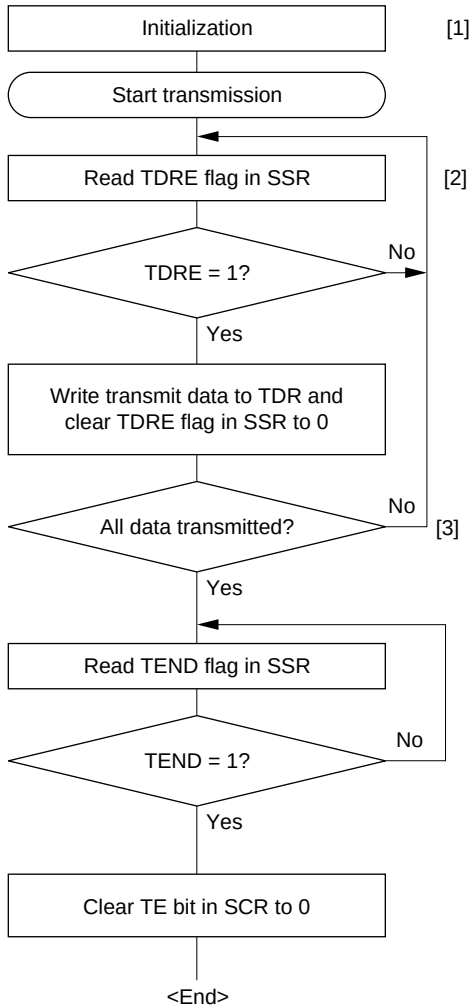
[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used.

Note: In simultaneous transmitting and receiving, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 15.15 Sample SCI Initialization Flowchart

Serial Data Transmission (Synchronous Mode): Figure 15.16 shows a sample flowchart for serial transmission.

The following procedure should be used for serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.

Figure 15.16 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit-data-empty interrupt (TXI) is generated.

   When clock output mode has been set, the SCI outputs 8 serial clock pulses. When use of an external clock has been specified, data is output synchronized with the input clock.

   The serial transmit data is sent from the TxD pin starting with the LSB (bit 0) and ending with the MSB (bit 7).

3. The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).

   If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.

   If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the MSB (bit 7) is sent, and the TxD pin maintains its state.

   If the TEIE bit in SCR is set to 1 at this time, a transmit-end interrupt (TEI) request is generated.

4. After completion of serial transmission, the SCK pin is held in a constant state.

Figure 15.17 shows an example of SCI operation in transmission.

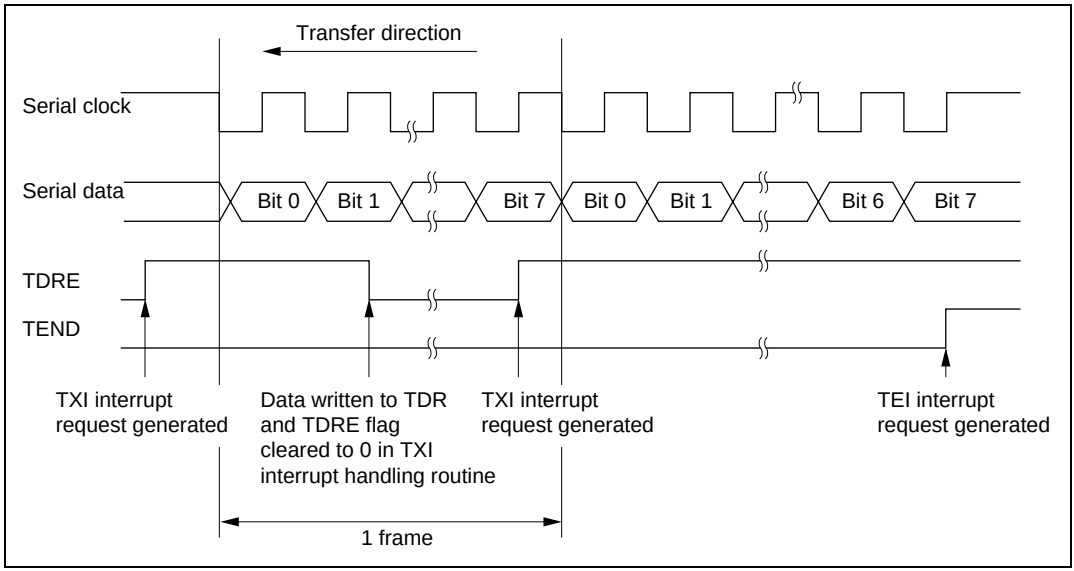


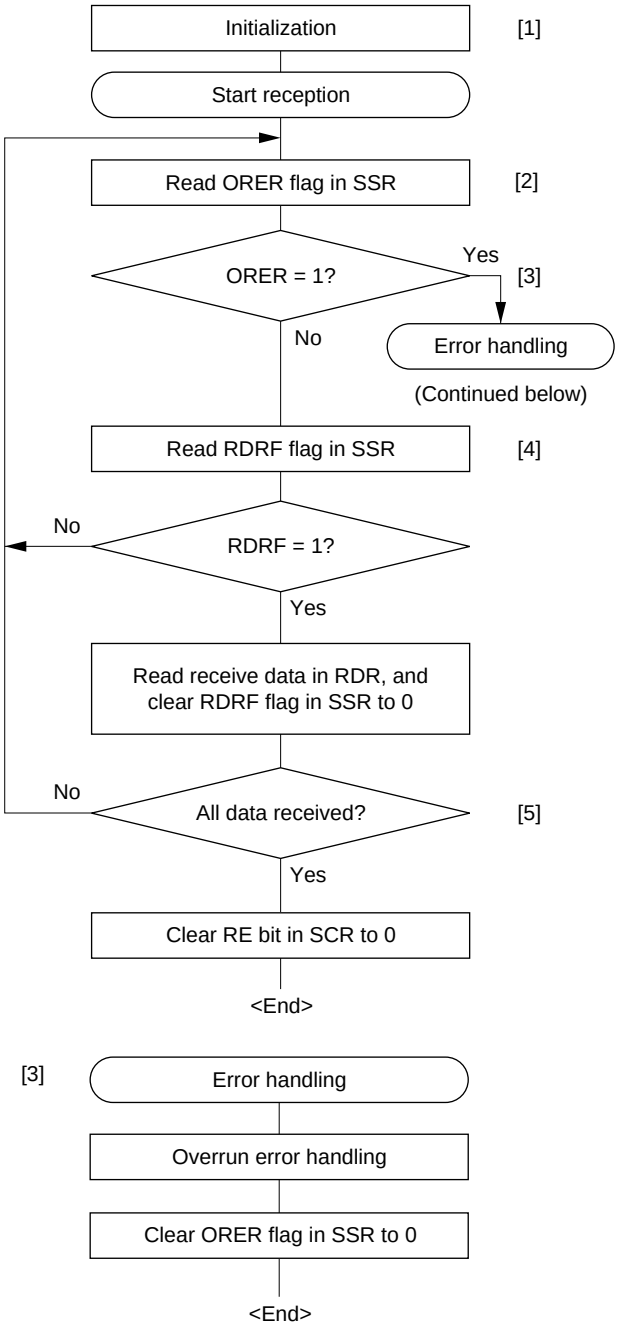
Figure 15.17 Example of SCI Operation in Transmission

Serial Data Reception (Synchronous Mode): Figure 15.18 shows a sample flowchart for serial reception.

The following procedure should be used for serial data reception.

When changing the operating mode from asynchronous to synchronous, be sure to check that the ORER, PER, and FER flags are all cleared to 0.

The RDRF flag will not be set if the FER or PER flag is set to 1, and neither transmit nor receive operations will be possible.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error handling:
If a receive error occurs, read the ORER flag in SSR , and after performing the appropriate error handling, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0.

Figure 15.18 Sample Serial Reception Flowchart

In serial reception, the SCI operates as described below.

1. The SCI performs internal initialization in synchronization with serial clock input or output.

2. The received data is stored in RSR in LSB-to-MSB order.
   After reception, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from RSR to RDR.
   If this check is passed, the RDRF flag is set to 1, and the receive data is stored in RDR. If a receive error is detected in the error check, the operation is as shown in table 15.11.
   Neither transmit nor receive operations can be performed subsequently when a receive error has been found in the error check.

3. If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated.
   Also, if the RIE bit in SCR is set to 1 when the ORER flag changes to 1, a receive-error interrupt (ERI) request is generated.

Figure 15.19 shows an example of SCI operation in reception.

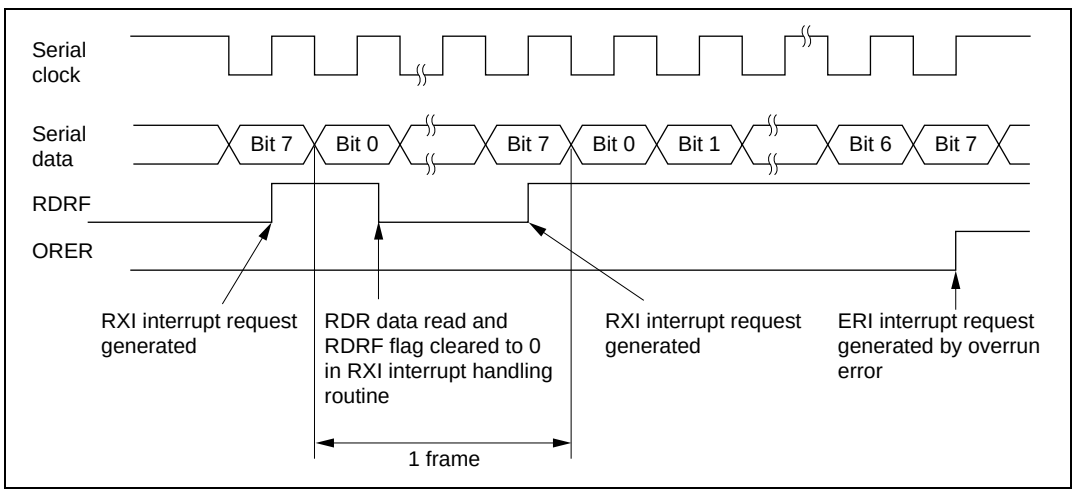
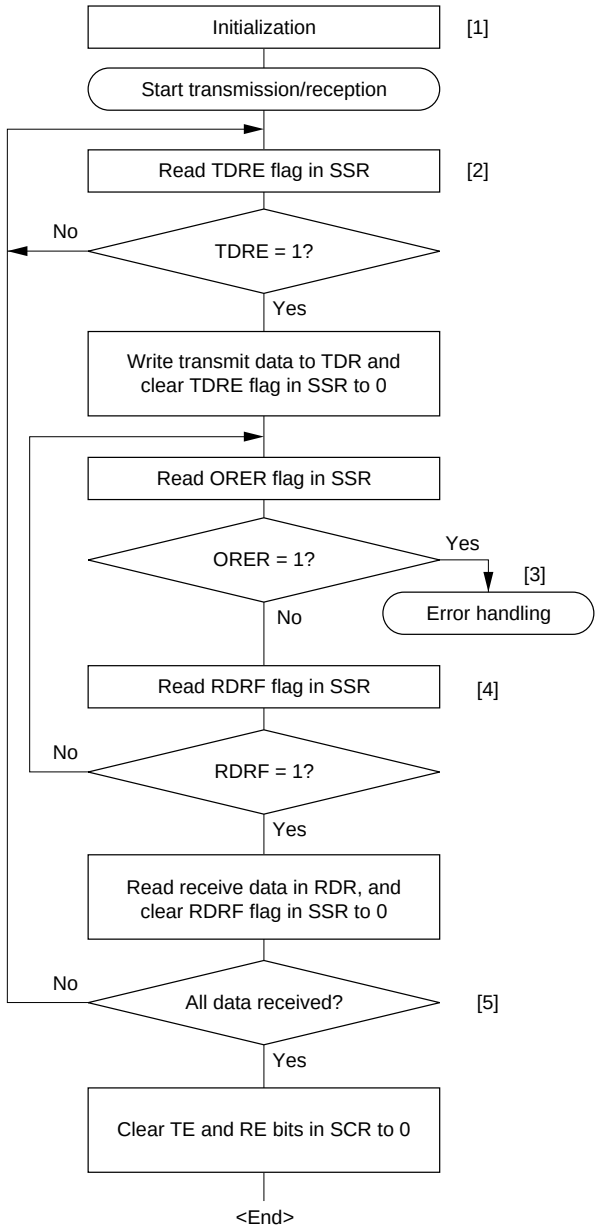


Figure 15.19 Example of SCI Operation in Reception

Simultaneous Serial Data Transmission and Reception (Synchronous Mode): Figure 15.20 shows a sample flowchart for simultaneous serial transmit and receive operations.

The following procedure should be used for simultaneous serial data transmit and receive operations.



[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error handling:
If a receive error occurs, read the ORER flag in SSR , and after performing the appropriate error handling, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/ reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR and clear the TDRE flag to 0.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

Figure 15.20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

15.4 SCI Interrupts

The SCI has four interrupt sources: the transmit-end interrupt (TEI) request, receive-error interrupt (ERI) request, receive-data-full interrupt (RXI) request, and transmit-data-empty interrupt (TXI) request. Table 15.12 shows the interrupt sources and their relative priorities. Individual interrupt sources can be enabled or disabled with the TIE, RIE, and TEIE bits in SCR. Each kind of interrupt request is sent to the interrupt controller independently.

Table 15.12 SCI Interrupt Sources

Interrupt Source	Description	Priority
ERI	Receive error (ORER, FER, or PER)	High
RXI	Receive data register full (RDRF)	↑
TXI	Transmit data register empty (TDRE)	
TEI	Transmit end (TEND)	Low

The TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1. The TEND flag is cleared at the same time as the TDRE flag. Consequently, if a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt will have priority for acceptance, and the TDRE flag and TEND flag may be cleared. Note that the TEI interrupt will not be accepted in this case.

15.5 Usage Notes

The following points should be noted when using the SCI.

Relation between Writes to TDR and the TDRE Flag: The TDRE flag in SSR is a status flag that indicates that transmit data has been transferred from TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to 1.

Data can be written to TDR regardless of the state of the TDRE flag. However, if new data is written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost since it has not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is set to 1 before writing transmit data to TDR.

Operation when Multiple Receive Errors Occur Simultaneously: If a number of receive errors occur at the same time, the state of the status flags in SSR is as shown in table 15.13. If there is an overrun error, data is not transferred from RSR to RDR, and the receive data is lost.

Table 15.13 State of SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	
RDRF	**ORER**	**FER**	**PER**	**RSR to RDR**	**Receive Errors**
1	1	0	0	X	Overrun error
0	0	1	0	O	Framing error
0	0	0	1	O	Parity error
1	1	1	0	X	Overrun error + framing error
1	1	0	1	X	Overrun error + parity error
0	0	1	1	O	Framing error + parity error
1	1	1	1	X	Overrun error + framing error + parity error

Legend:

O: Receive data is transferred from RSR to RDR.

X: Receive data is not transferred from RSR to RDR.

Break Detection and Processing: When a framing error (FER) is detected, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is set, and the parity error flag (PER) may also be set.

Note that, since the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

Sending a Break: The TxD pin has a dual function as an I/O port whose direction (input or output) is determined by DR and DDR. This feature can be used to send a break.

Between serial transmission initialization and setting of the TE bit to 1, the mark state is replaced by the value of DR (the pin does not function as the TxD pin until the TE bit is set to 1). Consequently, DDR and DR for the port corresponding to the TxD pin should first be set to 1.

To send a break during serial transmission, first clear DR to 0, then clear the TE bit to 0.

When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

Receive Error Flags and Transmit Operations (Synchronous Mode Only): Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission.

Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

Receive Data Sampling Timing and Reception Margin in Asynchronous Mode:
In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times the transfer rate.

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the base clock. This is illustrated in figure 15.21.

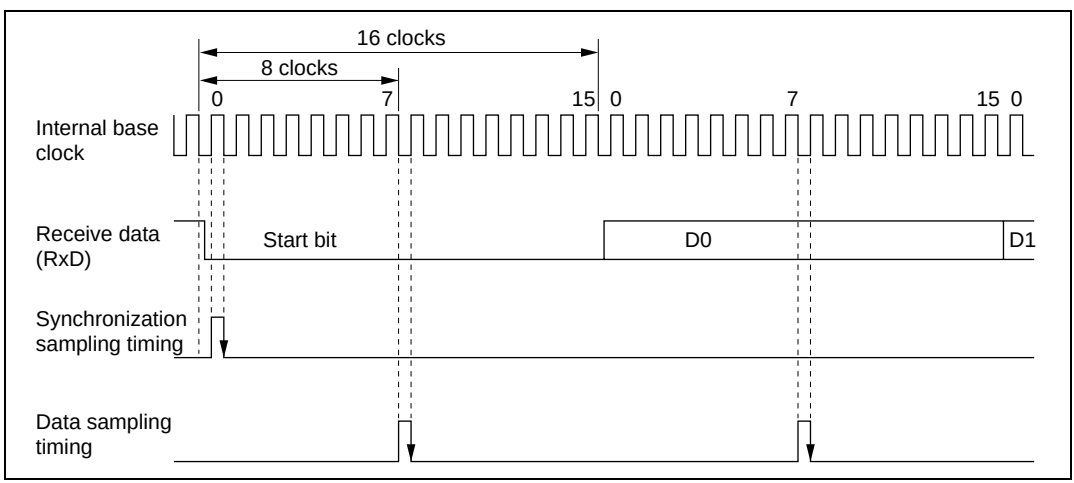


Figure 15.21 Receive Data Sampling Timing in Asynchronous Mode

Thus the receive margin in asynchronous mode is given by equation (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\% \quad \text{..... (1)}$$

Where M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in equation (1), a receive margin of 46.875% is given by equation (2) below.

When D = 0.5 and F = 0,

$$M = \left(0.5 - \frac{1}{2 \times 16}\right) \times 100\%$$
$$= 46.875\% \quad \text{..... (2)}$$

However, this is only a theoretical value, and a margin of 20% to 30% should be allowed in system design.

Section 16 I²C Bus Interface (IIC)

16.1 Overview

The H8/3577 Group and H8/3567 Group have an on-chip two-channel I²C bus interface. The I²C bus interface conforms to and provides a subset of the Philips I²C bus (inter-IC bus) interface functions. The register configuration that controls the I²C bus differs partly from the Philips configuration, however.

Each I²C bus interface channel uses only one data line (SDA) and one clock line (SCL) to transfer data, saving board and connector space.

16.1.1 Features

- Selection of addressing format or non-addressing format
  - — I²C bus format: addressing format with acknowledge bit, for master/slave operation
  - — Serial format: non-addressing format without acknowledge bit, for master operation only
- Conforms to Philips I²C bus interface (I²C bus format)
- Two ways of setting slave address (I²C bus format)
- Start and stop conditions generated automatically in master mode (I²C bus format)
- Selection of acknowledge output levels when receiving (I²C bus format)
- Automatic loading of acknowledge bit when transmitting (I²C bus format)
- Wait function in master mode (I²C bus format)

  A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement. The wait can be cleared by clearing the interrupt flag.
- Wait function in slave mode (I²C bus format)

  A wait request can be generated by driving the SCL pin low after data transfer, excluding acknowledgement. The wait request is cleared when the next transfer becomes possible.
- Three interrupt sources
  - — Data transfer end (including transmission mode transition with I²C bus format and address reception after loss of master arbitration)
  - — Address match: when any slave address matches or the general call address is received in slave receive mode (I²C bus format)
  - — Stop condition detection
- Selection of 16 internal clocks (in master mode)

- Direct bus drive (with SCL and SDA pins)
  - Two pins—$P5_2$/SCL_0 and $P4_7$/SDA_0—(normally NMOS push-pull outputs) function as NMOS open-drain outputs when the bus drive function is selected.
  - Two pins—$P2_4$/SCL_1 and $P2_3$/SDA_1 in the H8/3577 Group, and $P1_7$/SCL_1 and $P1_6$/SDA_1 in the H8/3567 Group—(normally CMOS pins) function as NMOS-only outputs when the bus drive function is selected.
- Automatic switching from formatless mode to I²C bus format (channel 0 only)
  - Formatless operation (no start/stop conditions, non-addressing mode) in slave mode
  - Operation using a common data pin (SDA) and independent clock pins (VSYNCI, SCL)
  - Automatic switching from formatless mode to I²C bus format on the fall of the SCL pin

16.1.2 Block Diagram

Figure 16.1 shows a block diagram of the I²C bus interface.

Figure 16.2 shows an example of I/O pin connections to external circuits. Channel 0 I/O pins and channel 1 I/O pins differ in structure, and have different specifications for permissible applied voltages. For details, see section 22, Electrical Characteristics.

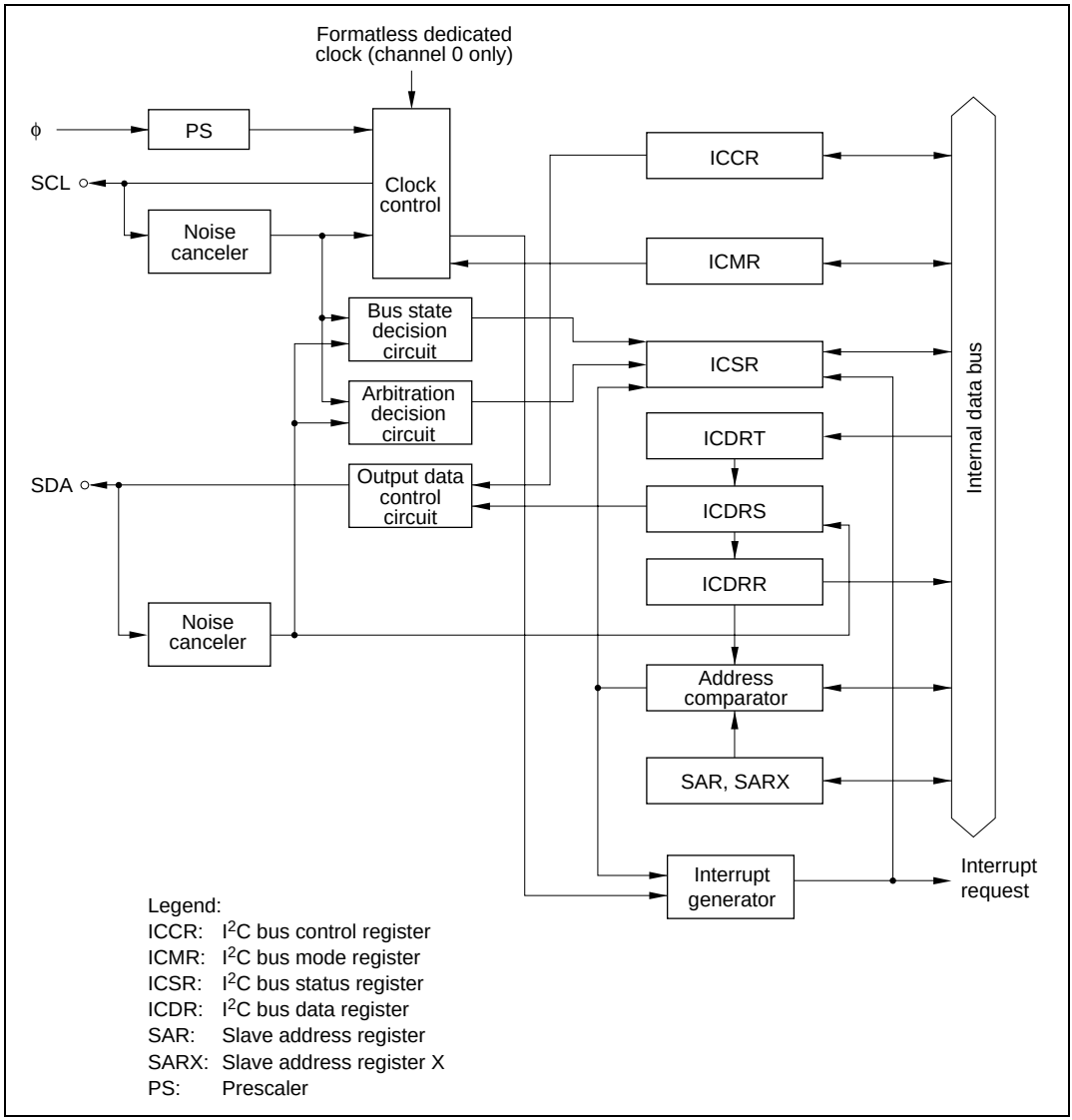


Figure 16.1 Block Diagram of I²C Bus Interface

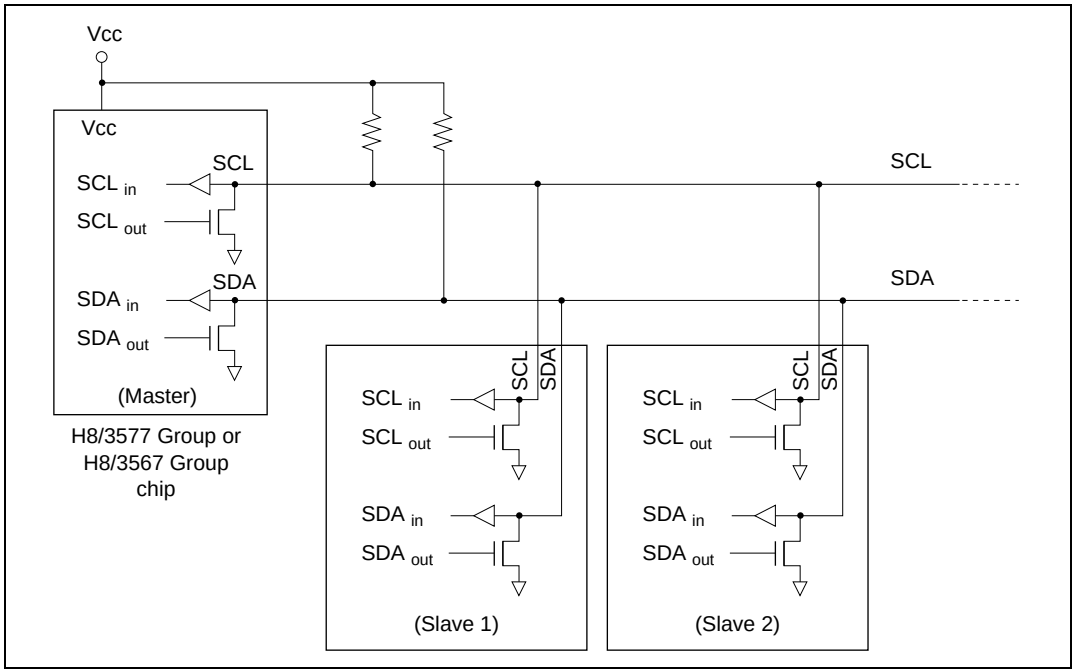


Figure 16.2 I²C Bus Interface Connections
(Example: H8/3577 Group or H8/3567 Group Chip as Master)

16.1.3 Input/Output Pins

Table 16.1 summarizes the input/output pins used by the I²C bus interface.

Table 16.1 I²C Bus Interface Pins

Channel	Name	Abbreviation*	I/O	Function
0	Serial clock	SCL_0	I/O	IIC0 serial clock input/output
	Serial data	SDA_0	I/O	IIC0 serial data input/output
	Formatless serial clock	VSYNCI	Input	IIC0 formatless serial clock input
1	Serial clock	SCL_1	I/O	IIC1 serial clock input/output
	Serial data	SDA_1	I/O	IIC1 serial data input/output

Note: * In the text, the channel subscript is omitted, and only SCL and SDA are used.

16.1.4 Register Configuration

Table 16.2 summarizes the registers of the I²C bus interface.

Table 16.2 Register Configuration

Channel	Name	Abbreviation	R/W	Initial Value	Address
0	I²C bus control register	ICCR0	R/W	H'01	H'FFD8
	I²C bus status register	ICSR0	R/W	H'00	H'FFD9
	I²C bus data register	ICDR0	R/W	—	H'FFDE*
	I²C bus mode register	ICMR0	R/W	H'00	H'FFDF*
	Slave address register	SAR0	R/W	H'00	H'FFDF*
	Second slave address register	SARX0	R/W	H'01	H'FFDE*
1	I²C bus control register	ICCR1	R/W	H'01	H'FF88
	I²C bus status register	ICSR1	R/W	H'00	H'FF89
	I²C bus data register	ICDR1	R/W	—	H'FF8E*
	I²C bus mode register	ICMR1	R/W	H'00	H'FF8F*
	Slave address register	SAR1	R/W	H'00	H'FF8F*
	Second slave address register	SARX1	R/W	H'01	H'FF8E*
Common	Serial timer control register	STCR	R/W	H'00	H'FFC3
	DDC switch register	DDCSWR	R/W	H'0F	H'FEE6
	Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
		MSTPCRL	R/W	H'FF	H'FF87

Note: * The register that can be written or read depends on the ICE bit in the I²C bus control register. The slave address register can be accessed when ICE = 0, and the I²C bus mode register can be accessed when ICE = 1.

The I²C bus interface registers are assigned to the same addresses as other registers. Register selection is performed by means of the IICE bit in the serial timer control register (STCR).

16.2 Register Descriptions

16.2.1 I²C Bus Data Register (ICDR)

Bit	7	6	5	4	3	2	1	0
	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ICDRR

Bit	7	6	5	4	3	2	1	0
	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R	R	R	R	R	R	R	R

- ICDRS

Bit	7	6	5	4	3	2	1	0
	ICDRS7	ICDRS6	ICDRS5	ICDRS4	ICDRS3	ICDRS2	ICDRS1	ICDRS0
Initial value	—	—	—	—	—	—	—	—
Read/Write	—	—	—	—	—	—	—	—

- ICDRT

Bit	7	6	5	4	3	2	1	0
	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0
Initial value	—	—	—	—	—	—	—	—
Read/Write	W	W	W	W	W	W	W	W

- TDRE, RDRF (internal flags)

Bit							—	—
							TDRE	RDRF
Initial value							0	0
Read/Write							—	—

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is divided internally into a shift register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). ICDRS cannot be read or written by the CPU, ICDRR is read-only, and ICDRT is write-only. Data transfers among the three registers are performed automatically in coordination with changes in the bus state, and affect the status of internal flags such as TDRE and RDRF.

If IIC is in transmit mode and the next data is in ICDRT (the TDRE flag is 0) following transmission of one frame of data using ICDRS, data is transferred automatically from ICDRT to ICDRS. If the IIC is in receive mode and none of the previous data remains in ICDRR (the RDRF flag is 0), after one frame of data has been received normally in ICDRS, the data is transferred automatically from ICDRS to ICDRR.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

ICDR is assigned to the same address as SARX, and can be written and read only when the ICE bit is set to 1 in ICCR.

The value of ICDR is undefined after a reset.

The TDRE and RDRF flags are set and cleared under the conditions shown below. Setting the TDRE and RDRF flags affects the status of the interrupt flags.

TDRE	Description
0	The next transmit data is in ICDR (ICDRT), or transmission cannot be started (Initial value) [Clearing conditions] • When transmit data is written in ICDR (ICDRT) in transmit mode (TRS = 1) • When a stop condition is detected in the bus line state after a stop condition is issued with the I²C bus format or serial format selected • When a stop condition is detected with the I²C bus format selected • In receive mode (TRS = 0) (A 0 write to TRS during transfer is valid after reception of a frame containing an acknowledge bit)
1	The next transmit data can be written in ICDR (ICDRT) [Setting conditions] • In transmit mode (TRS = 1), when a start condition is detected in the bus line state after a start condition is issued in master mode with the I²C bus format or serial format selected • When using formatless mode in transmit mode (TRS = 1) • When data is transferred from ICDRT to ICDRS (Data transfer from ICDRT to ICDRS when TRS = 1 and TDRE = 0, and ICDRS is empty) • When a switch is made from receive mode (TRS = 0) to transmit mode (TRS = 1) after detection of a start condition

RDRF	Description
0	The data in ICDR (ICDRR) is invalid (Initial value) [Clearing condition] When ICDR (ICDRR) receive data is read in receive mode
1	The ICDR (ICDRR) receive data can be read [Setting condition] When data is transferred from ICDRS to ICDRR (Data transfer from ICDRS to ICDRR in case of normal termination with TRS = 0 and RDRF = 0)

16.2.2 Slave Address Register (SAR)

Bit	7	6	5	4	3	2	1	0
	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SAR is an 8-bit readable/writable register that stores the slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SAR is assigned to the same address as ICMR, and can be written and read only when the ICE bit is cleared to 0 in ICCR.

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

Bits 7 to 1—Slave Address (SVA6 to SVA0): Set a unique address in bits SVA6 to SVA0, differing from the addresses of other slave devices connected to the I²C bus.

Bit 0—Format Select (FS): Used together with the FSX bit in SARX and the SW bit in DDCSWR to select the communication format.

- I²C bus format: addressing format with acknowledge bit
- Synchronous serial format: non-addressing format without acknowledge bit, for master mode only
- Formatless mode (channel 0 only): non-addressing format with or without acknowledge bit, slave mode only, start/stop conditions not detected

The FS bit also specifies whether or not SAR slave address recognition is performed in slave mode.

DDCSWR Bit 6	SAR Bit 0	SARX Bit 0	
SW	**FS**	**FSX**	**Operating Mode**
0	0	0	I²C bus format • SAR and SARX slave addresses recognized
		1	I²C bus format (Initial value) • SAR slave address recognized • SARX slave address ignored
	1	0	I²C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	0	0	Formatless mode (start/stop conditions not detected)
		1	• Acknowledge bit used
	1	0	
		1	Formatless mode* (start/stop conditions not detected) • No acknowledge bit

Note: * Do not set this mode when automatic switching to the I²C bus format is performed by means of the DDCSWR setting.

16.2.3 Second Slave Address Register (SARX)

Bit	7	6	5	4	3	2	1	0
	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SARX is an 8-bit readable/writable register that stores the second slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SARX match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SARX is assigned to the same address as ICDR, and can be written and read only when the ICE bit is cleared to 0 in ICCR.

SARX is initialized to H'01 by a reset and in hardware standby mode.

Bits 7 to 1—Second Slave Address (SVAX6 to SVAX0): Set a unique address in bits SVAX6 to SVAX0, differing from the addresses of other slave devices connected to the I²C bus.

Bit 0—Format Select X (FSX): Used together with the FS bit in SAR and the SW bit in DDCSWR to select the communication format.

- I²C bus format: addressing format with acknowledge bit
- Synchronous serial format: non-addressing format without acknowledge bit, for master mode only
- Formatless mode: non-addressing format with or without acknowledge bit, slave mode only, start/stop conditions not detected

The FSX bit also specifies whether or not SARX slave address recognition is performed in slave mode. For details, see the description of the FS bit in SAR.

16.2.4 I²C Bus Mode Register (ICMR)

Bit	7	6	5	4	3	2	1	0
	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the master mode transfer clock frequency and the transfer bit count. ICMR is assigned to the same address as SAR. ICMR can be written and read only when the ICE bit is set to 1 in ICCR.

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

Bit 7—MSB-First/LSB-First Select (MLS): Selects whether data is transferred MSB-first or LSB-first.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

Do not set this bit to 1 when the I²C bus format is used.

Bit 7

MLS	Description	
0	MSB-first	(Initial value)
1	LSB-first	

Bit 6—Wait Insertion Bit (WAIT): Selects whether to insert a wait between the transfer of data and the acknowledge bit, in master mode with the I²C bus format. When WAIT is set to 1, after the fall of the clock for the final data bit, the IRIC flag is set to 1 in ICCR, and a wait state begins (with SCL at the low level). When the IRIC flag is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted.

The IRIC flag in ICCR is set to 1 on completion of the acknowledge bit transfer, regardless of the WAIT setting.

The setting of this bit is invalid in slave mode.

Bit 6

WAIT	Description	
0	Data and acknowledge bits transferred consecutively	(Initial value)
1	Wait inserted between data and acknowledge bits	

Bits 5 to 3—Serial Clock Select (CKS2 to CKS0): These bits, together with the IICX1 (channel 1) or IICX0 (channel 0) bit in the STCR register, select the serial clock frequency in master mode. They should be set according to the required transfer rate.

STCR Bit 5 or 6	Bit 5	Bit 4	Bit 3		Transfer Rate				
IICX	CKS2	CKS1	CKS0	Clock	φ = 5 MHz	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ = 20 MHz
0	0	0	0	φ/28	179 kHz	286 kHz	357 kHz	571 kHz*	714 kHz*
			1	φ/40	125 kHz	200 kHz	250 kHz	400 kHz	500 kHz*
		1	0	φ/48	104 kHz	167 kHz	208 kHz	333 kHz	417 kHz*
			1	φ/64	78.1 kHz	125 kHz	156 kHz	250 kHz	313 kHz
	1	0	0	φ/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
			1	φ/100	50.0 kHz	80.0 kHz	100 kHz	160 kHz	200 kHz
		1	0	φ/112	44.6 kHz	71.4 kHz	89.3 kHz	143 kHz	179 kHz
			1	φ/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	0	0	0	φ/56	89.3 kHz	143 kHz	179 kHz	286 kHz	357 kHz
			1	φ/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
		1	0	φ/96	52.1 kHz	83.3 kHz	104 kHz	167 kHz	208 kHz
			1	φ/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
	1	0	0	φ/160	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz	125 kHz
			1	φ/200	25.0 kHz	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz
		1	0	φ/224	22.3 kHz	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz
			1	φ/256	19.5 kHz	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz

Note: * Outside the I²C bus interface specification range (normal mode: max. 100 kHz; high-speed mode: max. 400 kHz).

Bits 2 to 0—Bit Counter (BC2 to BC0): Bits BC2 to BC0 specify the number of bits to be transferred next. With the I²C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the data is transferred with one addition acknowledge bit. Bits BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL line is low.

The bit counter is initialized to 000 by a reset and when a start condition is detected. The value returns to 000 at the end of a data transfer, including the acknowledge bit.

Bit 2	Bit 1	Bit 0	Bits/Frame		
BC2	BC1	BC0	Synchronous Serial Format	I²C Bus Format	
0	0	0	8	9	(Initial value)
		1	1	2	
	1	0	2	3	
		1	3	4	
1	0	0	4	5	
		1	5	6	
	1	0	6	7	
		1	7	8	

16.2.5 I²C Bus Control Register (ICCR)

Bit	7	6	5	4	3	2	1	0
	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	W

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

ICCR is an 8-bit readable/writable register that enables or disables the I²C bus interface, enables or disables interrupts, selects master or slave mode and transmission or reception, enables or disables acknowledgement, confirms the I²C bus interface bus status, issues start/stop conditions, and performs interrupt flag confirmation.

ICCR is initialized to H'01 by a reset and in hardware standby mode.

Bit 7—I²C Bus Interface Enable (ICE): Selects whether or not the I²C bus interface is to be used. When ICE is set to 1, port pins function as SCL and SDA input/output pins and transfer operations are enabled. When ICE bit is cleared to 0, the module stops the functions and clears the internal state.

The SAR and SARX registers can be accessed when ICE is 0. The ICMR and ICDR registers can be accessed when ICE is 1.

Bit 7

ICE	Description
0	I²C bus interface module disabled, with SCL and SDA signal pins set to port function (Initial value)
	Initialization of IIC module internal state
	SAR and SARX can be accessed
1	I²C bus interface module enabled for transfer operations (pins SCL and SCA are driving the bus)
	ICMR and ICDR can be accessed

Bit 6—I²C Bus Interface Interrupt Enable (IEIC): Enables or disables interrupts from the I²C bus interface to the CPU.

Bit 6

IEIC	Description
0	Interrupts disabled (Initial value)
1	Interrupts enabled

Bit 5—Master/Slave Select (MST)

Bit 4—Transmit/Receive Select (TRS)

MST selects whether the I²C bus interface operates in master mode or slave mode.

TRS selects whether the I²C bus interface operates in transmit mode or receive mode.

In master mode with the I²C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. In slave receive mode with the addressing format (FS = 0 or FSX = 0), hardware automatically selects transmit or receive mode according to the R/W bit in the first frame after a start condition.

Modification of the TRS bit during transfer is deferred until transfer of the frame containing the acknowledge bit is completed, and the changeover is made after completion of the transfer.

MST and TRS select the operating mode as follows.

Bit 5	Bit 4		
MST	**TRS**	**Operating Mode**	
0	0	Slave receive mode	(Initial value)
	1	Slave transmit mode	
1	0	Master receive mode	
	1	Master transmit mode	

Bit 5

MST	Description
0	Slave mode (Initial value) [Clearing conditions] 1. When 0 is written by software 2. When bus arbitration is lost after transmission is started in I²C bus format master mode
1	Master mode [Setting conditions] 1. When 1 is written by software (in cases other than clearing condition 2) 2. When 1 is written in MST after reading MST = 0 (in case of clearing condition 2)

Bit 4

TRS	Description
0	Receive mode (Initial value) [Clearing conditions] 1. When 0 is written by software (in cases other than setting condition 3) 2. When 0 is written in TRS after reading TRS = 1 (in case of clearing condition 3) 3. When bus arbitration is lost after transmission is started in I²C bus format master mode 4. When the SW bit in DDCSWR changes from 1 to 0
1	Transmit mode [Setting conditions] 1. When 1 is written by software (in cases other than clearing conditions 3 and 4) 2. When 1 is written in TRS after reading TRS = 0 (in case of clearing conditions 3 and 4) 3. When a 1 is received as the R/W bit of the first frame in I²C bus format slave mode

Bit 3—Acknowledge Bit Judgement Selection (ACKE): Specifies whether the value of the acknowledge bit returned from the receiving device when using the I²C bus format is to be ignored and continuous transfer is performed, or transfer is to be aborted and error handling, etc., performed if the acknowledge bit is 1. When the ACKE bit is 0, the value of the received acknowledge bit is not indicated by the ACKB bit, which is always 0.

Bit 3

ACKE	Description
0	The value of the acknowledge bit is ignored, and continuous transfer is performed (Initial value)
1	If the acknowledge bit is 1, continuous transfer is interrupted

Bit 2—Bus Busy (BBSY): The BBSY flag can be read to check whether the I²C bus (SCL, SDA) is busy or free. In master mode, this bit is also used to issue start and stop conditions.

A high-to-low transition of SDA while SCL is high is recognized as a start condition, setting BBSY to 1. A low-to-high transition of SDA while SCL is high is recognized as a stop condition, clearing BBSY to 0.

To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, use a MOV instruction to write 0 in BBSY and 0 in SCP. It is not possible to write to BBSY in slave mode; the I²C bus interface must be set to master transmit mode before issuing a start condition. MST and TRS should both be set to 1 before writing 1 in BBSY and 0 in SCP.

Bit 2

BBSY	Description
0	Bus is free (Initial value) [Clearing condition] When a stop condition is detected
1	Bus is busy [Setting condition] When a start condition is detected

Bit 1—I²C Bus Interface Interrupt Request Flag (IRIC): Indicates that the I²C bus interface has issued an interrupt request to the CPU. IRIC is set to 1 at the end of a data transfer, when a slave address or general call address is detected in slave receive mode, when bus arbitration is lost in master transmit mode, and when a stop condition is detected. IRIC is set at different times depending on the FS bit in SAR and the WAIT bit in ICMR. See section 16.3.6, IRIC Setting

Timing and SCL Control. The conditions under which IRIC is set also differ depending on the setting of the ACKE bit in ICCR.

IRIC is cleared by reading IRIC after it has been set to 1, then writing 0 in IRIC.

Bit 1

IRIC	Description
0	Waiting for transfer, or transfer in progress (Initial value) [Clearing condition] When 0 is written in IRIC after reading IRIC = 1
1	Interrupt requested [Setting conditions] • I²C bus format master mode — When a start condition is detected in the bus line state after a start condition is issued (when the TDRE flag is set to 1 because of first frame transmission) — When a wait is inserted between the data and acknowledge bit when WAIT = 1 — At the end of data transfer (at the rise of the 9th transmit/receive clock pulse, and, when a wait is inserted, at the fall of the 8th transmit/receive clock pulse) — When a slave address is received after bus arbitration is lost (when the AL flag is set to 1) — When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) • I²C bus format slave mode — When the slave address (SVA, SVAX) matches (when the AAS and AASX flags are set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) — When the general call address is detected (when the FS = 0 and the ADZ flag is set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) — When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) — When a stop condition is detected (when the STOP or ESTP flag is set to 1) • Synchronous serial format, and formatless mode — At the end of data transfer (when the TDRE or RDRF flag is set to 1) — When a start condition is detected with serial format selected — When the SW bit is set to 1 in DDCSWR Besides the above, when a condition that sets the TDRE or RDRF internal flag to 1 occurs

When, with the I²C bus format selected, IRIC is set to 1 and an interrupt is generated, other flags must be checked in order to identify the source that set IRIC to 1. Although each source has a corresponding flag, caution is needed at the end of a transfer.

When the TDRE or RDRF internal flag is set, the readable IRTR flag may or may not be set. The IRTR flag is not set at the end of a data transfer up to detection of a retransmission start condition or stop condition after a slave address (SVA) or general call address match in I²C bus format slave mode.

Even when the IRIC flag and IRTR flag are set, the TDRE or RDRF internal flag may not be set.

Table 16.3 shows the relationship between the flags and the transfer states.

Table 16.3 Flags and Transfer States

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	State
1/0	1/0	0	0	0	0	0	0	0	0	0	Idle state (flag clearing required)
1	1	0	0	0	0	0	0	0	0	0	Start condition issuance
1	1	1	0	0	1	0	0	0	0	0	Start condition established
1	1/0	1	0	0	0	0	0	0	0	0/1	Master mode wait
1	1/0	1	0	0	1	0	0	0	0	0/1	Master mode transmit/receive end
0	0	1	0	0	0	1/0	1	1/0	1/0	0	Arbitration lost
0	0	1	0	0	0	0	0	1	0	0	SAR match by first frame in slave mode
0	0	1	0	0	0	0	0	1	1	0	General call address match
0	0	1	0	0	0	1	0	0	0	0	SARX match
0	1/0	1	0	0	0	0	0	0	0	0/1	Slave mode transmit/receive end (except after SARX match)
0	1/0	1	0	0	1	1	0	0	0	0	Slave mode transmit/receive end (after SARX match)
0	1	1	0	0	0	1	0	0	0	1	
0	1/0	0	1/0	1/0	0	0	0	0	0	0/1	Stop condition detected

Bit 0—Start Condition/Stop Condition Prohibit (SCP): Controls the issuing of start and stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored.

Bit 0

SCP	Description
0	Writing 0 issues a start or stop condition, in combination with the BBSY flag
1	Reading always returns a value of 1 (Initial value) Writing is ignored

16.2.6 I²C Bus Status Register (ICSR)

Bit	7	6	5	4	3	2	1	0
	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

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

ICSR is an 8-bit readable/writable register that performs flag confirmation and acknowledge confirmation and control.

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

Bit 7—Error Stop Condition Detection Flag (ESTP): Indicates that a stop condition has been detected during frame transfer in I²C bus format slave mode.

Bit 7

ESTP	Description	
0	No error stop condition	(Initial value)
	[Clearing conditions]	
	• When 0 is written in ESTP after reading ESTP = 1	
	• When the IRIC flag is cleared to 0	
1	• In I²C bus format slave mode Error stop condition detected [Setting condition] When a stop condition is detected during frame transfer	
	• In other modes No meaning	

Bit 6—Normal Stop Condition Detection Flag (STOP): Indicates that a stop condition has been detected after completion of frame transfer in I²C bus format slave mode.

Bit 6

STOP	Description	
0	No normal stop condition	(Initial value)
	[Clearing conditions]	
	• When 0 is written in STOP after reading STOP = 1	
	• When the IRIC flag is cleared to 0	
1	• In I²C bus format slave mode Normal stop condition detected [Setting condition] When a stop condition is detected after completion of frame transfer	
	• In other modes No meaning	

Bit 5—I²C Bus Interface Continuous Transmission/Reception Interrupt Request Flag (IRTR): Indicates that the I²C bus interface has issued an interrupt request to the CPU, and the source is completion of reception/transmission of one frame in continuous transmission/reception for which DTC activation is possible. As the H8/3577 Group and H8/3567 Group do not have an on-chip DTC, the IRTR flag is used by the CPU to determine the source that set IRIC. When the IRTR flag is set to 1, the IRIC flag is also set to 1 at the same time.

IRTR flag setting is performed when the TDRE or RDRF flag is set to 1. IRTR is cleared by reading IRTR after it has been set to 1, then writing 0 in IRTR. IRTR is also cleared automatically when the IRIC flag is cleared to 0.

Bit 5

IRTR	Description
0	Waiting for transfer, or transfer in progress (Initial value) [Clearing conditions] • When 0 is written in IRTR after reading IRTR = 1 • When the IRIC flag is cleared to 0
1	Continuous transfer state [Setting conditions] • In I²C bus interface slave mode When the TDRE or RDRF flag is set to 1 when AASX = 1 • In other modes When the TDRE or RDRF flag is set to 1

Bit 4—Second Slave Address Recognition Flag (AASX): In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVAX6 to SVAX0 in SARX.

AASX is cleared by reading AASX after it has been set to 1, then writing 0 in AASX. AASX is also cleared automatically when a start condition is detected.

Bit 4

AASX	Description
0	Second slave address not recognized (Initial value) [Clearing conditions] • When 0 is written in AASX after reading AASX = 1 • When a start condition is detected • In master mode
1	Second slave address recognized [Setting condition] When the second slave address is detected in slave receive mode while FSX = 0

Bit 3—Arbitration Lost (AL): This flag indicates that arbitration was lost in master mode. The I²C bus interface monitors the SDA. When two or more master devices attempt to seize the bus at nearly the same time, if the I²C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master.

AL is cleared by reading AL after it has been set to 1, then writing 0 in AL. In addition, AL is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 3

AL	Description
0	Bus arbitration won (Initial value) [Clearing conditions] • When ICDR data is written (transmit mode) or read (receive mode) • When 0 is written in AL after reading AL = 1
1	Arbitration lost [Setting conditions] • If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode • If the internal SCL line is high at the fall of SCL in master transmit mode

Bit 2—Slave Address Recognition Flag (AAS): In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR, or if the general call address (H'00) is detected.

AAS is cleared by reading AAS after it has been set to 1, then writing 0 in AAS. In addition, AAS is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 2

AAS	Description
0	Slave address or general call address not recognized (Initial value) [Clearing conditions] • When ICDR data is written (transmit mode) or read (receive mode) • When 0 is written in AAS after reading AAS = 1 • In master mode
1	Slave address or general call address recognized [Setting condition] When the slave address or general call address is detected in slave receive mode while FS = 0

Bit 1—General Call Address Recognition Flag (ADZ): In I²C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition is the general call address (H'00).

ADZ is cleared by reading ADZ after it has been set to 1, then writing 0 in ADZ. In addition, ADZ is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 1

ADZ	Description
0	General call address not recognized (Initial value) [Clearing conditions] • When ICDR data is written (transmit mode) or read (receive mode) • When 0 is written in ADZ after reading ADZ = 1 • In master mode
1	General call address recognized [Setting condition] When the general call address is detected in slave receive mode while FSX = 0 or FS = 0

Bit 0—Acknowledge Bit (ACKB): Stores acknowledge data. In transmit mode, after the receiving device receives data, it returns acknowledge data, and this data is loaded into ACKB. In receive mode, after data has been received, the acknowledge data set in this bit is sent to the transmitting device.

When this bit is read, in transmission (when TRS = 1), the value loaded from the bus line (returned by the receiving device) is read. In reception (when TRS = 0), the value set by internal software is read.

Also, when this bit is written, the set value of the acknowledge data to be issued upon receiving is rewritten, regardless of the TRS value. Since the value loaded from the receiving device is held, as is, in this case, care is required when rewriting this register using a bit operation command.

Bit 0

ACKB	Description
0	Receive mode: 0 is output at acknowledge output timing (Initial value) Transmit mode: Indicates that the receiving device has acknowledged the data (signal is 0)
1	Receive mode: 1 is output at acknowledge output timing Transmit mode: Indicates that the receiving device has not acknowledged the data (signal is 1)

16.2.7 Serial Timer Control Register (STCR)

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

STCR is an 8-bit readable/writable register that controls register access, the IIC interface operating mode (when the on-chip IIC option is included), selects the TCNT input clock source, and controls the USB. For details of functions not related to the I²C bus interface, see section 3.2.3, Serial Timer Control Register (STCR), and the descriptions of the relevant modules. If a module controlled by STCR is not used, do not write 1 to the corresponding bit.

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

Bit 7—Reserved: This bit must not be set to 1.

Bits 6 and 5—I²C Transfer Select 1 and 0 (IICX1, IICX0): These bits, together with bits CKS2 to CKS0 in ICMR, select the transfer rate in master mode. For details, see section 16.2.4, I²C Bus Mode Register (ICMR).

Bit 4—I²C Master Enable (IICE): Controls CPU access to the I²C bus interface data and control registers (ICCR, ICSR, ICDR/SARX, ICMR/SAR).

Bit 4

IICE	Description	
0	CPU access to I²C bus interface data and control registers is disabled	(Initial value)
1	CPU access to I²C bus interface data and control registers is enabled	

Bit 3—Reserved: This bit must not be set to 1.

Bit 2—USB Enable (USBE): This bit controls CPU access to the USB data register and control register.

Bit 2

USBE	Description	
0	Prohibition of the above register access	(Initial value)
1	Permission of the above register access	

Bits 1 and 0—Internal Clock Source Select 1 and 0 (ICKS1, ICSK0): These bits, together with bits CKS2 to CKS0 in TCR, select the clock input to the timer counters (TCNT). For details, see section 12.2.4, Timer Control Register.

16.2.8 DDC Switch Register (DDCSWR)

Bit	7	6	5	4	3	2	1	0
	SWE	SW	IE	IF	CLR3	CLR2	CLR1	CLR0
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/(W)*1	W*2	W*2	W*2	W*2

Notes: 1. Only 0 can be written, to clear the flag.
2. Always read as 1.

DDCSWR is an 8-bit readable/writable register that controls the IIC channel 0 automatic format switching function.

DDCSWR is initialized to H'0F by a reset and in hardware standby mode.

Bit 7—DDC Mode Switch Enable (SWE): Selects the function for automatically switching IIC channel 0 from formatless mode to the I²C bus format.

Bit 7

SWE	Description
0	Automatic switching of IIC channel 0 from formatless mode to I²C bus format is disabled (Initial value)
1	Automatic switching of IIC channel 0 from formatless mode to I²C bus format is enabled

Bit 6—DDC Mode Switch (SW): Selects either formatless mode or the I²C bus format for IIC channel 0.

Bit 6

SW	Description
0	IIC channel 0 is used with the I²C bus format (Initial value) [Clearing conditions] • When 0 is written by software • When a falling edge is detected on the SCL pin when SWE = 1
1	IIC channel 0 is used in formatless mode [Setting condition] When 1 is written in SW after reading SW = 0

Bit 5—DDC Mode Switch Interrupt Enable Bit (IE): Enables or disables an interrupt request to the CPU when automatic format switching is executed for IIC channel 0.

Bit 5

IE	Description
0	Interrupt when automatic format switching is executed is disabled (Initial value)
1	Interrupt when automatic format switching is executed is enabled

Bit 4—DDC Mode Switch Interrupt Flag (IF): Flag that indicates an interrupt request to the CPU when automatic format switching is executed for IIC channel 0.

Bit 4	
IF	**Description**
0	No interrupt is requested when automatic format switching is executed (Initial value) [Clearing condition] When 0 is written in IF after reading IF = 1
1	An interrupt is requested when automatic format switching is executed [Setting condition] When a falling edge is detected on the SCL pin when SWE = 1

Bits 3 to 0—IIC Clear 3 to 0 (CLR3 to CLR0): These bits control initialization of the internal state of IIC0 and IIC1.

These bits can only be written to; if read they will always return a value of 1.

When a write operation is performed on these bits, a clear signal is generated for the internal latch circuit of the corresponding module(s), and the internal state of the IIC module(s) is initialized.

The write data for these bits is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR.

When clearing is required again, all the bits must be writen to in accordance with the setting.

Bit 3	Bit 2	Bit 1	Bit 0	
CLR3	**CLR2**	**CLR1**	**CLR0**	**Description**
0	0	—	—	Setting prohibited
	1	0	0	Setting prohibited
			1	IIC0 internal latch cleared
		1	0	IIC1 internal latch cleared
			1	IIC0 and IIC1 internal latches cleared
1	—	—	—	Invalid setting

16.2.9 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers, and is used to perform module stop mode control.

When the MSTP4 or MSTP3 bit is set to 1, operation of the corresponding IIC channel is halted at the end of the bus cycle, and a transition is made to module stop mode. For details, see section 21.5, Module Stop Mode.

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

MSTPCRL Bit 4—Module Stop (MSTP4): Specifies IIC channel 0 module stop mode.

MSTPCRL Bit 4	
MSTP4	**Description**
0	IIC channel 0 module stop mode is cleared
1	IIC channel 0 module stop mode is set (Initial value)

MSTPCRL Bit 3—Module Stop (MSTP3): Specifies IIC channel 1 module stop mode.

MSTPCRL Bit 3	
MSTP3	**Description**
0	IIC channel 1 module stop mode is cleared
1	IIC channel 1 module stop mode is set (Initial value)

16.3 Operation

16.3.1 I²C Bus Data Format

The I²C bus interface has serial and I²C bus formats.

The I²C bus formats are addressing formats with an acknowledge bit. These are shown in figures 16.3 (a) and (b). The first frame following a start condition always consists of 8 bits.

IIC channel 0 only is capable of formatless operation, as shown in figure 16.3 (c).

The serial format is a non-addressing format with no acknowledge bit. This is shown in figure 16.4.

Figure 16.5 shows the I²C bus timing.

The symbols used in figures 16.3 to 16.5 are explained in table 16.4.

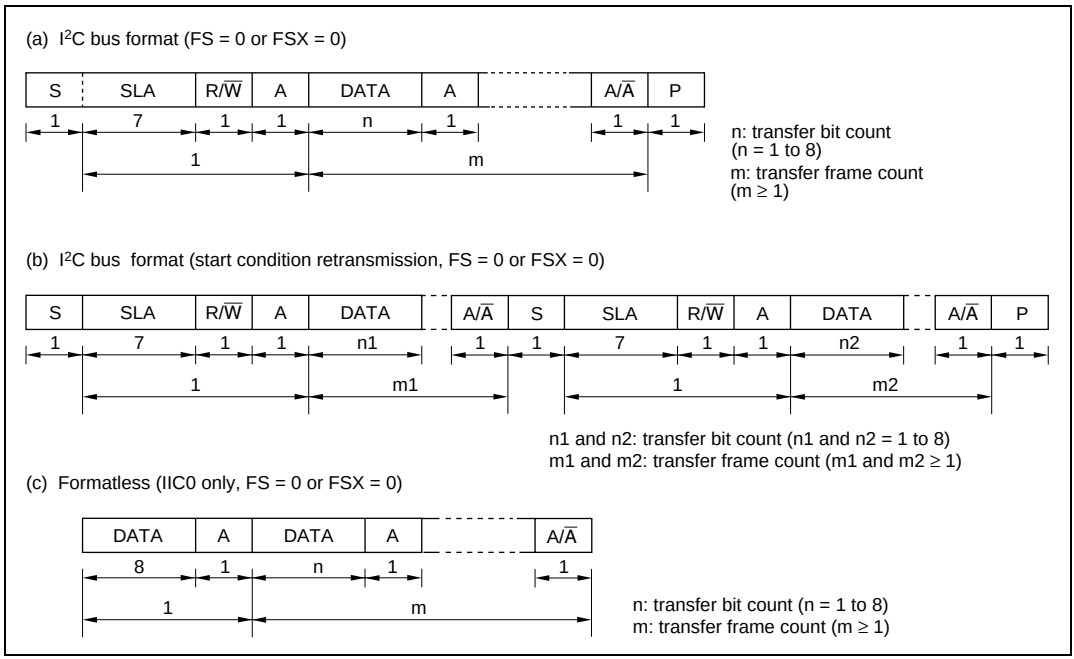


Figure 16.3 I²C Bus Data Formats (I²C Bus Formats)

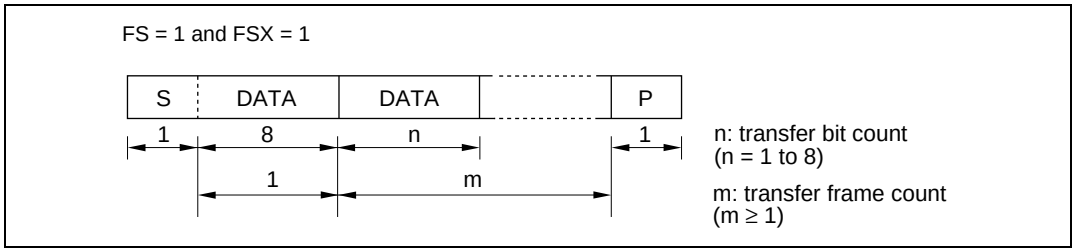


Figure 16.4 I²C Bus Data Format (Serial Format)

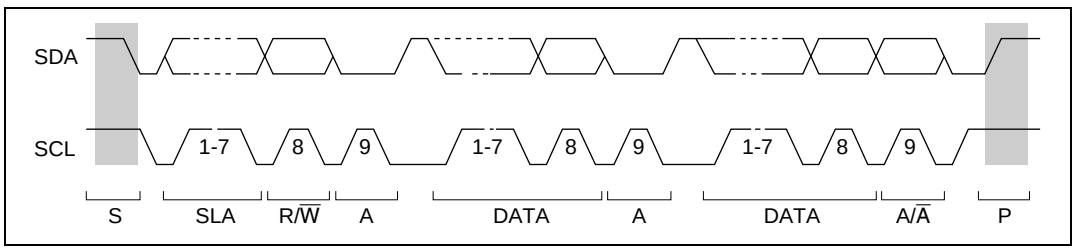


Figure 16.5 I²C Bus Timing

Table 16.4 Description of I²C Bus Data Format Symbols

S	Indicates a start condition. When SCL is high level, the master device changes SDA from high to low level.
SLA	Indicates a slave address. The master device selects the slave device.
R/$\overline{W}$	Indicates the transmit/receive direction. When the value of the R/$\overline{W}$ bit is 1, data is transferred from the slave device to the master device. When it is 0, data is transferred from the master device to the slave device.
A	Indicates an acknowledge response. The receiving device drives SDA low level. (In master transmit mode the slave device, and in master receive mode the master device, returns the acknowledge response.)
DATA	Indicates transmit/receive data. The bit length of the transmit/receive data is set by bits BC2 to BC0 in ICMR. The MLS bit in ICMR is used to select between MSB-first or LSB-first format.
P	Indicates a stop condition. When SCL is high level, the master device changes SDA from low to high level.

16.3.2 Master Transmit Operation

In I²C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal.

The transmission procedure and operations by which data is sequentially transmitted in synchronization with ICDR write operations, are described below.

[1] Set the ICE bit in ICCR to 1. Set bits MLS, WAIT, and CKS2 to CKS0 in ICMR, and bit IICX in STCR, according to the operation mode.

[2] Read the BBSY flag to confirm that the bus is free.

[3] Set the MST and TRS bits to 1 in ICCR to select master transmit mode.

[4] Write 1 to BBSY and 0 to SCP. This switches SDA from high to low when SCL is high, and generates the start condition.

[5] When the start condition is generated, the IRIC and IRTR flags are set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.

[6] Write data to ICDR (slave address + R/$\overline{W}$)

With the I2C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the first frame data following the start condition indicates the 7-bit slave address and transmit/receive direction.

Then clear the IRIC flag to indicate the end of transfer.

Writing to ICDR and clearing of the IRIC flag must be executed continuously, so that no interrupt is inserted.

If a period of time that is equal to transfer one byte has elapsed by the time the IRIC flag is cleared, the end of transfer cannot be identified.

The master device sequentially sends the transmit clock and the data written to ICDR with the timing shown in Figure 16.6. The selected slave device (i.e., the slave device with the matching slave address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge signal.

[7] When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.

[8] Read the ACKB bit to confirm that ACKB is 0.

When the slave device has not returned an acknowledge signal and ACKB remains 1, execute the transmit end processing described in step [12] and perform transmit operation again.

[9] Write the next data to be transmitted in ICDR. To identify the end of data transfer, clear the IRIC flag to 0.

As described in step [6] above, writing to ICDR and clearing of the IRIC flag must be executed continuously so that no interrupt is inserted.

The next frame is transmitted in synchronization with the internal clock.

[10] When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse. After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.

[11] Read the ACKB bit of ICSR. Confirm that the slave device has returned an acknowledge signal and ACKB is 0. When more data is to be transmitted, return to step [9] to execute next transmit operation. If the slave device has not returned an acknowledge signal and ACKB is 1, execute the transmit end processing described in step [12].

[12] Clear the IRIC flag to 0. Write BBSY and CSP of ICCR to 0. By doing so, SDA is changed from low to high while SCL is high and the transmit stop condition is generated.

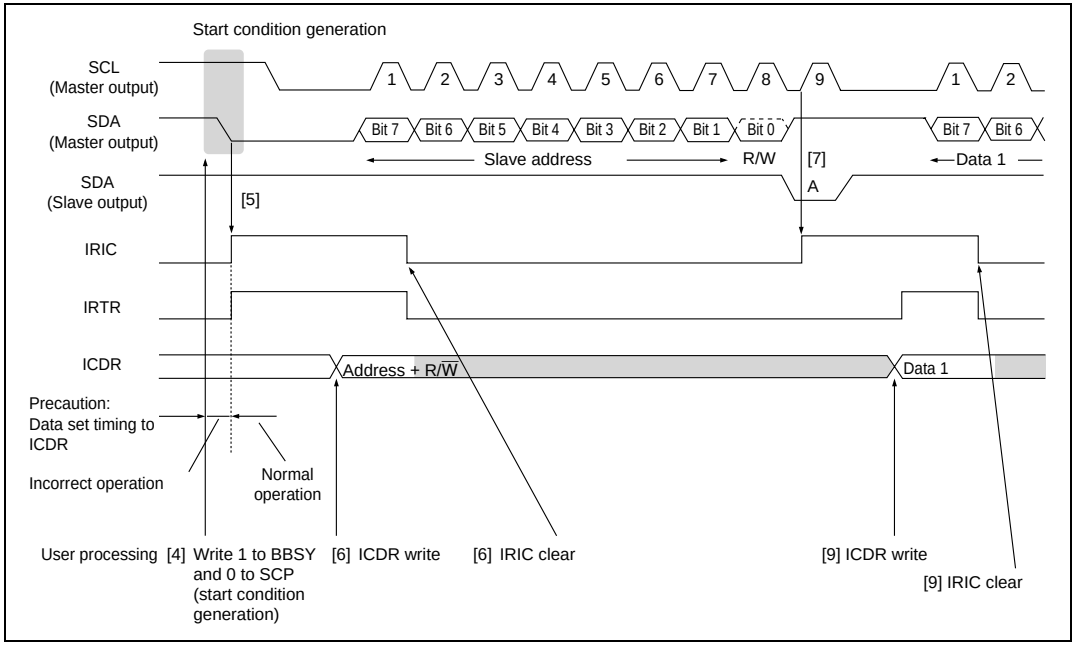


Figure 16.6 Example of Master Transmit Mode Operating Timing (MLS = WAIT = 0)

16.3.3 Master Receive Operation

In master receive mode, the master device outputs the receive clock, receives data, and returns an acknowledge signal. The slave device transmits data.

The transmission procedure and operations by which data is sequentially transmitted in synchronization with ICDR write operations, are described below.

[1] Clear the TRS bit of ICCR to 0 and switch from transmit mode to receive mode. Set the WAIT bit to 1 and clear the ACKB bit of ICSR to 0 (acknowledge data setting).

[2] When ICDR is read (dummy data read), reception is started and the receive clock is output, and data is received, in synchronization with the internal clock. To indicate the wait, clear the IRIC flag to 0.

Reading from ICDR and clearing of the IRIC flag must be executed continuously so that no interrupt is inserted.

If a period of time that is equal to transfer one byte has elapsed by the time the IRIC flag is cleared, the end of transfer cannot be identified.

[3] The IRIC flag is set to 1 at the fall of the 8th clock of a one-frame reception clock. At this point, if the IEIC bit of ICCR is set to 1, an interrupt request is generated to the CPU.

SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag is cleared. If the first frame is the final reception frame, execute the end processing as described in [10].

[4] Clear the IRIC flag to 0 to negate the wait.

The master device outputs the 9th receive clock pulse, sets SDA to low, and returns an acknowledge signal.

[5] When one frame of data has been transmitted, the IRIC and IRTR flags are set to 1 at the rise of the 9th transmit clock pulse.

The master device continues to output the receive clock for the next receive data.

[6] Read the ICDR receive data.

[7] Clear the IRIC flag to indicate the next wait.

From clearing of the IRIC flag to negation of a wait as described in step [4] (and [9]) to clearing of the IRIC flag as described in steps [5], [6], and [7], must be performed within the time taken to transfer one byte.

[8] The IRIC flag is set to 1 at the fall of the 8th one-frame reception clock pulse.

SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag is cleared.

If this frame is the final reception frame, execute the end processing as described in [10].

[9] Clear the IRIC flag to 0 to negate the wait.

The master device outputs the 9th reception clock pulse, sets SDA to low, and returns an acknowledge signal.

By repeating steps [5] to [9] above, more data can be received.

[10] Set the ACKB bit of ICSR to 1 and set the acknowledge data for the final reception.

Set the TRS bit of ICCR to 1 to change receive mode to transmit mode.

[11] Clear the IRIC flag to negate the wait.

[12] When one frame of data has been received, the IRIC flag is set to 1 at the rise of the 9th reception clock pulse.

[13] Clear the WAIT bit of ICMR to 0 to cancel wait mode. Read the ICDR receive data and clear the IRIC flag to 0.

Clear the IRIC flag only when WAIT = 0.

If the stop-condition generation command is executed after clearing the IRIC flag to 0 and then clearing the WAIT bit to 0, the SDA line is fixed low and the stop condition cannot be generated.

[14] Write 0 to BBSY and SCP. This changes SDA from low to high when SCL is high, and generates the stop condition.

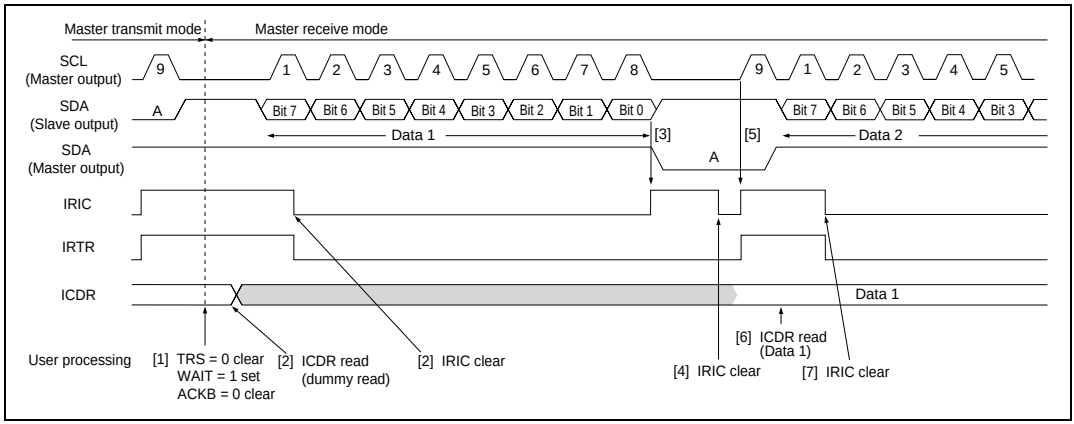


Figure 16.7 (1) Example of Master Receive Mode Operating Timing (MLS = ACKB = 0 and WAIT = 1)

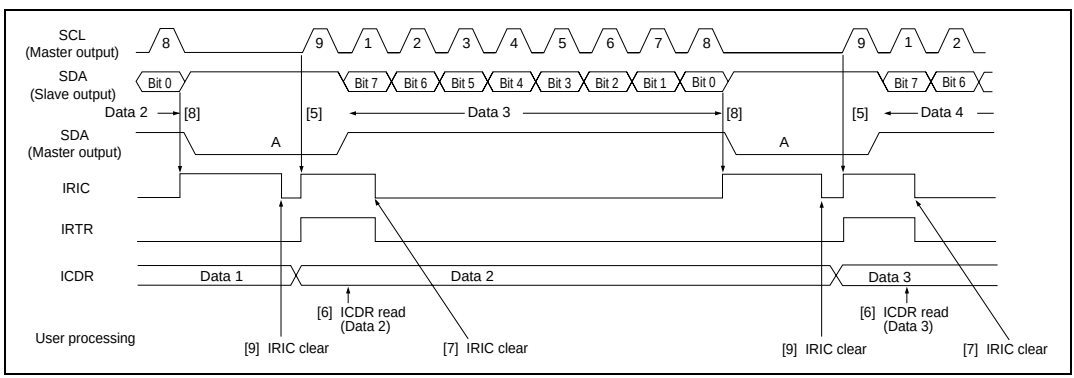


Figure 16.7 (2) Example of Master Receive Mode Operating Timing (MLS = ACKB = 0 and WAIT = 1)

16.3.4 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The reception procedure and operations in slave receive mode are described below.

[1] Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

[2] When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1.

[3] When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$) is 0, the TRS bit in ICCR remains cleared to 0, and slave receive operation is performed.

[4] At the 9th clock pulse of the receive frame, the slave device drives SDA low and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the RDRF internal flag has been cleared to 0, it is set to 1, and the receive operation continues. If the RDRF internal flag has been set to 1, the slave device drives SCL low from the fall of the receive clock until data is read into ICDR.

[5] Read ICDR and clear the IRIC flag in ICCR to 0. The RDRF flag is cleared to 0.

Receive operations can be performed continuously by repeating steps [4] and [5]. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

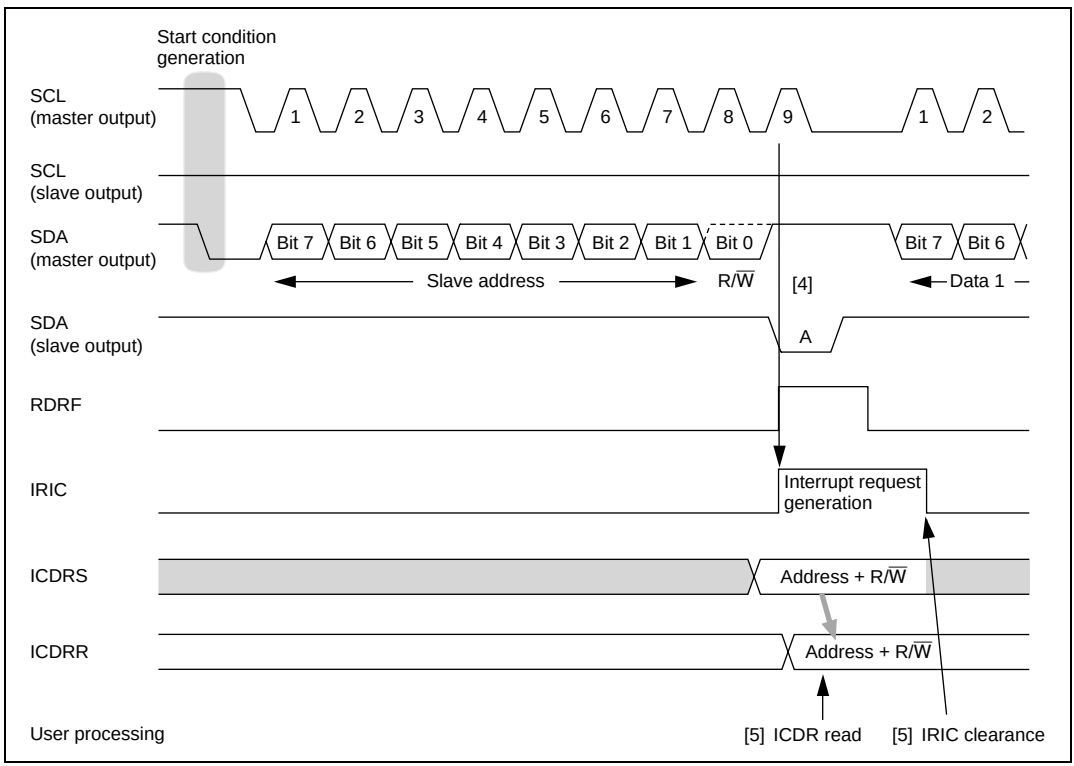


Figure 16.8 Example of Slave Receive Mode Operation Timing (1) (MLS = ACKB = 0)

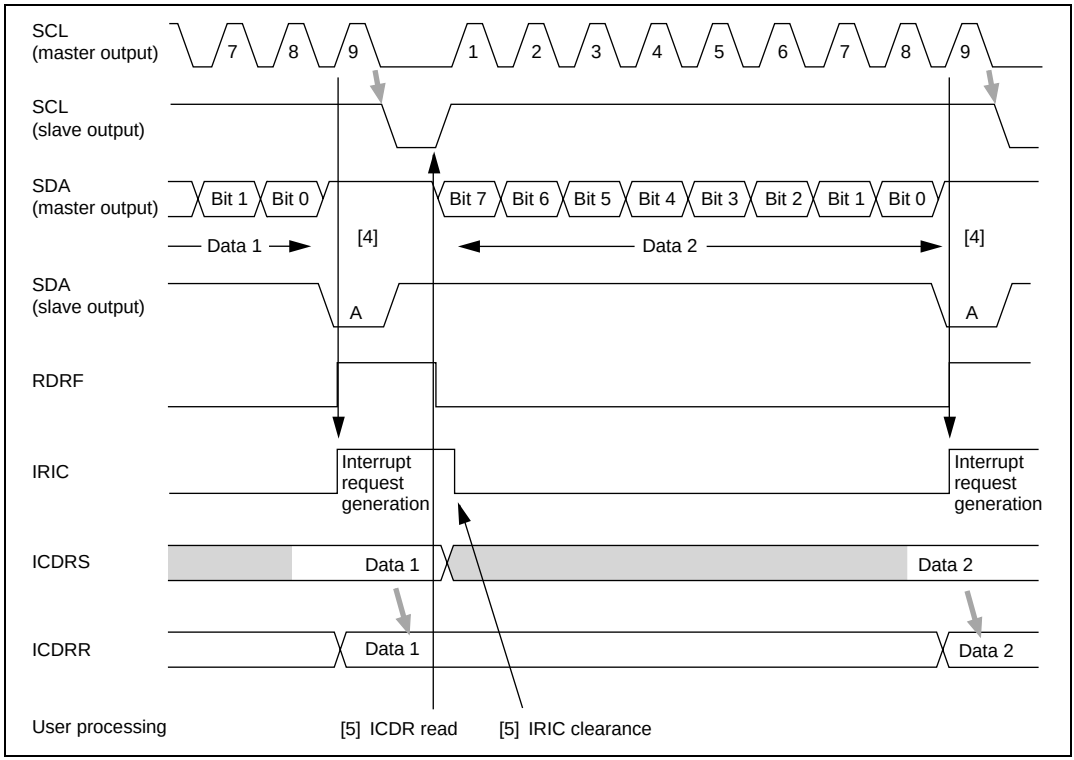


Figure 16.9 Example of Slave Receive Mode Operation Timing (2) (MLS = ACKB = 0)

16.3.5 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

[1] Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

[2] When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the 8th data bit (R/$\overline{W}$) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The TDRF internal flag is set to 1. The slave device drives SCL low from the fall of the transmit clock until ICDR data is written.

[3] After clearing the IRIC flag to 0, write data to ICDR. The TDRE internal flag is cleared to 0. The written data is transferred to ICDRS, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again. After clearing the IRIC flag to 0, write the next data to ICDR. The slave device sequentially sends the data written into ICDR in accordance with the clock output by the master device at the timing shown in figure 16.10.

[4] When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. If the TDRE internal flag has been set to 1, this slave device drives SCL low from the fall of the transmit clock until data is written to ICDR. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed normally. When the TDRE internal flag is 0, the data written into ICDR is transferred to ICDRS, transmission is started, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again.

[5] To continue transmission, clear the IRIC flag to 0, then write the next data to be transmitted into ICDR. The TDRE flag is cleared to 0.

Transmit operations can be performed continuously by repeating steps [4] and [5]. To end transmission, write H'FF to ICDR to release SDA on the slave side. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

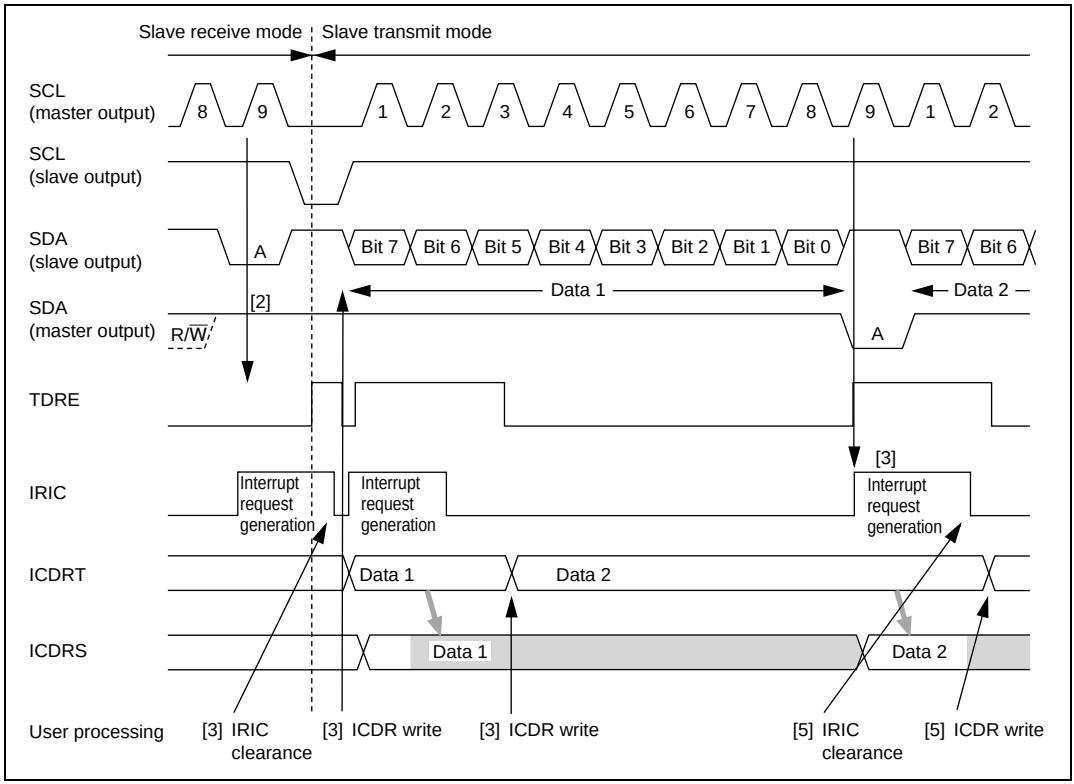


Figure 16.10 Example of Slave Transmit Mode Operation Timing (MLS = 0)

16.3.6 IRIC Setting Timing and SCL Control

The interrupt request flag (IRIC) is set at different times depending on the WAIT bit in ICMR, the FS bit in SAR, and the FSX bit in SARX. If the TDRE or RDRF internal flag is set to 1, SCL is automatically held low after one frame has been transferred; this timing is synchronized with the internal clock. Figure 16.11 shows the IRIC set timing and SCL control.

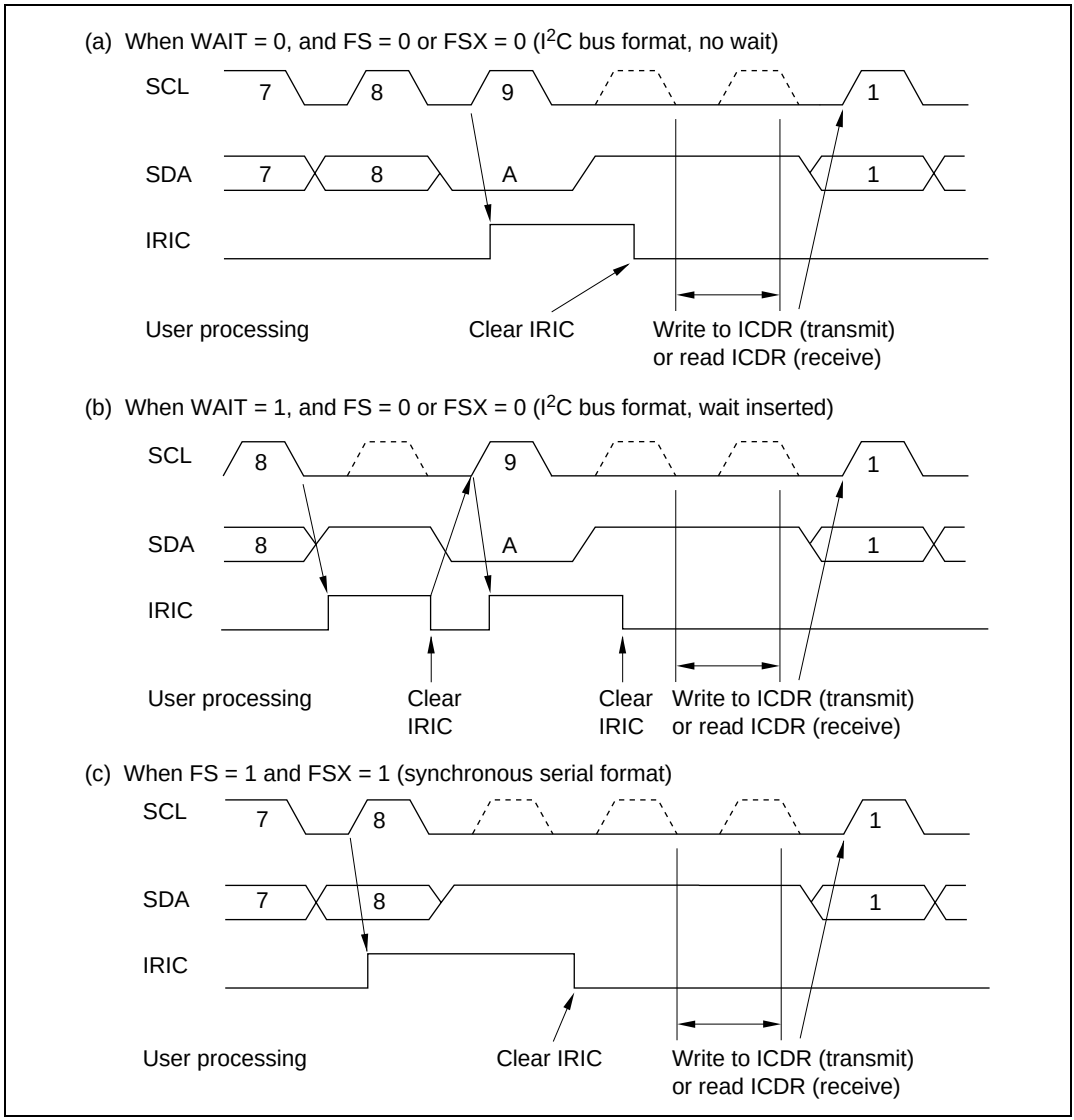


Figure 16.11 IRIC Setting Timing and SCL Control

16.3.7 Automatic Switching from Formatless Mode to I²C Bus Format

Setting the SW bit to 1 in DDCSWR enables formatless mode to be selected as the IIC0 operating mode. Switching from formatless mode to the I²C bus format (slave mode) is performed automatically when a falling edge is detected on the SCL pin.

The following four preconditions are necessary for this operation:

- A common data pin (SDA) for formatless and I²C bus format operation
- Separate clock pins for formatless operation (VSYNCI) and I²C bus format operation (SCL)
- A fixed 1 level for the SCL pin during formatless operation (is not driven to low)
- Settings of bits other than TRS in ICCR that allow I²C bus format operation

Automatic switching is performed from formatless mode to the I²C bus format when the SW bit in DDCSWR is automatically cleared to 0 on detection of a falling edge on the SCL pin. Switching from the I²C bus format to formatless mode is achieved by having software set the SW bit in DDCSWR to 1.

In formatless mode, bits (such as MSL and TRS) that control the I²C bus interface operating mode must not be modified. When switching from the I²C bus format to formatless mode, set the TRS bit to 1 or clear it to 0 according to the transmit data (transmission or reception) in formatless mode, then set the SW bit to 1. After automatic switching from formatless mode to the I²C bus format (slave mode), in order to wait for slave address reception, the TRS bit is automatically cleared to 0.

If a falling edge is detected on the SCL pin during formatless operation, I²C bus interface operation is deferred until a stop condition is detected.

16.3.8 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 16.12 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

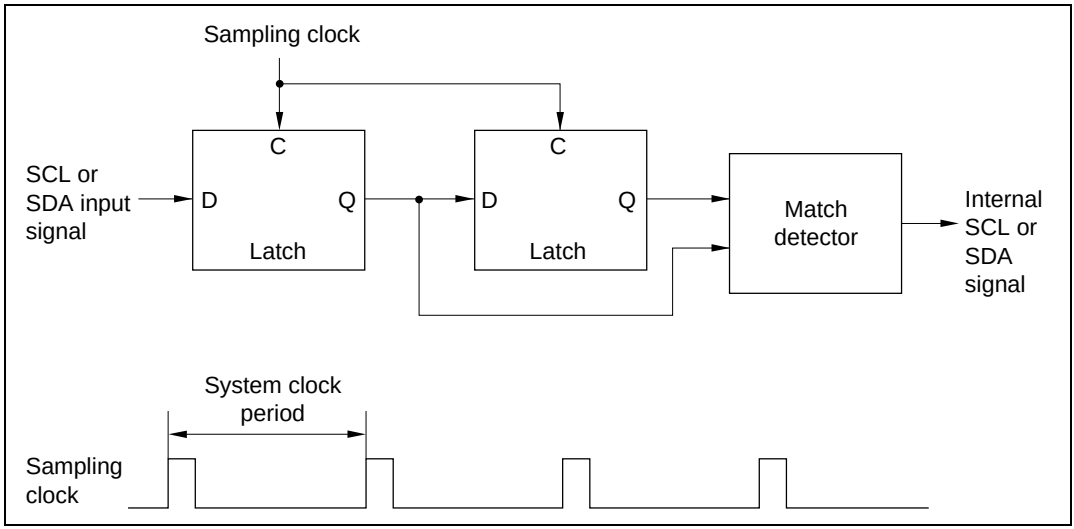


Figure 16.12 Block Diagram of Noise Canceler

16.3.9 Sample Flowcharts

Figures 16.13 to 16.16 show sample flowcharts for using the I²C bus interface in each mode.

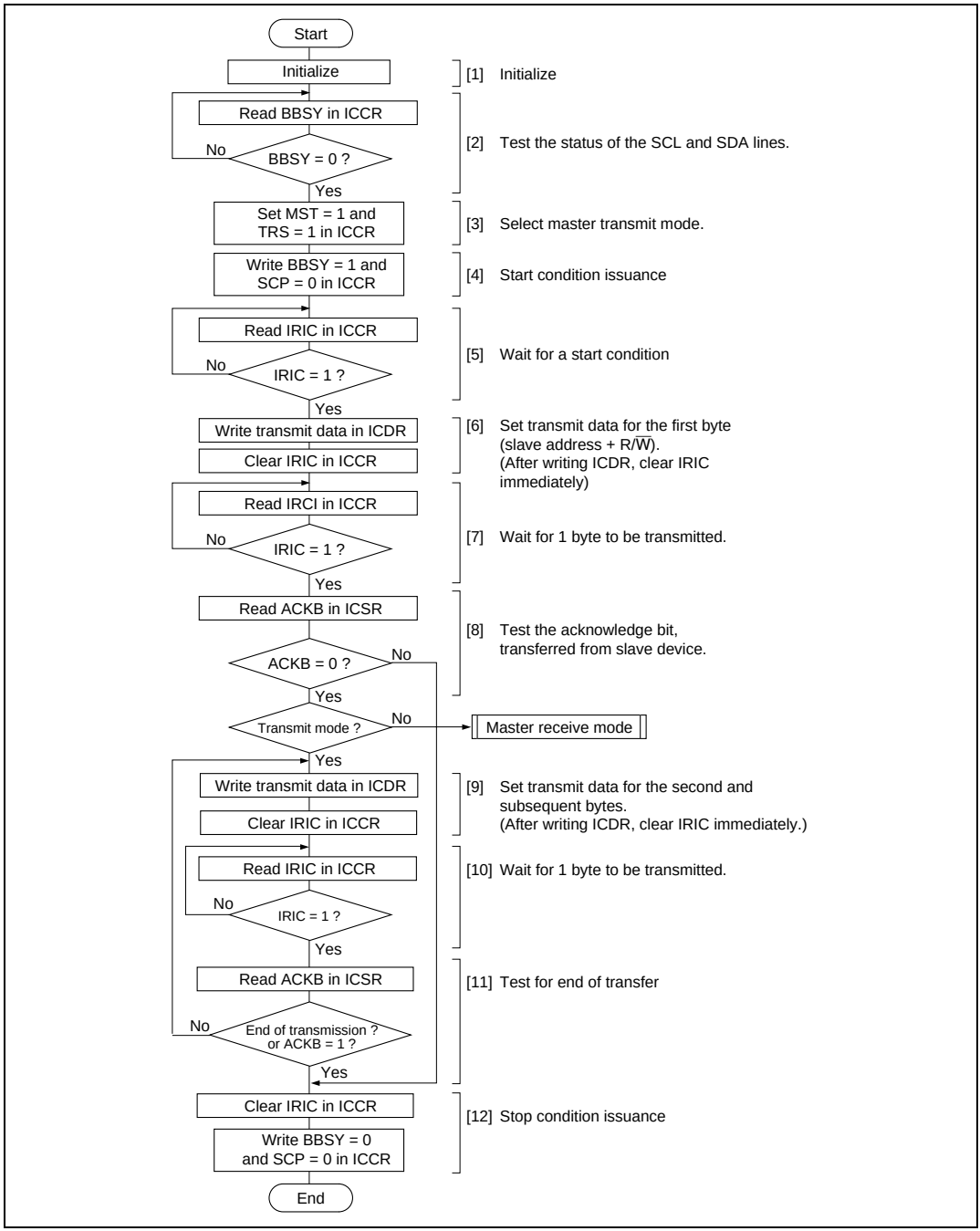


Figure 16.13 Flowchart for Master Transmit Mode (Example)

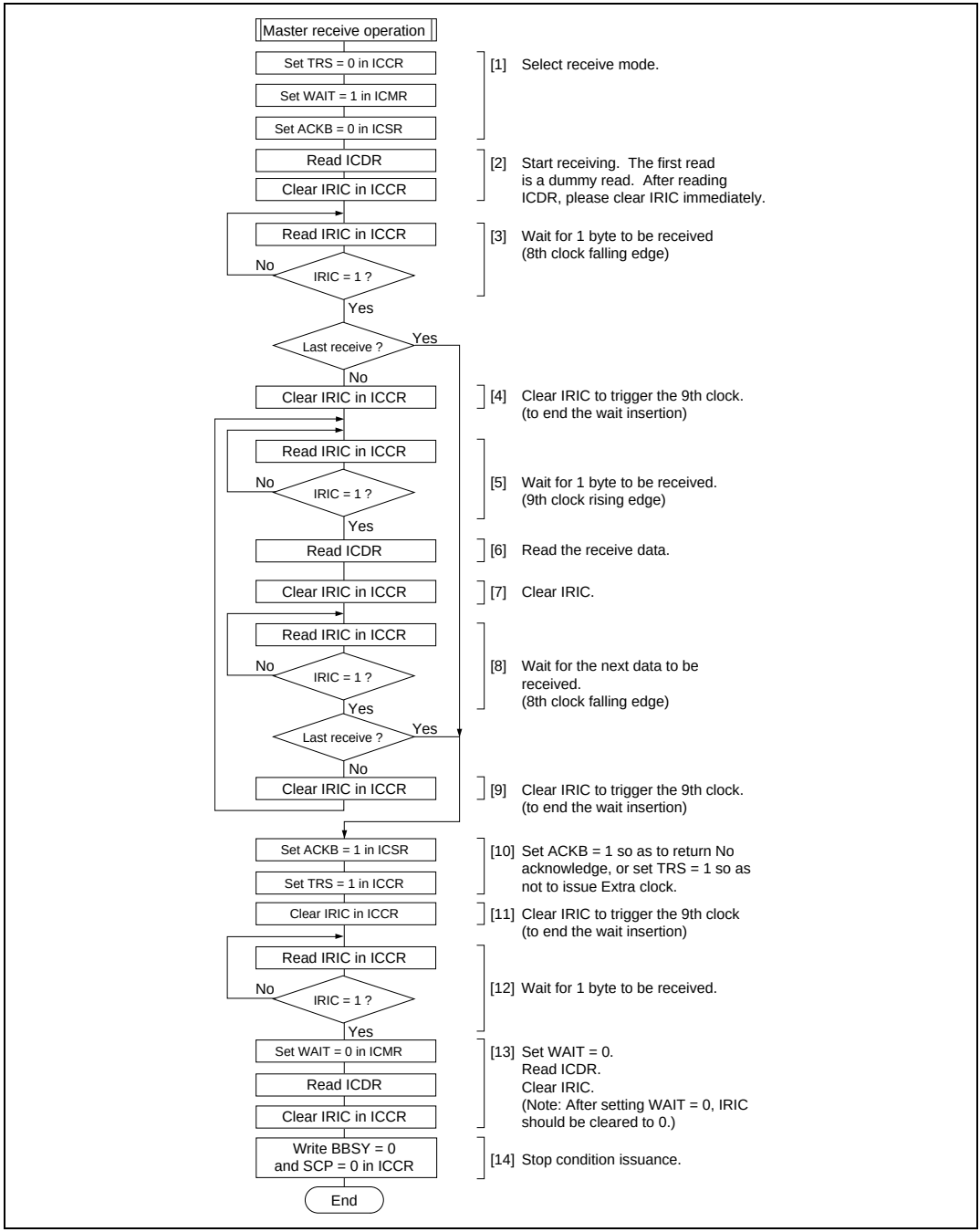


Figure 16.14 Flowchart for Master Receive Mode (Example)

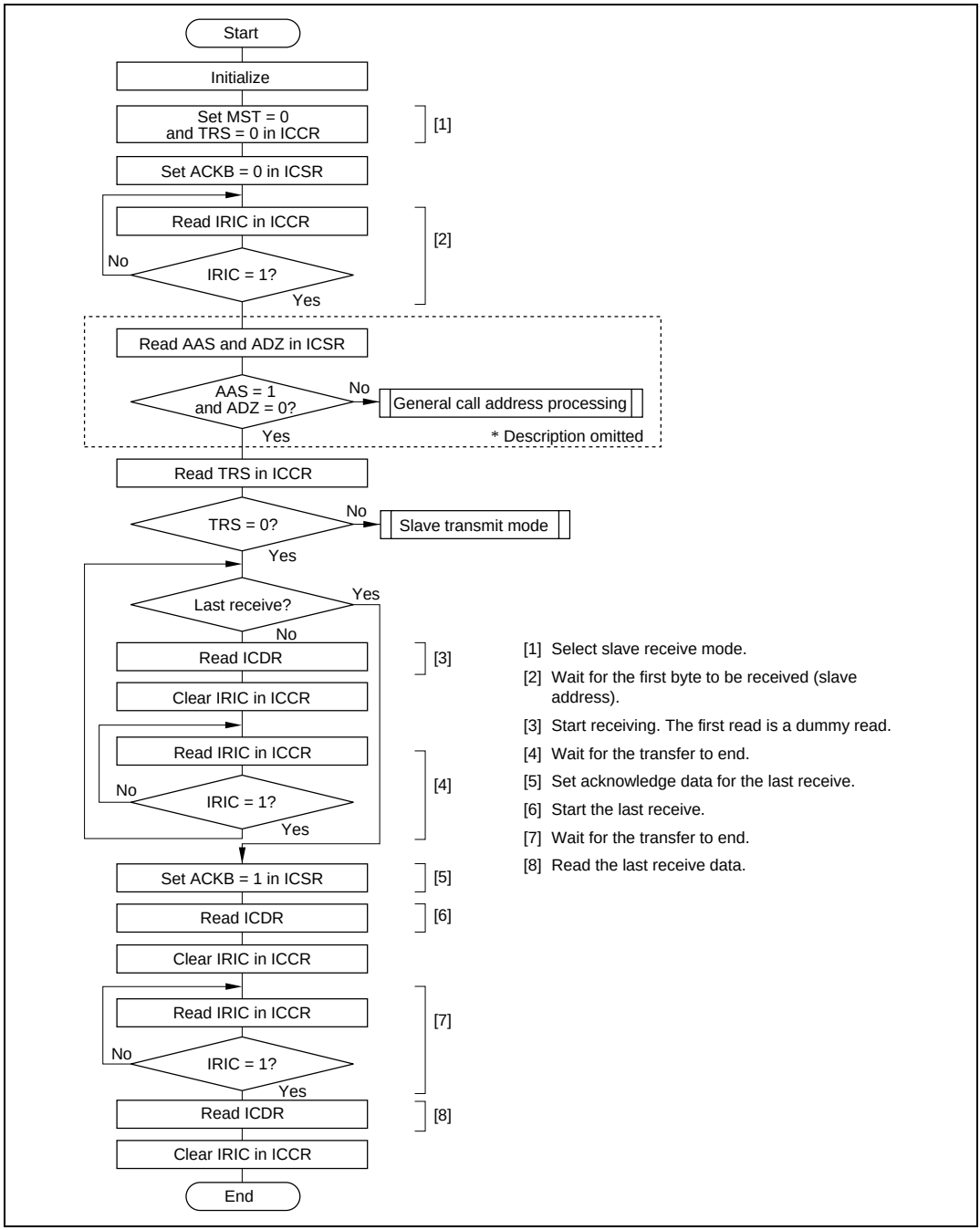


Figure 16.15 Flowchart for Slave Receive Mode (Example)

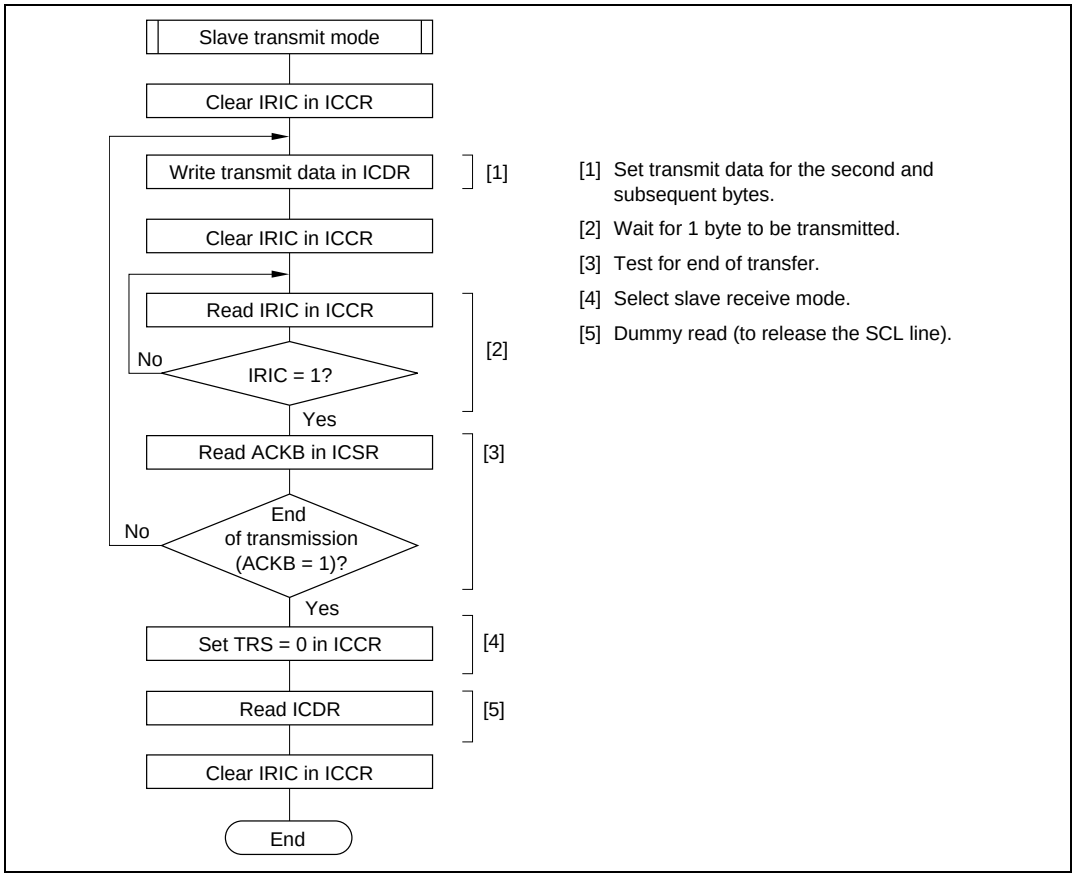


Figure 16.16 Flowchart for Slave Transmit Mode (Example)

16.3.10 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs during communication.

Initialization is executed in accordance with the setting of bits CLR3 to CLR0 in the DDCSWR register or clearing ICE bit. For details the setting of bits CLR3 to CLR0, see section 16.2.8, DDC Switch Register (DDCSWR).

Scope of Initialization: The initialization executed by this function covers the following items:

- TDRE and RDRF internal flags
- Transmit/receive sequencer and internal operating clock counter

- Internal latches for retaining the output state of the SCL and SDA pins (wait, clock, data output, etc.)

The following items are not initialized:

- Actual register values (ICDR, SAR, SARX, ICMR, ICCR, ICSR, DDCSWR, STCR)
- Internal latches used to retain register read information for setting/clearing flags in the ICMR, ICCR, ICSR, and DDCSWR registers
- The value of the ICMR register bit counter (BC2 to BC0)
- Generated interrupt sources (interrupt sources transferred to the interrupt controller)

Notes on Initialization:

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures must be taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures must be taken as necessary.
- When initialization is executed by the DDCSWR register, the write data for bits CLR3 to CLR0 is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR. Similarly, when clearing is required again, all the bits must be written to simultaneously in accordance with the setting.
- If a flag clearing setting is made during transmission/reception, the IIC module will stop transmitting/receiving at that point and the SCL and SDA pins will be released. When transmission/reception is started again, register initialization, etc., must be carried out as necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, but since the stop condition pin waveform is generated according to the state and release timing of the SCL and SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other bits and flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used when initializing the IIC state.

1. Execute initialization of the internal state according to the setting of bits CLR3 to CLR0.
2. Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the BBSY bit to 0, and wait for two transfer rate clock cycles.
3. Re-execute initialization of the internal state according to the setting of bits CLR3 to CLR0.
4. Initialize (re-set) the IIC registers.

16.4 Usage Notes

- In master mode, if an instruction to generate a start condition is immediately followed by an instruction to generate a stop condition, neither condition will be output correctly. To output consecutive start and stop conditions, after issuing the instruction that generates the start condition, read the relevant ports, check that SCL and SDA are both low, then issue the instruction that generates the stop condition. Note that SCL may not yet have gone low when BBSY is cleared to 0.

- Either of the following two conditions will start the next transfer. Pay attention to these conditions when reading or writing to ICDR.
  - Write access to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICDRT to ICDRS)
  - Read access to ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRS to ICDRR)

- Table 16.5 shows the timing of SCL and SDA output in synchronization with the internal clock. Timings on the bus are determined by the rise and fall times of signals affected by the bus load capacitance, group resistance, and parallel resistance.

Table 16.5 I²C Bus Timing (SCL and SDA Output)

Item	Symbol	Output Timing	Unit	Notes
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $256t_{cyc}$	ns	Figure 22.18 (reference)
SCL output high pulse width	t_{SCLHO}	$0.5t_{SCLO}$	ns	
SCL output low pulse width	t_{SCLLO}	$0.5t_{SCLO}$	ns	
SDA output bus free time	t_{BUFO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Start condition output hold time	t_{STAHO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Retransmission start condition output setup time	t_{STASO}	$1t_{SCLO}$	ns	
Stop condition output setup time	t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	t_{SDASO}	$1t_{SCLLO} – 3t_{cyc}$	ns	
Data output setup time (slave)		$1t_{SCLL} – (6t_{cyc}$ or $12t_{cyc}$ *)		
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

Note: * $6t_{cyc}$ when IICX is 0, $12t_{cyc}$ when 1.

- SCL and SDA input is sampled in synchronization with the internal clock. The AC timing therefore depends on the system clock cycle t_{cyc}, as shown in table 22.8 in section 22, Electrical Characteristics. Note that the I²C bus interface AC timing specifications will not be met with a system clock frequency of less than 5 MHz.

- The I²C bus interface specification for the SCL rise time t_{sr} is under 1000 ns (300 ns for high-speed mode). In master mode, the I²C bus interface monitors the SCL line and synchronizes one bit at a time during communication. If t_{sr} (the time for SCL to go from low to V_{IH}) exceeds the time determined by the input clock of the I²C bus interface, the high period of SCL is extended. The SCL rise time is determined by the pull-up resistance and load capacitance of the SCL line. To insure proper operation at the set transfer rate, adjust the pull-up resistance and load capacitance so that the SCL rise time does not exceed the values given in the table below.

Table 16.6 Permissible SCL Rise Time (t_{SR}) Values

		Time Indication						
IICX	t_{cyc} Indication		I²C Bus Specification (Max.)	φ = 5 MHz	φ = 8 MHz	φ = 10 MHz	φ = 16 MHz	φ = 20 MHz
0	$7.5t_{cyc}$	Normal mode	1000 ns	1000 ns	937 ns	750 ns	468 ns	375 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns
1	$17.5t_{cyc}$	Normal mode	1000 ns	1000 ns	1000 ns	1000 ns	1000 ns	875 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns

- The I²C bus interface specifications for the SCL and SDA rise and fall times are under 1000 ns and 300 ns. The I²C bus interface SCL and SDA output timing is prescribed by t_{cyc}, as shown in table 16.6. However, because of the rise and fall times, the I²C bus interface specifications may not be satisfied at the maximum transfer rate. Table 16.7 shows output timing calculations for different operating frequencies, including the worst-case influence of rise and fall times.

  t_{BUFO} fails to meet the I²C bus interface specifications at any frequency. The solution is either (a) to provide coding to secure the necessary interval (approximately 1 µs) between issuance of a stop condition and issuance of a start condition, or (b) to select devices whose input timing permits this output timing for use as slave devices connected to the I²C bus.

  t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I²C bus interface specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be investigated include (a) adjusting the rise and fall times by means of a pull-up resistor and capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecting devices whose input timing permits this output timing for use as slave devices connected to the I²C bus.

Table 16.7 I²C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf})

Item	t_{cyc} Indication		Time Indication (at Maximum Transfer Rate) [ns]						
			t_{Sr}/t_{Sf} Influence (Max.)	I²C Bus Specification (Min.)	ϕ = 5 MHz	ϕ = 8 MHz	ϕ = 10 MHz	ϕ = 16 MHz	ϕ = 20 MHz
t_{SCLHO}	$0.5t_{SCLO}$ $(-t_{Sr})$	Standard mode	–1000	4000	4000	4000	4000	4000	4000
		High-speed mode	–300	600	950	950	950	950	950
t_{SCLLO}	$0.5t_{SCLO}$ $(-t_{Sf})$	Standard mode	–250	4700	4750	4750	4750	4750	4750
		High-speed mode	–250	1300	1000*1	1000*1	1000*1	1000*1	1000*1
t_{BUFO}	$0.5t_{SCLO}$ $-1t_{cyc}$ $(-t_{Sr})$	Standard mode	–1000	4700	3800*1	3875*1	3900*1	3938*1	3950*1
		High-speed mode	–300	1300	750*1	825*1	850*1	888*1	900*1
t_{STAHO}	$0.5t_{SCLO}$ $-1t_{cyc}$ $(-t_{Sf})$	Standard mode	–250	4000	4550	4625	4650	4688	4700
		High-speed mode	–250	600	800	875	900	938	950
t_{STASO}	$1t_{SCLO}$ $(-t_{Sr})$	Standard mode	–1000	4700	9000	9000	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200	2200	2200
t_{STOSO}	$0.5t_{SCLO}$ + $2t_{cyc}$ $(-t_{Sr})$	Standard mode	–1000	4000	4400	4250	4200	4125	4100
		High-speed mode	–300	600	1350	1200	1150	1075	1050
t_{SDASO} (master)	$1t_{SCLLO}$*3 $-3t_{cyc}$ $(-t_{Sr})$	Standard mode	–1000	250	3100	3325	3400	3513	3550
		High-speed mode	–300	100	400	625	700	813	850
t_{SDASO} (slave)	$1t_{SCLL}$*3 $-12t_{cyc}$*2 $(-t_{Sr})$	Standard mode	–1000	250	1300	2200	2500	2950	3100
		High-speed mode	–300	100	–1400*1	–500*1	–200*1	250	400
t_{SDAHO}	$3t_{cyc}$	Standard mode	0	0	600	375	300	188	150
		High-speed mode	0	0	600	375	300	188	150

Notes: 1. Does not meet the I²C bus interface specification. Remedial action such as the following is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the rise and fall times by means of a pull-up resistor and capacitive load; (c) reduce the transfer rate; (d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the IICX bit and bits CKS0 to CKS2. Depending on the frequency it may not be possible to achieve the maximum transfer rate; therefore, whether or not the I²C bus interface specifications are met must be determined in accordance with the actual setting conditions.

2. Value when the IICX bit is set to 1. When the IICX bit is cleared to 0, the value is (t_{SCLL} – $6t_{cyc}$).
3. Calculated using the I²C bus specification values (standard mode: 4700 ns min.; high-speed mode: 1300 ns min.).

- Note on ICDR Read at End of Master Reception

  To halt reception at the end of a receive operation in master receive mode, set the TRS bit to 1 and write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition. After this, receive data can be read by means of an ICDR read, but if data remains in the buffer the ICDRS receive data will not be transferred to ICDR, and so it will not be possible to read the second byte of data.

  If it is necessary to read the second byte of data, issue the stop condition in master receive mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm that the BBSY bit in the ICCR register is cleared to 0, the stop condition has been generated, and the bus has been released, then read the ICDR register with TRS cleared to 0.

  Note that if the receive data (ICDR data) is read in the interval between execution of the instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR) and the actual generation of the stop condition, the clock may not be output correctly in subsequent master transmission.

  Clearing of the MST bit after completion of master transmission/reception, or other modifications of IIC control bits to change the transmit/receive operating mode or settings, must be carried out during interval (a) in figure 16.17 (after confirming that the BBSY bit has been cleared to 0 in the ICCR register).

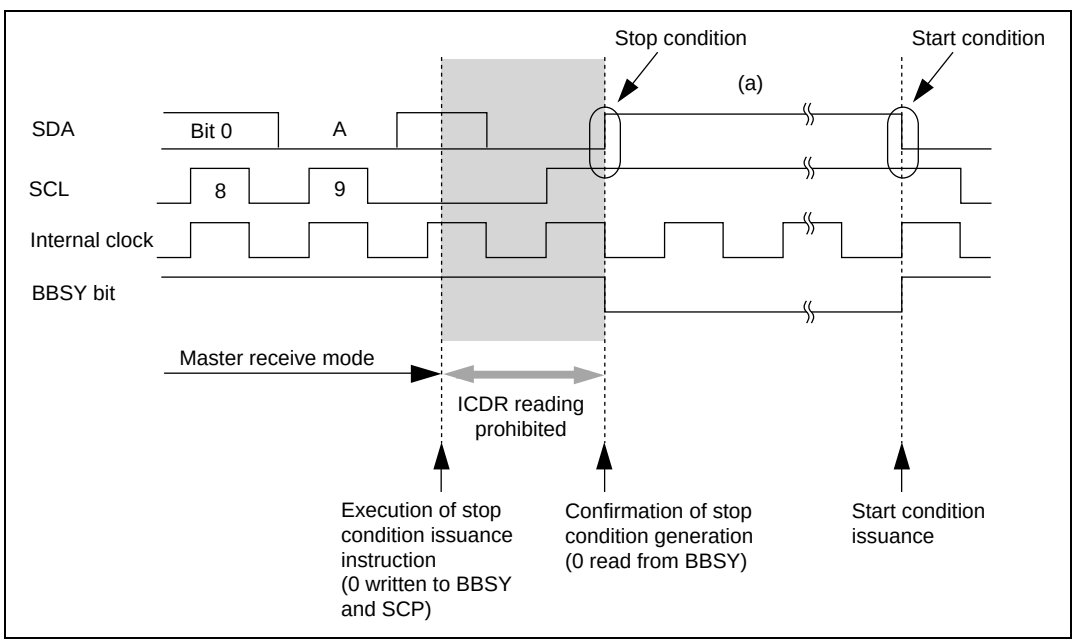


Figure 16.17 Points for Attention Concerning Reading of Master Receive Data

- Notes on Start Condition Issuance for Retransmission

  Figure 16.18 shows the timing of start condition issuance for retransmission, and the timing for subsequently writing data to ICDR, together with the corresponding flowchart. After start condition issuance is done and determined the start condition, write the transmit data to ICDR, as shown below.

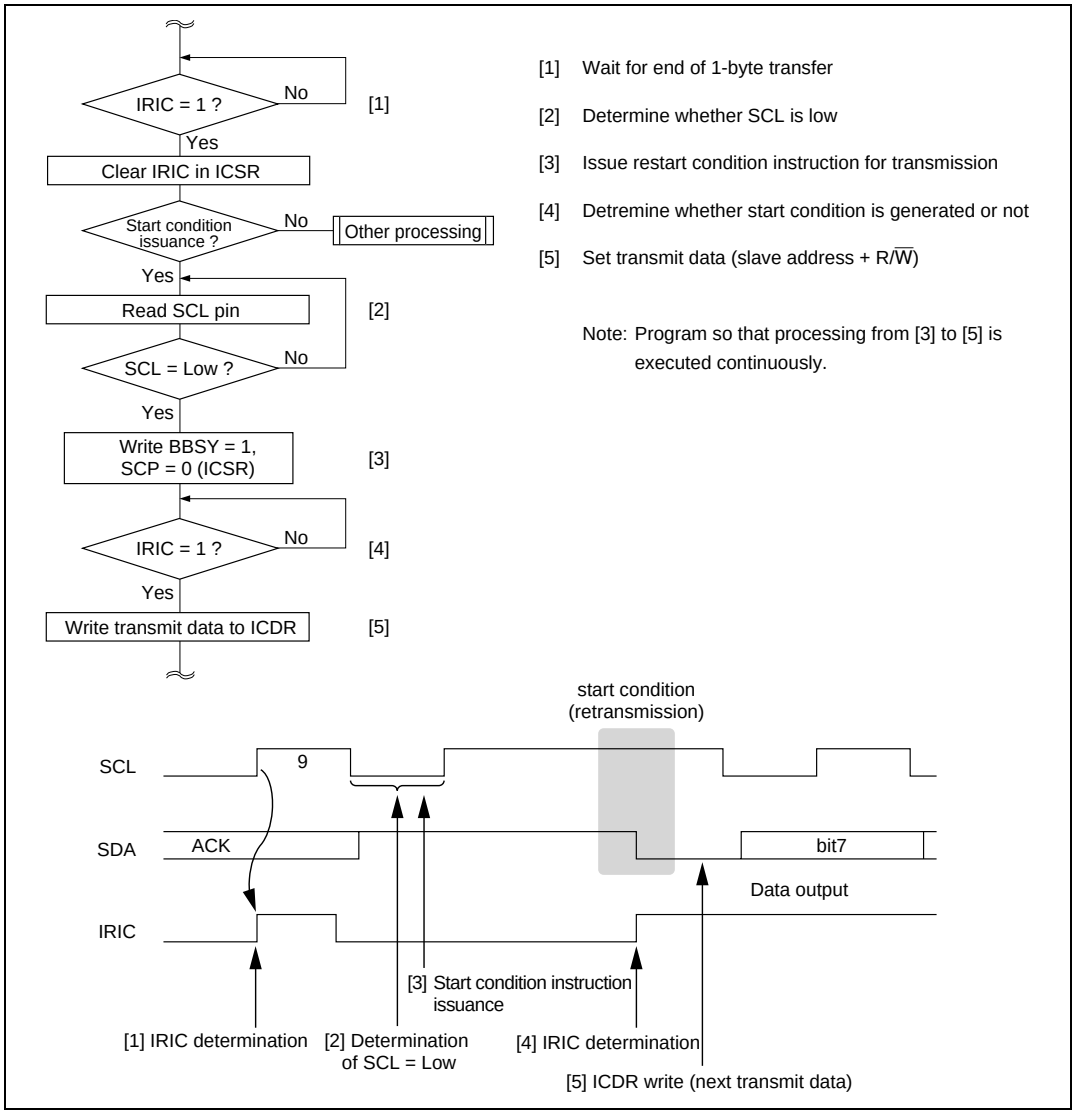


Figure 16.18 Flowchart and Timing of Start Condition Instruction Issuance for Retransmission

- Notes on I²C Bus Interface Stop Condition Instruction Issuance

  If the rise time of the 9th SCL clock exceeds the specification because the bus load capacitance is large, or if there is a slave device of the type that drives SCL low to effect a wait, after rising of the 9th SCL clock, issue the stop condition after reading SCL and determining it to below, as shown below.

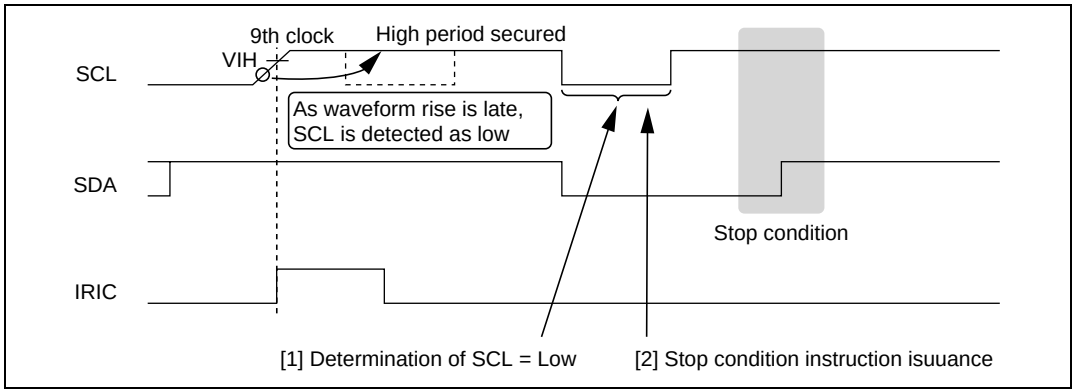


Figure 16.19 Timing of Stop Condition Issuance

Section 17 A/D Converter

17.1 Overview

The H8/3577 Group and H8/3567 Group have an on-chip 10-bit successive-approximations A/D converter that allows up to eight analog input channels to be selected.

The H8/3577 Group has eight analog input channels, and the H8/3567 Group has four.

17.1.1 Features

A/D converter features are listed below.

- 10-bit resolution (analog input)
- Input channels
  - 8 channels (H8/3577 Group)
  - 4 channels (H8/3567 Group)
- Settable analog conversion voltage range
  - The analog conversion voltage range is set using the analog power supply voltage pin (AVcc) as the analog reference voltage
- High-speed conversion
  - Minimum conversion time: 6.7 µs per channel (at 20 MHz operation)
- Choice of single mode or scan mode
  - Single mode: Single-channel A/D conversion
  - Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  - Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  - Choice of software or timer conversion start trigger (8-bit timer), or $\overline{\text{ADTRG}}$ pin
- A/D conversion end interrupt generation
  - An A/D conversion end interrupt (ADI) request can be generated at the end of A/D conversion

17.1.2 Block Diagram

Figure 17.1 shows a block diagram of the A/D converter.

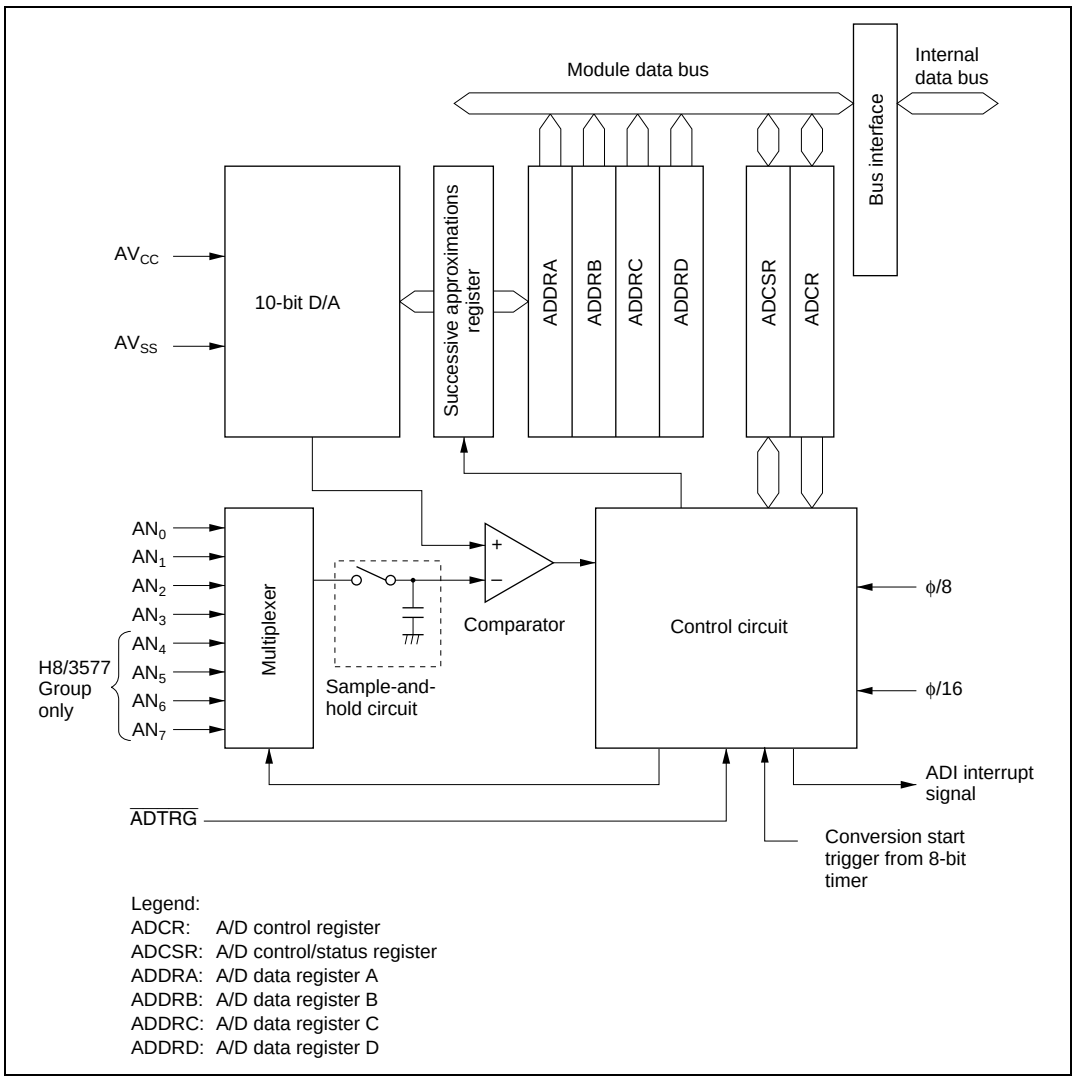


Figure 17.1 Block Diagram of A/D Converter

17.1.3 Pin Configuration

Table 17.1 summarizes the input pins used by the A/D converter.

The AV_{CC} and AV_{SS} pins are the power supply pins for the analog block in the A/D converter.

Table 17.1 A/D Converter Pins

Pin Name	Symbol	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog block power supply
Analog ground pin	AV_{SS}	Input	Analog block ground and A/D conversion reference voltage
Analog input pin 0	AN_0	Input	Analog input channel 0
Analog input pin 1	AN_1	Input	Analog input channel 1
Analog input pin 2	AN_2	Input	Analog input channel 2
Analog input pin 3	AN_3	Input	Analog input channel 3
Analog input pin 4	AN_4	Input	Analog input channel 4 (H8/3577 Group only)
Analog input pin 5	AN_5	Input	Analog input channel 5 (H8/3577 Group only)
Analog input pin 6	AN_6	Input	Analog input channel 6 (H8/3577 Group only)
Analog input pin 7	AN_7	Input	Analog input channel 7 (H8/3577 Group only)
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

17.1.4 Register Configuration

Table 17.2 summarizes the registers of the A/D converter.

Table 17.2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address
A/D data register AH	ADDRAH	R	H'00	H'FFE0
A/D data register AL	ADDRAL	R	H'00	H'FFE1
A/D data register BH	ADDRBH	R	H'00	H'FFE2
A/D data register BL	ADDRBL	R	H'00	H'FFE3
A/D data register CH	ADDRCH	R	H'00	H'FFE4
A/D data register CL	ADDRCL	R	H'00	H'FFE5
A/D data register DH	ADDRDH	R	H'00	H'FFE6
A/D data register DL	ADDRDL	R	H'00	H'FFE7
A/D control/status register	ADCSR	R/(W)*	H'00	H'FFE8
A/D control register	ADCR	R/W	H'3F	H'FFE9
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

Note: * Only 0 can be written in bit 7, to clear the flag.

17.2 Register Descriptions

17.2.1 A/D Data Registers A to D (ADDRA to ADDRD)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

There are four 16-bit read-only ADDR registers, ADDRA to ADDRD, used to store the results of A/D conversion.

The 10-bit data resulting from A/D conversion is transferred to the ADDR register for the selected channel and stored there. The upper 8 bits of the converted data are transferred to the upper byte (bits 15 to 8) of ADDR, and the lower 2 bits are transferred to the lower byte (bits 7 and 6) and stored. Bits 5 to 0 are always read as 0.

The correspondence between the analog input channels and ADDR registers is shown in table 17.3.

The ADDR registers can always be read by the CPU. The upper byte can be read directly, but for the lower byte, data transfer is performed via a temporary register (TEMP). For details, see section 17.3, Interface to Bus Master.

The ADDR registers are initialized to H'0000 by a reset, and in standby mode, and module stop mode.

Table 17.3 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
AN_0	AN_4	ADDRA
AN_1	AN_5	ADDRB
AN_2	AN_6	ADDRC
AN_3	AN_7	ADDRD

17.2.2 A/D Control/Status Register (ADCSR)

Bit	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written in bit 7, to clear the flag.

ADCSR is an 8-bit readable/writable register that controls A/D conversion operations.

ADCSR is initialized to H'00 by a reset, and in standby mode, and module stop mode.

Bit 7—A/D End Flag (ADF): Status flag that indicates the end of A/D conversion.

Bit 7

ADF	Description	
0	[Clearing condition]	(Initial value)
	When 0 is written in the ADF flag after reading ADF = 1	
1	[Setting conditions]	
	• Single mode: When A/D conversion ends	
	• Scan mode: When A/D conversion ends on all specified channels	

Bit 6—A/D Interrupt Enable (ADIE): Selects enabling or disabling of interrupt (ADI) requests at the end of A/D conversion.

Bit 6

ADIE	Description	
0	A/D conversion end interrupt (ADI) request is disabled	(Initial value)
1	A/D conversion end interrupt (ADI) request is enabled	

Bit 5—A/D Start (ADST): Selects starting or stopping of A/D conversion. Holds a value of 1 during A/D conversion.

The ADST bit can be set to 1 by software, a timer conversion start trigger, or the A/D external trigger input pin ($\overline{\text{ADTRG}}$).

Bit 5

ADST	Description	
0	A/D conversion stopped	(Initial value)
1	Single mode: A/D conversion is started. Cleared to 0 automatically when conversion on the specified channel ends	
	Scan mode: A/D conversion is started. Conversion continues sequentially on the selected channels until ADST is cleared to 0 by software, a reset, or a transition to standby mode or module stop mode	

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode as the A/D conversion operating mode. See section 17.4, Operation, for single mode and scan mode operation. Only set the SCAN bit while conversion is stopped.

Bit 4

SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

Bit 3—Clock Select (CKS): Sets the A/D conversion time. Only change the conversion time while ADST = 0.

Bit 3

CKS	Description	
0	Conversion time = 266 states (max.)	(Initial value)
1	Conversion time = 134 states (max.)	

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): Together with the SCAN bit, these bits select the analog input channel(s).

Only set the input channel while conversion is stopped.

	Group Selection	Channel Selection		Description	
	CH2	CH1	CH0	Single Mode	Scan Mode
H8/3577 Group and H8/3567 Group	0	0	0	AN_0 (Initial value)	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
H8/3577 Group only	1	0	0	AN_4	AN_4
			1	AN_5	AN_4, AN_5
		1	0	AN_6	AN_4, AN_5, AN_6
			1	AN_7	AN_4 to AN_7

17.2.3 A/D Control Register (ADCR)

Bit	7	6	5	4	3	2	1	0
	TRGS1	TRGS0	—	—	—	—	—	—
Initial value	0	0	1	1	1	1	1	1
Read/Write	R/W	R/W	—	—	—	—	—	—

ADCR is an 8-bit readable/writable register that enables or disables external triggering of A/D conversion operations.

ADCR is initialized to H'3F by a reset, and in standby mode, and module stop mode.

Bits 7 and 6—Timer Trigger Select 1 and 0 (TRGS1, TRGS0): These bits select enabling or disabling of the start of A/D conversion by a trigger signal. Only set bits TRGS1 and TRGS0 while conversion is stopped.

Bit 7	Bit 6		
TRGS1	**TRGS0**	**Description**	
0	0	Start of A/D conversion by external trigger is disabled	(Initial value)
	1	Start of A/D conversion by external trigger is disabled	
1	0	Start of A/D conversion by external trigger (8-bit timer) is enabled	
	1	Start of A/D conversion by external trigger pin is enabled	

Bits 5 to 0—Reserved: These bits cannot be modified and are always read as 1.

17.2.4 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

When the MSTP9 bit in MSTPCR is set to 1, A/D converter operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 21.5, Module Stop Mode.

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

MSTPCRH Bit 1—Module Stop (MSTP9): Specifies the A/D converter module stop mode.

MSTPCRH Bit 1		
MSTP9	**Description**	
0	A/D converter module stop mode is cleared	
1	A/D converter module stop mode is set	(Initial value)

17.3 Interface to Bus Master

ADDRA to ADDRD are 16-bit registers, but the data bus to the bus master is only 8 bits wide. Therefore, in accesses by the bus master, the upper byte is accessed directly, but the lower byte is accessed via a temporary register (TEMP).

A data read from ADDR is performed as follows. When the upper byte is read, the upper byte value is transferred to the CPU and the lower byte value is transferred to TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading ADDR, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 17.2 shows the data flow for ADDR access.

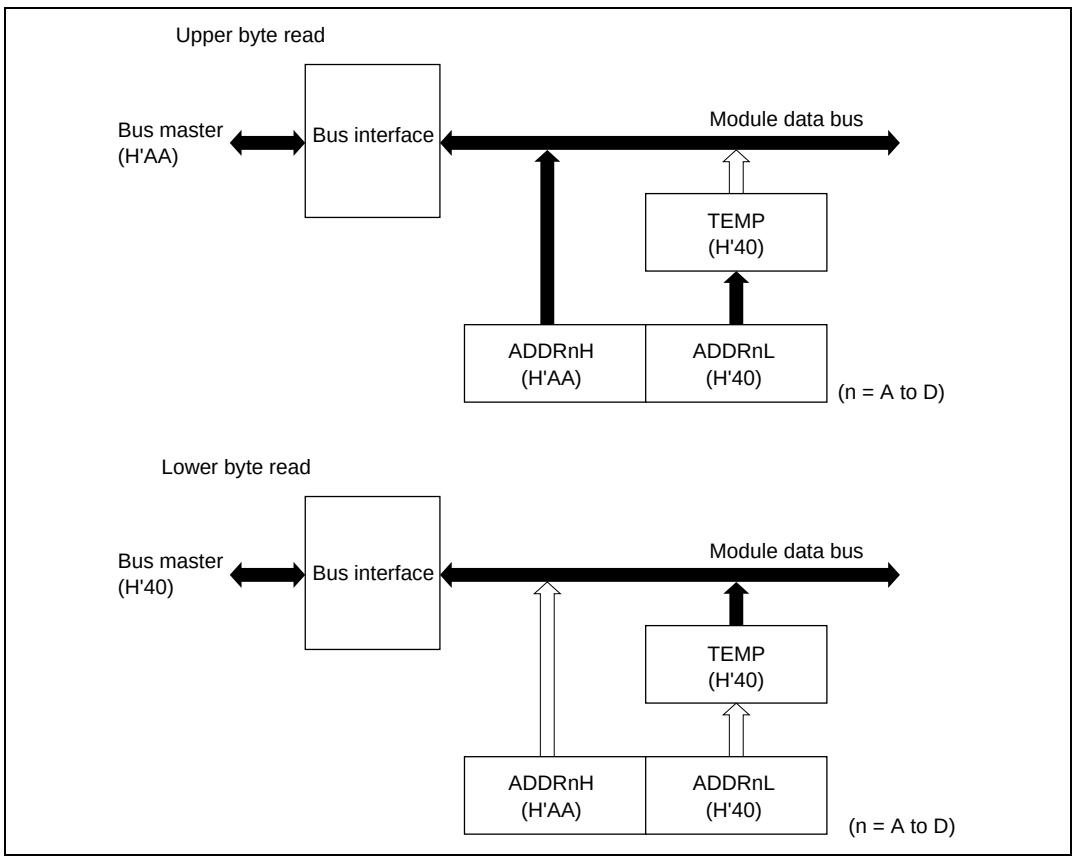


Figure 17.2 ADDR Access Operation (Reading H'AA40)

17.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

17.4.1 Single Mode (SCAN = 0)

Single mode is selected when A/D conversion is to be performed on a single channel only. A/D conversion is started when the ADST bit is set to 1 by software, or by external trigger input. The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends.

On completion of conversion, the ADF flag is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The ADF flag is cleared by writing 0 after reading ADCSR.

When the operating mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the operating mode or input channel is changed.

Typical operations when channel 1 (AN_1) is selected in single mode are described next. Figure 17.3 shows a timing diagram for this example.

1. Single mode is selected (SCAN = 0), input channel AN_1 is selected (CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred to ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 to the ADF flag.
6. The routine reads and processes the conversion result (ADDRB).
7. Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is set to 1, A/D conversion starts again and steps 2 to 7 are repeated.

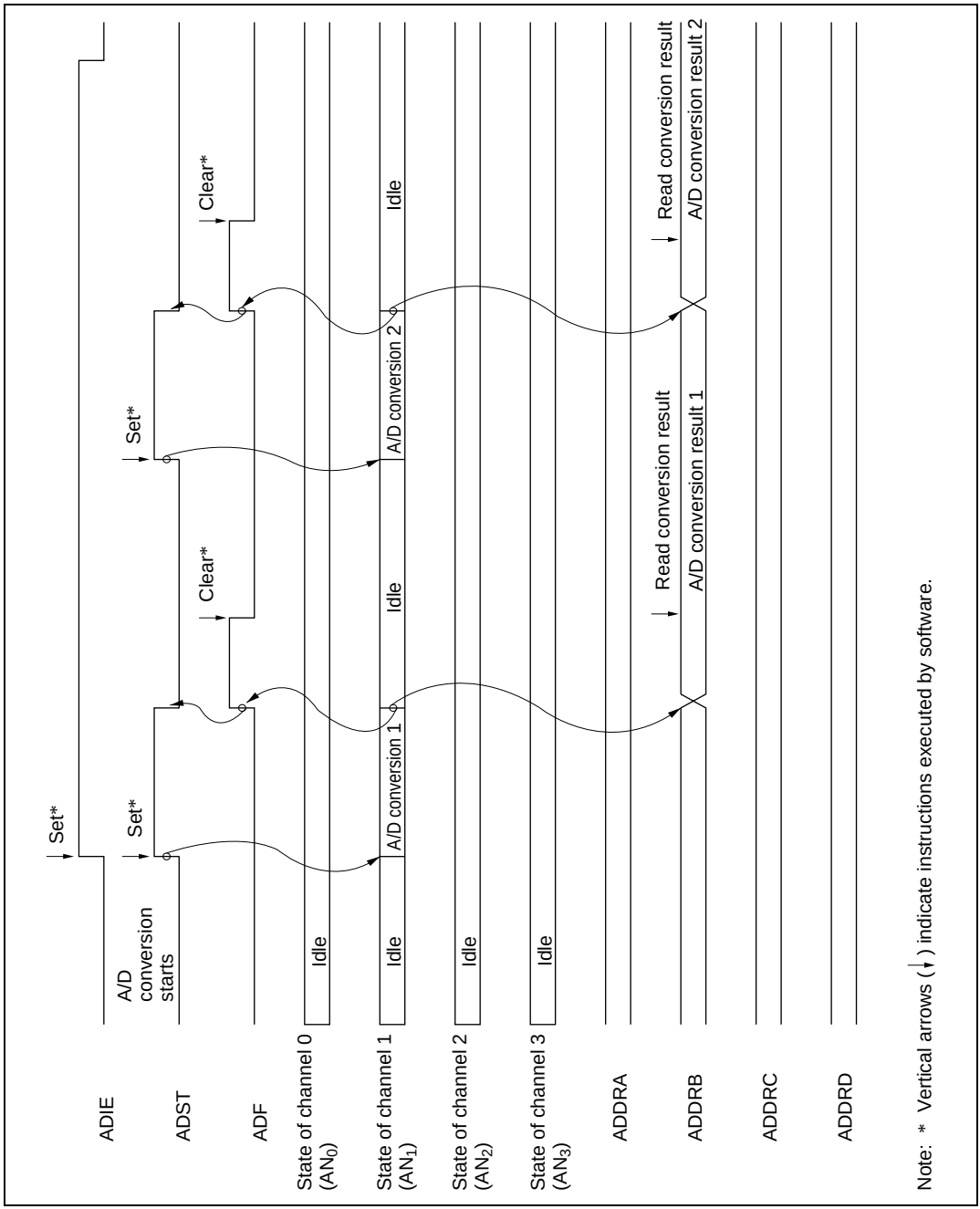


Figure 17.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

17.4.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by software, or by timer or external trigger input, A/D conversion starts on the first channel in the group (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 ADDR registers corresponding to the channels.

When the operating mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the operating mode or input channel is changed.

Typical operations when three channels (AN_0 to AN_2) are selected in scan mode are described next. Figure 17.4 shows a timing diagram for this example.

1. Scan mode is selected (SCAN = 1), scan group 0 is selected (CH2 = 0), analog input channels 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 to 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 the 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 at this time, an ADI interrupt is requested after A/D conversion ends.

5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN_0).

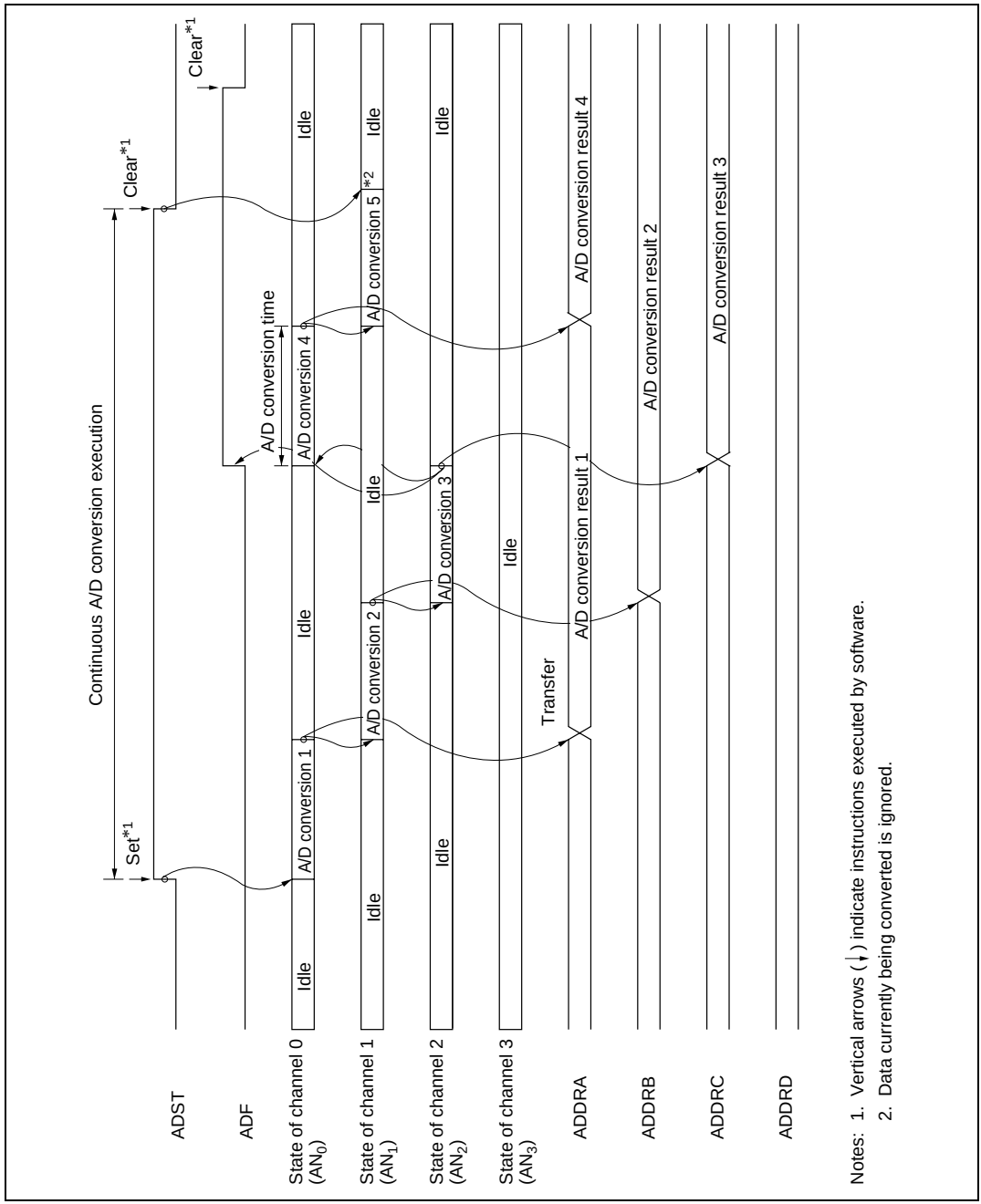


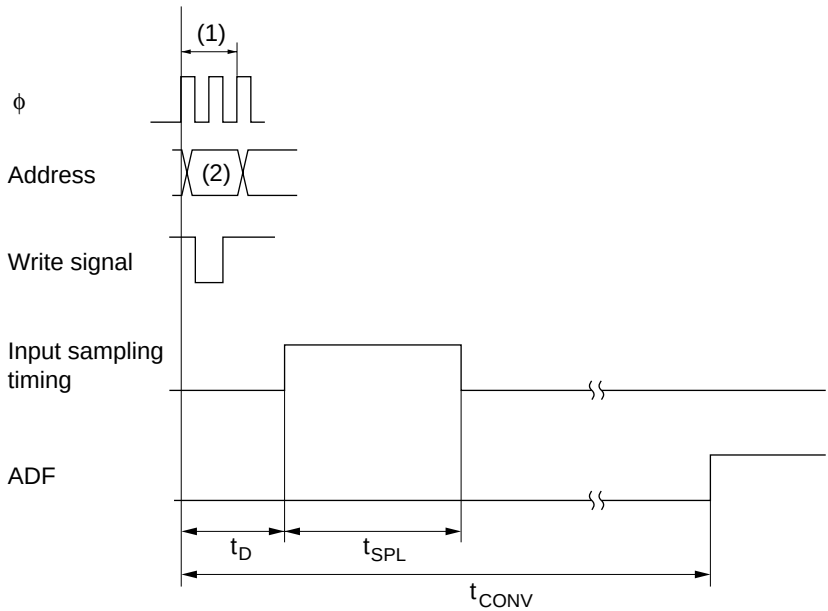
Figure 17.4 Example of A/D Converter Operation (Scan Mode, Channels AN_0 to AN_2 Selected)

17.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit is set to 1, then starts conversion. Figure 17.5 shows the A/D conversion timing. Table 17.4 indicates the A/D conversion time.

As indicated in figure 17.5, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 17.4.

In scan mode, the values given in table 17.4 apply to the first conversion time. In the second and subsequent conversions the conversion time is fixed at 256 states when CKS = 0 or 128 states when CKS = 1.



Legend:
(1): ADCSR write cycle
(2): ADCSR address
t_D: A/D conversion start delay
t_{SPL}: Input sampling time
t_{CONV}: A/D conversion time

Figure 17.5 A/D Conversion Timing

Table 17.4 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1		
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay	t_D	10	—	17	6	—	9
Input sampling time	t_{SPL}	—	63	—	—	31	—
A/D conversion time	t_{CONV}	259	—	266	131	—	134

Note: Values in the table are the number of states.

17.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to 11 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as when the ADST bit is set to 1 by software. Figure 17.6 shows the timing.

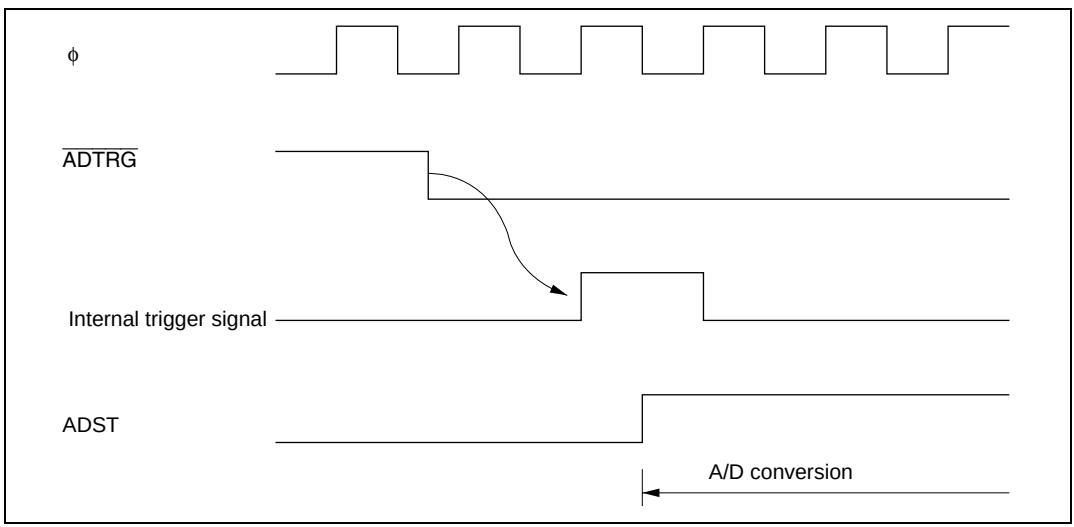


Figure 17.6 External Trigger Input Timing

17.5 Interrupts

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

17.6 Usage Notes

The following points should be noted when using the A/D converter.

Setting Range of Analog Power Supply and Other Pins

1. Analog input voltage range
   The voltage applied to the ANn analog input pins during A/D conversion should be in the range $AV_{SS} \leq ANn \leq AV_{CC}$ (n = 0 to 7).

2. Relation between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}
   As the relationship between AV_{CC}, AV_{SS} and V_{CC}, V_{SS}, set $AV_{SS} = V_{SS}$. If the A/D converter is not used, the AV_{CC} and AV_{SS} pins must on no account be left open.

If conditions 1 and 2 above are not met, the reliability of the device may be adversely affected.

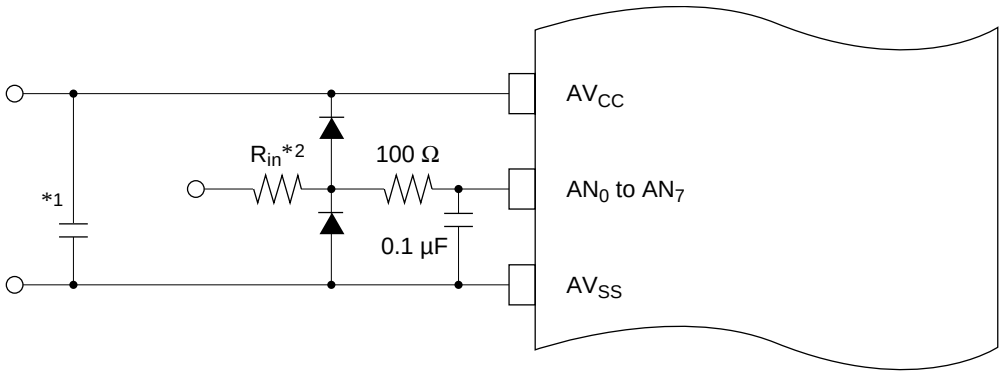
Notes on Board Design: In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN_0 to AN_7), and analog power supply (AV_{CC}) by the analog ground (AV_{SS}). Also, the analog ground (AV_{SS}) should be connected at one point to a stable digital ground (V_{SS}) on the board.

Notes on Noise Countermeasures: A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN_0 to AN_7) should be connected between AV_{CC} and AV_{SS} as shown in figure 17.7.

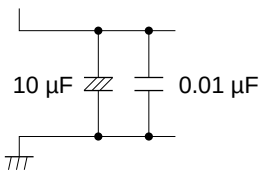
Also, the bypass capacitors connected to AV_{CC} and the filter capacitor connected to AN_0 to AN_7 must be connected to AV_{SS}.

If a filter capacitor is connected as shown in figure 17.7, the input currents at the analog input pins (AN_0 to AN_7) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.



Notes: Figures are reference values.

1.



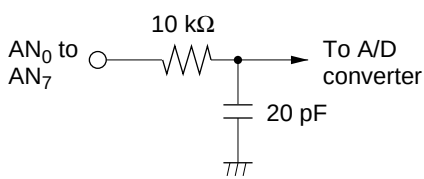
2. R_{in}: Input impedance

Figure 17.7 Example of Analog Input Protection Circuit

Table 17.5 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	10*	kΩ

Note: * When V_{cc} = 4.5 V to 5.5 V and $\phi \leq$ 12 MHz



Note: Figures are reference values.

Figure 17.8 Analog Input Pin Equivalent Circuit

A/D Conversion Precision Definitions: A/D conversion precision definitions for the H8/3577 Group and H8/3567 Group are given below.

- Resolution

  The number of A/D converter digital output codes

- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 17.10).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3FE) to B'111111111 (H'3FF) (see figure 17.10).

- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 17.9).

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error.

- Absolute precision

  The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

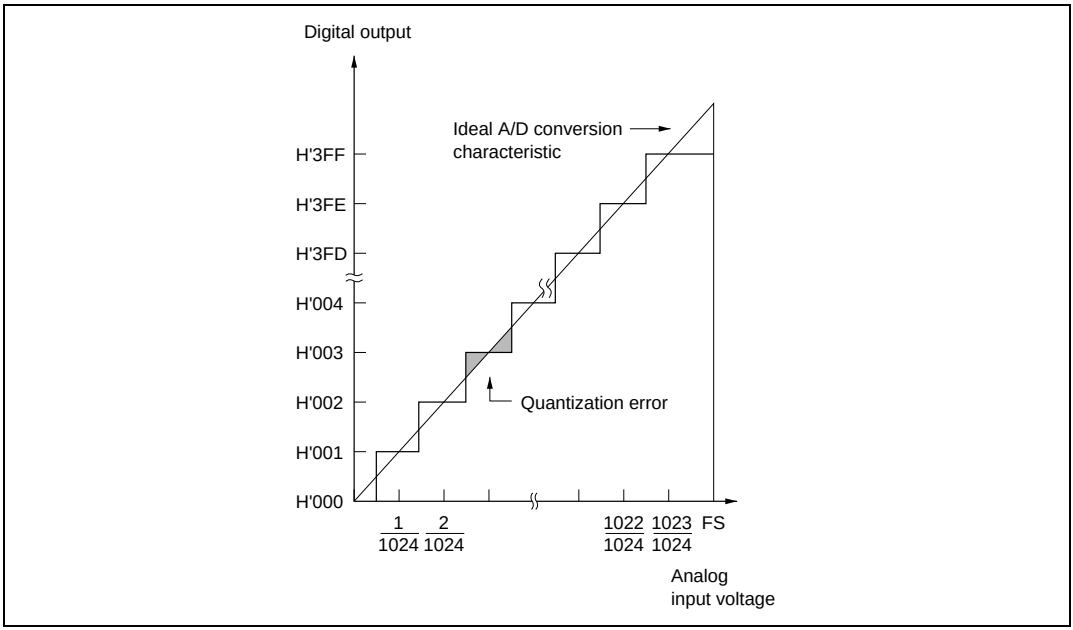


Figure 17.9 A/D Conversion Precision Definitions (1)

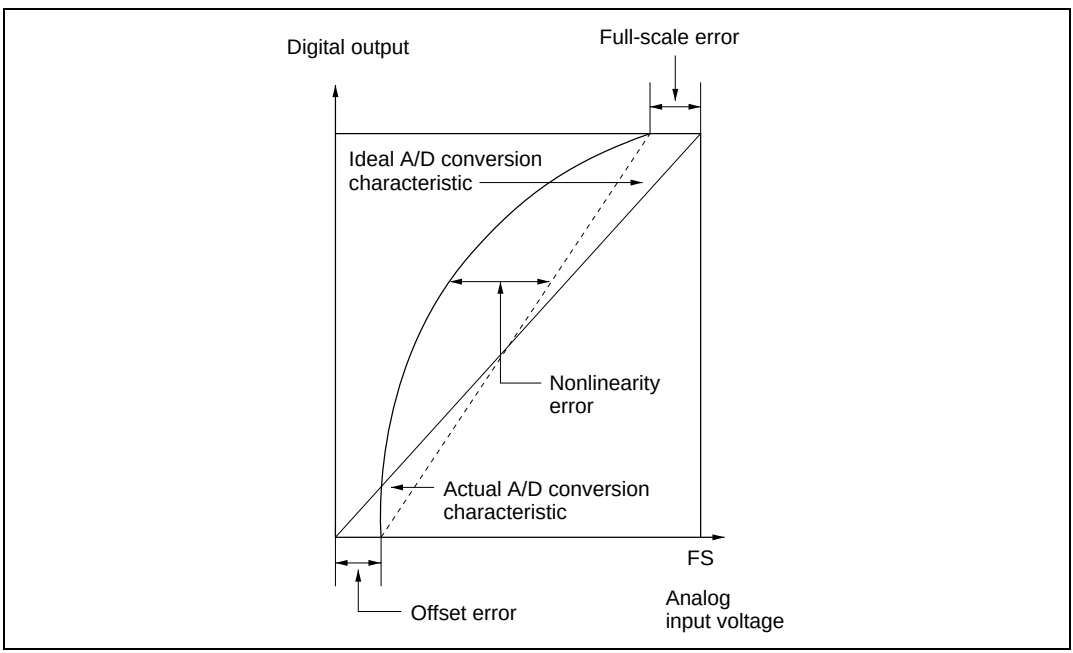


Figure 17.10 A/D Conversion Precision Definitions (2)

Permissible Signal Source Impedance: Analog input is designed so that conversion precision is guaranteed for an input signal for which the signal source impedance is 10 kΩ (when AV_{CC} = 4.5 to 5.5 V and $\phi \leq 12$ MHz, or when CSK = 0) or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 10 kΩ (when AV_{CC} = 4.5 to 5.5 V and $\phi \leq 12$ MHz, or when CSK = 0), charging may be insufficient and it may not be possible to guarantee the A/D conversion precision.

However, if a large capacitance is provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored.

But since a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µsec or greater).

When converting a high-speed analog signal, a low-impedance buffer should be inserted.

Influences on Absolute Precision: Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AV_{SS}.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, so acting as antennas.

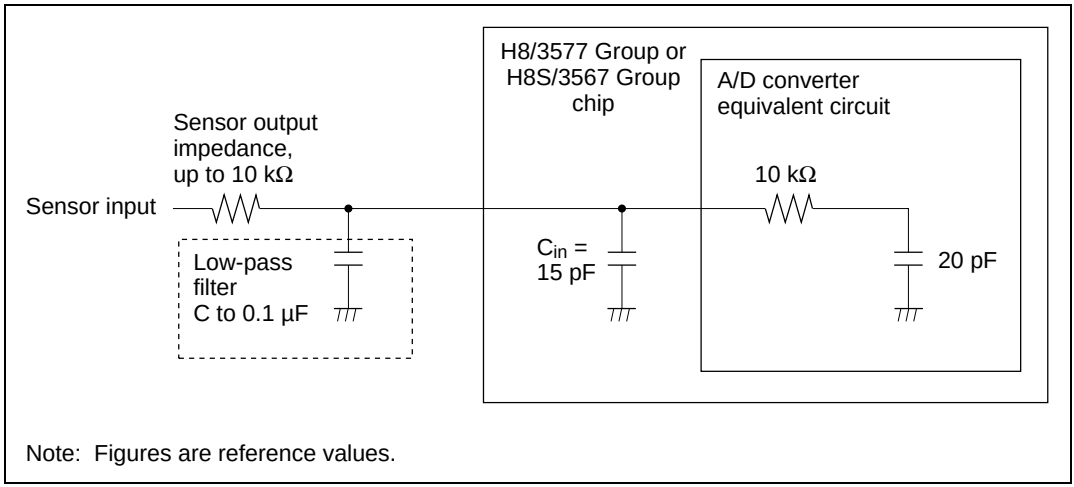


Figure 17.11 Example of Analog Input Circuit

Section 18 RAM

18.1 Overview

The H8/3577 Group and H8/3567 Group have 2 kbytes of on-chip high-speed static RAM. The on-chip RAM is connected to the bus master by a 16-bit data bus, enabling both byte data and word data to be accessed in two states. This makes it possible to perform fast word data transfer.

The on-chip RAM can be enabled or disabled by means of the RAM enable bit (RAME) in the system control register (SYSCR).

18.1.1 Block Diagram

Figure 18.1 shows a block diagram of the on-chip RAM.

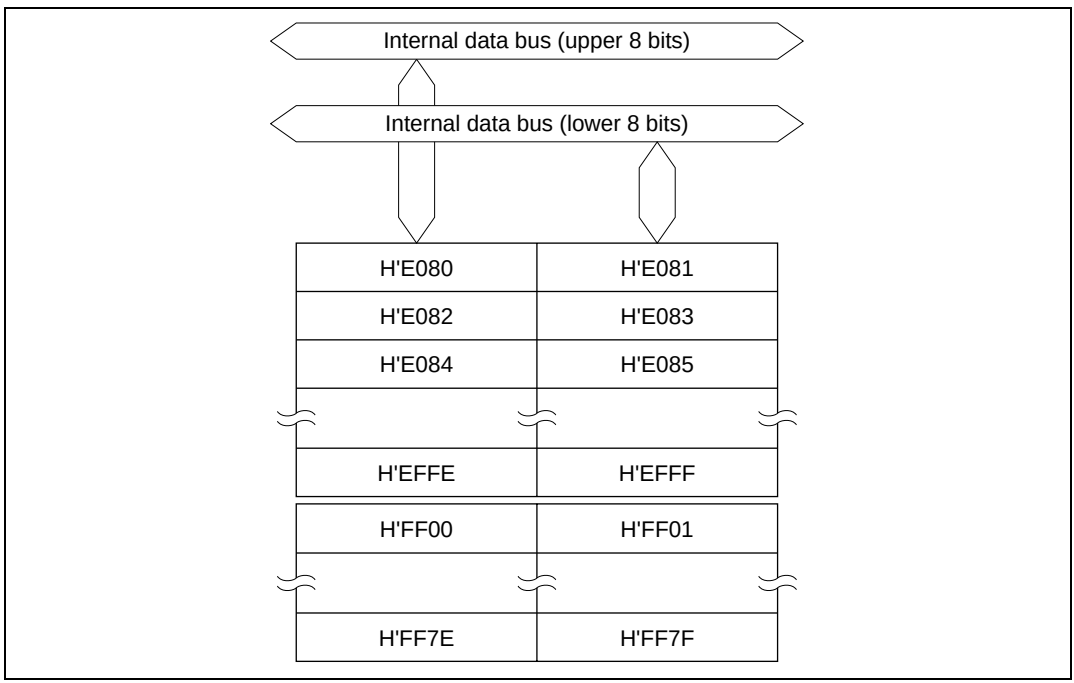


Figure 18.1 Block Diagram of RAM

18.1.2 Register Configuration

The on-chip RAM is controlled by SYSCR. Table 18.1 shows the register configuration.

Table 18.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address
System control register	SYSCR	R/W	H'09	H'FFC4

18.2 System Control Register (SYSCR)

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

The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. For details of other bits in SYSCR, see section 3.2.2, System Control Register.

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

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

18.3 Operation

When the RAME bit is set to 1, accesses to addresses H'E880 to H'EFFF and H'FF00 to H'FF7F are directed to the on-chip RAM. When the RAME bit is cleared to 0, the on-chip RAM is not accessed; a read will return an undefined value, and writes are invalid.

Since the on-chip RAM is connected to the bus master by a 16-bit data bus, it can be written to and read in byte or word units. Each type of access is performed in two states.

Even addresses use the upper 8 bits, and odd addresses use the lower 8 bits. Word data must start at an even address.

Section 19 ROM

19.1 Overview

The H8/3577, H8/3567, and H8/3567U have 56 kbytes of on-chip ROM (PROM or mask ROM), and the H8/3574, H8/3564, and H8/3564U have 32 kbytes. The ROM is connected to the bus master by a 16-bit data bus. The CPU accesses both byte and word data in two states, enabling faster instruction fetches and higher processing speed.

Figure 19.1 shows a block diagram of the ROM.

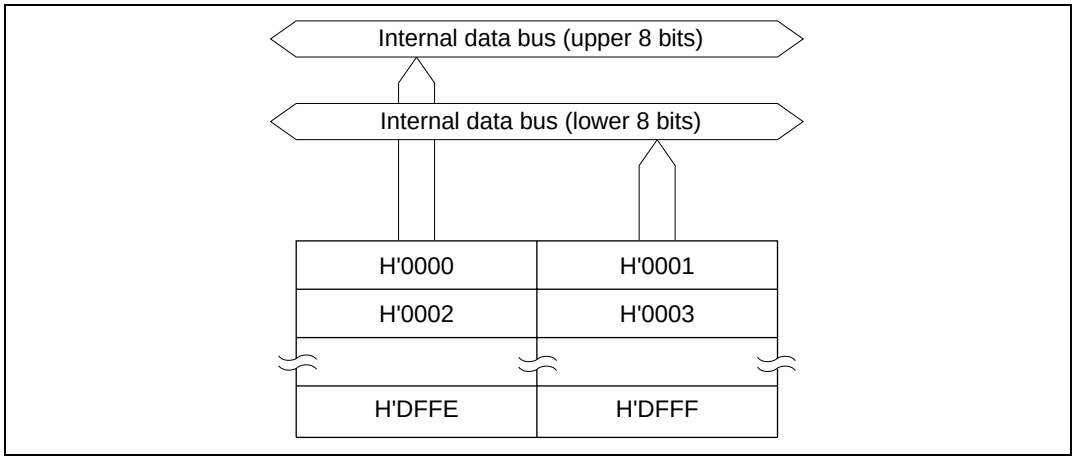


Figure 19.1 ROM Block Diagram (H8/3577, H8/3567, H8/3567U)

19.2 Operation

The on-chip ROM is connected to the CPU by a 16-bit data bus, and both byte and word data is accessed in two states. Even addresses are connected to the upper 8 bits, and odd addresses to the lower 8 bits. Word data must start at an even address.

19.3 Writer Mode (H8/3577, H8/3567, H8/3567U)

19.3.1 Writer Mode Setup

In writer mode the PROM versions of the H8/3577, H8/3567, and H8/3567U suspend the usual microcomputer functions to allow the on-chip PROM to be programmed. The programming method is the same as for the HN27C101.

To select writer mode, apply the signal inputs listed in table 19.1.

Table 19.1 Selection of Writer Mode

Pin		Input
H8/3577	Mode pin MD_1	Low
	Mode pin MD_0	Low
	$\overline{STBY}$ pin	Low
	Pins $P6_3$ and $P6_4$	High
H8/3567, H8/3567U	Mode pin $\overline{TEST}$	Low
	$\overline{STBY}$ pin	Low
	Pins $P4_7$ and $P5_2$	High

19.3.2 Socket Adapter Pin Assignments and Memory Map

The H8/3577, H8/3567, and H8/3567U can be programmed with a general-purpose PROM programmer by using a socket adapter to change the pin-out to 32 pins. See table 19.2. The same socket adapter can be used for H8/3577, H8/3567, and H8/3567U. Figures 19.2 to 19.4 show the socket adapter pin assignments.

Table 19.2 Socket Adapter

Package	Socket Adapter
64-pin QFP (H8/3577)	HS3297ESHS1H
64-pin shrink DIP (H8/3577)	HS3297ESSS1H
44-pin QFP (H8/3567)	TBD
42-pin shrink DIP (H8/3567)	TBD
64-pin QFP (H8/3567U)	TBD
64-pin shrink DIP (H8/3567U)	TBD

The PROM size is 56 kbytes for the H8/3577, H8/3567, and H8/3567U. Figure 19.5 shows memory maps of the H8/3577, H8/3567, and H8/3567U in writer mode. H'FF data should be specified for unused address areas in the on-chip PROM.

When programming with a PROM programmer, limit the program address range to H'0000 to H'DFFF for the H8/3577, H8/3567, and H8/3567U. Specify H'FF data for addresses H'E000 and above. If these addresses are programmed by mistake, it may become impossible to program or verify the PROM data. The same problem may occur if an attempt is made to program the chip in page programming mode. Note that the PROM versions are one-time programmable (OTP) microcomputers, packaged in plastic packages, and cannot be reprogrammed.

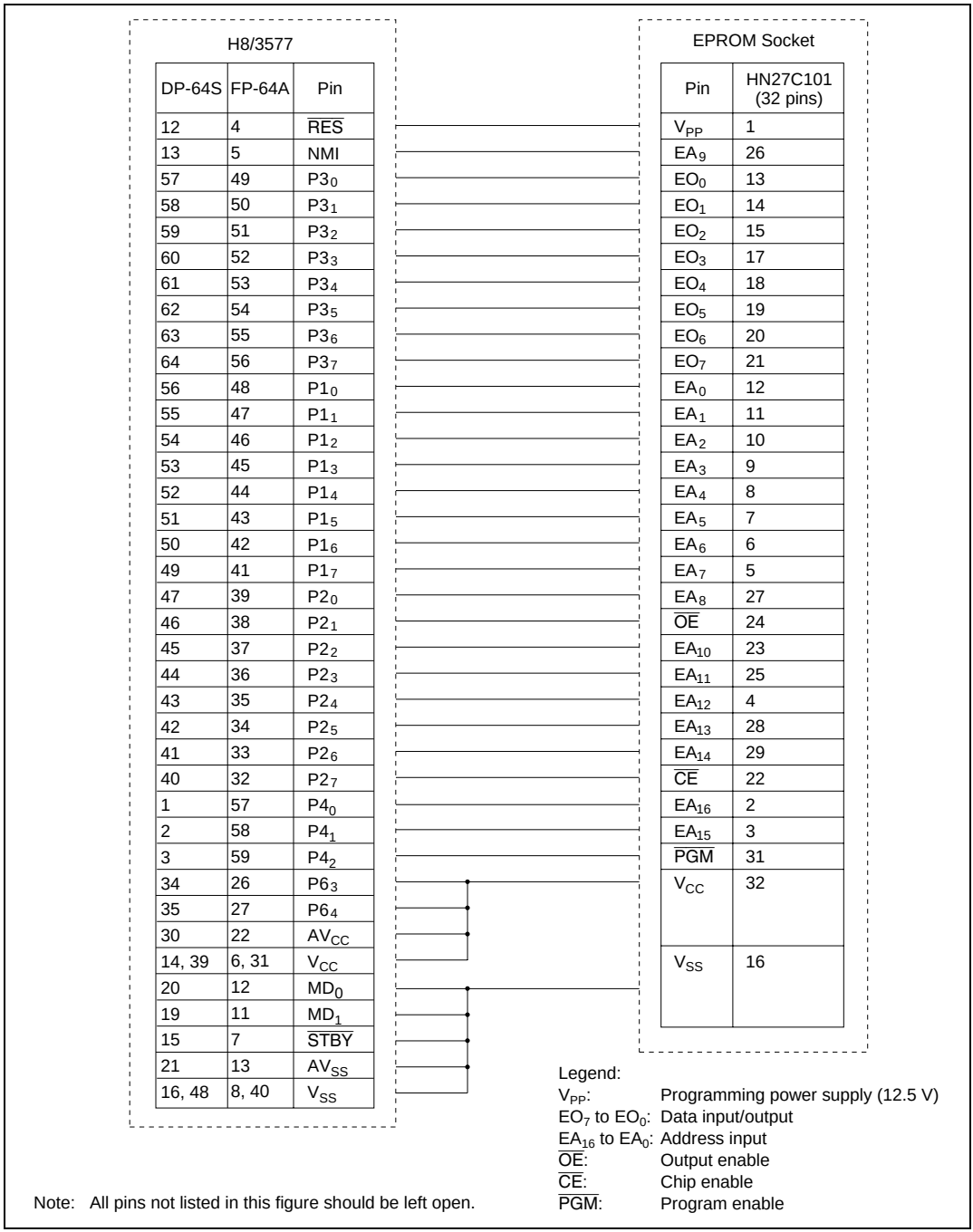


Figure 19.2 Socket Adapter Pin Assignments (H8/3577)

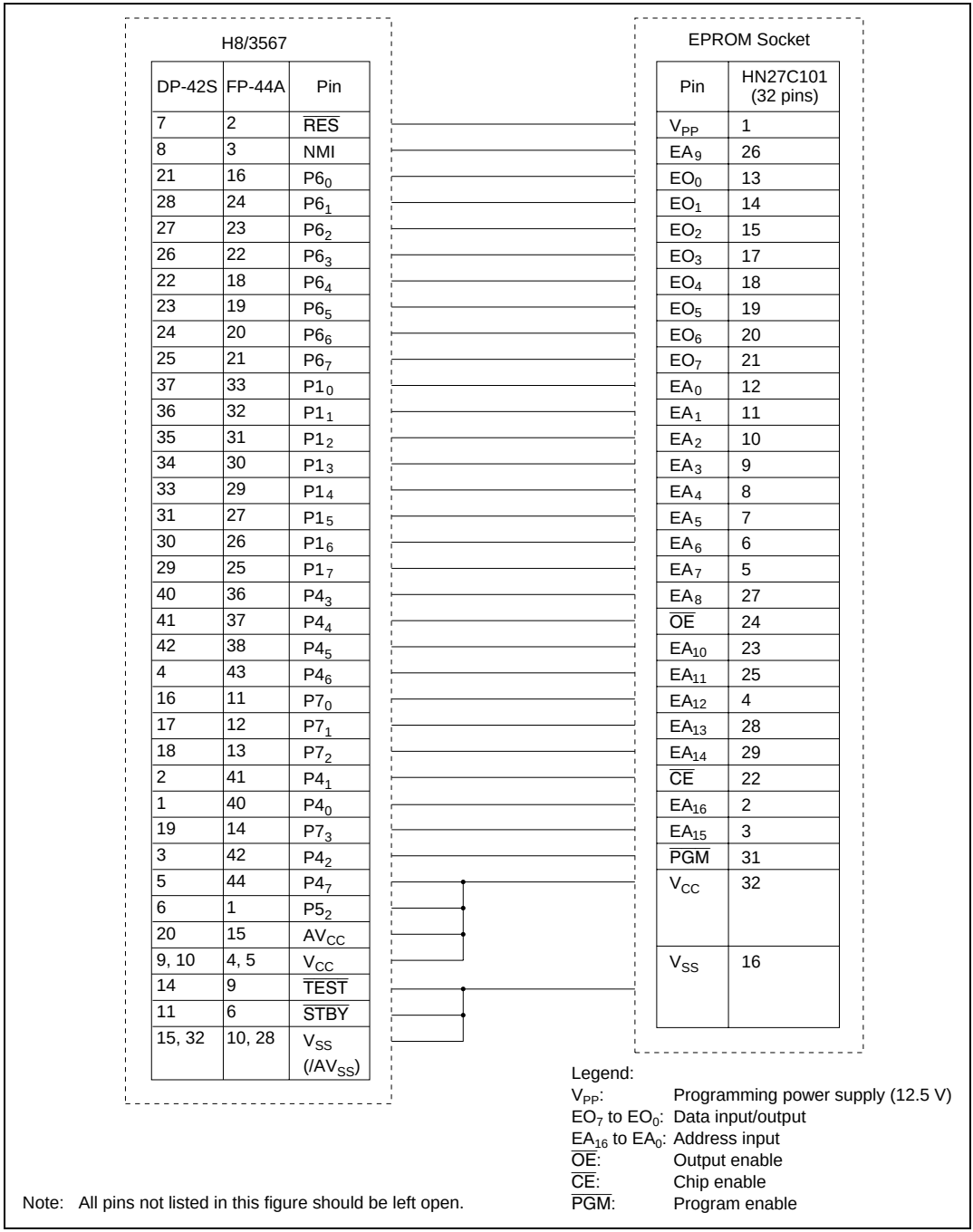


Figure 19.3 Socket Adapter Pin Assignments (H8/3567)

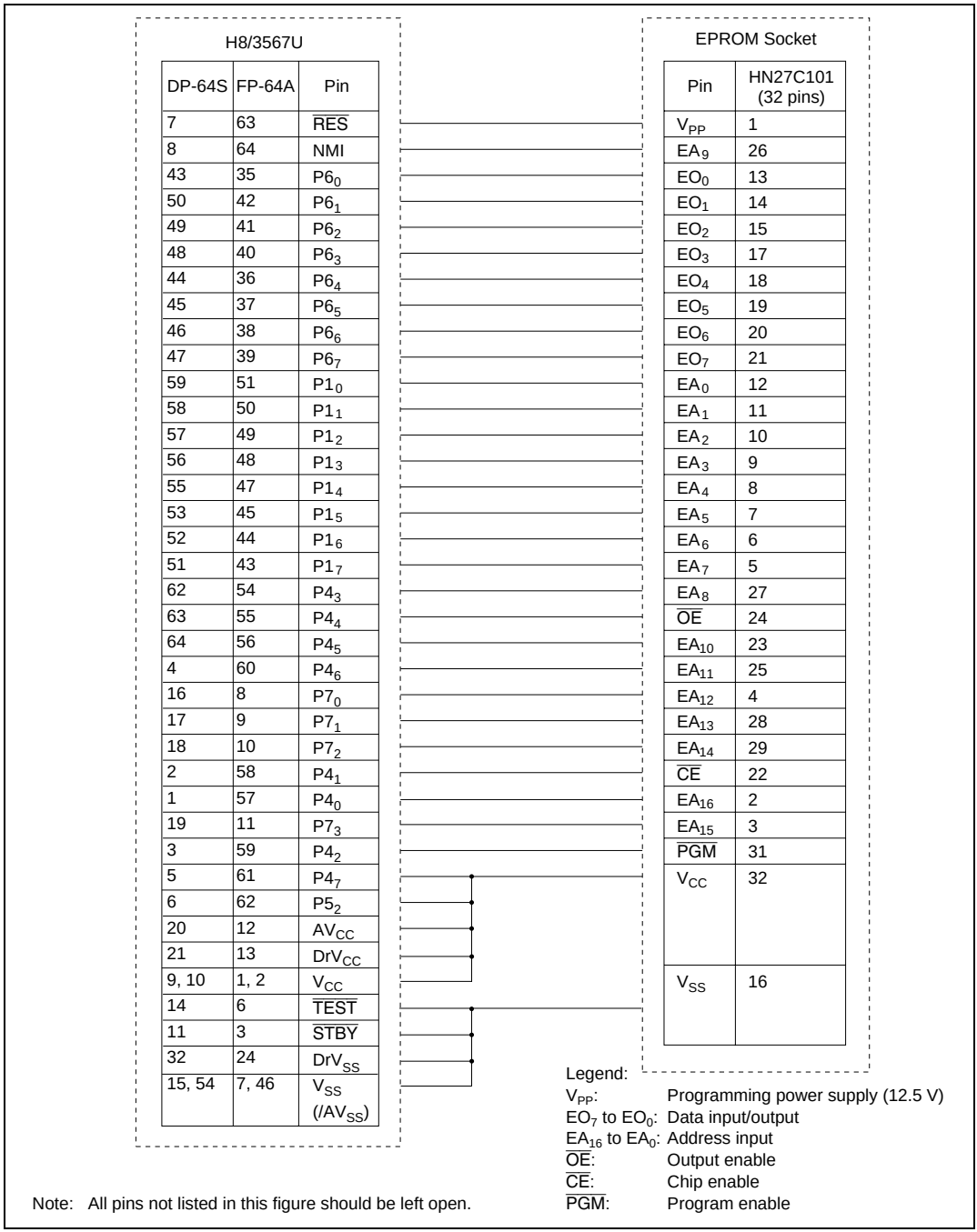


Figure 19.4 Socket Adapter Pin Assignments (H8/3567U)

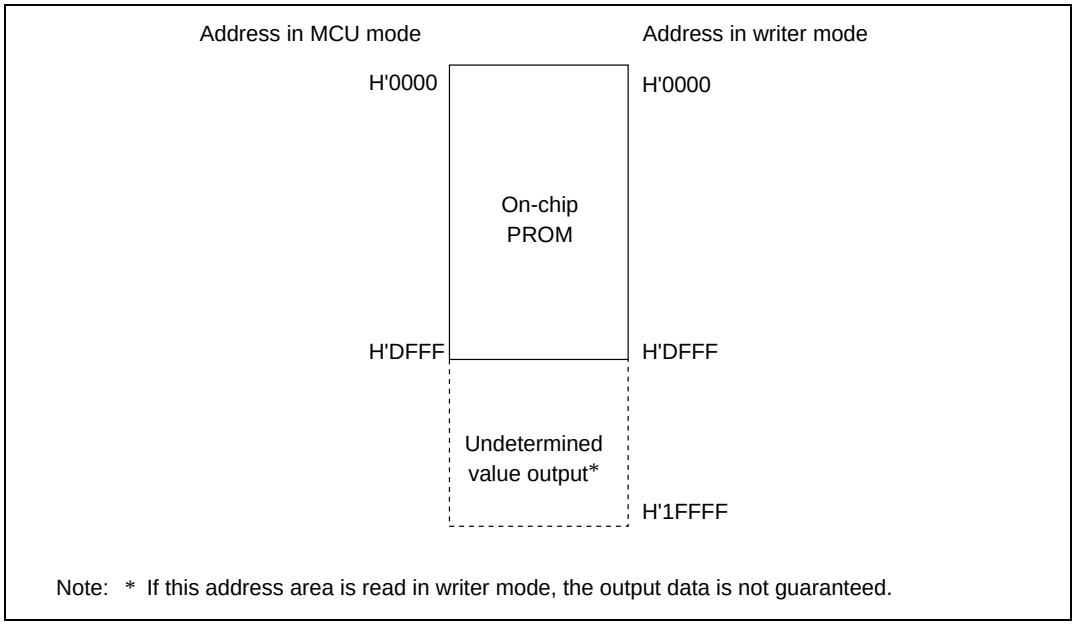


Figure 19.5 Memory Map in Writer Mode

19.4 PROM Programming

The write, verify, and other sub-modes of the writer mode are selected as shown in table 19.3.

Table 19.3 Selection of Sub-Modes in Writer Mode

Sub-Mode	$\overline{CE}$	$\overline{OE}$	$\overline{PGM}$	V_{PP}	V_{CC}	EO_7 to EO_0	EA_{16} to EA_0
Write	Low	High	Low	V_{PP}	V_{CC}	Data input	Address input
Verify	Low	Low	High	V_{PP}	V_{CC}	Data output	Address input
Programming inhibited	Low Low High High	Low High Low High	Low High Low High	V_{PP}	V_{CC}	High impedance	Address input

The H8/3577, H8/3567, and H8/3567U PROM have the same standard read/write specifications as the HN27C101 EPROM. Page programming is not supported, however, so do not select page programming mode. PROM programmers that provide only page programming cannot be used. When selecting a PROM programmer, check that it supports a byte-at-a-time high-speed programming mode. Be sure to set the address range to H'0000 to H'DFFF for the H8/3577, H8/3567, and H8/3567U.

19.4.1 Programming and Verification

An efficient, high-speed programming procedure can be used to program and verify PROM data. This procedure programs data quickly without subjecting the chip to voltage stress and without sacrificing data reliability. It leaves the data undefined in unused addresses.

Figure 19.6 shows the basic high-speed programming flowchart.

Tables 19.4 and 19.5 list the electrical characteristics of the chip in writer mode. Figure 19.7 shows a program/verify timing chart.

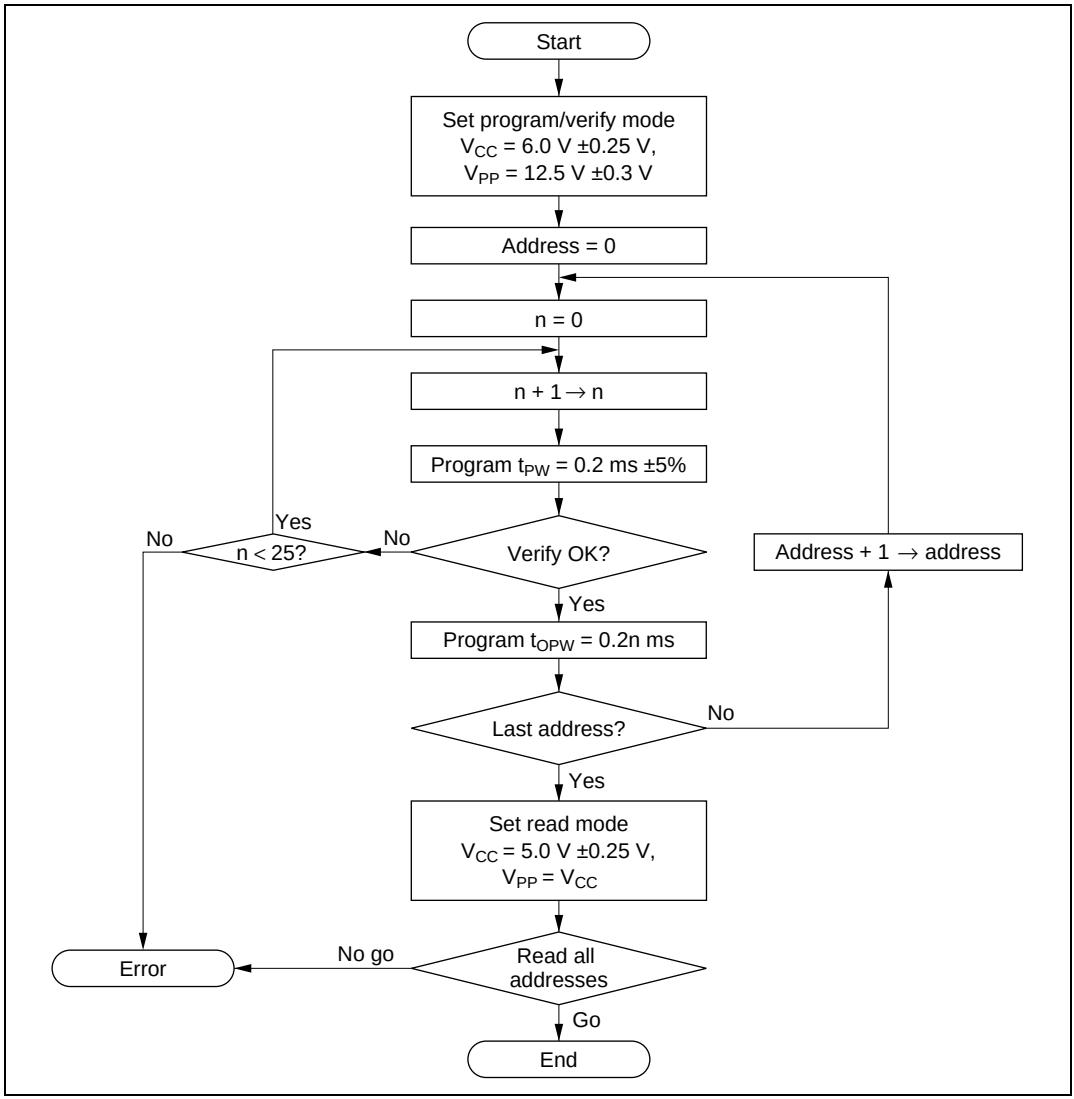


Figure 19.6 High-Speed Programming Flowchart

Table 19.4 DC Characteristics

When V_{CC} = 6.0 V ±0.25 V, V_{PP} = 12.5 V ±0.3 V, V_{SS} = 0 V, Ta = 25°C ±5°C

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input high voltage	EO_7–EO_0, EA_{16}–EA_0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	V_{IH}	2.4	—	V_{CC} + 0.3	V	
Input low voltage	EO_7–EO_0, EA_{16}–EA_0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	V_{IL}	–0.3	—	0.8	V	
Output high voltage	EO_7–EO_0	V_{OH}	2.4	—	—	V	I_{OH} = –200 µA
Output low voltage	EO_7–EO_0	V_{OL}	—	—	0.45	V	I_{OL} = 1.6 mA
Input leakage current	EO_7–EO_0, EA_{16}–EA_0, $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	$\|I_{LI}\|$	—	—	2	µA	V_{in} = 5.25 V/0.5 V
V_{CC} current		I_{CC}	—	—	40	mA	
V_{PP} current		I_{PP}	—	—	40	mA	

Table 19.5 AC Characteristics

When V_{CC} = 6.0 V ±0.25 V, V_{PP} = 12.5 V ±0.3 V, Ta = 25°C ±5°C

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Address setup time	t_{AS}	2	—	—	µs	See figure 19.7*
$\overline{OE}$ setup time	t_{OES}	2	—	—	µs	
Data setup time	t_{DS}	2	—	—	µs	
Address hold time	t_{AH}	0	—	—	µs	
Data hold time	t_{DH}	2	—	—	µs	
Data output disable time	t_{DF}	—	—	130	ns	
V_{PP} setup time	t_{VPS}	2	—	—	µs	
Program pulse width	t_{PW}	0.19	0.20	0.21	ms	
$\overline{OE}$ pulse width for overwrite-programming	t_{OPW}	0.19	—	5.25	ms	
V_{CC} setup time	t_{VCS}	2	—	—	µs	
$\overline{CE}$ setup time	t_{CES}	2	—	—	µs	
Data output delay time	t_{OE}	0	—	150	ns	

Note: * Input pulse level: 0.8 V to 2.2 V
Input rise/fall time ≤ 20 ns
Timing reference levels: input—1.0 V, 2.0 V; output—0.8 V, 2.0 V

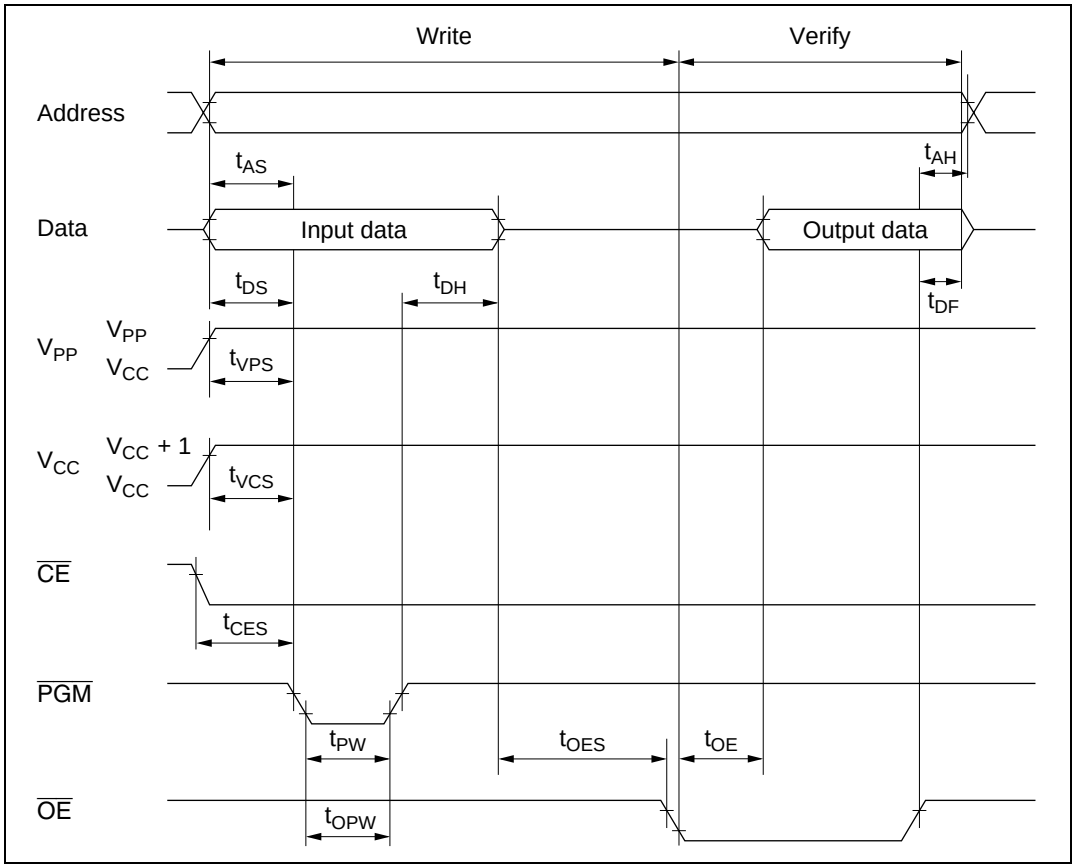


Figure 19.7 PROM Program/Verify Timing

19.4.2 Notes on Programming

(1) Program with the specified voltages and timing. The programming voltage (V_{PP}) is 12.5 V.

Caution: Applied voltages in excess of the specified values can permanently destroy the chip. Be particularly careful about the PROM programmer's overshoot characteristics.

If the PROM programmer is set to HN27C101 specifications, V_{PP} will be 12.5 V.

(2) Before writing data, check that the socket adapter and chip are correctly mounted in the PROM writer. Overcurrent damage to the chip can result if the index marks on the PROM programmer, socket adapter, and chip are not correctly aligned.

(3) Don't touch the socket adapter or chip while writing. Touching either of these can cause contact faults and write errors.

(4) Page programming is not supported. Do not select page programming mode.

(5) The PROM size is 56 kbytes. Set the address range to H'0000 to H'DFFF for the H8/3577, H8/3567, and H8/3567U. When programming, specify H'FF data for unused address areas (H'E000 to H'1FFFF).

19.4.3 Reliability of Programmed Data

An effective way to assure the data holding characteristics of the programmed chips is to bake them at 150°C, then screen them for data errors. This procedure quickly eliminates chips with PROM memory cells prone to early failure.

Figure 19.8 shows the recommended screening procedure.

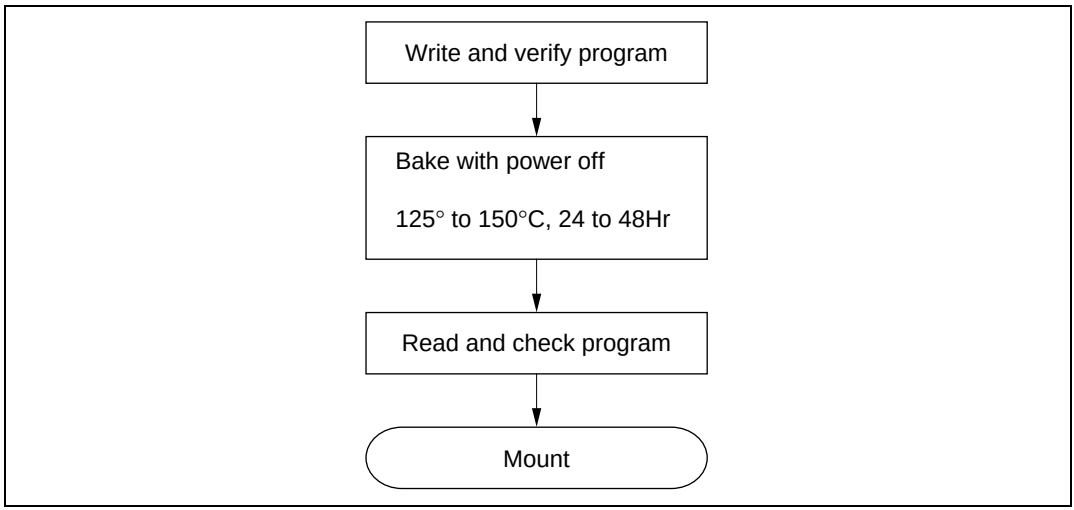


Figure 19.8 Recommended Screening Procedure

If a group of write errors occurs while the same PROM programmer is in use, stop programming and check the PROM programmer and socket adapter for defects.

Please inform Renesas Technology of any abnormal conditions noted during programming or in screening of program data after high-temperature baking.

Section 20 Clock Pulse Generator

20.1 Overview

The H8/3577 Group and H8/3567 Group have an on-chip clock pulse generator (CPG) that generates the system clock (φ), the bus master clock, and internal clocks.

The clock pulse generator consists of an oscillator circuit, a duty adjustment circuit, clock selection circuit, medium-speed clock divider, bus master clock selection circuit.

20.1.1 Block Diagram

Figure 20.1 shows a block diagram of the clock pulse generator.

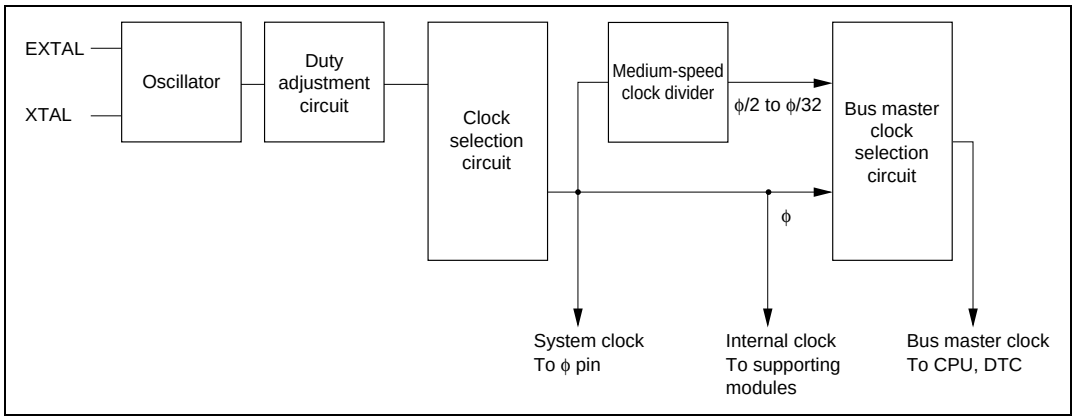


Figure 20.1 Block Diagram of Clock Pulse Generator

20.1.2 Register Configuration

The clock pulse generator is controlled by the standby control register (SBYCR). Table 20.1 shows the register configuration.

Table 20.1 CPG Registers

Name	Abbreviation	R/W	Initial Value	Address
Standby control register	SBYCR	R/W	H'00	H'FF84

20.2 Register Descriptions

20.2.1 Standby Control Register (SBYCR)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

Only bits 0 to 2 are described here. For a description of the other bits, see section 21.2.1, Standby Control Register (SBYCR).

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

Bits 2 to 0—System Clock Select 2 to 0 (SCK2 to SCK0): These bits select the bus master clock for high-speed mode and medium-speed mode.

Bit 2	Bit 1	Bit 0		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master is in high-speed mode	(Initial value)
		1	Medium-speed clock is ϕ/2	
	1	0	Medium-speed clock is ϕ/4	
		1	Medium-speed clock is ϕ/8	
1	0	0	Medium-speed clock is ϕ/16	
		1	Medium-speed clock is ϕ/32	
	1	—	—	

20.3 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

20.3.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 20.2. Select the damping resistance R_d according to table 20.2. An AT-cut parallel-resonance crystal should be used.

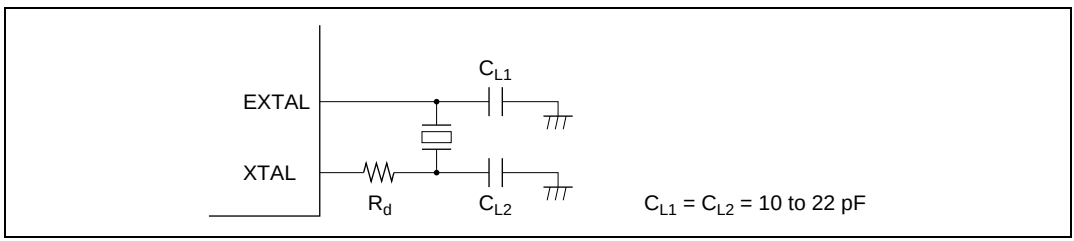


Figure 20.2 Connection of Crystal Resonator (Example)

Table 20.2 Damping Resistance Value

Frequency (MHz)	2	4	8	10	12	16	20
R_d (Ω)	1 k	500	200	0	0	0	0

Crystal resonator: Figure 20.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 20.3 and the same frequency as the system clock (ϕ).

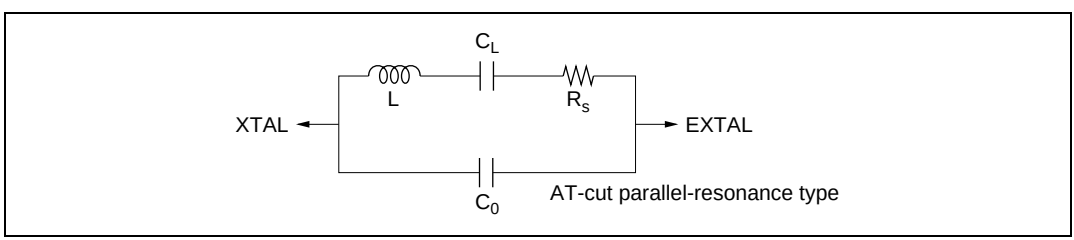


Figure 20.3 Crystal Resonator Equivalent Circuit

Table 20.3 Crystal Resonator Parameters

Frequency (MHz)	2	4	8	10	12	16	20
R_S max (Ω)	500	120	80	70	60	50	40
C_0 max (pF)	7	7	7	7	7	7	7

Note on Board Design: When a crystal resonator is connected, the following points should be noted.

Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 20.4.

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins.

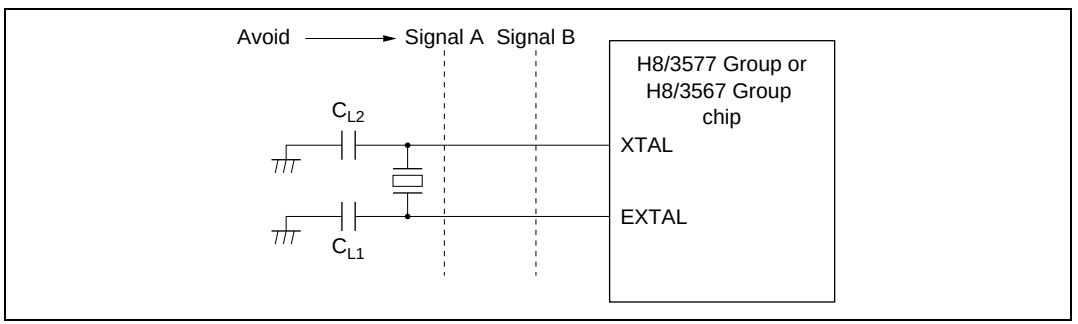
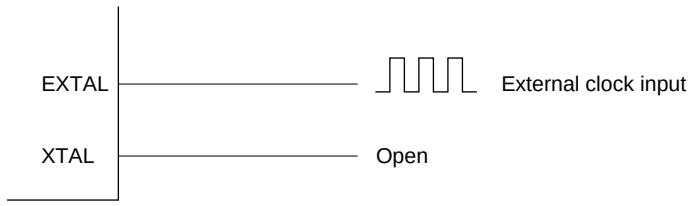


Figure 20.4 Example of Incorrect Board Design

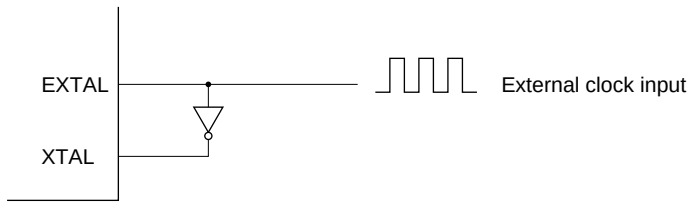
20.3.2 External Clock Input

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 20.5. If the XTAL pin is left open, make sure that stray capacitance is no more than 10 pF.

In example (b), make sure that the external clock is held high in standby mode.



(a) XTAL pin left open



(b) Complementary clock input at XTAL pin

Figure 20.5 External Clock Input (Examples)

External Clock: The external clock signal should have the same frequency as the system clock (ϕ).

Table 20.4 and figure 20.6 show the input conditions for the external clock.

Table 20.4 External Clock Input Conditions

		V_{cc} = 5.0 V ±10%				
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**	
External clock input low pulse width	t_{EXL}	20	—	ns	Figure 20.6	
External clock input high pulse width	t_{EXH}	20	—	ns		
External clock rise time	t_{EXr}	—	5	ns		
External clock fall time	t_{EXf}	—	5	ns		
Clock low pulse width	t_{CL}	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz	Figure 22.4
		80	—	ns	$\phi < 5$ MHz	
Clock high pulse width	t_{CH}	0.4	0.6	t_{cyc}	$\phi \geq 5$ MHz	
		80	—	ns	$\phi < 5$ MHz	

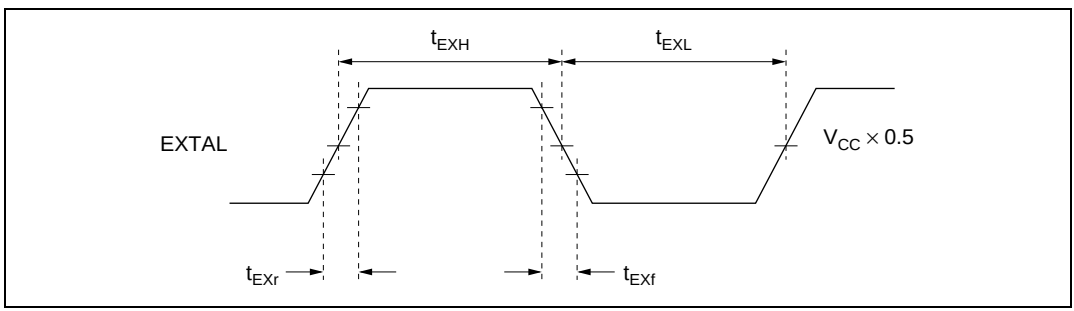


Figure 20.6 External Clock Input Timing

Table 20.5 shows the external clock output settling delay time, and figure 20.7 shows the external clock output settling delay timing. The oscillator and duty adjustment circuit have a function for adjusting the waveform of the external clock input at the EXTAL pin. When the prescribed clock signal is input at the EXTAL pin, internal clock signal output is fixed after the elapse of the external clock output settling delay time (t_{DEXT}). As the clock signal output is not fixed during the t_{DEXT} period, the reset signal should be driven low to maintain the reset state.

Table 20.5 External Clock Output Settling Delay Time

Conditions: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0 V

Item	Symbol	Min	Max	Unit	Notes
External clock output settling delay time	t_{DEXT}*	500	—	µs	Figure 20.7

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

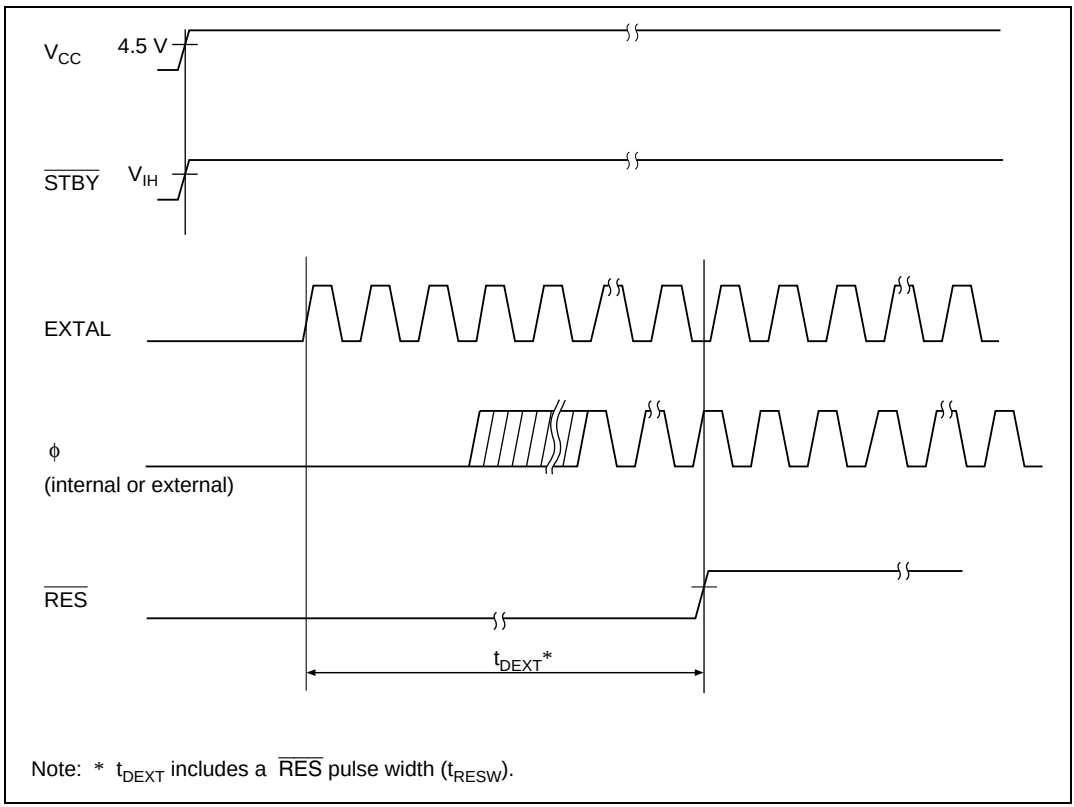


Figure 20.7 External Clock Output Settling Delay Timing

20.4 Duty Adjustment Circuit

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

20.5 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate ϕ/2, ϕ/4, ϕ/8, ϕ/16, and ϕ/32 clocks.

20.6 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the system clock (ϕ) or one of the medium-speed clocks (ϕ/2, ϕ/4, ϕ/8, ϕ/16, or ϕ/32) to be supplied to the bus master, according to the settings of bits SCK2 to SCK0 in SBYCR.

20.7 Universal Clock Pulse Generator [H8/3567 Group Version with On-Chip USB]

The H8/3567 Group version with an on-chip USB has a USB clock pulse generator (UCPG) that generates the 48 MHz USB clock (CLK48) from an 8, 12, 16, or 20 MHz input clock. The input clock can be selected from (1) the 12 MHz crystal oscillator or (2) the system clock (only when the system clock is 8, 12, 16, or 20 MHz).

The USB clock pulse generator consists of an oscillator, clock selection circuit, and frequency division/multiplication circuit.

20.7.1 Block Diagram

Figure 20.8 shows a block diagram of the USB clock pulse generator.

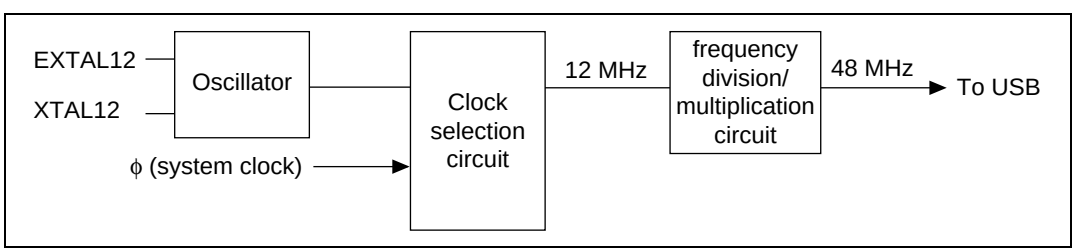


Figure 20.8 Block Diagram of USB Clock Pulse Generator

20.7.2 Registers

Table 20.6 USB Clock Pulse Generator Registers

Name	Abbreviation	R/W	Initial Value	Address
USB control/status register 0	USBCSR0	R/W	H'00	H'FDF5
USB control register	USBCR	R/W	H'7F	H'FDFD
USB PLL control register	UPLLCR	R/W	H'01	H'FDFE

USB Control/Status Register 0 (USBCSR0)

Bit	7	6	5	4	3	2	1	0
	DP5CNCT	DP4CNCT	DP3CNCT	DP2CNCT	EP0STOP	EPIVLD	EP0OTC	CKSTOP
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W

USBCSR0 contains flags (DPCNCT) that indicate the USB hubs' downstream port connection status, and bits that control the operation of the USB function.

Only bit 0 is described here. For details of the other bits, see section 7.2.11, USB Control/Status Register 0 (USBCSR0).

USBCSR0 is initialized to H'00 by a system reset, and bits 3 to 0 are also cleared to 0 by a function soft reset.

Bit 0—Clock Stop (CKSTOP): Controls the USB function operating clock. When the USB function is placed in the suspend state due to a bus idle condition, this bit should be set to 1 after the necessary processing is completed. The clock supply to the USB function is then stopped, reducing power consumption.

When the CKSTOP bit is set to 1, writes to USB module registers are invalid. If these registers are read, the contents of the read data are not guaranteed, but there are no read-related status changes (such as decrementing of FVSR).

If a bus idle condition of the specified duration or longer is detected, the suspend IN interrupt flag is set, and when a change in the bus status is subsequently detected the suspend OUT interrupt flag is set. When the suspend OUT interrupt flag is set, the CKSTOP bit is simultaneously cleared to 0.

Bit 0

CKSTOP	Description
0	Clock is supplied to USB function (Initial value) [Clearing conditions] • System reset • Function soft reset • Suspend OUT interrupt flag setting
1	Clock supply to USB function is stopped [Setting condition] When 1 is written to CKSTOP after reading CKSTOP = 0 in the function suspend state.

USB Control Register (USBCR)

Bit	7	6	5	4	3	2	1	0
	FADSEL	FONLY	FNCSTP	UIFRST	HPLLRST	HSRST	FPLLRST	FSRST
Initial value	0	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

USBCR contains bits (FADSEL, FONLY, FNCSTP) that control USB function and USB hub internal connection, and reset control bits for sequential enabling of the operation of each part according to the procedure in USB module initialization.

Only bits 3 and 1 are described here. For details of the other bits, see section 7.2.18, USB Control Register (USBCR).

USBCR is initialized to H'7F by a system reset [in an H8/3567 reset (by $\overline{\text{RES}}$ input or the watchdog timer), and in hardware standby mode]. It is not initialized in software standby mode.

Bit 3—Hub Block PLL Soft Reset (HPLLRST): Resets the USB bus clock synchronization circuit (DPLL) in the hub.

When HPLLRST is set to 1, the DPLL circuit in the USB hub block is reset, and bus clock synchronous operation halts. HPLLRST is cleared to 0 after PLL operation stabilizes.

Bit 3

HPLLRST	Description	
0	USB hub block DPLL is placed in operational state	
1	USB hub block DPLL is placed in reset state	(Initial value)

Bit 1—Function Block PLL Soft Reset (FPLLRST): Resets the USB bus clock synchronization circuit (DPLL) in the USB function block.

When FPLLRST is set to 1, the DPLL circuit in the USB function block is reset, and bus clock synchronous operation halts. FPLLRST is cleared to 0 after PLL operation stabilizes.

Bit 1

FPLLRST	Description	
0	USB function block DPLL is placed in operational state	
1	USB function block DPLL is placed in reset state	(Initial value)

USB PLL Control Register (UPLLCR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	CKSEL2	CKSEL1	CKSEL0	PFSEL1	PFSEL0
Initial value	0	0	0	0	0	0	0	1
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W

UPLLCR contains bits that control the method of generating the USB function and USB hub operating clock.

UPLLCR is initialized to H'01 by a system reset [in an H8/3567 reset (by $\overline{\text{RES}}$ input or the watchdog timer), and in hardware standby mode]. It is not initialized in software standby mode.

Bits 4 to 2—Clock Source Select 2 to 0 (CKSEL2 to CKSEL0): These bits select the source of the clock supplied to the USB operating clock generator (PLL).

CKSEL0 selects either the USB clock pulse generator (XTAL12) or the system clock pulse generator (XTATL) as the clock source. The USB clock pulse generator starts operating when it is selected as a clock source. It operates with CKSEL2 = 1, CKSEL0 = 1.

When CKSEL2 = 1 and CKSEL1 = 1, the PLL operates.

When CKSEL1 is cleared to 0, a clock is not input to the PLL, and PLL operation halts. The 48 MHz signal from the USB clock pulse generator can be input directly as the USB operating clock.

When CKSEL2 is cleared to 0, a clock is not input to the PLL, and PLL operation halts.

Bit 4	Bit 3	Bit 2		
CKSEL2	**CKSEL1**	**CKSEL0**	**Description**	
0	0	0	PLL operation halted, clock input halted	(Initial value)
	—	—	PLL operation halted, clock input halted	
1	0	0	Setting prohibited	
		1	PLL operation halted USB clock pulse generator (XTAL12: 48 MHz) used directly instead of PLL output	
	1	0	PLL operates with system clock pulse generator (XTAL) as clock source	
		1	PLL operates with USB clock pulse generator (XTAL12) as clock source	

Bits 1 and 0—PLL Frequency Select 1 and 0 (PFSEL1, PFSEL0): These bits select the frequency of the clock supplied to the USB operating clock generator (PLL).

The PLL generates the 48 MHz USB operating clock using the frequency selected with these bits as the clock source frequency.

Bit 1	Bit 0		
PFSEL1	**PFSEL0**	**Description**	
0	0	PLL input clock is 8 MHz	
	1	PLL input clock is 12 MHz	(Initial value)
1	0	PLL input clock is 16 MHz	
	1	PLL input clock is 20 MHz	

Section 21 Power-Down State

21.1 Overview

In addition to the normal program execution state, the H8/3577 Group and H8/3567 Group have a power-down state in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip supporting modules, and so on.

The operating modes are as follows:

1. High-speed mode
2. Medium-speed mode
3. Sleep mode
4. Module stop mode
5. Software standby mode
6. Hardware standby mode

Of these, 2 to 6 are power-down modes. Sleep mode is a CPU mode, medium-speed mode is a CPU operating clock state, and module stop mode is an on-chip supporting module mode. Certain combinations of these modes can be set.

After a reset, the MCU is in high-speed mode and module stop mode.

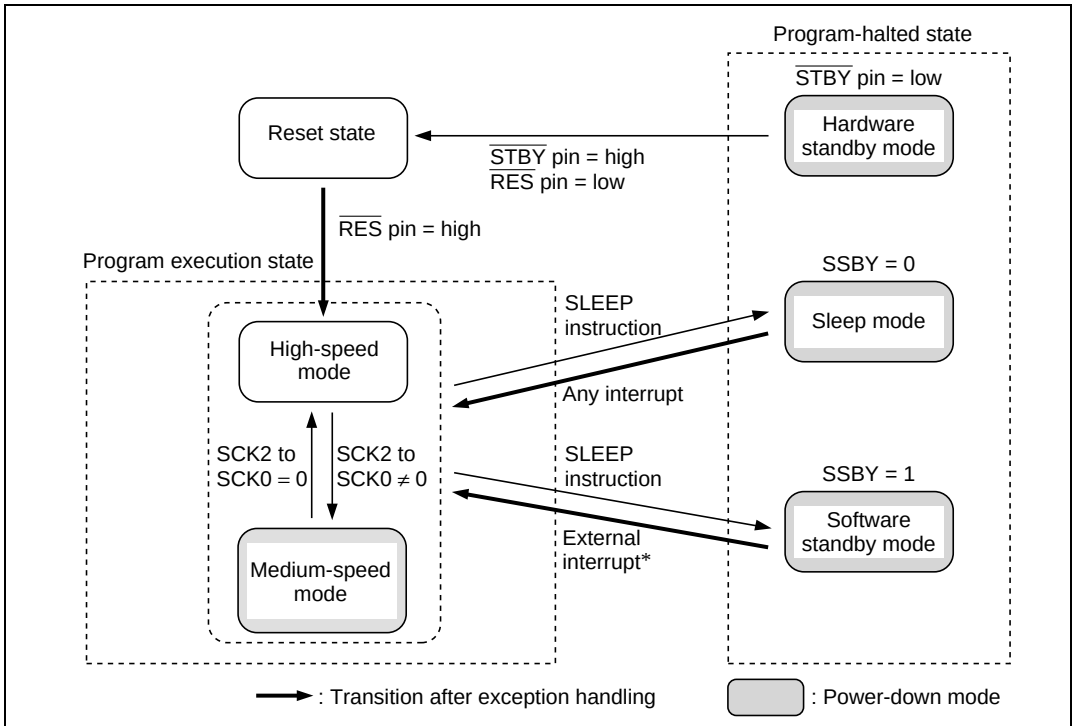
Table 21.1 shows the internal chip states in each mode, and table 21.2 shows the conditions for transition to the various modes. Figure 21.1 shows a mode transition diagram.

Table 21.1 H8/3577 Group and H8/3567 Group Internal States in Each Mode

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby
System clock oscillator		Functioning	Functioning	Functioning	Functioning	Halted	Halted
CPU operation	Instructions	Functioning	Medium-speed	Halted	Functioning	Halted	Halted
	Registers	Functioning	Medium-speed	Retained	Functioning	Retained	Undefined
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Halted
	IRQ0						
	IRQ1						
	IRQ2						
On-chip supporting module operation	WDT0	Functioning	Functioning	Functioning	Functioning	Halted (retained)	Halted (reset)
	TMR_0, TMR_1	Functioning	Functioning	Functioning	Functioning/ halted (retained)	Halted (retained)	Halted (reset)
	FRT						
	TMRX, Y						
	Timer connection						
	IIC0						
	IIC1						
	SCI0	Functioning	Functioning	Functioning	Functioning/ halted (reset)	Halted (reset)	Halted (reset)
	PWM						
	PWMX						
	A/D						
	RAM	Functioning	Functioning	Functioning	Functioning	Retained	Retained
	I/O	Functioning	Functioning	Functioning	Functioning	Retained	High impedance
	USB	Functioning	Functioning	Functioning	Functioning/ halted*	Functioning/ halted*	Halted (reset)

Note: "Halted (retained)" means that internal register values are retained. The internal state is "operation suspended."
"Halted (reset)" means that internal register values and internal states are initialized.
In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

* Functioning (USB hub part only) when the USB clock (XTAL12, EXTAL12) is selected as a USB operating clock, and halted (retained) when not selected.



Notes: When a transition is made between modes by means of an interrupt, transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
From any state except hardware standby mode, a transition to the reset state occurs whenever RES goes low.
From any state, a transition to hardware standby mode occurs when STBY goes low.

* NMI, IRQ0 to IRQ2

Figure 21.1 Mode Transitions

Table 21.2 Power-Down Mode Transition Conditions

State before Transition	Control Bit States at Time of Transition: SSBY	State after Transition by SLEEP Instruction	State after Return by Interrupt
High-speed/ medium-speed	0	Sleep	High-speed/ medium-speed
	1	Software standby	High-speed/ medium-speed

21.1.1 Register Configuration

The power-down state is controlled by the SBYCR and MSTPCR registers. Table 21.3 summarizes these registers.

Table 21.3 Power-Down State Registers

Name	Abbreviation	R/W	Initial Value	Address
Standby control register	SBYCR	R/W	H'00	H'FF84
Module stop control register	MSTPCRH	R/W	H'3F	H'FF86
	MSTPCRL	R/W	H'FF	H'FF87

21.2 Register Descriptions

21.2.1 Standby Control Register (SBYCR)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

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

Bit 7—Software Standby (SSBY): Determines the operating mode, in combination with other control bits, when a power-down mode transition is made by executing a SLEEP instruction. The SSBY setting is not changed by a mode transition due to an interrupt, etc.

Bit 7

SSBY	Description
0	Transition to sleep mode after execution of SLEEP instruction in high-speed mode or medium-speed mode (Initial value)
1	Transition to software standby mode, after execution of SLEEP instruction in high-speed mode or medium-speed mode

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the time the MCU waits for the clock to stabilize when software standby mode is cleared and a transition is made to high-speed mode or medium-speed mode by means of a specific interrupt or instruction. With crystal oscillation, refer to table 21.4 and make a selection according to the operating frequency so that the standby time is at least 8 ms (the oscillation settling time). With an external clock, any selection can be made.

Bit 6	Bit 5	Bit 4		
STS2	**STS1**	**STS0**	**Description**	
0	0	0	Standby time = 8192 states	(Initial value)
		1	Standby time = 16384 states	
	1	0	Standby time = 32768 states	
		1	Standby time = 65536 states	
1	0	0	Standby time = 131072 states	
		1	Standby time = 262144 states	
	1	0	Reserved	
		1	Standby time = 16 states	

Bit 3—Reserved: This bit cannot be modified and is always read as 0.

Bits 2 to 0—System Clock Select (SCK2 to SCK0): These bits select the clock for the bus master in high-speed mode and medium-speed mode.

Bit 2	Bit 1	Bit 0		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master is in high-speed mode	(Initial value)
		1	Medium-speed clock is ϕ/2	
	1	0	Medium-speed clock is ϕ/4	
		1	Medium-speed clock is ϕ/8	
1	0	0	Medium-speed clock is ϕ/16	
		1	Medium-speed clock is ϕ/32	
	1	—	—	

21.2.2 Module Stop Control Register (MSTPCR)

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR comprises two 8-bit readable/writable registers that perform module stop mode control.

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

MSTRCRH and MSTPCRL Bits 7 to 0—Module Stop (MSTP 15 to MSTP 0): These bits specify module stop mode. See table 21.3 for the method of selecting on-chip supporting modules.

MSTPCRH, MSTPCRL Bits 7 to 0

MSTP15 to MSTP0	Description	
0	Module stop mode is cleared	(Initial value of MSTP15, MSTP14)
1	Module stop mode is set	(Initial value of MSTP13 to MSTP0)

21.3 Medium-Speed Mode

When the SCK2 to SCK0 bits in SBYCR are set to 1 in high-speed mode, the operating mode changes to medium-speed mode at the end of the bus cycle. In medium-speed mode, the CPU operates on the operating clock ($\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, or $\phi/32$) specified by the SCK2 to SCK0 bits. On-chip supporting modules other than the bus masters always operate on the high-speed clock (ϕ).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if $\phi/4$ is selected as the operating clock, on-chip memory is accessed in 8 states, and internal I/O registers in 12 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, a transition is made to software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is driven low, a transition is made to the reset state, and medium-speed mode is cleared. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 21.2 shows the timing for transition to and clearance of medium-speed mode.

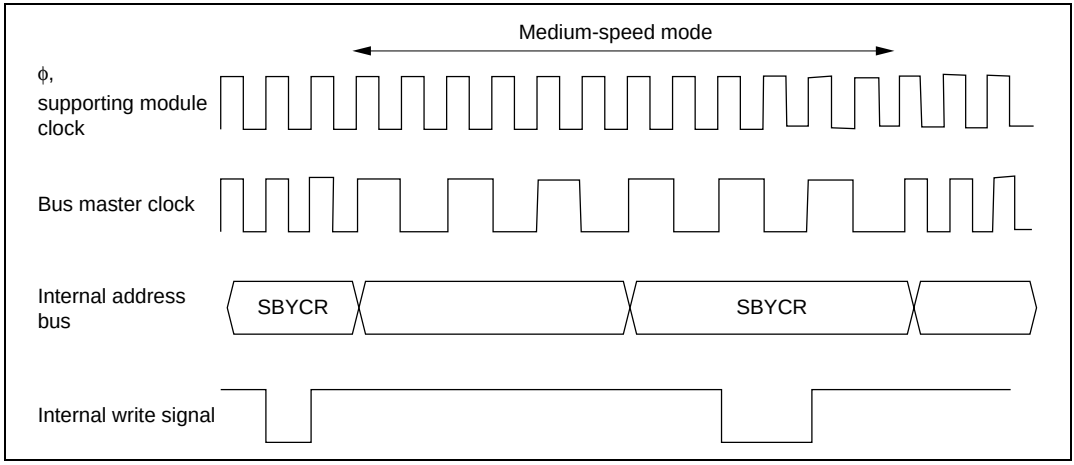


Figure 21.2 Medium-Speed Mode Transition and Clearance Timing

21.4 Sleep Mode

21.4.1 Sleep Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, the CPU enters sleep mode. In sleep mode, CPU operation stops but the contents of the CPU's internal registers are retained. Other supporting modules do not stop.

21.4.2 Clearing Sleep Mode

Sleep mode is cleared by any interrupt, or with the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

Clearing with an Interrupt: When an interrupt request signal is input, sleep mode is cleared and interrupt exception handling is started. Sleep mode will not be cleared if interrupts are disabled, or if interrupts other than NMI have been masked by the CPU.

Clearing with the $\overline{\text{RES}}$ Pin: When the $\overline{\text{RES}}$ pin is driven low, the reset state is entered. When the $\overline{\text{RES}}$ pin is driven high after the prescribed reset input period, the CPU begins reset exception handling.

Clearing with the $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

21.5 Module Stop Mode

21.5.1 Module Stop Mode

Module stop mode can be set for individual on-chip supporting modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

Table 21.4 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating again at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI, A/D converter, 8-bit PWM module, and 14-bit PWM module, are retained. Additionally, when the USB clock (XTAL12, EXTAL12) is selected as a USB operating clock, the USB module does not stop operating even when the MSTP1 bit is set to 1. To stop the

USB module, initialize UPLLCR to H'01 before setting the MSTP1 bit to 1. Also, it is recommended to initialize USBCR to H'7F to prepare for cancellation of the module stop state.

After reset release, all modules other than the DTC are in module stop mode.

When an on-chip supporting module is in module stop mode, read/write access to its registers is disabled.

Table 21.4 MSTP Bits and Corresponding On-Chip Supporting Modules

Register	Bit	Module
MSTPCRH	MSTP15*	—
	MSTP14*	—
	MSTP13	16-bit free-running timer (FRT)
	MSTP12	8-bit timers (TMR_0, TMR_1)
	MSTP11	8-bit PWM timer (PWM), 14-bit PWM timer (PWMX)
	MSTP10*	—
	MSTP9	A/D converter
	MSTP8	8-bit timers (TMRX, TMRY), timer connection
MSTPCRL	MSTP7	Serial communication interface 0 (SCI0)
	MSTP6*	—
	MSTP5*	—
	MSTP4	I^2C bus interface (IIC) channel 0
	MSTP3	I^2C bus interface (IIC) channel 1
	MSTP2*	—
	MSTP1	Universal serial bus interface (USB)
	MSTP0*	—

Note: * Bits 15, 14, 10, 6, 5, 2, and 0 can be read or written to, must be set to 1.

21.5.2 Usage Note

The MSTP bit for modules not included on-chip must be set to 1.

21.6 Software Standby Mode

21.6.1 Software Standby Mode

If a SLEEP instruction is executed when the SSBY bit in SBYCR is set to 1, software standby mode is entered. In this mode, the CPU, on-chip supporting modules, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip supporting modules other than the SCI, PWM, and PWMX, and of the I/O ports, are retained.*

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

Note: * When the USB clock (XTAL12, EXTAL12) is selected as a USB operating clock, the USB module does not stop operating even under the software standby mode. To realize the power save state, initialize UPLLCR to H'01 and USBCR to H'7F.

21.6.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pin $\overline{IRQ_0}$, $\overline{IRQ_1}$, or $\overline{IRQ_2}$), or by means of the $\overline{RES}$ pin or $\overline{STBY}$ pin.

Clearing with an Interrupt: When an NMI, IRQ0, IRQ1, or IRQ2 interrupt request signal is input, clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SYSCR, stable clocks are supplied to the entire chip, software standby mode is cleared, and interrupt exception handling is started.

Software standby mode cannot be cleared with an IRQ0, IRQ1, or IRQ2 interrupt if the corresponding enable bit has been cleared to 0 or has been masked by the CPU.

Clearing with the $\overline{RES}$ Pin: When the $\overline{RES}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire chip. Note that the $\overline{RES}$ pin must be held low until clock oscillation stabilizes. When the $\overline{RES}$ pin goes high, the CPU begins reset exception handling.

Clearing with the $\overline{STBY}$ Pin: When the $\overline{STBY}$ pin is driven low, a transition is made to hardware standby mode.

21.6.3 Setting Oscillation Settling Time after Clearing Software Standby Mode

Bits STS2 to STS0 in SBYCR should be set as described below.

Using a Crystal Oscillator: Set bits STS2 to STS0 so that the standby time is at least 8 ms (the oscillation settling time).

Table 21.5 shows the standby times for different operating frequencies and settings of bits STS2 to STS0.

Table 21.5 Oscillation Settling Time Settings

STS2	STS1	STS0	Standby Time	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz	2 MHz	Unit
0	0	0	8192 states	0.41	0.51	0.65	0.8	1.0	1.3	2.0	4.1	ms
		1	16384 states	0.82	1.0	1.3	1.6	2.0	2.7	4.1	[8.2]	
	1	0	32768 states	1.6	2.0	2.7	3.3	4.1	5.5	[8.2]	16.4	
		1	65536 states	3.3	4.1	5.5	6.6	[8.2]	[10.9]	16.4	32.8	
1	0	0	131072 states	6.6	[8.2]	[10.9]	[13.1]	16.4	21.8	32.8	65.5	
		1	262144 states	[13.1]	16.4	21.8	26.2	32.8	43.6	65.6	131.2	
	1	0	Reserved	—	—	—	—	—	—	—	—	µs
		1	16 states	0.8	1.0	1.3	1.6	2.0	2.7	4.0	8.0	

Legend:

[ ]: Recommended time setting

—: Don't care

Using an External Clock: Any value can be set. Normally, use of the minimum time is recommended.

21.6.4 Software Standby Mode Application Example

Figure 21.3 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software standby mode is cleared at the rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.

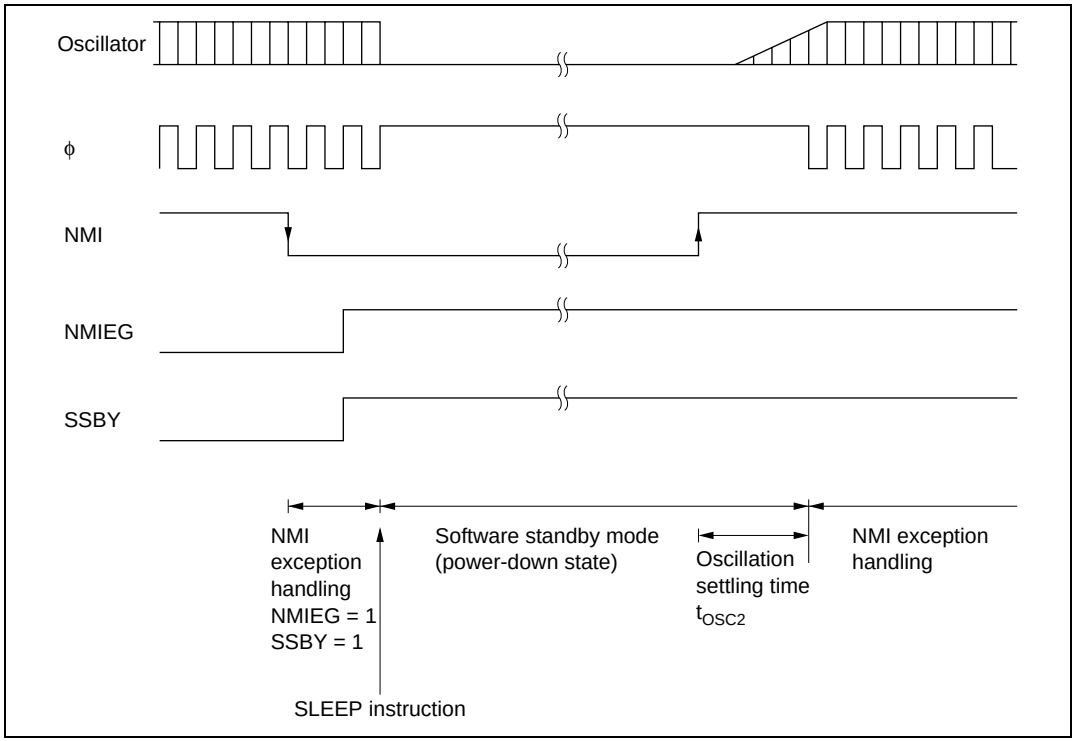


Figure 21.3 Software Standby Mode Application Example

21.6.5 Usage Note

In software standby mode, I/O port states are retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

Current dissipation increases while waiting for oscillation to settle.

21.7 Hardware Standby Mode

21.7.1 Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power dissipation. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD_1 and MD_0, $\overline{\text{TEST}}$) while the chip is in hardware standby mode.

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillation settles (at least 8 ms—the oscillation settling time—when using a crystal oscillator). When the $\overline{\text{RES}}$ pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

21.7.2 Hardware Standby Mode Timing

Figure 21.4 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high, waiting for the oscillation settling time, then changing the $\overline{\text{RES}}$ pin from low to high.

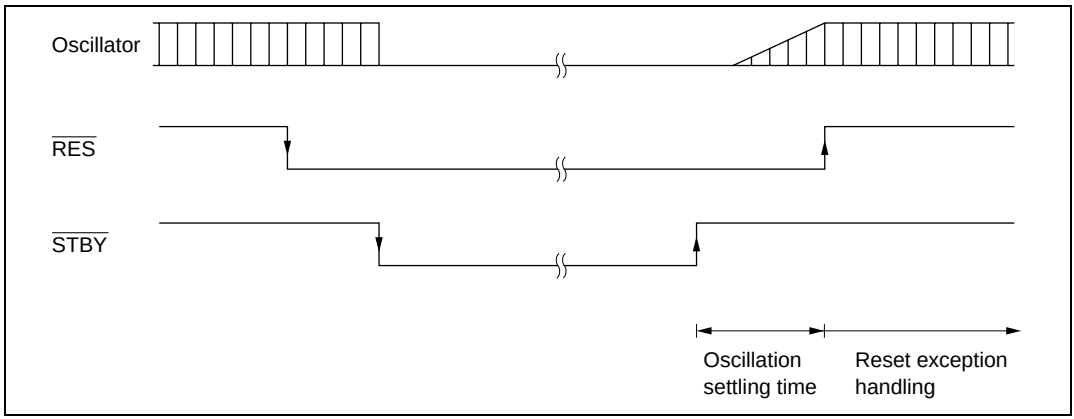


Figure 21.4 Hardware Standby Mode Timing

Section 22 Electrical Characteristics

22.1 Absolute Maximum Ratings

Table 22.1 lists the absolute maximum ratings.

Table 22.1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Program voltage	V_{PP}	–0.3 to +13.5	V
Bus driver power supply voltage (H8/3567U Group only)	DrV_{CC}	–0.3 to +4.3	V
Input voltage (except port 7)	V_{in}	–0.3 to V_{CC} + 0.3	V
Input voltage (port 7)	V_{in}	–0.3 to AV_{CC} + 0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} + 0.3	V
Operating temperature	T_{opr}	–20 to +75	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum ratings are exceeded.

22.2 DC Characteristics

Table 22.2 lists the DC characteristics. Table 22.3 lists the permissible output currents.

Table 22.2 DC Characteristics

Conditions: V_{CC} = 5.0 V ±10%, AV_{CC}*1 = 5.0 V ±10%, V_{SS} = AV_{SS}*1 = 0 V,
T_a = –20 to +75°C

Item			Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	$P6_7$ to $P6_0$*2, $\overline{IRQ}_2$ to $\overline{IRQ}_0$*3	(1)	V_T^-	1.0	—	—	V	
			V_T^+	—	—	$V_{CC} \times 0.7$	V	
			$V_T^+ - V_T^-$	0.4	—	—	V	
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, MD_1*7, MD_0*7, $\overline{TEST}$*8	(2)	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	EXTAL			$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Port 7			2.0	—	AV_{CC} + 0.3	V	
	Input pins except (1) and (2) above			2.0	—	V_{CC} + 0.3	V	
Input low voltage	$\overline{RES}$, $\overline{STBY}$, MD_1*7, MD_0*7, $\overline{TEST}$*8	(3)	V_{IL}	–0.3	—	0.5	V	
	NMI, EXTAL, input pins except (1) and (3) above			–0.3	—	0.8	V	
Output high voltage	All output pins (except $P4_7$, and $P5_2$)		V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
				3.5	—	—	V	I_{OH} = –1 mA
	$P4_7$, $P5_2$*4			2.0	—	—	V	I_{OH} = –200 µA
Output low voltage	All output pins		V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
Input leakage current	$\overline{RES}$		$\lvert I_{in} \rvert$	—	—	10.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
	$\overline{STBY}$, NMI, MD_1*7, MD0*7, $\overline{TEST}$*8			—	—	1.0	µA	
	Port 7			—	—	1.0	µA	V_{in} = 0.5 to AV_{CC} – 0.5 V

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Three-state leakage current (off state)	Ports 1 to 6	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1 to 3*7	$-I_P$	30	—	300	µA	V_{in} = 0 V
Input capacitance	$\overline{RES}$ (4)	C_{in}	—	—	80	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
	NMI		—	—	50	pF	
	$P5_2$, $P4_7$, $P2_4$*7, $P2_3$*7, $P1_7$, $P1_6$, $\overline{TEST}$*8		—	—	20	pF	
	Input pins except (4) above		—	—	15	pF	
Current dissipation*5	Normal operation (with on-chip USB)	I_{CC}	—	80	100	mA	f = 20 MHz
	Normal operation (other than the above)		—	60	80	mA	
	Sleep mode (with on-chip USB)		—	60	80	mA	
	Sleep mode (other than the above)		—	45	63	mA	
	Standby mode*6		—	0.2	5.0	µA	$T_a \leq 50°C$
			—	—	20.0	µA	$50°C < T_a$
Analog power supply current	During A/D conversion	AI_{CC}	—	1.5	3.0	mA	
	Idle		—	0.01	5.0	µA	AV_{CC} = 2.0 V to 5.5 V
Analog power supply voltage*1		AV_{CC}	4.5	—	5.5	V	Operating
			2.0	—	5.5	V	Idle/not used
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. Do not leave the AV_{CC}, and AV_{SS} pins open even if the A/D converter is not used.
Even if the A/D converter is not used, apply a value in the range 2.0 V to 5.5 V to AV_{CC} by connection to the power supply (V_{CC}), or some other method.
2. $P6_7$ to $P6_0$ include supporting module inputs multiplexed on those pins.
3. $\overline{IRQ_2}$ includes the $\overline{ADTRG}$ signal multiplexed on that pin.

4. $P5_2/SCK_0/SCL_0$ and $P4_7/SDA_0$ are NMOS push-pull outputs.
   An external pull-up resistor is necessary to provide high-level output from SCL_0 and SDA_0 (ICE = 1).
   $P5_2/SCK_0$ and $P4_7$ (ICE = 0) high levels are driven by NMOS.
5. Current dissipation values are for V_{IH} min = V_{CC} – 0.2 V and V_{IL} max = 0.2 V with all output pins unloaded and the on-chip pull-up MOSs in the off state.
6. The values are for $V_{RAM} \leq V_{CC} < 4.5$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.
7. In the H8/3577
8. In the H8/3567

Table 22.3 Permissible Output Currents

Conditions: V_{CC} = 4.0 to 5.5 V, AV_{CC} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = 0 V, Ta = –20 to +75°C

Item		Symbol	Min	Typ	Max	Unit
Permissible output low current (per pin)	SCL_1, SCL_0, SDA_1, SDA_0	I_{OL}	—	—	20	mA
	Other output pins		—	—	2	mA
Permissible output low current (total)	Total of all output pins, including the above	$\sum I_{OL}$	—	—	120	mA
Permissible output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	mA
Permissible output high current (total)	Total of all output pins	$\sum -I_{OH}$	—	—	40	mA

Notes: 1. To protect chip reliability, do not exceed the output current values in table 22.3.
2. When driving a Darlington pair or LED, always insert a current-limiting resistor in the output line, as show in figure 22.1.

Table 22.4 Bus Drive Characteristics

Conditions: V_{CC} = 4.5 to 5.5 V, V_{SS} = 0 V, Ta = –20 to +75°C

Applicable Pins: SCL1, SCL0, SDA1, SDA0 (bus drive function selected)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	V_T^-	$V_{CC} \times 0.3$	—	—	V	
	V_T^+	—	—	$V_{CC} \times 0.7$		
	$V_T^+ - V_T^-$	$V_{CC} \times 0.05$	—	—		
Input high voltage	V_{IH}	$V_{CC} \times 0.7$	—	V_{CC} + 0.5	V	
Input low voltage	V_{IL}	–0.5	—	$V_{CC} \times 0.3$	V	
Output low voltage	V_{OL}	—	—	0.8	V	I_{OL} = 16 mA
		—	—	0.5		I_{OL} = 8 mA
		—	—	0.4		I_{OL} = 3 mA
Input capacitance	C_{in}	—	—	20	pF	V_{in} = 0 V, f = 1 MHz, T_a = 25°C
Three-state leakage current (off state)	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5 V
SCL, SDA output fall time	t_{Of}	20 + 0.1Cb	—	250	ns	

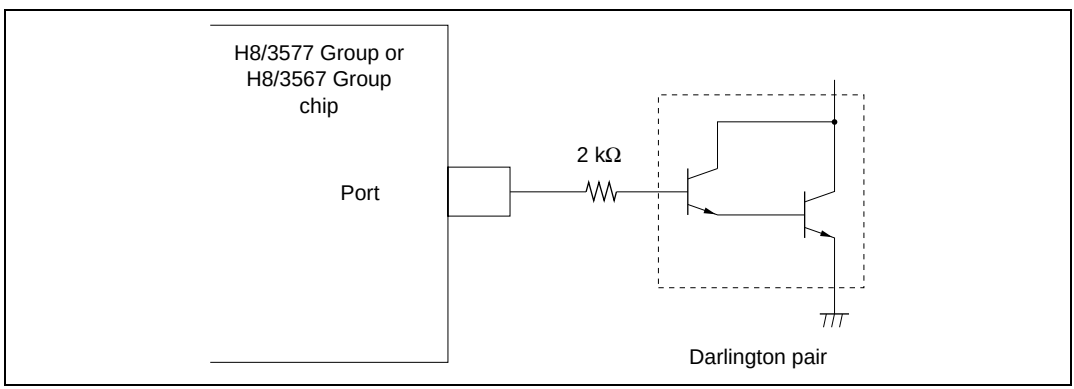


Figure 22.1 Darlington Pair Drive Circuit (Example)

22.3 AC Characteristics

Figure 22.2 shows the test conditions for the AC characteristics.

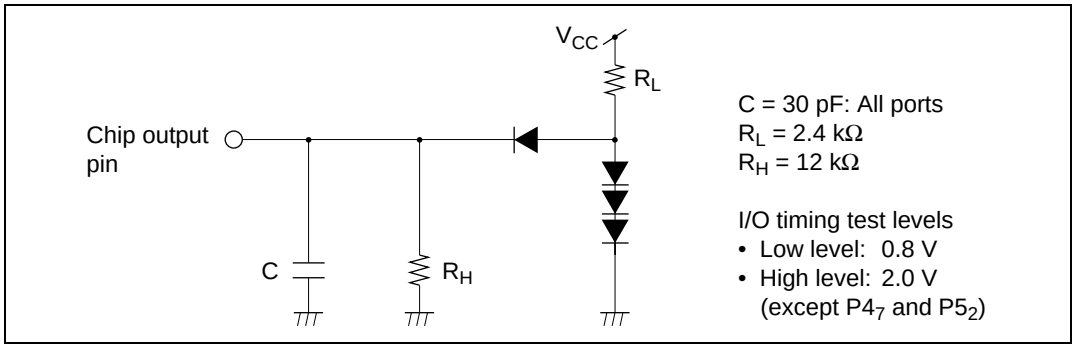


Figure 22.2 Output Load Circuit

22.3.1 Clock Timing

Table 22.5 shows the clock timing. The clock timing specified here covers clock (φ) output and clock pulse generator (crystal) and external clock input (EXTAL pin) oscillation settling times. For details of external clock input (EXTAL pin) timing, see section 20, Clock Pulse Generator.

Table 22.5 Clock Timing

Condition A: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, φ = 2 MHz to maximum operating frequency, T_a = –20 to +75°C

		Condition A			
		20 MHz			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
Clock cycle time	t_{cyc}	50	500	ns	Figure 22.3
Clock high pulse width	t_{CH}	17	—	ns	
Clock low pulse width	t_{CL}	17	—	ns	
Clock rise time	t_{Cr}	—	8	ns	
Clock fall time	t_{Cf}	—	8	ns	
Oscillation settling time at reset (crystal)	t_{OSC1}	10	—	ms	Figure 22.4 Figure 22.5
Oscillation settling time in software standby (crystal)	t_{OSC2}	8	—	ms	
External clock output stabilization delay time	t_{DEXT}	500	—	µs	

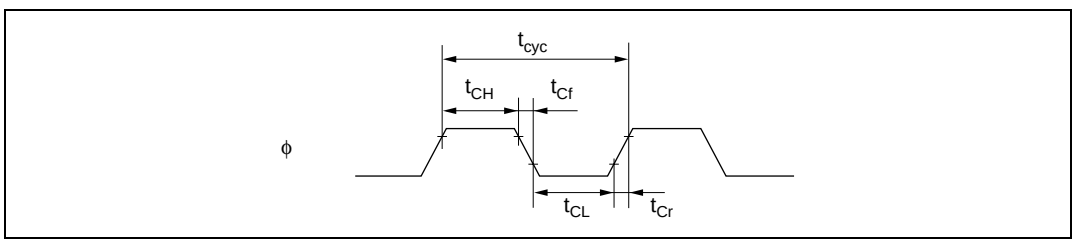


Figure 22.3 System Clock Timing

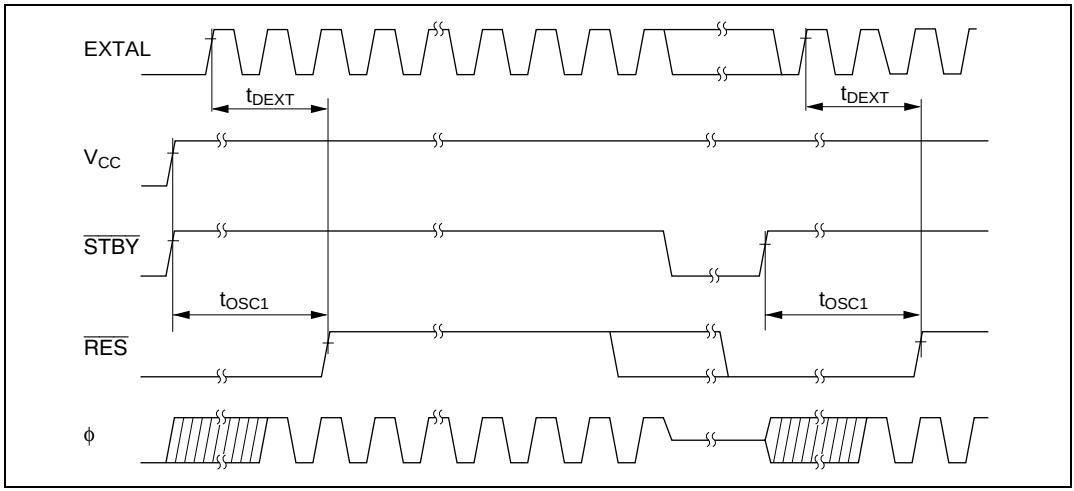


Figure 22.4 Oscillation Settling Timing

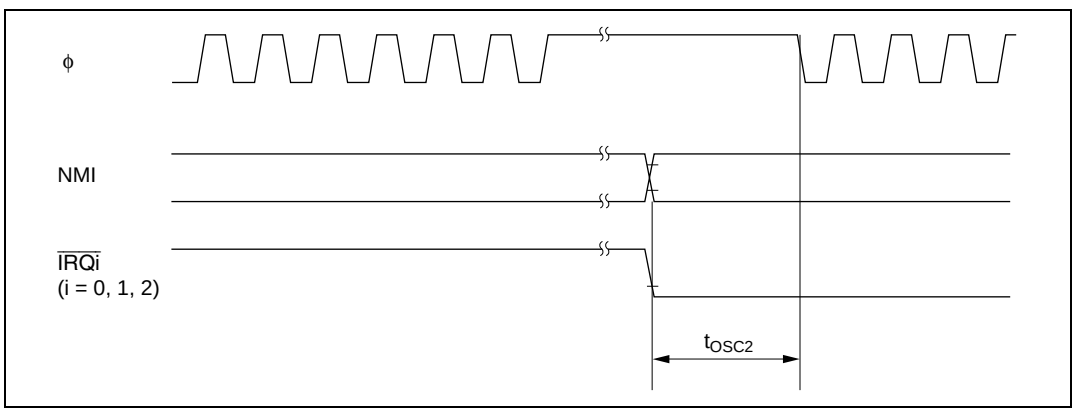


Figure 22.5 Oscillation Setting Timing (Exiting Software Standby Mode)

22.3.2 Control Signal Timing

Table 22.6 shows the control signal timing.

Table 22.6 Control Signal Timing

Condition A: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, ϕ = 2 MHz to maximum operating frequency, T_a = –20 to +75°C

		Condition A			
		20 MHz			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 22.6
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time (NMI)	t_{NMIS}	150	—	ns	Figure 22.7
NMI hold time (NMI)	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
IRQ setup time ($\overline{IRQ2}$ to $\overline{IRQ0}$)	t_{IRQS}	150	—	ns	
IRQ hold time ($\overline{IRQ2}$ to $\overline{IRQ0}$)	t_{IRQH}	10	—	ns	
IRQ pulse width ($\overline{IRQ2}$ to $\overline{IRQ0}$) (exiting software standby mode)	t_{IRQW}	200	—	ns	

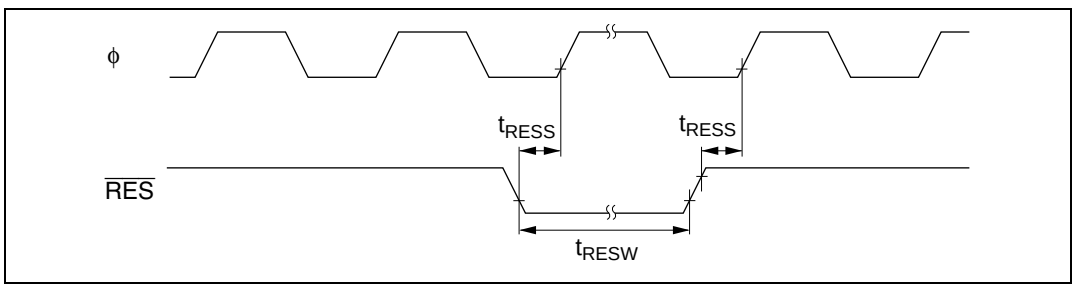


Figure 22.6 Reset Input Timing

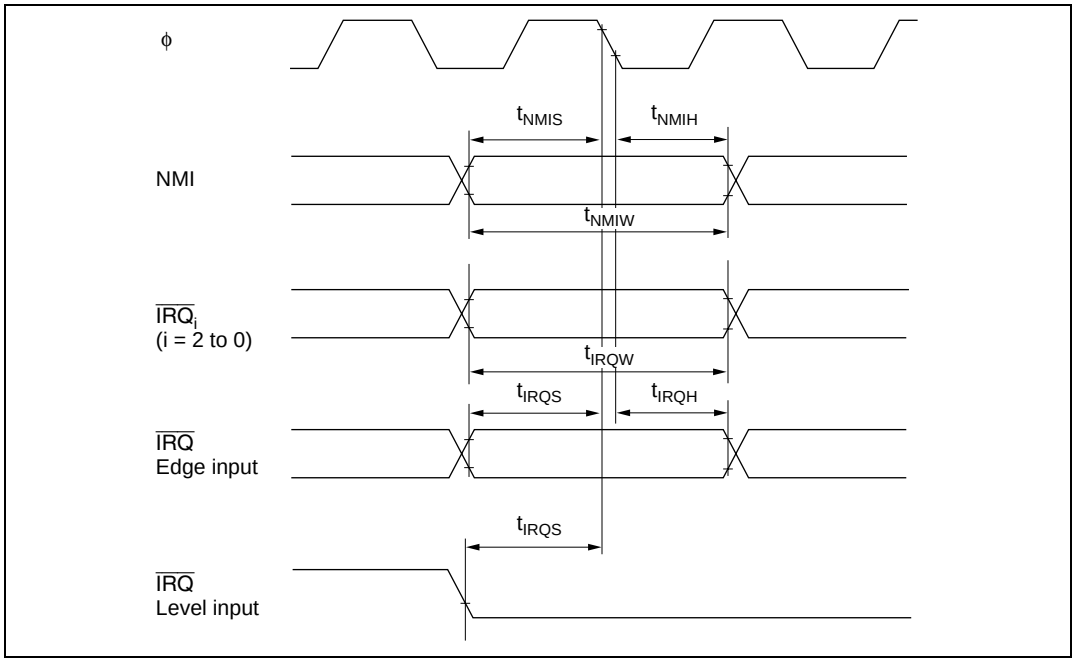


Figure 22.7 Interrupt Input Timing

22.3.3 Timing of On-Chip Supporting Modules

Tables 22.7 and 22.8 show the on-chip supporting module timing.

Table 22.7 Timing of On-Chip Supporting Modules

Condition A: V_{CC} = 5.0 V ±10%, V_{SS} = 0 V, φ = 2 MHz to maximum operating frequency, T_a = –20 to +75°C

				Condition A			
				20 MHz			
Item			**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
I/O ports	Output data delay time		t_{PWDA}, t_{PWDB}	—	50	ns	Figure 22.8 (1)
	Input data setup time		t_{PRSA}, t_{PRSB}	30	—		Figure 22.8 (2)
	Input data hold time		t_{PRHA}, t_{PRHB}	30	—		
FRT	Timer output delay time		t_{FTOD}	—	50	ns	Figure 22.9
	Timer input setup time		t_{FTIS}	30	—		
	Timer clock input setup time		t_{FTCS}	30	—		Figure 22.10
	Timer clock pulse width	Single edge	t_{FTCWH}	1.5	—	t_{cyc}	
		Both edges	t_{FTCWL}	2.5	—		
TMR	Timer output delay time		t_{TMOD}	—	50	ns	Figure 22.11
	Timer reset input setup time		t_{TMRS}	30	—		Figure 22.13
	Timer clock input setup time		t_{TMCS}	30	—		Figure 22.12
	Timer clock pulse width	Single edge	t_{TMCWH}	1.5	—	t_{cyc}	
		Both edges	t_{TMCWL}	2.5	—		
PWM, PWMX	Pulse output delay time		t_{PWOD}	—	50	ns	Figure 22.14
SCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{cyc}	Figure 22.15
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time (synchronous)		t_{TXD}	—	50	ns	Figure 22.16
	Receive data setup time (synchronous)		t_{RXS}	50	—		

Item		Symbol	Condition A 20 MHz Min	Max	Unit	Test Conditions
SCI	Receive data hold time (synchronous)	t_{RXH}	50	—	ns	Figure 22.16
A/D converter	Trigger input setup time	t_{TRGS}	30	—	ns	Figure 22.17

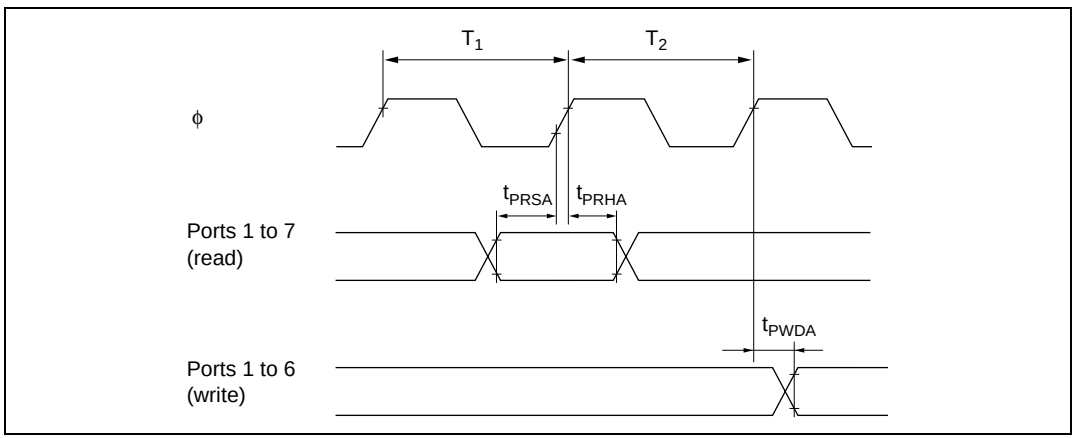


Figure 22.8 (1) I/O Port Input/Output Timing

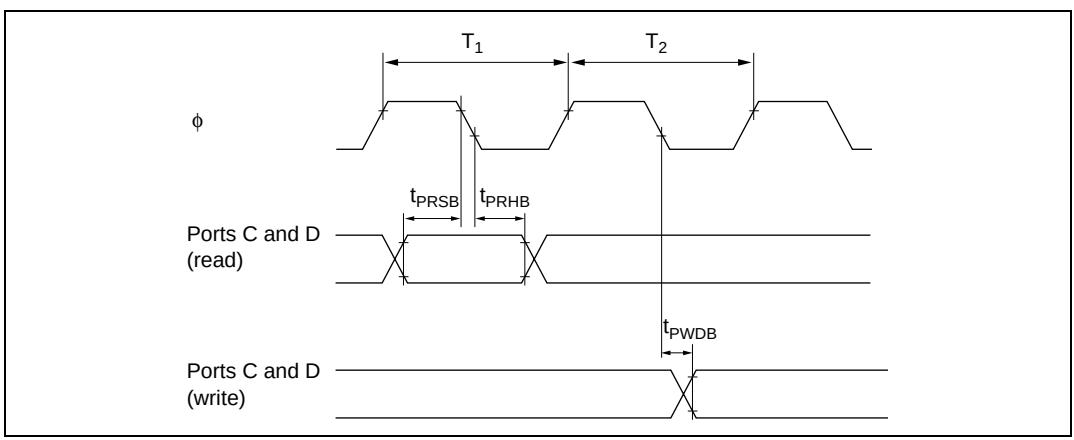


Figure 22.8 (2) I/O Port Input/Output Timing (USB On-Chip Version)

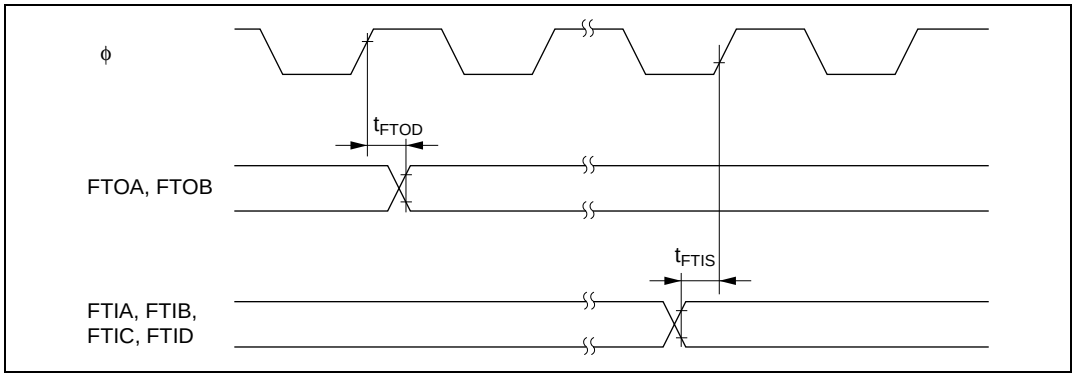


Figure 22.9 FRT Input/Output Timing

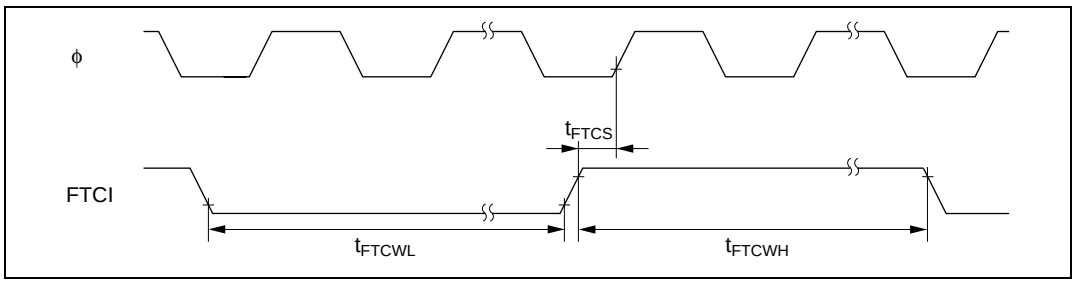


Figure 22.10 FRT Clock Input Timing

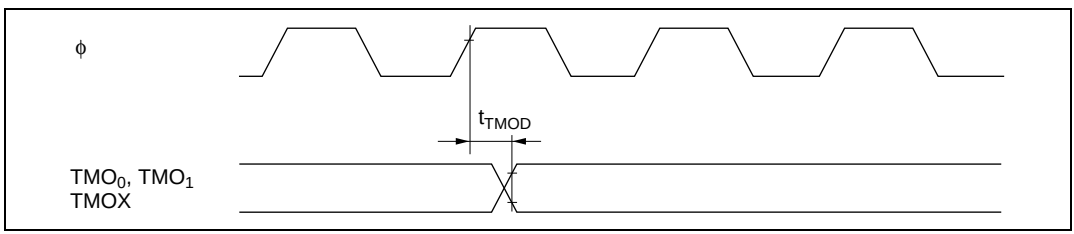


Figure 22.11 8-Bit Timer Output Timing

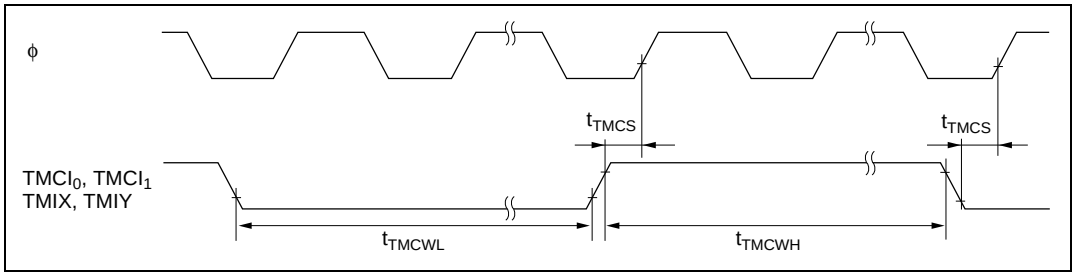


Figure 22.12 8-Bit Timer Clock Input Timing

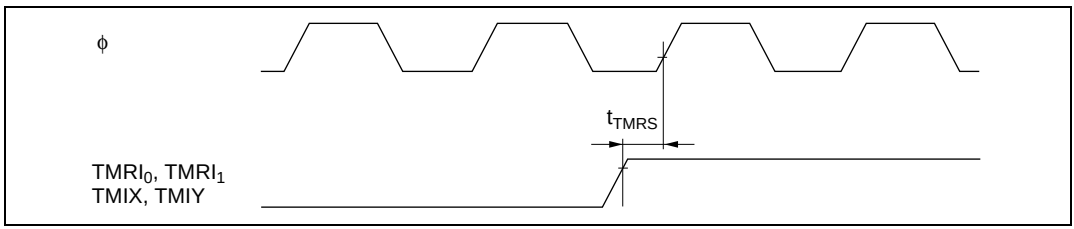


Figure 22.13 8-Bit Timer Reset Input Timing

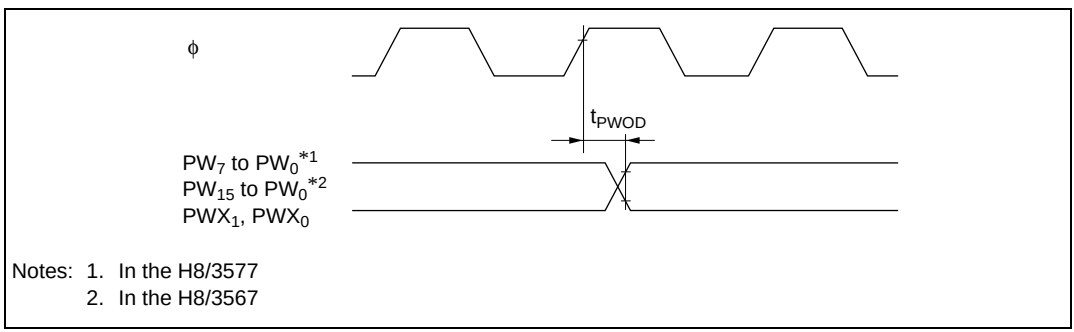


Figure 22.14 PWM, PWMX Output Timing

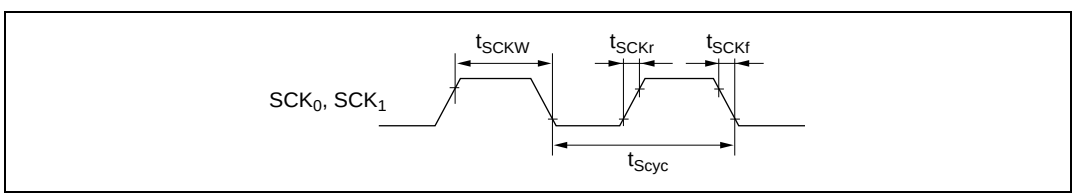


Figure 22.15 SCK Clock Input Timing

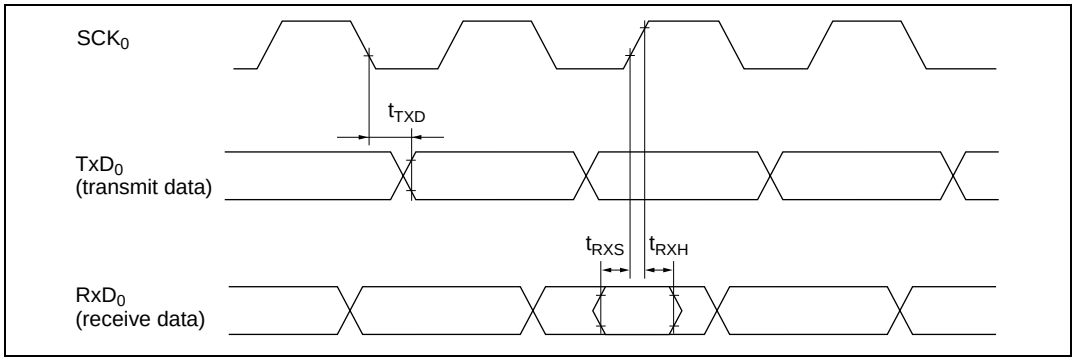


Figure 22.16 SCI Input/Output Timing (Synchronous Mode)

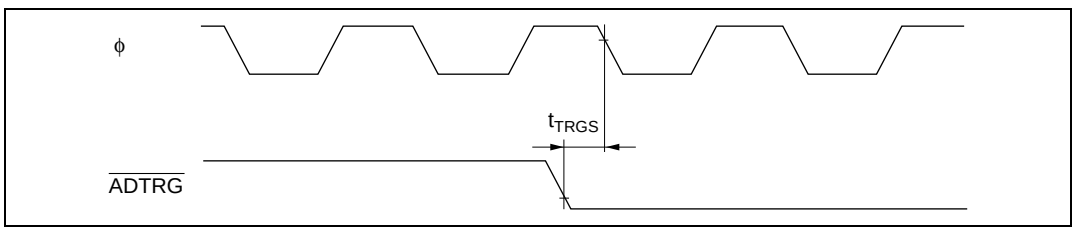


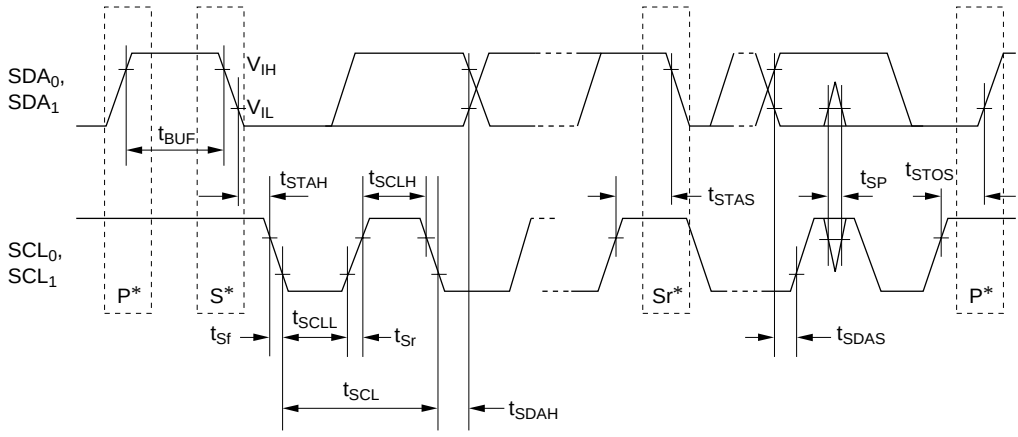
Figure 22.17 A/D Converter External Trigger Input Timing

Table 22.8 I^2C Bus Timing

Conditions: V_{CC} = 4.5 V to 5.5 V, V_{SS} = 0 V, ϕ = 5 MHz to maximum operating frequency, T_a = –20 to +75°C

Item	Symbol	Min	Typ	Max	Unit	Test Conditions	Notes
SCL clock cycle time	t_{SCL}	12	—	—	t_{cyc}		Figure 22.18
SCL clock high pulse width	t_{SCLH}	3	—	—	t_{cyc}		
SCL clock low pulse width	t_{SCLL}	5	—	—	t_{cyc}		
SCL, SDA input rise time	t_{Sr}	—	—	7.5*	t_{cyc}		
SCL, SDA input fall time	t_{Sf}	—	—	300	ns		
SCL, SDA input spike pulse elimination time	t_{SP}	—	—	1	t_{cyc}		
SDA input bus free time	t_{BUF}	5	—	—	t_{cyc}		
Start condition input hold time	t_{STAH}	3	—	—	t_{cyc}		
Retransmission start condition input setup time	t_{STAS}	3	—	—	t_{cyc}		
Stop condition input setup time	t_{STOS}	3	—	—	t_{cyc}		
Data input setup time	t_{SDAS}	0.5	—	—	t_{cyc}		
Data input hold time	t_{SDAH}	0	—	—	ns		
SCL, SDA capacitive load	C_b	—	—	400	pF		

Note: * $17.5t_{cyc}$ can be set according to the clock selected for use by the I^2C module. For details, see section 16.4, Usage Notes.



Note: * S, P, and Sr indicate the following conditions.

S: Start condition
P: Stop condition
Sr: Retransmission start condition

Figure 22.18 I²C Bus Interface Input/Output Timing

22.4 A/D Conversion Characteristics

Table 22.9 lists the A/D conversion characteristics.

Table 22.9 A/D Conversion Characteristics
(AN_7 to AN_0 Input[*1]: 134/266-State Conversion)

Condition A: V_{CC} = 5.0 V ±10%, AV_{CC} = 4.5 V to 5.5 V
$V_{SS} = AV_{SS}$ = 0 V[*2], ϕ = 2 MHz to maximum operating frequency,
T_a = –20 to +75°C

	Condition A			
	20 MHz			
Item	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	Bits
Conversion time (single mode)[*5]	—	—	6.7	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	10[*3]	kΩ
			5[*4]	
Nonlinearity error	—	—	±3.0	LSB
Offset error	—	—	±3.5	LSB
Full-scale error	—	—	±3.5	LSB
Quantization error	—	—	±0.5	LSB
Absolute accuracy			±4.0	LSB

Notes: 1. In the H8/3577
(AN_3 to AN_0 in the H8/3567)
2. The voltage applied to the ANn analog input pins during A/D conversion must be in the range $AV_{SS} \le ANn \le AV_{CC}$ (where n = 0 to 3). For the relationship between AV_{CC}/AV_{SS} and V_{CC}/V_{SS}, set $AV_{SS} = V_{SS}$. The AV_{CC} and AV_{SS} pins must not be left open when the A/D converter is not used.
3. When conversion time ≥ 11. 17 µs (CKS = 1 and $\phi \le$ 12 MHz, or CKS = 0)
4. When conversion time < 11. 17 µs (CKS = 1 and ϕ > 12 Mhz)
5. Value when using the maximum operating frequency.

22.5 USB Function Pin Characteristics

Table 22.10 shows the USB function pin characteristics.

Table 22.10 DC Characteristics

Conditions: V_{CC} = 5.0 V ±10%, DrV_{CC} = 3.3 V ± 0.3 V, DrV_{SS} = V_{SS} = 0 V, Ta = –20°C to +75°C

Pin Functions: Transceiver input/output (USD+, USD–, DS2D+, DS2D–, DS3D+, DS3D–, DS4D+, DS4D–, DS5D+, DS5D–), ports C and D, $\overline{ENP_2}$ to $\overline{ENP_5}$, $\overline{OCP_2}$ to $\overline{OCP_5}$, EXTAL12, XTAL12

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Differential input sensitivity		V_{DI}	0.2	—	—	V	| (D+) – (D–) |
Differential common mode range		V_{CM}	0.8	—	2.5	V	Including V_{DI}
Schmitt trigger input voltages	$\overline{OCP_2}$ to $\overline{OCP_5}$	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input high*1 voltage	EXTAL12	V_{IH}	$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
	Port D		2.0	—	DrV_{CC} + 0.3	V	
	Other than the above		2.0	—	V_{CC} + 0.3	V	
Input low*1 voltage	EXTAL12	V_{IL}	–0.3	—	$V_{CC} \times 0.2$	V	
	Other than the above		–0.3	—	0.8	V	
Output high voltage	Transceiver	V_{OH}	2.8	—	3.6	V	RL = 15 kΩ connected between pin and GND
	Port D		DrV_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			DrV_{CC} – 1.0	—	—	V	I_{OH} = –1 mA
	Other than the above		V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1 mA

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Output low voltage	Transceiver	V_{OL}	—	—	0.3	V	RL = 1.5 kΩ connected between pin and power supply
	Other than the above		—	—	0.4	V	I_{OL} = 1.6 mA
Output resistance		Z_{DRV}	28	—	44	Ω	
Input pin capacitance		C_{IN}	—	—	35	pF	Between pin and GND
Three-state leakage current		I_{LO}	—	—	1.0	µA	0.5 V < V_{in} < DrV_{CC} – 0.5 V 0.5 V < V_{in} < V_{CC} – 0.5 V*2
DrV_{CC} current dissipation	Normal operation	DI_{CC}	—	5	10	mA	
	Standby mode		—	0.2	5.0	µA	

Notes: 1. Excluding transceiver input/output (USD+, USD–, DS2D+, DS2D–, DS3D+, DS3D–, DS4D+, DS4D–, DS5D+, DS5D–)

2. Upper row applies to transceiver input/output and port D, and lower row to other pins.

Table 22.11 AC Characteristics

Conditions: V_{CC} = 5.0 V ±10%, DrV_{CC} = 3.3 V ± 0.3 V, DrV_{SS} = V_{SS} = 0 V, Ta = –20°C to +75°C

Pin Functions: Transceiver input/output (USD+, USD–, DS2D+, DS2D–, DS3D+, DS3D–, DS4D+, DS4D–, DS5D+, DS5D–), ports C and D, $\overline{ENP_2}$ to $\overline{ENP_5}$, $\overline{OCP_2}$ to $\overline{OCP_5}$, EXTAL12, XTAL12

Item		Symbol	Min	Max	Unit	Figure	Notes
Transceiver full speed	Rise time	t_{FR}	4	20	ns	Figure 22.19	
	Fall time	t_{FF}	4	20			
	Differential signal time difference	t_{FRFM}	90.0	111.11	%		t_{FR}/t_{FF}
Transceiver low speed	Rise time	t_{LR}	75	300	ns	Figure 22.19	
	Fall time	t_{LF}	75	300			
	Differential signal time difference	t_{LRFM}	80.0	125	%		t_{LR}/t_{LF}
Transceiver output signal crossing voltage		V_{CRS}	1.3	2.0	V		
Ports C and D	Output data delay time	t_{PWDB}	—	50	ns	Figure 22.8 (2)	
	Input data setup time	t_{PRSB}	30	—			
	Input data hold time	t_{PRHB}	30	—			
USB clock oscillation settling time (crystal)		t_{OSCU}	10	—	ms		

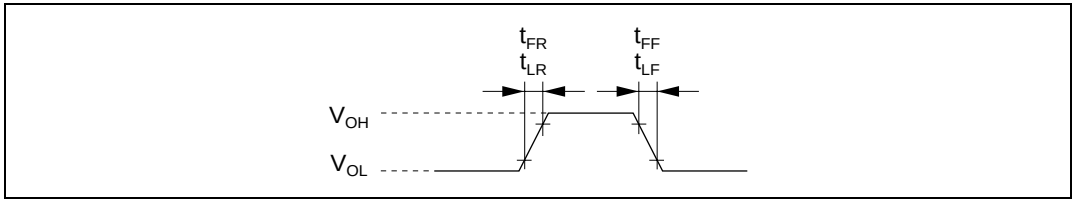


Figure 22.19 Transceiver Output Timing

22.6 Usage Notes

ZTAT Version and Mask ROM Version: The ZTAT and mask ROM versions satisfy the electrical characteristics shown in this manual, but actual electrical characteristic values, operating margins, noise margins, and other properties may vary due to differences in manufacturing process, on-chip ROM, layout patterns, and so on.

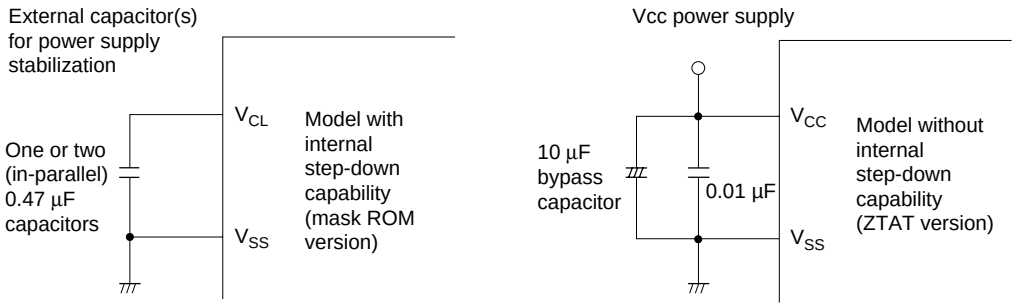
When system evaluation testing is carried out using the ZTAT version, the same evaluation testing should also be conducted for the mask ROM version when changing over to that version.

Models with Internal Step-Down Circuit: H8/3577, H8/3567, and H8/3567U mask ROM models (HD6433577, HD6433574, HD6433567, HD6433564-20, HD6433564-10, HD6433567U, and HD6433564U) incorporate an internal step-down circuit to lower the MCU's internal power supply voltage to the optimum level automatically.

One or two (in-parallel) 0.47 µF internal voltage stabilization capacitors must be connected between the internal step-down pin (V_{CL}) and the V_{SS} pin.

The method of connecting the external capacitor(s) is shown in figure 22.20. Do not apply a voltage exceeding 3.6 V to the V_{CL} pin.

When switching from a ZTAT version with no internal step-down capability to a mask ROM version with the step-down facility, the differences in the circuitry before and after the changeover must be taken into consideration when designing the board pattern.



Do not connect the V_{CC} power supply to the V_{CL} pin of a model with internal step-down capability. (Connect the V_{CC} power supply to other V_{CC} pins as usual.)
A power supply stabilization capacitor must be connected to the V_{CL} pin. Use one or two (in-parallel) 0.47 µF laminated ceramic capacitors, placed close to the pin.

Models with internal step-down capability:
HD6433577, HD6433574, HD6433567, HD6433564-20, HD6433564-10, HD6433567U, HD6433564U

Models with no internal step-down capability have a V_{CC} pin (V_{CC} power supply pin) in the pin position occupied by the V_{CL} pin in internal step-down models.
It is recommended that a bypass capacitor be connected to the power supply pins. (Values are for reference.)

Models without internal step-down capability:
HD6473577, HD6473567, HD6473567U

Figure 22.20 Method of Connecting V_{CL} Capacitor(s) to Mask ROM Model with Internal Step-Down Capability, and Differences between Models with and without Internal Step-Down Capability

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)

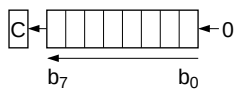
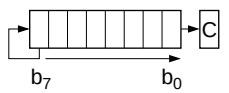
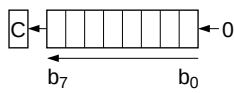
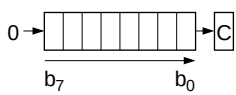
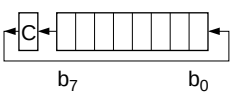
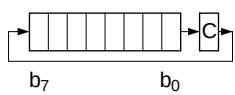
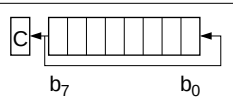
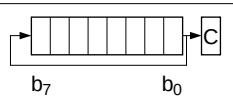
Condition Code Notation

↕	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									—	—	↕	↕	0	—	2
MOV.B Rs, Rd	B	Rs8 → Rd8		2								—	—	↕	↕	0	—	2
MOV.B @Rs, Rd	B	@Rs16 → Rd8			2							—	—	↕	↕	0	—	4
MOV.B @(d:16, Rs), Rd	B	@(d:16, Rs16) → Rd8				4						—	—	↕	↕	0	—	6
MOV.B @Rs+, Rd	B	@Rs16 → Rd8 Rs16+1 → Rs16					2					—	—	↕	↕	0	—	6
MOV.B @aa:8, Rd	B	@aa:8 → Rd8						2				—	—	↕	↕	0	—	4
MOV.B @aa:16, Rd	B	@aa:16 → Rd8						4				—	—	↕	↕	0	—	6
MOV.B Rs, @Rd	B	Rs8 → @Rd16			2							—	—	↕	↕	0	—	4
MOV.B Rs, @(d:16, Rd)	B	Rs8 → @(d:16, Rd16)				4						—	—	↕	↕	0	—	6
MOV.B Rs, @–Rd	B	Rd16–1 → Rd16 Rs8 → @Rd16					2					—	—	↕	↕	0	—	6
MOV.B Rs, @aa:8	B	Rs8 → @aa:8						2				—	—	↕	↕	0	—	4
MOV.B Rs, @aa:16	B	Rs8 → @aa:16						4				—	—	↕	↕	0	—	6
MOV.W #xx:16, Rd	W	#xx:16 → Rd16	4									—	—	↕	↕	0	—	4
MOV.W Rs, Rd	W	Rs16 → Rd16		2								—	—	↕	↕	0	—	2
MOV.W @Rs, Rd	W	@Rs16 → Rd16			2							—	—	↕	↕	0	—	4
MOV.W @(d:16, Rs), Rd	W	@(d:16, Rs16) → Rd16				4						—	—	↕	↕	0	—	6
MOV.W @Rs+, Rd	W	@Rs16 → Rd16 Rs16+2 → Rs16					2					—	—	↕	↕	0	—	6
MOV.W @aa:16, Rd	W	@aa:16 → Rd16						4				—	—	↕	↕	0	—	6
MOV.W Rs, @Rd	W	Rs16 → @Rd16			2							—	—	↕	↕	0	—	4
MOV.W Rs, @(d:16, Rd)	W	Rs16 → @(d:16, Rd16)				4						—	—	↕	↕	0	—	6
MOV.W Rs, @–Rd	W	Rd16–2 → Rd16 Rs16 → @Rd16					2					—	—	↕	↕	0	—	6
MOV.W Rs, @aa:16	W	Rs16 → @aa:16						4				—	—	↕	↕	0	—	6
POP Rd	W	@SP → Rd16 SP+2 → SP					2					—	—	↕	↕	0	—	6
PUSH Rs	W	SP–2 → SP Rs16 → @SP					2					—	—	↕	↕	0	—	6

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																
MOVTPE Rs, @aa:16	B	Not supported																
ADD.B #xx:8, Rd	B	Rd8+#xx:8 → Rd8	2									—	↕	↕	↕	↕	↕	2
ADD.B Rs, Rd	B	Rd8+Rs8 → Rd8		2								—	↕	↕	↕	↕	↕	2
ADD.W Rs, Rd	W	Rd16+Rs16 → Rd16		2								—	(1)	↕	↕	↕	↕	2
ADDX.B #xx:8, Rd	B	Rd8+#xx:8 +C → Rd8	2									—	↕	↕	(2)	↕	↕	2
ADDX.B Rs, Rd	B	Rd8+Rs8 +C → Rd8		2								—	↕	↕	(2)	↕	↕	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								—	—	↕	↕	↕	—	2
DAA.B Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*	(3)	2
SUB.B Rs, Rd	B	Rd8–Rs8 → Rd8		2								—	↕	↕	↕	↕	↕	2
SUB.W Rs, Rd	W	Rd16–Rs16 → Rd16		2								—	(1)	↕	↕	↕	↕	2
SUBX.B #xx:8, Rd	B	Rd8–#xx:8 –C → Rd8	2									—	↕	↕	(2)	↕	↕	2
SUBX.B Rs, Rd	B	Rd8–Rs8 –C → Rd8		2								—	↕	↕	(2)	↕	↕	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								—	—	↕	↕	↕	—	2
DAS.B Rd	B	Rd8 decimal adjust → Rd8		2								—	*	↕	↕	*	—	2
NEG.B Rd	B	0–Rd8 → Rd8		2								—	↕	↕	↕	↕	↕	2
CMP.B #xx:8, Rd	B	Rd8–#xx:8	2									—	↕	↕	↕	↕	↕	2
CMP.B Rs, Rd	B	Rd8–Rs8		2								—	↕	↕	↕	↕	↕	2
CMP.W Rs, Rd	W	Rd16–Rs16		2								—	(1)	↕	↕	↕	↕	2
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									—	—	↕	↕	0	—	2
AND.B Rs, Rd	B	Rd8∧Rs8 → Rd8		2								—	—	↕	↕	0	—	2

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	
OR.B #xx:8, Rd	B	Rd8∨#xx:8 → Rd8	2									—	—	↕	↕	0	—	2
OR.B Rs, Rd	B	Rd8∨Rs8 → Rd8		2								—	—	↕	↕	0	—	2
XOR.B #xx:8, Rd	B	Rd8⊕#xx:8 → Rd8	2									—	—	↕	↕	0	—	2
XOR.B Rs, Rd	B	Rd8⊕Rs8 → Rd8		2								—	—	↕	↕	0	—	2
NOT.B Rd	B	$\overline{\text{Rd8}}$ → Rd8		2								—	—	↕	↕	0	—	2
SHAL.B Rd	B			2								—	—	↕	↕	↕	↕	2
SHAR.B Rd	B			2								—	—	↕	↕	0	↕	2
SHLL.B Rd	B			2								—	—	↕	↕	0	↕	2
SHLR.B Rd	B			2								—	—	0	↕	0	↕	2
ROTXL.B Rd	B			2								—	—	↕	↕	0	↕	2
ROTXR.B Rd	B			2								—	—	↕	↕	0	↕	2
ROTL.B Rd	B			2								—	—	↕	↕	0	↕	2
ROTR.B Rd	B			2								—	—	↕	↕	0	↕	2

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	
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) ← ($\overline{\text{#xx:3 of Rd8}}$)		2								—	—	—	—	—	—	2
BNOT #xx:3, @Rd	B	(#xx:3 of @Rd16) ← ($\overline{\text{#xx:3 of @Rd16}}$)			4							—	—	—	—	—	—	8
BNOT #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← ($\overline{\text{#xx:3 of @aa:8}}$)						4				—	—	—	—	—	—	8
BNOT Rn, Rd	B	(Rn8 of Rd8) ← ($\overline{\text{Rn8 of Rd8}}$)		2								—	—	—	—	—	—	2
BNOT Rn, @Rd	B	(Rn8 of @Rd16) ← ($\overline{\text{Rn8 of @Rd16}}$)			4							—	—	—	—	—	—	8
BNOT Rn, @aa:8	B	(Rn8 of @aa:8) ← ($\overline{\text{Rn8 of @aa:8}}$)						4				—	—	—	—	—	—	8
BTST #xx:3, Rd	B	($\overline{\text{#xx:3 of Rd8}}$) → Z		2								—	—	—	↕	—	—	2
BTST #xx:3, @Rd	B	($\overline{\text{#xx:3 of @Rd16}}$) → Z			4							—	—	—	↕	—	—	6
BTST #xx:3, @aa:8	B	($\overline{\text{#xx:3 of @aa:8}}$) → Z						4				—	—	—	↕	—	—	6
BTST Rn, Rd	B	($\overline{\text{Rn8 of Rd8}}$) → Z		2								—	—	—	↕	—	—	2
BTST Rn, @Rd	B	($\overline{\text{Rn8 of @Rd16}}$) → Z			4							—	—	—	↕	—	—	6
BTST Rn, @aa:8	B	($\overline{\text{Rn8 of @aa:8}}$) → Z						4				—	—	—	↕	—	—	6

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	
BLD #xx:3, Rd	B	(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BLD #xx:3, @Rd	B	(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BLD #xx:3, @aa:8	B	(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6
BILD #xx:3, Rd	B	($\overline{\text{\#xx:3 of Rd8}}$) → C		2								—	—	—	—	—	↕	2
BILD #xx:3, @Rd	B	($\overline{\text{\#xx:3 of @Rd16}}$) → C			4							—	—	—	—	—	↕	6
BILD #xx:3, @aa:8	B	($\overline{\text{\#xx:3 of @aa:8}}$) → C						4				—	—	—	—	—	↕	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	$\overline{\text{C}}$ → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2
BIST #xx:3, @Rd	B	$\overline{\text{C}}$ → (#xx:3 of @Rd16)			4							—	—	—	—	—	—	8
BIST #xx:3, @aa:8	B	$\overline{\text{C}}$ → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8
BAND #xx:3, Rd	B	C∧(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BAND #xx:3, @Rd	B	C∧(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BAND #xx:3, @aa:8	B	C∧(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6
BIAND #xx:3, Rd	B	C∧($\overline{\text{\#xx:3 of Rd8}}$) → C		2								—	—	—	—	—	↕	2
BIAND #xx:3, @Rd	B	C∧($\overline{\text{\#xx:3 of @Rd16}}$) → C			4							—	—	—	—	—	↕	6
BIAND #xx:3, @aa:8	B	C∧($\overline{\text{\#xx:3 of @aa:8}}$) → C						4				—	—	—	—	—	↕	6
BOR #xx:3, Rd	B	C∨(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BOR #xx:3, @Rd	B	C∨(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BOR #xx:3, @aa:8	B	C∨(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6
BIOR #xx:3, Rd	B	C∨($\overline{\text{\#xx:3 of Rd8}}$) → C		2								—	—	—	—	—	↕	2
BIOR #xx:3, @Rd	B	C∨($\overline{\text{\#xx:3 of @Rd16}}$) → C			4							—	—	—	—	—	↕	6
BIOR #xx:3, @aa:8	B	C∨($\overline{\text{\#xx:3 of @aa:8}}$) → C						4				—	—	—	—	—	↕	6
BXOR #xx:3, Rd	B	C⊕(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BXOR #xx:3, @Rd	B	C⊕(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BXOR #xx:3, @aa:8	B	C⊕(#xx:3 of @aa:8) → C						4				—	—	—	—	—	↕	6
BIXOR #xx:3, Rd	B	C⊕($\overline{\text{\#xx:3 of Rd8}}$) → C		2								—	—	—	—	—	↕	2

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	
BIXOR #xx:3, @Rd	B	C⊕(#xx:3 of @Rd16) → C				4							—	—	—	—	—	↕	6
BIXOR #xx:3, @aa:8	B	C⊕(#xx:3 of @aa:8) → C							4				—	—	—	—	—	↕	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 is true then PC ← PC+d:8 else next;	C ∨ Z = 0							2			—	—	—	—	—	—	4
BLS d:8	—		C ∨ Z = 1							2			—	—	—	—	—	—	4
BCC d:8 (BHS d:8)	—		C = 0							2			—	—	—	—	—	—	4
BCS d:8 (BLO d:8)	—		C = 1							2			—	—	—	—	—	—	4
BNE d:8	—		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
JSR @Rn	—	SP–2 → SP PC → @SP PC ← Rn16				2							—	—	—	—	—	—	6
JSR @aa:16	—	SP–2 → SP PC → @SP PC ← aa:16							4				—	—	—	—	—	—	8

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 @@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	↕	↕	↕	↕	↕	↕	10
SLEEP	—	Transit to sleep mode.									2	—	—	—	—	—	—	2
LDC #xx:8, CCR	B	#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2
LDC Rs, CCR	B	Rs8 → CCR		2								↕	↕	↕	↕	↕	↕	2
STC CCR, Rd	B	CCR → Rd8		2								—	—	—	—	—	—	2
ANDC #xx:8, CCR	B	CCR∧#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2
ORC #xx:8, CCR	B	CCR∨#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2
XORC #xx:8, CCR	B	CCR⊕#xx:8 → CCR	2									↕	↕	↕	↕	↕	↕	2
NOP	—	PC ← PC+2									2	—	—	—	—	—	—	2
EEPMOV	—	EEPMOV	These cannot be used in this LSI.															(5)

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. For other cases, see section A.3, Number of Instruction Execution States.

(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 cleared to 0.

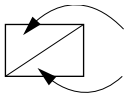
(5) These instructions are not supported by the H8/3577 Group and H8/3567 Group.

(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).



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	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BST / BIST	MOV*1							
7								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:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples: Mode 1, 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

Execution Status (Instruction Cycle)		Access Location		
		On-Chip Memory	On-Chip Reg. Field	External Memory
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

Note: 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		

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		

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	This cannot be used in these H8/3577 Group and H8/3567 Group.					
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					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
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		
	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	MOVTPE.Rs, @aa:16	Not supported					
MULXU	MULXU.B Rs, Rd	1					12
NEG	NEG.B Rd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
OR	OR.B #xx:8, Rd	1					
	OR.B Rs, Rd	1					
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
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 Registers

B.1 Addresses

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Bus Width
H'FDC0	UPRTCR	—	—	DSPSEL2	DSPSEL1	DSPSEL0	PCNMD2	PCNMD1	PCNMD0	USB	8
H'FDC1	UTESTR0										
H'FDC2	UTESTR1										
H'FDE1	EPDR2	D7	D6	D5	D4	D3	D2	D1	D0		
H'FDE2	FVSR2H	—	—	—	—	—	—	N9	N8		
H'FDE3	FVSR2L	N7	N6	N5	N4	N3	N2	N1	N0		
H'FDE4	EPSZR1	EP1SZ3	EP1SZ2	EP1SZ1	EP1SZ0	EP2SZ3	EP2SZ2	EP2SZ1	EP2SZ0		
H'FDE5	EPDR1	D7	D6	D5	D4	D3	D2	D1	D0		
H'FDE6	FVSR1H	—	—	—	—	—	—	N9	N8		
H'FDE7	FVSR1L	N7	N6	N5	N4	N3	N2	N1	N0		
H'FDE9	EPDR0O	D7	D6	D5	D4	D3	D2	D1	D0		
H'FDEA	FVSR0OH	—	—	—	—	—	—	N9	N8		
H'FDEB	FVSR0OL	N7	N6	N5	N4	N3	N2	N1	N0		
H'FDED	EPDR0I	D7	D6	D5	D4	D3	D2	D1	D0		
H'FDEE	FVSR0IH	—	—	—	—	—	—	N9	N8		
H'FDEF	FVSR0IL	N7	N6	N5	N4	N3	N2	N1	N0		
H'FDF0	PTTER	—	—	—	—	EP2TE	EP1TE	EP0ITE	—		
H'FDF1	USBIER	—	—	BRSTE	SOFE	SPNDE	TFE	TSE	SETUPE		
H'FDF2	USBIFR	TS	TF	—	BRSTF	SOFF	SPNDOF	SPNDIF	SETUPF		
H'FDF3	TSFR	—	—	—	—	EP2TS	EP1TS	EP0ITS	EP0OTS		
H'FDF4	TFFR	—	—	—	—	EP2TF	EP1TF	EP0ITF	EP0OTF		
H'FDF5	USBCSR0	DP5CNCT	DP4CNCT	DP3CNCT	DP2CNCE	EP0STOP	EPIVLD	EP0OTC	CKSTOP		
H'FDF6	EPSTLR	—	—	—	—	EP2STL	EP1STL	—	EP0STL		
H'FDF7	EPDIR	—	—	—	—	EP2DIR	EP1DIR	—	—		
H'FDF8	EPRSTR	—	—	—	—	EP2RST	EP1RST	EP0IRST	—		
H'FDF9	DEVRSMR	—	—	—	—	—	—	—	DVR		
H'FDFA	INTSELR0	TSELB	EPIBS2	EPIBS1	EPIBS0	TSELC	EPICS2	EPICS1	EPICS0		
H'FDFB	INTSELR1	—	—	—	—	—	—	DTCBE	DTCCE		
H'FDFC	HOCCR	—	—	PCSP	OCDSP	HOC5E	HOC4E	HOC3E	HOC2E		
H'FDFD	USBCR	FADSEL	FONLY	FNCSTP	UIFRST	HPLLRST	HSRST	FPLLRST	FSRST		
H'FDFE	UPLLCR	—	—	—	CKSEL2	CKSEL1	CKSEL0	PFSEL1	PFSEL0		
H'FDFF	UTESTR2	TESTA	TESTB	TESTC	TESTD	TESTE	TESTF	TESTG	TESTH		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Bus Width
H'FE4C	PCODR	PC_7ODR	PC_6ODR	PC_5ODR	PC_4ODR	PC_3ODR	PC_2ODR	PC_1ODR	PC_0ODR	Ports	8
H'FE4D	PDODR	PD_7ODR	PD_6ODR	PD_5ODR	PD_4ODR	PD_3ODR	PD_2ODR	PD_1ODR	PD_0ODR		
H'FE4E	PCDDR	PC_7DDR	PC_6DDR	PC_5DDR	PC_4DDR	PC_3DDR	PC_2DDR	PC_1DDR	PC_0DDR		
	PCPIN	PC_7PIN	PC_6PIN	PC_5PIN	PC_4PIN	PC_3PIN	PC_2PIN	PC_1PIN	PC_0PIN		
H'FE4F	PDDDR	PD_7DDR	PD_6DDR	PD_5DDR	PD_4DDR	PD_3DDR	PD_2DDR	PD_1DDR	PD_0DDR		
	PDPIN	PD_7PIN	PD_6PIN	PD_5PIN	PD_4PIN	PD_3PIN	PD_2PIN	PD_1PIN	PD_0PIN		
H'FEE6	DDCSWR	SWE	SW	IE	IF	CLR3	CLR2	CLR1	CLR0	IIC0	8
H'FEEB	ISR	—	—	—	—	—	IRQ2F	IRQ1F	IRQ0F	Interrupt controller	8
H'FEEC	ISCRH	—	—	—	—	—	—	—	—		
H'FEED	ISCRL	—	—	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA		
H'FF82	PCSR	—	—	—	—	—	PWCKB	PWCKA	—	PWM	8
H'FF84	SBYCR	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0	System	8
H'FF86	MSTPCRH	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8		
H'FF87	MSTPCRL	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0		
H'FF88	ICCR1	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC1	8
H'FF89	ICSR1	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB		
H'FF8E	ICDR1	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0		
	SARX	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX		
H'FF8F	ICMR1	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0		
	SAR	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS		
H'FF90	TIER	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	FRT	8
H'FF91	TCSR	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA		
H'FF92	FRCH										16
H'FF93	FRCL										
H'FF94	OCRAH										
	OCRBH										
H'FF95	OCRAL										
	OCRBL										
H'FF96	TCR	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0		
H'FF97	TOCR	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB		
H'FF98	ICRAH										
	OCRARH										
H'FF99	ICRAL										
	OCRARL										
H'FF9A	ICRBH										
	OCRAFH										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Bus Width
H'FF9B	ICRBL									FRT	16
	OCRAFL										
H'FF9C	ICRCH										
	OCRDMH										
H'FF9D	ICRCL										
	OCRDML										
H'FF9E	ICRDH										
H'FF9F	ICRDL										
H'FFA0	DADRAH	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	PWMX	8
	DACR	TEST	PWME	—	—	OEB	OEA	OS	CKS		
H'FFA1	DADRAL	DA5	DA4	DA3	DA2	DA1	DA0	CFS	—		
H'FFA6	DADRBH	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6		
	DACNTH										
H'FFA7	DADRBL	DA5	DA4	DA3	DA2	DA1	DA0	CFS	REGS		
	DACNTL							—	REGS		
H'FFA8	TCSR0	OVF	WT/$\overline{IT}$	TME	RSTS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT0	16
	TCNT0 (write)										
H'FFA9	TCNT0 (read)										
H'FFAC	P1PCR	$P1_7$PCR	$P1_6$PCR	$P1_5$PCR	$P1_4$PCR	$P1_3$PCR	$P1_2$PCR	$P1_1$PCR	$P1_0$PCR	Port	8
H'FFAD	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		
H'FFAE	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		
H'FFB0	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		
H'FFB1	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		
H'FFB2	P1DR	$P1_7$DR	$P1_6$DR	$P1_5$DR	$P1_4$DR	$P1_3$DR	$P1_2$DR	$P1_1$DR	$P1_0$DR		
H'FFB3	P2DR	$P2_7$DR	$P2_6$DR	$P2_5$DR	$P2_4$DR	$P2_3$DR	$P2_2$DR	$P2_1$DR	$P2_0$DR		
H'FFB4	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		
H'FFB5	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		
H'FFB6	P3DR	$P3_7$DR	$P3_6$DR	$P3_5$DR	$P3_4$DR	$P3_3$DR	$P3_2$DR	$P3_1$DR	$P3_0$DR		
H'FFB7	P4DR	$P4_7$DR	$P4_6$DR	$P4_5$DR	$P4_4$DR	$P4_3$DR	$P4_2$DR	$P4_1$DR	$P4_0$DR		
H'FFB8	P5DDR	—	—	—	—	—	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR		
H'FFB9	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		
H'FFBA	P5DR	—	—	—	—	—	$P5_2$DR	$P5_1$DR	$P5_0$DR		
H'FFBB	P6DR	$P6_7$DR	$P6_6$DR	$P6_5$DR	$P6_4$DR	$P6_3$DR	$P6_2$DR	$P6_1$DR	$P6_0$DR		
H'FFBE	P7PIN	$P7_7$PIN	$P7_6$PIN	$P7_5$PIN	$P7_4$PIN	$P7_3$PIN	$P7_2$PIN	$P7_1$PIN	$P7_0$PIN		
H'FFC2	IER	—	—	—	—	—	IRQ2E	IRQ1E	IRQ0E	Interrupts	8

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Bus Width
H'FFC3	STCR	—	IICX1	IICX0	IICE	—	USBE	ICKS1	ICKS0	System	8
H'FFC4	SYSCR	CS2E	IOSE	INTM1	INTM0	XRST	NMIEG	HIE	RAME		
H'FFC5	MDCR	EXPE	—	—	—	—	—	MDS1	MDS0		
H'FFC6	BCR	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	IOS1	IOS0		
H'FFC7	WSCR	RAMS	RAM0	ABW	AST	WMS1	WMS0	WC1	WC0		
H'FFC8	TCR0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR_0, TMR_1	8
H'FFC9	TCR1	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0		
H'FFCA	TCSR0	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0		
H'FFCB	TCSR1	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0		
H'FFCC	TCORA0									TMR_0, TMR_1	16, 8
H'FFCD	TCORA1										
H'FFCE	TCORB0										
H'FFCF	TCORB1										
H'FFD0	TCNT0										
H'FFD1	TCNT1										
H'FFD2	PWOERB	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8	PWM	8
H'FFD3	PWOERA	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0		
H'FFD4	PWDPRB	OS15	OS14	OS13	OS12	OS11	OS10	OS9	OS8		
H'FFD5	PWDPRA	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0		
H'FFD6	PWSL	PWCKE	PWCKS	—	—	RS3	RS2	RS1	RS0		
H'FFD7	PWDR0 to PWDR15										
H'FFD8	SMR0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI0	8
	ICCR0	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP	IIC0	
H'FFD9	BRR0									SCI0	
	ICSR0	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	IIC0	
H'FFDA	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	SCI0	
H'FFDB	TDR0										
H'FFDC	SSR0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
H'FFDD	RDR0										
H'FFDE	SCMR0	—	—	—	—	SDIR	SINV	—	SMIF		
	ICDR0	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0	IIC0	
	SARX0	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX		
H'FFDF	ICMR0	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0		
	SAR0	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS		
H'FFE0	ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D	8
H'FFE1	ADDRAL	AD1	AD0	—	—	—	—	—	—		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Bus Width
H'FFE2	ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D	8
H'FFE3	ADDRBL	AD1	AD0	—	—	—	—	—	—		
H'FFE4	ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FFE5	ADDRCL	AD1	AD0	—	—	—	—	—	—		
H'FFE6	ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FFE7	ADDRDL	AD1	AD0	—	—	—	—	—	—		
H'FFE8	ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0		
H'FFE9	ADCR	TRGS1	TRGS0	—	—	—	—	—	—		
H'FFF0	TCRX	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMRX	8
	TCRY	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMRY	
H'FFF1	TCSRX	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0	TMRX	
	TCSRY	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0	TMRY	
H'FFF2	TICRR									TMRX	
	TCORAY									TMRY	
H'FFF3	TICRF									TMRX	
	TCORBY									TMRY	
H'FFF4	TCNTX									TMRX	
	TCNTY									TMRY	
H'FFF5	TCORC									TMRX	
	TISR	—	—	—	—	—	—	—	IS	TMRY	
H'FFF6	TCORAX									TMRX	
H'FFF7	TCORBX										
H'FFFC	TCONRI	SIMOD1	SIMOD0	SCONE	ICST	HFINV	VFINV	HIINV	VIINV	Timer connection	
H'FFFD	TCONRO	HOE	VOE	CLOE	CBOE	HOINV	VOINV	CLOINV	CBOINV		
H'FFFE	TCONRS	TMRX/Y	ISGENE	HOMOD1	HOMOD0	VOMOD1	VOMOD0	CLMOD1	CLMOD0		
H'FFFF	SEDGR	VEDG	HEDG	CEDG	HFEDG	VFEDG	PREQF	IHI	IVI		

B.2 Register Selection Conditions

Address	Register Name	Register Selection Conditions	Module Name
H'FDC0	UPRTCR	MSTP1 = 0	USB
H'FDC1	UTESTR0	USBE = 1 in STCR	
H'FDC2	UTESTR1		
H'FDE1	EPDR2		
H'FDE2	FVSR2H		
H'FDE3	FVSR2L		
H'FDE4	EPSZR1		
H'FDE5	EPDR1		
H'FDE6	FVSR1H		
H'FDE7	FVSR1L		
H'FDE9	EPDR0O		
H'FDEA	FVSR0OH		
H'FDEB	FVSR0OL		
H'FDED	EPDR0I		
H'FDEE	FVSR0IH		
H'FDEF	FVSR0IL		
H'FDF0	PTTER		
H'FDF1	USBIER		
H'FDF2	USBIFR		
H'FDF3	TSFR0		
H'FDF4	TFFR0		
H'FDF5	USBCSR0		
H'FDF6	EPSTLR		
H'FDF7	EPDIR		
H'FDF8	EPRSTR		
H'FDF9	DEVRSMR		
H'FDFA	INTSELR0		
H'FDFB	INTSELR1		
H'FDFC	HOCCR		
H'FDFD	USBCR		
H'FDFE	UPLLCR		
H'FDFF	UTESTR2		

Address	Register Name	Register Selection Conditions		Module Name
H'FE4C	PCODR			Port
H'FE4D	PDODR			
H'FE4E	PCDDR			
	PCPIN			
H'FE4F	PDDDR			
	PDPIN			
H'FEE6	DDCSWR	MSTP4 = 0		IIC0
H'FEEB	ISR	No conditions		Interrupt controller
H'FEEC	ISCRH			
H'FEED	ISCRL			
H'FF82	PCSR	FLSHE = 0 in STCR		PWM
H'FF84	SBYCR	FLSHE = 0 in STCR		System
H'FF86	MSTPCRH			
H'FF87	MSTPCRL			
H'FF88	ICCR1	MSTP3 = 0, IICE = 1 in STCR		IIC1
H'FF89	ICSR1			
H'FF8E	ICDR1	MSTP3 = 0, IICE = 1 in STCR	ICE = 1 in ICCR1	
	SARX1		ICE = 0 in ICCR1	
H'FF8F	ICMR1		ICE = 1 in ICCR1	
	SAR1		ICE = 0 in ICCR1	
H'FF90	TIER	MSTP13 = 0		FRT
H'FF91	TCSR			
H'FF92	FRCH			
H'FF93	FRCL			
H'FF94	OCRAH		OCRS = 0 in TOCR	
	OCRBH		OCRS = 1 in TOCR	
H'FF95	OCRAL		OCRS = 0 in TOCR	
	OCRBL		OCRS = 1 in TOCR	
H'FF96	TCR			
H'FF97	TOCR			

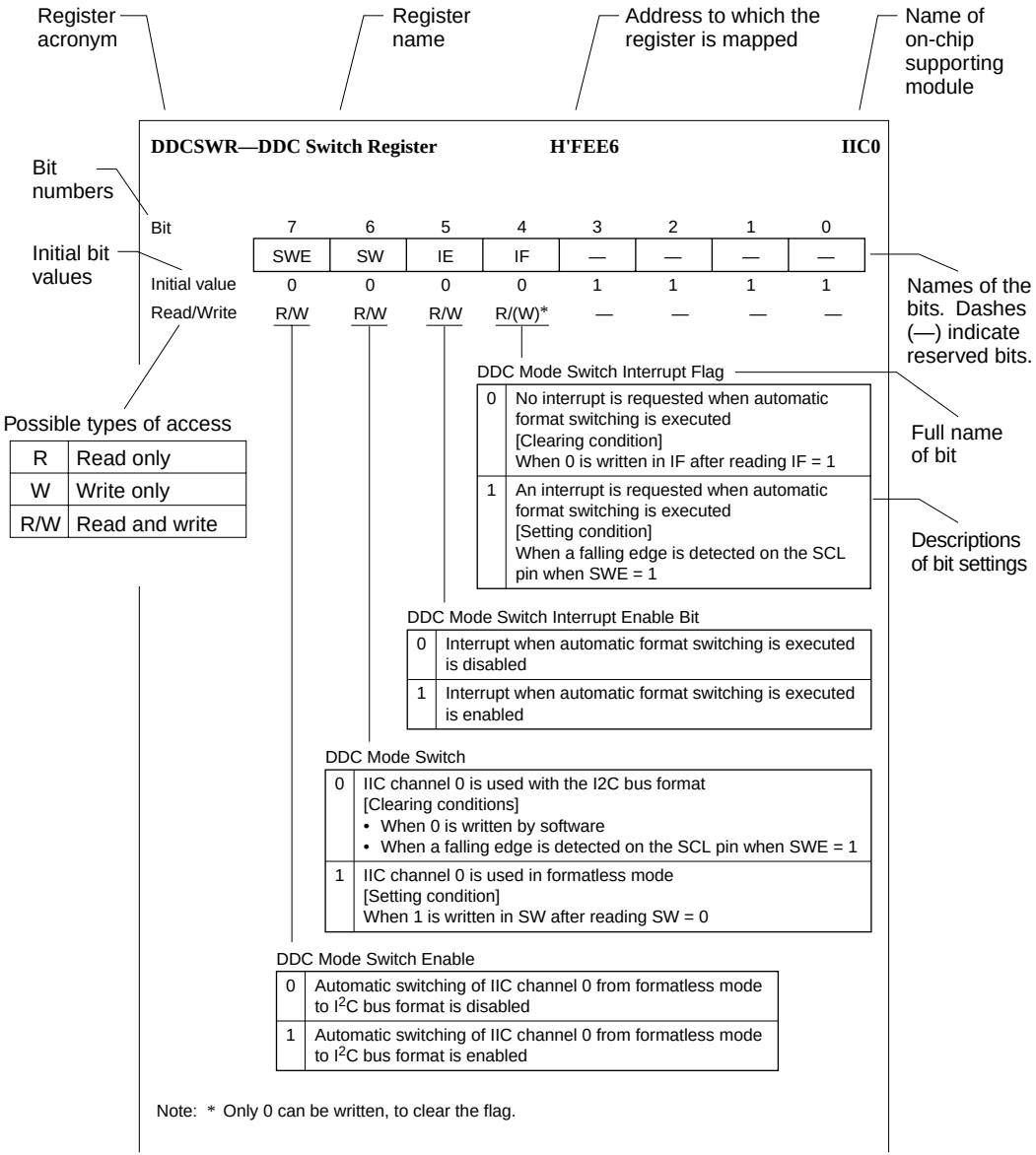
Address	Register Name	Register Selection Conditions		Module Name
H'FF98	ICRAH	MSTP13 = 0	ICRS = 0 in TOCR	FRT
	OCRARH		ICRS = 1 in TOCR	
H'FF99	ICRAL		ICRS = 0 in TOCR	
	OCRARL		ICRS = 1 in TOCR	
H'FF9A	ICRBH		ICRS = 0 in TOCR	
	OCRAFH		ICRS = 1 in TOCR	
H'FF9B	ICRBL		ICRS = 0 in TOCR	
	OCRAFL		ICRS = 1 in TOCR	
H'FF9C	ICRCH		ICRS = 0 in TOCR	
	OCRDMH		ICRS = 1 in TOCR	
H'FF9D	ICRCL		ICRS = 0 in TOCR	
	OCRDML		ICRS = 1 in TOCR	
H'FF9E	ICRDH			
H'FF9F	ICRDL			
H'FFA0	DADRAH	MSTP11 = 0, IICE = 1 in STCR	REGS=0 in DACNT/DADRB	PWMX
	DACR		REGS=1 in DACNT/DADRB	
H'FFA1	DADRAL	MSTP11 = 0, IICE = 1 in STCR	REGS=0 in DACNT/DADRB	
H'FFA6	DADRBH	MSTP11 = 0, IICE = 1 in STCR	REGS=0 in DACNT/DADRB	
	DACNTH		REGS=1 in DACNT/DADRB	
H'FFA7	DADRBL		REGS=0 in DACNT/DADRB	
	DACNTL		REGS=1 in DACNT/DADRB	
H'FFA8	TCSR0	No conditions		WDT0
	TCNT0 (write)			
H'FFA9	TCNT0 (read)			
H'FFAC	P1PCR	No conditions		Ports
H'FFAD	P2PCR			
H'FFAE	P3PCR			
H'FFB0	P1DDR			

Address	Register Name	Register Selection Conditions	Module Name
H'FFB1	P2DDR	No conditions	Ports
H'FFB2	P1DR		
H'FFB3	P2DR		
H'FFB4	P3DDR		
H'FFB5	P4DDR		
H'FFB6	P3DR		
H'FFB7	P4DR		
H'FFB8	P5DDR		
H'FFB9	P6DDR		
H'FFBA	P5DR		
H'FFBB	P6DR		
H'FFBE	P7PIN		
H'FFC2	IER	No conditions	Interrupts
H'FFC3	STCR	No conditions	System
H'FFC4	SYSCR		
H'FFC5	MDCR		
H'FFC6	BCR	No conditions	Bus controller
H'FFC7	WSCR		
H'FFC8	TCR0	MSTP12 = 0	TMR_0, TMR_1
H'FFC9	TCR1		
H'FFCA	TCSR0		
H'FFCB	TCSR1		
H'FFCC	TCORA0		
H'FFCD	TCORA1		
H'FFCE	TCORB0		
H'FFCF	TCORB1		
H'FFD0	TCNT0		
H'FFD1	TCNT1		
H'FFD2	PWOERB	No conditions	PWM
H'FFD3	PWOERA		
H'FFD4	PWDPRB		
H'FFD5	PWDPRA		

Address	Register Name	Register Selection Conditions		Module Name
H'FFD6	PWSL	MSTP11 = 0		PWM
H'FFD7	PWDR0 to 15			
H'FFD8	SMR0	MSTP7 = 0, IICE = 0 in STCR		SCI0
	ICCR0	MSTP4 = 0, IICE = 1 in STCR		IIC0
H'FFD9	BRR0	MSTP7 = 0, IICE = 0 in STCR		SCI0
	ICSR0	MSTP4 = 0, IICE = 1 in STCR		IIC0
H'FFDA	SCR0	MSTP7 = 0		SCI0
H'FFDB	TDR0			
H'FFDC	SSR0			
H'FFDD	RDR0			
H'FFDE	SCMR0	MSTP7 = 0, IICE = 0 in STCR		
	ICDR0	MSTP4 = 0, IICE = 1 in STCR	ICE = 1 in ICCR0	IIC0
	SARX0		ICE = 0 in ICCR0	
H'FFDF	ICMR0		ICE = 1 in ICCR0	
	SAR0		ICE = 0 in ICCR0	
H'FFE0	ADDRAH	MSTP9 = 0		A/D
H'FFE1	ADDRAL			
H'FFE2	ADDRBH			
H'FFE3	ADDRBL			
H'FFE4	ADDRCH			
H'FFE5	ADDRCL			
H'FFE6	ADDRDH			
H'FFE7	ADDRDL			
H'FFE8	ADCSR			
H'FFE9	ADCR			
H'FFF0	TCRX	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	TMRX
	TCRY		TMRX/Y = 1 in TCONRS	TMRY
H'FFF1	TCSRX	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	TMRX
	TCSRY		TMRX/Y = 1 in TCONRS	TMRY
H'FFF2	TICRR	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	TMRX
	TCORAY		TMRX/Y = 1 in TCONRS	TMRY

Address	Register Name	Register Selection Conditions		Module Name
H'FFF3	TICRF	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	TMRX
	TCORBY		TMRX/Y = 1 in TCONRS	TMRY
H'FFF4	TCNTX	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	TMRX
	TCNTY		TMRX/Y = 1 in TCONRS	TMRY
H'FFF5	TCORC	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	TMRX
	TISR		TMRX/Y = 1 in TCONRS	TMRY
H'FFF6	TCORAX	MSTP8 = 0, HIE = 0 in SYSCR	TMRX/Y = 0 in TCONRS	TMRX
H'FFF7	TCORBX			
H'FFFC	TCONRI	MSTP8 = 0, HIE = 0 in SYSCR		Timer connection
H'FFFD	TCONRO			
H'FFFE	TCONRS			
H'FFFF	SEDGR			

B.3 Functions



UPRTCR—USB Port Control Register **H'FDC0** **USB**

Bit	7	6	5	4	3	2	1	0
	—	—	DSPSEL2	DSPSEL1	DSPSEL0	PCNMD2	PCNMD1	PCNMD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Port Connection Mode Select 2 to 0

0	0	0	User mode
		1	Digital upstream mode
	1	0	Digital downstream mode
		1	Digital upstream/downstream mode
1	0	0	Upstream transceiver/receiver monitor mode
		1	Downstream transceiver/receiver monitor mode
	1	—	Reserved

Downstream Port Select 2 to 0

0	0	0	Downstream port 2 selected
		1	Downstream port 3 selected
	1	0	Downstream port 4 selected
		1	Downstream port 5 selected
1	—	—	Downstream port 1 selected

UTESTR0—USB Test Register 0 **H'FDC1** **USB**
UTESTR1—USB Test Register 1 **H'FDC2** **USB**

UTESTR0

Bit	7	6	5	4	3	2	1	0
	TEST15	TEST14	TEST13	TEST12	TEST11	TEST10	TEST9	TEST8
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

UTESTR1

Bit	7	6	5	4	3	2	1	0
	TEST7	TEST6	TEST5	TEST4	TEST3	TEST2	TEST1	TEST0
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

EPDR2—Endpoint Data Register 2 **H'FDE1** **USB**

Bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
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*

Mediates data transfer between CPU and FIFO for each USB function endpoint host input transfer/host output transfer

Note: * The EPDR2 transfer direction is determined by the endpoint direction register. EPDR2 is a write-only register when designated for host input transfer, and a read-only register when designated for host output transfer.

FVSR2H—FIFO Valid Size Register 2H **H'FDE2** **USB**
FVSR2L—FIFO Valid Size Register 2L **H'FDE3** **USB**

	FVSR2H								FVSR2L							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	N9	N8	N7	N6	N5	N4	N3	N2	N1	N0
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

Indicates number of valid data bytes in FIFO for each USB function endpoint host input/host output

EPSZR1—Endpoint Size Register 1 **H'FDE4** **USB**

Bit	7	6	5	4	3	2	1	0
	EP1SZ3	EP1SZ2	EP1SZ1	EP1SZ0	EP2SZ3	EP2SZ2	EP2SZ1	EP2SZ0
Initial value	0	1	0	0	0	1	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Specifies number of FIFO bytes used

Bits 7 to 4	EP1 FIFO size
Bits 3 to 0	EP2 FIFO size

EPnSZ3	EPnSZ2	EPnSZ1	EPnSZ0	Operating Mode
0	0	0	0	FIFO size = 0 bytes (settable for EP2 only)
			1	Setting prohibited
		1	0	Setting prohibited
			1	Setting prohibited
	1	0	0	FIFO size = 16 bytes (Initial value)
			1	FIFO size = 32 bytes (settable for EP1 only)
		1	0	Setting prohibited
			1	Setting prohibited
1	—	—	—	Setting prohibited

n = 1, 2

EPDR1—Endpoint Data Register 1 **H'FDE5** **USB**

Bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Mediates data transfer between CPU and FIFO for each USB function endpoint host input transfer/host output transfer

FVSR1H—FIFO Valid Size Register 1H **H'FDE6** **USB**
FVSR1L—FIFO Valid Size Register 1H **H'FDE7** **USB**

	FVSR1H								FVSR1L							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	N9	N8	N7	N6	N5	N4	N3	N2	N1	N0
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

Indicates number of valid data bytes in FIFO for each USB function endpoint host input/host output

EPDR0O—Endpoint Data Register 0O **H'FDE9** **USB**

Bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Mediates data transfer between CPU and FIFO for each USB function endpoint host input transfer/host output transfer

FVSR0OH—FIFO Valid Size Register 0OH **H'FDEA** **USB**
FVSR0OL—FIFO Valid Size Register 0OL **H'FDEB** **USB**

	FVSR0OH								FVSR0OL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	N9	N8	N7	N6	N5	N4	N3	N2	N1	N0
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

Indicates number of valid data bytes in FIFO for each USB function endpoint host input/host output

EPDR0I—Endpoint Data Register 0I **H'FDED** **USB**

Bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Mediates data transfer between CPU and FIFO for each USB function endpoint host input transfer/host output transfer

FVSR0IH—FIFO Valid Size Register 0IH **H'FDEE** **USB**
FVSR0IL—FIFO Valid Size Register 0IL **H'FDEF** **USB**

	FVSR0IH								FVSR0IL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	N9	N8	N7	N6	N5	N4	N3	N2	N1	N0
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

Indicates number of valid data bytes in FIFO for each USB function endpoint host input/host output

PTTER—Packet Transmit Enable Register H'FDF0 USB

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2TE	EP1TE	EP0ITE	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R

Endpoint 0I Packet Transmit Enable

0	Initial set value
(1)	[1 write] Endpoint 0 IN-FIFO FVSR0I is updated

Endpoint 1 Packet Transmit Enable

0	Initial set value
(1)	[1 write] Endpoint 1 IN-FIFO FVSR1 is updated

Endpoint 2 Packet Transmit Enable

0	Initial set value
(1)	[1 write] Endpoint 2 IN-FIFO FVSR2 is updated

Note: * Only 1 can be written.

USBIER—USB Interrupt Enable Register **H'FDF1** **USB**

Bit	7	6	5	4	3	2	1	0
	—	—	BRSTE	SOFE	SPNDE	TFE	TSE	SETUPE
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Setup Interrupt Enable

0	USB function setup interrupts disabled
1	USB function setup interrupts enabled

Transfer Successful Interrupt Enable

0	USB function transfer successful interrupts disabled
1	USB function transfer successful interrupts enabled

Transfer Failed Interrupt Enable

0	USB function transfer failed interrupts disabled
1	USB function transfer failed interrupts enabled

Suspend Interrupt Enable

0	USB function suspend OUT interrupts and suspend IN interrupts disabled
1	USB function suspend OUT interrupts and suspend IN interrupts enabled

SOF Interrupt Enable

0	USB function SOF interrupts disabled
1	USB function SOF interrupts enabled

Bus Reset Interrupt Enable

0	USB function bus reset interrupts disabled
1	USB function bus reset interrupts enabled

USBIFR—USB Interrupt Flag Register **H'FDF2** **USB**

Bit	7	6	5	4	3	2	1	0
	TS	TF	—	BRSTF	SOFF	SPNDOF	SPNDIF	SETUPF
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Setup Interrupt Flag

0	[Clearing condition] When 0 is written in SETUPF after reading SETUPF = 1
1	[Setting condition] When USB function endpoint 0 receives SETUP token

Suspend IN Interrupt Flag

0	[Clearing condition] When 0 is written in SPNDIF after reading SPNDIF = 1
1	[Setting condition] When USB function switches from normal state to suspend state

Suspend OUT Interrupt Flag

0	[Clearing condition] When 0 is written in SPNDOF after reading SPNDOF = 1
1	[Setting condition] When USB function switches from suspend state to normal state

SOF Interrupt Flag

0	[Clearing condition] When 0 is written in SOFF after reading SOFF = 1
1	[Setting condition] When USB function detects SOF (Start of Frame)

Bus Reset Interrupt Flag

0	[Clearing condition] When 0 is written in BRSTF after reading BRSTF = 1
1	[Setting condition] When USB function detects a bus reset from upstream

Transfer Failed Interrupt Status

0	All bits in transfer fail flag register (TFFR) are 0
1	At least one bit in transfer fail flag register (TFFR) is 1

Transfer Successful Interrupt Status

0	All bits in transfer success flag register (TSFR) are 0
1	At least one bit in transfer success flag register (TSFR) is 1

Note: * Only 0 can be written, after reading 1, to clear the flag.

TSFR—Transfer Success Flag Register H'FDF3 USB

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2TS	EP1TS	EP0ITS	EP0OTS
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Endpoint 0 Host Output Transfer Success Flag

0	Endpoint 0 is in host output transfer standby state [Clearing conditions] • When 0 is written in EP0OTS after reading EP0OTS = 1 • When endpoint 0 receives a SETUP token
1	Endpoint 0 host output transfer (OUT transaction or SETUP transaction) has ended normally [Setting conditions] • ACK handshake established after OUT token reception and data transfer (ACK transmission) • When command received after SETUP token reception requires processing by the slave CPU

Endpoint 0 Host Input Transfer Success Flag

0	Endpoint 0 is in host input transfer standby state [Clearing conditions] • When 0 is written in EP0ITS after reading EP0ITS = 1 • When endpoint 0 receives a SETUP token
1	Endpoint 0 host input transfer (IN transaction) has ended normally [Setting condition] ACK handshake established after IN token reception and data transfer (ACK reception)

Endpoint 1 Transfer Success Flag

0	Endpoint 1 is in transfer standby state [Clearing condition] When 0 is written in EP1TS after reading EP1TS = 1
1	Endpoint 1 host input transfer (IN transaction) has ended normally [Setting condition] ACK handshake established after IN token reception and data transfer (ACK reception)

Endpoint 2 Transfer Success Flag

0	Endpoint 2 is in transfer standby state) [Clearing condition] When 0 is written in EP2TS after reading EP2TS = 1
1	Endpoint 2 host input transfer (IN transaction) or host output transfer (OUT transaction) has ended normally [Setting conditions] • ACK handshake established after IN token reception and data transfer (ACK reception) • ACK handshake established after OUT token reception and data transfer (ACK transmission)

Note: * Only 0 can be written, after reading 1, to clear the flag.

TFFR—Transfer Fail Flag Register **H'FDF4** **USB**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2TF	EP1TF	EP0ITF	EP0OTF
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Endpoint 0 Host Output Transfer Fail Flag

0	Endpoint 0 is in host output transfer standby state [Clearing conditions] • When 0 is written in EP0OTF after reading EP0OTF = 1 • When endpoint 0 receives a SETUP token
1	Endpoint 0 host output transfer (OUT transaction or SETUP transaction) has ended abnormally [Setting conditions] • Data transfer not possible due to FIFO full condition after OUT token reception (NAK transmission) • Data transfer not possible because EP0OTC = 1 after OUT token reception (NAK transmission) • Communication error after OUT token reception • When command received after SETUP token reception can be processed within the USB function core

Endpoint 0 Host Input Transfer Fail Flag

0	Endpoint 0 is in host input transfer standby state [Clearing conditions] • When 0 is written in EP0ITF after reading EP0ITF = 1 • When endpoint 0 receives a SETUP token
1	Endpoint 0 host input transfer (IN transaction) has ended abnormally [Setting conditions] • ACK handshake not established after IN token reception and data transfer • Data transfer not possible due to FIFO empty condition after IN token reception (NAK transmission)

Endpoint 1 Transfer Fail Flag

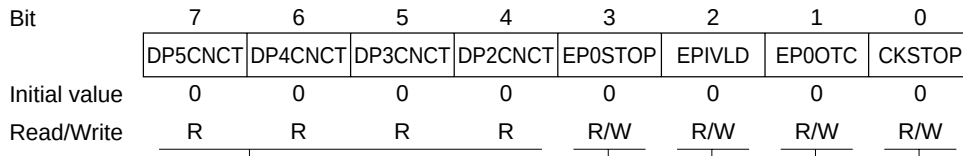
0	Endpoint 1 is in transfer standby state [Clearing condition] When 0 is written in EP1TF after reading EP1TF = 1
1	Endpoint 1 host input transfer (IN transaction) has ended abnormally [Setting conditions] • ACK handshake not established after IN token reception and data transfer • Data transfer not possible due to FIFO empty condition after IN token reception (NAK transmission)

Endpoint 2 Transfer Fail Flag

0	Endpoint 2 is in transfer standby state [Clearing condition] When 0 is written to EP2TF after reading EP2TF = 1
1	Endpoint 2 host input transfer (IN transaction) or host output transfer (OUT transaction) has ended abnormally [Setting conditions] • ACK handshake not established after IN token reception and data transfer • Data transfer not possible due to FIFO empty condition after IN token reception (NAK transmission) • Data reception not possible due to FIFO full condition after OUT token reception (NAK transmission) • DATA0/DATA1 PID toggle error after OUT token reception

Note: * Only 0 can be written, after reading 1, to clear the flag.

USBCSR0—USB Control/Status Register 0 H'FDF5 USB



Clock Stop

0	Clock is supplied to USB function [Clearing conditions] • System reset • Function soft reset • Suspend OUT interrupt flag setting
1	Clock supply to USB function is stopped [Setting condition] When 1 is written in CKSTOP after reading CKSTOP = 0

Endpoint 0O Transfer Control

0	EP0 OUT-FIFO writing stopped • Subsequent writes to EP0 OUT-FIFO are invalid [Clearing conditions] • System reset • Function soft reset • Command data reception in SETUP transaction (EP0OTS flag setting)
1	EP0 OUT-FIFO operational [Setting conditions] • SETUP token reception • When 1 is written in EP0OTC after reading EP0OTC = 0

Endpoint Information Valid

0	Endpoint information (EPINFO) has not been set [Clearing conditions] • System reset • Function soft reset
1	Endpoint information (EPINFO) has been set

Endpoint 0 Stop

0	EP0 OUT-FIFO, IN-FIFO operational [Clearing conditions] • System reset • Function soft reset
1	EP0 OUT-FIFO reading stopped • FVSR0O contents are not changed by an EPDR0O read EP0 IN-FIFO writing and transfer stopped • FIFO contents are not changed by an EPDR0I write • FVSR0I contents are not changed by setting EP0IPTE

Downstream Port Connect 5 to 2

0	Cable is not connected to downstream port [Clearing conditions] • System reset • Downstream port disconnect • USB hub upstream port disconnect (Total downstream disconnect by software in reconnect process)
1	Cable is connected to downstream port, and power is being supplied [Setting condition] Downstream port connect

EPSTLR—Endpoint Stall Register **H'FDF6** **USB**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2STL	EP1STL	—	EP0STL
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/W	R/W	R	R/W

Endpoint 0 Stall

0	Endpoint 0 is operational [Clearing condition] When endpoint 0 receives a SETUP token
1	Endpoint 0 is in stall state [Setting condition] When 1 is written in EP0STL after reading EP0STL = 0

Endpoint 1 Stall

0	Endpoint 1 is operational
1	Endpoint 1 is in stall state

Endpoint 2 Stall

0	Endpoint 2 is operational
1	Endpoint 2 is in stall state

EPDIR—Endpoint Direction Register **H'FDF7** **USB**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2DIR	EP1DIR	—	—
Initial value	1	1	1	1	1	1	0	0
Read/Write	R	R	R	R	R/W	R/W	R	R

Endpoint 1 Data Transfer Direction Control Flag

0	Setting prohibited
1	Endpoint 1 is designated for host input transfer

Endpoint 2 Data Transfer Direction Control Flag

0	Endpoint 2 is designated for host output transfer
1	Endpoint 2 is designated for host input transfer

EPRSTR—Endpoint Reset Register **H'FDF8** **USB**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EP2RST	EP1RST	EP0IRST	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R/(W)*	R/(W)*	R/(W)*	R

Endpoint 0I Reset

0	Initial set value
(1)	[1 write] FVSR0I is initialized to H'0010

Endpoint 1 Reset

0	Initial set value
(1)	[1 write] EP1 FIFO size = 16 bytes: FVSR1 is initialized to H'0010 EP1 FIFO size = 32 bytes: FVSR1 is initialized to H'0020

Endpoint 2 Reset

0	Initial set value
(1)	[1 write] EP2DIR = 0: FVSR2 is initialized to H'0000 EP2DIR = 1: FVSR2 is initialized to H'0010

Note: * Only 1 can be written.

DEVRSMR—Device Resume Register **H'FDF9** **USB**

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

Device Resume (DVR)

0	(Initial value)
(1)	[1 write] Suspend state is cleared (remote wakeup)

Note: * Only 1 can be written.

INTSELR0—Interrupt Source Select Register 0 H'FDFA USB

Bit	7	6	5	4	3	2	1	0
	TSELB	EPIBS2	EPIBS1	EPIBS0	TSELC	EPICS2	EPICS1	EPICS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Interrupt C Endpoint Select 2 to 0

0	0	0	Initial set value
		1	Endpoint 1 selected
	1	0	Endpoint 2 selected
		1	Setting prohibited
1	—	—	Setting prohibited

Transfer Select C

0	USBIC is requested by a TS interrupt; the endpoint constituting the TS interrupt source is specified by bits EPICS2 to EPICS0
1	USBIC is requested by a TF interrupt; the endpoint constituting the TF interrupt source is specified by bits EPICS2 to EPICS0

Interrupt B Endpoint Select 2 to 0

0	0	0	Initial set value
		1	Endpoint 1 selected
	1	0	Endpoint 2 selected
		1	Setting prohibited
1	—	—	Setting prohibited

Transfer Select B

0	USBIB is requested by a TS interrupt; the endpoint constituting the TS interrupt source is specified by bits EPIBS2 to EPIBS0
1	USBIB is requested by a TF interrupt; the endpoint constituting the TF interrupt source is specified by bits EPIBS2 to EPIBS0

INTSELR1—Interrupt Source Select Register 1 H'FDFB USB

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

Note: Do not write 1 to the bits in this register.

HOCCR—Hub Overcurrent Control Register **H'FDFC** **USB**

Bit	7	6	5	4	3	2	1	0
	—	—	PCSP	OCDSP	HOC5E	HOC4E	HOC3E	HOC2E
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Overcurrent Detection Control Enable 2

0	Pins $\overline{ENP}_2$ and $\overline{OCP}_2$ are general ports (PC_4, PC_0)
1	Pins $\overline{ENP}_2$ and $\overline{OCP}_2$ have output enable and overcurrent detection functions

Overcurrent Detection Control Enable 3

0	Pins $\overline{ENP}_3$ and $\overline{OCP}_3$ are general ports (PC_5, PC_1)
1	Pins $\overline{ENP}_3$ and $\overline{OCP}_3$ have output enable and overcurrent detection functions

Overcurrent Detection Control Enable 4

0	Pins $\overline{ENP}_4$ and $\overline{OCP}_4$ are general ports (PC_6, PC_2)
1	Pins $\overline{ENP}_4$ and $\overline{OCP}_4$ have output enable and overcurrent detection functions

Overcurrent Detection Control Enable 5

0	Pins $\overline{ENP}_5$ and $\overline{OCP}_5$ are general ports (PC_7, PC_3)
1	Pins $\overline{ENP}_5$ and $\overline{OCP}_5$ have output enable and overcurrent detection functions

Overcurrent Detection Polarity

0	Power supply control IC outputs low level in case of overcurrent detection
1	Power supply control IC outputs high level in case of overcurrent detection

Power Supply Enable Control Polarity

0	Power supply control IC requires low-level input for enabling
1	Power supply control IC requires high-level input for enabling

USBCR—USB Control Register **H'FDFD** **USB**

Bit	7	6	5	4	3	2	1	0
	FADSEL	FONLY	FNCSTP	UIFRST	HPLLRST	HSRST	FPLLRST	FSRST
Initial value	0	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

Function Block Internal State Soft Reset

0	Internal state of USB function block is set to operational state
1	Internal state of USB function block is set to reset state (excluding DPLL)

Function Block PLL Soft Reset

0	Function DPLL is placed in operational state
1	Function DPLL is placed in reset state

Hub Block Internal State Soft Reset

0	Internal state of USB hub block is set to operational state
1	Internal state of USB hub block is set to reset state (excluding DPLL)

Hub Block PLL Soft Reset

0	Hub DPLL is placed in operational state
1	Hub DPLL is placed in reset state

USB Interface Soft Reset

0	EPSZR1, USBIER, EPDIR, INTSELR0, and INTSELR1 are placed in operational state
1	EPSZR1, USBIER, EPDIR, INTSELR0, and INTSELR1 are placed in reset state

USB Function Stop/Suspend

0	For USB function block, USB hub downstream port 1 internal connection is set to connected state
1	For USB function block, USB hub downstream port 1 internal connection is set to disconnected state, and power-down state is set

USB Function Select

0	USB function block is connected internally to USB hub downstream port 1; USB hub block is enabled
1	USB function block is directly connected to upstream port; USB hub block is disabled

USB Function I/O Analog/Digital Select

0	USD+ and USD– pins are used for USB function block data input/output
1	USB function block data input/output is implemented by multiplexing Philips transceiver/receiver (PIDUSB11A) compatible control input/output with port C pins

UPLLCR—USB PLL Control Register **H'FDFE** **USB**

Bit	7	6	5	4	3	2	1	0
	—	—	—	CKSEL2	CKSEL1	CKSEL0	PFSEL1	PFSEL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W

PLL Frequency Select

0	0	PLL input clock is 8 MHz
	1	PLL input clock is 12 MHz
1	0	PLL input clock is 16 MHz
	1	PLL input clock is 20 MHz

Clock Source Select 2 to 0

0	0	0	PLL operation halted, clock input halted
	—	—	PLL operation halted, clock input halted
1	0	0	Setting prohibited
		1	PLL operation halted USB clock pulse generator (XTAL12: 48 MHz) used directly instead of PLL output
	1	0	PLL operates with system clock pulse generator (XTAL) as clock source
		1	PLL operates with USB clock pulse generator (XTAL12) as clock source

UTESTR2—USB Test Register 2 **H'FDFF** **USB**

UTESTR2

Bit	7	6	5	4	3	2	1	0
	TESTA	TESTB	TESTC	TESTD	TESTE	TESTF	TESTG	TESTH
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

PCODR—Port C Data Output Register **H'FE4C** **Port C**

Bit	7	6	5	4	3	2	1	0
	PC_7ODR	PC_6ODR	PC_5ODR	PC_4ODR	PC_3ODR	PC_2ODR	PC_1ODR	PC_0ODR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output data for port C pins

PDODR—Port D Data Output Register **H'FE4D** **Port D**

Bit	7	6	5	4	3	2	1	0
	PD_7ODR	PD_6ODR	PD_5ODR	PD_4ODR	PD_3ODR	PD_2ODR	PD_1ODR	PD_0ODR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output data for port D pins

PCDDR—Port C Data Direction Register **H'FE4E** **Port C**

Bit	7	6	5	4	3	2	1	0
	PC_7DDR	PC_6DDR	PC_5DDR	PC_4DDR	PC_3DDR	PC_2DDR	PC_1DDR	PC_0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specify input or output for port C pins

PCPIN—Port C Input Data Register **H'FE4E** **Port C**

Bit	7	6	5	4	3	2	1	0
	PC_7PIN	PC_6PIN	PC_5PIN	PC_4PIN	PC_3PIN	PC_2PIN	PC_1PIN	PC_0PIN
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Port C pin states

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

PDDDR—Port D Data Direction Register **H'FE4F** **Port D**

Bit	7	6	5	4	3	2	1	0
	PD_7DDR	PD_6DDR	PD_5DDR	PD_4DDR	PD_3DDR	PD_2DDR	PD_1DDR	PD_0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specify input or output for port D pins

PDPIN—Port D Input Data Register **H'FE4F** **Port D**

Bit	7	6	5	4	3	2	1	0
	PD_7PIN	PD_6PIN	PD_5PIN	PD_4PIN	PD_3PIN	PD_2PIN	PD_1PIN	PD_0PIN
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Port D pin states

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

DDCSWR—DDC Switch Register **H'FEE6** **IIC0**

Bit	7	6	5	4	3	2	1	0
	SWE	SW	IE	IF	CLR3	CLR2	CLR1	CLR0
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/(W)*1	W*2	W*2	W*2	W*2

IIC Clear 3 to 0

0	0	—	—	Setting prohibited
	1	0	0	Setting prohibited
			1	IIC0 internal latch clearance
		1	0	IIC1 internal latch clearance
			1	IIC0 and IIC1 internal latch clearance
1	—	—	—	Invalid setting

DDC Mode Switch Interrupt Flag

0	No interrupt is requested when automatic format switching is executed [Clearing condition] When 0 is written in IF after reading IF = 1
1	An interrupt is requested when automatic format switching is executed [Setting condition] When a falling edge is detected on the SCL pin when SWE = 1

DDC Mode Switch Interrupt Enable Bit

0	Interrupt when automatic format switching is executed is disabled
1	Interrupt when automatic format switching is executed is enabled

DDC Mode Switch

0	IIC channel 0 is used with the I^2C bus format [Clearing conditions] • When 0 is written by software • When a falling edge is detected on the SCL pin when SWE = 1
1	IIC channel 0 is used in formatless mode [Setting condition] When 1 is written in SW after reading SW = 0

DDC Mode Switch Enable

0	Automatic switching of IIC channel 0 from formatless mode to I^2C bus format is disabled
1	Automatic switching of IIC channel 0 from formatless mode to I^2C bus format is enabled

Notes: 1. Only 0 can be written, to clear the flag.
2. Always read as 1.

ISR—IRQ Status Register **H'FEEB** **Interrupt Controller**

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

IRQ2 to IRQ0 Flags

0	[Clearing conditions] • When 0 is written in IRQnF after reading IRQnF = 1 • When interrupt exception handling is executed while low-level detection is set (IRQnSCB = IRQnSCA = 0) and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed while falling, rising, or both-edge detection is set (IRQnSCB = 1 or IRQnSCA = 1)
1	[Setting conditions] • When $\overline{\text{IRQn}}$ input goes low while low-level detection is set (IRQnSCB = IRQnSCA = 0) • When a falling edge occurs in $\overline{\text{IRQn}}$ input while falling edge detection is set (IRQnSCB = 0, IRQnSCA = 1) • When a rising edge occurs in $\overline{\text{IRQn}}$ input while rising edge detection is set (IRQnSCB = 1, IRQnSCA = 0) • When a falling or rising edge occurs in $\overline{\text{IRQn}}$ input while both-edge detection is set (IRQnSCB = IRQnSCA = 1)

Notes: n = 2 to 0
* Only 0 can be written, to clear the flag.

ISCRH—IRQ Sense Control Register H **H'FEEC** **Interrupt Controller**
ISCRL—IRQ Sense Control Register L **H'FEED** **Interrupt Controller**

ISCRH

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

Reserved

ISCRL

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

IRQ2 to IRQ0 Sense Control A and B

ISCRL bits 5 to 0		Description
IRQ2SCB to IRQ0SCB	IRQ2SCA to IRQ0SCA	
0	0	Interrupt request generated by low level of $\overline{IRQ}_2$–$\overline{IRQ}_0$ input
	1	Interrupt request generated by falling edge of $\overline{IRQ}_2$–$\overline{IRQ}_0$ input
1	0	Interrupt request generated by rising edge of $\overline{IRQ}_2$–$\overline{IRQ}_0$ input
	1	Interrupt request generated by rising and falling edges of $\overline{IRQ}_2$–$\overline{IRQ}_0$ input

PCSR—Peripheral Clock Select Register **H'FF82** **PWM**

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

PWM Clock Select

PWSL		PCSR		Description
Bit 7	Bit 6	Bit 2	Bit 1	
PWCKE	PWCKS	PWCKB	PWCKA	
0	—	—	—	Clock input stopped
1	0	—	—	ϕ (system clock) selected
	1	0	0	ϕ/2 selected
			1	ϕ/4 selected
		1	0	ϕ/8 selected
			1	ϕ/16 selected

SBYCR—Standby Control Register **H'FF84** **System**

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	—	SCK2	SCK1	SCK0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

System Clock Select 2 to 0

0	0	0	Bus master is in high-speed mode
		1	Medium-speed clock = ϕ/2
	1	0	Medium-speed clock = ϕ/4
		1	Medium-speed clock = ϕ/8
1	0	0	Medium-speed clock = ϕ/16
		1	Medium-speed clock = ϕ/32
	1	—	—

Standby Timer Select 2 to 0

0	0	0	Standby time = 8,192 states
		1	Standby time = 16,384 states
	1	0	Standby time = 32,768 states
		1	Standby time = 65,536 states
1	0	0	Standby time = 131,072 states
		1	Standby time = 262,144 states
	1	0	Reserved
		1	Standby time = 16 states

Software Standby

0	Transition to sleep mode on execution of SLEEP instruction in high-speed mode or medium-speed mode
1	Transition to software standby mode on execution of SLEEP instruction in high-speed mode or medium-speed mode

MSTPCRH—Module Stop Control Register H **H'FF86** **System**
MSTPCRL—Module Stop Control Register L **H'FF87** **System**

	MSTPCRH								MSTPCRL							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	MSTP15	MSTP14	MSTP13	MSTP12	MSTP11	MSTP10	MSTP9	MSTP8	MSTP7	MSTP6	MSTP5	MSTP4	MSTP3	MSTP2	MSTP1	MSTP0
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Module Stop

0	Module stop mode cleared
1	Module stop mode set

The correspondence between MSTPCR bits and on-chip supporting modules is shown below.

Register	Bit	Module
MSTPCRH	MSTP15*	—
	MSTP14*	—
	MSTP13	16-bit free-running timer (FRT)
	MSTP12	8-bit timers (TMR0, TMR1)
	MSTP11	8-bit PWM timer (PWM), 14-bit PWM timer (PWMX)
	MSTP10*	—
	MSTP9	A/D converter
	MSTP8	8-bit timers (TMRX, TMRY), timer connection
MSTPCRL	MSTP7	Serial communication interface 0 (SCI0)
	MSTP6*	—
	MSTP5*	—
	MSTP4	I^2C bus interface (IIC) channel 0
	MSTP3	I^2C bus interface (IIC) channel 1
	MSTP2*	—
	MSTP1	Universal serial bus interface (USB)
	MSTP0*	—

Note: * Bits 15, 14, 10, 6, 5, 2, and 0 can be read and written but must always be set to 1.

ICCR1—I²C Bus Control Register 1 H'FF88 IIC1

Bit	7	6	5	4	3	2	1	0
	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	W

Start Condition/Stop Condition Prohibit

0	Writing issues a start or stop condition, in combination with the BBSY flag
1	Reading always returns a value of 1; writing is invalid

I²C Bus Interface Interrupt Request Flag

0	Waiting for transfer, or transfer in progress
1	Interrupt requested

Note: For the clearing and setting conditions, see section 16.2.5, I²C Bus Control Register (ICCR).

Bus Busy

0	Bus is free [Clearing condition] When a stop condition is detected
1	Bus is busy [Setting condition] When a start condition is detected

Acknowledge Bit Judgement Select

0	Acknowledge bit is ignored and continuous transfer is performed
1	If acknowledge bit is 1, continuous transfer is interrupted

Master/Slave Select (MST), Transmit/Receive Select (TRS)

0	0	Slave receive mode
	1	Slave transmit mode
1	0	Master receive mode
	1	Master transmit mode

Note: For details see section 16.2.5, I²C Bus Control Register (ICCR).

I²C Bus Interface Interrupt Enable

0	Interrupt requests disabled
1	Interrupt requests enabled

I²C Bus Interface Enable

0	I²C bus interface module disabled, with SCL and SDA signal pins set to port function SAR and SARX can be accessed
1	I²C bus interface module enabled for transfer operations (pins SCL and SDA are driving the bus) ICMR and ICDR can be accessed

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

ICSR1—I²C Bus Status Register 1 H'FF89 IIC1

Bit	7	6	5	4	3	2	1	0
	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/W

Acknowledge Bit

0	Receive mode: 0 is output at acknowledge output timing Transmit mode: indicates that the receiving device has acknowledged the data (0 value)
1	Receive mode: 1 is output at acknowledge output timing Transmit mode: indicates that the receiving device has not acknowledged the data (1 value)

General Call Address Recognition Flag*2

0	General call address not recognized
1	General call address recognized

Slave Address Recognition Flag*2

0	Slave address or general call address not recognized
1	Slave address or general call address recognized

Arbitration Lost Flag*2

0	Bus arbitration won
1	Bus arbitration lost

Second Slave Address Recognition Flag*2

0	Second slave address not recognized
1	Second slave address recognized

I²C Bus Interface Continuous Transmission/Reception Interrupt Request Flag*2

0	Waiting for transfer, or transfer in progress
1	Continuous transfer state

Normal Stop Condition Detection Flag*2

0	No normal stop condition
1	In I²C bus format slave mode: Normal stop condition detected In other modes: No meaning

Error Stop Condition Detection Flag*2

0	No error stop condition
1	In I²C bus format slave mode: Error stop condition detected In other modes: No meaning

Notes: 1. Only 0 can be written, to clear the flag.
2. For the clearing and setting conditions, see section 16.2.6, I²C Bus Status Register (ICSR).

ICDR1—I²C Bus Data Register 1 **H'FF8E** **IIC1**

Bit	7	6	5	4	3	2	1	0
	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ICDRR

Bit	7	6	5	4	3	2	1	0
	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R	R	R	R	R	R	R	R

- ICDRS

Bit	7	6	5	4	3	2	1	0
	ICDRS7	ICDRS6	ICDRS5	ICDRS4	ICDRS3	ICDRS2	ICDRS1	ICDRS0
Initial value	—	—	—	—	—	—	—	—
Read/Write	—	—	—	—	—	—	—	—

- ICDRT

Bit	7	6	5	4	3	2	1	0
	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0
Initial value	—	—	—	—	—	—	—	—
Read/Write	W	W	W	W	W	W	W	W

- TDRE, RDRF (internal flags)

Bit							—	—
							TDRE	RDRF
Initial value							0	0
Read/Write							—	—

Note: For details see section 16.2.1, I²C Bus Data Register (ICDR).

SARX—Second Slave Address Register 1 **H'FF8E** **IIC1**

Bit	7	6	5	4	3	2	1	0
	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Second Slave Address

Format Select X

DDCSWR Bit 6	SAR Bit 0	SARX Bit 0	Operating Mode
SW	FS	FSX	
0	0	0	I²C bus format • SAR and SARX slave addresses recognized
		1	I²C bus format • SAR slave address recognized • SARX slave address ignored
	1	0	I²C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	0	0	Formatless mode (start/stop conditions not detected) • Acknowledge bit present
		1	
	1	0	
		1	Formatless mode* (start/stop conditions not detected) • No acknowledge bit

Note: * Do not select this mode when automatic switching to the I²C bus format is performed by means of a DDCSWR setting.

SAR—Slave Address Register **H'FF8F** **IIC1**

Bit	7	6	5	4	3	2	1	0
	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Slave Address

Format Select

DDCSWR Bit 6	SAR Bit 0	SARX Bit 0	Operating Mode
SW	FS	FSX	
0	0	0	I²C bus format • SAR and SARX slave addresses recognized
		1	I²C bus format • SAR slave address recognized • SARX slave address ignored
	1	0	I²C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	0	0	Formatless mode (start/stop conditions not detected) • Acknowledge bit present
		1	
	1	0	
		1	Formatless mode* (start/stop conditions not detected) • No acknowledge bit

Note: * Do not select this mode when automatic switching to the I²C bus format is performed by means of a DDCSWR setting.

ICMR1—I²C Bus Mode Register 1 H'FF8F IIC1

Bit	7	6	5	4	3	2	1	0
	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Counter

BC2	BC1	BC0	Synchronous Serial Format	I²C Bus Format
0	0	0	8	9
		1	1	2
	1	0	2	3
		1	3	4
1	0	0	4	5
		1	5	6
	1	0	6	7
		1	7	8

Transfer Clock Select

IICX	CKS2	CKS1	CKS0	Clock
0	0	0	0	φ/28
			1	φ/40
		1	0	φ/48
			1	φ/64
	1	0	0	φ/80
			1	φ/100
		1	0	φ/112
			1	φ/128
1	0	0	0	φ/56
			1	φ/80
		1	0	φ/96
			1	φ/128
	1	0	0	φ/160
			1	φ/200
		1	0	φ/224
			1	φ/256

Wait Insertion Bit

0	Data and acknowledge transferred consecutively
1	Wait inserted between data and acknowledge

MSB-First/LSB-First Select*

0	MSB-first
1	LSB-first

Note: * Do not set this bit to 1 when using the I²C bus format.

TIER—Timer Interrupt Enable Register **H'FF90** **FRT**

Bit	7	6	5	4	3	2	1	0
	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

Timer Overflow Interrupt Enable

0	OVF interrupt request (FOVI) is disabled
1	OVF interrupt request (FOVI) is enabled

Output Compare Interrupt B Enable

0	OCFB interrupt request (OCIB) is disabled
1	OCFB interrupt request (OCIB) is enabled

Output Compare Interrupt A Enable

0	OCFA interrupt request (OCIA) is disabled
1	OCFA interrupt request (OCIA) is enabled

Input Capture Interrupt D Enable

0	ICFD interrupt request (ICID) is disabled
1	ICFD interrupt request (ICID) is enabled

Input Capture Interrupt C Enable

0	ICFC interrupt request (ICIC) is disabled
1	ICFC interrupt request (ICIC) is enabled

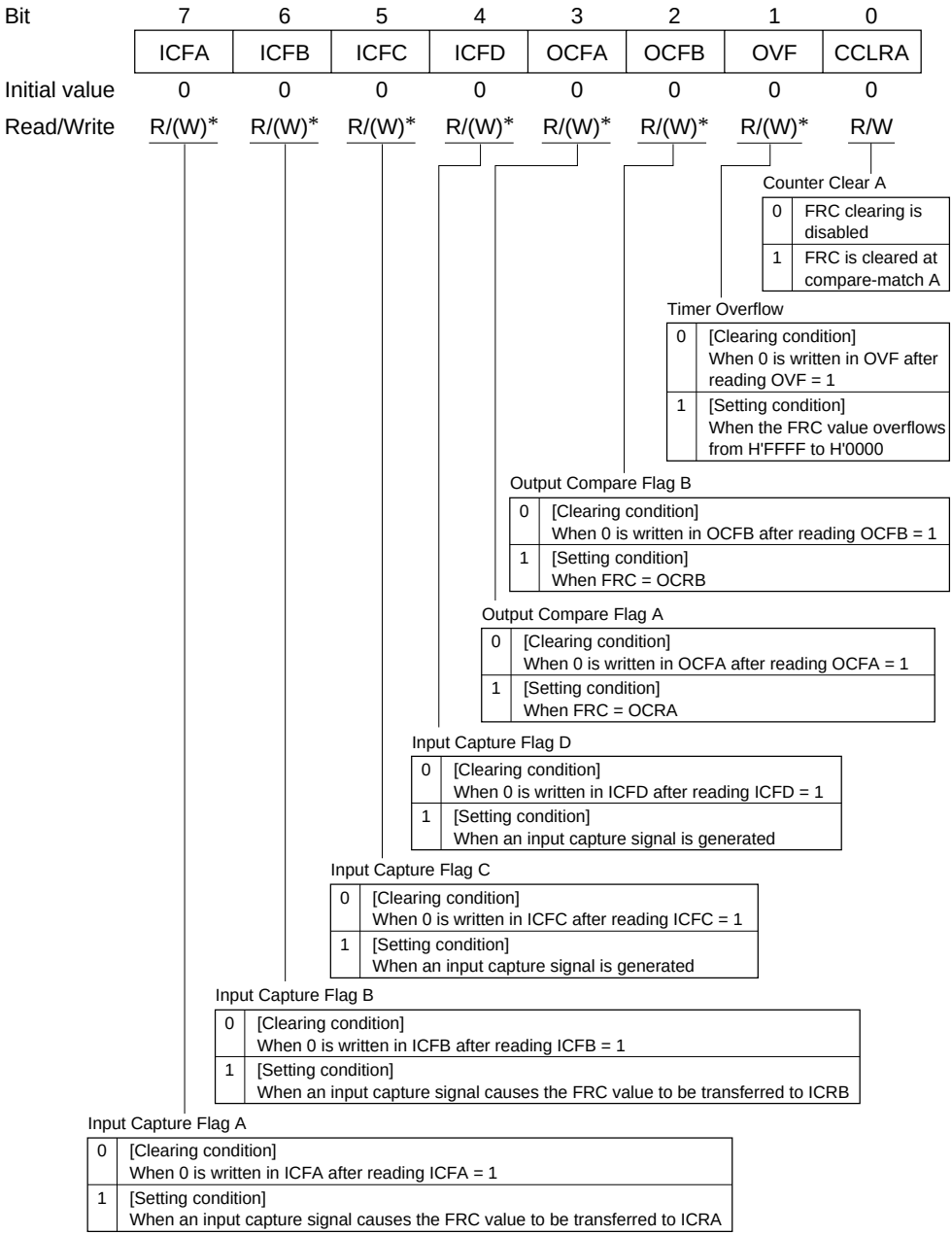
Input Capture Interrupt B Enable

0	ICFB interrupt request (ICIB) is disabled
1	ICFB interrupt request (ICIB) is enabled

Input Capture Interrupt A Enable

0	ICFA interrupt request (ICIA) is disabled
1	ICFA interrupt request (ICIA) is enabled

TCSR—Timer Control/Status Register **H'FF91** **FRT**



Note: * Only 0 can be written in bits 7 to 1, to clear the flags.

FRCH—Free-Running Counter H **H'FF92** **FRT**
FRCL—Free-Running Counter L **H'FF93** **FRT**

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

Up-counter

OCRAH—Output Compare Register AH **H'FF94** **FRT**
OCRAL—Output Compare Register AL **H'FF95** **FRT**
OCRBH—Output Compare Register BH **H'FF94** **FRT**
OCRBL—Output Compare Register BL **H'FF95** **FRT**

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

Constantly compared with FRC value; OCF is set when OCR = FRC

TCR—Timer Control Register **H'FF96** **FRT**

Bit	7	6	5	4	3	2	1	0
	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	φ/2 internal clock source
	1	φ/8 internal clock source
1	0	φ/32 internal clock source
	1	External clock source (rising edge)

Buffer Enable B

0	ICRD is not used as ICRB buffer register
1	ICRD is used as ICRB buffer register

Buffer Enable A

0	ICRC is not used as ICRA buffer register
1	ICRC is used as ICRA buffer register

Input Edge Select D

0	Capture on falling edge of input capture input D
1	Capture on rising edge of input capture input D

Input Edge Select C

0	Capture on falling edge of input capture input C
1	Capture on rising edge of input capture input C

Input Edge Select B

0	Capture on falling edge of input capture input B
1	Capture on rising edge of input capture input B

Input Edge Select A

0	Capture on falling edge of input capture input A
1	Capture on rising edge of input capture input A

TOCR—Timer Output Compare Control Register H'FF97 FRT

Bit	7	6	5	4	3	2	1	0
	ICRDMS	OCRAMS	ICRS	OCRS	OEA	OEB	OLVLA	OLVLB
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output Level B

0	0 output at compare-match B
1	1 output at compare-match B

Output Level A

0	0 output at compare-match A
1	1 output at compare-match A

Output Enable B

0	Output compare B output disabled
1	Output compare B output enabled

Output Enable A

0	Output compare A output disabled
1	Output compare A output enabled

Output Compare Register Select

0	OCRA register selected
1	OCRB register selected

Input Capture Register Select

0	ICRA, ICRB, and ICRC registers selected
1	OCRAR, OCRAF, and OCRDM registers selected

Output Compare A Mode Select

0	OCRA set to normal operating mode
1	OCRA set to operating mode using OCRAR and OCRAF

Input Capture D Mode Select

0	ICRD set to normal operating mode
1	ICRD set to operating mode using OCRDM

OCRARH—Output Compare Register ARH	**H'FF98**	**FRT**
OCRARL—Output Compare Register ARL	**H'FF99**	**FRT**
OCRAFH—Output Compare Register AFH	**H'FF9A**	**FRT**
OCRAFL—Output Compare Register AFL	**H'FF9B**	**FRT**

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

Used for OCRA operation when OCRAMS = 1 in TOCR
(For details see section 11.2.4, Output Compare Registers AR and AF (OCRAR, OCRAF).)

ICRAH—Input Capture Register AH	**H'FF98**	**FRT**
ICRAL—Input Capture Register AL	**H'FF99**	**FRT**
ICRBH—Input Capture Register BH	**H'FF9A**	**FRT**
ICRBL—Input Capture Register BL	**H'FF9B**	**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

Stores FRC value when input capture signal is input

OCRDMH—Output Compare Register DMH	**H'FF9C**	**FRT**
OCRDML—Output Compare Register DML	**H'FF9D**	**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/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Used for ICRD operation when ICRDMS = 1 in TOCR
(For details see section 11.2.5, Output Compare Register DM (OCRDM).)

ICRCH—Input Capture Register CH	**H'FF9C**	**FRT**
ICRCL—Input Capture Register CL	**H'FF9D**	**FRT**
ICRDH—Input Capture Register DH	**H'FF9E**	**FRT**
ICRDL—Input Capture Register DL	**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

Stores FRC value when input capture signal is input
(ICRC and ICRD can be used for buffer operation.
For details see section 11.2.3, Input Capture Registers A to D (ICRA to ICRD).)

DACR—PWM (D/A) Control Register **H'FFA0** **PWMX**

Bit	7	6	5	4	3	2	1	0
	TEST	PWME	—	—	OEB	OEA	OS	CKS
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

Clock Select

0	Operates at resolution (T) = system clock cycle time (t_{cyc})
1	Operates at resolution (T) = system clock cycle time (t_{cyc}) × 2

Output Select

0	Direct PWM output
1	Inverted PWM output

Output Enable A

0	PWM (D/A) channel A output (PWX0 output pin) disabled
1	PWM (D/A) channel A output (PWX0 output pin) enabled

Output Enable B

0	PWM (D/A) channel B output (PWX1 output pin) disabled
1	PWM (D/A) channel B output (PWX1 output pin) enabled

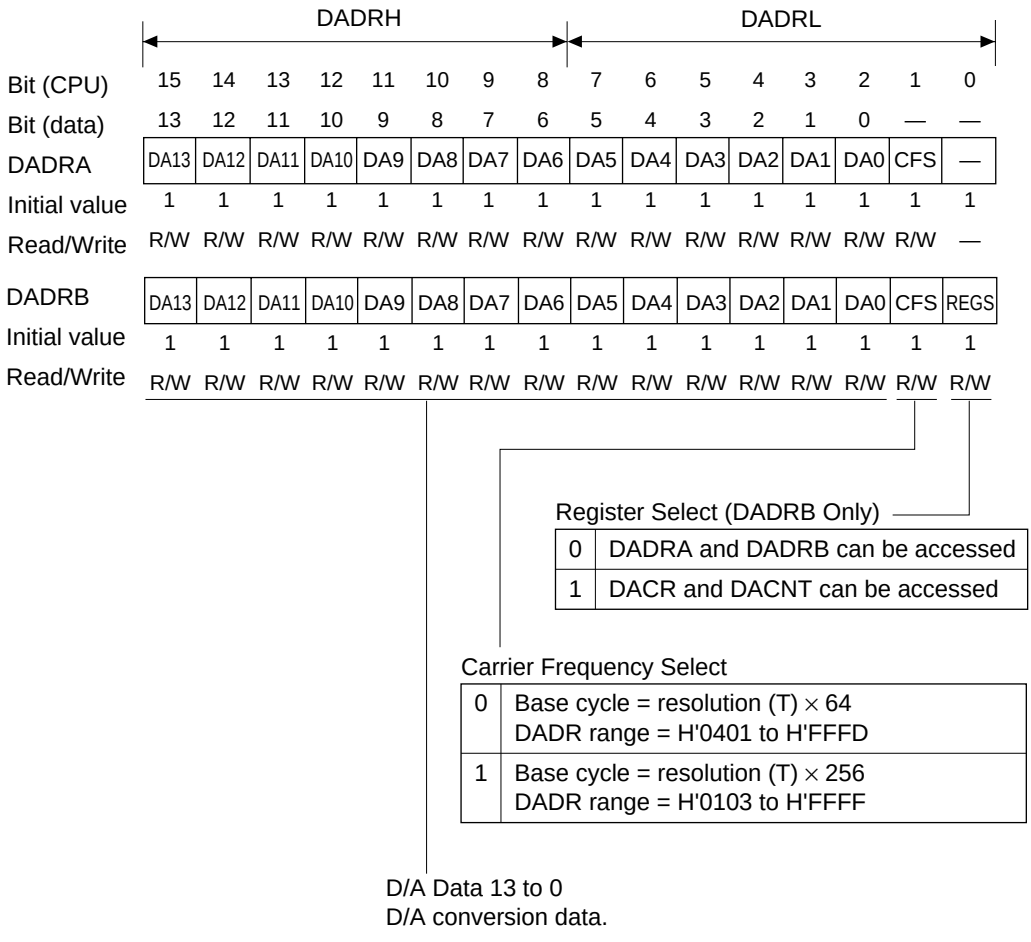
PWM Enable

0	DACNT operates as 14-bit up-counter
1	DACNT halts at H'0003

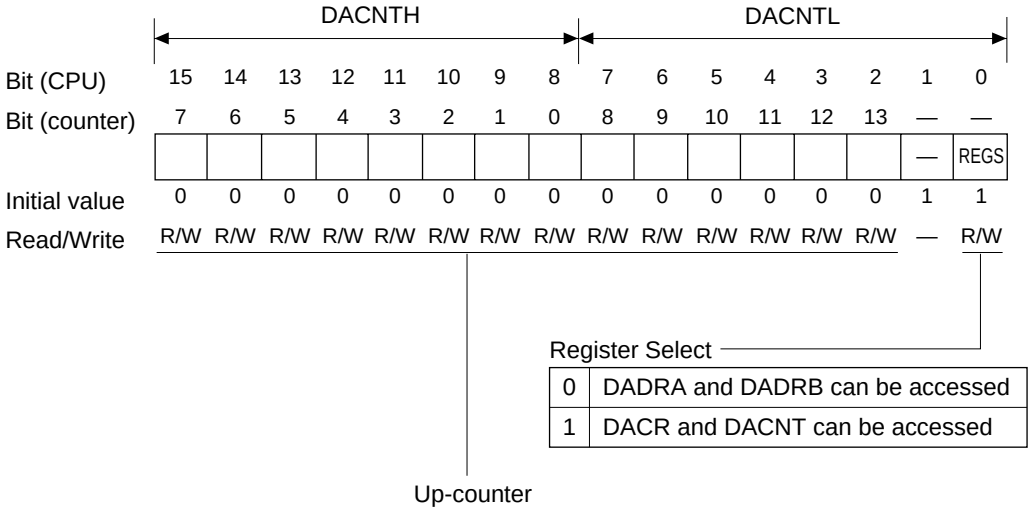
Test Mode

0	PWM (D/A) in user state, normal operation
1	PWM (D/A) in test state, correct conversion results unobtainable

DADRAH—PWM (D/A) Data Register AH	**H'FFA0**	**PWMX**
DADRAL—PWM (D/A) Data Register AL	**H'FFA1**	**PWMX**
DADRBH—PWM (D/A) Data Register BH	**H'FFA6**	**PWMX**
DADRBL—PWM (D/A) Data Register BL	**H'FFA7**	**PWMX**



DACNTH—PWM (D/A) Counter H **H'FFA6** **PWMX**
DACNTL—PWM (D/A) Counter L **H'FFA7** **PWMX**



TCSR0—Timer Control/Status Register 0 H'FFA8 WDT0

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	RSTS	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select 2 to 0

CKS2	CKS1	CKS0	Clock	Overflow Period (when ϕ = 20 MHz)
0	0	0	ϕ/2	25.6 µs
		1	ϕ/64	819.2 µs
	1	0	ϕ/128	1.6 ms
		1	ϕ/512	6.6 ms
1	0	0	ϕ/2048	26.2 ms
		1	ϕ/8192	104.9 ms
	1	0	ϕ/32768	419.4 ms
		1	ϕ/131072	1.68 s

Reset or NMI

0	NMI interrupt requested
1	Internal reset requested

Reserved

Timer Enable

0	TCNT is initialized to H'00 and halted
1	TCNT counts

Timer Mode Select

0	Interval timer mode: Sends the CPU an interval timer interrupt request (WOVI) when TCNT overflows
1	Watchdog timer mode: Generates a reset or NMI interrupt when TCNT overflows

Overflow Flag

0	[Clearing conditions] • When 0 is written in the TME bit • When 0 is written in OVF after reading TCSR when OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00 (When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset)

Notes: The method of writing to TCSR is more complicated that for most other registers, to prevent accidental overwriting. For details see section 14.2.4, Notes on Register Access.
* Only 0 can be written, to clear the flag.

TCNT0—Timer Counter 0 **H'FFA8 (W), H'FFA9 (R)** **WDT0**

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

Up-counter

P1PCR—Port 1 MOS Pull-Up Control Register **H'FFAC** **Port 1**

Bit	7	6	5	4	3	2	1	0
	$P1_7PCR$	$P1_6PCR$	$P1_5PCR$	$P1_4PCR$	$P1_3PCR$	$P1_2PCR$	$P1_1PCR$	$P1_0PCR$
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

Control port 1 MOS input pull-ups

P2PCR—Port 2 MOS Pull-Up Control Register **H'FFAD** **Port 2**

Bit	7	6	5	4	3	2	1	0
	$P2_7PCR$	$P2_6PCR$	$P2_5PCR$	$P2_4PCR$	$P2_3PCR$	$P2_2PCR$	$P2_1PCR$	$P2_0PCR$
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

Control port 2 MOS input pull-ups

P3PCR—Port 3 MOS Pull-Up Control Register H'FFAE Port 3

Bit	7	6	5	4	3	2	1	0
	$P3_7PCR$	$P3_6PCR$	$P3_5PCR$	$P3_4PCR$	$P3_3PCR$	$P3_2PCR$	$P3_1PCR$	$P3_0PCR$
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

Control port 3 MOS input pull-ups

P1DDR—Port 1 Data Direction Register H'FFB0 Port 1

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

Specify input or output for port 1 pins

P2DDR—Port 2 Data Direction Register H'FFB1 Port 2

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

Specify input or output for port 2 pins

P1DR—Port 1 Data Register **H'FFB2** **Port 1**

Bit	7	6	5	4	3	2	1	0
	$P1_7DR$	$P1_6DR$	$P1_5DR$	$P1_4DR$	$P1_3DR$	$P1_2DR$	$P1_1DR$	$P1_0DR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output data for port 1 pins

P2DR—Port 2 Data Register **H'FFB3** **Port 2**

Bit	7	6	5	4	3	2	1	0
	$P2_7DR$	$P2_6DR$	$P2_5DR$	$P2_4DR$	$P2_3DR$	$P2_2DR$	$P2_1DR$	$P2_0DR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output data for port 2 pins

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

Specify input or output for port 3 pins

P4DDR—Port 4 Data Direction Register **H'FFB5** **Port 4**

Bit	7	6	5	4	3	2	1	0
	$P4_7DDR$	$P4_6DDR$	$P4_5DDR$	$P4_4DDR$	$P4_3DDR$	$P4_2DDR$	$P4_1DDR$	$P4_0DDR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specify input or output for port 4 pins

P3DR—Port 3 Data Register **H'FFB6** **Port 3**

Bit	7	6	5	4	3	2	1	0
	$P3_7DR$	$P3_6DR$	$P3_5DR$	$P3_4DR$	$P3_3DR$	$P3_2DR$	$P3_1DR$	$P3_0DR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output data for port 3 pins

P4DR—Port 4 Data Register **H'FFB7** **Port 4**

Bit	7	6	5	4	3	2	1	0
	$P4_7DR$	$P4_6DR$	$P4_5DR$	$P4_4DR$	$P4_3DR$	$P4_2DR$	$P4_1DR$	$P4_0DR$
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

Output data for port 4 pins

Note: * Determined by the state of pin $P4_6$.

P5DDR—Port 5 Data Direction Register **H'FFB8** **Port 5**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	$P5_2DDR$	$P5_1DDR$	$P5_0DDR$
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	W	W	W

Specify input or output for port 5 pins

P6DDR—Port 6 Data Direction Register **H'FFB9** **Port 6**

Bit	7	6	5	4	3	2	1	0
	$P6_7DDR$	$P6_6DDR$	$P6_5DDR$	$P6_4DDR$	$P6_3DDR$	$P6_2DDR$	$P6_1DDR$	$P6_0DDR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specify input or output for port 6 pins

P5DR—Port 5 Data Register **H'FFBA** **Port 5**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	$P5_2DR$	$P5_1DR$	$P5_0DR$
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

Output data for port 5 pins

P6DR—Port 6 Data Register **H'FFBB** **Port 6**

Bit	7	6	5	4	3	2	1	0
	$P6_7DR$	$P6_6DR$	$P6_5DR$	$P6_4DR$	$P6_3DR$	$P6_2DR$	$P6_1DR$	$P6_0DR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output data for port 6 pins

P7PIN—Port 7 Input Data Register **H'FFBE** **Port 7**

Bit	7	6	5	4	3	2	1	0
	$P7_7PIN$	$P7_6PIN$	$P7_5PIN$	$P7_4PIN$	$P7_3PIN$	$P7_2PIN$	$P7_1PIN$	$P7_0PIN$
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Port 7 pin states

Note: * Determined by the state of pins $P7_7$ to $P7_0$.

IER—IRQ Enable Register **H'FFC2** **Interrupt Controller**

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

IRQ2 to IRQ0 Enable

0	IRQn interrupt disabled
1	IRQn interrupt enabled

(n = 2 to 0)

STCR—Serial Timer Control Register **H'FFC3** **System**

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

Internal Clock Source select*1

USB Enable

0	CPU access to USB data registers and control registers is disabled
1	CPU access to USB data registers and control registers is enabled

Reserved

I2C Master Enable

0	CPU access to I2C bus interface data registers and control registers is disabled
1	CPU access to I2C bus interface data registers and control registers is enabled

I2C Transfer Rate Select 1 and 0*2

Reserved

Notes: 1. Used for 8-bit timer input clock selection. For details see section 12.2.4, Timer Control Register (TCR).
2. Used for I2C bus interface transfer clock selection. For details see section 16.2.4, I2C Bus Mode Register (ICMR).

SYSCR—System Control Register **H'FFC4** **System**

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

RAM Enable

0	On-chip RAM is disabled
1	On-chip RAM is enabled

Host Interface Enable

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

NMI Edge Select

0	Interrupt request generated by NMI falling edge
1	Interrupt request generated by NMI rising edge

External Reset

0	Reset generated by watchdog timer overflow
1	Reset generated by external reset

Interrupt Control Mode 1 and 0

INTM1	INTM0	Interrupt Control Mode	Description
0	0	0	Interrupts are controlled by I bit
	1	1	Cannot be used in these groups
1	0	2	Cannot be used in these groups
	1	3	Cannot be used in these groups

IOS Enable
Do not set this bit to 1.

Chip Select 2 Enable
Do not set this bit to 1.

MDCR—Mode Control Register **H'FFC5** **System**

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

Expanded Mode Enable

Mode Select 1 and 0
Mode pin states.

Note: * Determined by the MD_1 and MD_0 pins (H8/3577 Group) or the $\overline{TEST}$ pin (H8/3567 Group).

BCR—Bus Control Register **H'FFC6** **Bus Controller**

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

Do not write any values other than the initial values.

WSCR—Wait State Control Register **H'FFC7** **Bus Controller**

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

Do not write any values other than the initial values.

TCR0—Timer Control Register 0 **H'FFC8** **TMR_0**
TCR1—Timer Control Register 1 **H'FFC9** **TMR_1**

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

Counter Clear 1 and 0

0	0	Clearing is disabled
	1	Cleared on compare-match A
1	0	Cleared on compare-match B
	1	Cleared on rising edge of external reset input

Timer Overflow Interrupt Enable

0	OVF interrupt request (OVI) is disabled
1	OVF interrupt request (OVI) is enabled

Compare-Match Interrupt Enable A

0	CMFA interrupt request (CMIA) is disabled
1	CMFA interrupt request (CMIA) is enabled

Compare-Match Interrupt Enable B

0	CMFB interrupt request (CMIB) is disabled
1	CMFB interrupt request (CMIB) is enabled

Clock Select 2 to 0

Channel	Bit 2	Bit 1	Bit 0	Description
	CKS2	CKS1	CKS0	
0	0	0	0	Clock input disabled
			1*1	φ/8 internal clock source, counted on falling edge
				φ/2 internal clock source, counted on falling edge
		1	0*1	φ/64 internal clock source, counted on falling edge
				φ/32 internal clock source, counted on falling edge
			1*1	φ/1024 internal clock source, counted on falling edge
				φ/256 internal clock source, counted on falling edge
	1	0	0	Counted on TCNT1 overflow signal*2
1	0	0	0	Clock input disabled
			1*1	φ/8 internal clock source, counted on falling edge
				φ/2 internal clock source, counted on falling edge
		1	0*1	φ/64 internal clock source, counted on falling edge
				φ/128 internal clock source, counted on falling edge
			1*1	φ/1024 internal clock source, counted on falling edge
				φ/2048 internal clock source, counted on falling edge
	1	0	0	Counted on TCNT0 compare-match A*2
X	0	0	0	Clock input disabled
			1	Counted on φ internal clock source
		1	0	φ/2 internal clock source, counted on falling edge
			1	φ/4 internal clock source, counted on falling edge
	1	0	0	Clock input disabled
Y	0	0	0	Clock input disabled
			1	φ/4 internal clock source, counted on falling edge
		1	0	φ/256 internal clock source, counted on falling edge
			1	φ/2048 internal clock source, counted on falling edge
	1	0	0	Clock input disabled
Common	1	0	1	External clock source, counted on rising edge
		1	0	External clock source, counted on falling edge
			1	External clock source, counted on both rising and falling edges

Notes: 1. Selected by ICKS1 and ICKS0 in STCR. For details see section 12.2.4, Timer Control Register (TCR).
2. If the count input of channel 0 is the TCNT1 overflow signal and that of channel 1 is the TCNT0 compare-match signal, no incrementing clock will be generated. Do not use this setting.

TCSR0—Timer Control/Status Register 0 **H'FFCA** **TMR_0**

TCSR0

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ADTE	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Output Select 1 and 0

0	0	No change when compare-match A occurs
	1	0 output when compare-match A occurs
1	0	1 output when compare-match A occurs
	1	Output inverted when compare-match A occurs (toggle output)

Output Select 3 and 2

0	0	No change when compare-match B occurs
	1	0 output when compare-match B occurs
1	0	1 output when compare-match B occurs
	1	Output inverted when compare-match B occurs (toggle output)

A/D Trigger Enable

0	A/D converter start requests by compare-match A are disabled
1	A/D converter start requests by compare-match A are enabled

Timer Overflow Flag

0	[Clearing condition] When 0 is written in OVF after reading OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00

Compare-Match Flag A

0	[Clearing condition] When 0 is written in CMFA after reading CMFA = 1
1	[Setting condition] When TCNT = TCORA

Compare-Match Flag B

0	[Clearing condition] When 0 is written in CMFB after reading CMFB = 1
1	[Setting condition] When TCNT = TCORB

Note: * Only 0 can be written in bits 7 to 5, to clear the flags.

TCSR1—Timer Control/Status Register 1 **H'FFCB** **TMR_1**

TCSR1

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/W

Output Select 1 and 0

0	0	No change when compare-match A occurs
	1	0 output when compare-match A occurs
1	0	1 output when compare-match A occurs
	1	Output inverted when compare-match A occurs (toggle output)

Output Select 3 and 2

0	0	No change when compare-match B occurs
	1	0 output when compare-match B occurs
1	0	1 output when compare-match B occurs
	1	Output inverted when compare-match B occurs (toggle output)

Timer Overflow Flag

0	[Clearing condition] When 0 is written in OVF after reading OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00

Compare-Match Flag A

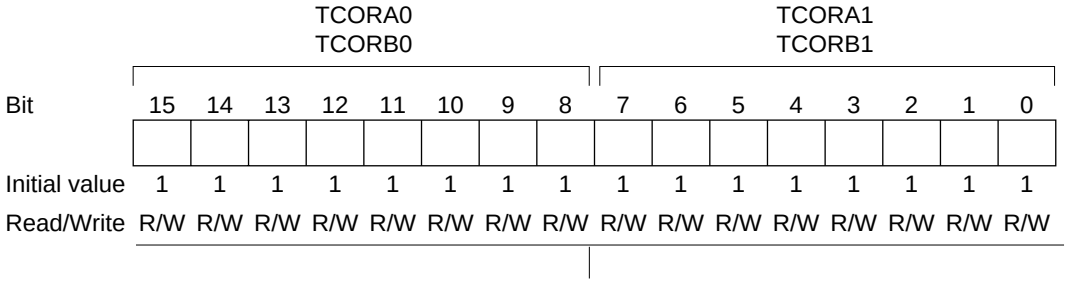
0	[Clearing condition] When 0 is written in CMFA after reading CMFA = 1
1	[Setting condition] When TCNT = TCORA

Compare-Match Flag B

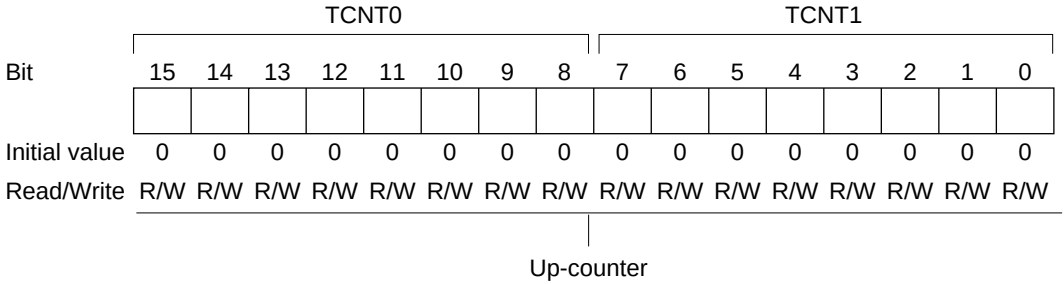
0	[Clearing condition] When 0 is written in CMFB after reading CMFB = 1
1	[Setting condition] When TCNT = TCORB

Note: * Only 0 can be written in bits 7 to 5, to clear the flags.

TCORA0—Time Constant Register A0 **H'FFCC** **TMR_0**
TCORA1—Time Constant Register A1 **H'FFCD** **TMR_1**
TCORB0—Time Constant Register B0 **H'FFCE** **TMR_0**
TCORB1—Time Constant Register B1 **H'FFCF** **TMR_1**



TCNT0—Timer Counter 0 **H'FFD0** **TMR_0**
TCNT1—Timer Counter 1 **H'FFD1** **TMR_1**



PWOERB—PWM Output Enable Register B **H'FFD2** **PWM**
PWOERA—PWM Output Enable Register A **H'FFD3** **PWM**

Bit	7	6	5	4	3	2	1	0
PWOERB	OE15	OE14	OE13	OE12	OE11	OE10	OE9	OE8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
PWOERA	OE7	OE6	OE5	OE4	OE3	OE2	OE1	OE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Switching between PWM output and port output

DDR	OE	Description
0	0	Port input
	1	Port input
1	0	Port output or PWM 256/256 output
	1	PWM output (0 to 255/256 output)

PWDPRB—PWM Data Polarity Register B **H'FFD4** **PWM**
PWDPRA—PWM Data Polarity Register A **H'FFD5** **PWM**

Bit	7	6	5	4	3	2	1	0
PWDPRB	OS15	OS14	OS13	OS12	OS11	OS10	OS9	OS8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
PWDPRA	OS7	OS6	OS5	OS4	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM output polarity control

0	PWM direct output (PWDR value corresponds to high width of output)
1	PWM inverted output (PWDR value corresponds to low width of output)

PWSL—PWM Register Select **H'FFD6** **PWM**

Bit	7	6	5	4	3	2	1	0
	PWCKE	PWCKS	—	—	RS3	RS2	RS1	RS0
Initial value	0	0	1	0	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

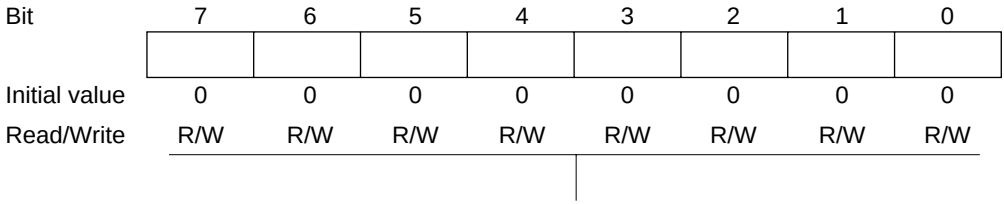
Register Select

0	0	0	0	PWDR0 selected
			1	PWDR1 selected
		1	0	PWDR2 selected
			1	PWDR3 selected
	1	0	0	PWDR4 selected
			1	PWDR5 selected
		1	0	PWDR6 selected
			1	PWDR7 selected
1	0	0	0	PWDR8 selected
			1	PWDR9 selected
		1	0	PWDR10 selected
			1	PWDR11 selected
	1	0	0	PWDR12 selected
			1	PWDR13 selected
		1	0	PWDR14 selected
			1	PWDR15 selected

PWM Clock Enable, PWM Clock Select

PWSL		PCSR		Description
Bit 7	Bit 6	Bit 2	Bit 1	
PWCKE	PWCKS	PWCKB	PWCKA	
0	—	—	—	Clock input disabled
1	0	—	—	ϕ (system clock) selected
	1	0	0	ϕ/2 selected
			1	ϕ/4 selected
		1	0	ϕ/8 selected
			1	ϕ/16 selected

PWDR0 to PWDR15—PWM Data Registers H'FFD7 PWM



ICCR0—I²C Bus Control Register 0 H'FFD8 IIC0

Bit	7	6	5	4	3	2	1	0
	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	W

Start Condition/Stop Condition Prohibit

0	Writing issues a start or stop condition, in combination with the BBSY flag
1	Reading always returns a value of 1; writing is invalid

I²C Bus Interface Interrupt Request Flag

0	Waiting for transfer, or transfer in progress
1	Interrupt requested

Note: For the clearing and setting conditions, see section 16.2.5, I²C Bus Control Register (ICCR).

Bus Busy

0	Bus is free [Clearing condition] When a stop condition is detected
1	Bus is busy [Setting condition] When a start condition is detected

Acknowledge Bit Judgement Select

0	Acknowledge bit is ignored and continuous transfer is performed
1	If acknowledge bit is 1, continuous transfer is interrupted

Master/Slave Select (MST), Transmit/Receive Select (TRS)

0	0	Slave receive mode
	1	Slave transmit mode
1	0	Master receive mode
	1	Master transmit mode

Note: For details see section 16.2.5, I²C Bus Control Register (ICCR).

I²C Bus Interface Interrupt Enable

0	Interrupt requests disabled
1	Interrupt requests enabled

I²C Bus Interface Enable

0	I²C bus interface module disabled, with SCL and SDA signal pins set to port function SAR and SARX can be accessed
1	I²C bus interface module enabled for transfer operations (pins SCL and SDA are driving the bus) ICMR and ICDR can be accessed

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

SMR0—Serial Mode Register 0 **H'FFD8** **SCI0**

Bit	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select 1 and 0

0	0	ϕ clock
	1	ϕ/4 clock
1	0	ϕ/16 clock
	1	ϕ/64 clock

Multiprocessor Mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop Bit Length

0	1 stop bit*1
1	2 stop bits*2

Notes: 1. In transmission, a single 1 bit (stop bit) is added to the end of a transmit character before it is sent.
2. In transmission, two 1 bits (stop bits) are added to the end of a transmit character before it is sent.

Parity Mode

0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is even.
2. When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled*

Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

Character Length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and LSB-first/MSB-first selection is not available.

Communication Mode

0	Asynchronous mode
1	Synchronous mode

BRR0—Bit Rate Register 0 **H'FFD9** **SCI0**

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

Sets the serial transmit/receive bit rate

ICSR0—I²C Bus Status Register 0 H'FFD9 IIC0

Bit	7	6	5	4	3	2	1	0
	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	R/W

Acknowledge Bit

0	Receive mode: 0 is output at acknowledge output timing Transmit mode: indicates that the receiving device has acknowledged the data (0 value)
1	Receive mode: 1 is output at acknowledge output timing Transmit mode: indicates that the receiving device has not acknowledged the data (1 value)

General Call Address Recognition Flag*2

0	General call address not recognized
1	General call address recognized

Slave Address Recognition Flag*2

0	Slave address or general call address not recognized
1	Slave address or general call address recognized

Arbitration Lost Flag*2

0	Bus arbitration won
1	Bus arbitration lost

Second Slave Address Recognition Flag*2

0	Second slave address not recognized
1	Second slave address recognized

I²C Bus Interface Continuous Transmission/Reception Interrupt Request Flag*2

0	Waiting for transfer, or transfer in progress
1	Continuous transfer state

Normal Stop Condition Detection Flag*2

0	No normal stop condition
1	In I²C bus format slave mode: Normal stop condition detected In other modes: No meaning

Error Stop Condition Detection Flag*2

0	No error stop condition
1	In I²C bus format slave mode: Error stop condition detected In other modes: No meaning

Notes: 1. Only 0 can be written, to clear the flag.
2. For the clearing and setting conditions, see section 16.2.6, I²C Bus Status Register (ICSR).

SCR0—Serial Control Register 0 H'FFDA SCI0

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable 1 and 0

0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output
		Synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input
		Synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input
		Synchronous mode	External clock/SCK pin functions as serial clock input

Transmit End Interrupt Enable

0	Transmit end interrupt (TEI) request disabled
1	Transmit end interrupt (TEI) request enabled

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled Receive interrupt (RXI) requests, receive error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received

Receive Enable

0	Reception disabled
1	Reception enabled

Transmit Enable

0	Transmission disabled
1	Transmission enabled

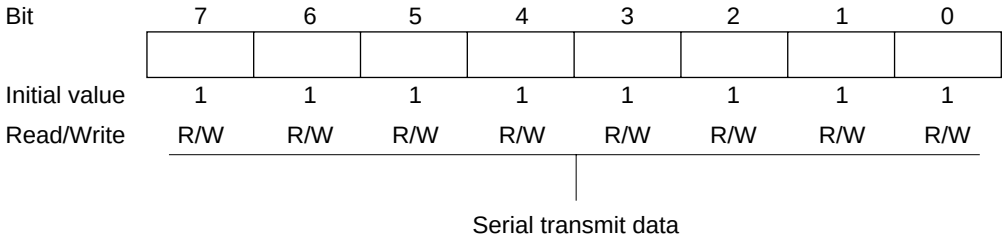
Receive Interrupt Enable

0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Transmit Interrupt Enable

0	Transmit data empty interrupt (TXI) request disabled
1	Transmit data empty interrupt (TXI) request enabled

TDR0—Transmit Data Register 0 **H'FFDB** **SCI0**



SSR0—Serial Status Register 0 **H'FFDC** **SCI0**

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor Bit Transfer

0	Data with a 0 multiprocessor bit is transmitted
1	Data with a 1 multiprocessor bit is transmitted

Multiprocessor Bit

0	[Clearing condition] When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Transmit End

0	[Clearing condition] When 0 is written in TDRE after reading TDRE = 1
1	[Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Parity Error

0	Clearing condition] When 0 is written in PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR

Framing Error

0	[Clearing condition] When 0 is written in FER after reading FER = 1
1	[Setting condition] When the SCI checks the stop bit at the end of the receive data when reception ends, and the stop bit is 0

Overrun Error

0	[Clearing condition] When 0 is written in ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1

Receive Data Register Full

0	[Clearing condition] When 0 is written in RDRF after reading RDRF = 1
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Transmit Data Register Empty

0	[Clearing condition] When 0 is written in TDRE after reading TDRE = 1
1	[Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written in TDR

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

RDR0—Receive Data Register 0 **H'FFDD** **SCI0**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Serial receive data

SCMR0—Serial Interface Mode Register 0 **H'FFDE** **SCI0**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	SMIF
Initial value	1	1	1	1	0	0	1	0
Read/Write	—	—	—	—	R/W	R/W	—	R/W

Serial Communication Interface Mode Select

0	Normal SCI mode
1	Setting prohibited

Data Invert

0	TDR contents are transmitted without modification Receive data is stored in RDR without modification
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Data Transfer Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

ICDR0—I²C Bus Data Register 0 **H'FFDE** **IIC0**

Bit	7	6	5	4	3	2	1	0
	ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ICDRR

Bit	7	6	5	4	3	2	1	0
	ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R	R	R	R	R	R	R	R

- ICDRS

Bit	7	6	5	4	3	2	1	0
	ICDRS7	ICDRS6	ICDRS5	ICDRS4	ICDRS3	ICDRS2	ICDRS1	ICDRS0
Initial value	—	—	—	—	—	—	—	—
Read/Write	—	—	—	—	—	—	—	—

- ICDRT

Bit	7	6	5	4	3	2	1	0
	ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0
Initial value	—	—	—	—	—	—	—	—
Read/Write	W	W	W	W	W	W	W	W

- TDRE, RDRF (internal flags)

Bit							—	—
							TDRE	RDRF
Initial value							0	0
Read/Write							—	—

Note: For details see section 16.2.1, I²C Bus Data Register (ICDR).

SARX0—Second Slave Address Register 0 **H'FFDE** **IIC0**

Bit	7	6	5	4	3	2	1	0
	SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Second Slave Address

Format Select

DDCSWR Bit 6	SAR Bit 0	SARX Bit 0	Operating Mode
SW	FS	FSX	
0	0	0	I²C bus format • SAR and SARX slave addresses recognized
		1	I²C bus format • SAR slave address recognized • SARX slave address ignored
	1	0	I²C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	0	0	Formatless mode (start/stop conditions not detected) • Acknowledge bit present
		1	
	1	0	
		1	Formatless mode* (start/stop conditions not detected) • No acknowledge bit

Note: * Do not select this mode when automatic switching to the I²C bus format is performed by means of a DDCSWR setting.

SAR0—Slave Address Register 0 **H'FFDF** **IIC0**

Bit	7	6	5	4	3	2	1	0
	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Slave Address

Format Select

DDCSWR Bit 6	SAR Bit 0	SARX Bit 0	Operating Mode
SW	FS	FSX	
0	0	0	I²C bus format • SAR and SARX slave addresses recognized
		1	I²C bus format • SAR slave address recognized • SARX slave address ignored
	1	0	I²C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	0	0	Formatless mode (start/stop conditions not detected) • Acknowledge bit present
		1	
	1	0	
		1	Formatless mode* (start/stop conditions not detected) • No acknowledge bit

Note: * Do not select this mode when automatic switching to the I²C bus format is performed by means of a DDCSWR setting.

ICMR0—I²C Bus Mode Register 0 **H'FFDF** **IIC0**

Bit	7	6	5	4	3	2	1	0
	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit Counter

BC2	BC1	BC0	Synchronous Serial Format	I²C Bus Format
0	0	0	8	9
		1	1	2
	1	0	2	3
		1	3	4
1	0	0	4	5
		1	5	6
	1	0	6	7
		1	7	8

Transfer Clock Select

IICX	CKS2	CKS1	CKS0	Clock
0	0	0	0	φ/28
			1	φ/40
		1	0	φ/48
			1	φ/64
	1	0	0	φ/80
			1	φ/100
		1	0	φ/112
			1	φ/128
1	0	0	0	φ/56
			1	φ/80
		1	0	φ/96
			1	φ/128
	1	0	0	φ/160
			1	φ/200
		1	0	φ/224
			1	φ/256

Wait Insertion Bit

0	Data and acknowledge transferred consecutively
1	Wait inserted between data and acknowledge

MSB-First/LSB-First Select*

0	MSB-first
1	LSB-first

Note: * Do not set this bit to 1 when using the I²C bus format.

ADDRAH—A/D Data Register AH	**H'FFE0**	**A/D**
ADDRAL—A/D Data Register AL	**H'FFE1**	**A/D**
ADDRBH—A/D Data Register BH	**H'FFE2**	**A/D**
ADDRBL—A/D Data Register BL	**H'FFE3**	**A/D**
ADDRCH—A/D Data Register CH	**H'FFE4**	**A/D**
ADDRCL—A/D Data Register CL	**H'FFE5**	**A/D**
ADDRDH—A/D Data Register DH	**H'FFE6**	**A/D**
ADDRDL—A/D Data Register DL	**H'FFE7**	**A/D**

Correspondence between analog input channels and ADDR registers

Analog Input Channel		A/D Data Register
Group 0	Group 1	
AN_0	AN_4	ADDRA
AN_1	AN_5	ADDRB
AN_2	AN_6	ADDRC
AN_3	AN_7	ADDRD

ADCSR—A/D Control/Status Register **H'FFE8** **A/D Converter**

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
H8/3577 Group / H8/3567 Group	0	0	0	AN_0	AN_0
			1	AN_1	AN_0, AN_1
		1	0	AN_2	AN_0, AN_1, AN_2
			1	AN_3	AN_0, AN_1, AN_2, AN_3
H8/3577 Group only	1	0	0	AN_4	AN_4
			1	AN_5	AN_4, AN_5
		1	0	AN_6	AN_4, AN_5, AN_6
			1	AN_7	AN_4, AN_5, AN_6, AN_7

Clock Select

0	Conversion time = 266 states (max.)
1	Conversion time = 134 states (max.)

Scan Mode

0	Single mode
1	Scan mode

A/D Start

0	A/D conversion stopped
1	• Single mode: A/D conversion is started. Cleared to 0 automatically when conversion on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues consecutively on the selected channels until ADST is cleared to 0 by software, a reset, or a transition to standby mode or module stop mode

A/D Interrupt Enable

0	A/D conversion end interrupt (ADI) request disabled
1	A/D conversion end interrupt (ADI) request enabled

A/D end flag

0	[Clearing condition] When 0 is written in ADF after reading ADF = 1
1	[Setting conditions] • Single mode: When A/D conversion ends • Scan mode: When A/D conversion ends on all specified channels

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
	TRGS1	TRGS0	—	—	—	—	—	—
Initial value	0	0	1	1	1	1	1	1
Read/Write	R/W	R/W	—	—	—	—	—	—

Timer Trigger Select

0	0	Start of A/D conversion by external trigger is disabled
	1	Start of A/D conversion by external trigger is disabled
1	0	Start of A/D conversion by external trigger (8-bit timer) is enabled
	1	Start of A/D conversion by external trigger pin is enabled

TCRX—Timer Control Register X **H'FFF0** **TMRX**
TCRY—Timer Control Register Y **H'FFF0** **TMRY**

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Counter Clear 1 and 0

0	0	Clearing is disabled
	1	Cleared on compare-match A
1	0	Cleared on compare-match B
	1	Cleared on rising edge of external reset input

Timer Overflow Interrupt Enable

0	OVF interrupt request (OVI) is disabled
1	OVF interrupt request (OVI) is enabled

Compare-Match Interrupt Enable A

0	CMFA interrupt request (CMIA) is disabled
1	CMFA interrupt request (CMIA) is enabled

Compare-Match Interrupt Enable B

0	CMFB interrupt request (CMIB) is disabled
1	CMFB interrupt request (CMIB) is enabled

Clock Select 2 to 0

Channel	Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Description
0	0	0	0	Clock input disabled
			1*1	φ/8 internal clock source, counted on falling edge
				φ/2 internal clock source, counted on falling edge
		1	0*1	φ/64 internal clock source, counted on falling edge
				φ/32 internal clock source, counted on falling edge
			1*1	φ/1024 internal clock source, counted on falling edge
				φ/256 internal clock source, counted on falling edge
	1	0	0	Counted on TCNT1 overflow signal*2
1	0	0	0	Clock input disabled
			1*1	φ/8 internal clock source, counted on falling edge
				φ/2 internal clock source, counted on falling edge
		1	0*1	φ/64 internal clock source, counted on falling edge
				φ/128 internal clock source, counted on falling edge
			1*1	φ/1024 internal clock source, counted on falling edge
				φ/2048 internal clock source, counted on falling edge
	1	0	0	Counted on TCNT0 compare-match A*2
X	0	0	0	Clock input disabled
			1	Counted on φ internal clock source
		1	0	φ/2 internal clock source, counted on falling edge
			1	φ/4 internal clock source, counted on falling edge
	1	0	0	Clock input disabled
Y	0	0	0	Clock input disabled
			1	φ/4 internal clock source, counted on falling edge
		1	0	φ/256 internal clock source, counted on falling edge
			1	φ/2048 internal clock source, counted on falling edge
	1	0	0	Clock input disabled
Common	1	0	1	External clock source, counted on rising edge
		1	0	External clock source, counted on falling edge
			1	External clock source, counted on both rising and falling edges

Notes: 1. Selected by ICKS1 and ICKS0 in STCR. For details see section 12.2.4, Timer Control Register (TCR).
2. If the count input of channel 0 is the TCNT1 overflow signal and that of channel 1 is the TCNT0 compare-match signal, no incrementing clock will be generated. Do not use this setting.

TCSRX—Timer Control/Status Register X **H'FFF1** **TMRX**

TCSRX

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ICF	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W

Output Select 1 and 0

0	0	No change when compare-match A occurs
	1	0 output when compare-match A occurs
1	0	1 output when compare-match A occurs
	1	Output inverted when compare-match A occurs (toggle output)

Output Select 3 and 2

0	0	No change when compare-match B occurs
	1	0 output when compare-match B occurs
1	0	1 output when compare-match B occurs
	1	Output inverted when compare-match B occurs (toggle output)

Input Capture Flag

0	[Clearing condition] When 0 is written in ICF after reading ICF = 1
1	[Setting condition] When a rising edge followed by a falling edge is detected in the external reset signal after the ICST bit in TCONRI has been set to 1

Timer Overflow Flag

0	[Clearing condition] When 0 is written in OVF after reading OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00

Compare-Match Flag A

0	[Clearing condition] When 0 is written in CMFA after reading CMFA = 1
1	[Setting condition] When TCNT = TCORA

Compare-Match Flag B

0	[Clearing condition] When 0 is written in CMFB after reading CMFB = 1
1	[Setting condition] When TCNT = TCORB

Note: * Only 0 can be written in bits 7 to 4, to clear the flags.

TCSRY—Timer Control/Status Register Y **H'FFF1** **TMRY**

TCSRY

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	ICIE	OS3	OS2	OS1	OS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W	R/W

Output Select 1 and 0

0	0	No change when compare-match A occurs
	1	0 output when compare-match A occurs
1	0	1 output when compare-match A occurs
	1	Output inverted when compare-match A occurs (toggle output)

Output Select 3 and 2

0	0	No change when compare-match B occurs
	1	0 output when compare-match B occurs
1	0	1 output when compare-match B occurs
	1	Output inverted when compare-match B occurs (toggle output)

Input Capture Interrupt Enable

0	ICF interrupt request (ICIX) is disabled
1	ICF interrupt request (ICIX) is enabled

Timer Overflow Flag

0	[Clearing condition] When 0 is written in OVF after reading OVF = 1
1	[Setting condition] When TCNT overflows from H'FF to H'00

Compare-Match Flag A

0	[Clearing condition] When 0 is written in CMFA after reading CMFA = 1
1	[Setting condition] When TCNT = TCORA

Compare-Match Flag B

0	[Clearing condition] When 0 is written in CMFB after reading CMFB = 1
1	[Setting condition] When TCNT = TCORB

Note: * Only 0 can be written in bits 7 to 5, to clear the flags.

TICRR—Input Capture Register R **H'FFF2** **TMRX**
TICRF—Input Capture Register F **H'FFF3** **TMRX**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Stores TCNT value at fall of external reset input

TCORAY—Time Constant Register AY **H'FFF2** **TMRY**
TCORBY—Time Constant Register BY **H'FFF3** **TMRY**

TCORAY, TCORBY

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

Compare-match flag (CMF) is set when TCOR and TCNT values match

TCNTX—Timer Counter X **H'FFF4** **TMRX**
TCNTY—Timer Counter Y **H'FFF4** **TMRY**

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

Up-counter

TISR—Timer Input Select Register **H'FFF5** **TMRY**

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

Input Select

0	IVG signal is selected
1	TMIY (TMCIY/TMRIY) is selected

TCORC—Time Constant Register C **H'FFF5** **TMRX**
TCORAX—Time Constant Register AX **H'FFF6** **TMRX**
TCORBX—Time Constant Register BX **H'FFF7** **TMRX**

TCORC

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

Compare-match C signal is generated when sum of TCORC and TICR contents match TCNT value

TCORAX, TCORBX

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

Compare-match flag (CMF) is set when TCOR and TCNT values match

TCONRI—Timer Connection Register I **H'FFFC** **Timer Connection**

Bit	7	6	5	4	3	2	1	0
	SIMOD1	SIMOD0	SCONE	ICST	HFINV	VFINV	HIINV	VIINV
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Input Synchronization Signal Inversion

0	The VSYNCI pin state is used directly as the VSYNCI input
1	The VSYNCI pin state is inverted before use as the VSYNCI input

Input Synchronization Signal Inversion

0	The HSYNCI and CSYNCI pin states are used directly as the HSYNCI and CSYNCI inputs
1	The HSYNCI and CSYNCI pin states are inverted before use as the HSYNCI and CSYNCI inputs

Input Synchronization Signal Inversion

0	The VFBACKI pin state is used directly as the VFBACKI input
1	The VFBACKI pin state is inverted before use as the VFBACKI input

Input Synchronization Signal Inversion

0	The HFBACKI pin state is used directly as the HFBACKI input
1	The HFBACKI pin state is inverted before use as the HFBACKI input

Input Capture Start Bit

0	The TICRR and TICRF input capture functions are stopped [Clearing condition] When a rising edge followed by a falling edge is detected on TMRIX
1	The TICRR and TICRF input capture functions are operating (Waiting for detection of a rising edge followed by a falling edge on TMRIX) [Setting condition] When 1 is written in ICST after reading ICST = 0

Synchronization Signal Connection Enable

SCONE	Mode	FTIA	FTIB	FTIC	FTID	TMCI1	TMRI1
0	Normal connection	FTIA input	FTIB input	FTIC input	FTID input	TMCI1 input	TMRI1 input
1	Synchronization signal connec-tion mode	IVI signal	TMO1 signal	VFBACKI input	IHI signal	IHI signal	IVI inverse signal

Input Synchronization Mode Select 1 and 0

SIMOD1	SIMOD0	Mode	IHI Signal	IVI Signal
0	0	No signal	HFBACKI input	VFBACKI input
	1	S-on-G mode	CSYNCI input	PDC input
1	0	Composite mode	HSYNCI input	PDC input
	1	Separate mode	HSYNCI input	VSYNCI input

TCONRO—Timer Connection Register O **H'FFFD** **Timer Connection**

Bit	7	6	5	4	3	2	1	0
	HOE	VOE	CLOE	CBOE	HOINV	VOINV	CLOINV	CBOINV
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output Synchronization Signal Inversion

0	The CBLANK signal is used directly as the CBLANK output
1	The CBLANK signal is inverted before use as the CBLANK output

Output Synchronization Signal Inversion

0	The CLO signal (CL_1, CL_2, CL_3, or CL_4 signal) is used directly as the CLAMPO output
1	The CLO signal (CL_1, CL_2, CL_3, or CL_4 signal) is inverted before use as the CLAMPO output

Output Synchronization Signal Inversion

0	The IVO signal is used directly as the VSYNCO output
1	The IVO signal is inverted before use as the VSYNCO output

Output Synchronization Signal Inversion

0	The IHO signal is used directly as the HSYNCO output
1	The IHO signal is inverted before use as the HSYNCO output

Output Enable

0	[H8/3577 Group] The $P2_7$/PW_{15}/CBLANK pin functions as the $P2_7$/PW_{15} pin [H8/3567 Group] The $P1_5$/PW_5/CBLANK pin functions as the $P1_5$/PW_5 pin
1	[H8/3577 Group] The $P2_7$/PW_{15}/CBLANK pin functions as the CBLANK pin [H8/3567 Group] The $P1_5$/PW_5/CBLANK pin functions as the CBLANK pin

Output Enable

0	The $P6_4$/FTIC/TMO_0/CLAMPO pin functions as the $P6_4$/FTIC/TMO_0 pin
1	The $P6_4$/FTIC/TMO_0/CLAMPO pin functions as the CLAMPO pin

Output Enable

0	The $P6_1$/FTOA/VSYNCO pin functions as the $P6_1$/FTOA pin
1	The $P6_1$/FTOA/VSYNCO pin functions as the VSYNCO pin

Output Enable

0	The $P6_7$/TMO_1/TMOX/HSYNCO pin functions as the $P6_7$/TMO_1/TMOX pin
1	The $P6_7$/TMO_1/TMOX/HSYNCO pin functions as the HSYNCO pin

TCONRS—Timer Connection Register S **H'FFFE** **Timer Connection**

Bit	7	6	5	4	3	2	1	0
	TMRX/Y	ISGENE	HOMOD1	HOMOD0	VOMOD1	VOMOD0	CLMOD1	CLMOD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clamp Waveform Mode Select 1 and 0

ISGENE	CLMOD1	CLMOD0	Description
0	0	0	The CL_1 signal is selected
		1	The CL_2 signal is selected
	1	0	The CL_3 signal is selected
		1	
1	0	0	The CL_4 signal is selected
		1	
	1	0	
		1	

Vertical Synchronization Output Mode Select 1 and 0

ISGENE	VOMOD1	VOMOD0	Description
0	0	0	The IVI signal (without fall modification or IHI synchronization) is selected
		1	The IVI signal (without fall modification, with IHI synchronization) is selected
	1	0	The IVI signal (with fall modification, without IHI synchronization) is selected
		1	The IVI signal (with fall modification and IHI synchronization) is selected
1	0	0	The IVG signal is selected
		1	
	1	0	
		1	

Horizontal Synchronization Output Mode Select 1 and 0

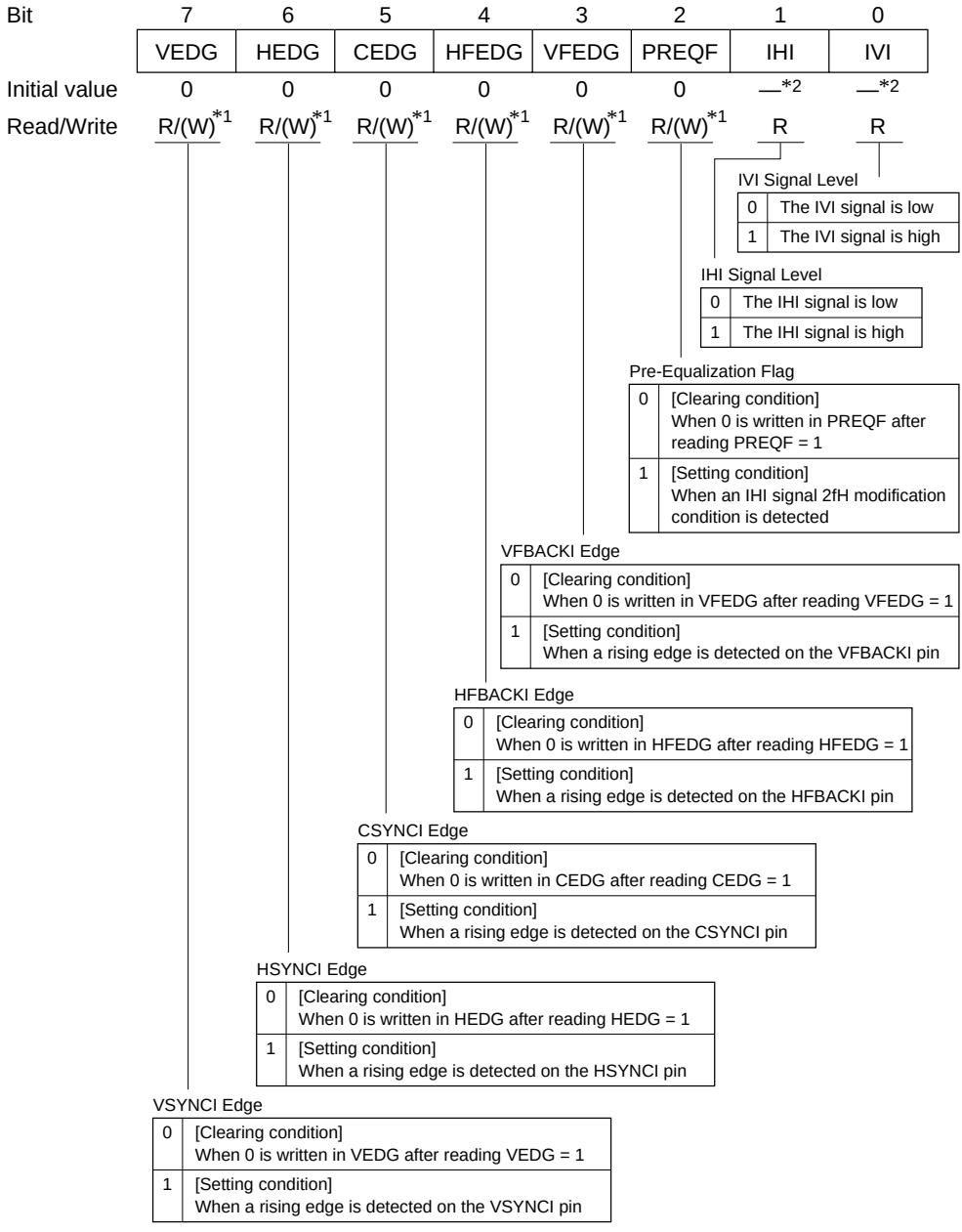
ISGENE	HOMOD1	HOMOD0	Description
0	0	0	The IHI signal (without 2fH modification) is selected
		1	The IHI signal (with 2fH modification) is selected
	1	0	The CL_1 signal is selected
		1	
1	0	0	The IHG signal is selected
		1	
	1	0	
		1	

Internal Synchronization Signal Select

TMRX/TMRY Access Select

0	The TMRX registers are accessed at addresses H'FFF0 to H'FFF5
1	The TMRY registers are accessed at addresses H'FFF0 to H'FFF5

SEDGR—Edge Sense Register **H'FFFF** **Timer Connection**



Notes: 1. Only 0 can be written, to clear the flag.
2. The initial value is undefined since it depends on the pin states.

Appendix C I/O Port Block Diagrams

C.1 Port 1 Block Diagrams

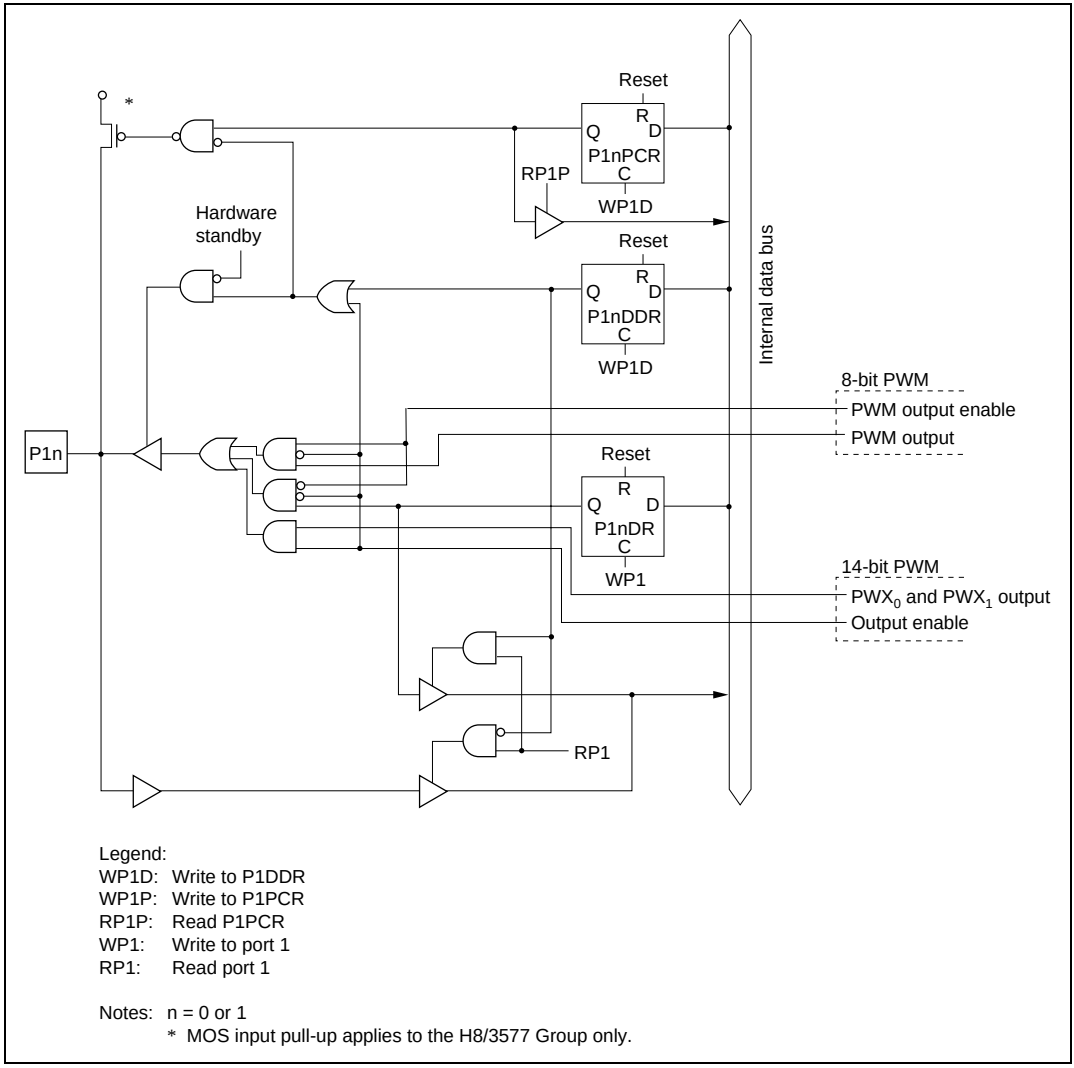


Figure C.1 Port 1 Block Diagram (Pins $P1_0$ and $P1_1$)

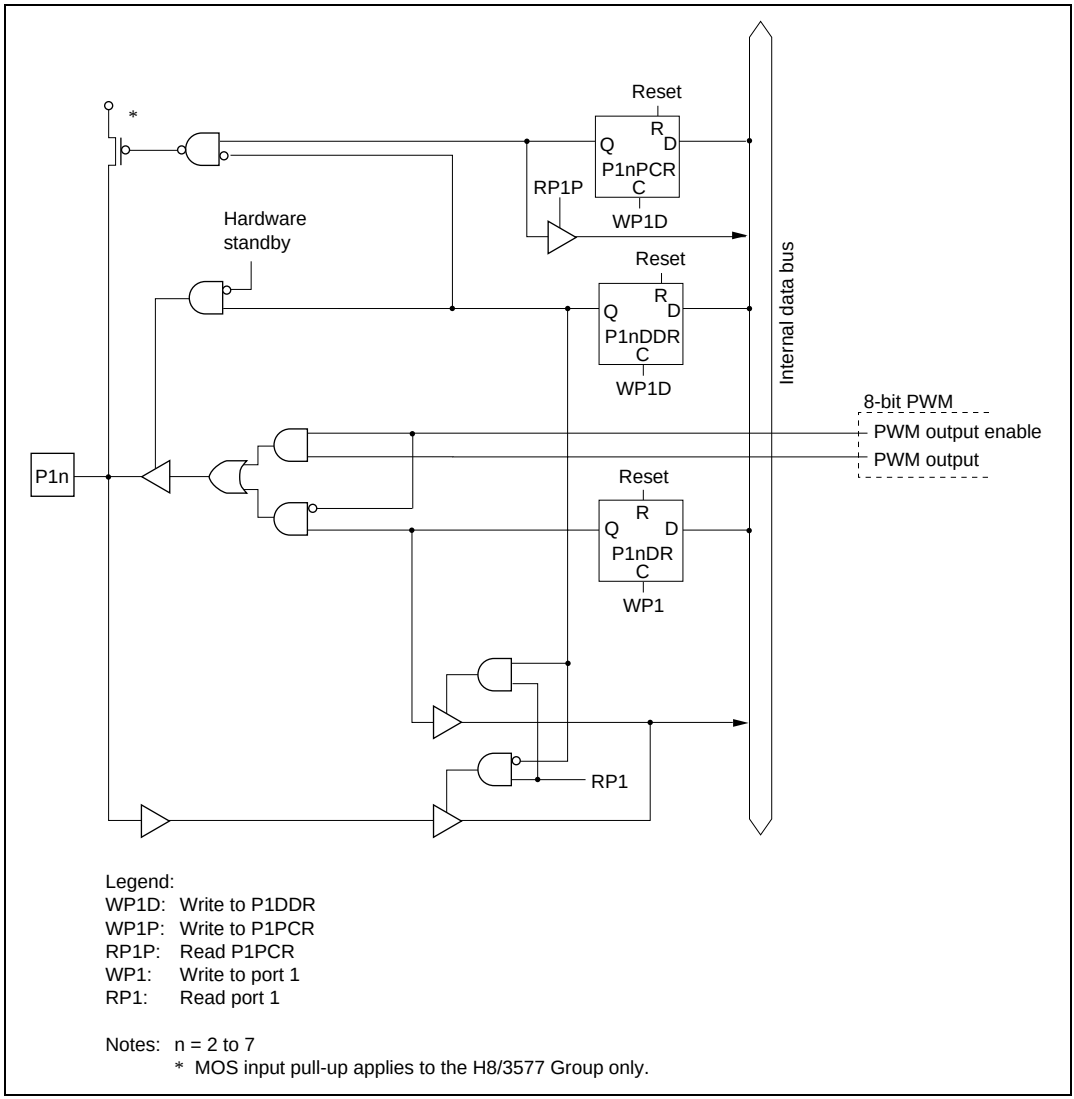


Figure C.2 Port 1 Block Diagram
(Pins $P1_2$ to $P1_7$ in H8/3577 Group, Pins $P1_2$ to $P1_4$ in H8/3567 Group)

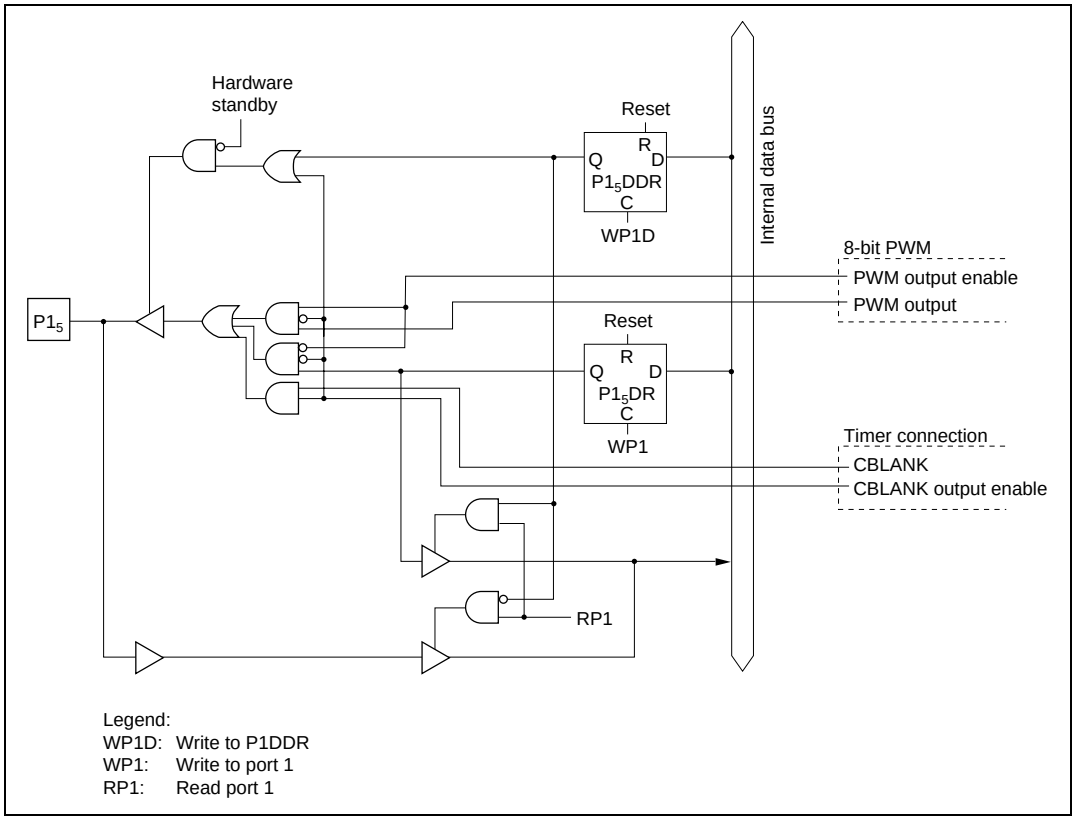


Figure C.3 Port 1 Block Diagram (Pin P1$_5$ in H8/3567 Group)

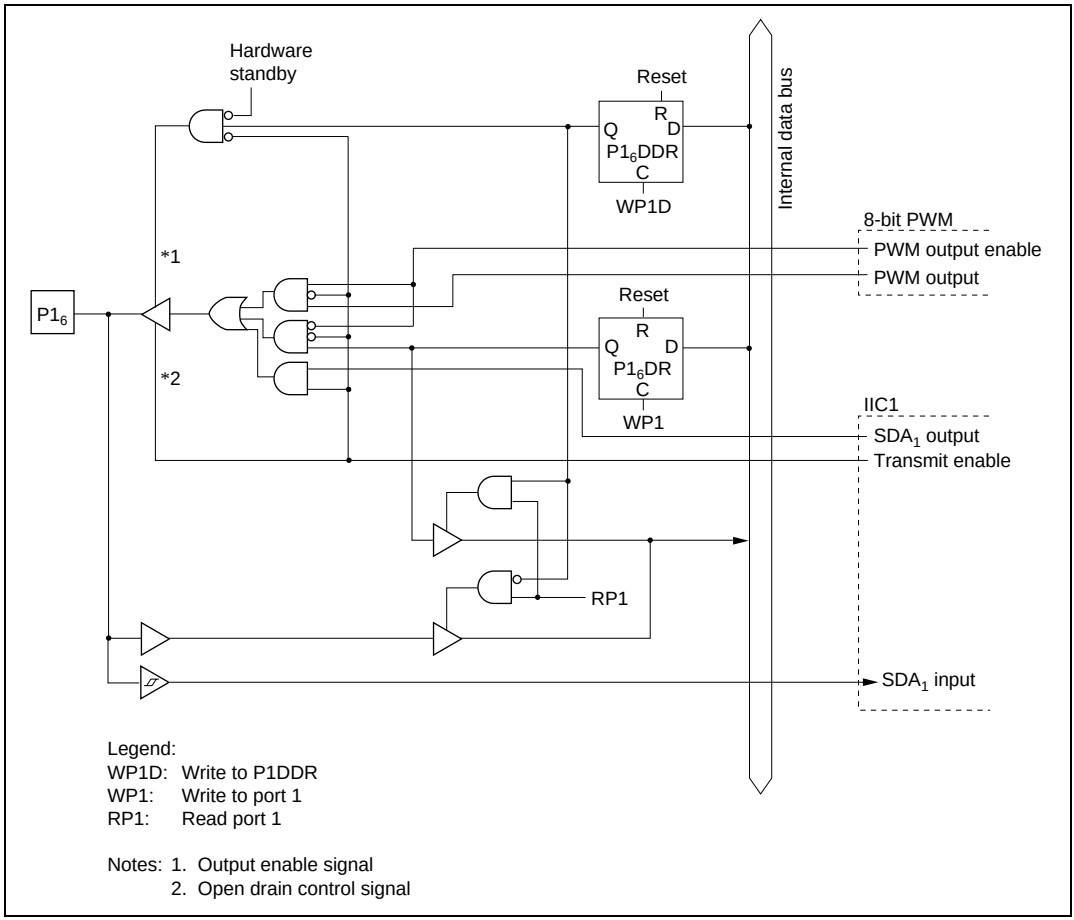


Figure C.4 Port 1 Block Diagram (Pin P1$_6$ in H8/3567 Group)

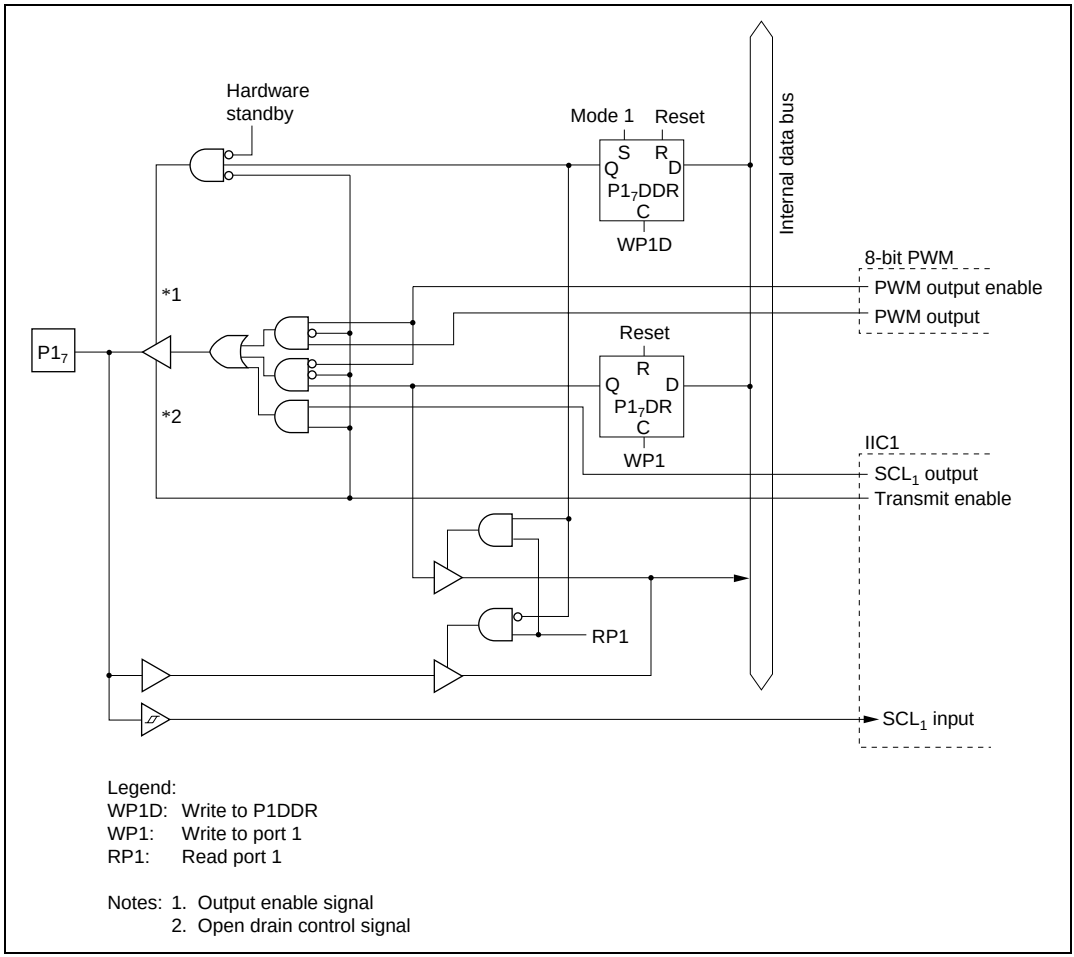


Figure C.5 Port 1 Block Diagram (Pin P1$_7$ in H8/3567 Group)

C.2 Port 2 Block Diagrams

Port 2 is provided only in the H8/3577 Group, and not in the H8/3567 Group.

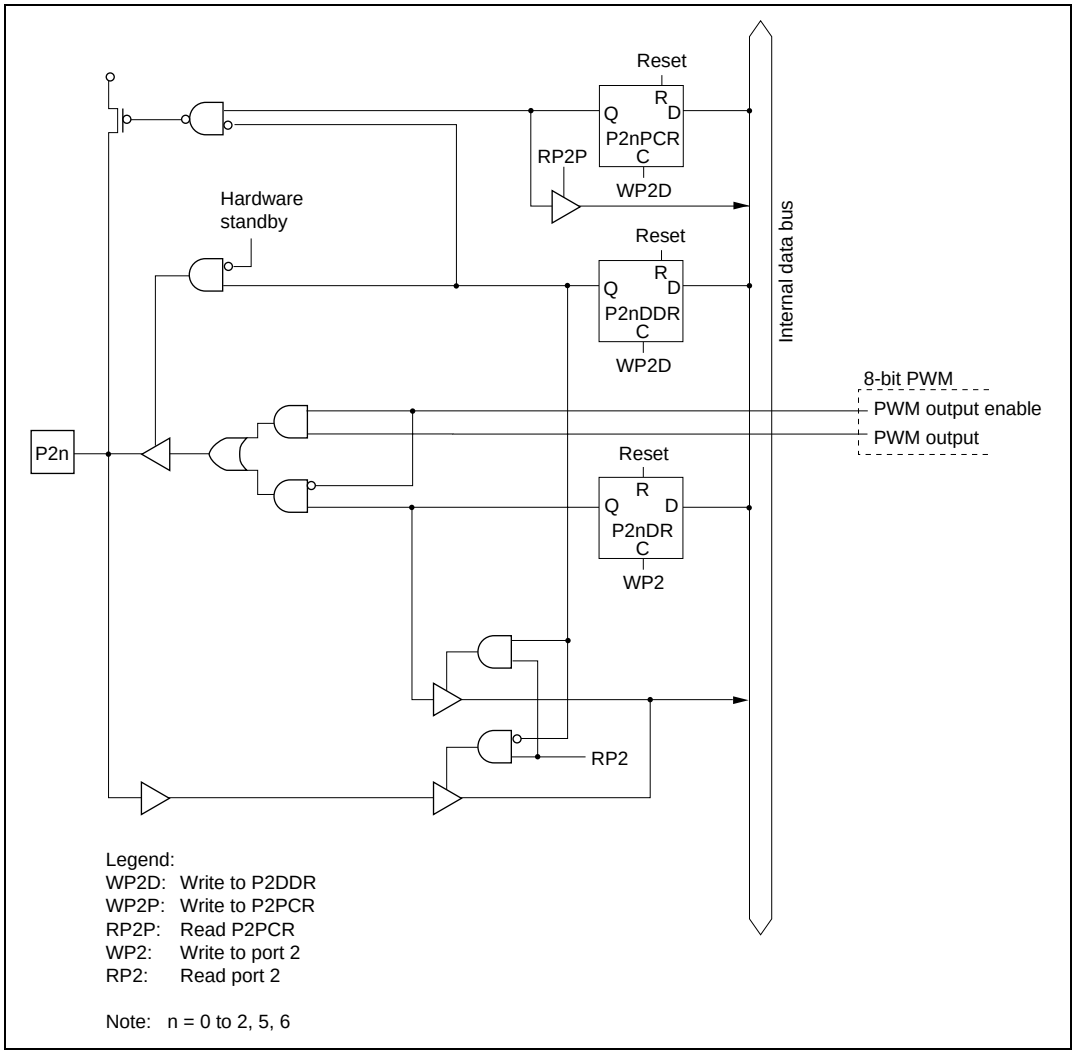


Figure C.6 Port 2 Block Diagram (Pins $P2_0$ to $P2_2$, $P2_5$, and $P2_6$ in H8/3577 Group)

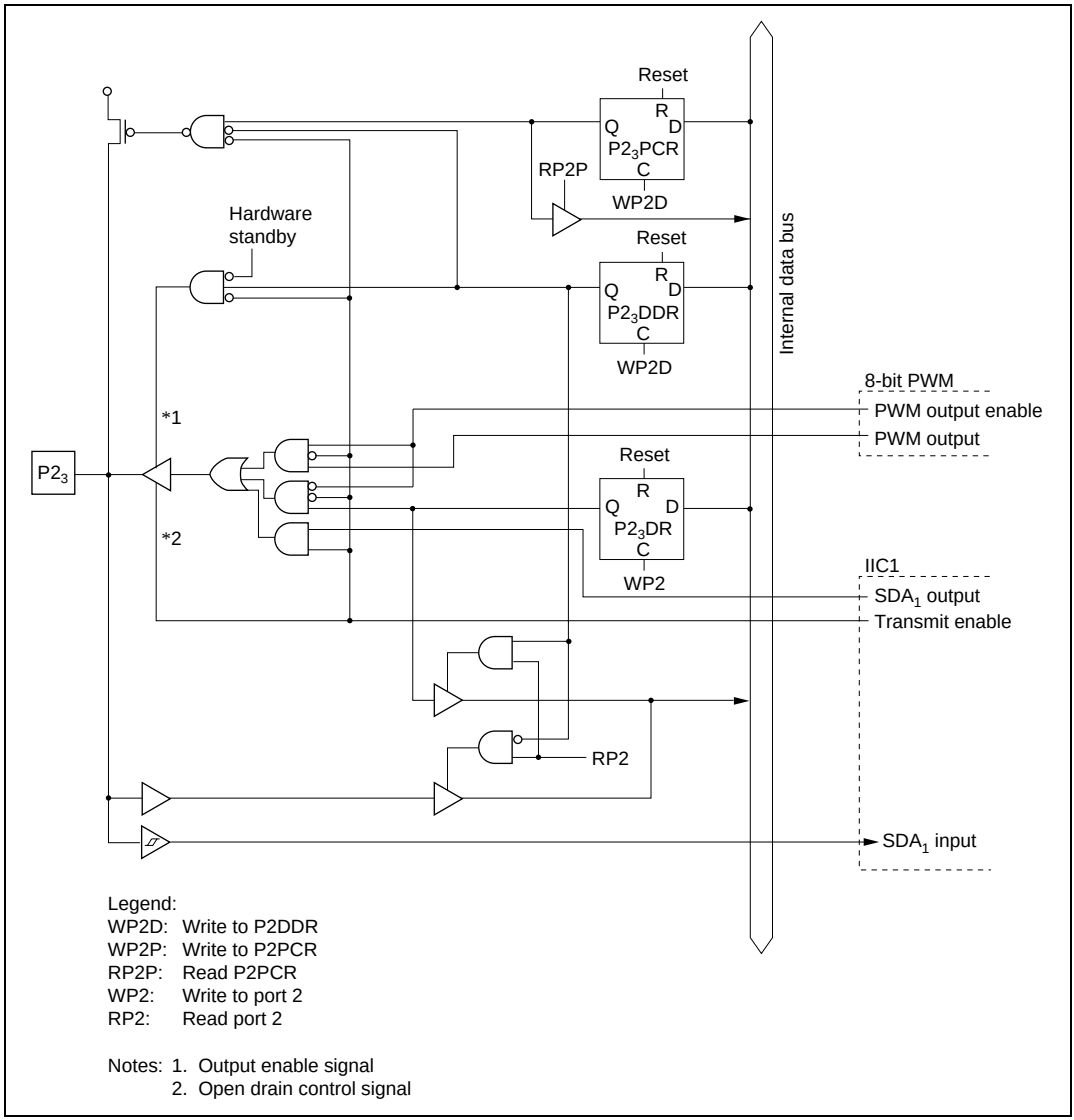


Figure C.7 Port 2 Block Diagram (Pin P2$_3$ in H8/3577 Group)

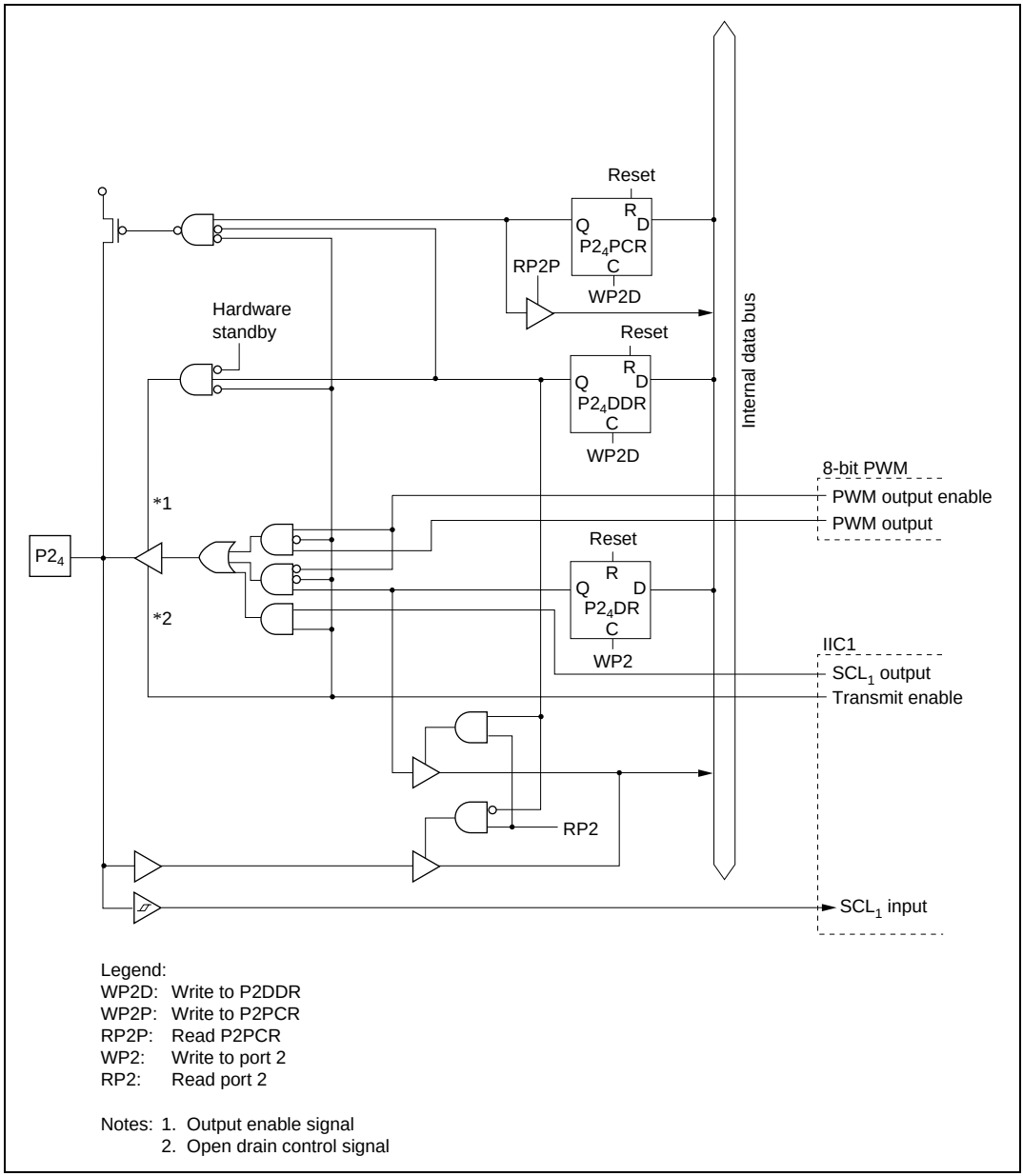


Figure C.8 Port 2 Block Diagram (Pin P2$_4$ in H8/3577 Group)

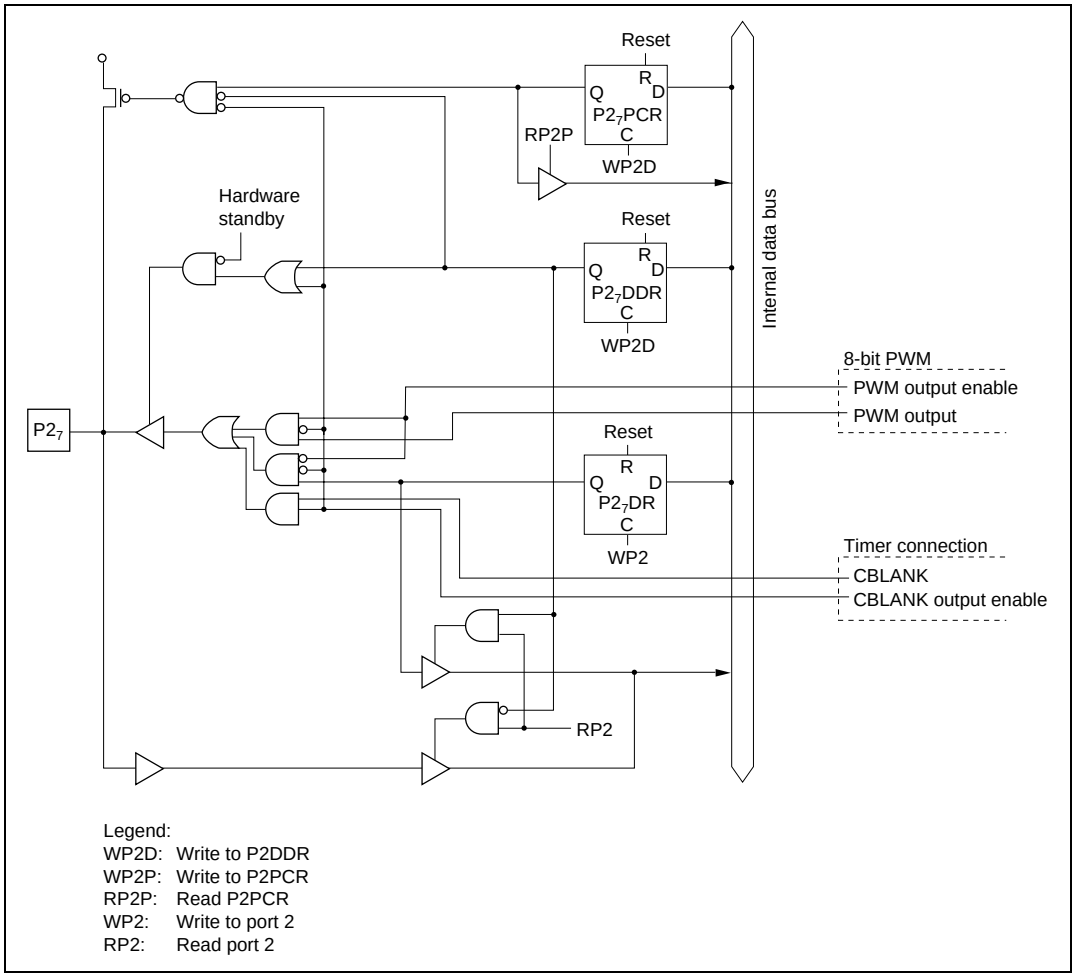


Figure C.9 Port 2 Block Diagram (Pin P2$_7$ in H8/3577 Group)

C.3 Port 3 Block Diagram

Port 3 is provided only in the H8/3577 Group, and not in the H8/3567 Group.

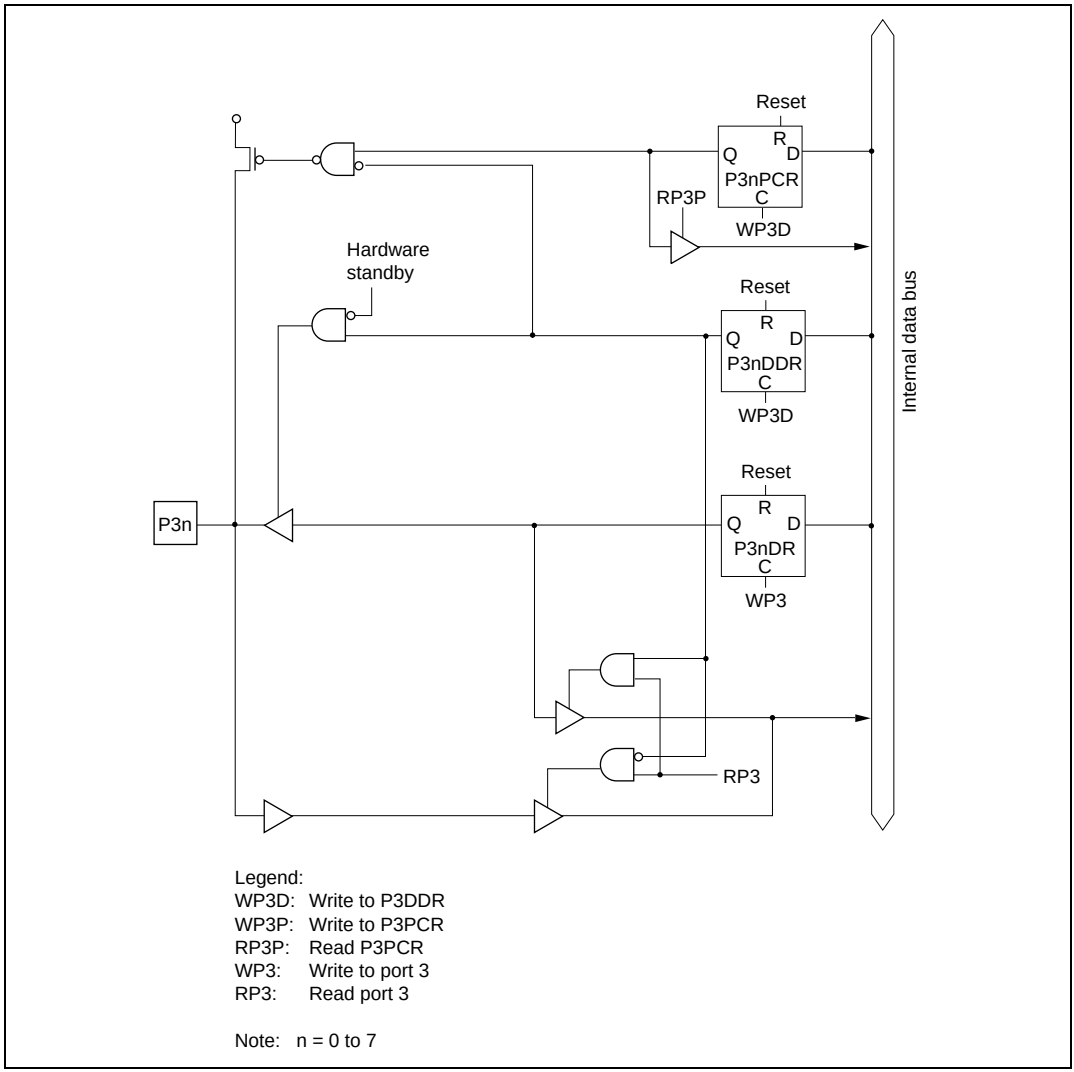


Figure C.10 Port 3 Block Diagram (Pins $P3_0$ to $P3_7$ in H8/3577 Group)

C.4 Port 4 Block Diagrams

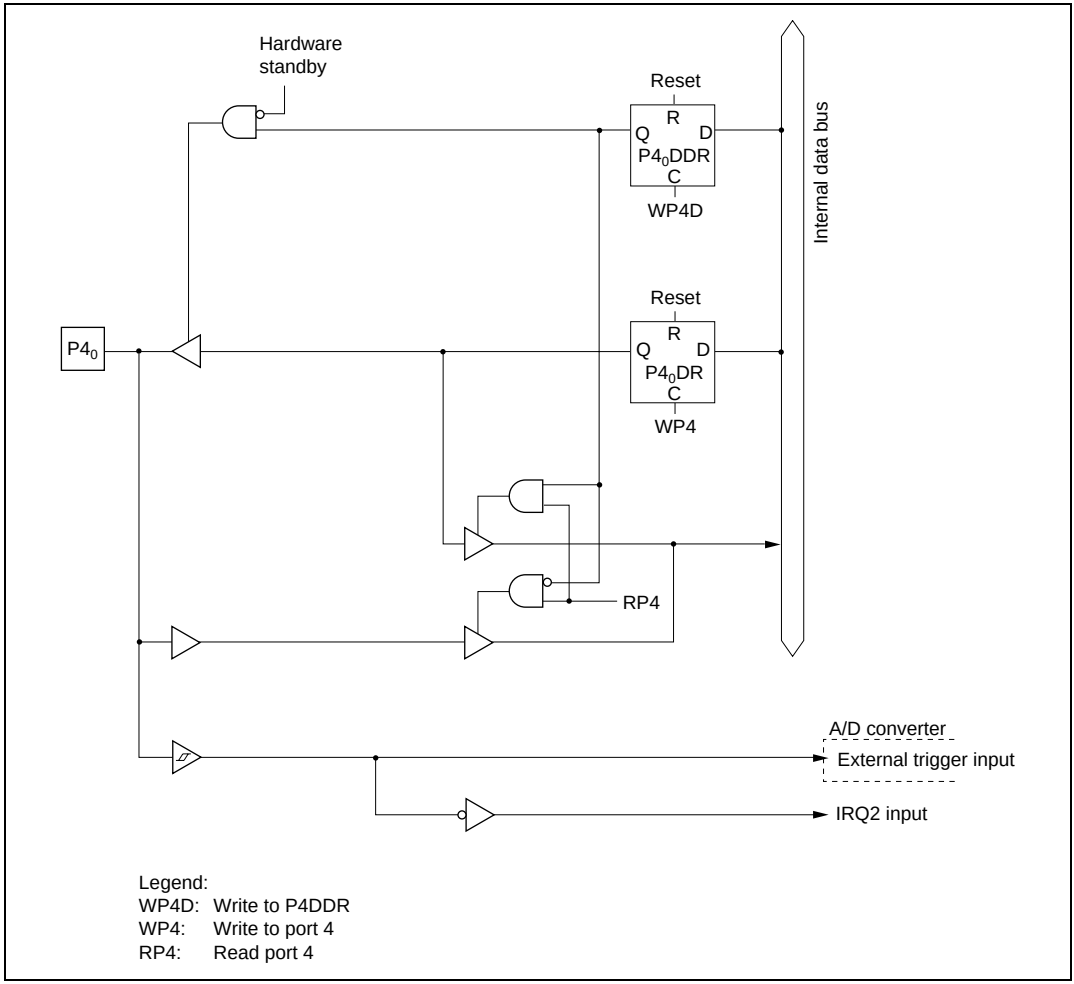


Figure C.11 Port 4 Block Diagram (Pin P4$_0$)

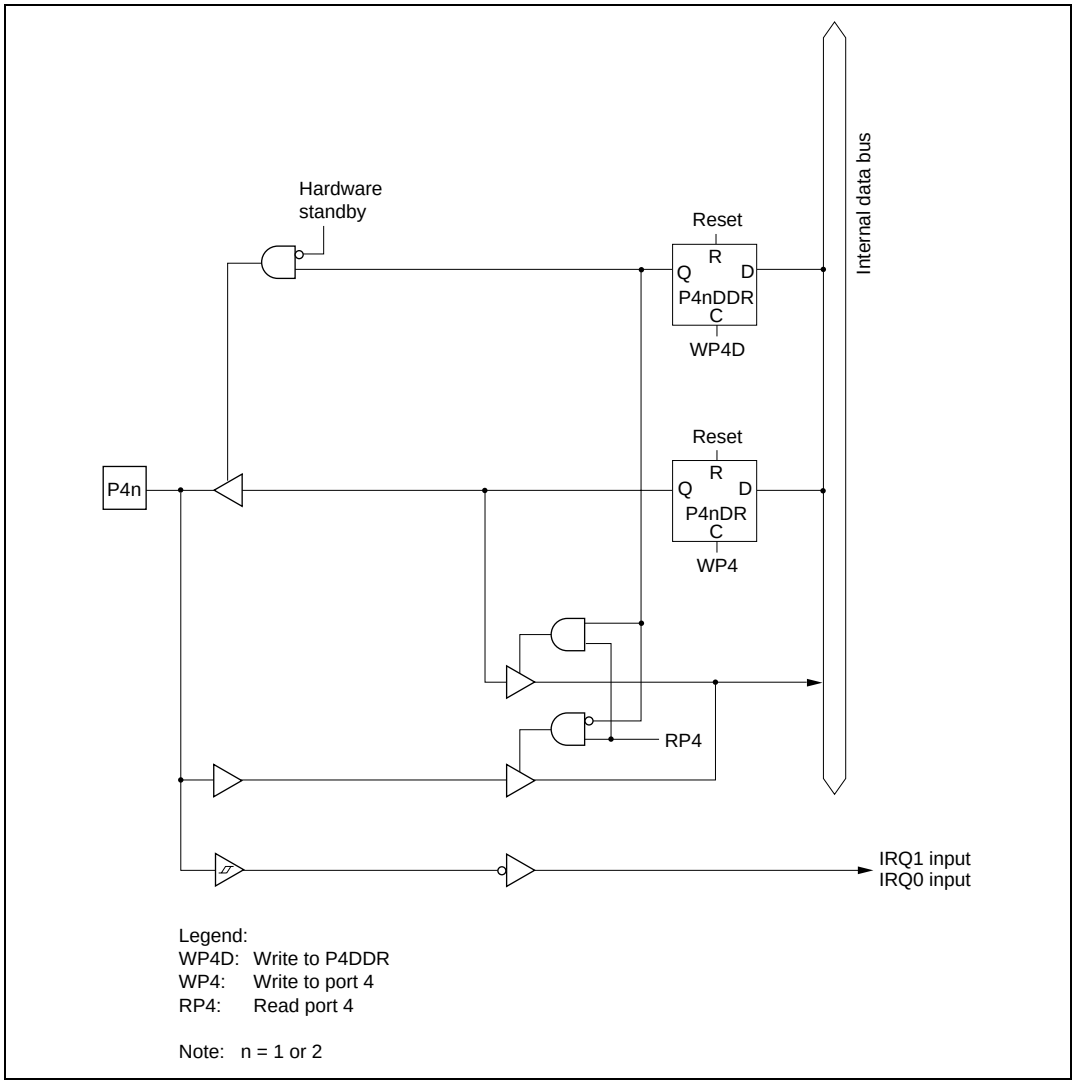


Figure C.12 Port 4 Block Diagram (Pins $P4_1$ and $P4_2$)

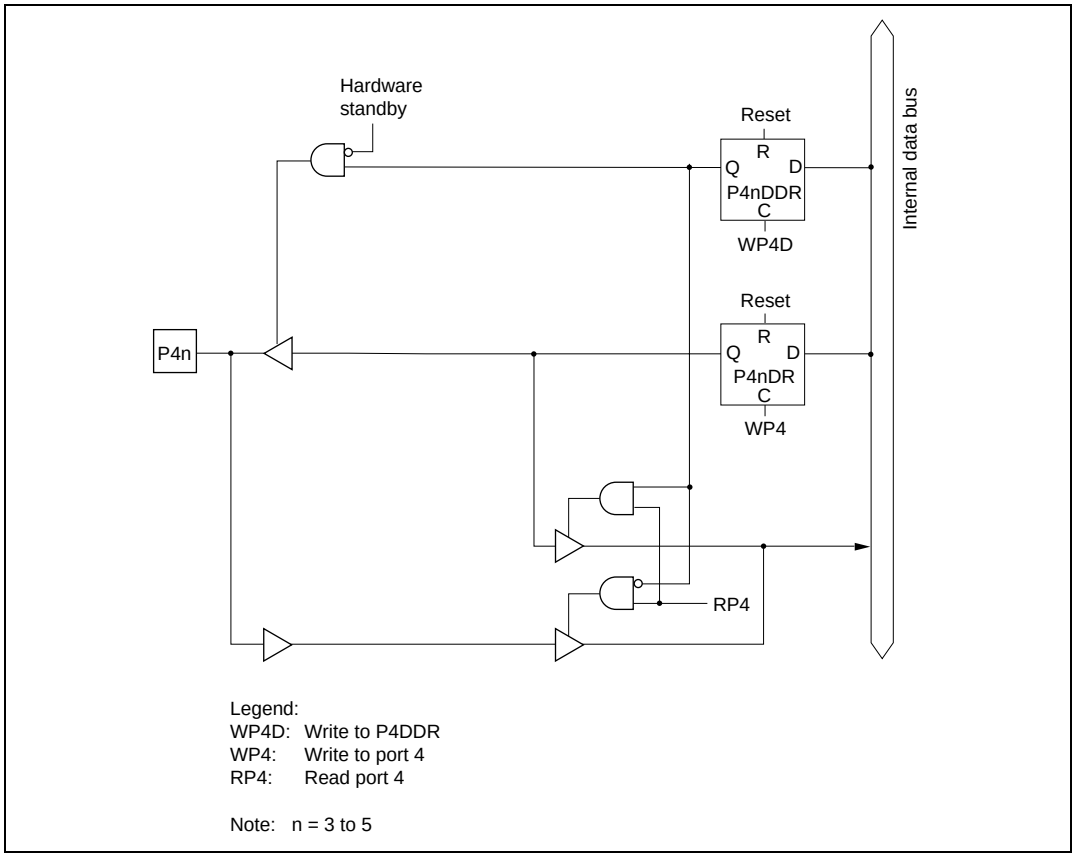


Figure C.13 Port 4 Block Diagram (Pins $P4_3$ to $P4_5$)

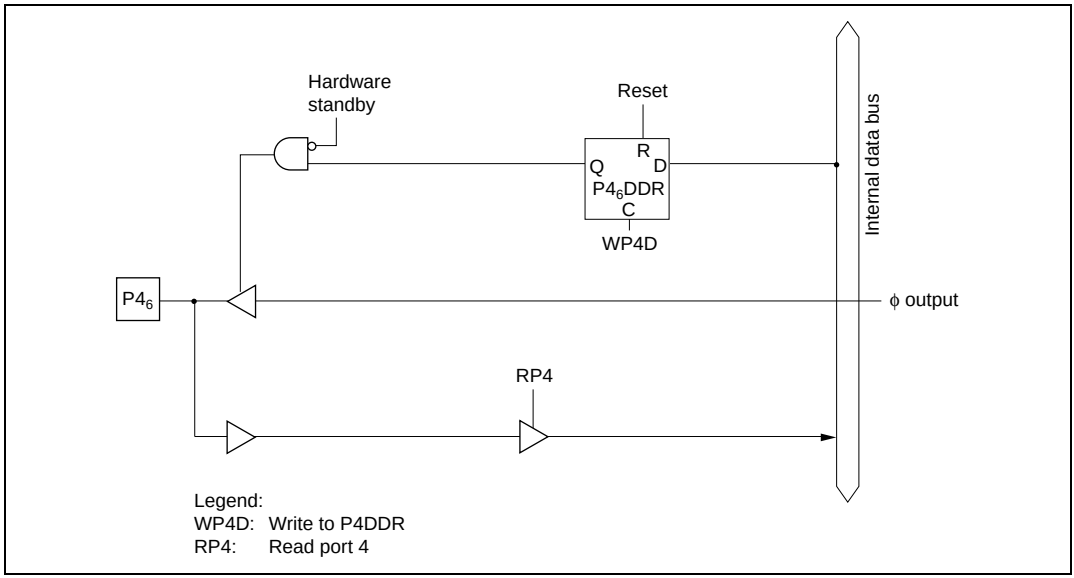
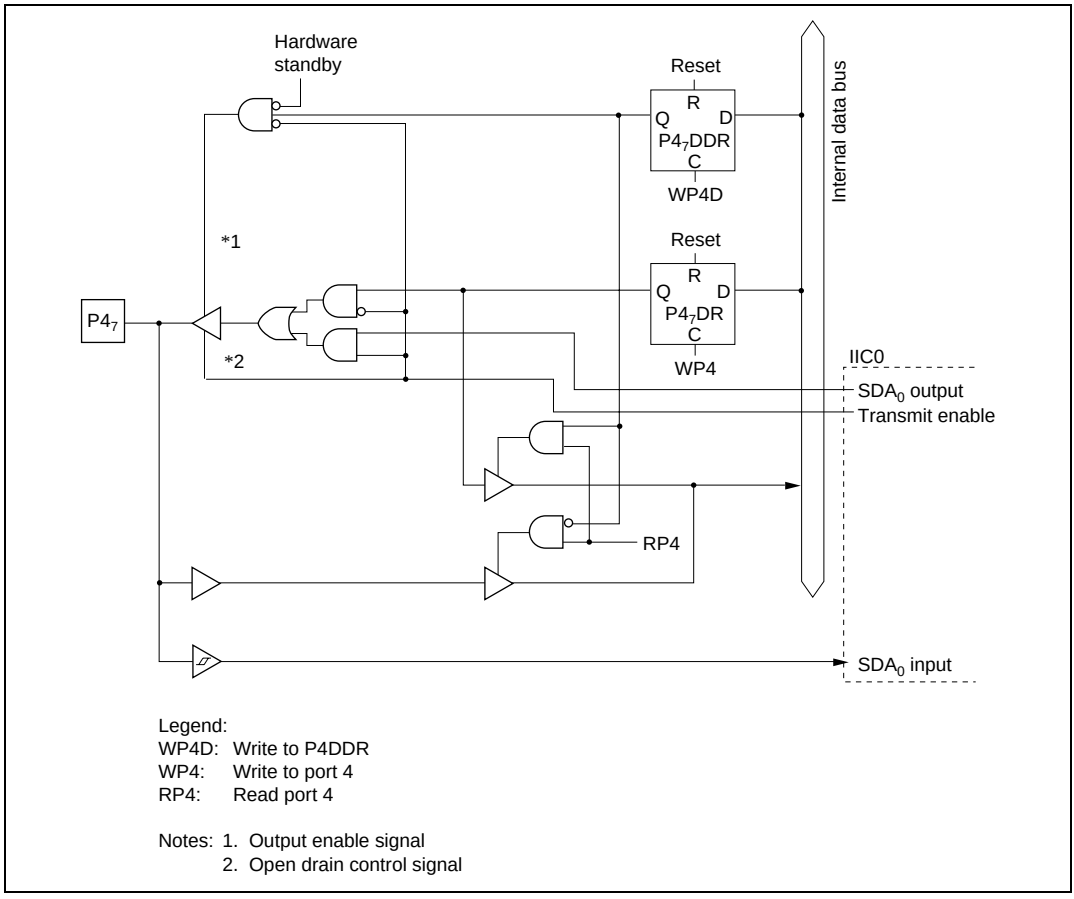


Figure C.14 Port 4 Block Diagram (Pin P4$_6$)



Legend:
WP4D: Write to P4DDR
WP4: Write to port 4
RP4: Read port 4

Notes: 1. Output enable signal
2. Open drain control signal

Figure C.15 Port 4 Block Diagram (Pin $P4_7$)

C.5 Port 5 Block Diagrams

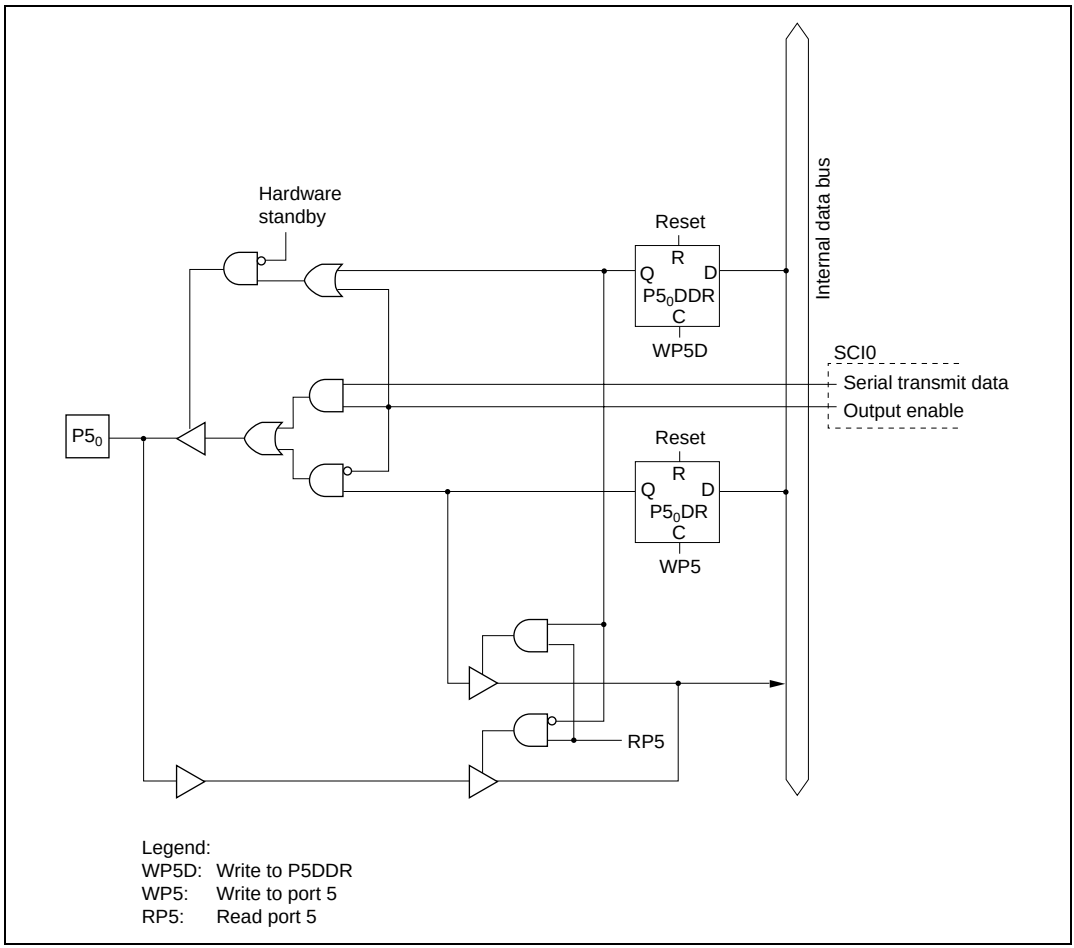


Figure C.16 Port 5 Block Diagram (Pin P5$_0$)

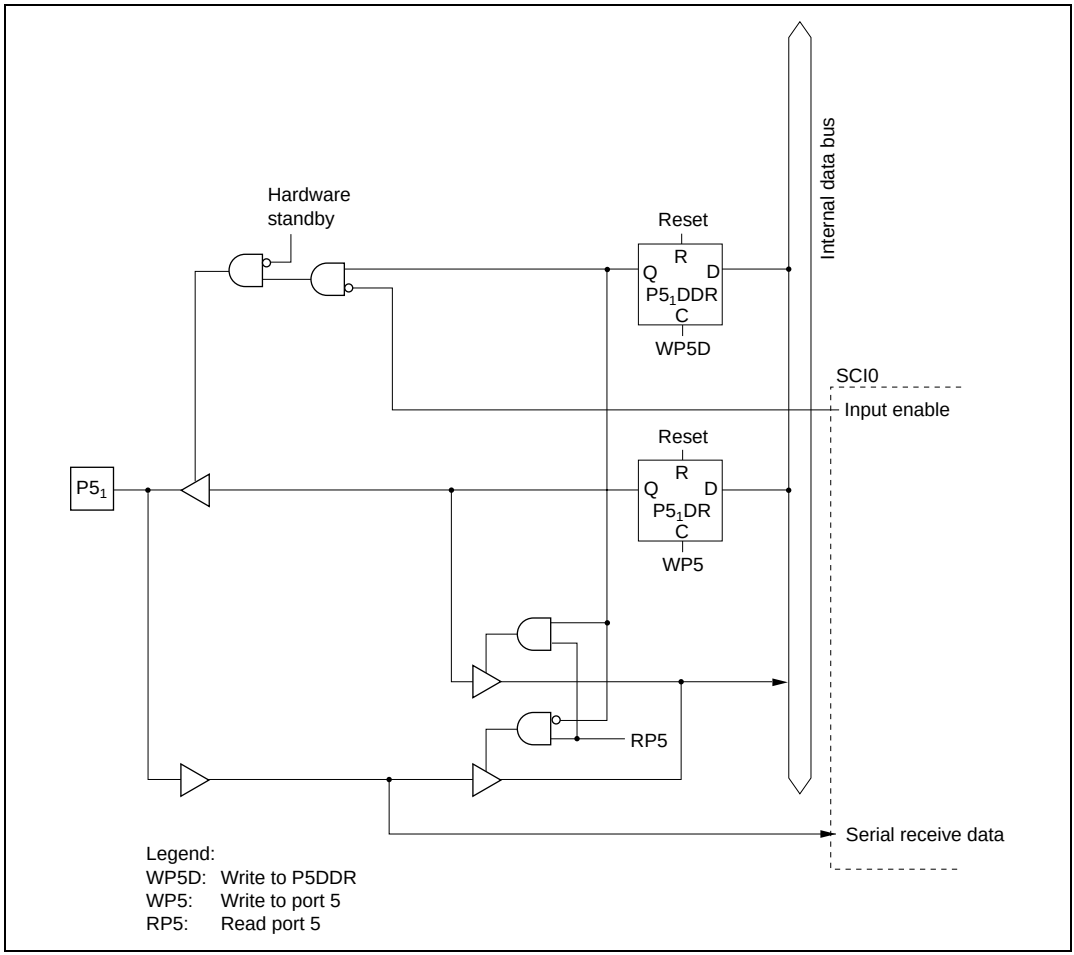


Figure C.17 Port 5 Block Diagram (Pin $P5_1$)

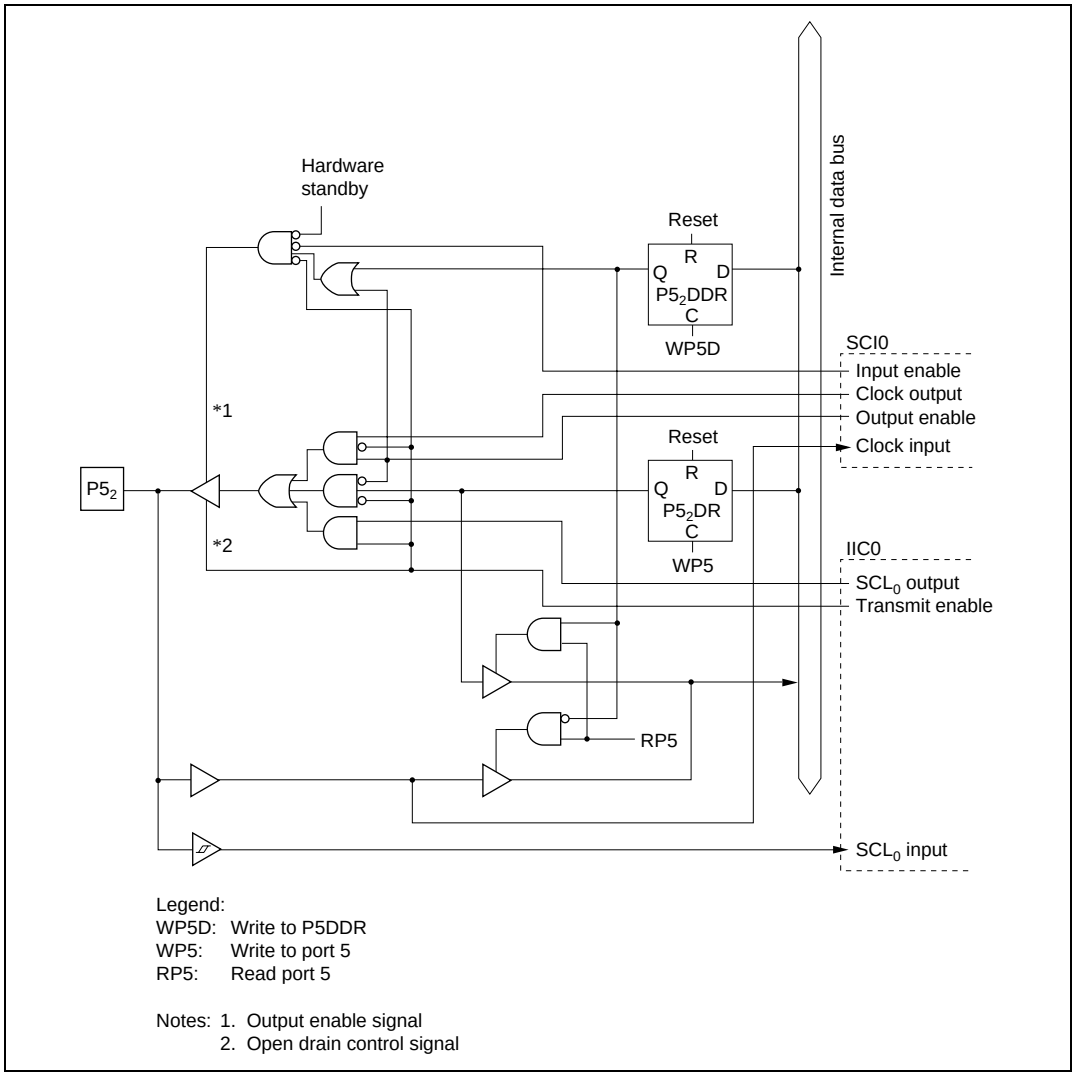


Figure C.18 Port 5 Block Diagram (Pin P5$_2$)

C.6 Port 6 Block Diagrams

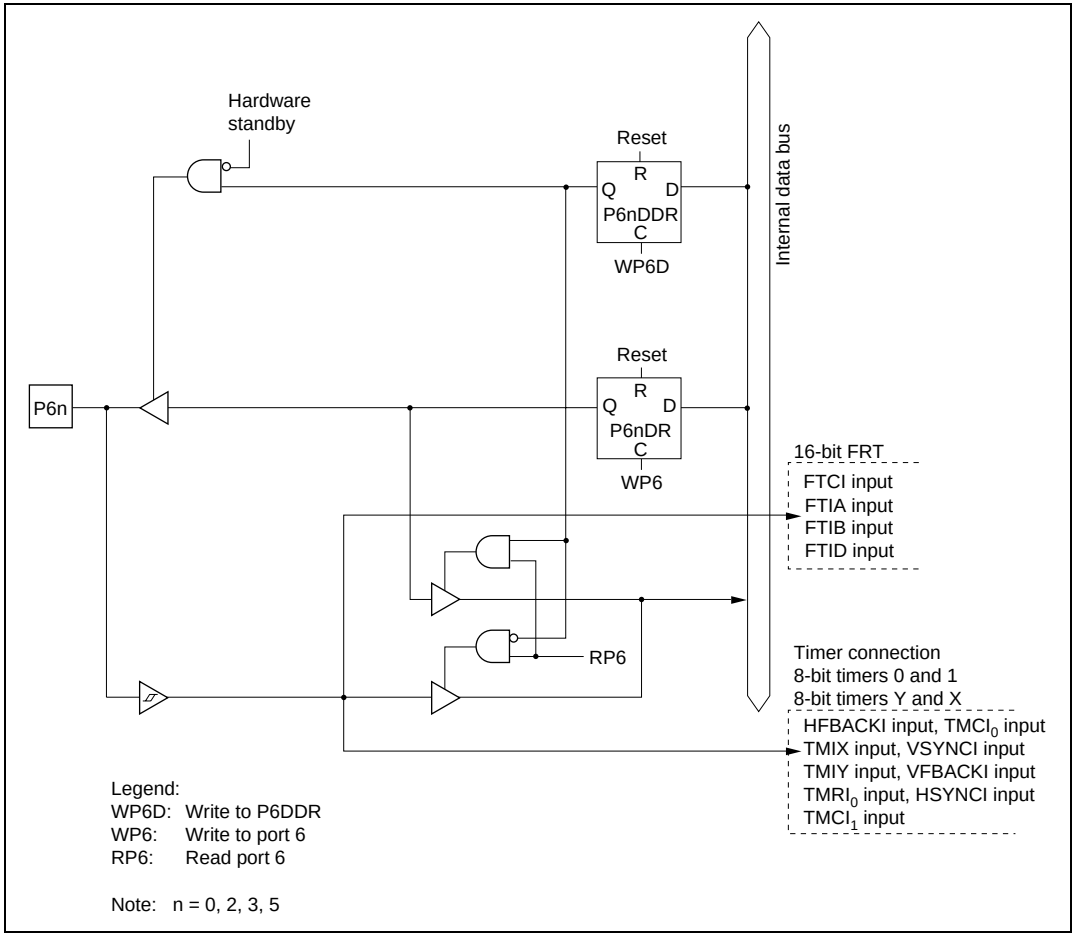


Figure C.19 Port 6 Block Diagram (Pins $P6_0$, $P6_2$, $P6_3$, and $P6_5$)

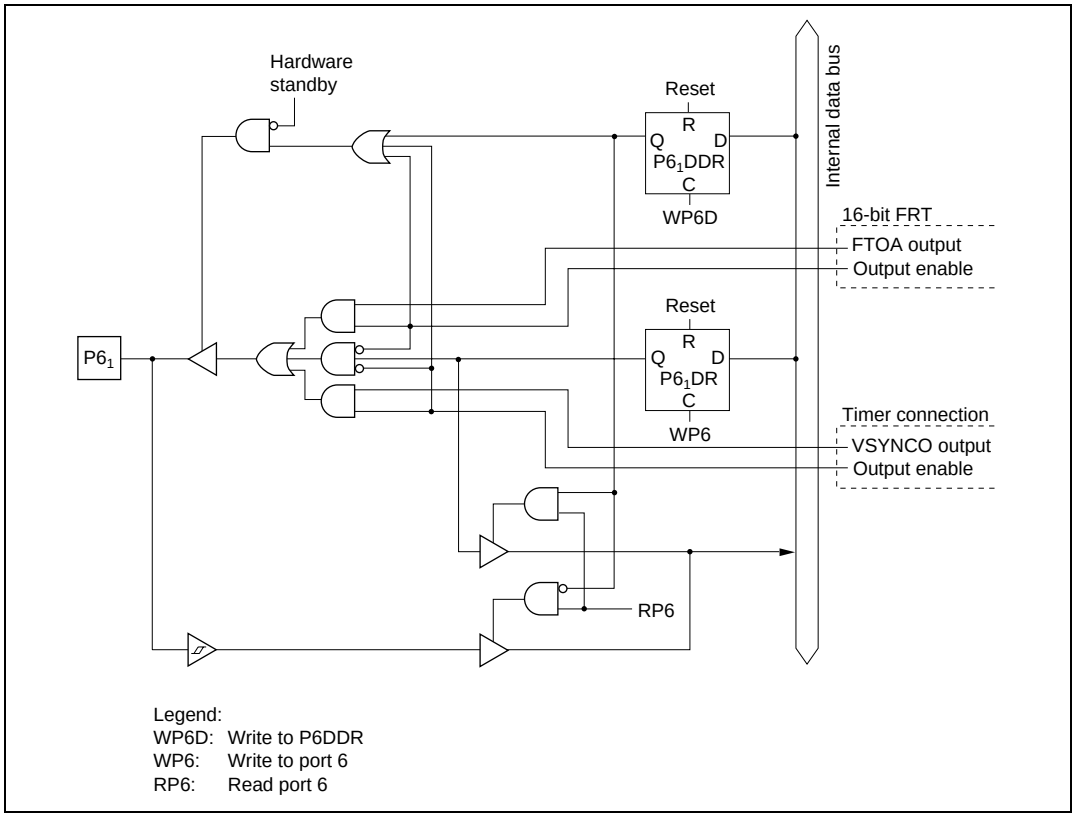


Figure C.20 Port 6 Block Diagram (Pin P6$_1$)

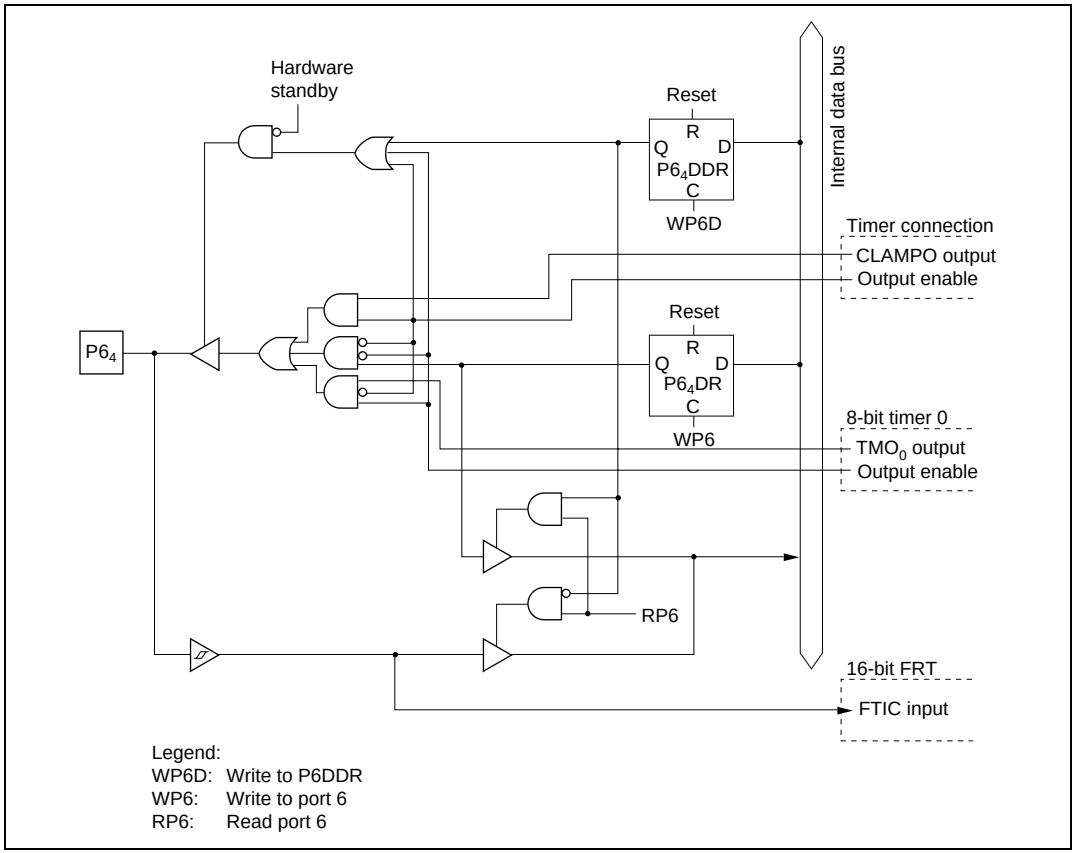


Figure C.21 Port 6 Block Diagram (Pin P6$_4$)

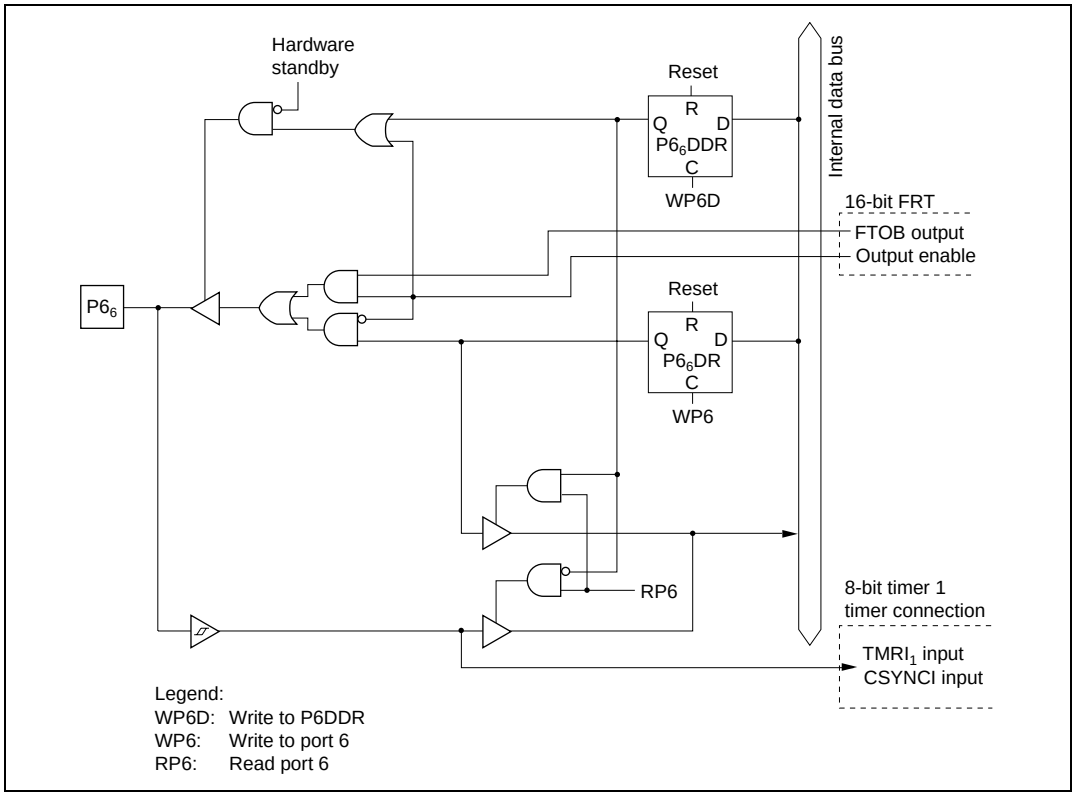


Figure C.22 Port 6 Block Diagram (Pin $P6_6$)

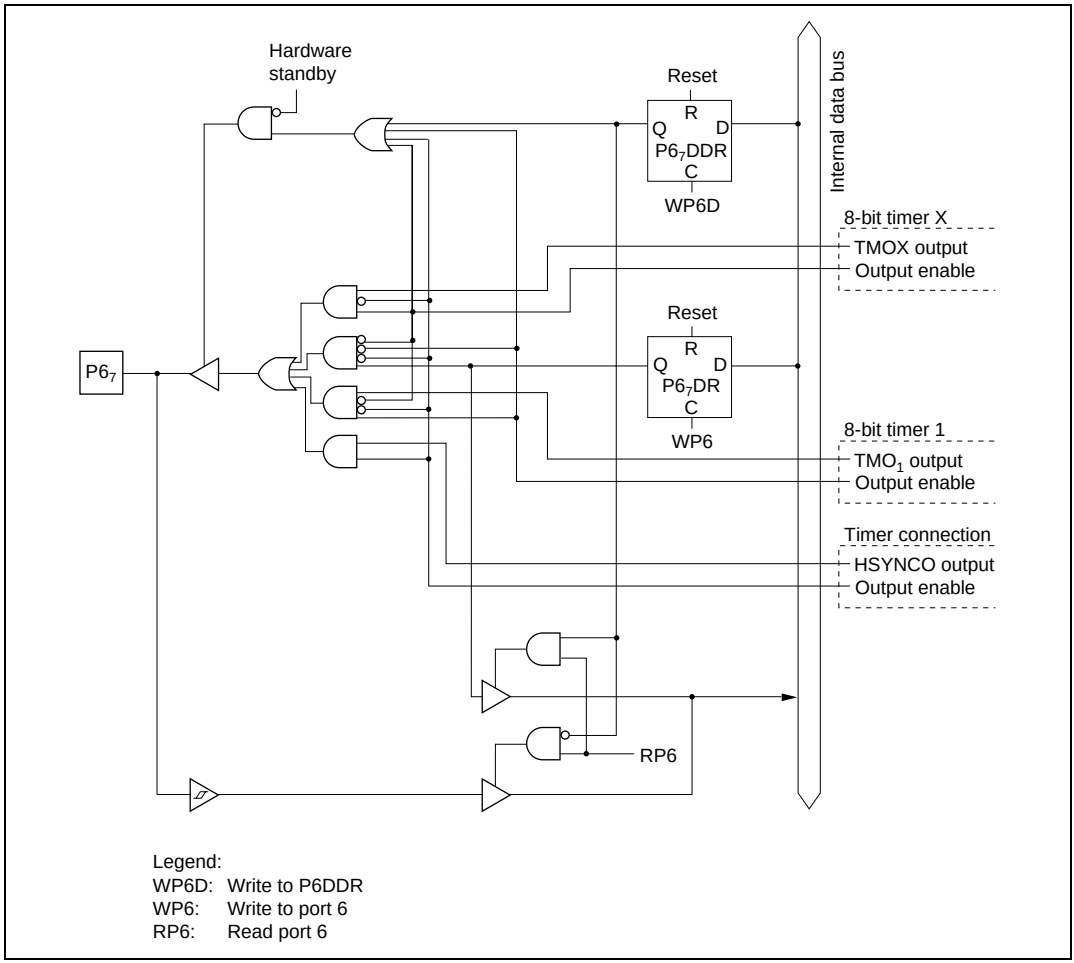


Figure C.23 Port 6 Block Diagram (Pin P6$_7$)

C.7 Port 7 Block Diagram

The H8/3577 Group has an 8-bit input port (pins $P7_0$ to $P7_7$) and the H8/3567 Group has a 4-bit input port (pins $P7_0$ to $P7_3$).

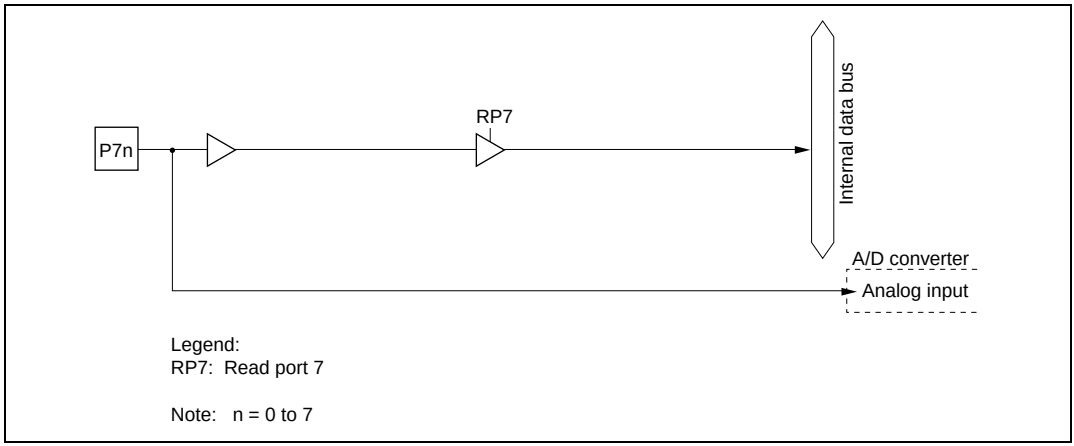


Figure C.24 Port 7 Block Diagram
(Pins $P7_0$ to $P7_7$ in H8/3577 Group, Pins $P7_0$ to $P7_3$ in H8/3567 Group)

C.8 Port 8 Block Diagrams

Port C is provided only in the H8/3567 Group version with an on-chip USB.

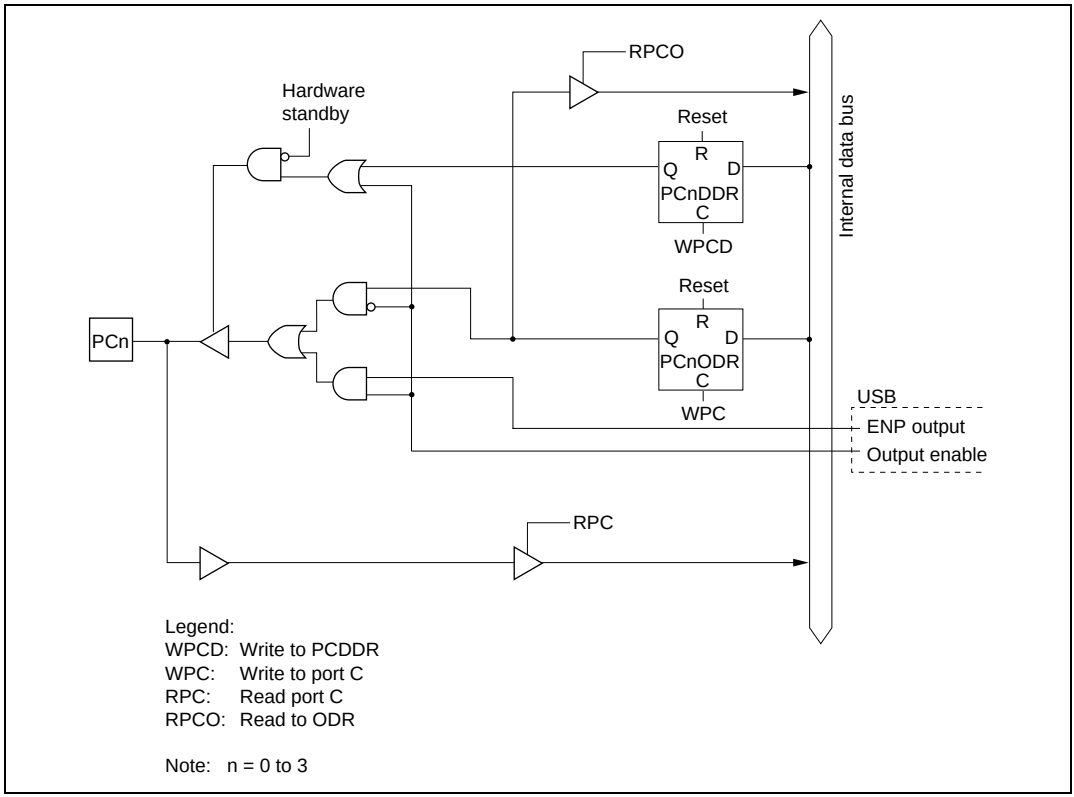


Figure C.25 Port C Block Diagram
(Pins PC_0 to PC_3 in H8/3567 Group Version with On-Chip USB)

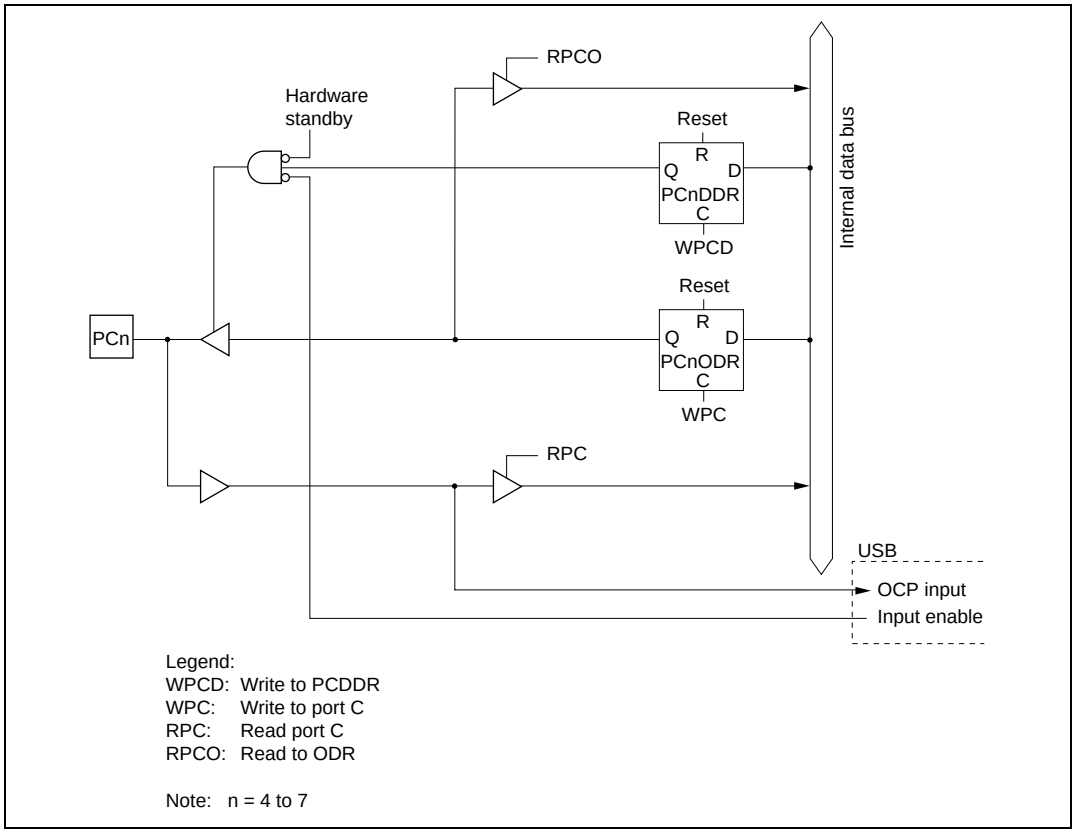


Figure C.26 Port C Block Diagram
(Pins PC_4 to PC_7 in H8/3567 Group Version with On-Chip USB)

C.9 Port D Block Diagram

Port D is provided only in the H8/3567 Group version with an on-chip USB.

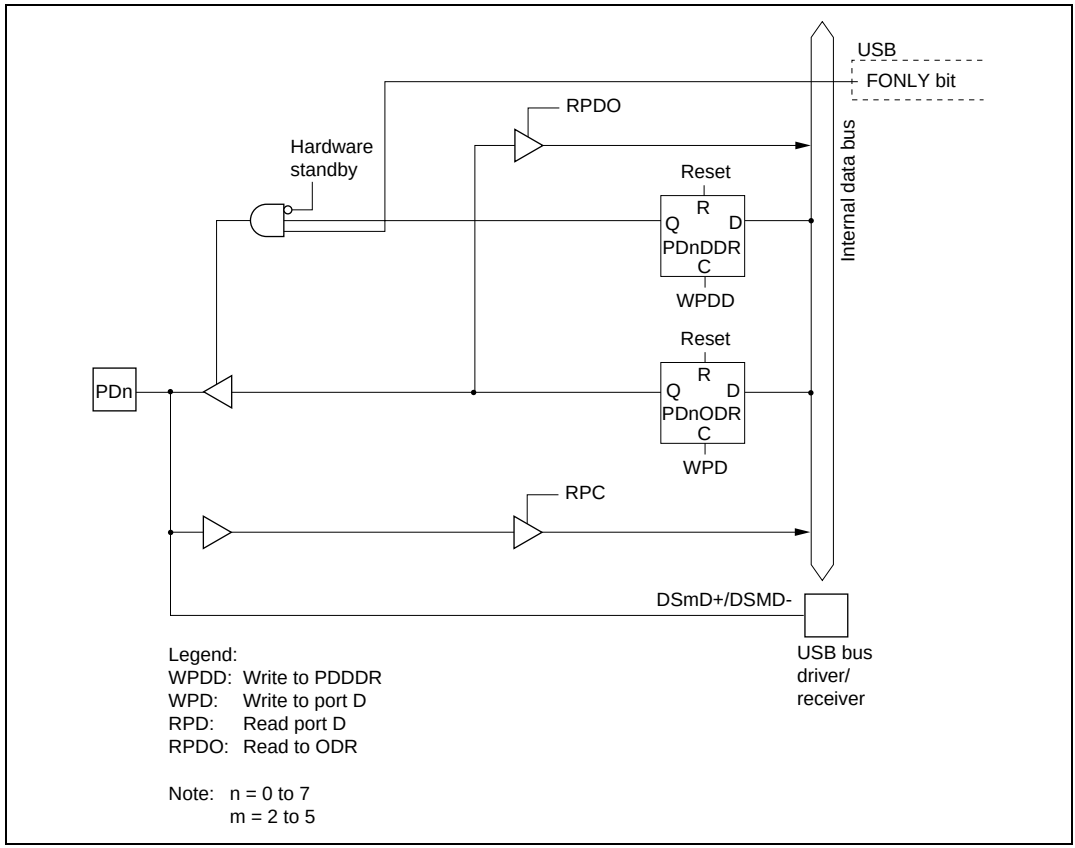


Figure C.27 Port D Block Diagram
(Pins PD_0 to PD_7 in H8/3567 Group Version with On-Chip USB)

Appendix D Pin States

D.1 Port States in Each Mode

Table D.1 I/O Port States in Each Processing State

Port Name Pin Name	Reset	Hardware Standby Mode	Software Standby Mode	Program Execution State
Port 1	T	T	kept	I/O port
Port 2	T	T	kept	I/O port
Port 3	T	T	kept	I/O port
Port 4_7	T	T	kept	I/O port
Port 4_6	T	T	[DDR = 1] H [DDR = 0] T	Clock output/input port
Port 4_5 to 4_0	T	T	kept	I/O port
Port 5	T	T	kept	I/O port
Port 6	T	T	kept	I/O port
Port 7	T	T	T	Input port
Port C	T	T	Functioning (HOCnE = 1) kept (HOCnE = 0)	USB input/output I/O port
Port D	T	T	Functioning (FONLY = 0) kept (FONLY = 1)	USB input/output I/O port

Legend:

H: High level

L: Low level

T: High impedance

kept: Input pins are in the high-impedance state (when DDR = 0 and PCR = 1, MOS input pull-ups remain in the on state).

Output ports retain their state.

In some cases, the on-chip supporting module is initialized and the pin is an input/output port, determined by the DDR and DR settings.

DDR: Data direction register

HOCnE: HOCnE bit in HOCCR of USB

FONLY: FONLY bit in USBCR of USB

Note: n = 2 to 5

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

E.1 Timing of Transition to Hardware Standby Mode

(1) To retain RAM contents 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 in figure E.1. $\overline{\text{RES}}$ must remain low until $\overline{\text{STBY}}$ goes low (minimum delay from $\overline{\text{STBY}}$ low to $\overline{\text{RES}}$ high: 0 ns).

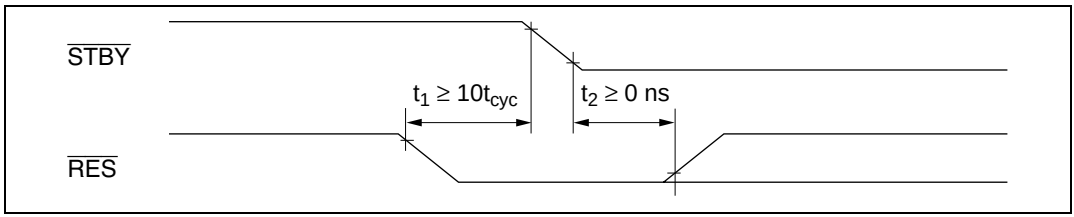


Figure E.1 Timing of Transition to Hardware Standby Mode

(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).

E.2 Timing of Recovery from Hardware Standby Mode

Drive the $\overline{\text{RES}}$ signal low approximately 100 ns or more before $\overline{\text{STBY}}$ goes high.

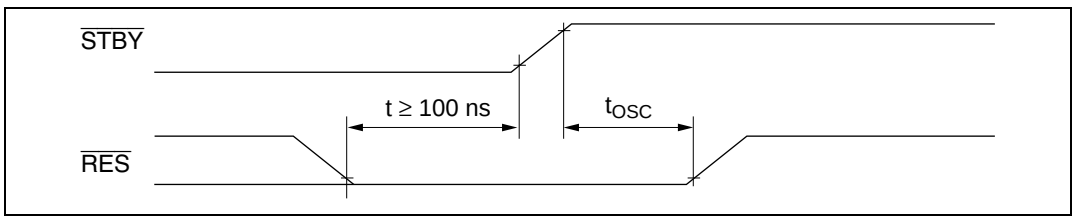


Figure E.2 Timing of Recovery from Hardware Standby Mode

Appendix F Product Code Lineup

Table F.1 H8/3577 Group and H8/3567 Group Product Code Lineup

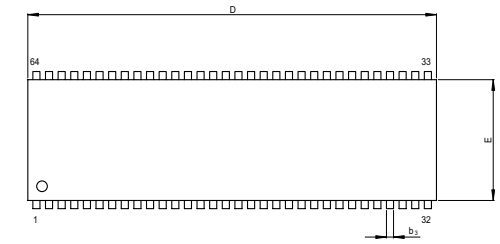
Product Type			Product Code	Mark Code	Package (Package Code)
H8/3577 Group	H8/3577	ZTAT version	HD6473577	HD6473577P20	64-pin shrink DIP (DP-64S)
				HD6476577F20	64-pin QFP (FP-64A)
		Mask ROM version	HD6433577	HD6433577(***)P20	64-pin shrink DIP (DP-64S)
				HD6433577(***)F20	64-pin QFP (FP-64A)
	H8/3574	Mask ROM version	HD6433574	HD6433574(***)P20	64-pin shrink DIP (DP-64S)
				HD6433574(***)F20	64-pin QFP (FP-64A)
H8/3567 Group	H8/3567	ZTAT version	HD6473567	HD6473567P20	42-pin shrink DIP (DP-42S)
				HD6476567F20	44-pin QFP (FP-44A)
		Mask ROM version	HD6433567	HD6433567(***)P20	42-pin shrink DIP (DP-42S)
				HD6433567(***)F20	44-pin QFP (FP-44A)
	H8/3564	Mask ROM version	HD6433564	HD6433564(***)P20	42-pin shrink DIP (DP-42S)
				HD6433564(***)F20	44-pin QFP (FP-44A)
		(10 MHz limit version)		HD6433564(***)P10	42-pin shrink DIP (DP-42S)
	H8/3567U	ZTAT version (on-chip USB)	HD6473567U	HD6473567UP20	64-pin shrink DIP (DP-64S)
				HD6473567UF20	64-pin QFP (FP-64A)
		Mask ROM version (on-chip USB)	HD6433567U	HD6433567U(***)P20	64-pin shrink DIP (DP-64S)
				HD6433567U(***)F20	64-pin QFP (FP-64A)
	H8/3564U	Mask ROM version (on-chip USB)	HD6433564U	HD6433564U(***)P20	64-pin shrink DIP (DP-64S)
				HD6433564U(***)F20	64-pin QFP (FP-64A)

Note: (***) is the ROM code.
When ordering, the frequency selection (20 or 10) is not indicated by the model name, but is identified by the ROM code.

Appendix G Package Dimensions

Figures G.1 to G.4 show package dimensions of H8/3577 Group and H8/3567 Group.

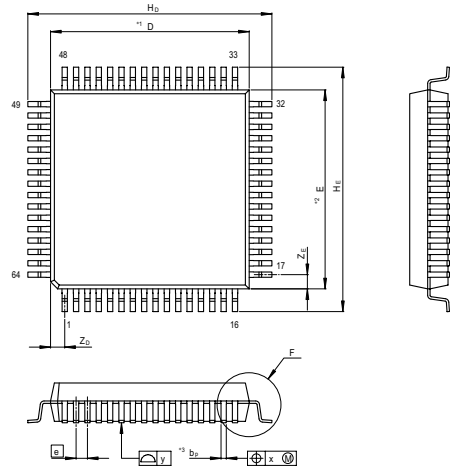
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-SDIP64-17x57.6-1.78	PRDP0064BB-A	DP-64S/DP-64SV	8.8g

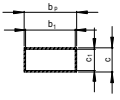


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
e_1	—	19.05	—
D	—	57.6	58.5
E	—	17.0	18.6
A	—	—	5.08
A_1	0.51	—	—
b_p	0.38	0.48	0.58
b_3	—	1.0	—
c	0.20	0.25	0.36
θ	0°	—	15°
e	1.53	1.78	2.03
Z	—	—	1.46
L	2.54	—	—

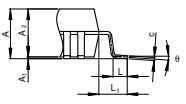
Figure G.1 DP-64S Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-QFP64-14x14-0.80	PRQP0064GB-A	FP-64A/FP-64AV	1.2g





Terminal cross section



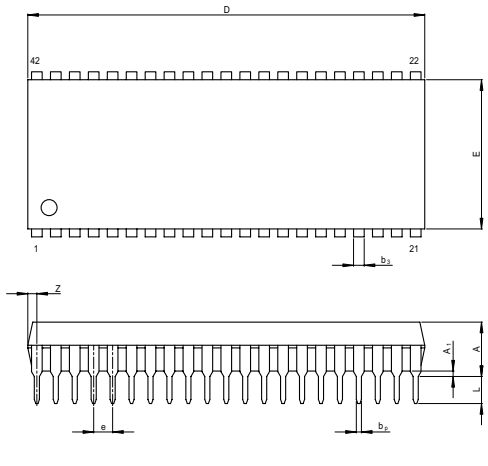
Detail F

NOTE)
1. DIMENSIONS"*1"AND"*2" DO NOT INCLUDE MOLD FLASH
2. DIMENSION"*3"DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	14	—
E	—	14	—
A_2	—	2.70	—
H_D	16.9	17.2	17.5
H_E	16.9	17.2	17.5
A	—	—	3.05
A_1	0.00	0.10	0.25
b_p	0.29	0.37	0.45
b_1	—	0.35	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.8	—
x	—	—	0.15
y	—	—	0.10
Z_D	—	1.0	—
Z_E	—	1.0	—
L	0.5	0.8	1.1
L_1	—	1.6	—

Figure G.2 FP-64A Package Dimensions

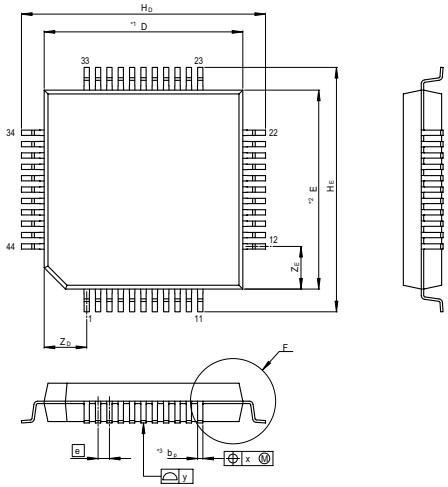
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-SDIP42-14x37.3-1.78	PRDP0042BB-A	DP-42S/DP-42SV	4.8g

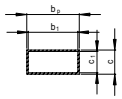


Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
e_1	—	15.24	—
D	—	37.3	38.6
E	—	14.0	14.6
A	—	—	5.10
A_1	0.51	—	—
b_p	0.38	0.48	0.58
b_3	—	1.0	—
c	0.20	0.25	0.35
θ	0°	—	15°
e	1.53	1.78	2.03
Z	—	—	1.38
L	2.54	—	—

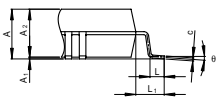
Figure G.3 DP-42S Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-QFP44-14x14-0.80	PRQP0044GC-A	FP-44A/FP-44AV	1.2g





Terminal cross section



Detail F

NOTE)
1. DIMENSIONS"*1"AND"*2" DO NOT INCLUDE MOLD FLASH
2. DIMENSION"*3"DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	14	—
E	—	14	—
A_2	—	2.70	—
H_D	16.9	17.2	17.5
H_E	16.9	17.2	17.5
A	—	—	3.05
A_1	0.00	0.10	0.25
b_p	0.29	0.37	0.45
b_1	—	0.35	—
c	0.12	0.17	0.22
c_1	—	0.15	—
θ	0°	—	8°
e	—	0.8	—
x	—	—	0.15
y	—	—	0.10
Z_D	—	3.0	—
Z_E	—	3.0	—
L	0.5	0.8	1.1
L_1	—	1.6	—

Figure G.4 FP-44A Package Dimensions

Renesas 8-Bit Single-Chip Microcomputer
Hardware Manual
H8/3577 Group, H8/3567 Group

Publication Date: 1st Edition, September 1999
Rev.3.00, March 17, 2006
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, 1 Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2730-6071

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jalan Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

H8/3577 Group, H8/3567 Group
Hardware Manual

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan