To all our customers

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

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

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

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

Cautions

Keep safety first in your circuit designs!

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

Notes regarding these materials

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

Hitachi Single-Chip Microcomputer
H8/3318

H8/3318
HD6473318, HD6433318

Hardware Manual

ADE-602-097A
Rev. 2.0
3/6/03
Hitachi, Ltd.

Cautions

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

Preface

The H8/3318 comprises high-performance microcontrollers with a fast H8/300 CPU core and a set of on-chip supporting functions optimized for embedded control. These include ROM, RAM, three types of timers, a programmable timing pattern controller, data transfer unit (with 256-byte DPRAM function), parallel buffer interface, serial communication interface, A/D converter, I/O ports, and other functions needed in control system configurations, so that compact, high-performance systems can be implemented easily. The H8/3318 includes the H8/3318, with 60-kbyte ROM and 4-kbyte RAM.

A ZTAT™ (Zero Turn Around Time) version of the H8/3318 is also available, with user-programmable PROM, providing a quick and flexible response under all sorts of production conditions, even for applications with frequently-changing specifications.

This manual describes the hardware of the H8/3318. Refer to the *H8/300 Series Programming Manual* for a detailed description of the instruction set.

Contents

Revisions and Additions in this Edition

Page	Item	Revision
All	H8/3315 deleted	
8 to 11	Table 1.2 Pin Assignments in Each Operating Mode	Amended
13	Table 1.3 Pin Functions	Operating mode control amended
57	3.1.1 Mode Selection	Description added
61	Figure 3.1 H8/3318 Address Space Map	Mode 1 amended
81 to 141	Section 5 Data Transfer Unit	Totally amended
144	6.2.1 Wait-State Control Register (WSCR)	Bit 4 initial value and Read/Write specification amended
149	7.1.2 Wait-State Control Register (WSCR)	Bit 4 initial value and Read/Write specification amended
203	Figure 8.20 Pin Functions in Mode 3 (Port 8)	Amended
211	8.11 Application Notes	Added
234	Figure 9.6 Timing of Output Compare A	Amended
261	Figure 10.5 Timing of Output Compare A	Amended
311	Figure 12.5 Normal TPC Output Example (Five-Phase Pulse Output)	Amended
319	13.2.2 Timer Control/Status Register (TCSR)	Notes added and amended
326	Figure 14.1 Block Diagram of Serial Communication Interface	Amended
335	14.2.7 Serial Status Register (SSR), Bit 7	Description added
336	14.2.7 Serial Status Register (SSR), Bit 6	Description added
338 to 347	Table 14.3 Examples of BRR Settings in Asynchronous Mode (When $ø_P$ = ø) Table 14.6 Examples of BRR Settings in Synchronous Mode (When $ø_P$ = ø/2)	Totally amended
356	Figure 14.5 Sample Flowchart for Transmitting Serial Data	Description added
361	Figure 14.8 Example of SCI Receive Operation (8-Bit Data with Parity and One Stop Bit)	Amended
364	Figure 14.11 Example of SCI Receive Operation (Eight-Bit Data with Multiprocessor Bit and One Stop Bit)	Amended

Page	Item	Revision
369	Figure 14.16 Example of SCI Receive Operation	Amended
370	Figure 14.17 Sample Flowchart for Serial Transmitting and Receiving	Note added
390 to 394	15.6 Usage Notes	Totally amended
397	16.3 Application Notes	Added
409	Figure 17.6 Recommended Screening Procedure	Amended, note added
431	Table 19.8 Control Signal Timing	Amended
439	Figure 19.5 Basic Bus Cycle (with 1 Wait State) in Expanded Modes (Modes 1, 2)	Amended
448	Figure 19.23 Receive Timing in Handshake Mode	Amended
477	B.2 Register Descriptions SSR—Serial Status Register	Description added
495	B.2 Register Descriptions WSCR—Wait-State Control Register	Amended
526	B.2 Register Descriptions DTCRB—Data Transfer Control Register B	Amended
528	B.2 Register Descriptions DTCRC—Data Transfer Control Register C	Amended
558	Figure C.9 (g) Port 9 Block Diagram (Pin $P9_7$)	Amended
562	Appendix F Product Code Lineup	Added
563 to 566	Appendix G Package Dimensions	Amended

Section 1 Overview

1.1 Overview

The H8/3318 consists of single-chip microcomputer units (MCUs) featuring an H8/300 CPU core and a complement of on-chip supporting modules implementing a variety of system functions.

The H8/300 CPU is a high-speed processor with an architecture featuring powerful bit-manipulation instructions, ideally suited for realtime control applications. The on-chip supporting modules implement peripheral functions needed in system configurations. These include ROM, RAM, three types of timers (16-bit free-running timers, 8-bit timers, and a watchdog timer), a programmable timing pattern controller (TPC), a data transfer unit (DTU) with a parallel buffer interface (PBI) supporting a 256-byte DPRAM function, a serial communication interface (SCI), an A/D converter, and I/O ports.

The H8/3318 can operate in single-chip mode or in two expanded modes, depending on the requirements of the application. The H8/3318 is available in a masked ROM version, or a ZTAT* version with electrically programmable ROM that can be programmed at the user site.

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

Table 1.1 lists the features of the H8/3318

Table 1.1 Features

Item	Description
CPU	Two-way general register configuration • Eight 16-bit registers, or • Sixteen 8-bit registers High-speed operation • Maximum clock rate: 16 MHz/5 V, 12 MHz/4 V, and 10 MHz/3 V (ø clock) • Add/subtract: 125 ns (16 MHz operation), 167 ns (12 MHz operation), and 200 ns (10 MHz operation) • Multiply/divide: 875 ns (16 MHz operation), 1167 ns (12 MHz operation), and 1400 ns (10 MHz operation) Streamlined, concise instruction set • Instruction length: 2 or 4 bytes • Register-register arithmetic and logic operations • MOV instruction for data transfer between registers and memory Instruction set features • Multiply instruction (8 bits × 8 bits) • Divide instruction (16 bits ÷ 8 bits) • Bit-accumulator instructions • Register-indirect specification of bit positions
Memory	• 60-kbyte ROM • 4-kbyte RAM
Data transfer unit (DTU) (4 channels)	• 1 channel for PBI transfer (single address) • 1 channel for I/O transfer (dual address) • 2 channels for either PBI or I/O transfer
16-bit free-running timer (2 channels)	• One 16-bit free-running counter per channel (can also count external events) • Two output-compare lines per channel • Channel 0: four input capture lines (can be buffered) • Channel 1: one input capture line

Table 1.1 Features (cont)

Item	Description
8-bit timer (2 channels)	Each channel has • One 8-bit up-counter (can also count external events) • Two time constant registers
Programmable timing pattern controller (TPC)	• Maximum 16-bit pulse output, using FRT as time base • Up to four 4-bit pulse output groups (or one 16-bit group, or two 8-bit groups) • Non-overlap mode available • Output data can be transferred by DTU
Watchdog timer (WDT) (1 channel)	• Overflow or NMI interrupt can reset the chip • Also usable as interval timer
Serial communication interface (SCI) (2 channels)	• Asynchronous or synchronous mode (selectable) • Full duplex: can transmit and receive simultaneously • On-chip baud rate generator
A/D converter	• 10-bit resolution • Eight channels: single or scan mode (selectable) • Start of A/D conversion can be externally triggered • Sample-and-hold function
I/O ports	• 58 input/output lines (16 of which can drive LEDs) • Eight input-only lines
Interrupts	• Nine external interrupt lines: $\overline{NMI}$, $\overline{IRQ}_0$ to $\overline{IRQ}_7$ • 33 on-chip interrupt sources
Operating modes	• Expanded mode with on-chip ROM disabled (mode 1) • Expanded mode with on-chip ROM enabled (mode 2) • Single-chip mode (mode 3)
Power-down modes	• Sleep mode • Software standby mode • Hardware standby mode
Other features	• On-chip oscillator

Table 1.1 Features (cont)

Item	Description				
Series lineup		Model			
	Product Name	5 V Version (16 MHz) 4 V Version (12 MHz)	3 V Version (10 MHz)	Package	ROM
	H8/3318 ZTAT	HD6473318CG16	HD6473318CG16	84-pin windowed LCC (CG-84)	PROM
		HD6473318CP16	HD6473318CP16	84-pin PLCC (CP-84)	
		HD6473318F16	HD6473318F16	80-pin QFP (FP-80A)	
		HD6473318TF16	HD6473318TF16	80-pin TQFP (TFP-80C)	
	H8/3318	HD6433318CP16 HD6433318CP12	HD6433318VCP10	84-pin PLCC (CP-84)	Mask ROM
		HD6433318F16 HD6433318F12	HD6433318VF10	80-pin QFP (FP-80A)	
		HD6433318TF16 HD6433318TF12	HD6433318VTF10	80-pin TQFP (TFP-80C)	

1.2 Block Diagram

Figure 1.1 shows a block diagram of the H8/3318.

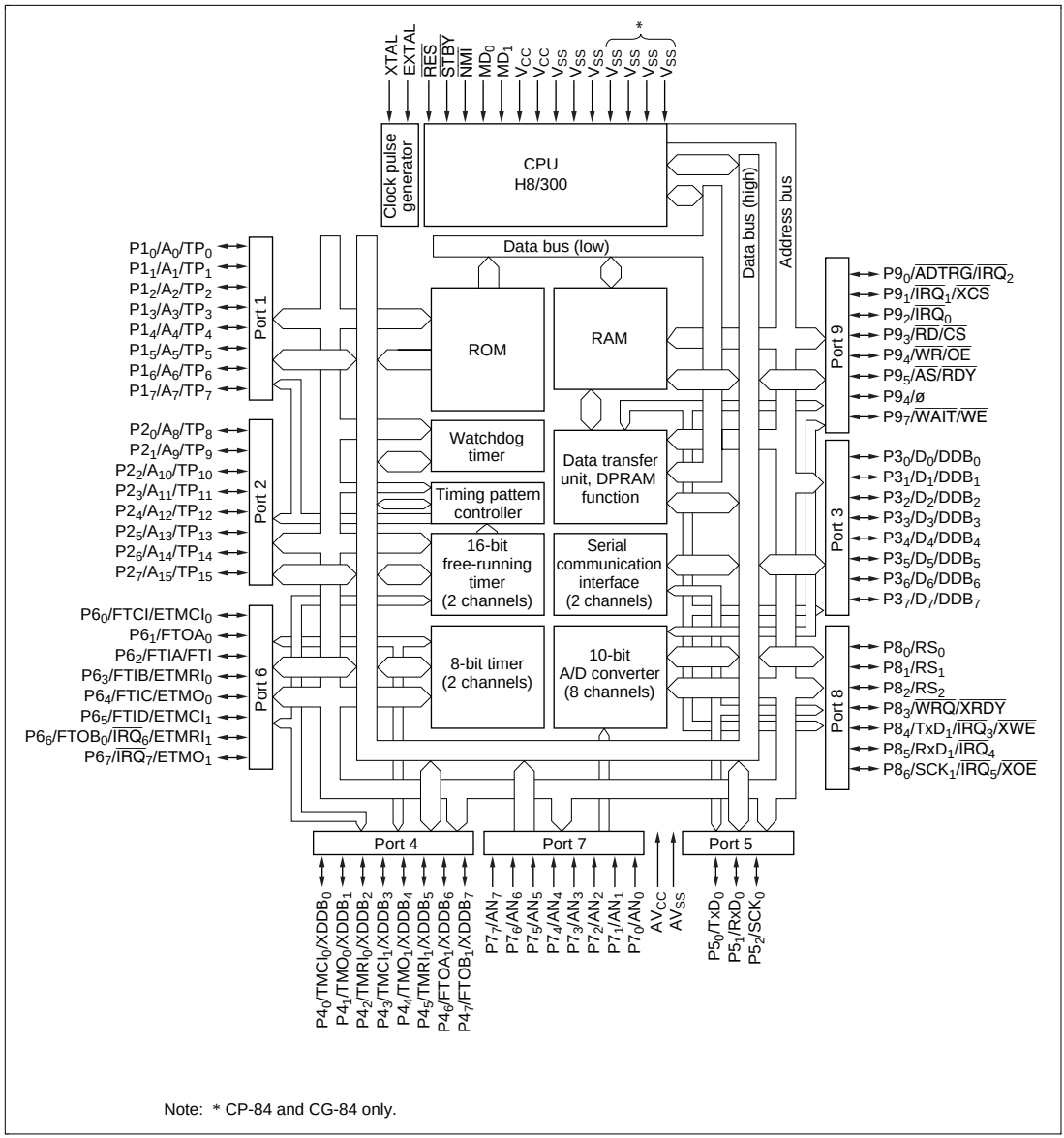


Figure 1.1 Block Diagram

1.3 Pin Assignments and Functions

1.3.1 Pin Arrangement

Figure 1.2 shows the pin arrangement of the CP-84 and CG-84 packages. Figure 1.3 shows the pin arrangement of the FP-80A and TFP-80C packages.

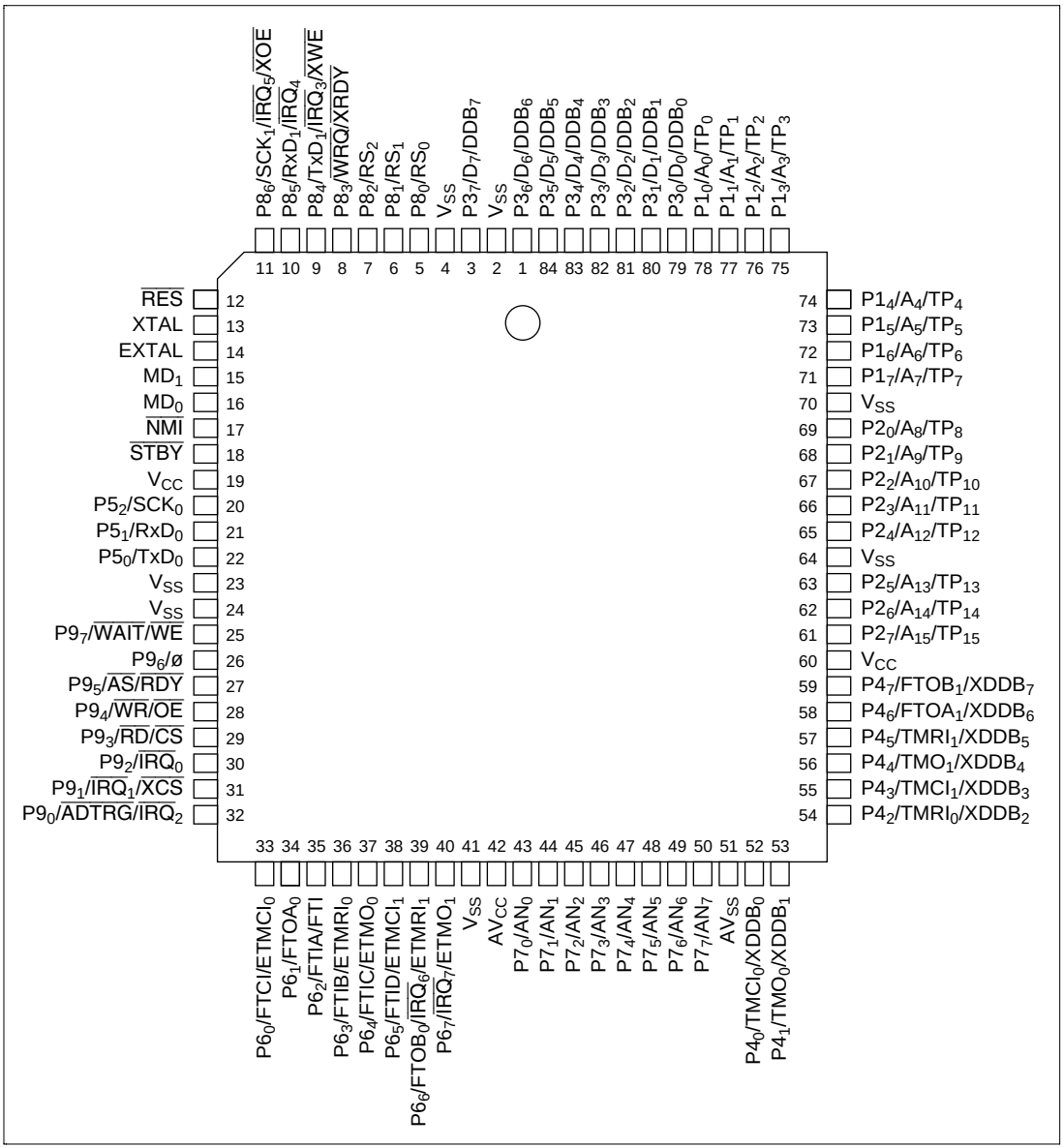


Figure 1.2 Pin Arrangement (CP-84 and CG-84, Top View)

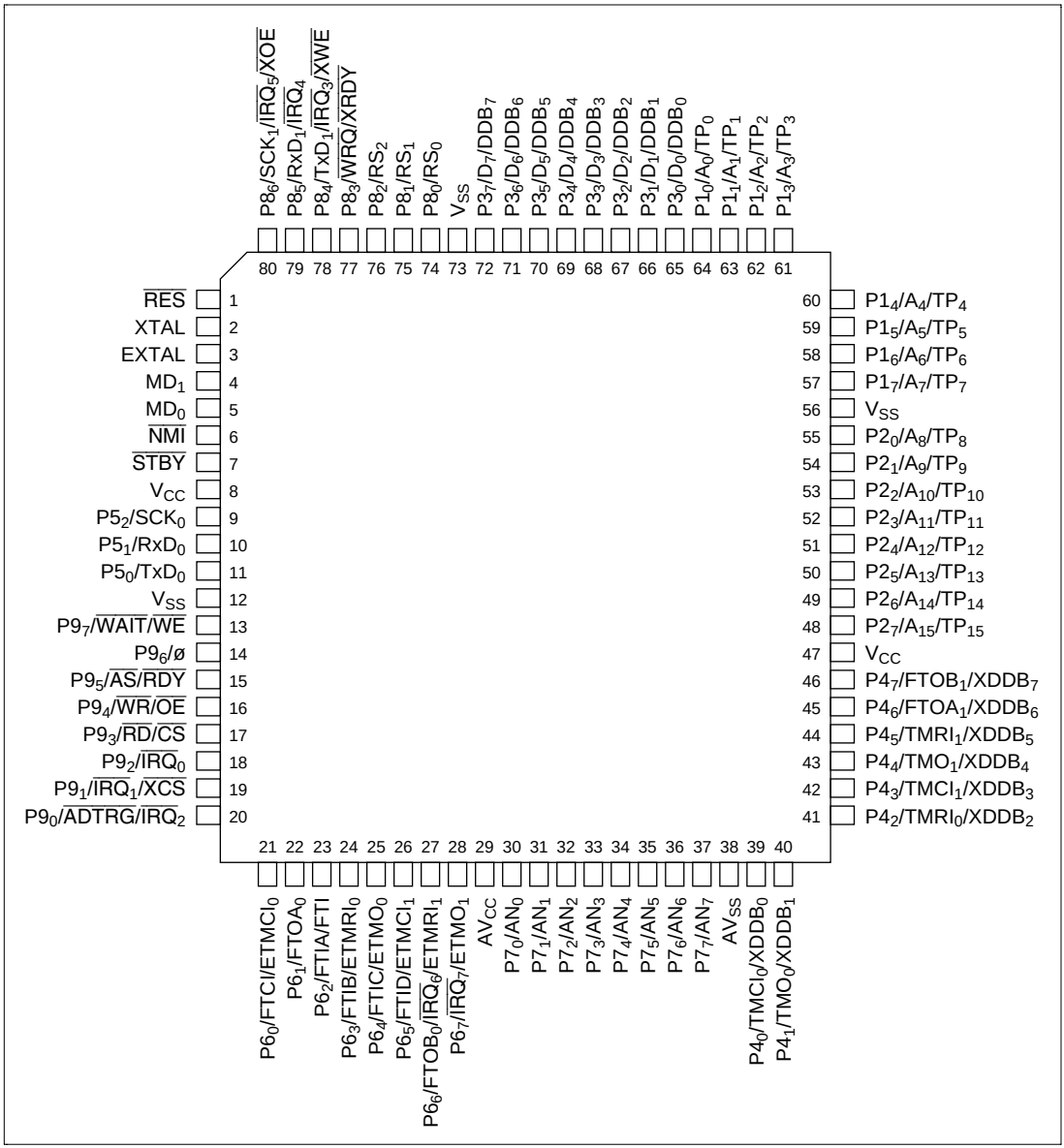


Figure 1.3 Pin Arrangement (FP-80A and TFP-80C, Top View)

1.3.2 Pin Assignments in Each Operating Mode

Table 1.2 lists the assignments of the pins of the CP-84, CG-84, FP-80A, and TFP-80C packages in each operating mode.

Table 1.2 Pin Assignments in Each Operating Mode

Pin No.		Expanded Modes			Single-Chip Mode		
			Mode 1	Mode 2	Mode 3		
CP-84 CG-84	FP-80A TFP-80C	PBI Enabled (DPME = 1)	PBI Disabled (DPME = 0)	PBI Disabled (DPME = 0)	PBI Disabled (DPME = 0)	PBI Enabled (DPME = 1)	PROM Mode
1	71	D_6	D_6	D_6	$P3_6$	DDB_6	EO_6
2	—	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
3	72	D_7	D_7	D_7	$P3_7$	DDB_7	EO_7
4	73	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
5	74	RS_0	$P8_0$	$P8_0$	$P8_0$	RS_0	NC
6	75	RS_1	$P8_1$	$P8_1$	$P8_1$	RS_1	NC
7	76	RS_2	$P8_2$	$P8_2$	$P8_2$	RS_2	NC
8	77	$\overline{WRQ}$/$\overline{XRDY}$	$P8_3$	$P8_3$	$P8_3$	$\overline{WRQ}$	NC
9	78	$\overline{XWE}$	$P8_4$/TxD_1/ $\overline{IRQ_3}$	$P8_4$/TxD_1/ $\overline{IRQ_3}$	$P8_4$/TxD_1/ $\overline{IRQ_3}$	$P8_4$/TxD_1/ $\overline{IRQ_3}$	NC
10	79	$P8_5$/RxD_1/ $\overline{IRQ_4}$	$P8_5$/RxD_1/ $\overline{IRQ_4}$	$P8_5$/RxD_1/ $\overline{IRQ_4}$	$P8_5$/RxD_1/ $\overline{IRQ_4}$	$P8_5$/RxD_1/ $\overline{IRQ_4}$	NC
11	80	$\overline{XOE}$	$P8_6$/SCK_1/ $\overline{IRQ_5}$	$P8_6$/SCK_1/ $\overline{IRQ_5}$	$P8_6$/SCK_1/ $\overline{IRQ_5}$	$P8_6$/SCK_1/ $\overline{IRQ_5}$	NC
12	1	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	$\overline{RES}$	V_{PP}
13	2	XTAL	XTAL	XTAL	XTAL	XTAL	NC
14	3	EXTAL	EXTAL	EXTAL	EXTAL	EXTAL	NC
15	4	MD_1	MD_1	MD_1	MD_1	MD_1	V_{SS}
16	5	MD_0	MD_0	MD_0	MD_0	MD_0	V_{SS}
17	6	$\overline{NMI}$	$\overline{NMI}$	$\overline{NMI}$	$\overline{NMI}$	$\overline{NMI}$	EA_9
18	7	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	$\overline{STBY}$	V_{SS}
19	8	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
20	9	$P5_2$/SCK_0	$P5_2$/SCK_0	$P5_2$/SCK_0	$P5_2$/SCK_0	$P5_2$/SCK_0	NC

Note: Pins marked NC should be left unconnected.
For details on PROM mode, refer to 17.2, PROM Mode.

Table 1.2 Pin Assignments in Each Operating Mode (cont)

Pin No.		Expanded Modes			Single-Chip Mode		
			Mode 1	Mode 2	Mode 3		
CP-84 CG-84	FP-80A TFP-80C	PBI Enabled (DPME = 1)	PBI Disabled (DPME = 0)	PBI Disabled (DPME = 0)	PBI Disabled (DPME = 0)	PBI Enabled (DPME = 1)	PROM Mode
21	10	$P5_1/RxD_0$	$P5_1/RxD_0$	$P5_1/RxD_0$	$P5_1/RxD_0$	$P5_1/RxD_0$	NC
22	11	$P5_0/TxD_0$	$P5_0/TxD_0$	$P5_0/TxD_0$	$P5_0/TxD_0$	$P5_0/TxD_0$	NC
23	12	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
24	—	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
25	13	$\overline{WAIT}$	$\overline{WAIT}$	$\overline{WAIT}$	$P9_7$	$\overline{WE}$	NC
26	14	ø	ø	ø	$P9_6$/ø	$P9_6$/ø	NC
27	15	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	$P9_5$	$\overline{RDY}$	NC
28	16	$\overline{WR}$	$\overline{WR}$	$\overline{WR}$	$P9_4$	$\overline{OE}$	NC
29	17	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	$P9_3$	$\overline{CS}$	NC
30	18	$P9_2/\overline{IRQ_0}$	$P9_2/\overline{IRQ_0}$	$P9_2/\overline{IRQ_0}$	$P9_2/\overline{IRQ_0}$	$P9_2/\overline{IRQ_0}$	$\overline{PGM}$
31	19	$\overline{XCS}$	$P9_1/\overline{IRQ_1}$	$P9_1/\overline{IRQ_1}$	$P9_1/\overline{IRQ_1}$	$P9_1/\overline{IRQ_1}$	EA_{15}
32	20	$P9_0/\overline{IRQ_2}/\overline{ADTRG}$	$P9_0/\overline{IRQ_2}/\overline{ADTRG}$	$P9_0/\overline{IRQ_2}/\overline{ADTRG}$	$P9_0/\overline{IRQ_2}/\overline{ADTRG}$	$P9_0/\overline{IRQ_2}/\overline{ADTRG}$	EA_{16}
33	21	$P6_0/ETMCI_0$/FTCI	$P6_0$/FTCI	$P6_0$/FTCI	$P6_0$/FTCI	$P6_0$/FTCI	NC
34	22	$P6_1/FTOA_0$	$P6_1/FTOA_0$	$P6_1/FTOA_0$	$P6_1/FTOA_0$	$P6_1/FTOA_0$	NC
35	23	$P6_2$/FTIA/FTI	$P6_2$/FTIA/FTI	$P6_2$/FTIA/FTI	$P6_2$/FTIA/FTI	$P6_2$/FTIA/FTI	NC
36	24	$P6_3/ETMRI_0$/FTIB	$P6_3$/FTIB	$P6_3$/FTIB	$P6_3$/FTIB	$P6_3$/FTIB	V_{CC}
37	25	$P6_4/ETMO_0$/FTIC	$P6_4$/FTIC	$P6_4$/FTIC	$P6_4$/FTIC	$P6_4$/FTIC	V_{CC}
38	26	$P6_5/ETMCI_1$/FTID	$P6_5$/FTID	$P6_5$/FTID	$P6_5$/FTID	$P6_5$/FTID	NC
39	27	$P6_6/ETMRI_1$/$FTOB_0/\overline{IRQ_6}$	$P6_6/FTOB_0/\overline{IRQ_6}$	$P6_6/FTOB_0/\overline{IRQ_6}$	$P6_6/FTOB_0/\overline{IRQ_6}$	$P6_6/FTOB_0/\overline{IRQ_6}$	NC
40	28	$P6_7/ETMO_1/\overline{IRQ_7}$	$P6_7/\overline{IRQ_7}$	$P6_7/\overline{IRQ_7}$	$P6_7/\overline{IRQ_7}$	$P6_7/\overline{IRQ_7}$	NC

Note: Pins marked NC should be left unconnected.
For details on PROM mode, refer to 17.2, PROM Mode.

Table 1.2 Pin Assignments in Each Operating Mode (cont)

Pin No.		Expanded Modes			Single-Chip Mode		
			Mode 1	Mode 2	Mode 3		
CP-84 CG-84	FP-80A TFP-80C	PBI Enabled (DPME = 1)	PBI Disabled (DPME = 0)	PBI Disabled (DPME = 0)	PBI Disabled (DPME = 0)	PBI Enabled (DPME = 1)	PROM Mode
41	—	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
42	29	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	AV_{CC}	V_{CC}
43	30	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	$P7_0/AN_0$	NC
44	31	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	$P7_1/AN_1$	NC
45	32	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	$P7_2/AN_2$	NC
46	33	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	$P7_3/AN_3$	NC
47	34	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	$P7_4/AN_4$	NC
48	35	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	$P7_5/AN_5$	NC
49	36	$P7_6/AN_6$	$P7_6/AN_6$	$P7_6/AN_6$	$P7_6/AN_6$	$P7_6/AN_6$	NC
50	37	$P7_7/AN_7$	$P7_7/AN_7$	$P7_7/AN_7$	$P7_7/AN_7$	$P7_7/AN_7$	NC
51	38	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	AV_{SS}	V_{SS}
52	39	$XDDB_0$	$P4_0/TMCI_0$	$P4_0/TMCI_0$	$P4_0/TMCI_0$	$P4_0/TMCI_0$	NC
53	40	$XDDB_1$	$P4_1/TMO_0$	$P4_1/TMO_0$	$P4_1/TMO_0$	$P4_1/TMO_0$	NC
54	41	$XDDB_2$	$P4_2/TMRI_0$	$P4_2/TMRI_0$	$P4_2/TMRI_0$	$P4_2/TMRI_0$	NC
55	42	$XDDB_3$	$P4_3/TMCI_1$	$P4_3/TMCI_1$	$P4_3/TMCI_1$	$P4_3/TMCI_1$	NC
56	43	$XDDB_4$	$P4_4/TMO_1$	$P4_4/TMO_1$	$P4_4/TMO_1$	$P4_4/TMO_1$	NC
57	44	$XDDB_5$	$P4_5/TMRI_1$	$P4_5/TMRI_1$	$P4_5/TMRI_1$	$P4_5/TMRI_1$	NC
58	45	$XDDB_6$	$P4_6/FTOA_1$	$P4_6/FTOA_1$	$P4_6/FTOA_1$	$P4_6/FTOA_1$	NC
59	46	$XDDB_7$	$P4_7/FTOB_1$	$P4_7/FTOB_1$	$P4_7/FTOB_1$	$P4_7/FTOB_1$	NC
60	47	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}	V_{CC}
61	48	A_{15}	A_{15}	$P2_7/A_{15}/TP_{15}$	$P2_7/TP_{15}$	$P2_7/TP_{15}$	$\overline{CE}$
62	49	A_{14}	A_{14}	$P2_6/A_{14}/TP_{14}$	$P2_6/TP_{14}$	$P2_6/TP_{14}$	EA_{14}
63	50	A_{13}	A_{13}	$P2_5/A_{13}/TP_{13}$	$P2_5/TP_{13}$	$P2_5/TP_{13}$	EA_{13}
64	—	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
65	51	A_{12}	A_{12}	$P2_4/A_{12}/TP_{12}$	$P2_4/TP_{12}$	$P2_4/TP_{12}$	EA_{12}
66	52	A_{11}	A_{11}	$P2_3/A_{11}/TP_{11}$	$P2_3/TP_{11}$	$P2_3/TP_{11}$	EA_{11}

Note: Pins marked NC should be left unconnected.
For details on PROM mode, refer to 17.2, PROM Mode.

Table 1.2 Pin Assignments in Each Operating Mode (cont)

Pin No.		Expanded Modes			Single-Chip Mode		
			Mode 1	Mode 2	Mode 3		
CP-84 CG-84	FP-80A TFP-80C	PBI Enabled (DPME = 1)	PBI Disabled (DPME = 0)	PBI Disabled (DPME = 0)	PBI Disabled (DPME = 0)	PBI Enabled (DPME = 1)	PROM Mode
67	53	A_{10}	A_{10}	$P2_2/A_{10}/TP_{10}$	$P2_2/TP_{10}$	$P2_2/TP_{10}$	EA_{10}
68	54	A_9	A_9	$P2_1/A_9/TP_9$	$P2_1/TP_9$	$P2_1/TP_9$	$\overline{OE}$
69	55	A_8	A_8	$P2_0/A_8/TP_8$	$P2_0/TP_8$	$P2_0/TP_8$	EA_8
70	56	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}	V_{SS}
71	57	A_7	A_7	$P1_7/A_7/TP_7$	$P1_7/TP_7$	$P1_7/TP_7$	EA_7
72	58	A_6	A_6	$P1_6/A_6/TP_6$	$P1_6/TP_6$	$P1_6/TP_6$	EA_6
73	59	A_5	A_5	$P1_5/A_5/TP_5$	$P1_5/TP_5$	$P1_5/TP_5$	EA_5
74	60	A_4	A_4	$P1_4/A_4/TP_4$	$P1_4/TP_4$	$P1_4/TP_4$	EA_4
75	61	A_3	A_3	$P1_3/A_3/TP_3$	$P1_3/TP_3$	$P1_3/TP_3$	EA_3
76	62	A_2	A_2	$P1_2/A_2/TP_2$	$P1_2/TP_2$	$P1_2/TP_2$	EA_2
77	63	A_1	A_1	$P1_1/A_1/TP_1$	$P1_1/TP_1$	$P1_1/TP_1$	EA_1
78	64	A_0	A_0	$P1_0/A_0/TP_0$	$P1_0/TP_0$	$P1_0/TP_0$	EA_0
79	65	D_0	D_0	D_0	$P3_0$	DDB_0	EO_0
80	66	D_1	D_1	D_1	$P3_1$	DDB_1	EO_1
81	67	D_2	D_2	D_2	$P3_2$	DDB_2	EO_2
82	68	D_3	D_3	D_3	$P3_3$	DDB_3	EO_3
83	69	D_4	D_4	D_4	$P3_4$	DDB_4	EO_4
84	70	D_5	D_5	D_5	$P3_5$	DDB_5	EO_5

Note: Pins marked NC should be left unconnected.
For details on PROM mode, refer to 17.2, PROM Mode.

1.3.3 Pin Functions

Table 1.3 gives a description of the function of each pin.

Table 1.3 Pin Functions

		Pin No.			
Type	**Symbol**	**CP-84 CG-84**	**FP-80A TFP-80C**	**I/O**	**Name and Function**
Power	V_{CC}	19, 60	8, 47	I	**Power:** Connected to the power supply. Connect both V_{CC} pins to the system power supply.
	V_{SS}	2, 4, 23, 24, 41, 64, 70	12, 56, 73	I	**Ground:** Connected to ground (0 V). Connect all V_{SS} pins to system ground (0 V).
Clock	XTAL	13	2	I	**Crystal:** Connected to a crystal oscillator. The crystal frequency should be the same as the desired system clock frequency. When an external clock is input from the EXTAL pin, an inverse-phase clock should be input to the XTAL pin.
	EXTAL	14	3	I	**External crystal:** Connected to a crystal oscillator or external clock. The frequency of the external clock should be the same as the desired system clock frequency. See section 7.2, Oscillator Circuit, for examples of connections to a crystal and external clock.
	ø	26	14	O	**System clock:** Supplies the system clock to peripheral devices.
System control	$\overline{RES}$	12	1	I	**Reset:** A low input causes the chip to reset.
	$\overline{STBY}$	18	7	I	**Standby:** A transition to hardware standby mode (a power-down state) occurs when a low input is received at the $\overline{STBY}$ pin.
Address bus	A_{15} to A_0	61 to 63, 65 to 69, 71 to 78	48 to 55, 57 to 64	O	**Address bus:** Address output pins.
Data bus	D_7 to D_0	3, 1, 84 to 79	72 to 65	I/O	**Data bus:** 8-bit bidirectional data bus.

Table 1.3 Pin Functions (cont)

Type	Symbol	Pin No. CP-84 CG-84	Pin No. FP-80A TFP-80C	I/O	Name and Function
Bus control	$\overline{WAIT}$	25	13	I	**Wait:** Requests the CPU to insert wait states into the bus cycle when an external address is accessed.
	$\overline{RD}$	29	17	O	**Read:** Goes low to indicate that the CPU is reading an external address.
	$\overline{WR}$	28	16	O	**Write:** Goes low to indicate that the CPU is writing to an external address.
	$\overline{AS}$	27	15	O	**Address strobe:** Goes low to indicate that there is a valid address on the address bus.
Interrupt signals	$\overline{NMI}$	17	6	I	**Non-maskable interrupt:** Highest-priority interrupt request. The NMIEG bit in the system control register determines whether the interrupt is recognized at the rising or falling edge of the NMI input.
	$\overline{IRQ_0}$ to $\overline{IRQ_7}$	30 to 32, 9 to 11, 39, 40	18 to 20, 78 to 80, 27, 28	I	**Interrupt request 0 to 7:** Maskable interrupt request pins.
Operating mode control	MD_1, MD_0	15 16	4 5	I	**Mode:** Input pins for setting the MCU operating mode according to the table below. These pins must not be changed during MCU operation.

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

Table 1.3 Pin Functions (cont)

Type	Symbol	Pin No. CP-84 CG-84	Pin No. FP-80A TFP-80C	I/O	Name and Function
16-bit free-running timer (FRT)	FTCI	33	21	I	**FRT counter clock input:** Input pin for an external clock signal for the free-running counter (FRC) in FRT0 and FRT1.
	$FTOA_0$, $FTOA_1$	34 58	22 45	O	**FRT output compare A:** Output pin for output compare A in FRT0 and FRT1.
	$FTOB_0$, $FTOB_1$	39 59	27 46	O	**FRT output compare B:** Output pin for output compare B in FRT0 and FRT1.
	FTIA, FTI	35 35	23 23	I	**FRT input capture A:** Input pin for input capture A in FRT0, input pin for input capture in FRT1.
	FTIB	36	24	I	**FRT input capture B:** Input pin for input capture B in FRT0.
	FTIC	37	25	I	**FRT input capture C:** Input pin for input capture C in FRT0.
	FTID	38	26	I	**FRT input capture D:** Input pin for input capture D in FRT0.
8-bit timer	TMO_0, TMO_1	53 56	40 43	O	**8-bit timer output (channels 0 and 1):** Compare-match output pins for the 8-bit timers.
	$TMCI_0$, $TMCI_1$	52 55	39 42	I	**8-bit timer counter clock input (channels 0 and 1):** External clock input pins for the 8-bit timer counters.
	$TMRI_0$, $TMRI_1$	54 57	41 44	I	**8-bit timer counter reset input (channels 0 and 1):** Inputs at these pins reset the 8-bit timer counters.
8-bit timer (pins used in expanded modes when PBI is enabled)	$ETMO_0$, $ETMO_1$	37 40	25 28	O	**8-bit timer output (channels 0 and 1):** Compare-match output pins for the 8-bit timers.
	$ETMCI_0$, $ETMCI_1$	33 38	21 26	I	**8-bit timer counter clock input (channels 0 and 1):** External clock input pins for the 8-bit timers.
	$ETMRI_0$, $ETMRI_1$	36 39	24 27	I	**8-bit timer counter reset input (channels 0 and 1):** Inputs at these pins reset the 8-bit timer counters.

Table 1.3 Pin Functions (cont)

Type	Symbol	Pin No. CP-84 CG-84	Pin No. FP-80A TFP-80C	I/O	Name and Function
Program-mable timing pattern controller (TPC)	TP_{15} to TP_0	61 to 63, 65 to 69, 71 to 78	48 to 55, 57 to 64	O	**TPC output 15 to 0:** Pulse output pins.
Serial com-munication interface (SCI)	TxD_0, TxD_1	22 9	11 78	O	**Transmit data (channels 0 and 1):** Data output pins for the serial communication interface.
	RxD_0, RxD_1	21 10	10 79	I	**Receive data (channels 0 and 1):** Data input pins for the serial communication interface.
	SCK_0, SCK_1	20 11	9 80	I/O	**Serial clock (channels 0 and 1):** Input/output pins for the serial clock.
A/D converter	AN_7 to AN_0	50 to 43	37 to 30	I	**Analog input:** Analog signal input pins for the A/D converter.
	$\overline{ADTRG}$	32	20	I	**A/D trigger:** External trigger input for starting the A/D converter.
	AV_{CC}	42	29	I	**Analog reference voltage:** Reference voltage pin for the A/D converter. If the A/D converter is not used, connect AV_{CC} to the system power supply. Refer to section 19, Electrical Specifications.
	AV_{SS}	51	38	I	**Analog ground:** Ground pin for the A/D converter. Connect to system ground (0 V).
Dual-port RAM (DPRAM)	DDB_7 to DDB_0	3, 1, 84 to 79	72 to 65	I/O	**DPRAM data bus:** 8-bit bidirectional data bus for DPRAM access by an external CPU.
	$\overline{CS}$	29	17	I	**Chip select:** Chip select input pin for selecting DPRAM.
	RS_2 to RS_0	7 to 5	76 to 74	I	**Register select:** Address input pins for accessing DPRAM.
	$\overline{OE}$	28	16	I	**Output enable:** Output enable input pin for reading DPRAM.
	$\overline{WE}$	25	13	I	**Write enable:** Write enable input pin for writing to DPRAM.

Table 1.3 Pin Functions (cont)

Type	Symbol	Pin No. CP-84 CG-84	Pin No. FP-80A TFP-80C	I/O	Name and Function
Dual-port RAM (DPRAM)	$\overline{RDY}$	27	15	O	**Ready:** Ready output pin for sending interrupt requests to an external CPU. NMOS open-drain output.
	$\overline{WRQ}$	8	77	O	**Wait request:** Output pin for sending wait requests to an external CPU.
Dual-port RAM (DPRAM) (pin functions in expanded modes when PBI is enabled)	$XDDB_7$ to $XDDB_0$	59 to 52	46 to 39	I/O	**DPRAM data bus:** 8-bit bidirectional data bus for DPRAM access by an external CPU.
	$\overline{XCS}$	31	19	I	**Chip select:** Chip select input pin for selecting DPRAM.
	RS_2 to RS_0	7 to 5	76 to 74	I	**Register select:** Address input pin for accessing DPRAM.
	$\overline{XOE}$	11	80	I	**Output enable:** Output enable input pin for reading DPRAM.
	$\overline{XWE}$	9	78	I	**Write enable:** Write enable input pin for writing to DPRAM.
	$\overline{WRQ}$/ $\overline{XRDY}$	8	77	O	**Ready/wait request:** Output pin for sending interrupt requests to an external CPU.
I/O ports	$P1_7$ to $P1_0$	71 to 78	57 to 64	I/O	**Port 1:** An 8-bit input/output port with programmable MOS input pull-ups and LED driving capability. The direction of each bit can be selected in the port 1 data direction register (P1DDR).
	$P2_7$ to $P2_0$	61 to 63, 65 to 69	48 to 55	I/O	**Port 2:** An 8-bit input/output port with programmable MOS input pull-ups and LED driving capability. The direction of each bit can be selected in the port 2 data direction register (P2DDR).
	$P3_7$ to $P3_0$	3, 1, 84 to 79	72 to 65	I/O	**Port 3:** An 8-bit input/output port with programmable MOS input pull-ups. The direction of each bit can be selected in the port 3 data direction register (P3DDR).

Table 1.3 Pin Functions (cont)

Type	Symbol	Pin No. CP-84 CG-84	Pin No. FP-80A TFP-80C	I/O	Name and Function
I/O ports	$P4_7$ to $P4_0$	59 to 52	46 to 39	I/O	**Port 4:** An 8-bit input/output port. The direction of each bit can be selected in the port 4 data direction register (P4DDR).
	$P5_2$ to $P5_0$	20 to 22	9 to 11	I/O	**Port 5:** A 3-bit input/output port. The direction of each bit can be selected in the port 5 data direction register (P5DDR).
	$P6_7$ to $P6_0$	40 to 33	28 to 21	I/O	**Port 6:** An 8-bit input/output port. The direction of each bit can be selected in the port 6 data direction register (P6DDR).
	$P7_7$ to $P7_0$	50 to 43	37 to 30	I	**Port 7:** An 8-bit input port.
	$P8_6$ to $P8_0$	11 to 5	80 to 74	I/O	**Port 8:** A 7-bit input/output port. The direction of each bit can be selected in the port 8 data direction register (P8DDR).
	$P9_7$ to $P9_0$	25 to 32	13 to 20	I/O	**Port 9:** An 8-bit input/output port. The direction of each bit (except for $P9_6$) can be selected in the port 9 data direction register (P9DDR).

Section 2 CPU

2.1 Overview

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

2.1.1 Features

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

- Two-way register configuration
  - — Sixteen 8-bit general registers, or
  - — Eight 16-bit general registers
- Instruction set with 57 basic instructions, including:
  - — Multiply and divide instructions
  - — Powerful bit-manipulation instructions
- Eight addressing modes
  - — Register direct (Rn)
  - — Register indirect (@Rn)
  - — Register indirect with displacement (@(d:16, Rn))
  - — Register indirect with post-increment or pre-decrement (@Rn+ or @–Rn)
  - — Absolute address (@aa:8 or @aa:16)
  - — Immediate (#xx:8 or #xx:16)
  - — PC-relative (@(d:8, PC))
  - — Memory indirect (@@aa:8)
- Maximum 64-kbyte address space
- High-speed operation
  - — All frequently-used instructions are executed in two to four states
  - — The maximum clock rate is 16 MHz/5 V, 12 MHz/4 V, or 10 MHz/3 V (ø clock)
  - — 8- or 16-bit register-register add or subtract: 125 ns (16 MHz operation), 167 ns (12 MHz operation), 200 ns (10 MHz operation)
  - — 8 × 8-bit multiply: 875 ns (16 MHz operation), 1167 ns (12 MHz operation), 1400 ns (10 MHz operation)
  - — 16 ÷ 8-bit divide: 875 ns (16 MHz operation), 1167 ns (12 MHz operation), 1400 ns (10 MHz operation)
- Power-down mode
  - — SLEEP instruction

2.1.2 Address Space

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

2.1.3 Register Configuration

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

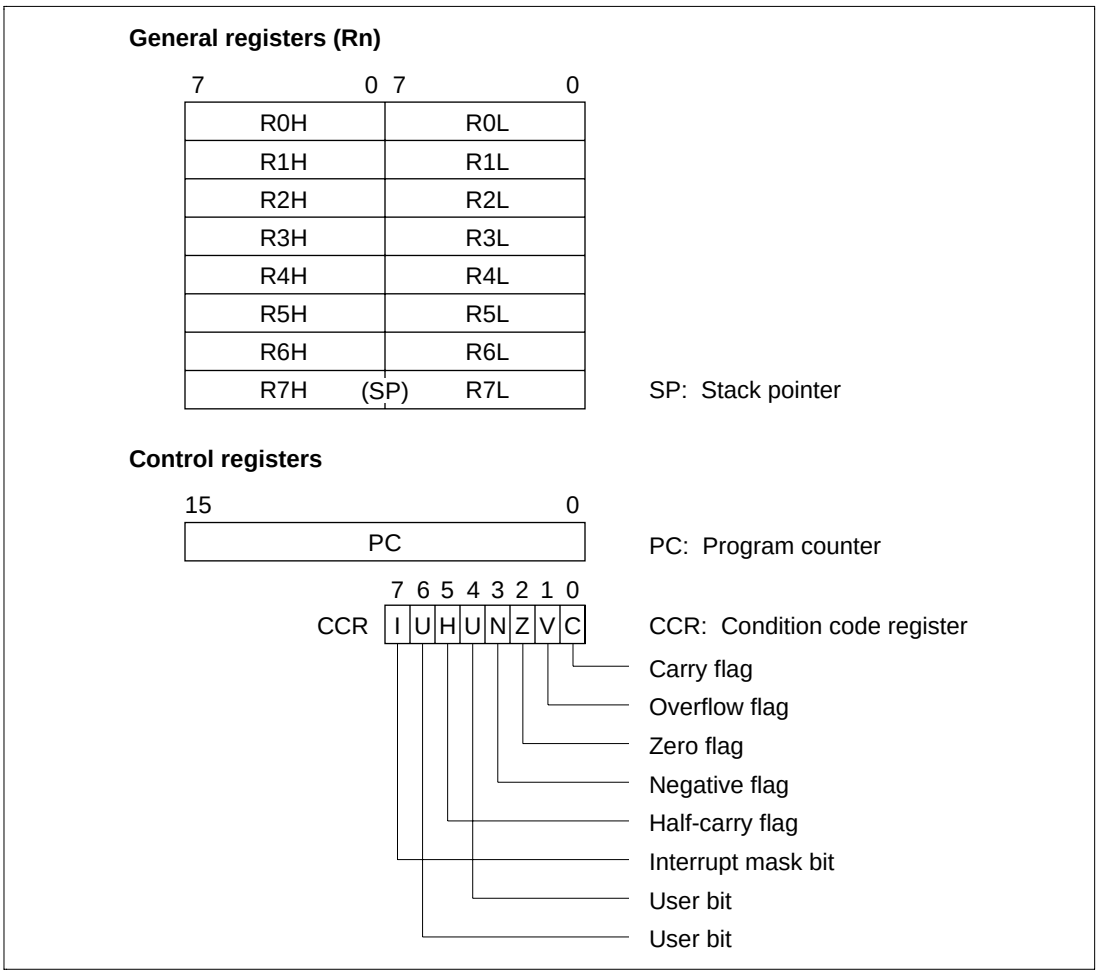


Figure 2.1 CPU Registers

2.2 Register Descriptions

2.2.1 General Registers

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

R7 also functions as the stack pointer, used implicitly by hardware in processing interrupts and subroutine calls. In assembly-language coding, R7 can also be denoted by the letters SP. As indicated in figure 2.2, R7 (SP) points to the top of the stack.

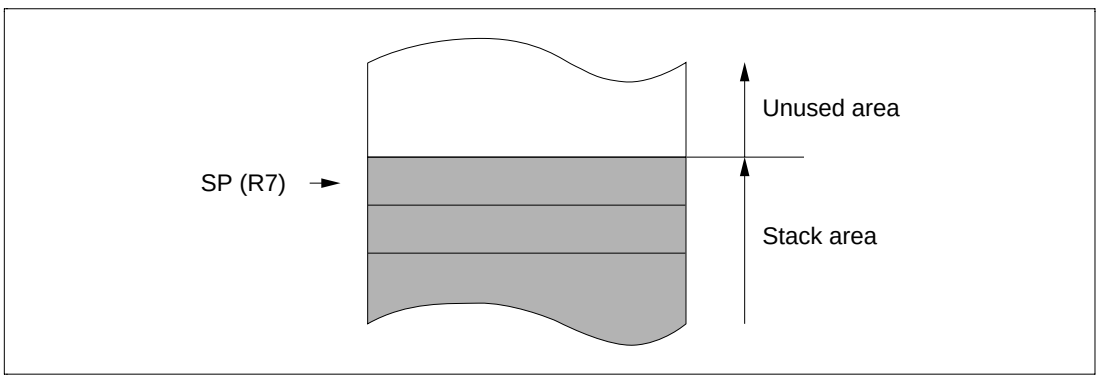


Figure 2.2 Stack Pointer

2.2.2 Control Registers

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

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

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

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

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

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

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

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

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

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

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

- Add and subtract instructions, to indicate a carry or borrow at the most significant bit of the result
- Shift and rotate instructions, to store the value shifted out of the most significant or least significant bit
- Bit manipulation and bit load instructions, as a bit accumulator

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

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

2.2.3 Initial Register Values

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

2.3 Data Formats

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

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

2.3.1 Data Formats in General Registers

Data of all the sizes above can be stored in general registers as shown in figure 2.3.

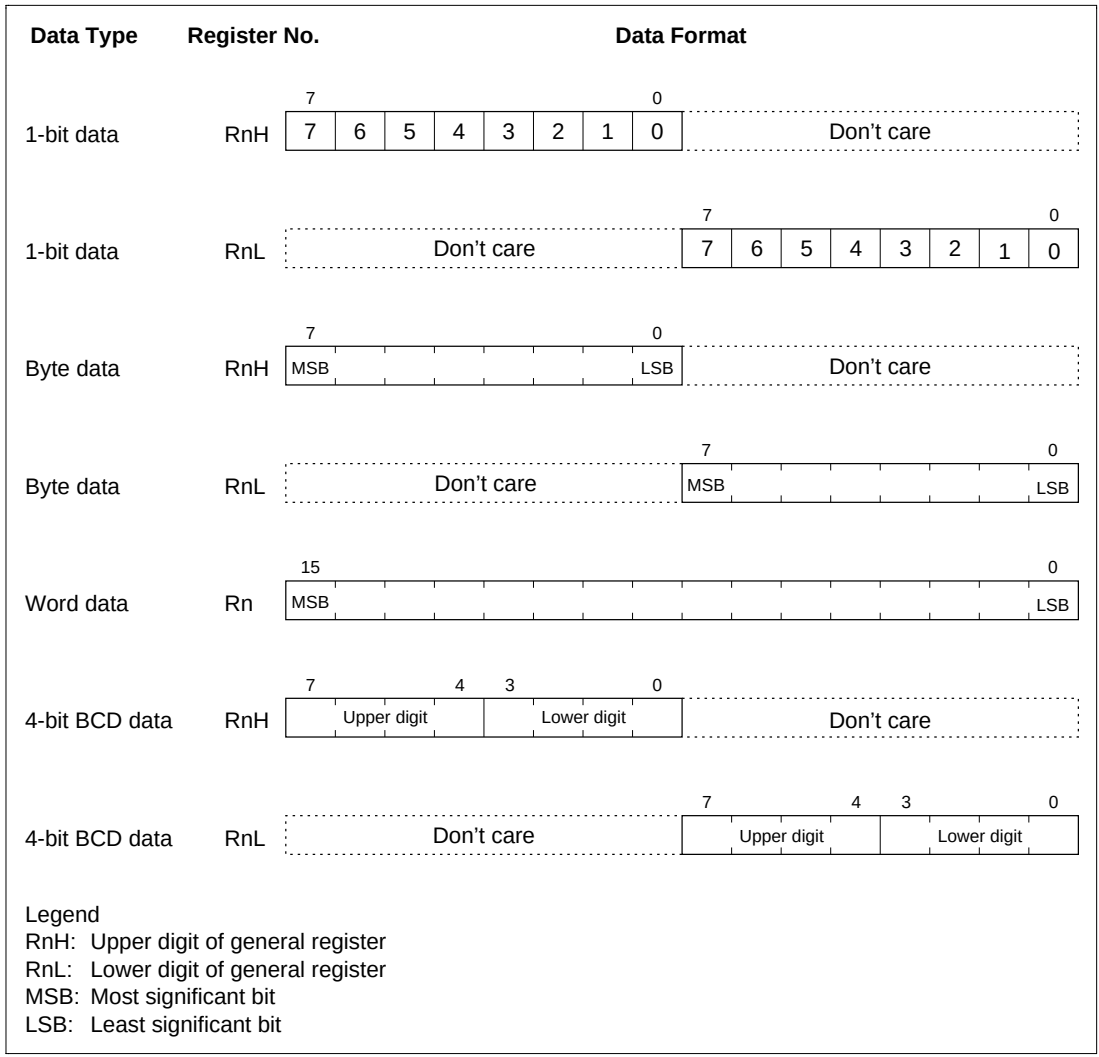


Figure 2.3 Register Data Formats

2.3.2 Memory Data Formats

Figure 2.4 indicates the data formats in memory.

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

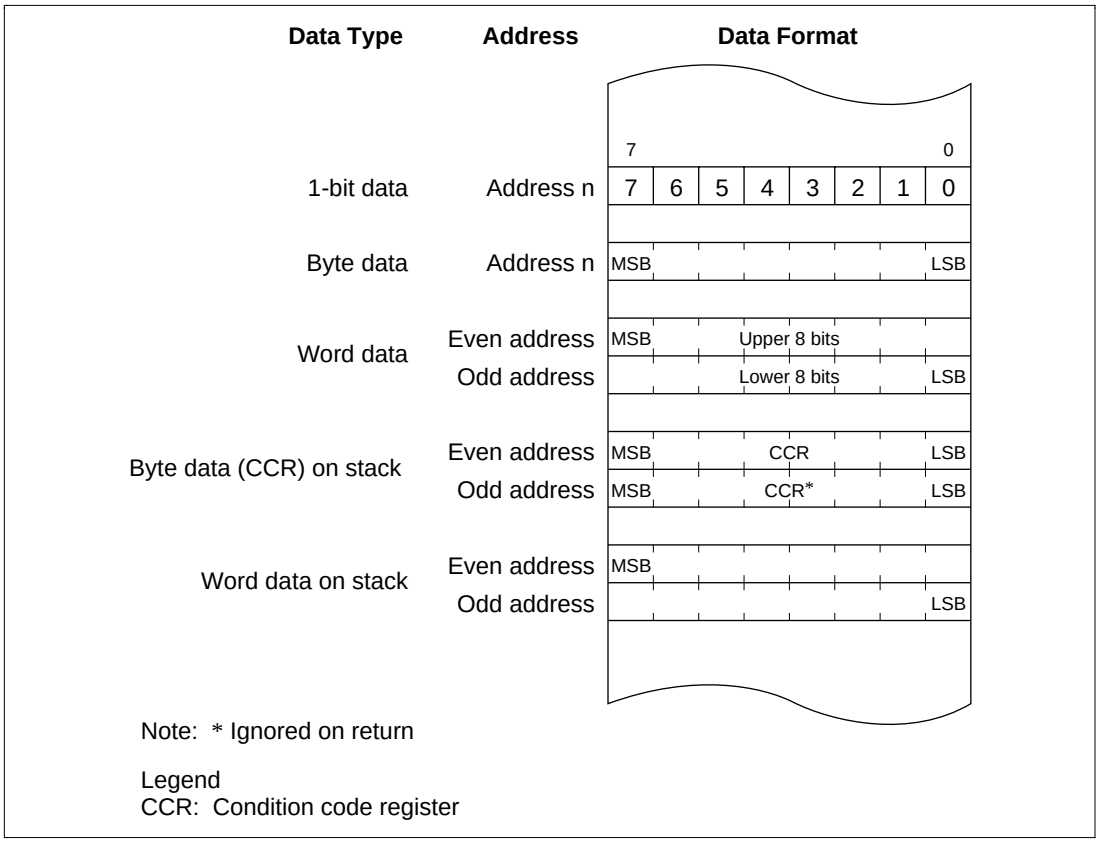


Figure 2.4 Memory Data Formats

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

2.4 Addressing Modes

2.4.1 Addressing Mode

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

Table 2.1 Addressing Modes

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

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

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

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

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

- Register Indirect with Post-Increment—@Rn+

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

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

- Register Indirect with Pre-Decrement—@–Rn

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

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

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

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

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

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

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

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

2.4.2 Calculation of Effective Address

Table 2.2 shows how the H8/300 calculates effective addresses in each addressing mode.

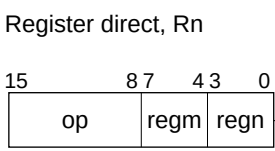
Arithmetic, logic, and shift instructions use register direct addressing (1). The ADD.B, ADDX.B, SUBX.B, CMP.B, AND.B, OR.B, and XOR.B instructions can also use immediate addressing (6).

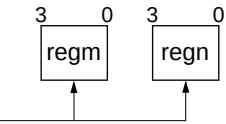
The MOV instruction uses all the addressing modes except program-counter relative (7) and memory indirect (8).

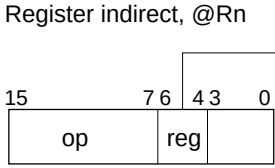
Bit manipulation instructions use register direct (1), register indirect (2), or 8-bit absolute (5) addressing to identify a byte operand, and 3-bit immediate addressing to identify a bit within the byte. The BSET, BCLR, BNOT, and BTST instructions can also use register direct addressing (1) to identify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

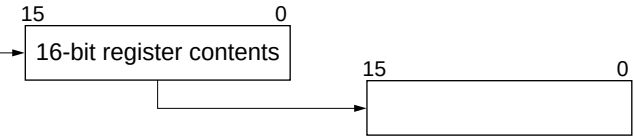
Table 2.2 Effective Address Calculation

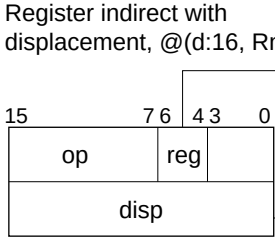
No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address
1	Register direct, Rn		Operands are contained in registers regm and regn
2	Register indirect, @Rn	16-bit register contents	
3	Register indirect with displacement, @(d:16, Rn)	16-bit register contents; disp	
4	Register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

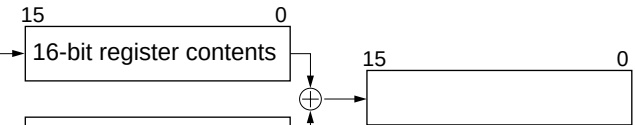


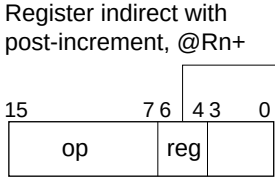


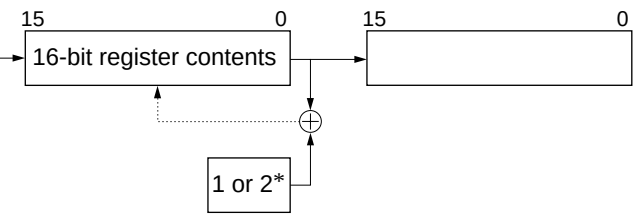


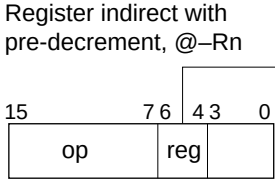


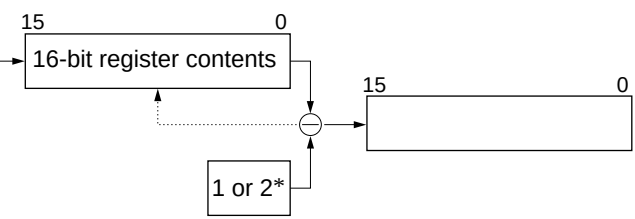












Note: * 1 for a byte operand,
2 for a word operand

Table 2.2 Effective Address Calculation (cont)

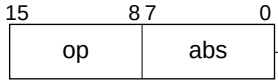
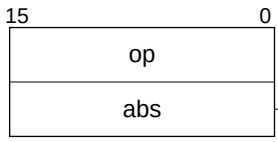
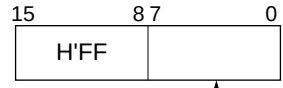
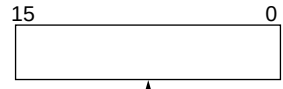
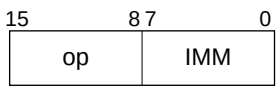
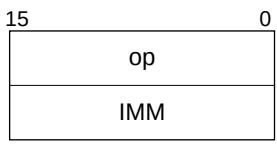
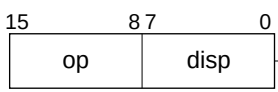
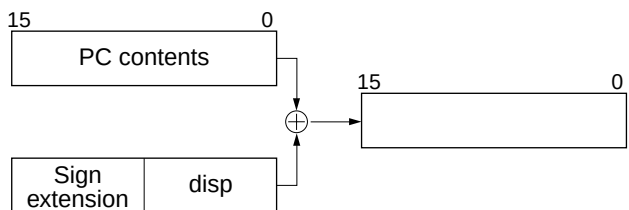
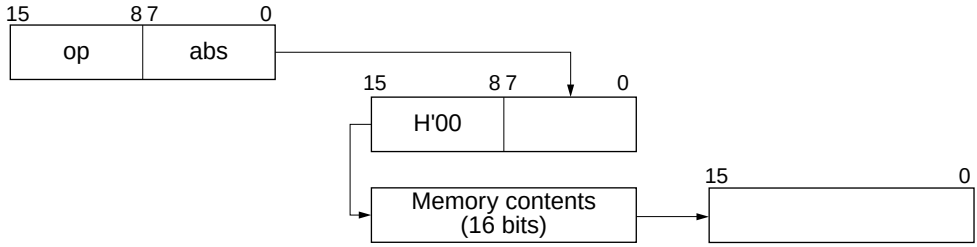
No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address
5	Absolute address @aa:8  @aa:16  	  	
6	Immediate #xx:8  #xx:16  		Operand is 1- or 2-byte immediate data
7	PC-relative @(d:8, PC)  	 	

Table 2.2 Effective Address Calculation (cont)

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

Legend

reg: General register
op: Operation code
disp: Displacement
IMM: Immediate data
abs: Absolute address

2.5 Instruction Set

The H8/300 CPU has 57 types of instructions, which are classified by function in table 2.3.

Table 2.3 Instruction Classification

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

Notes: 1. PUSH Rn is equivalent to MOV.W Rn, @–SP.
POP Rn is equivalent to MOV.W @SP+, Rn.
2. Bcc is a conditional branch instruction in which cc represents a condition code.
3. Not supported by the H8/3318.

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

Operation Notation

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

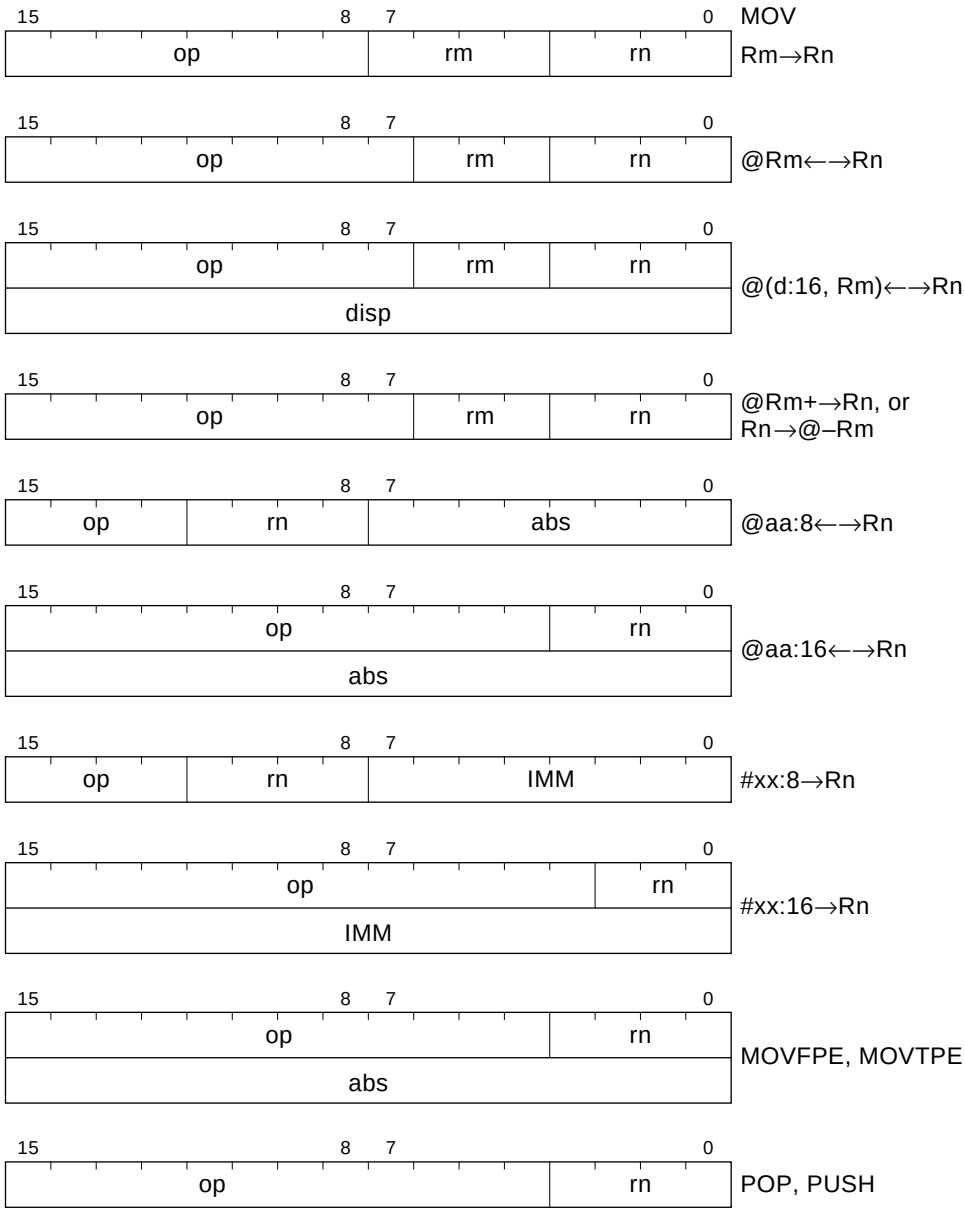
2.5.1 Data Transfer Instructions

Table 2.4 describes the data transfer instructions. Figure 2.5 shows their object code formats.

Table 2.4 Data Transfer Instructions

Instruction	Size*	Function
MOV	B/W	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register. The Rn, @Rn, @(d:16, Rn), @aa:16, #xx:8 or #xx:16, @–Rn, and @Rn+ addressing modes are available for byte or word data. The @aa:8 addressing mode is available for byte data only. The @–R7 and @R7+ modes require word operands. Do not specify byte size for these two modes.
MOVTPE	B	Not supported by the H8/3318.
MOVFPE	B	Not supported by the H8/3318.
PUSH	W	Rn → @–SP Pushes a 16-bit general register onto the stack. Equivalent to MOV.W Rn, @–SP.
POP	W	@SP+ → Rn Pops a 16-bit general register from the stack. Equivalent to MOV.W @SP+, Rn.

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



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

Figure 2.5 Data Transfer Instruction Codes

2.5.2 Arithmetic Operations

Table 2.5 describes the arithmetic instructions. See figure 2.6 in section 2.5.4, Shift Operations for their object codes.

Table 2.5 Arithmetic Instructions

Instruction	Size*	Function
ADD	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 or borrow on byte data in two general registers, or addition or subtraction 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 immediate data to or from data in a general register. The immediate data must be 1 or 2.
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. Word data can be compared only between two general registers.
NEG	B	0 – Rd → Rd
		Obtains the two's complement (arithmetic complement) of data in a general register.

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

2.5.3 Logic Operations

Table 2.6 describes the four instructions that perform logic operations. See figure 2.6 in section 2.5.4, Shift Operations, for their object codes.

Table 2.6 Logic Operation Instructions

Instruction	Size*	Function
AND	B	Rd ∧ Rs → Rd, Rd ∧ #imm → Rd Performs a logical AND operation on a general register and another general register or immediate data.
OR	B	Rd ∨ Rs → Rd, Rd ∨ #imm → Rd Performs a logical OR operation on a general register and another general register or immediate data.
XOR	B	Rd ⊕ Rs → Rd, Rd ⊕ #imm → Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data.
NOT	B	¬ (Rd) → (Rd) Obtains the one's complement (logical complement) of general register contents.

Note: * Size: Operand size
B: Byte

2.5.4 Shift Operations

Table 2.7 describes the eight shift instructions. Figure 2.6 shows the object code formats of the arithmetic, logic, and shift instructions.

Table 2.7 Shift Instructions

Instruction	Size*	Function
SHAL	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 through carry → Rd
ROTXR		Rotates general register contents through the C (carry) bit.

Note: * Size: Operand size
B: Byte

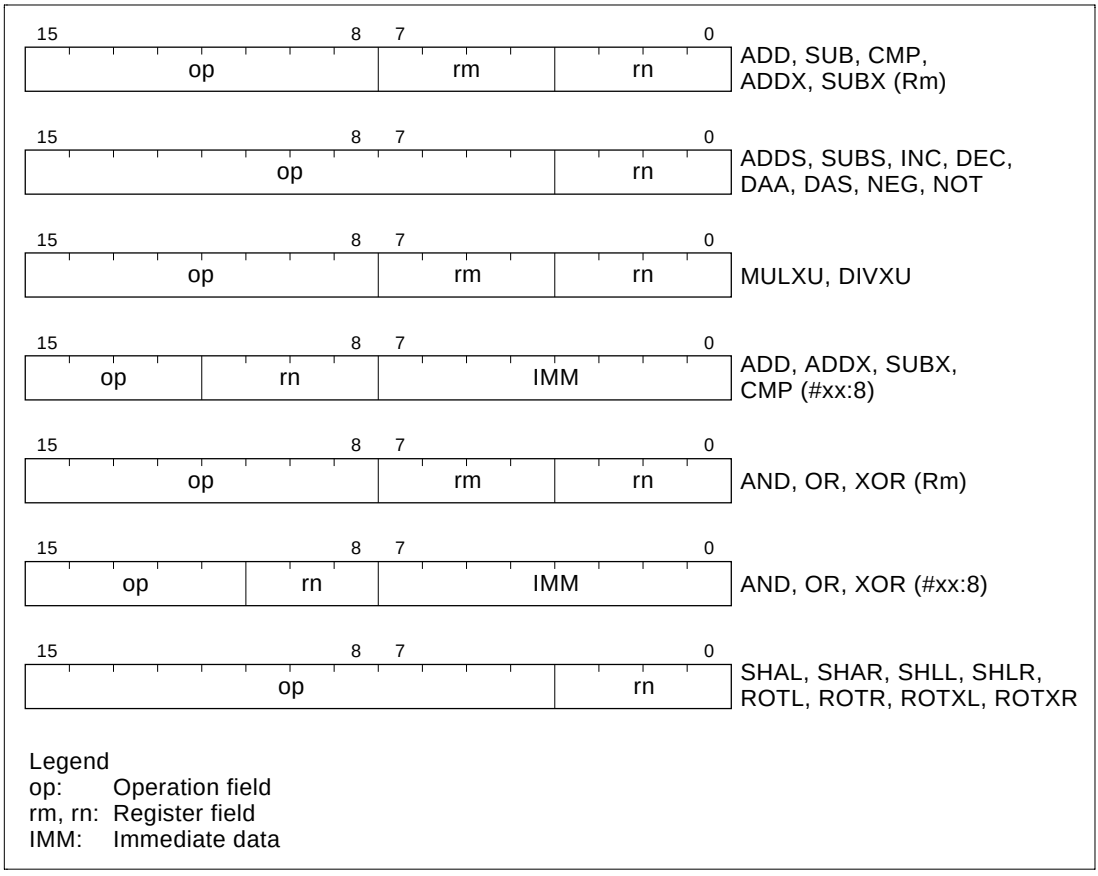


Figure 2.6 Arithmetic, Logic, and Shift Instruction Codes

2.5.5 Bit Manipulations

Table 2.8 describes the bit-manipulation instructions. Figure 2.7 shows their object code formats.

Table 2.8 Bit-Manipulation Instructions

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

Note: * Size: Operand size
B: Byte

Table 2.8 Bit-Manipulation Instructions (cont)

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

Note: * Size: Operand size
B: Byte

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

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

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

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

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

Before Execution of BCLR Instruction

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

Execution of BCLR Instruction

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

After Execution of BCLR Instruction

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

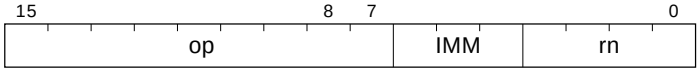
Explanation: To execute the BCLR instruction, the CPU begins by reading P4DDR. Since P4DDR is a write-only register, it is read as H'FF, even though its true value is H'3F.

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

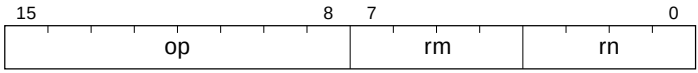
Finally, the CPU writes this value (H'FE) back to P4DDR to complete the BCLR instruction.

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

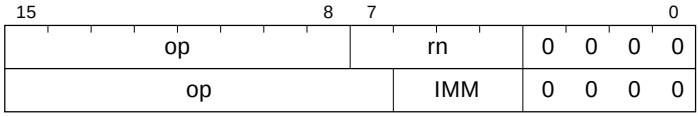
BSET, BCLR, BNOT, BTST



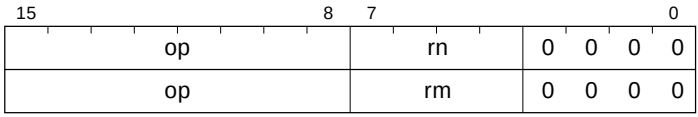
Operand: register direct (Rn)
Bit No.: immediate (#xx:3)



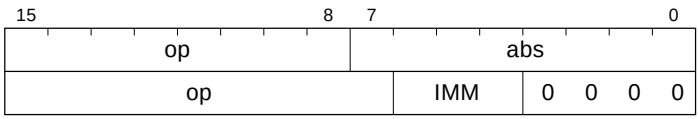
Operand: register direct (Rn)
Bit No.: register direct (Rm)



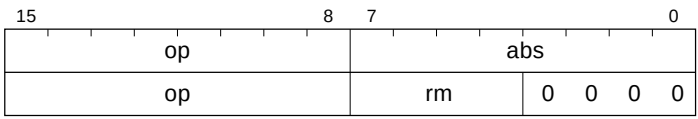
Operand: register indirect (@Rn)
Bit No.: immediate (#xx:3)



Operand: register indirect (@Rn)
Bit No.: register direct (Rm)

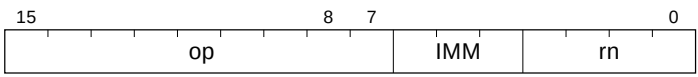


Operand: absolute (@aa:8)
Bit No.: immediate (#xx:3)

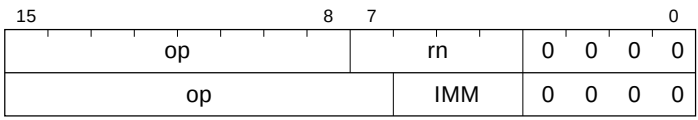


Operand: absolute (@aa:8)
Bit No.: register direct (Rm)

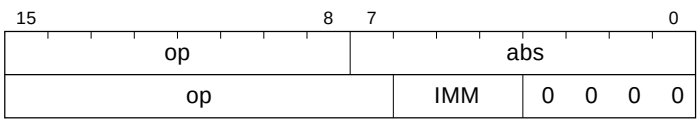
BAND, BOR, BXOR, BLD, BST



Operand: register direct (Rn)
Bit No.: immediate (#xx:3)



Operand: register indirect (@Rn)
Bit No.: immediate (#xx:3)



Operand: absolute (@aa:8)
Bit No.: immediate (#xx:3)

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

Figure 2.7 Bit Manipulation Instruction Codes

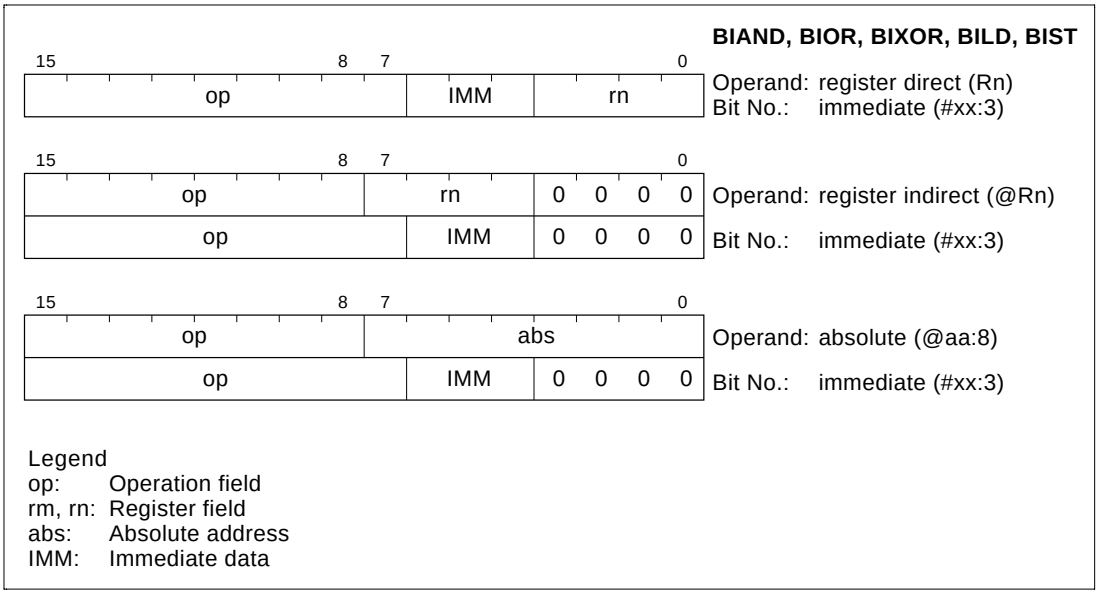


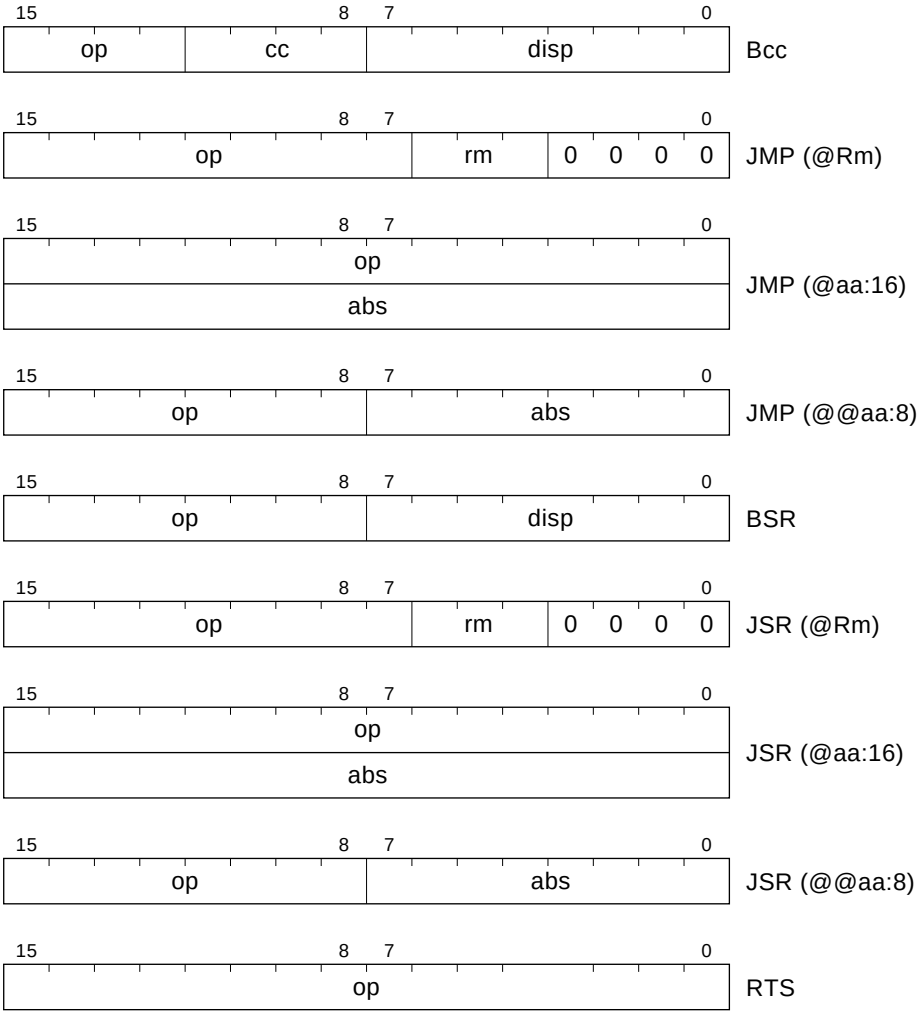
Figure 2.7 Bit Manipulation Instruction Codes (cont)

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



Legend
op: Operation field
cc: Condition field
rm: Register field
disp: Displacement
abs: Absolute address

Figure 2.8 Branching Instruction Codes

2.5.7 System Control Instructions

Table 2.10 describes the system control instructions. Figure 2.9 shows their object code formats.

Table 2.10 System Control Instructions

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

Note: * Size: Operand size
B: Byte

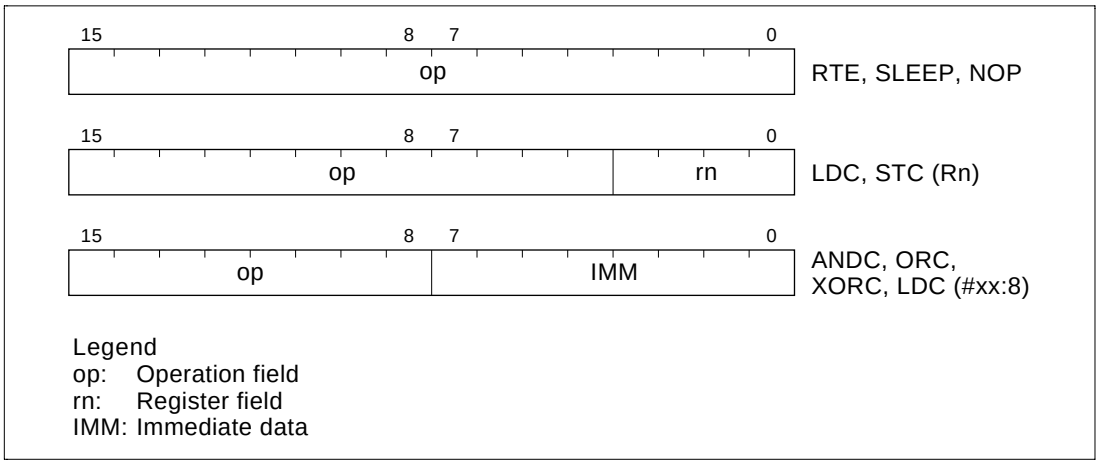


Figure 2.9 System Control Instruction Codes

2.5.8 Block Data Transfer Instruction

Table 2.11 describes the EEPMOV instruction. Figure 2.10 shows its object code format.

Table 2.11 Block Data Transfer Instruction/EEPROM Write Operation

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

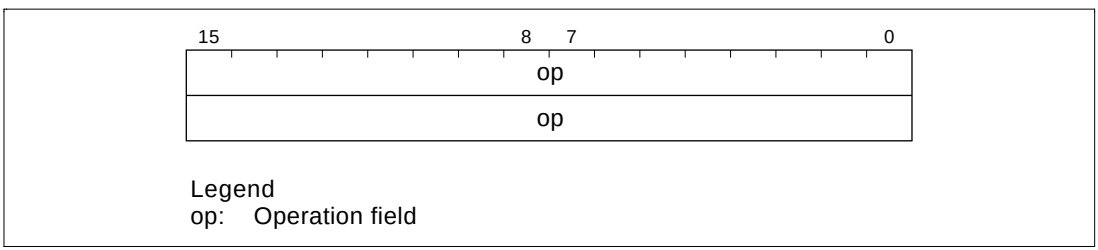
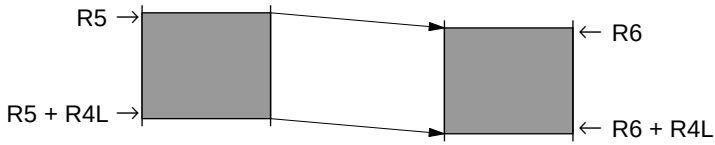


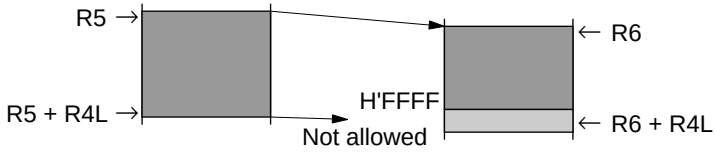
Figure 2.10 Block Data Transfer Instruction/EEPROM Write Operation Code

Notes on EEPMOV Instruction

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



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



2.6 CPU States

2.6.1 Overview

The CPU has three states: the program execution state, exception-handling state, and power-down state. The power-down state is further divided into three modes: sleep mode, software standby mode, and hardware standby mode. Figure 2.11 summarizes these states, and figure 2.12 shows a map of the state transitions.

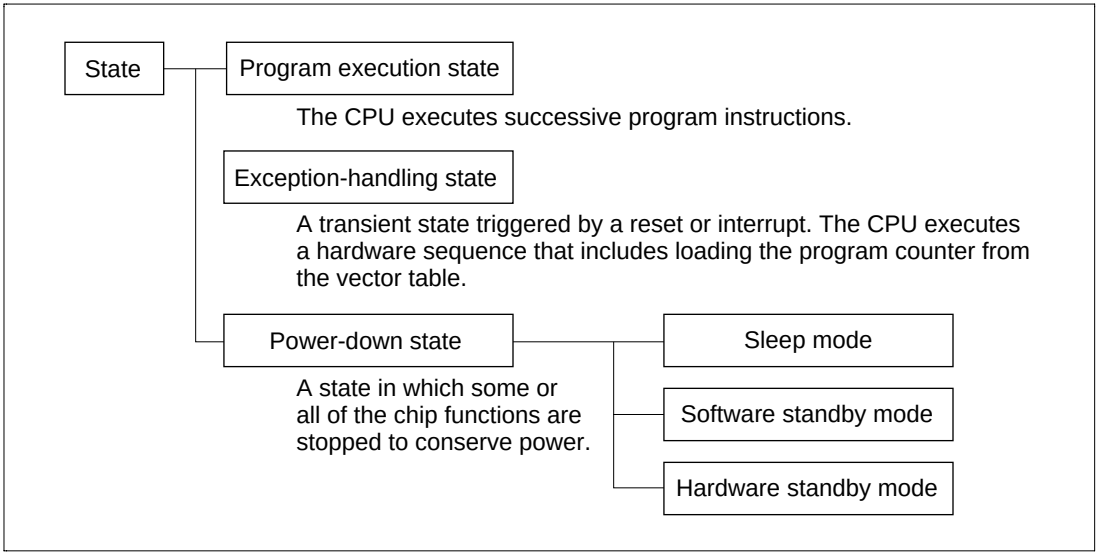
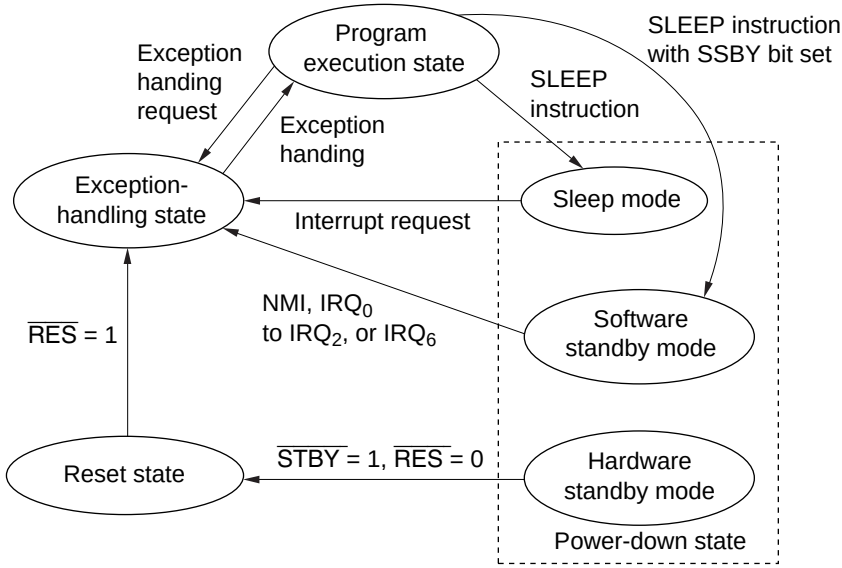


Figure 2.11 Operating States



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

Figure 2.12 State Transitions

2.6.2 Program Execution State

In this state the CPU executes program instructions.

2.6.3 Exception-Handling State

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

2.6.4 Power-Down State

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

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

Software Standby Mode: Is entered if the SLEEP instruction is executed while the SSBY (software standby) bit in the system control register (SYSCR) is set to 1. The CPU and all on-chip supporting modules halt. The on-chip supporting modules are initialized, but the contents of the on-chip RAM and CPU registers remain unchanged as long as a specified voltage is supplied. I/O port outputs also remain unchanged.

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

See section 18, Power-Down State, for further information.

2.7 Access Timing and Bus Cycle

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

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

On-chip ROM and RAM are accessed in a cycle of two states designated T_1 and T_2. Either byte or word data can be accessed, via a 16-bit data bus. Figure 2.13 shows the on-chip memory access cycle. Figure 2.14 shows the associated pin states.

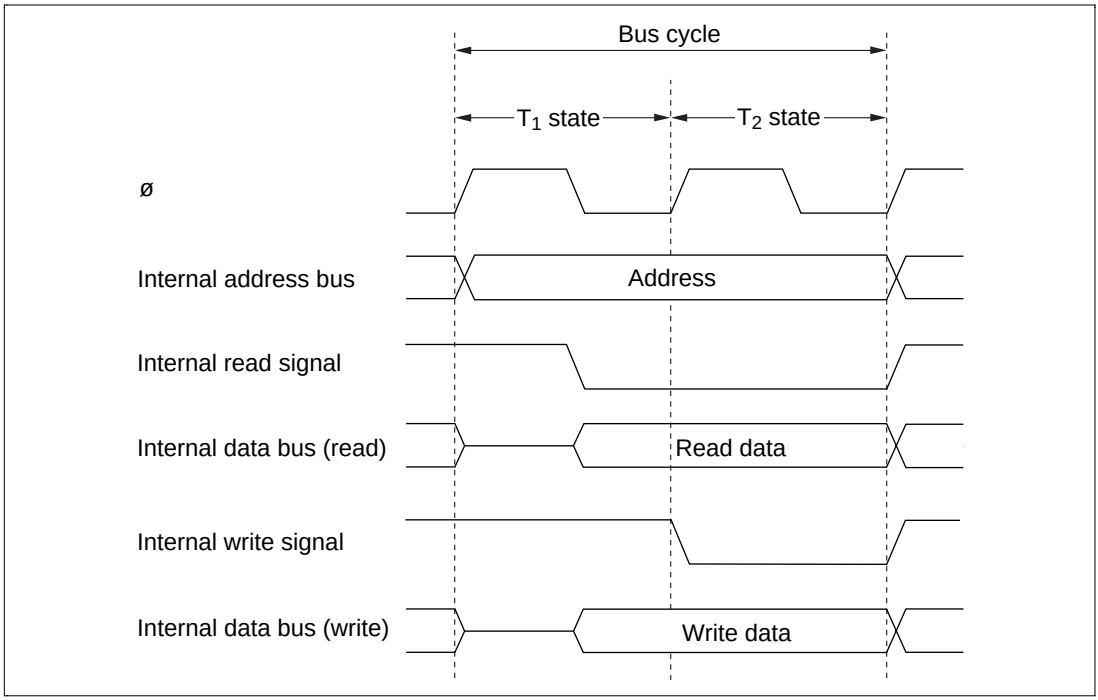


Figure 2.13 On-Chip Memory Access Cycle

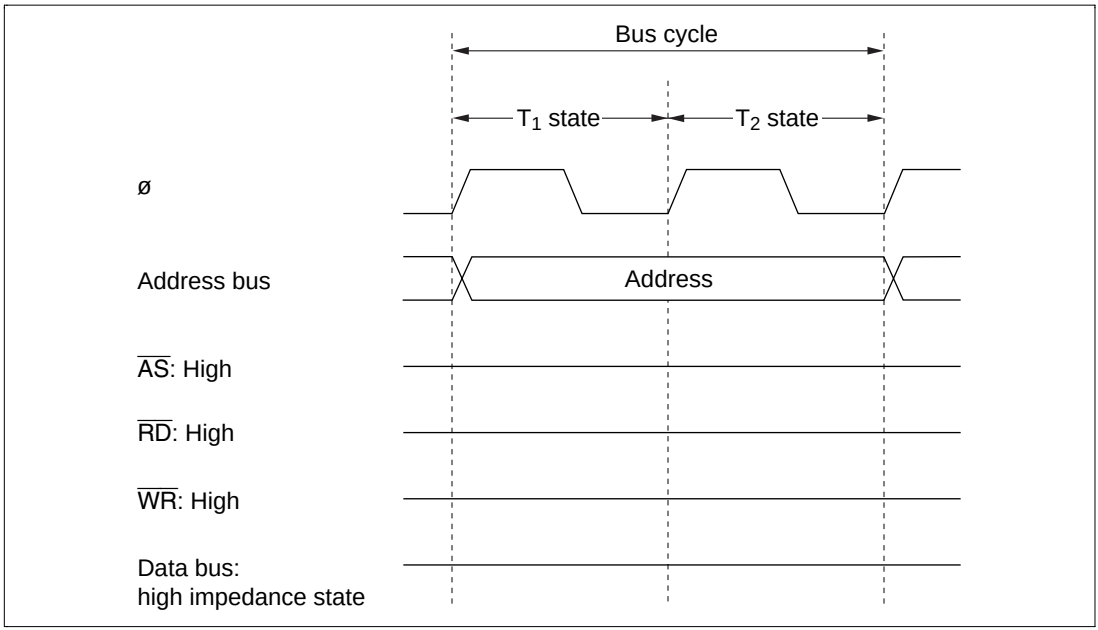


Figure 2.14 Pin States during On-Chip Memory Access Cycle

2.7.2 Access to On-Chip Register Field and External Devices

The on-chip supporting module registers and external devices are accessed in a cycle consisting of three states: T_1, T_2, and T_3. Only one byte of data can be accessed per cycle, via an 8-bit data bus. Access to word data or instruction codes requires two consecutive cycles (six states). Figure 2.15 shows the access cycle for the on-chip register field. Figure 2.16 shows the associated pin states. Figures 2.17 (a) and (b) show the read and write access timing for external devices.

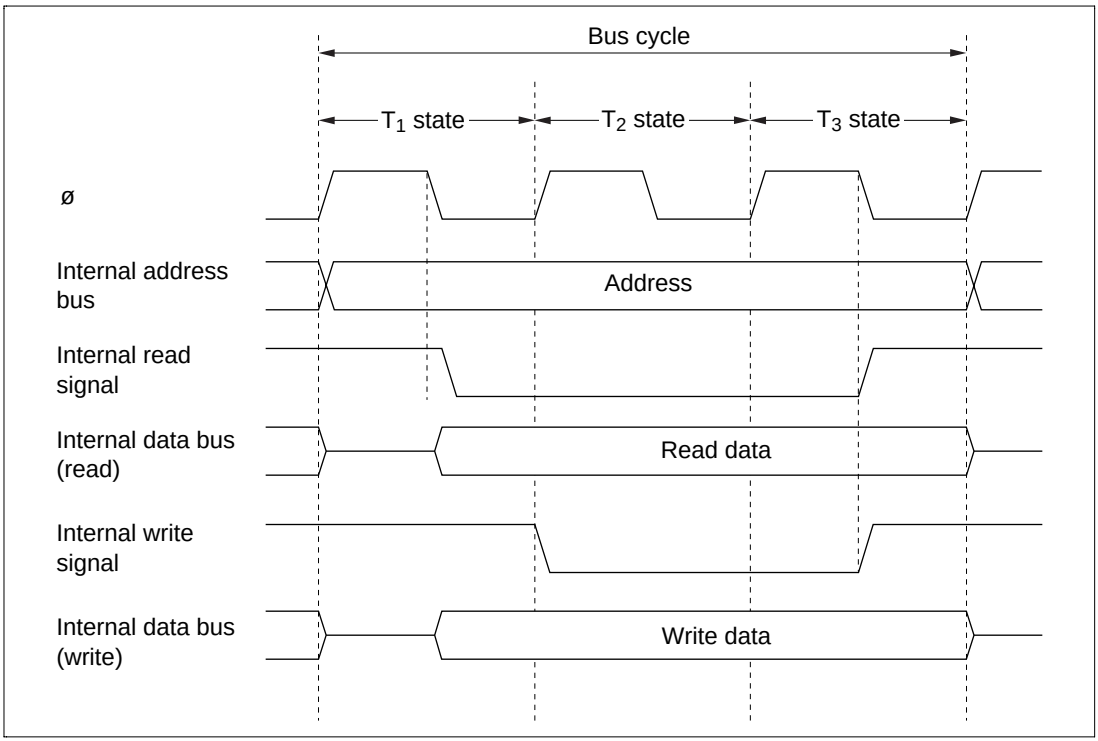


Figure 2.15 On-Chip Register Field Access Cycle

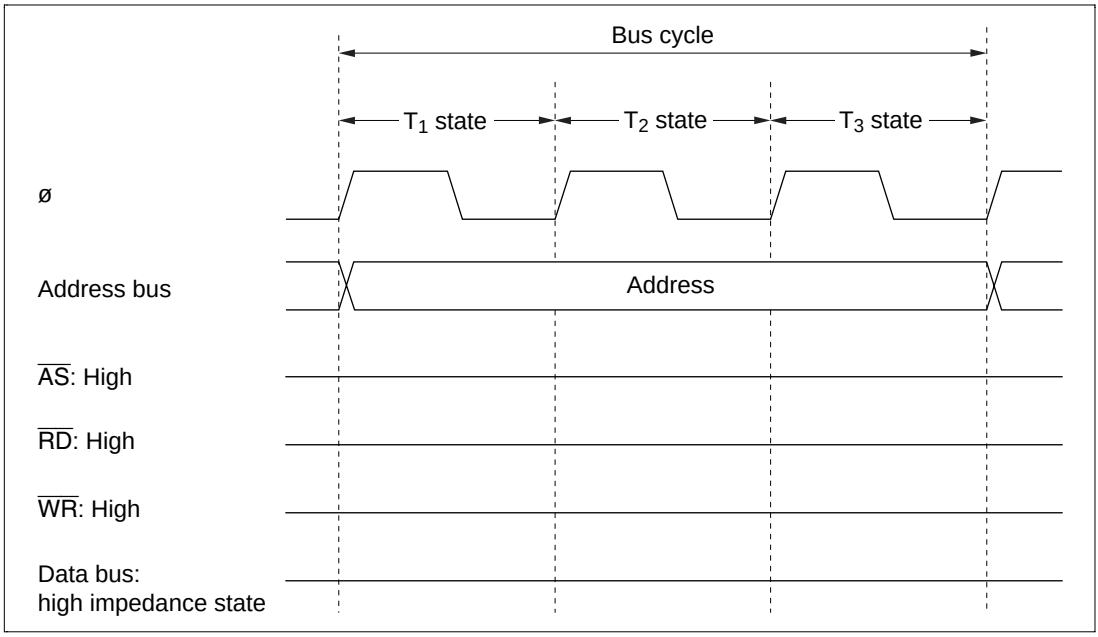


Figure 2.16 Pin States during On-Chip Register Field Access Cycle

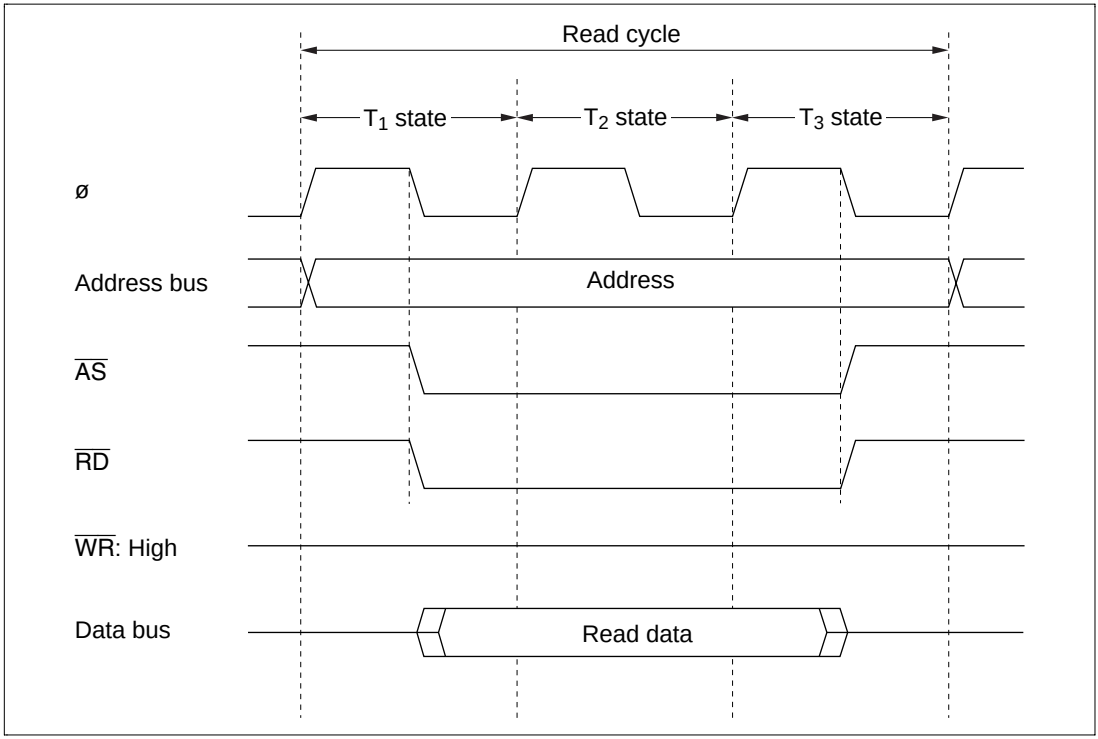


Figure 2.17 (a) External Device Access Timing (Read)

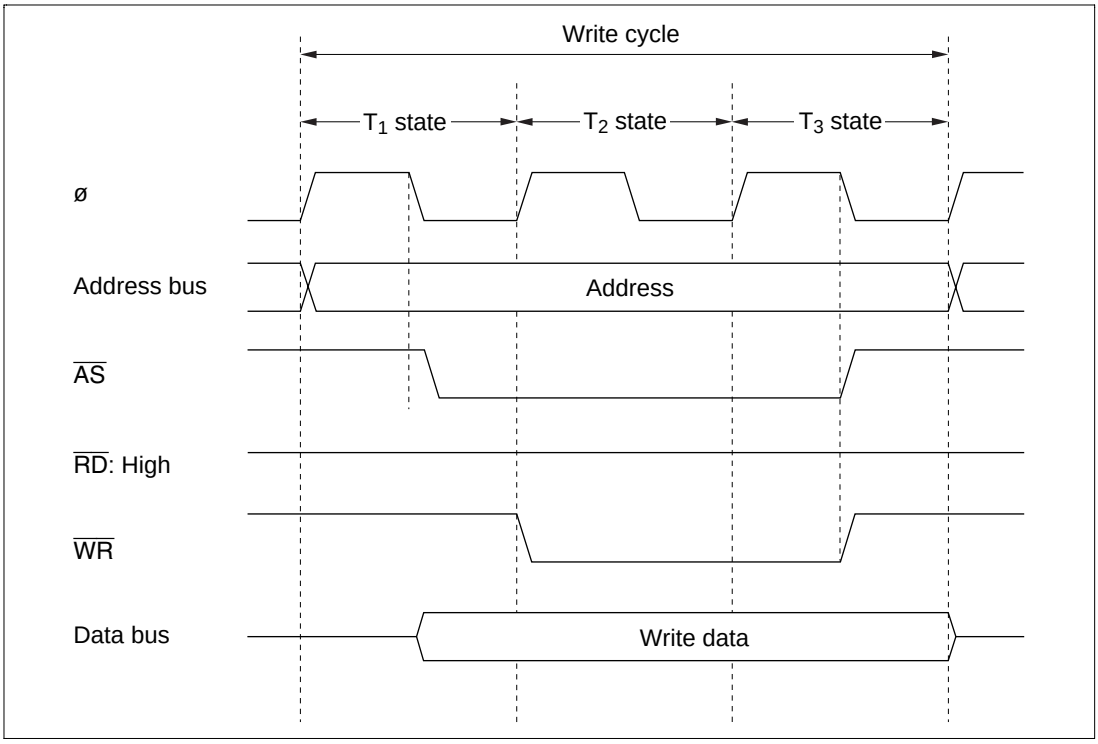


Figure 2.17 (b) External Device Access Timing (Write)

Section 3 MCU Operating Modes and Address Space

3.1 Overview

3.1.1 Mode Selection

The H8/3318 operates in three modes numbered 1, 2, and 3. The mode is selected by the inputs at the mode pins (MD_1 and MD_0) when the chip comes out of a reset. See table 3.1.

Table 3.1 Operating Modes

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

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

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

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

Mode 0 is inoperative in the H8/3318. Avoid setting the mode pins to mode 0. In addition, the mode pins must not be changed during MCU operation.

3.1.2 Mode and System Control Registers

Table 3.2 lists the registers related to the chip's operating mode: the system control register (SYSCR) and mode control register (MDCR). The mode control register indicates the inputs to the mode pins MD_1 and MD_0.

Table 3.2 Mode and System Control Registers

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

3.2 System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	XRST	NMIEG	DPME	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

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

Bit 7—Software Standby (SSBY): Enables transition to the software standby mode. For details, see section 18, Power-Down State.

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

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

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the clock settling time when the chip recovers from the software standby mode by an external interrupt. During the selected time the CPU and on-chip supporting modules continue to stand by. These bits should be set according to the clock frequency so that the settling time is at least 8 ms. For specific settings, see section 18.3.3, Clock Settling Time for Exit from Software Standby Mode.

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

Bit 3—External Reset (XRST): Indicates the source of a reset. A reset can be triggered by an external reset input, or by a watchdog timer overflow when the watchdog timer is in operation. XRST is set to 1 by an external reset, and cleared to 0 by a watchdog timer overflow.

Bit 3 XRST	Description	
0	Reset generated by watchdog timer overflow	
1	Reset generated by external reset input	(Initial value)

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

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

Bit 1—Dual-Port RAM Mode Enable (DPME): Selects whether to put the chip into slave mode.

Bit 1 DPME	Description	
0	The chip is not put into slave mode	(Initial value)
1	The chip is put into slave mode	

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized by the rising edge of the $\overline{\text{RES}}$ signal, but is not initialized in the software standby mode.

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

3.3 Mode Control Register (MDCR)

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

Note: * Initialized according to MD_1 and MD_0 inputs.

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

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

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

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

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

3.4 Address Space Map in Each Operating Mode

Figure 3.1 shows memory map of the H8/3318 in modes 1, 2, and 3.

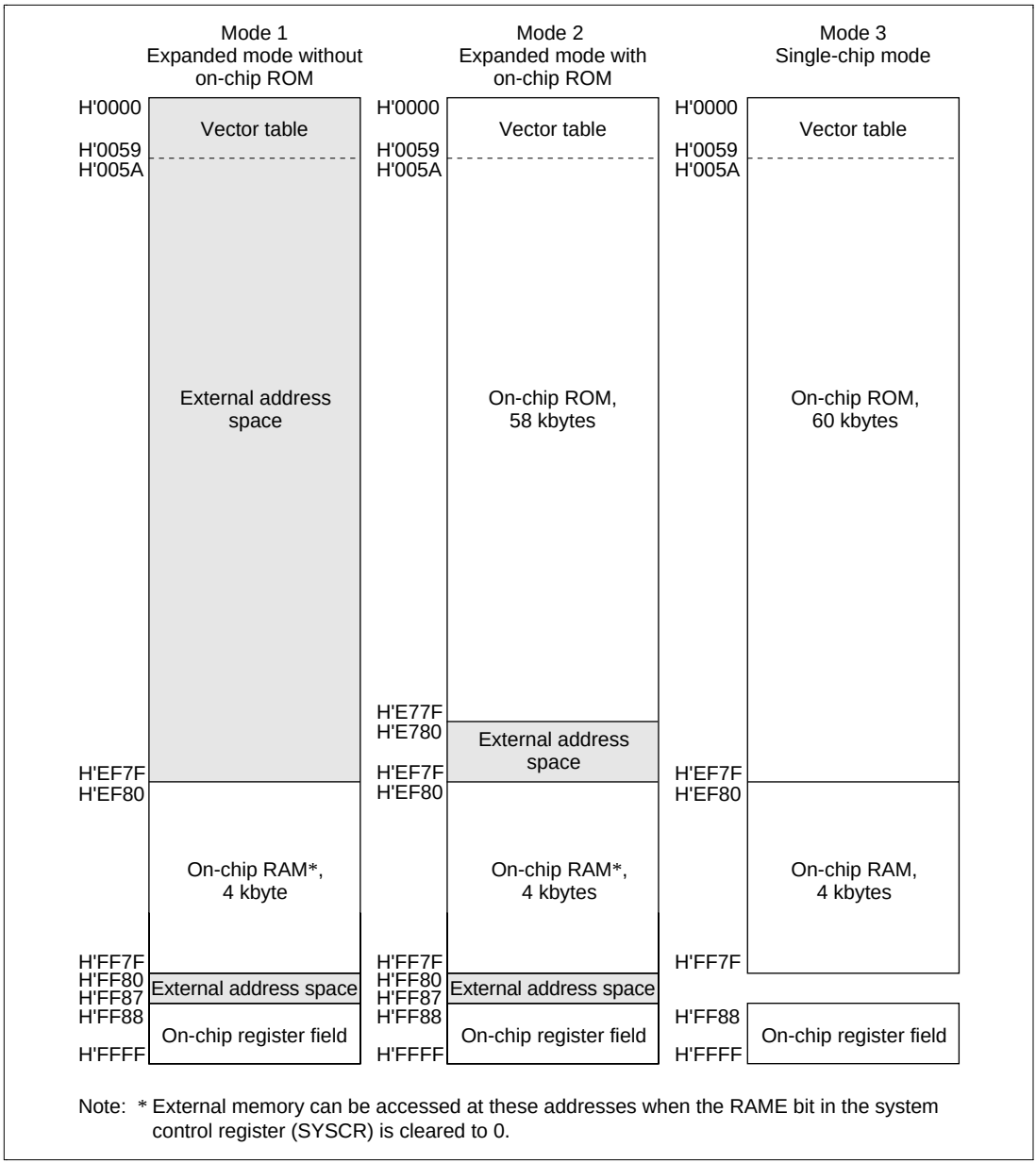


Figure 3.1 H8/3318 Address Space Map

Section 4 Exception Handling

4.1 Overview

The H8/3318 recognizes only two kinds of exceptions: interrupts and the reset. Table 4.1 indicates their priority and the timing of their hardware exception-handling sequence.

Table 4.1 Hardware Exception-Handling Sequences and Priority

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

Note: * Not detected in the case of the ANDC, ORC, XORC, and LDC instructions.

4.2 Reset

4.2.1 Overview

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

4.2.2 Reset Sequence

The reset state begins when $\overline{\text{RES}}$ goes low or if there is a watchdog timer reset. To ensure correct resetting, at power-on the $\overline{\text{RES}}$ pin should be held low for at least 20 ms. In a reset during operation, the $\overline{\text{RES}}$ pin should be held low for at least 10 system clock cycles. Watchdog timer reset pulse widths must be 518 system clock cycles. For the pin states during a reset, see appendix D, Pin States.

When a reset occurs, hardware carries out the following reset exception-handling sequence.

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

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

Figure 4.1 indicates the timing of the reset sequence in modes 2 and 3. Figure 4.2 indicates the timing in mode 1.

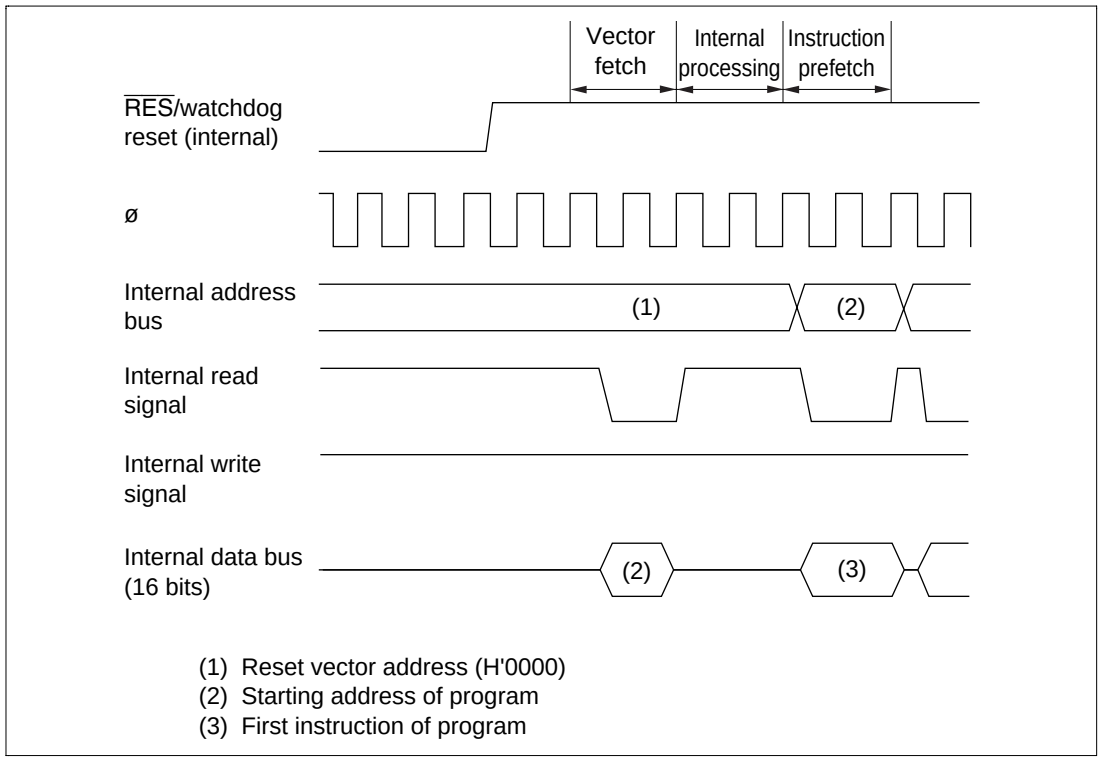


Figure 4.1 Reset Sequence (Mode 2 or 3, Program Stored in On-Chip ROM)

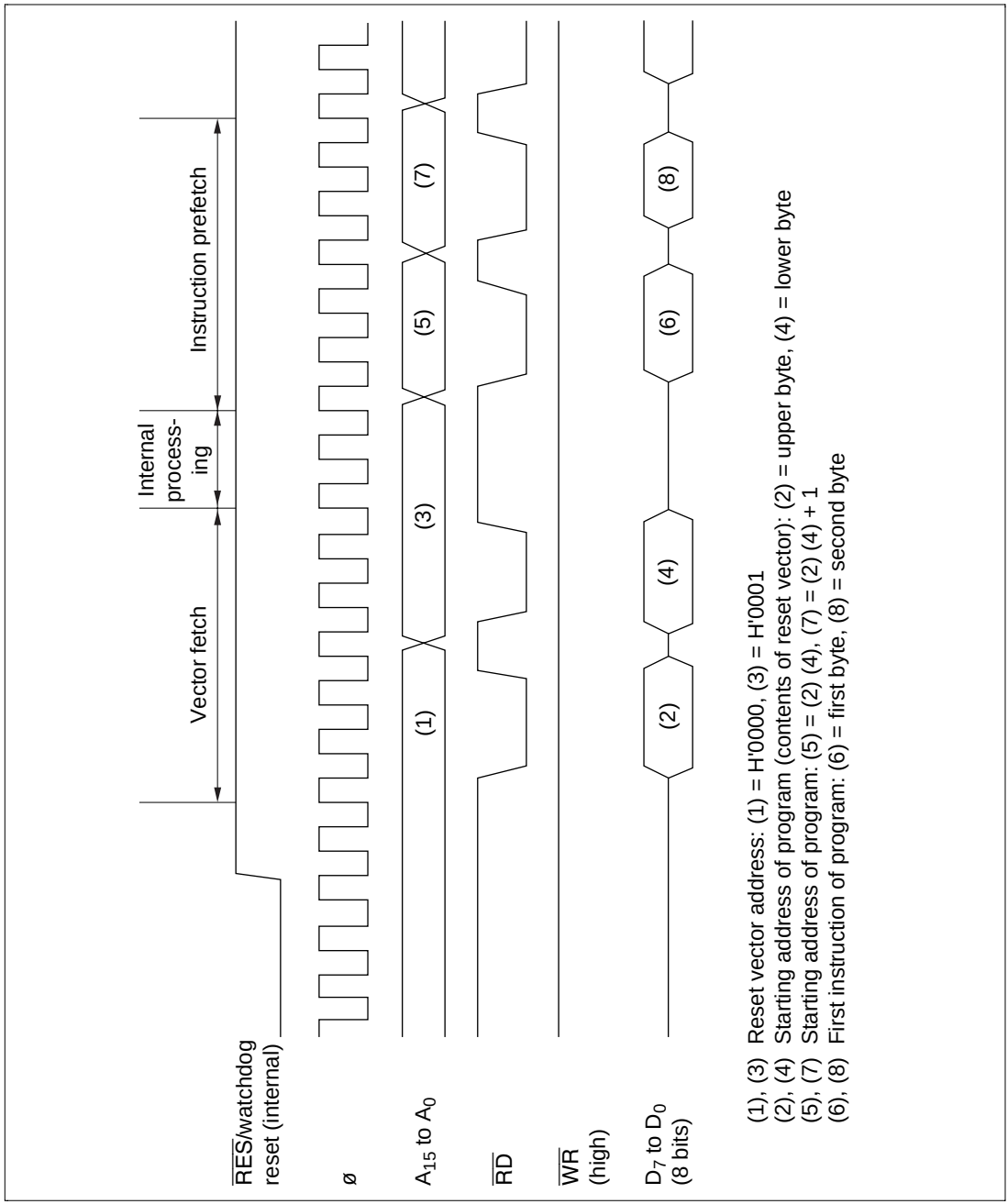


Figure 4.2 Reset Sequence (Mode 1)

4.2.3 Disabling of Interrupts after Reset

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

To confirm the CCR contents after reset exception handling, a CCR manipulation instruction can be executed before the instruction that initializes the stack pointer. All interrupts, including NMI, are disabled immediately after execution of a CCR manipulation instruction. The next instruction should be the instruction that initializes the stack pointer.

4.3 Interrupts

4.3.1 Overview

The interrupt sources include nine input pins for external interrupts (NMI, IRQ0 to IRQ7) and 33 internal sources in the on-chip supporting modules. Table 4.2 lists the interrupt sources in priority order and gives their vector addresses. When two or more interrupts are requested, the interrupt with highest priority is served first.

The features of these interrupts are:

- NMI has the highest priority and is always accepted. All internal and external interrupts except NMI can be masked by the I bit in the CCR. When the I bit is set to 1, interrupts other than NMI are not accepted.
- IRQ_0 to IRQ_7 can be sensed on the falling edge of the input signal, or level-sensed. The type of sensing can be selected for each interrupt individually. NMI is edge-sensed, and either the rising or falling edge can be selected.
- All interrupts are individually vectored. The software interrupt-handling routine does not have to determine what type of interrupt has occurred.
- The watchdog timer can be used to generate either an NMI interrupt or OVF interrupt, as needed. Refer to section 13, Watchdog Timer, for details.

Table 4.2 Interrupts

Interrupt Source			Vector No.	Vector Address	Priority
Reset			0	H'0000 to H'0001	High
Reserved			1 to 2	H'0002 to H'0005	
NMI			3	H'0006 to H'0007	
IRQ_0			4	H'0008 to H'0009	
IRQ_1			5	H'000A to H'000B	
IRQ_2			6	H'000C to H'000D	
IRQ_3			7	H'000E to H'000F	
IRQ_4			8	H'0010 to H'0011	
IRQ_5			9	H'0012 to H'0013	
IRQ_6			10	H'0014 to H'0015	
IRQ_7			11	H'0016 to H'0017	
16-bit free-running timer 0	ICIA0	(interrupt capture A)	12	H'0018 to H'0019	
	ICIB0	(interrupt capture B)	13	H'001A to H'001B	
	ICIC0	(interrupt capture C)	14	H'001C to H'001D	
	ICID0	(interrupt capture D)	15	H'001E to H'001F	
	OCIA0	(output compare A)	16	H'0020 to H'0021	
	OCIB0	(output compare B)	17	H'0022 to H'0023	
	FOVI0	(overflow)	18	H'0024 to H'0025	
16-bit free-running timer 1	ICI1	(input capture)	19	H'0026 to H'0027	
	OCIA1	(output compare A)	20	H'0028 to H'0029	
	OCIB1	(output compare B)	21	H'002A to H'002B	
	FOVI1	(overflow)	22	H'002C to H'002D	
8-bit timer 0	CMI0A	(compare match A)	23	H'002E to H'002F	
	CMI0B	(compare match B)	24	H'0030 to H'0031	
	OVI0	(overflow)	25	H'0032 to H'0033	
8-bit timer 1	CMI1A	(compare match A)	26	H'0034 to H'0035	
	CMI1B	(compare match B)	27	H'0036 to H'0037	
	OVI1	(overflow)	28	H'0038 to H'0039	
Serial communi-cation interface 0	ERI0	(receive error)	29	H'003A to H'003B	
	RXI0	(receive end)	30	H'003C to H'003D	
	TXI0	(TDR empty)	31	H'003E to H'003F	
	TEI0	(TSR empty)	32	H'0040 to H'0041	Low

Table 4.2 Interrupts (cont)

Interrupt Source			Vector No.	Vector Address	Priority
Serial communi-cation interface 1	ERI1	(receive error)	33	H'0042 to H'0043	High
	RXI1	(receive end)	34	H'0044 to H'0045	
	TXI1	(TDR empty)	35	H'0046 to H'0047	
	TEI1	(TSR empty)	36	H'0048 to H'0049	
A/D converter	ADI	(conversion end)	37	H'004A to H'004B	
Data transfer unit	MWEI	(write end)	38	H'004C to H'004D	
	MREI	(read end)	39	H'004E to H'004F	
	DTIA	(transfer end A)	40	H'0050 to H'0051	
	DTIB	(transfer end B)	41	H'0052 to H'0053	
	DTIC	(transfer end C)	42	H'0054 to H'0055	
	CMPI	(overrun error)	43	H'0056 to H'0057	
Watchdog timer	OVF	(watchdog overflow)	44	H'0058 to H'0059	Low

Notes: 1. Reset vectors are located at H'0000 and H'0001.
2. H'0002 to H'0005 is a reserved area unavailable to the user.

4.3.2 Interrupt-Related Registers

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

Table 4.3 Registers Read by Interrupt Controller

Name	Abbreviation	Read/Write	Address
System control register	SYSCR	R/W	H'FFC4
IRQ sense control register	ISCR	R/W	H'FFC6
IRQ enable register	IER	R/W	H'FFC7

System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	XRST	NMIEG	DPME	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

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

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

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

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

IRQ Sense Control Register (ISCR)

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

Bits 7 to 0—IRQ7 to IRQ0 Sense Control (IRQ7SC to IRQ0SC): These bits determine whether $\overline{\text{IRQ}_7}$ to $\overline{\text{IRQ}_0}$ are level-sensed or sensed on the falling edge.

Bits 7 to 0 IRQ7SC to IRQ0SC	Description	
0	An interrupt is generated when $\overline{\text{IRQ}_7}$ to $\overline{\text{IRQ}_0}$ inputs are low	(Initial state)
1	An interrupt is generated by the falling edge of the $\overline{\text{IRQ}_7}$ to $\overline{\text{IRQ}_0}$ inputs	

IRQ Enable Register (IER)

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

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

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

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

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

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

4.3.3 External Interrupts

The nine external interrupts are NMI and IRQ_0 to IRQ_7. NMI, IRQ_0 to IRQ_2, and IRQ_6 can be used to recover from software standby mode.

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

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

When one of these interrupts is accepted, the I bit is set to 1. IRQ_0 to IRQ_7 have interrupt vector numbers 4 to 11. They are prioritized in order from IRQ_7 (low) to IRQ_0 (high). For details, see table 4.2.

Interrupts IRQ_0 to IRQ_7 do not depend on whether pins $\overline{\text{IRQ}}_0$ to $\overline{\text{IRQ}}_7$ are input or output pins. When using external interrupts IRQ_0 to IRQ_7, clear the corresponding DDR bits to 0 to set these pins to the input state, and do not use these pins as input or output pins for the timers, serial communication interface, or A/D converter.

4.3.4 Internal Interrupts

Thirty-three internal interrupts can be requested by the on-chip supporting modules. Each interrupt source has its own vector number, so the interrupt-handling routine does not have to determine which interrupt has occurred. All internal interrupts are masked when the I bit in the CCR is set to 1. When one of these interrupts is accepted, the I bit is set to 1 to mask further interrupts (except $\overline{\text{NMI}}$). The vector numbers are 12 to 44. For the priority order, see table 4.2.

4.3.5 Interrupt Handling

Interrupts are controlled by an interrupt controller that arbitrates between simultaneous interrupt requests, commands the CPU to start the hardware interrupt exception-handling sequence, and furnishes the necessary vector number. Figure 4.3 shows a block diagram of the interrupt controller.

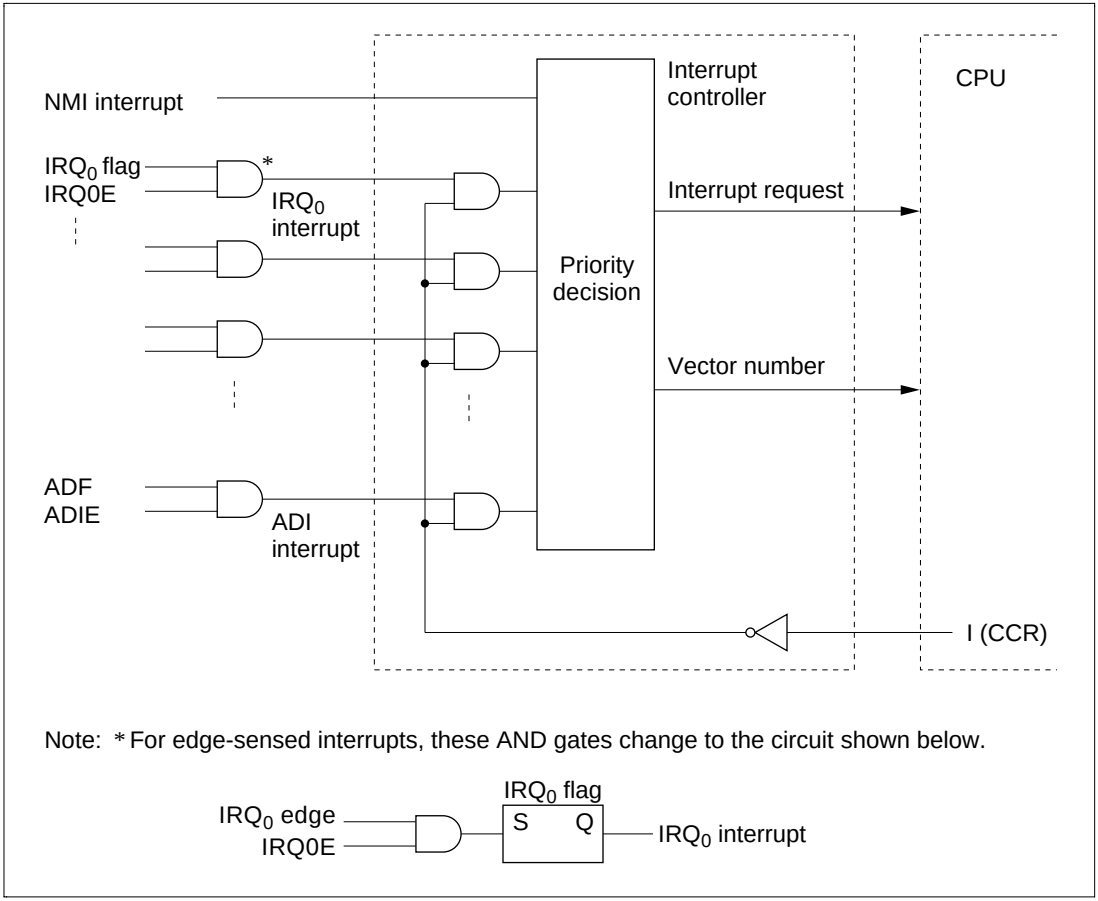


Figure 4.3 Block Diagram of Interrupt Controller

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

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

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

Figure 4.4 is a flowchart of the interrupt (and reset) operations. Figure 4.6 shows the interrupt timing sequence for the case in which the software interrupt-handling routine is in on-chip ROM and the stack is in on-chip RAM.

1. An interrupt request is sent to the interrupt controller when an NMI interrupt occurs, and when an interrupt occurs on an IRQ input line or in an on-chip supporting module provided the enable bit of that interrupt is set to 1.

2. The interrupt controller checks the I bit in the CCR and accepts the interrupt request if the I bit is cleared to 0. If the I bit is set to 1 only NMI requests are accepted; other interrupt requests remain pending.

3. Among all accepted interrupt requests, the interrupt controller selects the request with the highest priority and passes it to the CPU. Other interrupt requests remain pending.

4. When it receives the interrupt request, the CPU waits until completion of the current instruction or hardware exception-handling sequence, then starts the hardware exception-handling sequence for the interrupt and latches the interrupt vector number.

5. In the hardware exception-handling sequence, the CPU first pushes the PC and CCR onto the stack. See figure 4.5. The stacked PC indicates the address of the first instruction that will be executed on return from the software interrupt-handling routine.

6. Next the I bit in the CCR is set to 1, masking all further interrupts except NMI.

7. The vector address corresponding to the vector number is generated, the vector table entry at this vector address is loaded into the program counter, and execution branches to the software interrupt-handling routine at the address indicated by that entry.

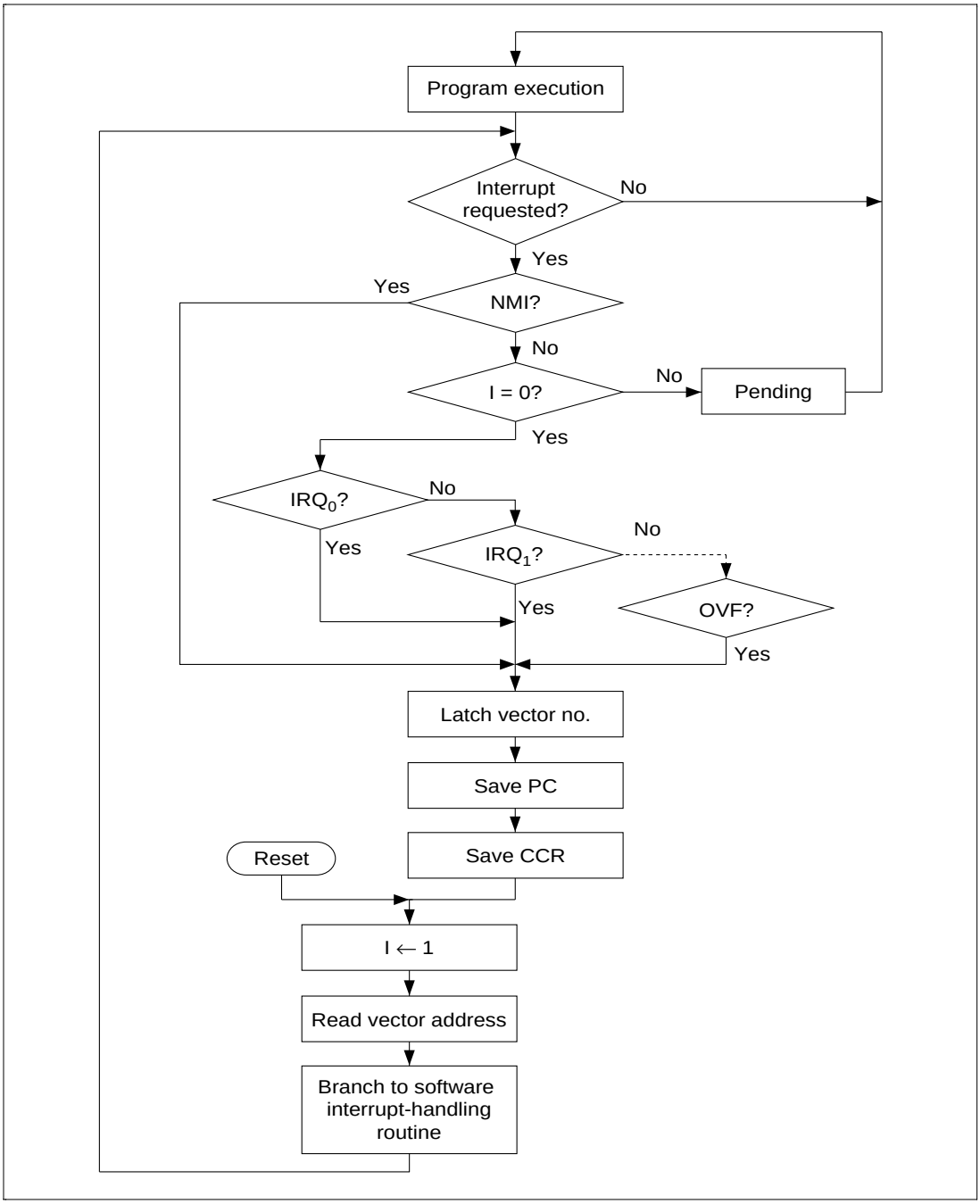


Figure 4.4 Hardware Interrupt-Handling Sequence

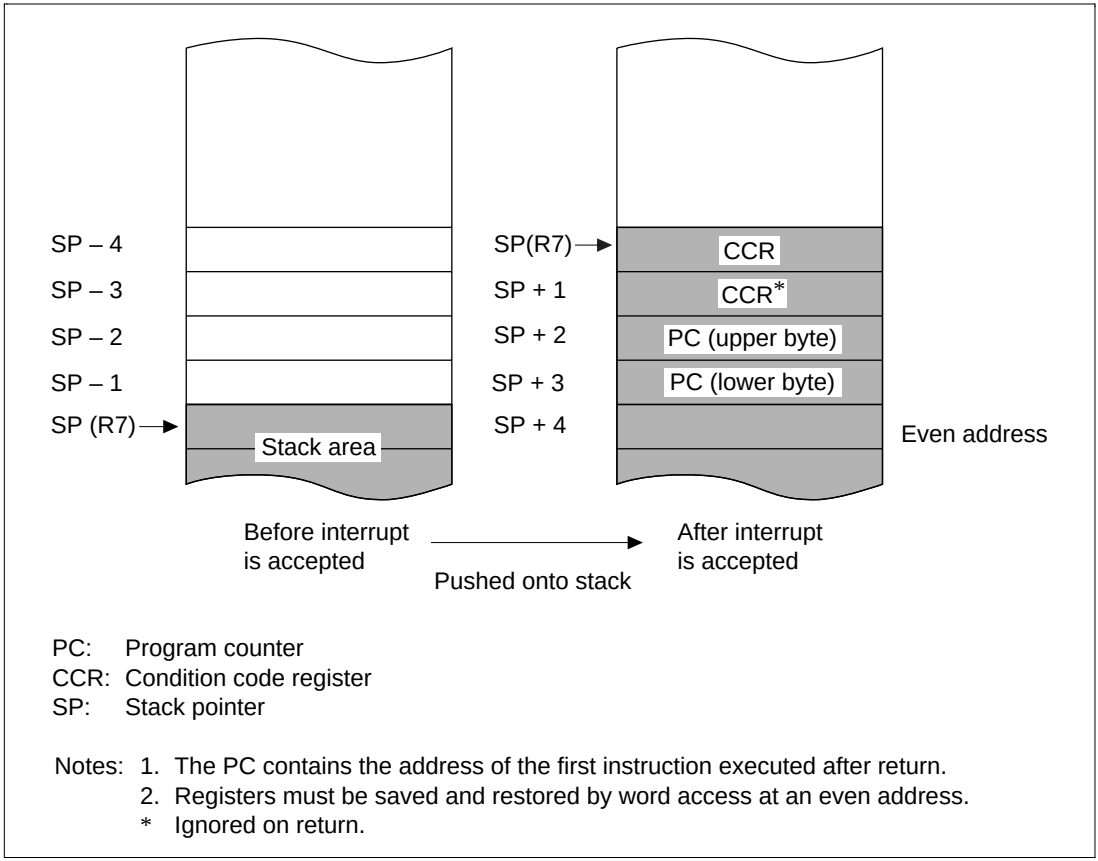
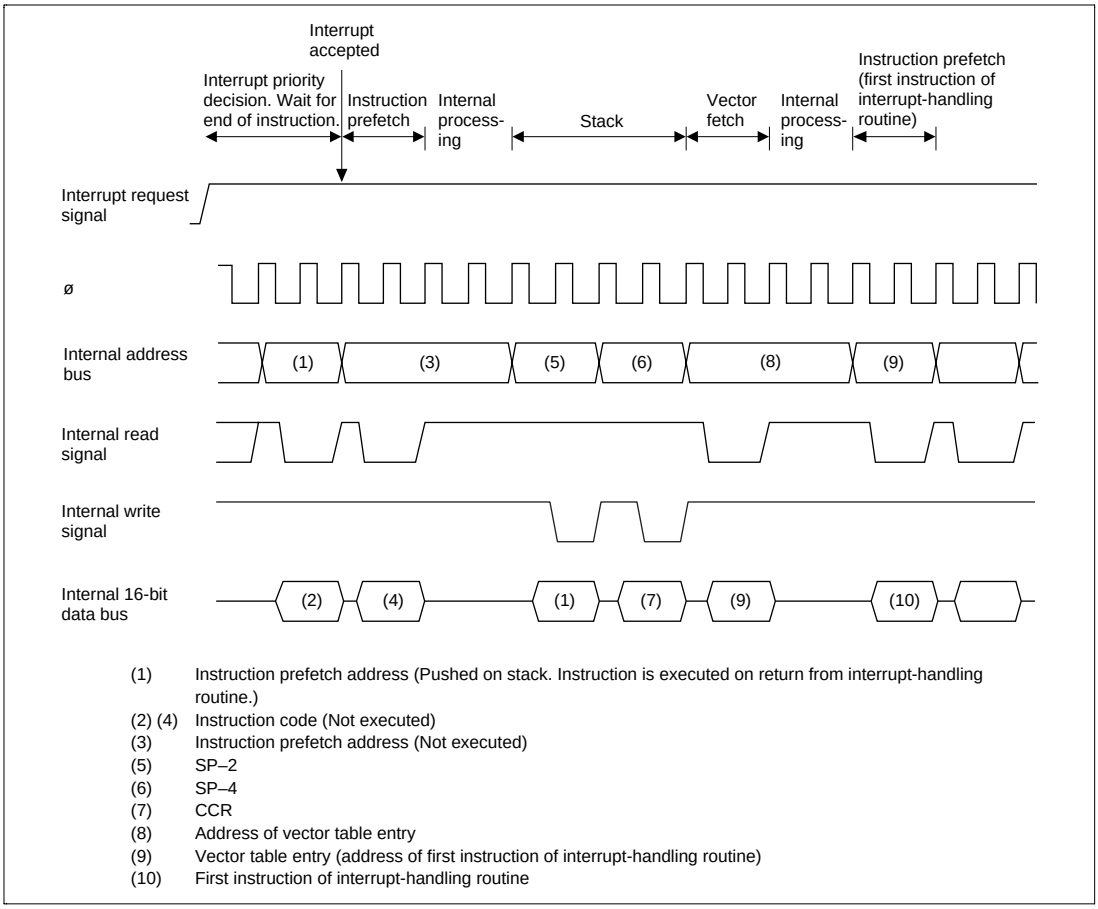


Figure 4.5 Usage of Stack in Interrupt Handling

Although the CCR consists of only one byte, it is treated as word data when pushed on the stack. In the hardware interrupt exception-handling sequence, two identical CCR bytes are pushed onto the stack to make a complete word. When popped from the stack by an RTE instruction, the CCR is loaded from the byte stored at the even address. The byte stored at the odd address is ignored.



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

Figure 4.6 Timing of Interrupt Sequence

4.3.6 Interrupt Response Time

Table 4.4 indicates the number of states that elapse from an interrupt request signal until the first instruction of the software interrupt-handling routine is executed. Since on-chip memory is accessed 16 bits at a time, very fast interrupt service can be obtained by placing interrupt-handling routines in on-chip ROM and the stack in on-chip RAM.

Table 4.4 Number of States before Interrupt Service

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

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

4.3.7 Precaution

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

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

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

Figure 4.7 shows an example in which the OCIAE bit is cleared to 0.

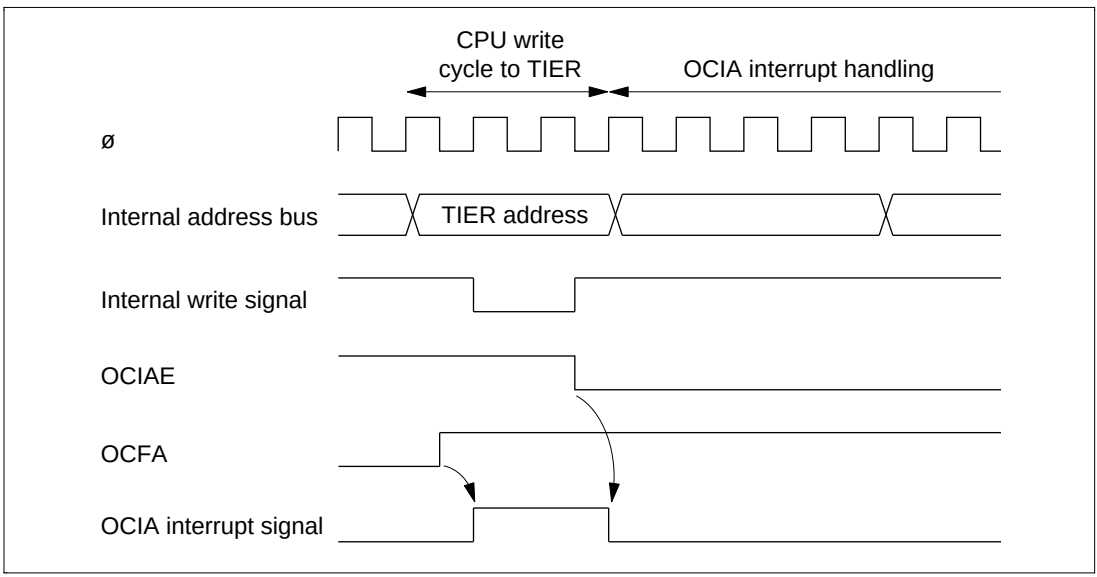


Figure 4.7 Contention between Interrupt and Disabling Instruction

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

4.4 Note on Stack Handling

In word access, the least significant bit of the address is always assumed to be 0. The stack is always accessed by word access. Care should be taken to keep an even value in the stack pointer (general register R7). Use the PUSH 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 4.8 shows an example of damage caused when the stack pointer contains an odd address.

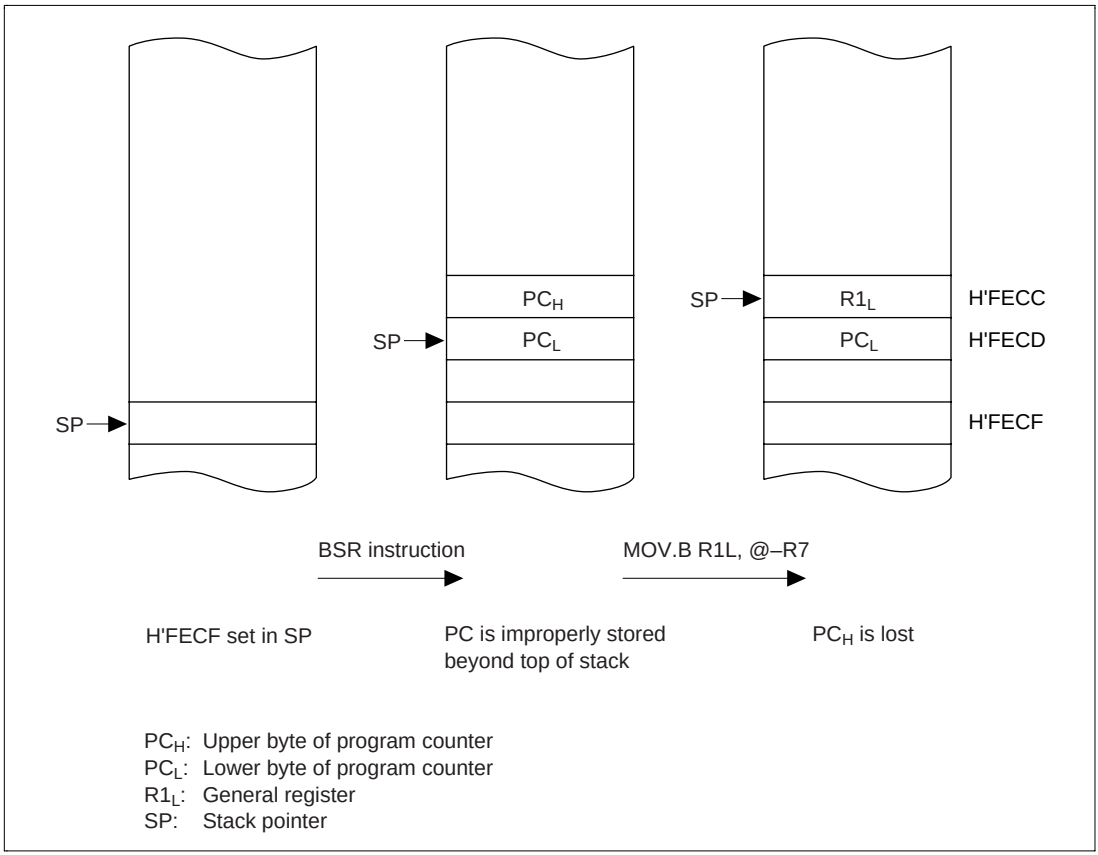


Figure 4.8 Example of Damage Caused by Setting an Odd Address in R7

Section 5 Data Transfer Unit

5.1 Overview

The H8/3318 Series has an on-chip data transfer unit (DTU) with four direct memory access (DMA) channels, and an on-chip 8-bit parallel buffer interface (PBI). When combined, these two functions enable up to 256 bytes of on-chip RAM to be easily accessed (read or write) from outside the chip.

In I/O transfer, the DTU can perform data transfer automatically between on-chip RAM and registers of the serial communication interface (SCI), 16-bit free-running timer (FRT), programmable timing pattern controller (TPC), and A/D converter (ADC), in response to an interrupt request.

In PBI transfer, the DTU can implement a large dual-port RAM (DPRAM) by enabling up to 256 bytes of on-chip RAM to be accessed from an external master CPU. In its DPRAM mode, the PBI has a buffer that can be queried to read one byte from a specified address. While accepting buffer queries, the PBI can also operate in bound buffer mode, which provides sequential read/write access starting from a specified address, or direct word mode, which does not use the DTU and on-chip RAM.

The PBI can operate in handshake mode as well as DPRAM mode.

5.1.1 Features

Features of I/O transfer, DPRAM mode, and handshake mode are listed below.

I/O Transfer (Figure 5.1)

- The three DTU channels with I/O transfer capability (channels A, B, and C) can perform DMA independently.
- The selectable types of transfer are: RAM to SCI (TDR), SCI (RDR) to RAM; RAM to FRT (OCRA, OCRB); RAM to TPC (NDR); and ADC (ADDRA) to RAM.
- The number of bytes transferred can be controlled by setting a starting address and boundary. Each channel can generate an independent interrupt when the designated number of bytes have been transferred.
- A maximum 256-byte RAM area can be used (maximum 128 bytes per channel).
- Data in a specified area can be transferred repeatedly by designating repeat mode.
- Channel B can be set to operate in ring buffer mode (FIFO mode).

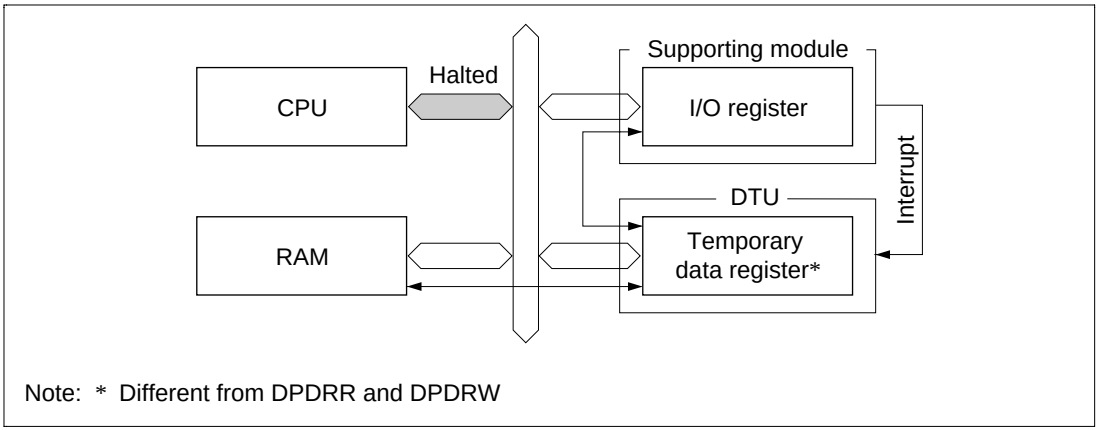


Figure 5.1 Concept of I/O Transfer

DPRAM Mode

- A master CPU can access on-chip RAM randomly via a buffer, by using DTU channel R as read DMA (buffer query).
- The master CPU can access on-chip RAM sequentially via a pair of buffers, using DTU channel A for read DMA and channel B for write DMA (bound buffer mode). Without using the DTU, the master CPU can also access the buffers in each channel as a dual-port RAM (direct word mode).
- A maximum 256-byte RAM area can be used (maximum 128 bytes each in channel A and B).
- Internal CPU interrupts can be generated when the designated number of bytes have been transferred (master read end, master write end).
- In single-chip mode, master CPU interrupt requests can be issued from the $\overline{\text{RDY}}$ pin, and wait requests from the $\overline{\text{WRQ}}$ pin.

  In the expanded modes, one of these two types of requests can be generated: master CPU interrupt requests can be issued from the $\overline{\text{RDY}}$ pin, or wait requests can be issued from the $\overline{\text{WRQ}}$ pin.

Buffer query operation (Figure 5.2)

- A 1-byte address/data register is available. By writing an on-chip RAM address, the master CPU can read any byte in a 256-byte on-chip RAM area.

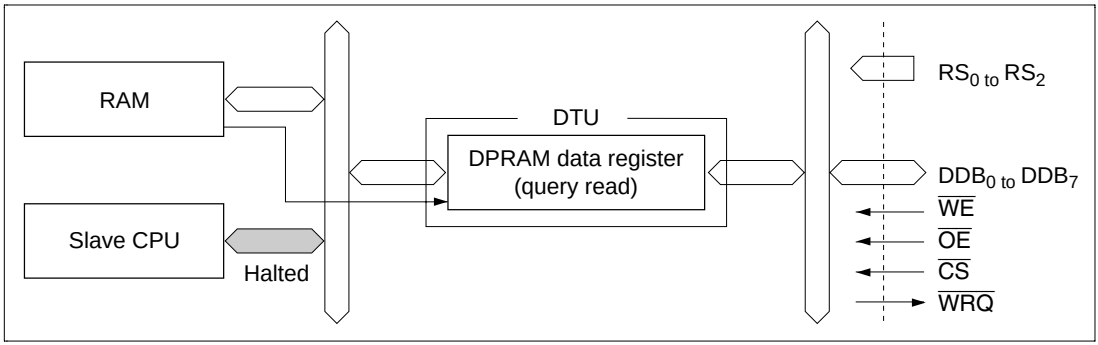


Figure 5.2 Concept of Buffer Queries

Bound buffer mode (Figure 5.3)

- Two pairs of 8-bit data registers are available, one for read and one for write. The master CPU can use these as buffers to transfer data to or from a 256-byte on-chip RAM area with no wait time.
- The number of bytes to be transferred can be designated by setting a starting address and boundary. Each channel can generate an internal CPU interrupt when the designated number of bytes have been transferred.
- In a read operation by the master CPU, data is transferred from addresses specified by the internal CPU.
- In a write operation by the master CPU, data is transferred to addresses specified by the master CPU.

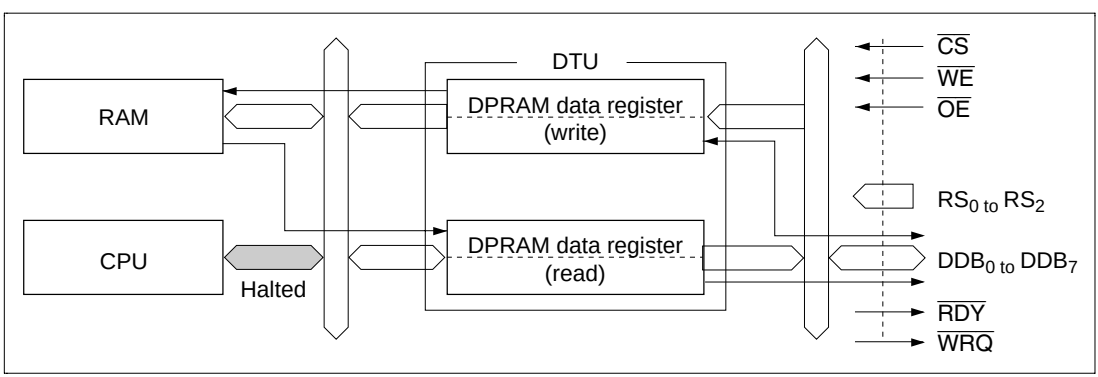


Figure 5.3 Concept of Bound Buffer Mode

Direct word mode (Figure 5.4)

- Two pairs of 8-bit data registers are available, one for read and one for write. These can be used as a dual-port RAM to exchange data with the master CPU. Channels not using bound buffer mode are automatically placed in direct word mode.

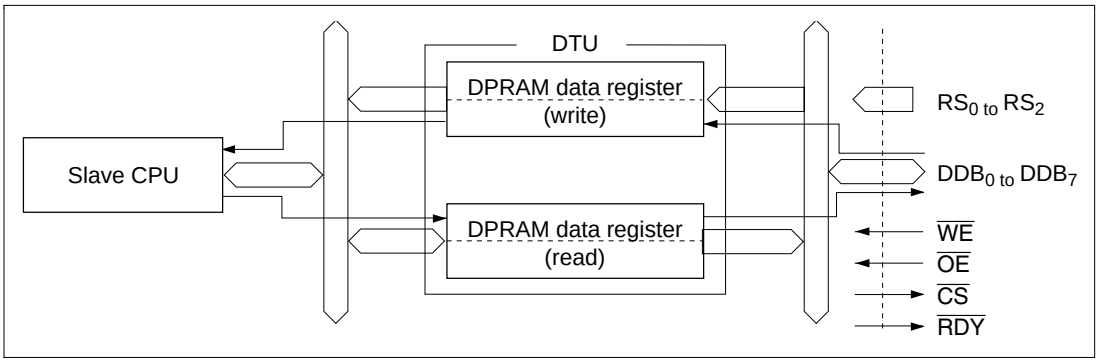


Figure 5.4 Concept of Direct Word Mode

Handshake Mode (Figure 5.5)

- Handshaking is carried out using $\overline{OE}$ and $\overline{WE}$ input signals and $\overline{RDY}$ and $\overline{WRQ}$ output signals.
- An internal CPU interrupt can be generated at the rise of the $\overline{OE}$ input (output data processing completed) and rise of the $\overline{WE}$ input (input data valid).
- The $\overline{RDY}$ and $\overline{WRQ}$ output signals can be used to send interrupt requests and data input/output requests to the master CPU.

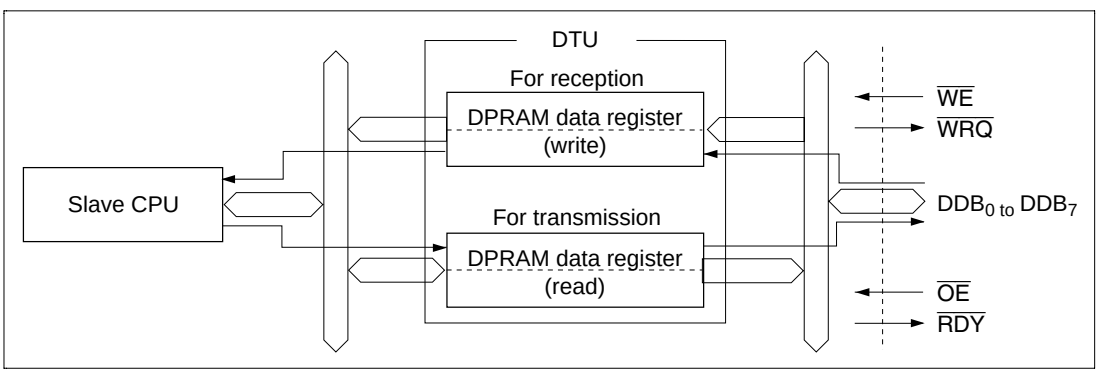


Figure 5.5 Concept of Handshake Mode

5.1.2 Block Diagram

Figure 5.6 shows a block diagram of the DTU and PBI.

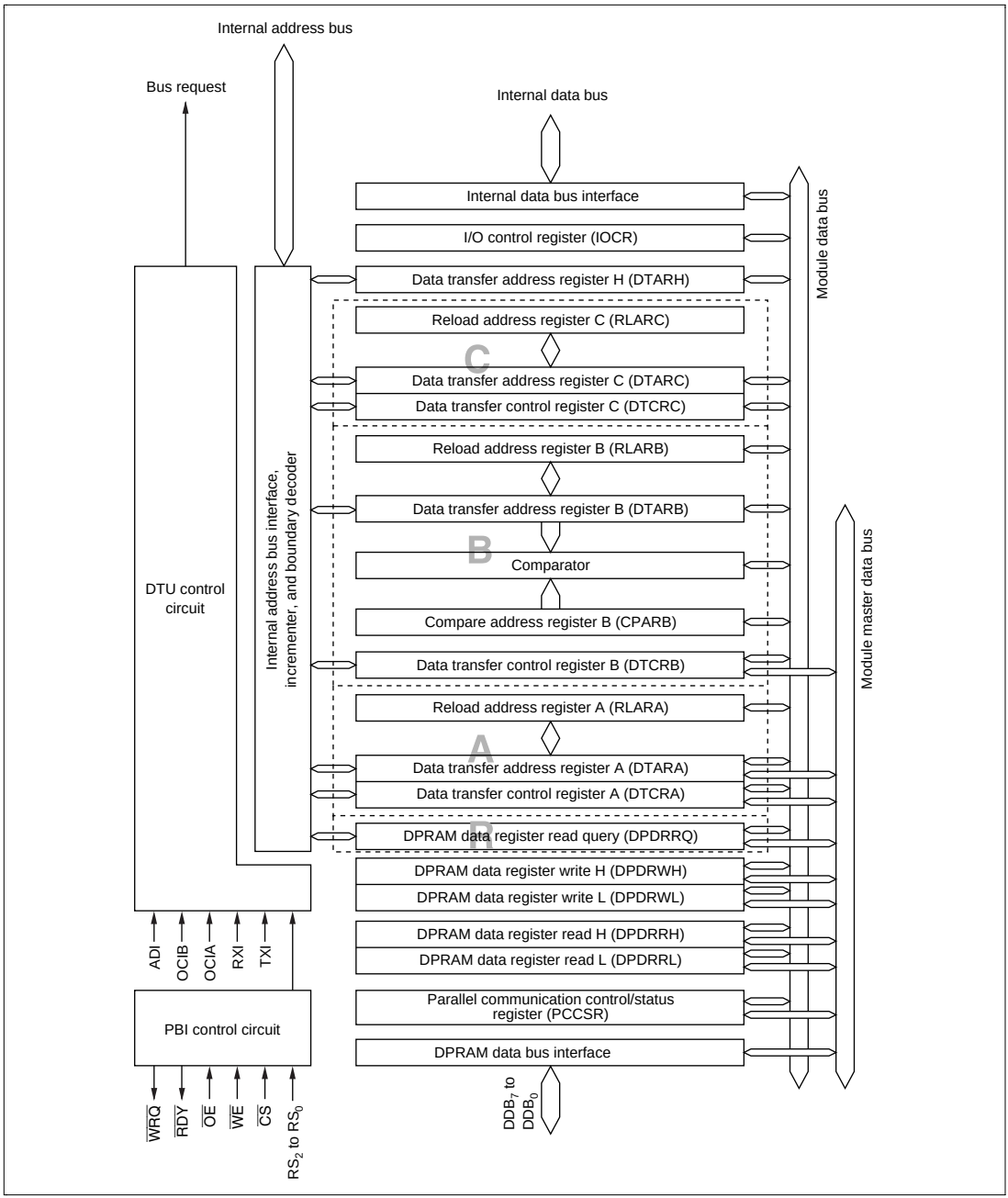


Figure 5.6 DTU and PBI Block Diagram

5.1.3 Input and Output Pins

Table 5.1 lists the input and output pins.
$\overline{CS}$, $\overline{OE}$, $\overline{WE}$, $\overline{RDY}$, and DDB are active only in single-chip mode. In expanded mode, $\overline{XCS}$, $\overline{XOE}$, $\overline{XWE}$, $\overline{XRDY}$, XDDB become active instead.

Table 5.1 Input and Output Pins

Name	Abbreviation*[1]	I/O	Function (DPRAM Mode)
Chip select	$\overline{CS}$, $\overline{XCS}$	Input	Selects the PBI
Register select	RS_2 to RS_0	Input	Used by the master CPU to select a PBI register*[2]
Output enable	$\overline{OE}$, $\overline{XOE}$	Input	Used by the master CPU to read a PBI register
Write enable	$\overline{WE}$, $\overline{XWE}$	Input	Used by the master CPU to write to a PBI register
Ready	$\overline{RDY}$, $\overline{XRDY}$	Output	Sends an interrupt request to the master CPU
Wait request	$\overline{WRQ}$	Output	Sends a wait request to the master CPU
DPRAM data bus	DDB_7 to DDB_0 $XDDB_7$ to $XDDB_0$	Input/ output	8-bit data bus providing a parallel interface between the master CPU and PBI

Notes: 1. Unless specifically noted, $\overline{XCS}$, $\overline{XOE}$, $\overline{XWE}$, $\overline{XRDY}$, and XDDB will be referred to as $\overline{CS}$, $\overline{OE}$, $\overline{WE}$, $\overline{RDY}$, and DDB, respectively.
2. The registers selected by the register select signals are listed in table 5.2.

5.1.4 Register Configuration

Table 5.2 lists the DTU and PBI registers

Table 5.2 DTU and PBI Registers

Name	Abbre-viation	R/W	Initial Value	Address		Description
				Internal	RS_2 to RS_0	
Parallel communication control/status register (PCCSR)	PCCSR	R/W	H'04	H'FFF0	000	P95
I/O control register	IOCR	R/W	H'03	H'FFF1	—	P84
DPRAM data register read query	DPDRRQ	—	Undetermined	—	001	P95
Data transfer address register H	DTARH	R/W	Undetermined	H'FFF5	—	P89
Data transfer control register A	DTCRA	*	H'00	H'FFF6	010	P86
Data transfer address register A	DTARA	*	Undetermined	H'FFF7	011	P89
Reload address register A	RLARA	R	Undetermined	H'FFF2	—	P90
Data transfer control register B	DTCRB	*	H'00	H'FFF8	010	P86
Data transfer address register B	DTARB	*	Undetermined	H'FFF9	011	P89
Reload address register B	RLARB	R	Undetermined	H'FFF3	—	P90
Compare address register B	CPARB	R/W	Undetermined	H'FFF4	—	P90
Data transfer control register C	DTCRC	R/W	H'00	H'FFFA	—	P86
Data transfer address register C	DTARC	R/W	Undetermined	H'FFFB	—	P89
Reload address register C	RLARC	—	Undetermined	—	—	P90
DPRAM data register write H	DPDRWH	*	Undetermined	H'FFFC	100	P93
DPRAM data register write L	DPDRWL	*	Undetermined	H'FFFD	101	P93
DPRAM data register read H	DPDRRH	*	Undetermined	H'FFFE	110	P94
DPRAM data register read L	DPDRRL	*	Undetermined	H'FFFF	111	P94
Serial/timer control register	STCR	R/W	H'1C	H'FFC3	—	P91
System control register	SYSCR	R/W	H'09	H'FFC4	—	P99

Note: * Register read/write specifications are given in table 5.3, Transfer Modes and Register Configuration.

Register accessibility differs depending on the transfer mode. Table 5.3 lists the register access conditions for each transfer mode, and table 5.4 shows how each transfer mode is selected.

Table 5.3 Transfer Modes and Register Configuration

		R/W									
		DPRAM Mode									
		I/O Transfer	PBI Transfer					Hand-shake Mode*[2]		Address	
			Buffer Query*[1]	Bound Buffer Mode		Direct Word Mode					
Chan-nel	Abbre-viation	Internal CPU	Master CPU	Master CPU	Internal CPU	Master CPU	Internal CPU	Internal CPU	Initial Value	Internal	RS_2 to RS_0
	PCCSR	×	R/W	R/W	R/W	R/W	R/W	R/W	H'04	H'FFF0	000
	IOCR	R/W	—	—	R/W	—	R/W	R/W	H'03	H'FFF1	—
R	DPDRRQ	—	R/W	×	—	×	—	—	*	—	001
	DTARH	R/W	—	—	R/W	—	×	×	*	H'FFF5	—
A	DTCRA	R/W | ×*[3]	—	— | R*[4]	× | R/W*[5]	—	×	×	H'00	H'FFF6	010
	DTARA	R/W | ×*[3]	—	— | R*[4]	× | R/W*[5]	—	×	×	*	H'FFF7	011
	RLARA	R | Δ	—	—	Δ | R*[5]	—	Δ	Δ	*	H'FFF2	—
B	DTCRB	R/W | ×*[3]	— | ×*[4]	— | W*[4]	× | R*[5]	— | ×*[4]		×	×	H'00	H'FFF8	010
	DTARB	R/W | ×*[3]	— | ×*[4]	— | W*[4]	× | R*[5]	— | ×*[4]		×	×	*	H'FFF9	011
	RLARB	R | Δ	—	—	Δ | R*[5]	—	Δ	Δ	*	H'FFF3	—
	CPARB	R/W | ×*[3]	—	—	× | R/W*[5]	—	×	×	*	H'FFF4	—
C	DTCRC	R/W	—	—	×	—	×	×	H'00	H'FFFA	—
	DTARC	R/W	—	—	×	—	×	×	*	H'FFFB	—
	RLARC	—	—	—		—	—	—	*	—	—
	DPDRWH	×	×	W	—	W	R	Δ	*	H'FFFC	100
	DPDRWL	×	×	W	—	W	R	R	*	H'FFFD	101
	DPDRRH	×	×	R		R	W	Δ	*	H'FFFE	110
	DPDRRL	×	×	R		R	W	W	*	H'FFFF	111

Legend

×: Must not be accessed (write access will affect operations of other functions).

Δ: Can be accessed, but has no effect on operation.

—: Cannot be accessed.

*: Undetermined

Notes:
1. Buffer queries can be made in parallel with bound buffer mode and direct word mode. DTARH and IOCR are used in all these modes, so care is required in modifying their contents.
2. In handshake mode, the master CPU can write to DPDRWL. The internal CPU can output the DPDRRL contents to the master CPU.
3. Must not be accessed if used in bound buffer mode.
4. Cannot be accessed if used for I/O transfer.
5. Must not be accessed if used for I/O transfer.

Table 5.4 Transfer Mode Selection

I/O transfer mode		Basically, I/O transfer mode can be started for channels A, B, and C at any time by setting DTE to 1. However, channels A and B may be assigned for transfer in bound buffer mode, depending on the IOCR setting. (See table 5.5 and section 5.3.3.)
DPRAM mode		
	Buffer queries	Channel R is used exclusively. The buffer query function operates when DPME = 1 in SYSCR and HSCE = 0 in IOCR. (See table 5.5 and section 5.3.4.)
	Bound buffer mode	Channel A is used for reads to the master. This mode can be used when DPME = 1 (SYSCR), HSCE = 0 (IOCR), and DPEA = 1 (IOCR). Channel B is used for writes from the master. This mode can be used when DPME = 1 (SYSCR), HSCE = 0 (IOCR), and DPEB = 1 (IOCR). (See table 5.5 and section 5.3.5.)
	Direct word mode	Channel A is used for reads to the master. This mode can be used when DPME = 1 (SYSCR), HSCE = 0 (IOCR), and DPEA = 0 (IOCR). Channel B is used for writes from the master. This mode can be used when DPME = 1 (SYSCR), HSCE = 0 (IOCR), and DPEB = 0 (IOCR). (See table 5.5 and section 5.3.6.)
Handshake mode		This mode can be used when DPME = 1 in SYSCR and HSCE = 1 in IOCR. (See table 5.5 and section 5.3.7.)

5.2 Register Descriptions

The registers of the DTU and PBI are described below.

Abbreviations shown in boxes after the register name indicate the modes in which the register can be used. The meaning of these abbreviations is as follows:

I/O: I/O transfer
Q: Buffer queries
BB: Bound buffer mode
DI: Direct word mode
H/S: Handshake mode

5.2.1 I/O Control Register (IOCR)

				I/O	Q	BB	DI	H/S
Bit	7	6	5	4	3	2	1	0
	HSCE	DPEA	DPEB	RPEA	RPEB	RPEC	—	—
Initial value	0	0	0	0	0	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	—	—

IOCR selects DTU and PBI operating modes, and controls the DTU.

Bit 7—Parallel Handshake Enable (HSCE)

Bits 6 and 5—DPRAM Enable A and B (DPEA, DPEB)

The DTU and PBI operating modes are selected by the DPME bit in SYSCR, and the HSCE, DPEA, and DPEB bits in IOCR. Table 5.5 lists the mode selections. Table 5.6 lists the corresponding pin functions.

Table 5.5 DTU/PBI Operating Mode Settings

DPME (SYSCR)	Bit 7 HSCE	Bit 6 DPEA	Bit 5 DPEB	Hand-shake Mode	Buffer Query (DTU Channel R)	Bound Buffer Mode (DTU Channel A)	Bound Buffer Mode (DTU Channel B)	Direct Word Mode (DPDRRH/L)	Direct Word Mode (DPDRWH/L)
1	1	—	—	O	×	I/O transfer*	I/O transfer*	Handshake	Handshake
	0	0	0	×	O (read)	I/O transfer*	I/O transfer*	O (read)	O (write)
		1	0	×	O (read)	O (read)	I/O transfer*	Bound buffer	O (write)
		0	1	×	O (read)	I/O transfer*	O (write)	O (read)	Bound buffer
		1	1	×	O (read)	O (read)	O (write)	Bound buffer	Bound buffer
0	—	—	—	×	×	I/O transfer*	I/O transfer*	×	×

O: Can be used

×: Cannot be used

—: Undetermined

*: Used in I/O transfer mode

Note: For I/O transfer, set the DTE bit to 1 in DTCRA, DTCRB, or DTCRC.
Handshake mode operation is supported only in single-chip mode. Do not set bit HSCE to 1 in expanded modes.

Table 5.6 Pin Functions

DPME (SYSCR)	Bit 7 HSCE	Mode Name	$\overline{CS}/\overline{XCS}$, RS_2 to RS_0	$\overline{OE}/\overline{XOE}$, $\overline{WE}/\overline{XWE}$	$\overline{RDY}/\overline{XRDY}$	$\overline{WRQ}$	DDB_7 to DDB_0/ $XDDB_7$ to $XDDB_0$
1	1	Hand-shake mode	Port function	Control input	Control output	Control output or port function, depending on EWRQ (PCCSR) bit	Data input/output
	0	DPRAM mode	Control input				
0	—	—	Port function	Port function	Port function	Port function	Port function or data bus

Note: "Port function" means that the port functions, supporting-module functions, and extended functions multiplexed with the DPRAM pins are available.

Bits 4, 3, and 2—Repeat Enable A, B, and C (RPEA, RPEB, RPEC): These bits are valid only in I/O transfers. They select repeat mode or normal mode for DTU channels A, B, and C.

In normal mode, the DTE bit in DTCRA, DTCRB, or DTCRC is cleared to 0 when the transfer reaches the boundary.

In repeat mode, when the transfer reaches the boundary, the DTE bit in DTCRA, DTCRB, or DTCRC is not cleared to 0, and the data in the area defined by DTARA, DTARB, or DTARC and the boundary is transferred repeatedly. Channels A and B also have reload address registers, and channel B can furthermore operate in ring buffer mode.

For usage of the boundary, normal mode, and repeat mode, see section 5.3, Operation.

Bits 4, 3, or 2 RPEA, B, C	Description	
0	Transfer in normal mode	(Initial value)
1	Transfer in repeat mode	

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

5.2.2 Data Transfer Control Registers A, B, and C (DTCRA, DTCRB, DTCRC)

I/O	BB

Bit	7	6	5	4	3	2	1	0
	DTE	DTIE	BUD2	BUD1	BUD0	SOS2	SOS1	SOS0
Initial value	0	0	0	0	0	0	0	0

Mode	Register									
I/O transfer	DTCRA DTCRB DTCRC	Internal CPU	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PBI transfer in DPRAM bound buffer mode	DTCRA	Internal CPU	Read/Write	0	(R/W)	R/W	R/W	R/W	(R/W)	(R/W)	(R/W)
		Master CPU	Read/Write	0	R	R	R	R	R	R	R
	DTCRB	Internal CPU	Read/Write	0	0	R	R	R	0	0	0
		Master CPU	Read/Write	—	—	W	W	W	—	—	—

—: Not used. Cannot be modified.

0: Always reads 0.

(R/W): Can be written, but has no effect on operation.

These registers are used in I/O transfers, and in PBI transfers in DPRAM bound buffer mode.

In I/O transfers, DTCRA, DTCRB, and DTCRC serve as control registers for DTU channels A, B, and C, respectively.

In PBI transfers in DPRAM bound buffer mode, DTCRA controls read access by the master CPU, and DTCRB controls write access by the master CPU.

Bit 7—Data Transfer Enable (DTE): Used in I/O transfers. Not used in PBI transfers.

When DTE is set to 1, the channel begins waiting for a transfer request. The DMA transfer is activated by the interrupt request signal selected by the source select bits (SOS2, SOS1, SOS0).

If a boundary overflow or the internal CPU clears the DTE bit to 0, the transfer is suspended. If the DTIE bit is set to 1, an interrupt is also requested. If the internal CPU sets the DTE bit to 1 again, the transfer resumes from the state in which it was suspended.

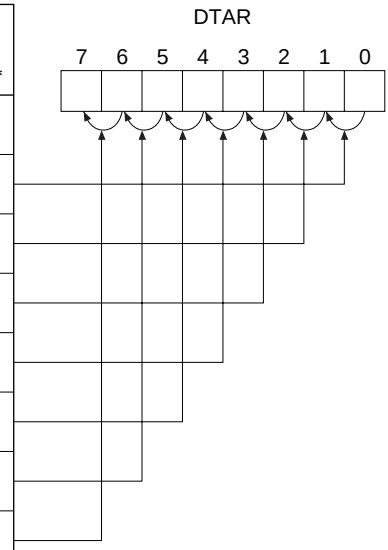
Bit 7 DTE	Description	
0	Indicates that I/O transfer is halted	(Initial value)
	[Clearing conditions]	
	1. 0 is written in DTE	
	2. The transfer terminates at the boundary in normal mode	
1	Indicates that I/O transfer is in progress	
	[Setting condition]	
	Read DTCR while DTE = 0, then write 1 in DTE	

Bit 6—Data Transfer Interrupt Enable (DTIE): Used in I/O transfers. Not used in PBI transfers. This bit enables or disables the interrupt generated when the DTE bit is cleared to 0.

Bit 6 DTIE	Description	
0	Disables the interrupt requested when the DTE bit is cleared to 0 (DTI)	(Initial value)
1	Enables the interrupt requested when the DTE bit is cleared to 0 (DTI)	

Bits 5, 4, and 3—Boundary 2, 1, and 0 (BUD2, BUD1, BUD0): These bits set a carry-control boundary in the data transfer address register (DTAR), which is the register that gives the lower byte of an on-chip RAM address. DTAR is an 8-bit counter that increments each time one byte or one word is transferred. The incrementing is held within the selected boundary. When a carry occurs at the boundary, causing a DTAR overflow, the bits above the boundary retain their existing values, while the bits below the boundary are reset to their initial value. The initial value may be 0, or a value stored in a reload register, depending on the channel and operating mode.

BUD2	BUD1	BUD0	DTAR Overflow Timing	Maximum Bytes Transferred*
0	0	0	End of each byte transfer	1
0	0	1	Carry from bit 0 to bit 1 in DTAR	2
0	1	0	Carry from bit 1 to bit 2 in DTAR	4
0	1	1	Carry from bit 2 to bit 3 in DTAR	8
1	0	0	Carry from bit 3 to bit 4 in DTAR	16
1	0	1	Carry from bit 4 to bit 5 in DTAR	32
1	1	0	Carry from bit 5 to bit 6 in DTAR	64
1	1	1	Carry from bit 6 to bit 7 in DTAR	128



Note: * Number of bytes transferred when bits below the boundary are initially cleared to 0.

Figure 5.7 Bit Settings and Boundary Positions

Bits 2, 1, and 0—Source Select 2, 1, and 0 (SOS2, SOS1, SOS0): Used in I/O transfers, but not in PBI transfers. These bits select the interrupt source that activates an I/O transfer by the DTU. When DTE is set to 1, the selected interrupt request signal is regarded as an I/O transfer request. When DTE is 0, the selected interrupt request signal is sent to the interrupt controller.

Table 5.7 lists the I/O transfer activation sources and the transfers they request. Some transfers clear the activation source while others do not. By using a combination of these types, it is possible to perform two consecutive transfer operations with a single activation source. In this case, the I/O transfers are performed in alphabetical channel order (channel A, then B, then C).

If channel A or channel B executes transfer operation number 9 or 10 in table 5.7, the activation source is not cleared, and the channel is stopped.

A stopped channel is automatically cleared by the completion of transfer operation 5 or 7 for stoppage due to transfer operation 9, or by completion of operation 4 or 6 for operation 10. A stopped channel can also be cleared by a dummy write to its data transfer control register (DTCR).

Table 5.7 Selection of Activation Sources and Resulting Transfers

No.	Interrupt Source	Module	Transfer	Clearing of Source	DTU (Channel A)			DTU (Channel B)			DTU (Channel C)		
					SOS2	SOS1	SOS0	SOS2	SOS1	SOS0	SOS2	SOS1	SOS0
1	RXI0	SCI0	RDR → RAM (byte)	Yes	0	0	1	0	0	1	0	0	1
2	TXI0	SCI0	RAM → TDR (byte)	Yes	0	1	0	0	1	0	0	1	0
3	ADI	ADC	ADDRA → RAM (word)	Yes	0	1	1	0	1	1	0	1	1
4	OCIB1	TPC	RAM → NDRB (word)	Yes							1	0	0
5	OCIA1	TPC	RAM → NDRB (word)	Yes				1	0	0	1	0	1
6	OCIB1	TPC	RAM → NDRA (byte)	Yes							1	1	0
7	OCIA1	TPC	RAM → NDRA (byte)	Yes				1	0	1	1	1	1
8	OCIA1	FRT1	RAM → OCRA (word)	Yes	1	0	0	1	1	0			
9	OCIA1	FRT1	RAM → OCRA (word)	No	1	0	1	1	1	1			
10	OCIB1	FRT1	RAM → OCRB (word)	No	1	1	0						

Note: When SOS2 = SOS1 = SOS0 = 0, an activation source is not selected.

5.2.3 Data Transfer Address Register H (DTARH)

I/O	Q	BB

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

DTARH is used whenever the DTU function is used. It is used both for I/O transfers and for PBI transfers in DPRAM mode, including buffer queries and transfers in bound buffer mode.

DTARH is the register that specifies the upper 8 bits of the on-chip RAM address. DTARH can be paired with DTARA, DTARB, or DTARC to generate a 16-bit address.

5.2.4 Data Transfer Address Registers A, B, and C (DTARA, DTARB, DTARC)

	I/O				Q		BB	
Bit	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—

Mode	Register									
I/O transfer	DTARA DTARB DTARC	Internal CPU	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PBI transfer in DPRAM bound buffer mode	DTARA	Internal CPU	Read/Write	R/W	R/W	R/W	R/W	R/W	(R/W)	(R/W)	(R/W)
		Master CPU	Read/Write	R	R	R	R	R	R	R	R
	DTARB	Internal CPU	Read/Write	R	R	R	R	R	R	R	R
		Master CPU	Read/Write	W	W	W	W	W	W	W	W

The DTAR registers are used when the DTU function is used for I/O transfers, and for PBI transfers in DPRAM bound buffer mode. DTARA, DTARB, and DTARC pair with DTARH to generate 16-bit on-chip RAM addresses which indicate DTU transfer addresses. A boundary can be set in the DTAR registers by the BUD bits, so that the DTAR registers increment only in the range up to the boundary.

In an I/O transfer, these registers are used as the lower 8-bit address registers in each channel.

In a PBI transfer in DPRAM bound buffer mode, DTARA is the lower 8-bit address register in read access by the master CPU, and DTARB is the lower 8-bit address register in write access by the master CPU.

5.2.5 Reload Address Registers A, B, and C (RLARA, RLARB, RLARC)

							I/O	BB
Bit	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The RLAR registers are used when the DTU function is used for I/O transfers, and for PBI transfers in DPRAM bound buffer mode.

The RLAR registers store data for initializing the DTAR registers when a boundary overflow occurs. Writes to the RLAR registers are performed automatically by master CPU or internal CPU writes. The DTAR value is then successively incremented, and if a boundary overflow occurs, the DTAR register will be initialized to the value in the RLAR register.

In ring buffer mode, RLARB functions as an auxiliary ring buffer pointer. DTARB is the main pointer, indicating the DTU transfer address. If a series of transfers is suspended and the stored data becomes invalid, DTARB can be initialized to the value in RLARB (loading). If a series of transfers ends normally and the stored data is valid, the contents of DTARB can be copied to RLARB (marking). Note, however, that in repeat mode the value in RLARB is copied to DTARB automatically when a boundary overflow occurs.

5.2.6 Compare Address Register B (CPARB)

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

CPARB is an auxiliary pointer used in ring buffer mode. The contents of CPARB are constantly compared with DTARB. When they match, an interrupt request can be generated.

CPARB should be updated by software so that it always indicates the top of the unprocessed data in the ring buffer. Then if the ring buffer becomes full of unprocessed data, causing an overrun error, CPARB and DTARB will match and an interrupt request will occur.

5.2.7 Serial/Timer Control Register (STCR)

BB

Bit	7	6	5	4	3	2	1	0
	RING	CMPF	CMPIE	LOAD	MARK	—	ICKS1	ICKS0
Initial value	0	0	0	1	1	1	0	0
Read/Write	R/W	R/(W)*	R/W	(W)	(W)	—	R/W	R/W

Note: * Software can write a 0 in bit 6 to clear the flag, but cannot write a 1 in this bit.

STCR is an 8-bit readable/writable register that controls DTU channel B, and selects the SCI operating mode and the TCNT clock source. STCR is initialized to H'1C by a reset.

Bit 7—Ring Buffer Mode (RING): Setting this bit to 1 places DTU channel B in ring buffer mode. To use ring buffer mode it is also necessary to set the REPB bit to 1 in IOCR.

Bit 7 RING	Description	
0	DTU channel B does not operate in ring buffer mode	(Initial value)
1	DTU channel B operates in ring buffer mode	

Bit 6—Compare Interrupt Flag (CMPF): Overrun error interrupt request flag for the ring buffer. This flag indicates that the contents of CPARB and DTARB match after DTARB was incremented due to the occurrence of a DTU cycle.

Bit 6 CMPF	Description	
0	[Clearing condition]	
	Read STCR while CMPF = 1, then write 0 in CMPF	(Initial value)
1	Ring buffer overrun error	
	[Setting condition]	
	When DTARB contents match CPARB contents after being incremented by DTU cycle occurrence	

Bit 5—Compare Interrupt Enable (CMPIE): Enables or disables the interrupt (CMPI) requested when CMPF is set to 1.

Bit 5 CMPIE	Description	
0	Interrupt request (CMPI) by CMPF is disabled	(Initial value)
1	Interrupt request (CMPI) by CMPF is enabled	

Bit 4—Pointer Load (LOAD): Controls the copying of the contents of the auxiliary pointer (RLARB) into the ring buffer pointer (DTARB). There is no latch to retain the value of the LOAD bit. The load operation is executed when the LOAD bit is cleared to 0.

Bit 4 LOAD	Description	
Cleared to 0	RLARB contents are copied to DTARB	
Set to 1	No operation	(Initial value)

Bit 3—Pointer Mark (MARK): Controls the copying of the contents of the ring buffer pointer (DTARB) into the auxiliary pointer (RLARB). There is no latch to retain the value of the MARK bit. The mark operation is executed when the MARK bit is cleared to 0.

Bit 3 MARK	Description	
Cleared to 0	DTARB contents are copied to RLARB	
Set to 1	No operation	(Initial value)

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

Bits 1 and 0—Internal Clock Source Select 1 and 0 (ICKS1 and ICKS0): These bits, together with bits CKS2 to CKS0 in TCR, select the TCNT clock source. For details see section 11, 8-Bit Timers.

5.2.8 DPRAM Data Registers (DPDRWH, DPDRWL, DPDRRH, DPDRRL)

BB	DI	H/S

The DPRAM data registers (DPDR registers) provide four bytes, of which the master CPU can write to two bytes (DPDRW) and read two bytes (DPDRR). These data registers are used in DPRAM bound buffer mode, DPRAM direct word mode, and handshake mode. They are not used for buffer queries in DPRAM mode, or for I/O transfers.

Read and write access to the DPDR registers sets interrupt flags (MWEF, MREF), activates the DTU, and changes the levels of control signals ($\overline{\text{RDY}}$, $\overline{\text{WRQ}}$) sent to the master CPU. For details of these operations, see section 5.3, Operation.

DPRAM Data Register Write H and L (DPDRWH, DPDRWL): These data registers are reserved for write access by the master CPU in DPRAM bound buffer mode, DPRAM direct word mode, and handshake mode. They are not used for buffer queries in DPRAM mode, or for I/O transfers.

			Bit	7	6	5	4	3	2	1	0
			Initial value	—	—	—	—	—	—	—	—
Mode	Register										
DRAM bound buffr mode	DPDRWH	Internal CPU	Read/Write	R*1	R*1	R*1	R*1	R*1	R*1	R*1	R*1
	DPDRWL	Master CPU	Read/Write	W	W	W	W	W	W	W	W
DRAM direct word mode	DPDRWH	Internal CPU	Read/Write	R	R	R	R	R	R	R	R
	DPDRWL	Master CPU	Read/Write	W	W	W	W	W	W	W	W
Handshake mode	DPDRWL*3	Internal CPU	Read/Write	R	R	R	R	R	R	R	R
		Master CPU	Read/Write	W*2	W*2	W*2	W*2	W*2	W*2	W*2	W*2

Notes: 1. Transferred to on-chip RAM automatically by the DTU.
2. Data on the DDB lines is latched at the rising edge of the $\overline{WE}$ input.
3. In handshake mode, the master CPU cannot access DPDRWH.

The two DPDRW bytes should be used in each mode as follows.

- DPRAM Bound Buffer Mode

  DPDRWH and DPDRWL are reserved for write access by the master CPU. Data written in these registers is transferred automatically to the on-chip RAM. The master CPU has two addresses for these two bytes, but write access to both bytes operates in the same way.

- DPRAM Direct Word Mode

  Read or write access to DPDRWL generates an interrupt request to the internal CPU or master CPU. If both bytes are used, read or write access should be performed to DPDRWH first, then DPDRWL. If only one byte is used, use DPDRWL.

- Handshake Mode

  Use DPDRWL.

DPRAM Data Register Read H and L (DPDRRH, DPDRRL): These data registers are reserved for read access by the master CPU in DPRAM bound buffer mode, DPRAM direct word mode, and handshake mode. They are not used for buffer queries in DPRAM mode, or for I/O transfers.

			Bit	7	6	5	4	3	2	1	0
			Initial value	—	—	—	—	—	—	—	—
Mode	Register										
DRAM bound buffr mode	DPDRRH	Internal CPU	Read/Write	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1
	DPDRRL	Master CPU	Read/Write	R	R	R	R	R	R	R	R
DRAM direct word mode	DPDRRH	Internal CPU	Read/Write	W	W	W	W	W	W	W	W
	DPDRRL	Master CPU	Read/Write	R	R	R	R	R	R	R	R
Handshake mode	DPDRRL*3	Internal CPU	Read/Write	W	W	W	W	W	W	W	W
		Master CPU	Read/Write	R*2	R*2	R*2	R*2	R*2	R*2	R*2	R*2

Notes: 1. Transferred from on-chip RAM automatically by the DTU.
2. Data is output on the DDB lines when the $\overline{OE}$ input is low.
3. In handshake mode, the master CPU cannot access DPDRRH.

The two DPDRR bytes should be used in each mode as follows.

- DPRAM Bound Buffer Mode

  DPDRRH and DPDRRL are reserved for read access by the master CPU. The read data is transferred automatically from on-chip RAM. The master CPU has two addresses for these two bytes, but read access to both bytes operates in the same way.

- DPRAM Direct Word Mode

  Read or write access to DPDRRL generates an interrupt request to the internal CPU or master CPU. If both bytes are used, DPDRRH should be accessed first, then DPDRRL. If only one byte is used, use DPDRRL.

- Handshake Mode

  Use DPDRRL.

5.2.9 DPRAM Data Register Read Query (DPDRRQ)

This register is used as a query buffer in DPRAM mode.

		Q
	Bit	7 6 5 4 3 2 1 0
	Initial value	— — — — — — — —
Internal CPU	Read/Write	W* W* W* W* W* W* W* W*
Master CPU	Read/Write	R/W R/W R/W R/W R/W R/W R/W R/W

Note: * Transferred automatically from on-chip RAM by the DTU.

DPDRRQ is reserved for read and write access by the master CPU. When a write access to DPDRRQ occurs, data stored in on-chip RAM at the address given by DTARH (upper byte) and DPDRRQ (lower byte) is transferred to DPDRRQ.

5.2.10 Parallel Communication Control/Status Register (PCCSR)

Q	BB	DI	H/S

This register is used in DPRAM mode and handshake mode. It is not used in I/O transfers.

PCCSR can be written and read by both the internal CPU and the master CPU. It controls data transfer between the internal CPU and master CPU, and indicates status.

		Bit	7	6	5	4	3	2	1	0
			QREF	EWRQ	EWAKAR	ERAKAR	MWEF	MREF	EMWI	EMRI
		Initial value	0	0	0	0	0	0	0	0
Mode										
All modes except handshake mode	Internal CPU	Read/Write	R	R	R	R	R/(W)*	R/(W)*	R/W	R/W
	Master CPU	Read/Write	R	R/W	R/W	R/W	R	R	R	R
Handshake mode	Internal CPU	Read/Write	R	R/W	R/W	R/W	R/(W)*	R/(W)*	R/W	R/W

Note: * Only 0 write after read is available.

Bit 7—Query Read End Flag (QREF): Indicates whether DPDRRQ contains an on-chip RAM address or data. This flag is useful in buffer query operations in DPRAM mode.

Bit 7 QREF	Description	
0	DPDRRQ contains data	(Initial value)
	[Clearing condition]	
	The DTU writes on-chip RAM data in DPDRRQ	
1	DPDRRQ contains the lower byte of an on-chip RAM address	
	[Setting condition]	
	The master CPU writes the lower byte of an on-chip RAM address in DPDRRQ	

Bit 6—Enable Wait Request (EWRQ): Enables operation of the $\overline{\text{WRQ}}$ pin.

Bit 5—Enable Write Acknowledge and Request (EWAKAR): Enables operation of the $\overline{\text{RDY}}$ pin in response to master write access.

Bit 4—Enable Read Acknowledge and Request (ERAKAR): Enables operation of the $\overline{\text{RDY}}$ pin in response to master read access.

Table 5.8 lists the pin states in single-chip mode. Table 5.9 lists the pin states in the expanded modes.

Table 5.8 Pin States in Single-Chip Mode with PCCSR Conditions

Condition	Bit 6 EWRQ	Bit 5 EWAKAR	Bit 4 ERAKAR	$P9_5/\overline{RDY}$	$P8_3/\overline{WRQ}/\overline{XRDY}$
DPME = 1, HSCE = 0	*	0	0	High level output (high impedance)	Depends on value of EWRQ as below
	*	1	0	$\overline{RDY}$ output enabled at master write	
	*	0	1	$\overline{RDY}$ output enabled at master read	
	*	1	1	$\overline{RDY}$ output enabled at master write and master read	
	0	*	*	Depends on value of ERAKAR and EWAKAR as above	Port function
	1	*	*		$\overline{WRQ}$ output enabled
DPME = 1, HSCE = 1	0	*	*	$\overline{RDY}$ output enabled	Port function
	1	*	*	$\overline{RDY}$ output enabled	$\overline{WRQ}$ output enabled

Table 5.9 Pin States in Expanded Modes with PCCSR Conditions

Condition	Bit 6 EWRQ	Bit 5 EWAKAR	Bit 4 ERAKAR	$P9_5/\overline{RDY}$	$P8_3/\overline{WRQ}/\overline{XRDY}$
DPME = 1, HSCE = 0	0	0	0	$\overline{AS}$ output	High level output
	0	1	0	$\overline{AS}$ output	$\overline{XRDY}$ output enabled at master write
	0	0	1	$\overline{AS}$ output	$\overline{XRDY}$ output enabled at master read
	0	1	1	$\overline{AS}$ output	$\overline{XRDY}$ output enabled at master write and master read
	1	*	*	$\overline{AS}$ output	$\overline{WRQ}$ output enabled
DPME = 1, HSCE = 1	0	*	*	$\overline{AS}$ output	$\overline{XRDY}$ output enabled
	1	*	*	$\overline{AS}$ output	$\overline{WRQ}$ output enabled

Table 5.10 describes $\overline{RDY}$ output operations. Table 5.11 describes $\overline{WRQ}$ operations.

Table 5.10 $\overline{RDY}$ Output Operations

Mode		Conditions for High Level Output*	Conditions for Low Level Output
DPRAM bound buffer mode	Master read	Master CPU reads DTARA	Internal CPU clears MREF
	Master write	Master CPU writes to DTARB	Internal CPU clears MWEF
DPRAM direct word mode	Master read	Master CPU reads DPDRRL	Internal CPU writes to DPDRRL
	Master write	Master CPU writes to DPDRWL	Internal CPU reads DPDRWL
Handshake mode		Internal CPU reads DPDRWL	$\overline{OE}$ is high and $\overline{WE}$ is low

Note: * During single-chip mode, pin $\overline{RDY}$ becomes an NMOS open drain output pin. High level output at this time refers to the high impedance state.

Table 5.11 $\overline{WRQ}$ Output Operations

Mode		Conditions for High Level Output	Conditions for Low Level Output
DPRAM bound buffer mode	Master read	DTU completes transfer to DPDRRH/L	Master CPU reads DPDRRH/L before DTU transfers new data
	Master write	DTU completes transfer from DPDRWH/L	Master CPU writes to DPDRWH/L before DTU transfers old data
Buffer query in DPRAM mode		DTU completes transfer to DPDRRQ	Master CPU reads DPDRRQ before DTU transfers new data
Handshake mode		8 system clocks after $\overline{OE}$ and $\overline{WRQ}$ both become low	Internal CPU writes to DPDRRL

Bit 3—Master Write End Flag (MWEF)

Bit 2—Master Read End Flag (MREF)

Table 5.12 describes the operation of MREF and MWEF. Table 5.13 explains the meaning of MREF and MWEF.

Table 5.12 MREF and MWEF Operations

Mode		Clearing Condition	Setting Condition
DPRAM bound buffer mode	MREF	Internal CPU reads MREF = 1, then writes 0 in MREF	DTU channel A has completed transfer up to boundary
	MWEF	Internal CPU reads MWEF = 1, then writes 0 in MWEF	DTU channel B has completed transfer up to boundary
DPRAM direct word mode	MREF	Internal CPU writes to or dummy-reads DPDRRL	Master CPU reads DPDRRL
	MWEF	Internal CPU reads or dummy-writes to DPDRWL	Master CPU writes to DPDRWL
Handshake mode	MREF	Internal CPU writes to or dummy-reads DPDRRL	$\overline{OE}$ goes high
	MWEF	Internal CPU reads or dummy-writes to DPDRWL	$\overline{WE}$ goes high

Table 5.13 Meaning of MREF and MWEF

Mode		0	1
DPRAM bound buffer and direct word modes	MREF	Internal CPU has finished preparing data to be read by master CPU	Master CPU has finished reading data prepared by internal CPU
	MWEF	Internal CPU has finished processing data written by master CPU	Master CPU has finished writing data to be processed by internal CPU
Handshake mode	MREF	Internal CPU has prepared data to be output to master CPU	Master CPU has read data being output by internal CPU
	MWEF	Internal CPU has read data latched by master CPU	Master CPU has caused internal CPU to latch data

Bit 1—Enable Master Write Interrupt (EMWI): Enables or disables the interrupt (MWEI) requested by MWEF when MWEF is set to 1.

Bit 1 EMWI	Description	
0	Disables interrupt request (MWEI) by MWEF	(Initial value)
1	Enables interrupt request (MWEI) by MWEF	

Bit 0—Enable Master Read Interrupt (EMRI): Enables or disables the interrupt (MREI) requested by MREF when MREF is set to 1.

Bit 0 EMRI	Description	
0	Disables interrupt request (MREI) by MREF	(Initial value)
1	Enables interrupt request (MREI) by MREF	

5.2.11 System Control Register (SYSCR)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	XRST	NMIEG	DPME	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit 1—DPRAM Enable Bit (DPME): Selects whether to put the chip into slave mode. When using a transfer mode other than I/O transfer, the chip must be put into slave mode. See tables 5.5 and 5.6. DPME is initialized at reset and in hardware standby mode.

Bit 1 DPME	Description	
0	The chip is not put into slave mode	(Initial value)
1	The chip is put into slave mode	

For descriptions of other bits of SYSCR, refer to section 3.2, System Control Register.

5.3 Operation

5.3.1 DTU Operation

DTU Operation Overview: The DTU operates by sharing the CPU's data bus, address bus, and bus control signals. To carry out a transfer, the DTU gets the bus right and generates a DTU bus cycle. During the DTU bus cycle, CPU bus cycles are held pending, while internal CPU operations can proceed concurrently.

A DTU bus cycle is initiated by a DTU bus request. In case of contention between CPU and DTU bus cycle requests, the DTU bus cycle has priority. A request for a DTU bus cycle can occur at the end of a CPU bus cycle or DTU bus cycle. When the CPU is operating internally or in sleep mode, a request can occur at any time.

A DTU bus cycle bus request is generated in response to a transfer request. The source of the transfer request may be an interrupt request from an on-chip supporting module, or a low-to-high transition at the $\overline{OE}$ or $\overline{WE}$ pin in the PBI. These transfer requests are recognized when the CPU is active or in sleep mode, but are not recognized in the reset state or standby modes.

Types of DTU Bus Cycles: Table 5.14 classifies the bus cycles generated by the DTU and indicates their transfer request sources.

A DPRAM cycle occurs when the DTU acquires the bus right for a PBI transfer request. There are two types of DPRAM cycles: master read, and master write. DTU channels R and A can generate master read cycles. DTU channel B can generate master write cycles.

Dead cycles, read cycles, and write cycles occur when the DTU acquires the bus right for an I/O transfer request. I/O transfers can be carried out on three DTU channels: channels A, B, and C. Each channel executes a series of DTU bus cycles consisting of a dead cycle, read cycle, and write cycle.

Table 5.14 Types of DTU Bus Cycles

Type	Name	Operation	Request Source
PBI transfer cycle	Master read cycle	Read on-chip RAM and simultaneously input data to PBI	PBI
	Master write cycle	Output data from PBI and simultaneously write to on-chip RAM	PBI
I/O transfer cycle	Dead cycle	Confirmation of transfer request	Supporting module
	Read cycle	Read on-chip RAM or register of on-chip supporting module	DTU
	Write cycle	Write to on-chip RAM or register of on-chip supporting module	DTU

DTU Bus Cycle Processing: Data is transferred from on-chip RAM to the PBI in a master read cycle, or from the PBI to on-chip RAM in a master write cycle.

A dead cycle is a one-state waiting interval to be certain of a transfer request. During a DTU bus cycle after the dead cycle, the transfer request cannot be cancelled by the CPU. During the dead cycle, the flag of the free-running timer (FRT) or A/D converter, the source of the transfer request, is cleared. Data is transferred from on-chip RAM or an on-chip supporting module to the DTU in a read cycle, and from the DTU to on-chip RAM or an on-chip supporting module in a write cycle. Interrupt flags in the SCI are cleared automatically when the receive data register is read or when data is written in the transmit data register.

Canceling an I/O Transfer Request: I/O transfer requests are generated by interrupt flags in the on-chip supporting modules. If the CPU clears the interrupt flag, the transfer request will be cancelled. If the transfer request is cancelled during a dead cycle, the following read cycle and write cycle will not be executed.

Priority of Transfer Requests and Bus Requests: Table 5.15 indicates the priority order of transfer requests. A transfer request for a master read cycle caused by a buffer query in DPRAM mode always has highest priority. Among transfer requests in DPRAM bound buffer mode, master read has priority over master write.

I/O transfers are basically performed in alphabetical order of channels (A,B,C), but in some cases a bus request will not be generated even though a transfer request is present. No bus request is generated for a one-state interval (dead cycle) after a write cycle.

PBI transfers take priority over I/O transfers, so a PBI transfer may be inserted during the execution of a series of I/O transfer cycles. If there is a PBI transfer request at the end of a read cycle, the write cycle will be held pending until all PBI transfer processing is completed.

Table 5.15 Priority Order of DTU Transfer Requests

	Priority of DTU Bus Cycle Activation						
Preceding Operation	**Master Read Cycle (Ch. R)**	**Master Read Cycle (Ch. A)**	**Master Write Cycle (Ch. B)**	**Dead Cycle**	**Read Cycle**	**Write Cycle**	**Pending Write Cycle**
Other than below	1	2	3	5	—	—	4
Dead cycle of cancelled bus request	1	2	3	4	—	—	—
Dead cycle	—	—	—	—	1	—	—
Read cycle	1	2	3	—	—	4*	—
Write cycle	1	2	3	—	—	—	—

Notes: Master read cycle in channel R: buffer query in DPRAM mode
Master read cycle in channel A: DPRAM bound buffer mode
Dead cycle priority order: channel A, channel B, channel C
* Write cycle only in the channel that generated the preceding read cycle.
Held pending if a higher-priority PBI transfer cycle is executed.

Address Bus Operations: Figure 5.8 shows a conceptual diagram of the DTU address-bus circuits. The DTARH contents, or H'FF, are output on the upper 8 bits of the 16-bit address bus. The contents of DTARA, DTARB, DTARC, or DPDRRQ, or the address of a supporting-module register determined by an I/O transfer request source, are output on the lower 8 bits.

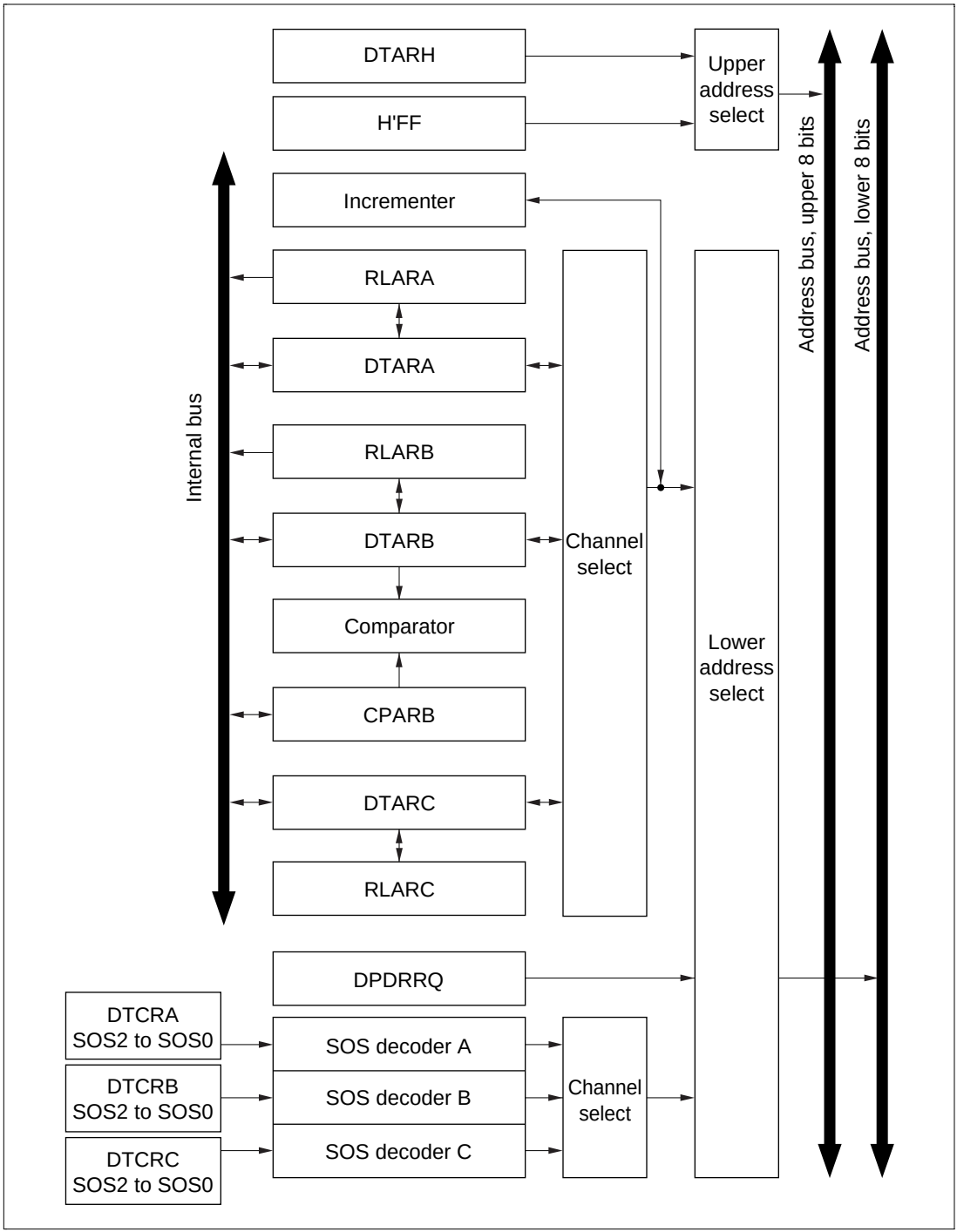


Figure 5.8 Concept of Address Bus Circuits

Data Bus Operations: Figure 5.9 shows a conceptual diagram of the DTU data-bus circuits. The DTU temporary data registers (DTDRH/L) are connected to a 16-bit data bus. Data is read into the DTDR registers in a read cycle, and written out from the DTDR registers in a write cycle. The PBI data registers (DPDR) are connected to an 8-bit data bus. Data is read into DPDRRQ or DPDRRH/L in a master read cycle, and written out from DPDRWH/L in a master write cycle.

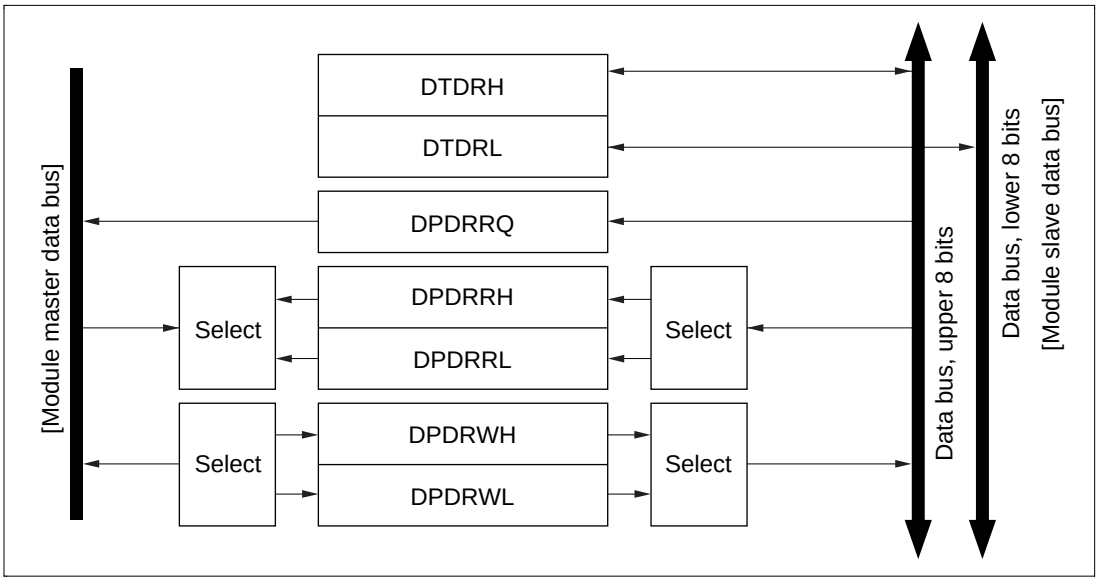


Figure 5.9 Concept of Data Bus Circuits

Table 5.16 Bus Operation in DTU Bus Cycles

Bus Cycle		Upper Address Bus	Lower Address Bus	Upper Data Bus	Lower Data Bus
Master read cycle (channel R)		DTARH →	DPDRRQ →	→ DPDRRQ	—
Master read cycle (channel A)		DTARH →	DTARA →	→ DPDRR	—
Master write cycle (channel B)		DTARH →	DTARB →	DPDRW →	—
Read cycle (RAM → I/O)	Byte	DTARH →	DTAR →	→ DTDRH	—
	Word			→ DTDRH	→ DTDRL
Write cycle (RAM → I/O)	Byte	H'FF →	SOS decoder →	DTDRH →	—
	Word			DTDRH, L →	—
Read cycle (I/O → RAM)	Byte	H'FF →	SOS decoder →	→ DTDRH	—
	Word			→ DTDRH, L	—
Write cycle (I/O → RAM)	Byte	DTARH →	DTAR →	DTDRH →	—
	Word			DTDRH →	DTDRL →

Address Register Boundary: The number of bytes of on-chip RAM transferred by the DTU is determined by the DTAR settings and the boundary designated by DTCR bits BUD2 to BUD0. (See page 87)

First, DTARH determines a 256-byte on-chip RAM area within which the DTU operates. BUD2 to BUD0 specify a 1-, 2-, 4-, 8-, 16-, 32-, 64-, or 128-byte boundary. These eight settings divide the 256-byte area into 256, 128, 64, 32, 16, 8, 4, or 2 smaller areas, respectively, with boundary addresses set on the boundaries between adjacent areas. The starting address of a transfer is given by the contents of DTARA, DTARB, or DTARC. The ending address is the next boundary address.

The contents of DTARA, DTARB, or DTARC are incremented at each transfer. When the transfer is completed as far as the boundary address, a boundary overflow occurs. The DTARA, DTARB, or DTARC contents do not increment past the boundary. The upper bits remain unchanged. The lower bits are initialized to the value stored in RLAR. In normal mode, the DTE bit is cleared.

Figure 5.10 shows an example with 64-byte boundaries, in which the starting address is in the exact center of a 64-byte area, so that a boundary overflow occurs when 32 bytes have been transferred. In repeat mode, if there are further transfer requests, the transfer is repeated on the same 32-byte area.

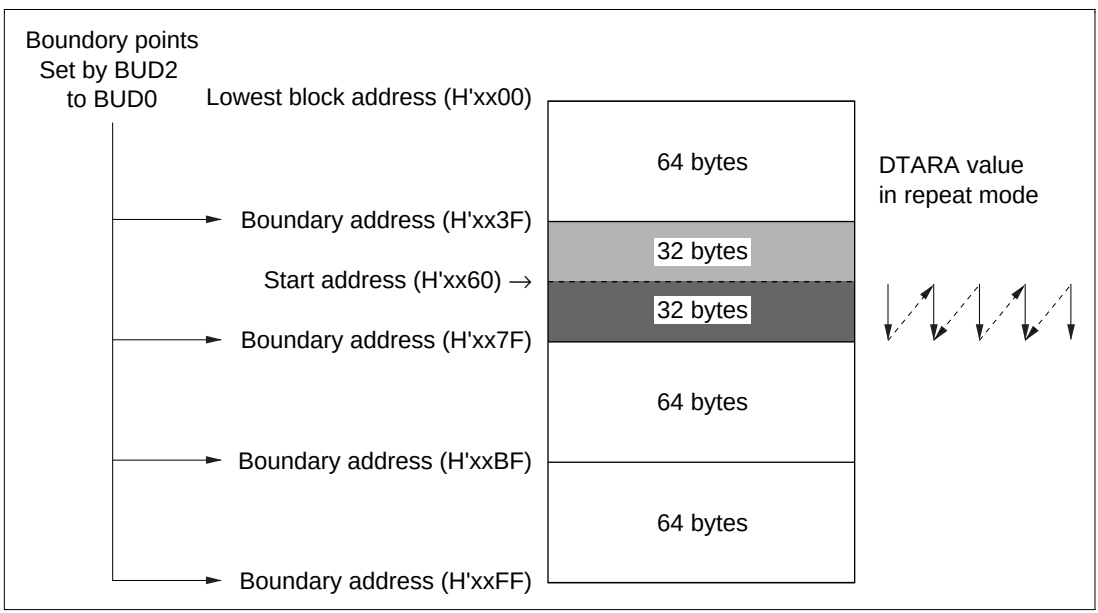


Figure 5.10 Example of Memory Map with 64-Byte Boundary (DTARH = H'xx, DTARA = H'60)

Ring Buffer Operation: In repeat mode, DTU channel B can operate as a ring buffer. In ring buffer operation, a number of bytes equal to the boundary designation can be treated as a ring-type FIFO buffer on RAM.

Data positions in this buffer are indicated by three pointers. The main pointer is DTARB, which points to the address that will be accessed at the next transfer request. One auxiliary pointer is RLARB, which points to the top address of a data block being transferred in accordance with a transfer protocol. If the transfer fails, it can be repeated from the address indicated by RLARB. The other auxiliary pointer is CPARB, which points to the top of the data that has not yet been processed by the CPU. If DTARB catches up with CPARB and the two match, an overrun error interrupt can be generated to report this abnormality.

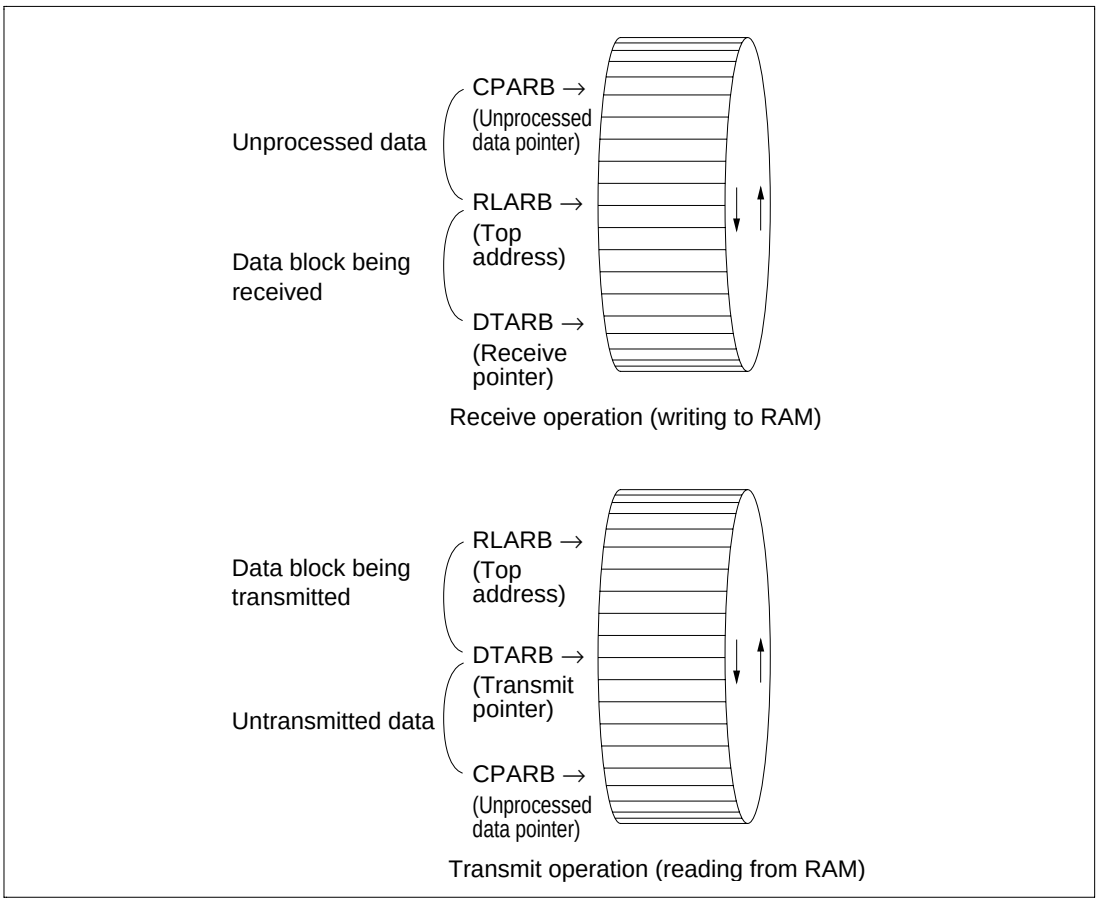


Figure 5.11 Pointer Operation in Ring Buffer

5.3.2 DTU and PBI Initialization

Care is required when setting up the DTU and PBI for DPRAM or handshake mode, because this causes changes in the operating states of input/output pins. An initialization procedure for this setup is given below and shown in figure 5.12. When some DTU channels are used for I/O transfer instead of PBI transfer, these channels should be initialized after the procedure below. Register contents are assumed to be initial values.

1. Write the upper byte of the address of the RAM area to be used by the DTU in DTARH.
   This register is also used for I/O transfers. It must be initialized before I/O transfer begins.

2. Set the EWRQ bit in PCCSR to 1, or clear it to 0.
   If this bit is set to 1, output from the $\overline{\text{WRQ}}$ pin is enabled when the DPME bit is set.

3-1. Set the HSCE bit in IOCR to 1, or clear it to 0.
   This bit selects handshake mode or DPRAM mode when the DPME bit is set. The treatment of the $\overline{\text{CS}}$ and RS_2 to RS_0 inputs differs between handshake mode and DPRAM mode.

3-2. Set the DPEA and DPEB bits in IOCR to 1 or 0.
   In DPRAM mode, these bits select direct word mode or bound buffer mode. They also limit access to DTARA, DTARB, DTCRA, and DTCRB. Some write accesses to registers that are shared with other operating modes are disabled by hardware if they would interfere with the operation of the other mode.

4. Set the DPME bit in SYSCR to 1.
   This sets the pins to slave mode (DPRAM mode or handshake mode) according to the chip operating mode (single-chip mode or expanded mode). Pins DDB_7 to DDB_0 (XDDB_7 to XDDB_0) are set for input/output, $\overline{\text{CS}}$ ($\overline{\text{XCS}}$), RS_2 to RS_0, $\overline{\text{WE}}$ ($\overline{\text{XWE}}$), and $\overline{\text{OE}}$ ($\overline{\text{XOE}}$) for input, and $\overline{\text{RDY}}$ ($\overline{\text{XRDY}}$) and $\overline{\text{WRQ}}$ for output. Pins $\overline{\text{WRQ}}$, $\overline{\text{CS}}$, and RS_2 to RS_0 also depend on the settings of the EWRQ and HSCE bits. At the same time, it becomes possible to set and clear the QREF, MWEF, and MREF flags in PCCSR. The inputs at $\overline{\text{CS}}$, $\overline{\text{WE}}$, $\overline{\text{OE}}$, etc. must be kept at the inactive level until initialization is completed.

5. Set the EWAKAR, ERAKAR, EMWI, and EMRI bits in PCCSR to 1, or clear them to 0.
   These bits select the usage of the $\overline{\text{RDY}}$ pin and enable or disable interrupts to the internal CPU. The usage of the $\overline{\text{RDY}}$ pin can be set from the master CPU.

Data can now be transferred in DPRAM mode or handshake mode by accessing registers PCCSR, DTCRA, DTCRB, DTARA, DTARB, DPDRRH, DPDRRL, DPDRWH, and DPDRWL according to the operating mode.

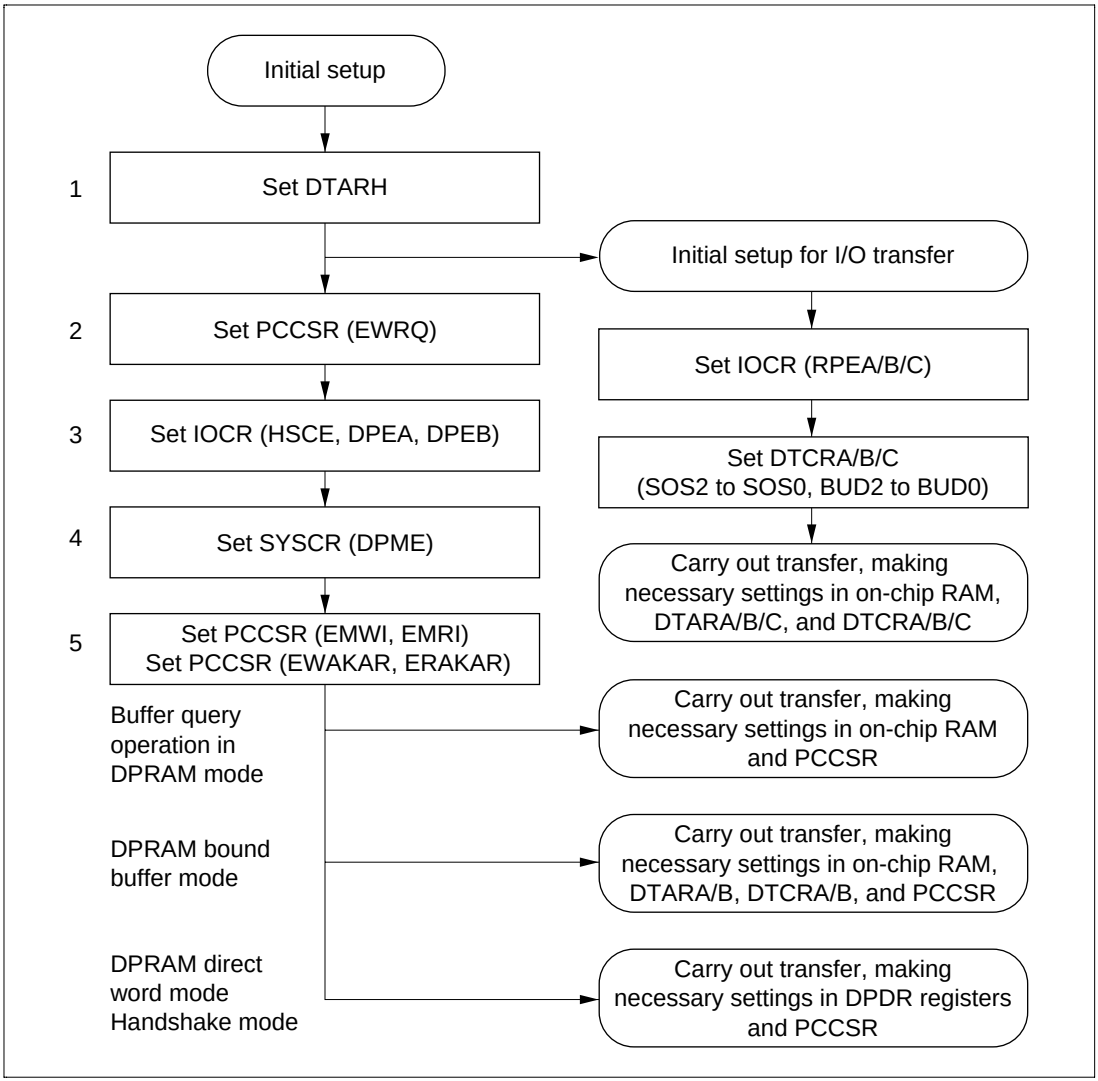


Figure 5.12 DTU and PBI Initialization Flowchart

5.3.3 I/O Transfer Operations

The conditions under which I/O transfer is available are given below, with initialization and operating procedures.

Conditions under Which I/O Transfer is Available

- Channel R
  Permanently connected to the PBI. This channel is not available for I/O transfer.
- Channel A
  Connected to the PBI when DPME = 1, HSCE = 0, and DPEA = 1. Available for I/O transfer at other times.
- Channel B
  Connected to the PBI when DPME = 1, HSCE = 0, and DPEB = 1. Available for I/O transfer at other times.
- Channel C
  Always available for I/O transfer.

I/O Transfer Procedure (See figure 5.13.)

1. Write the upper byte of the address of the RAM area to be used by the DTU in DTARH.
   This register is also used for PBI transfers. It must be initialized before PBI transfer begins.
2. Set bits RPEA, RPEB, and RPEC in IOCR to 1 or clear them to 0 as necessary.
3. Set bits SOS2 to SOS0 and BUD2 to BUD0 in DTCRA, DTCRB, or DTCRC.
4. Initialize the on-chip supporting modules so that they will generate transfer requests for the desired I/O transfer.
5. For a transfer from RAM to an on-chip supporting module register, place the data to be transferred on RAM. Write the top address of the data on RAM in DTARA, DTARB, or DTARC.
6. Set the DTE bit to 1 in DTARA, DTARB, or DTARC.
   Set the DTIE bit to 1 in DTARA, DTARB, or DTARC as necessary.
7. Transfer requests are input from the on-chip supporting modules. The DTU responds by transferring data, then clearing the transfer request. DTARA, DTARB, or DTARC is incremented each time one byte is transferred. After repeated transfer requests, when the transfer reaches the boundary, in normal mode, a boundary overflow occurs and the DTE bit is cleared to 0. If the DTIE bit is set to 1, an interrupt is requested.
8. For a transfer from an on-chip supporting module to RAM, process the data transferred onto RAM.

Steps 5 to 8 can be carried out repeatedly.

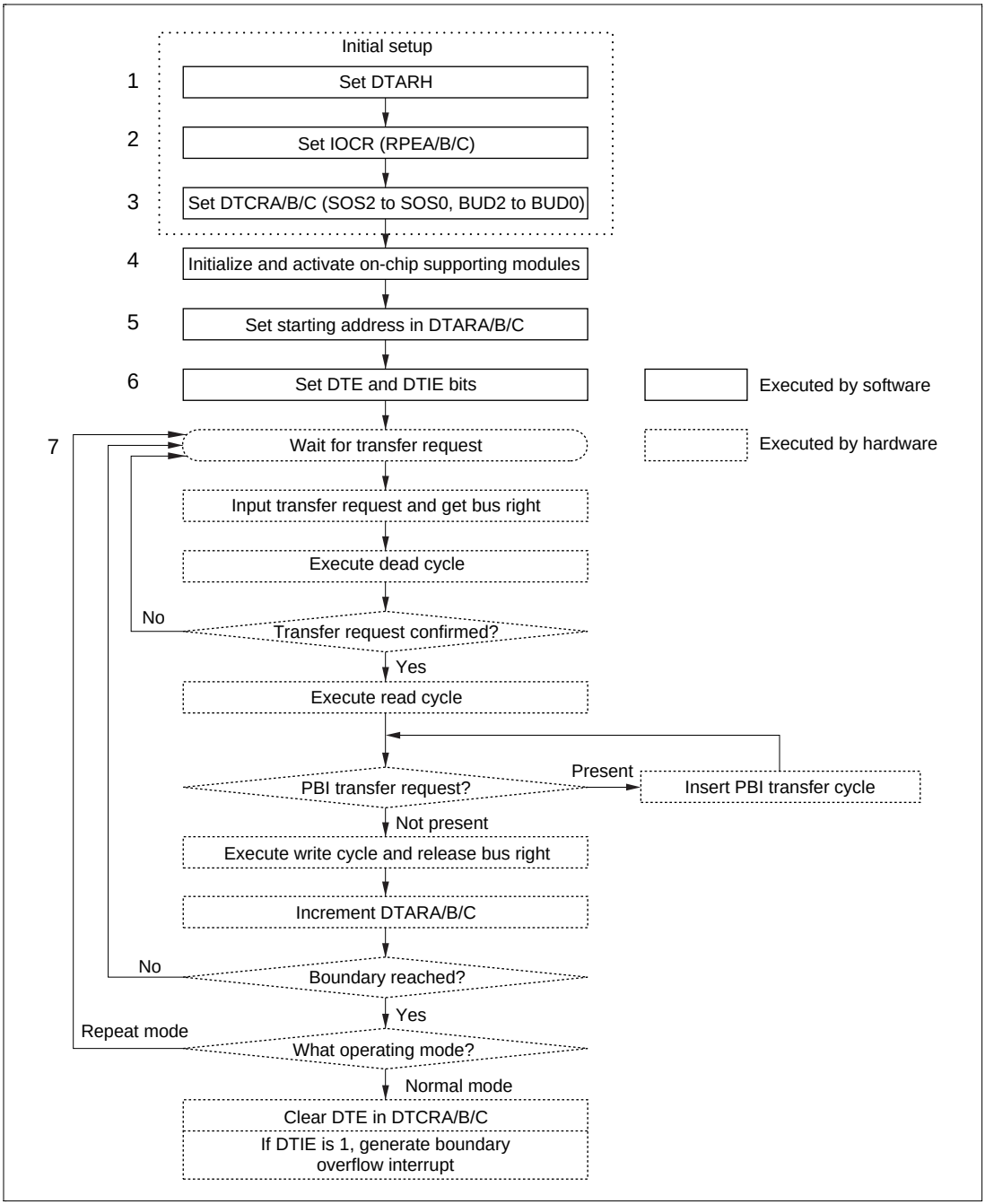


Figure 5.13 I/O Transfer Flowchart

I/O Transfer Precautions: When using I/O transfer, note the following points.

- If software clears the DTE bit to 0, I/O operations will be suspended after the end of the bus cycle currently being executed. Although this is not a boundary overflow, if the condition DTE = 0 and DTIE = 1 is true, an interrupt request will be generated. To avoid generating an interrupt request, when clearing DTE, clear the DTIE bit to 0 at the same time.

- While the DTE bit is set to 1, interrupt requests from the source selected by SOS2 to SOS0 will not be sent to the interrupt controller.

  Data should be prepared and processed after the DTE bit is cleared to 0, but if an interrupt request is generated before the DTE flag is set to 1 again, this interrupt request will go to the interrupt controller.

  To have interrupts handled by I/O transfer and not treated as CPU interrupt requests, the time taken to prepare and process data must be shorter than the time between interrupt requests. If this is not possible, software should implement a double-buffering scheme by writing to DTARA, DTARB, or DTARC at each boundary overflow to change the address.

- After a transfer request is accepted from a supporting module, if the same transfer request occurs again before the old request is cleared (before the interrupt flag is cleared), the DTU will be unable to recognize the new request. The interval between transfer requests should be comfortably longer than the DTU's turnaround time.

- If an I/O transfer is cancelled just when a transfer request occurs, the cancellation is carried out during the dead cycle.

- The address relations of word access by the DTU need to be noted. In an I/O transfer, some supporting-module registers are accessed by word access, in which case the on-chip RAM is also accessed by word access. In word access, address bit 0 is disregarded, so the upper byte must be located at an even address and the lower byte at an odd address. The supporting-module registers satisfy this condition, but it is possible to write odd addresses in the DTU address registers (DTARA, DTARB, DTARC) that specify the RAM address.

  When making a word-access I/O transfer, write an even address in the DTU address register. The transfer may not be performed as intended if an odd address is written. For example, if H'01 is written and word access is performed, the lower address bits will be output as H'00 in access to the upper byte, and H'01 in access to the lower byte.

5.3.4 Buffer Query in DPRAM Mode

Buffer query in DPRAM mode provides a parallel interface with random access (read access) to a maximum 256 bytes of on-chip RAM. The interface makes use of DTU channel R to execute PBI transfers. Buffer query operations are depicted in block-diagram form in figure 5.14.

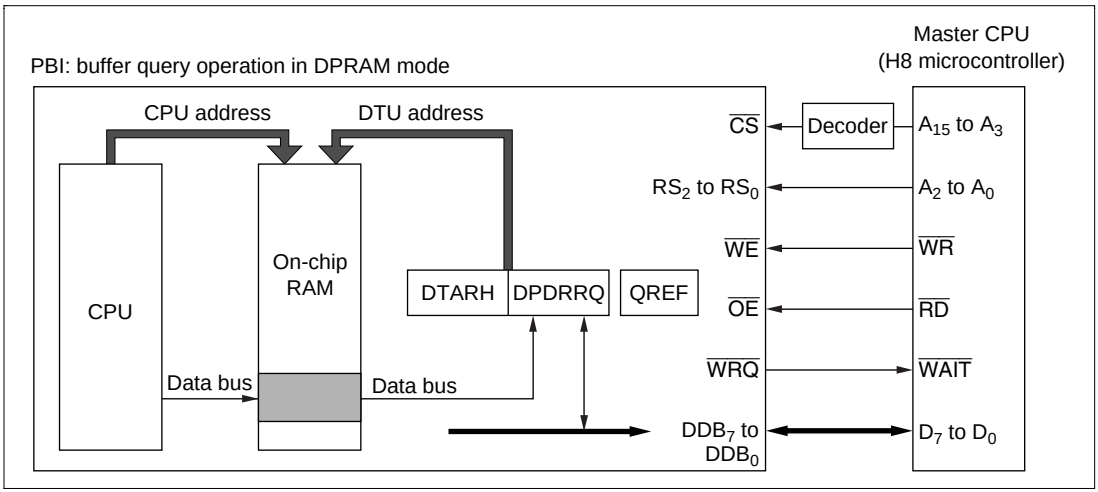


Figure 5.14 Buffer Query in DPRAM Mode

The conditions under which buffer query is available are given below, with initialization and operating procedures.

Conditions under Which Buffer Query is Available: Buffer query is available whenever the HSCE bit is cleared to 0 and the DPME bit is set to 1, enabling use of DPRAM-related pin functions. Availability is not affected by the mode of DTU channels A and B (DPRAM bound buffer mode or DPRAM direct word mode).

Initialization and Operating Procedures (See figure 5.15.)

1. Initialize the DTU and PBI by the DTU/PBI initialization procedure.
   DTARH is also used in I/O transfers and DPRAM bound buffer mode. It must be initialized before these transfers begin. The EWRQ bit in PCCSR and the $\overline{\text{WRQ}}$ pin are also used in DPRAM bound buffer mode and DPRAM direct word mode. The initial settings must be free of conflict between modes.

2. When the master CPU writes to DPDRRQ, the QREF flag in PCCSR is set to 1 to indicate that DPDRRQ contains the lower 8 bits of an on-chip RAM address.

3. A transfer request is generated for DTU channel R, and the bus right is acquired for a master read cycle.

4. The contents of the on-chip RAM address indicated by DTARH and DPDRRQ are transferred to DPDRRQ. The QREF flag is cleared to 0 to indicate that DPDRRQ contains a copy of on-chip RAM data.
5. The master CPU should check that the QREF flag is 0, then read DPDRRQ.

Precaution on Use of Buffer Query in DPRAM Mode: When using buffer query in DPRAM mode, note the following point:

If the master CPU reads DPDRRQ while QREF = 1, the $\overline{\text{WRQ}}$ pin will stay at the low output level until QREF is cleared. If the master CPU does not use $\overline{\text{WRQ}}$ as a wait request signal, leave a sufficient interval from DPDRRQ write to DPDRRQ read to be sure that the transfer has been carried out.

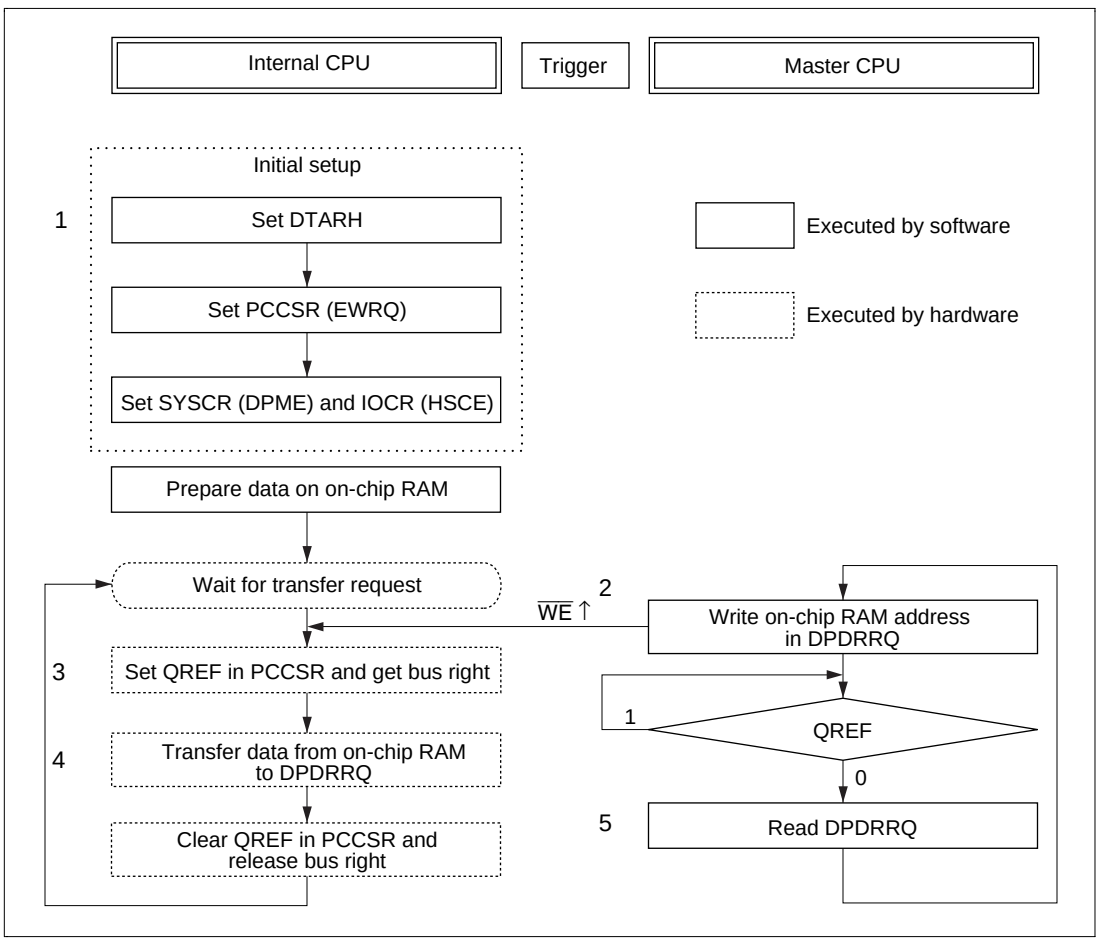


Figure 5.15 Flowchart of Buffer Query in DPRAM Mode

5.3.5 Operation in DPRAM Bound Buffer Mode

DPRAM bound buffer mode provides a parallel interface with sequential access to a maximum 128 bytes of on-chip RAM. The interface makes use of PBI transfers executed by DTU channels A and B.

The conditions under which DPRAM bound buffer mode is available are given below, with initialization and operating procedures.

Conditions under Which DPRAM Bound Buffer Mode is Available

- DPRAM bound buffer mode, read access (DTU channel A)
  Available when DPME = 1, HSCE = 0, and DPEA = 1.
- DPRAM bound buffer mode, write access (DTU channel B)
  Available when DPME = 1, HSCE = 0, and DPEB = 1.

Initialization Procedure: Initialize the DTU and PBI by the DTU/PBI initialization procedure.

DTARH is also used in I/O transfers, and in DPRAM buffer queries. It must be initialized before these transfers begin.

The EWRQ bit in PCCSR and the $\overline{\text{WRQ}}$ and $\overline{\text{RDY}}$ pins are also used in DPRAM buffer queries and DPRAM direct word mode. The initial settings must be free of conflict between modes. (See tables 5.8 and 5.9.)

Read Procedure for DPRAM Bound Buffer Mode: DPRAM bound buffer mode provides a parallel interface with sequential read access to a maximum 128 bytes of on-chip RAM, using PBI transfers executed by DTU channel A. Read operations in DPRAM bound buffer mode are depicted in block-diagram form in figure 5.16.

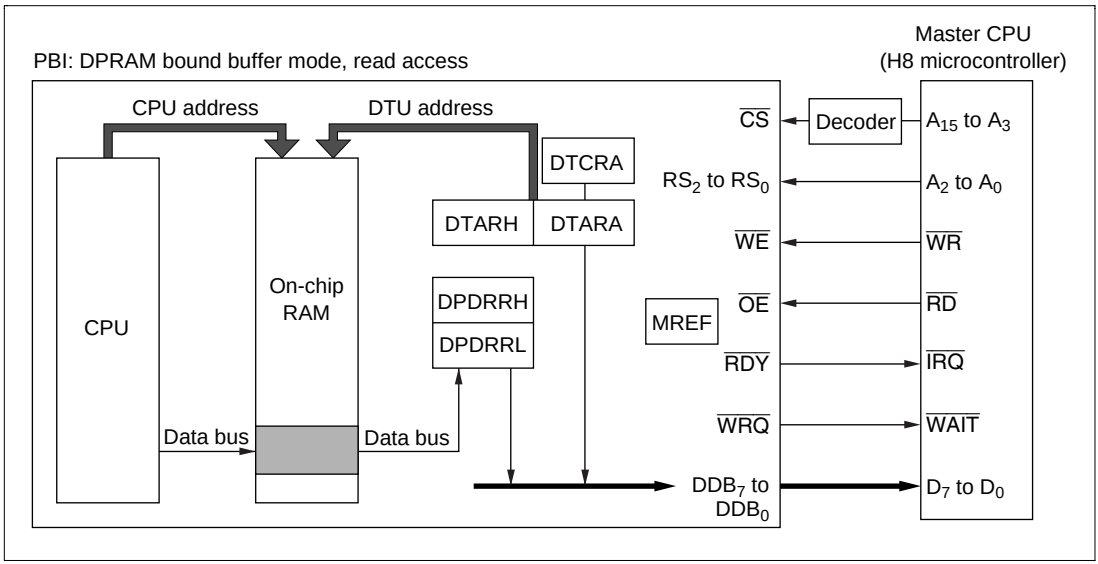


Figure 5.16 Read Access in DPRAM Bound Buffer Mode

The read procedure for DPRAM bound buffer mode is given below. (See figure 5.17.)

1. The internal CPU prepares the data to be transferred on on-chip RAM. The data should be located on contiguous addresses. Position the top address so that the last data will be located on a boundary address. The internal CPU sets the boundary in DTCRA and the top address in DTARA.

2. The internal CPU clears the MREF flag in PCCSR to notify the master CPU that preparations for the transfer are completed. If the ERAKAR bit is set to 1 in PCCSR, a low $\overline{\text{RDY}}$ signal is output.

3. The master CPU learns from the state of the $\overline{\text{RDY}}$ signal or MREF flag that transfer preparations have been completed, reads the boundary value in DTCRA and top address in DTARA, and calculates the number of bytes of data. The $\overline{\text{RDY}}$ output goes high. Then the master CPU reads the DPDRR registers a number of times equal to the number of data bytes.

4. When the master CPU reads DPDRR, the PBI requests a transfer on DTU channel A. The DTU gets the bus right and executes a master read cycle, transferring the contents of the on-chip RAM address specified by DTARH and DTARA to an empty DPDRR register. Even and odd addresses in DTARA correspond to DPDRRH and DPDRRL, respectively. When the master CPU reads the DPDRR registers, it receives their data in the same order in which the data was transferred to the DPDRR registers. After being read, each DPDRR register becomes empty. Each time a DPDRR register becomes empty, the PBI requests another transfer on DTU channel A, to keep the DPDRR registers in the full state.

5. Each time it transfers one byte, the DTU increments DTARA. When the transfer reaches the boundary, a boundary overflow occurs and the MREF flag is set to 1. If the EMRI bit is set to 1, an interrupt is requested.

Steps 1 to 5 can be carried out repeatedly.

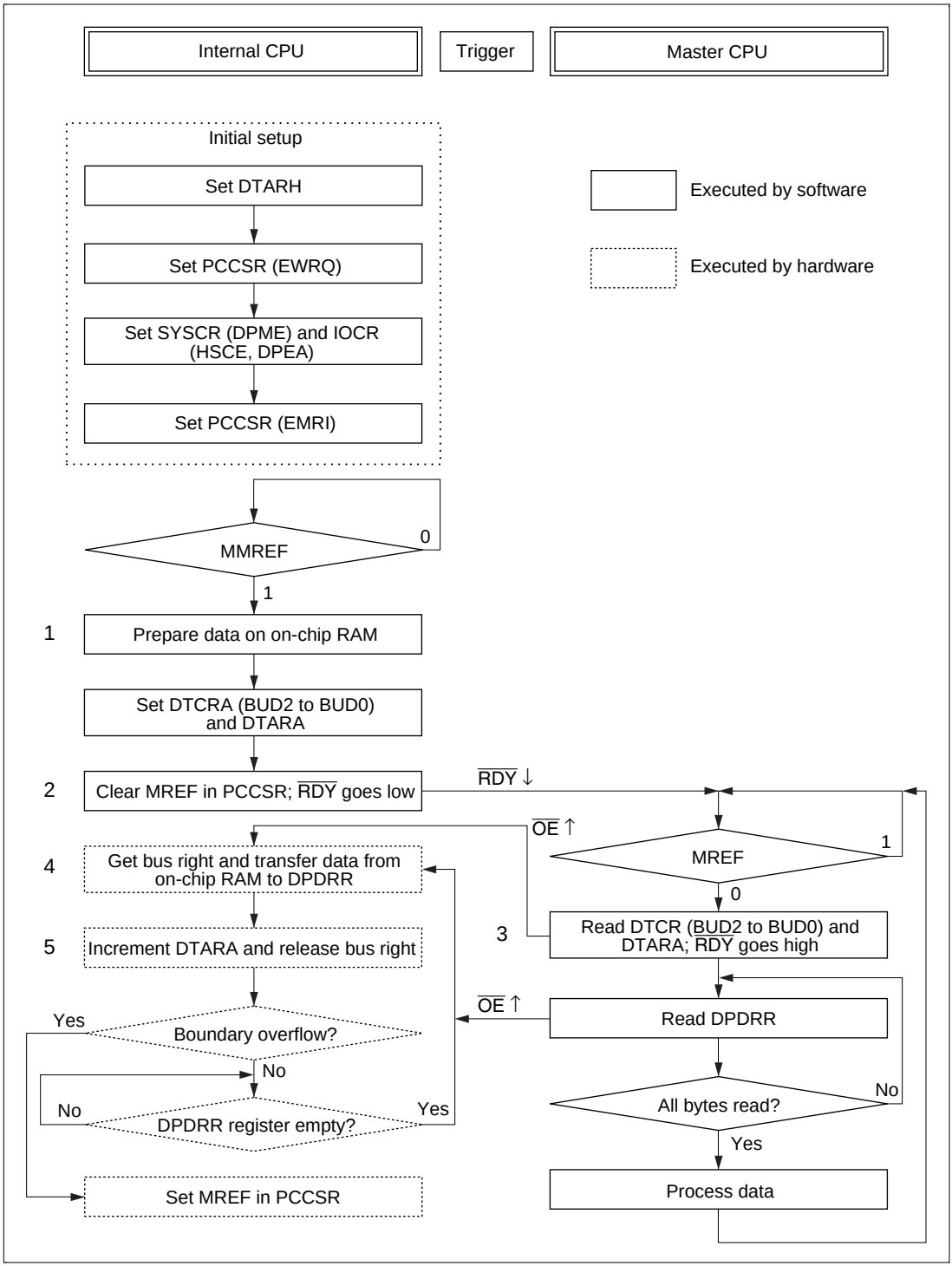


Figure 5.17 Flowchart of Read Operations in DPRAM Bound Buffer Mode

Write Procedure for DPRAM Bound Buffer Mode: DPRAM bound buffer mode provides a parallel interface with sequential write access to a maximum 128 bytes of on-chip RAM, using PBI transfers executed by DTU channel B. Write operations in DPRAM bound buffer mode are depicted in block-diagram form in figure 5.18.

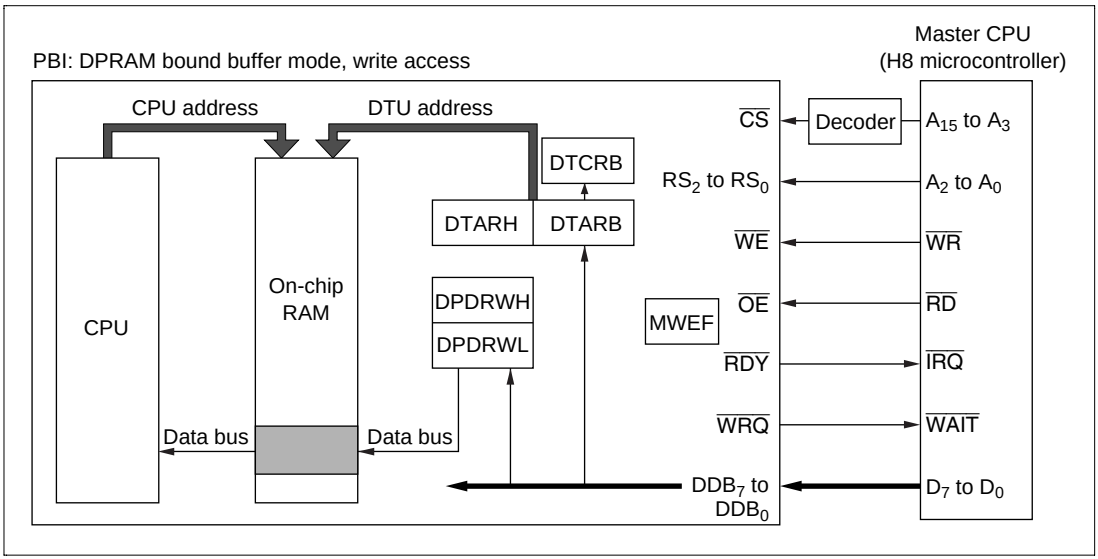


Figure 5.18 Write Access in DPRAM Bound Buffer Mode

The write procedure for DPRAM bound buffer mode is given below. (See figure 5.19.)

1. The internal CPU processes the data transferred to on-chip RAM, then clears the MWEF flag in PCCSR to notify the master CPU that it is ready for the next transfer. If the EWAKAR bit is set to 1 in PCCSR, a low $\overline{\text{RDY}}$ signal is output.

2. The master CPU learns from the state of the $\overline{\text{RDY}}$ signal or MWEF flag that transfer preparations have been completed, prepares data, and writes the top address in DTARB and the boundary of a destination area in on-chip RAM in DTCRB. The $\overline{\text{RDY}}$ output goes high. The master CPU writes all the data it has prepared to the DPDRW registers.

3. Each time the master CPU writes to the DPDRW registers, the data goes into a DPDRW register, which thereby becomes full. The PBI generates a transfer request for DTU channel B. The DTU transfers the data from the full DPDRW register to the on-chip RAM address specified by DTARH and DTARB. Even and odd addresses in DTARB correspond to DPDRWH and DPDRWL, respectively. Data is transferred in the order in which written. Each time data is written, the PBI requests a transfer on DTU channel B, to keep the DPDRW registers in the empty state.

4. Each time it transfers one byte, the DTU increments DTARB. When the transfer reaches the boundary, a boundary overflow occurs and the MWEF flag is set to 1. If the EMWI bit is set to 1, an interrupt is requested.

Steps 1 to 4 can be carried out repeatedly.

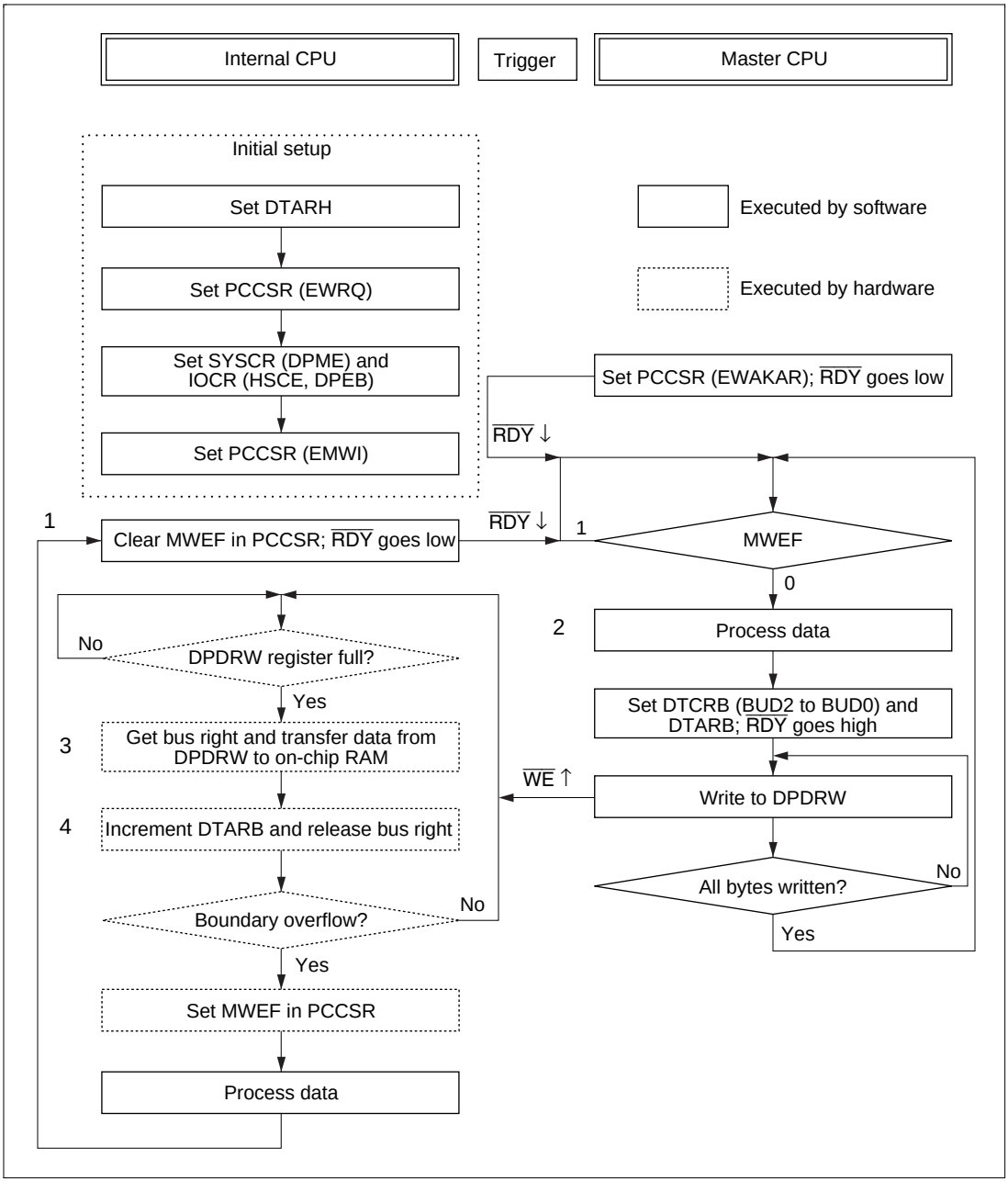


Figure 5.19 Flowchart of Write Operations in DPRAM Bound Buffer Mode

Precautions on Use of DPRAM Bound Buffer Mode: When using DPRAM bound buffer mode, note the following points:

- If the master CPU reads a DPDRR register while it is empty, the $\overline{\text{WRQ}}$ pin will stay at the low output level until the register becomes full. If the master CPU does not use $\overline{\text{WRQ}}$ as a wait request signal, leave sufficient intervals between reads to be sure that the transfer is carried out.
- If the master CPU writes to a DPDRW register while it is full, the $\overline{\text{WRQ}}$ pin will stay at the low output level until the register becomes empty. If the master CPU does not use $\overline{\text{WRQ}}$ as a wait request signal, leave sufficient intervals between writes to be sure that the transfer is carried out.
- MWEF is initially 0, so when the EWAKAR bit is set to 1, the $\overline{\text{RDY}}$ output will go low immediately. Be careful of this at the start of the first transfer.
- The $\overline{\text{RDY}}$ pin is shared by master read and master write in both DPRAM bound buffer mode and DPRAM direct word mode. If the master CPU uses $\overline{\text{RDY}}$ as an interrupt input, when it tries to learn the internal state of the DPRAM, there may be competing interrupt conditions. The master CPU should control the $\overline{\text{RDY}}$ output by using the ERAKAR and EWAKAR bits, or should check the MREF and MWEF flags at the start and end of transfer processing.

5.3.6 Operation in DPRAM Direct Word Mode

DPRAM direct word mode does not use the DTU. Read and write operations in DPRAM direct word mode are depicted in block-diagram form in figure 5.20.

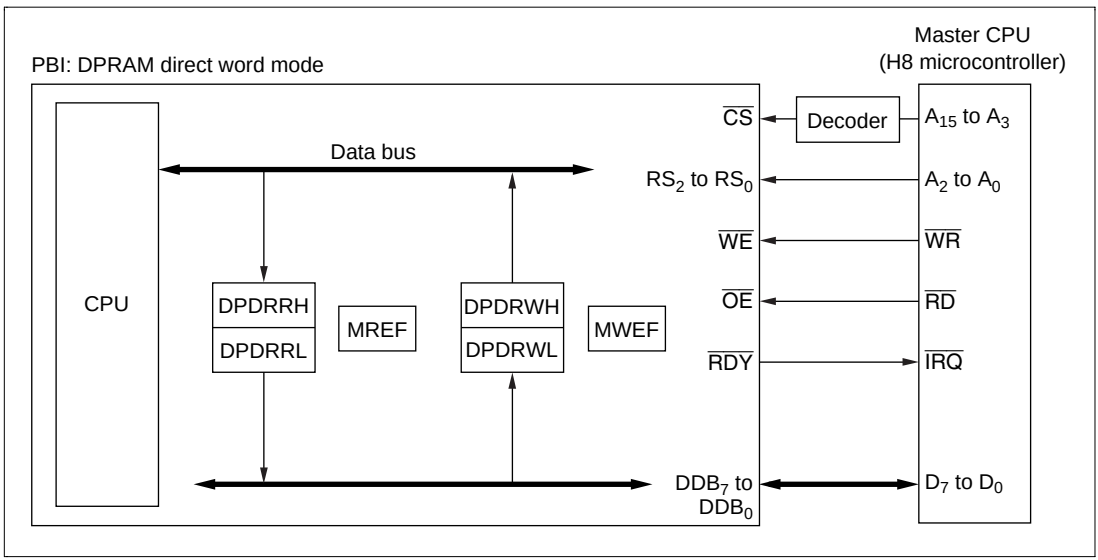


Figure 5.20 Read/Write Access in DPRAM Direct Word Mode

The conditions under which DPRAM direct word mode is available are given below, with initialization and operating procedures.

Conditions under Which DPRAM Direct Word Mode is Available

- DPRAM direct word mode, read access
  Available when DPME = 1, HSCE = 0, and DPEA = 0.

- DPRAM direct word mode, write access
  Available when DPME = 1, HSCE = 0, and DPEB = 0.

Initialization Procedure: Initialize the DTU and PBI by the DTU/PBI initialization procedure.

The $\overline{\text{RDY}}$ pin is also used in DPRAM bound buffer mode. The initial settings must be free of conflict between modes. The initial state of the $\overline{\text{RDY}}$ pin is the high impedance state in single-chip mode, and the high output state in expanded modes.

Read Procedure for DPRAM Direct Word Mode: For read access, DPRAM direct word mode provides a parallel interface with two data-register bytes that can be read by the master CPU and written to by the internal CPU. The read procedure for DPRAM direct word mode is given below. (See figure 5.21.)

1. The internal CPU writes data in DPDRRH and DPDRRL. Writing to DPDRRL clears the MREF flag in PCCSR. If the ERAKAR bit is set to 1 in PCCSR, a low $\overline{\text{RDY}}$ signal is output.

2. From the state of the $\overline{\text{RDY}}$ signal or MREF flag, the master CPU learns that the internal CPU has finished writing data. The master CPU now reads DPDRRH and DPDRRL. Reading DPDRRL sets the MREF flag to 1 in PCCSR. If the EMRI bit is set to 1, an interrupt is requested. The $\overline{\text{RDY}}$ output goes to the high impedance state in single-chip mode, and to the high output state in expanded modes.

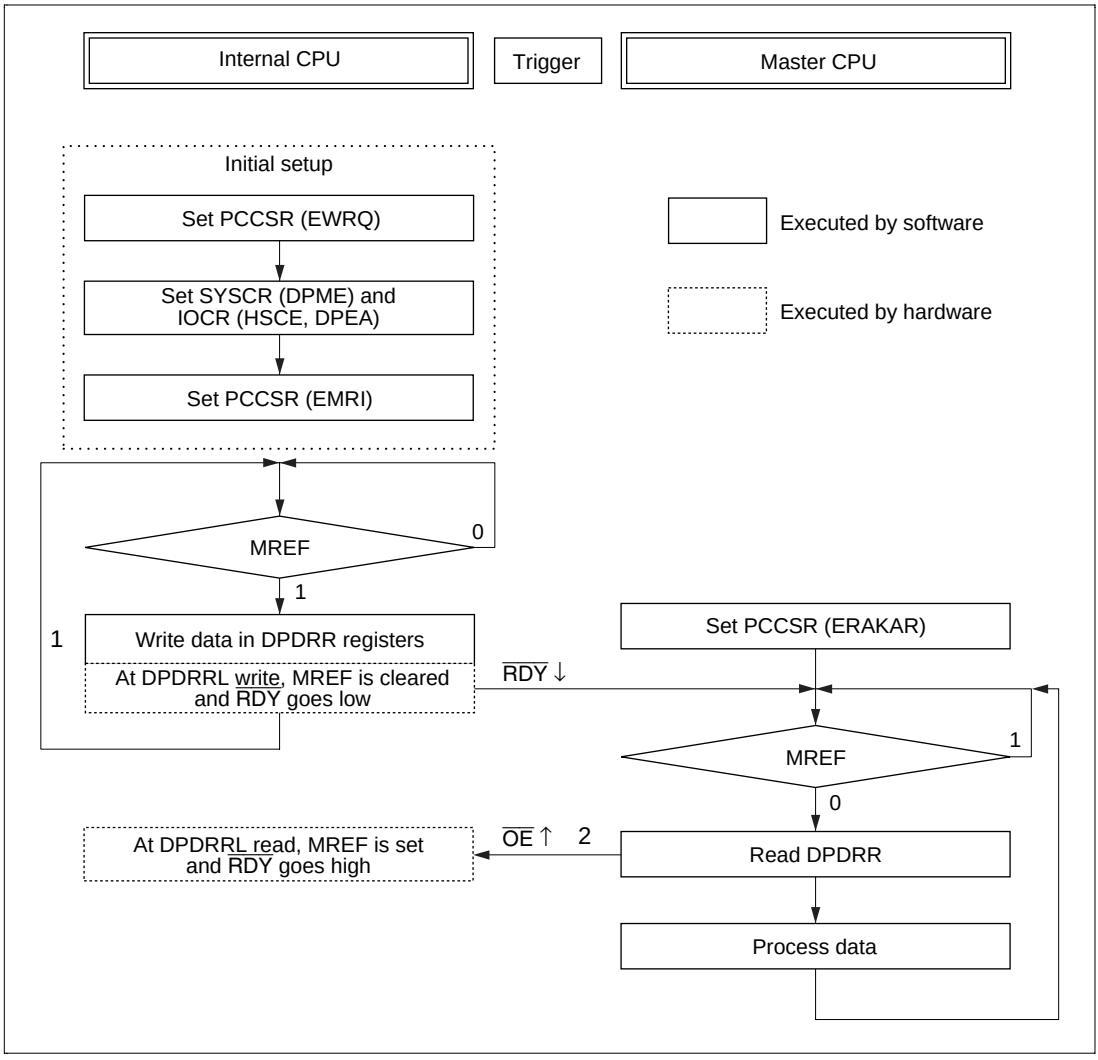


Figure 5.21 Flowchart of Read Operations in DPRAM Direct Word Mode

Write Procedure for DPRAM Direct Word Mode: For write access, DPRAM direct word mode provides a parallel interface with two data-register bytes that can be written to by the master CPU and read by the internal CPU. The write procedure for DPRAM direct word mode is given below. (See figure 5.22.)

1. From the state of the $\overline{\text{RDY}}$ signal or MWEF flag, the master CPU learns that the internal CPU has finished reading, and writes to DPDRWH and DPDRRL. Writing to DPDRWL sets the MWEF flag to 1 in PCCSR. If the EMWI bit is set to 1, an interrupt is requested. The $\overline{\text{RDY}}$ output goes to high impedance state in single-chip mode, and to high output state in expanded modes.

2. The internal CPU reads the data in DPDRWH and DPDRWL. Reading DPDRWL clears the MWEF flag in PCCSR. If the EWAKAR bit is set to 1 in PCCSR, a low $\overline{\text{RDY}}$ signal is output.

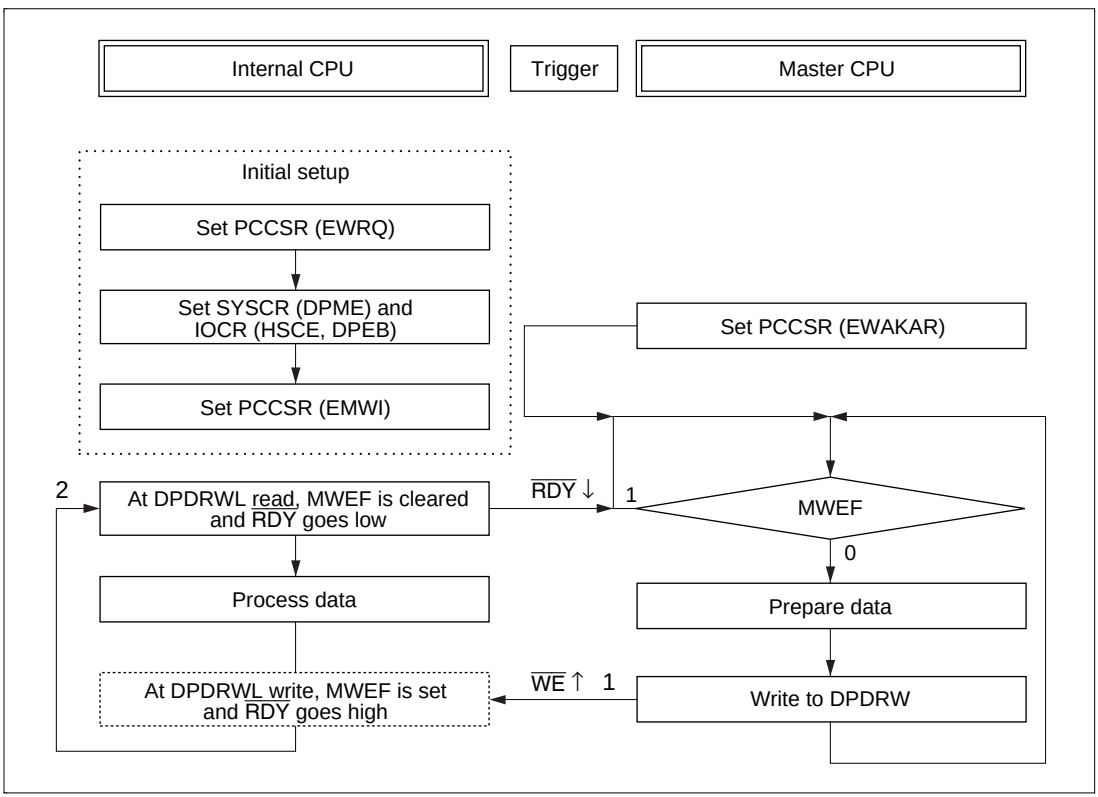


Figure 5.22 Flowchart of Write Operations in DPRAM Direct Word Mode

Precautions on Use of DPRAM Direct Word Mode: When using DPRAM direct word mode, note the following points:

- MWEF is initially 0, so when the EWAKAR bit is set to 1, the $\overline{\text{RDY}}$ output will go low immediately. Be careful of this at the start of the first transfer.
- The $\overline{\text{RDY}}$ pin is shared by master read and master write in both DPRAM bound buffer mode and DPRAM direct word mode. If the master CPU uses $\overline{\text{RDY}}$ as an interrupt input, when it tries to learn the internal state of the DPRAM, there may be competing interrupt conditions. The master CPU should control the $\overline{\text{RDY}}$ output by using the ERAKAR and EWAKAR bits, or should check the MREF and MWEF bits at the start and end of transfer processing.

5.3.7 Operation in Handshake Mode

Handshake mode provides a parallel interface in which data is passed through an 8-bit port one byte at a time. Figure 5.23 depicts handshake mode in block-diagram form. Figure 5.24 shows typical interconnections for handshaking communication.

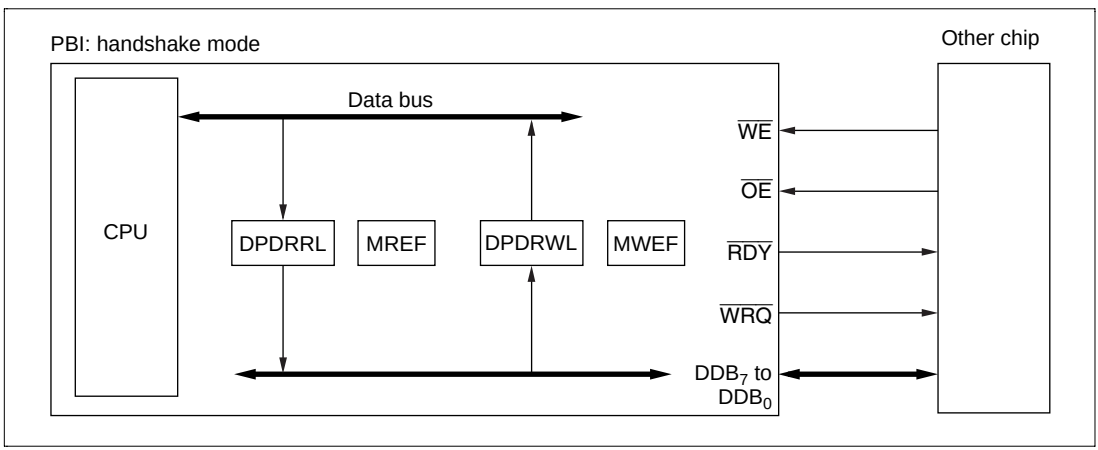


Figure 5.23 Transfers in Handshake Mode

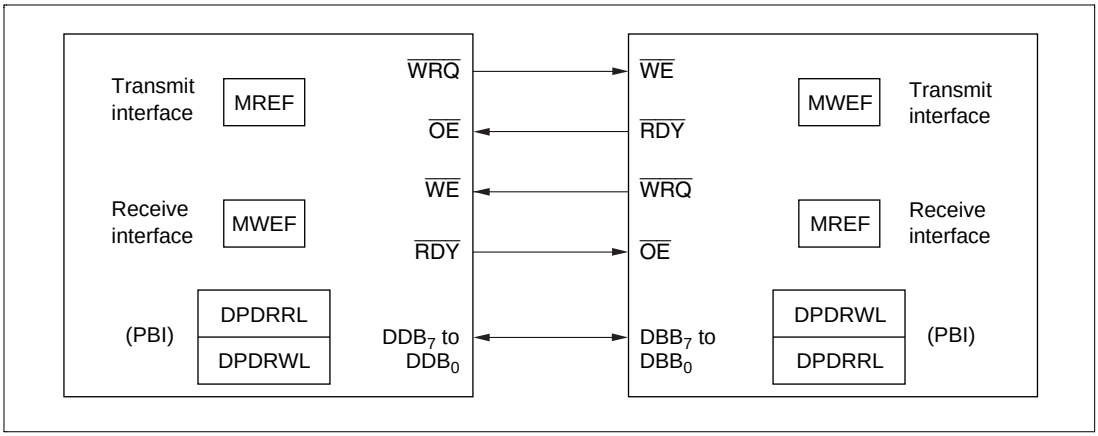


Figure 5.24 Interconnections for Handshake Mode (Example)

The conditions under which handshake mode is available are given below, with initialization, transmit, and receive procedures.

Conditions under Which Handshake Mode is Available, and Initialization Procedure

- Conditions under which handshake mode is available
  Available when DPME = 1 and HSCE = 1.

- Initialization procedure
  Initialize the DTU and PBI by the DTU/PBI initialization procedure.

Transmit Procedure in Handshake Mode: In transmitting in handshake mode, when the data register contents are output, a latch pulse is output from the $\overline{WRQ}$ pin to write the data into the handshake interface circuits in the receiving device. The transmit procedure for handshake mode is given below. (See figure 5.25.)

1. The internal CPU reads the MREF flag, checks that it is set to 1, and writes data in DPDRRL. The PBI then clears the MREF flag to 0 and outputs a low signal from the $\overline{WRQ}$ pin.

2. The PBI checks the state of the $\overline{OE}$ pin and outputs the DPDRRL contents if $\overline{OE}$ is low. After eight system clocks, the PBI automatically drives the $\overline{WRQ}$ output to the high level.

3. The PBI checks the state of the $\overline{OE}$ pin. When $\overline{OE}$ goes high, the PBI stops data output, places the data lines in the high-impedance state, and sets the MREF flag to 1.

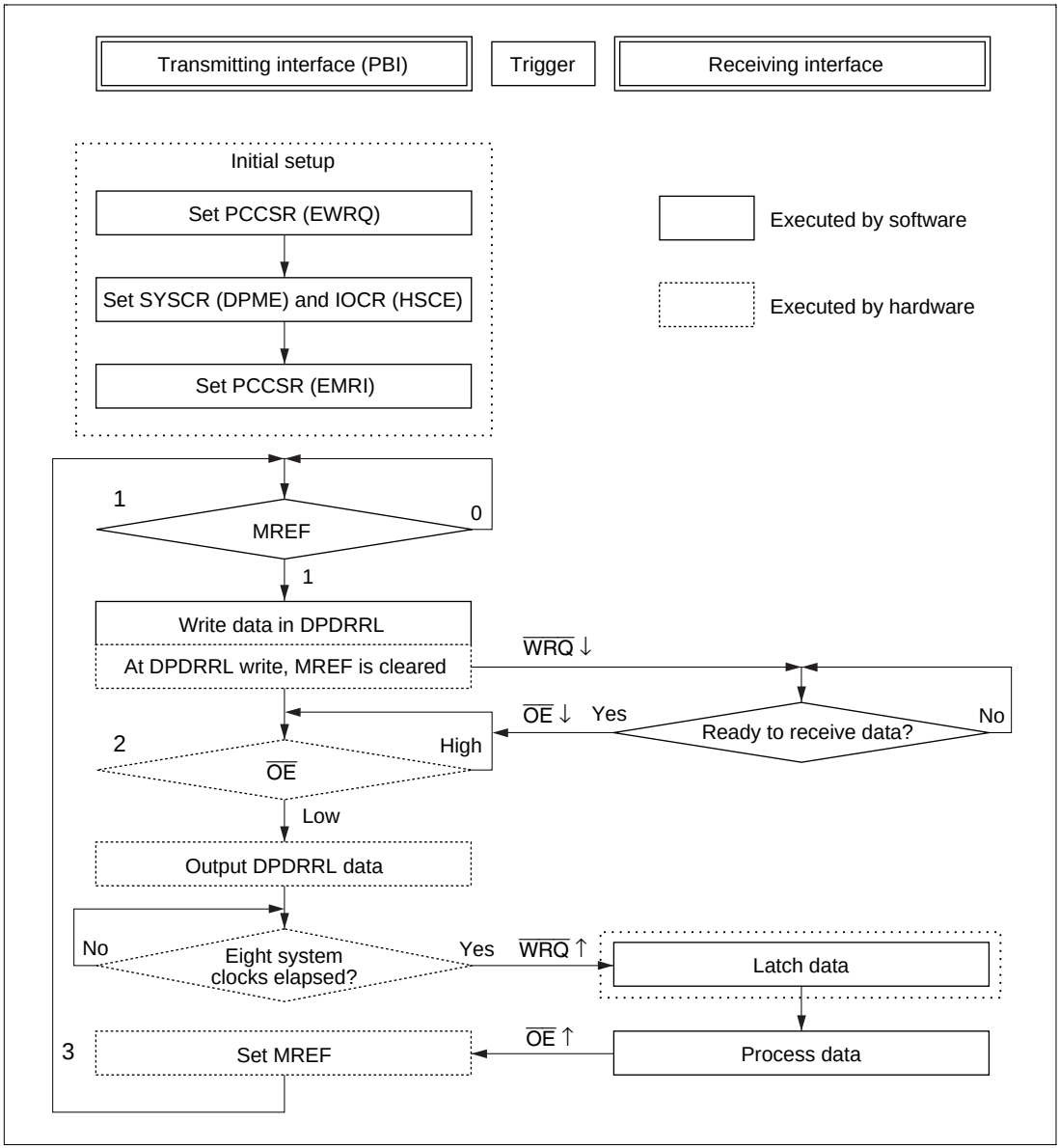


Figure 5.25 Flowchart of Transmit Operations in Handshake Mode

Receive Procedure in Handshake Mode: In receiving in handshake mode, data input from the handshake interface in the transmitting device is read into the data register on a latch pulse input from the $\overline{WE}$ pin. (See figure 5.26.)

1. The PBI checks the states of the $\overline{OE}$ and $\overline{WE}$ pins. If a high-to-low transition of $\overline{WE}$ occurs while $\overline{OE}$ is high, a low signal is output from the $\overline{RDY}$ pin.

2. When $\overline{WE}$ goes high, the PBI latches the data into DPDRWL on the rising edge of $\overline{WE}$ and sets the MWEF bit to 1.

3. The internal CPU reads MWEF, checks that it is set to 1, and reads DPDRWL. The PBI then clears MWEF to 0 and drives the $\overline{RDY}$ output to the high level.

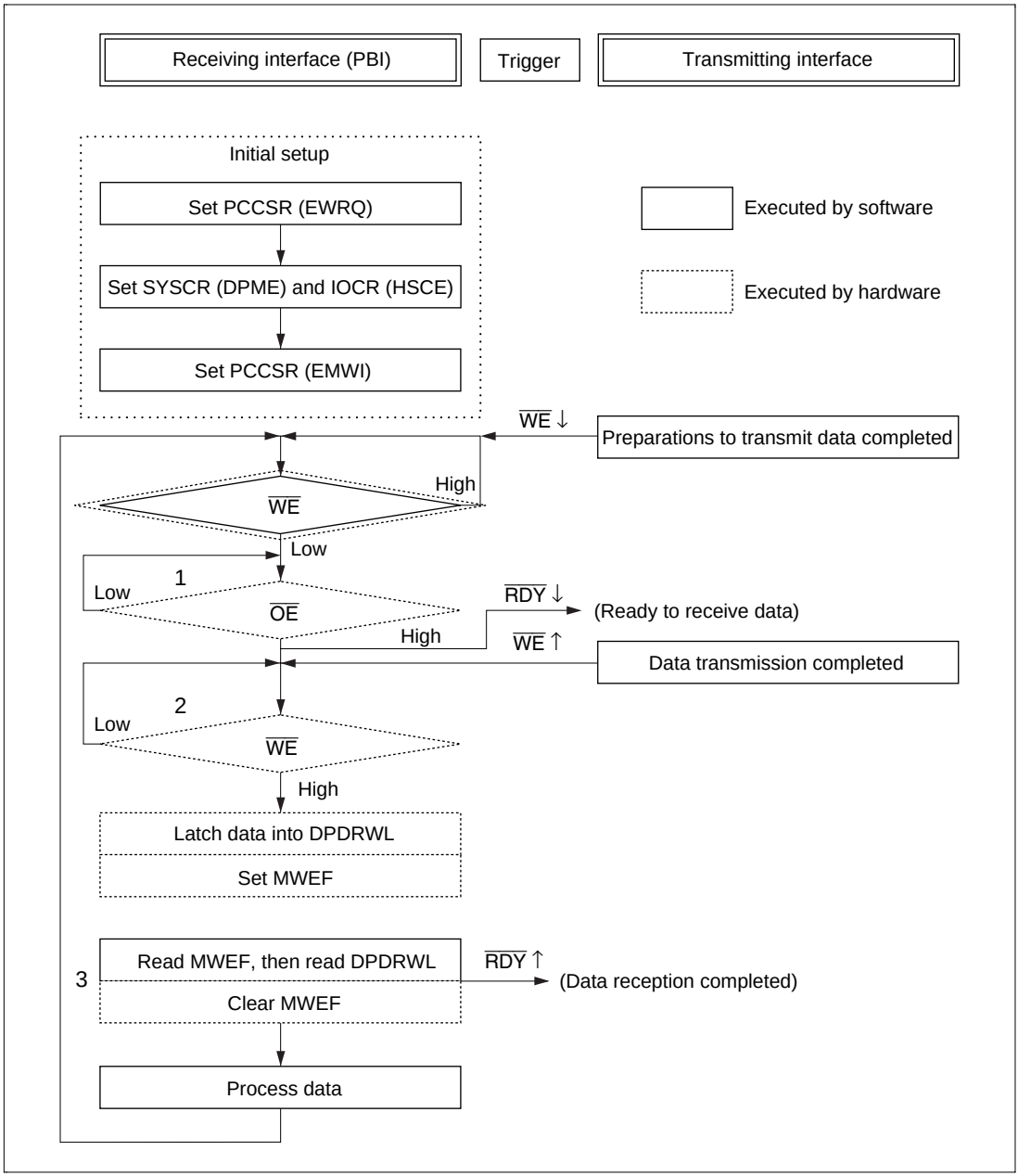


Figure 5.26 Flowchart of Receive Operations in Handshake Mode

Precautions on Use of Handshake Mode: When using handshake mode, note the following points.

- In handshake mode, it is possible to transmit and receive simultaneously, but care is required because the data bus is shared.

  Figures 5.27 and 5.28 (transmit) and 5.29 and 5.30 (receive) show typical interconnections for simultaneous transmitting and receiving with a device having an identical handshake mode. If both ends of the bidirectional $\overline{WRQ}$ line are brought from high to low at the same time, both $\overline{OE}$ pins will go high simultaneously and the data will collide. To avoid this, both devices should use a protocol in which they transmit and receive in turn, so that transmit requests will not occur simultaneously.

- If the interface is used only for transmitting, the input at the $\overline{WE}$ pin should be held at the high level. For receive-only usage, the $\overline{OE}$ input should be held at the high level and the EWRQ bit should be cleared to 0 in PCCSR, since the $\overline{WRQ}$ output is not needed.

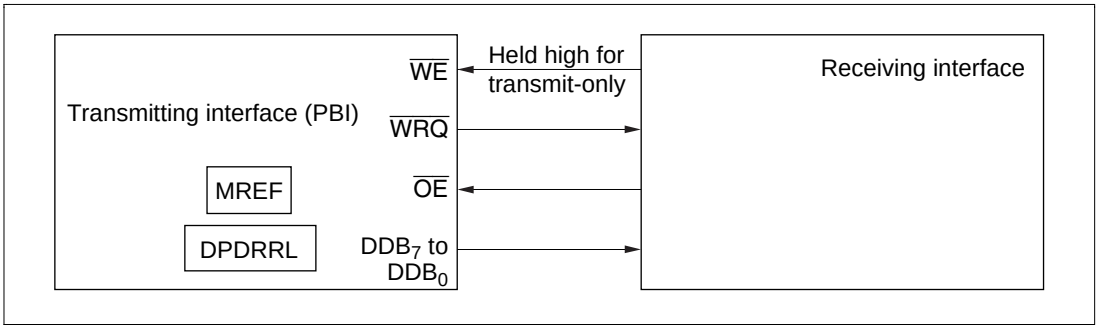


Figure 5.27 Transmitting in Handshake Mode

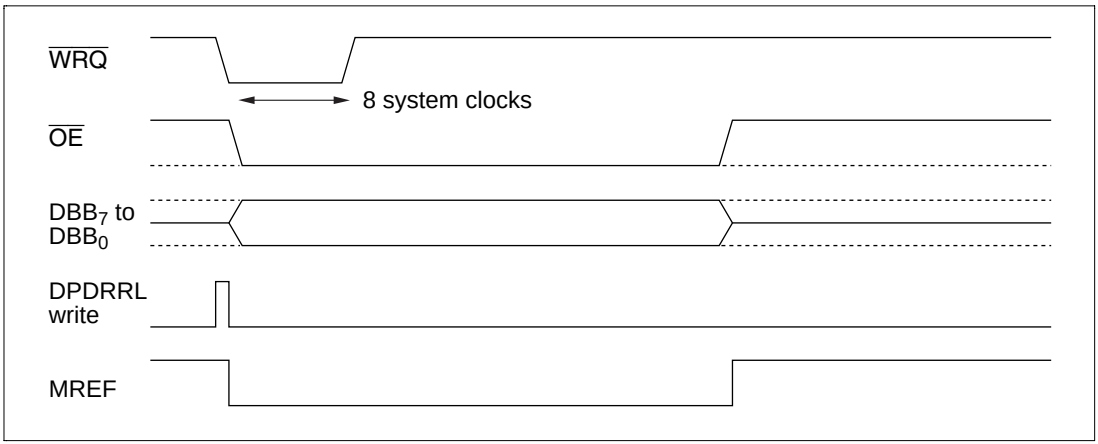


Figure 5.28 Transmit Timing in Handshake Mode

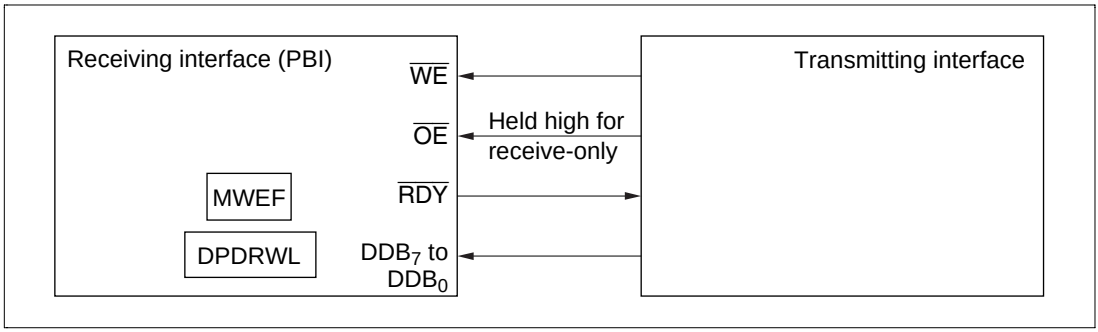


Figure 5.29 Receiving in Handshake Mode

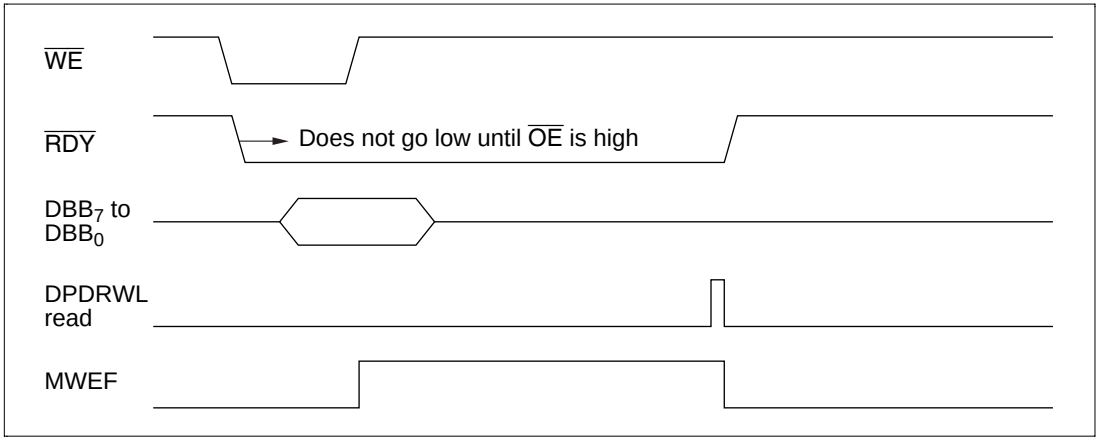


Figure 5.30 Receive Timing in Handshake Mode

5.4 Application Notes

5.4.1 DTU and PBI Processing Time

PBI transfers due to buffer queries in DPRAM mode and some PBI transfers in DPRAM bound buffer mode are initiated by the rising edge of the $\overline{WE}$ or $\overline{OE}$ input signal when the PBI is accessed. The master CPU may receive a wait request from the $\overline{WRQ}$ pin if it accesses the PBI again after the edge is detected but before the PBI transfer is completed.

The way to avoid this situation is to know how much time DTU and PBI processing takes, and allow sufficient intervals between PBI accesses. Examples are given below. In all these examples the master CPU is an H8 microcontroller.

(1) When the master CPU accesses a PBI data register (DPDRRQ, DPDRRH/L, DPDRWH/L), the DTU is activated. The time until the same data register can be accessed again is the sum of the following three periods:

- Transfer request detection lag: time lag from the detection of a rising edge at the $\overline{WE}$ or $\overline{OE}$ pin until the transfer request occurs
- Bus cycle waiting time (maximum 6 states + wait states): time until the end of the bus cycle being executed when the transfer request occurs
- DPRAM cycle (2 states + wait states): time for executing the bus cycle that transfers data between on-chip RAM and the PBI

The transfer request detection lag depends on the operating mode. In DPRAM bound buffer mode, it is a maximum of 3.5 states, and in buffer query in DPRAM mode, it is a maximum of 3 states. If external memory is used instead of on-chip memory, and if the external memory is accessed with wait states, the maximum value of the bus cycle waiting time depends on the number of wait states.

(2) In DPRAM bound buffer mode, the master CPU may access the PBI by byte access or word access. In word access, the minimum detectable interval (strobe interval) between the rising edges of $\overline{OE}$ and $\overline{WE}$ of two bytes is two states.

(3) If input pins RS_2 to RS_0 specify an unusable DPRAM data register and if pin $\overline{CS}$ is low, then pin $\overline{WRQ}$ also goes low. When pin $\overline{WRQ}$ is not used and the access interval is the time between the rising edge of $\overline{OE}$ and $\overline{WE}$ and the next confirmation of RS_2 to RS_0 and $\overline{CS}$ input, the following intervals become necessary.

- Bound buffer mode access interval: 9.5 states (see figure 5.31)
- Buffer query address write to data read access interval: 11 states (see figure 5.32)

The above are consecutive accesses with the same channel. If different channels are used simultaneously, the PBI executes in order of the priority sequence of the channels. For accesses

with a low priority channel, the DPRAM cycle time executed first must be added to the access time.

(4) Allowances for the times described in (1) to (3) above should be made with the master CPU operating frequency and AC characteristics taken into consideration. To make allowances for the strobe interval in a 3-state no-wait master CPU access, a master CPU with an operating frequency up to 1.5 times that of the interval CPU can be connected. In this case, the access intervals in (3) above become 14.25 states and 16.5 states of the master CPU. However, the output delay from the master CPU and the PBI input setup time for $\overline{OE}$, $\overline{WE}$, $\overline{CS}$, and RS are different for each signal, and so adequate allowances must be made for these differences.

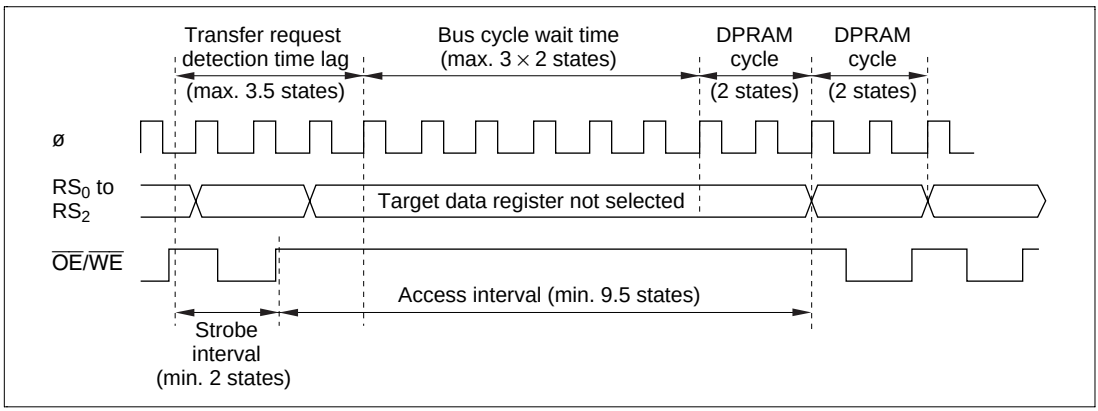


Figure 5.31 Bound Buffer Mode Access Interval

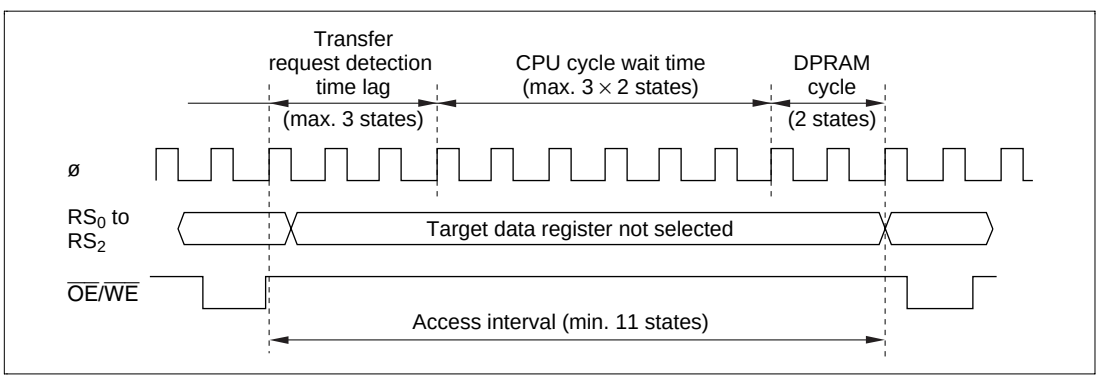


Figure 5.32 Buffer Query Mode Address Write to Data Read Access Interval

Section 6 Wait-State Controller

6.1 Overview

The H8/3318 has an on-chip wait-state controller that enables insertion of wait states into bus cycles for interfacing to low-speed external devices.

6.1.1 Features

Features of the wait-state controller are listed below.

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

6.1.2 Block Diagram

Figure 6.1 shows a block diagram of the wait-state controller.

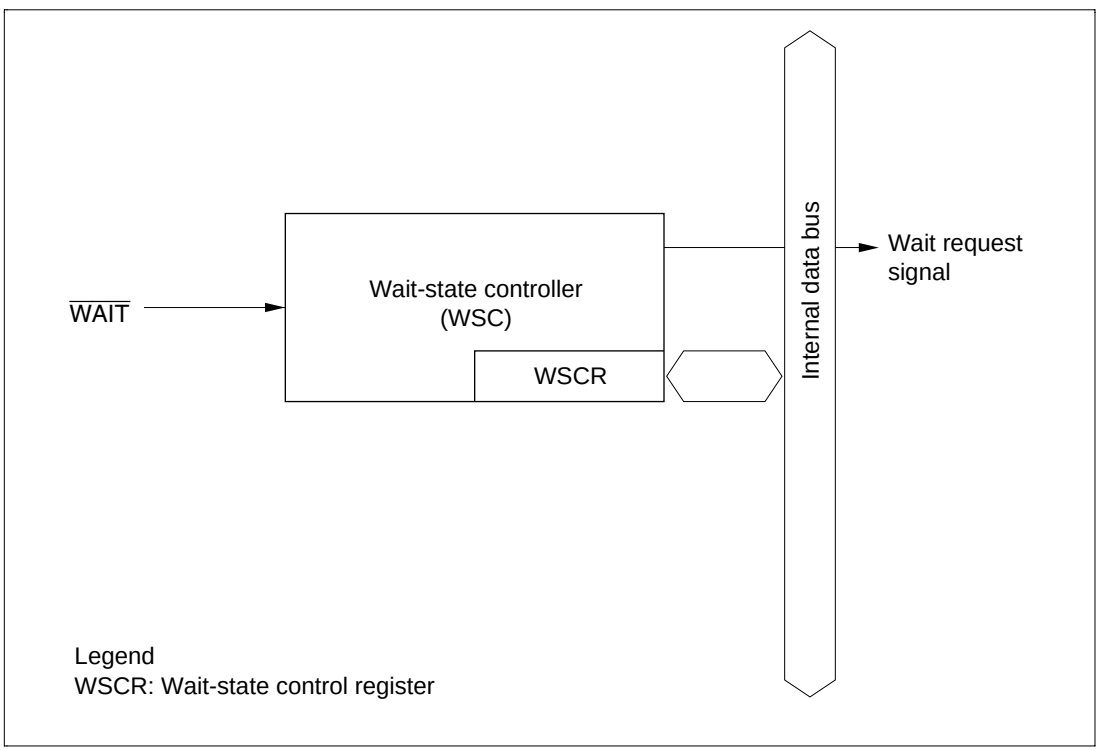


Figure 6.1 Block Diagram of Wait-State Controller

6.1.3 Input/Output Pins

Table 6.1 summarizes the wait-state controller's input pin.

Table 6.1 Wait-State Controller Pins

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

6.1.4 Register Configuration

Table 6.2 summarizes the wait-state controller's register.

Table 6.2 Register Configuration

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

6.2 Register Description

6.2.1 Wait-State Control Register (WSCR)

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

Bit	7	6	5	4	3	2	1	0
	—	—	CKDBL	—	WMS1	WMS0	WC1	WC0
Initial value	1	1	0	1	1	0	0	0
Read/Write	—	—	R/W	—	R/W	R/W	R/W	R/W

WSCR is initialized to H'C8 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 and 6—Reserved: These bits cannot be modified and are always read as 1.

Bit 5—Clock Double (CKDBL): Controls frequency division of clock signals supplied to supporting modules. For details, see section 7, Clock Pulse Generator.

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

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

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

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

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

6.3 Wait Modes

Programmable Wait Mode: The number of wait states (T_W) selected by bits WC1 and WC0 are inserted in all accesses to external addresses. Figure 6.2 shows the timing when the wait count is 1 (WC1 = 0, WC0 = 1).

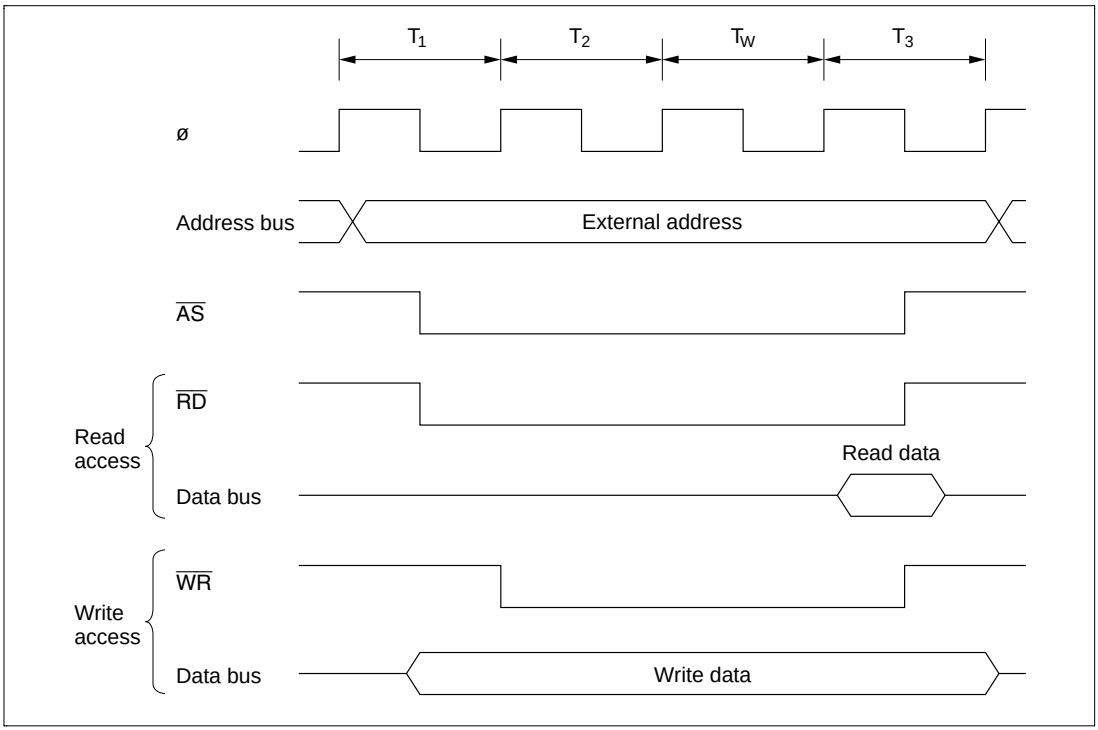


Figure 6.2 Programmable Wait Mode

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

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

Figure 6.3 shows the timing when the wait count is 1 (WC1 = 0, WC0 = 1) and one additional wait state is inserted by $\overline{\text{WAIT}}$ input.

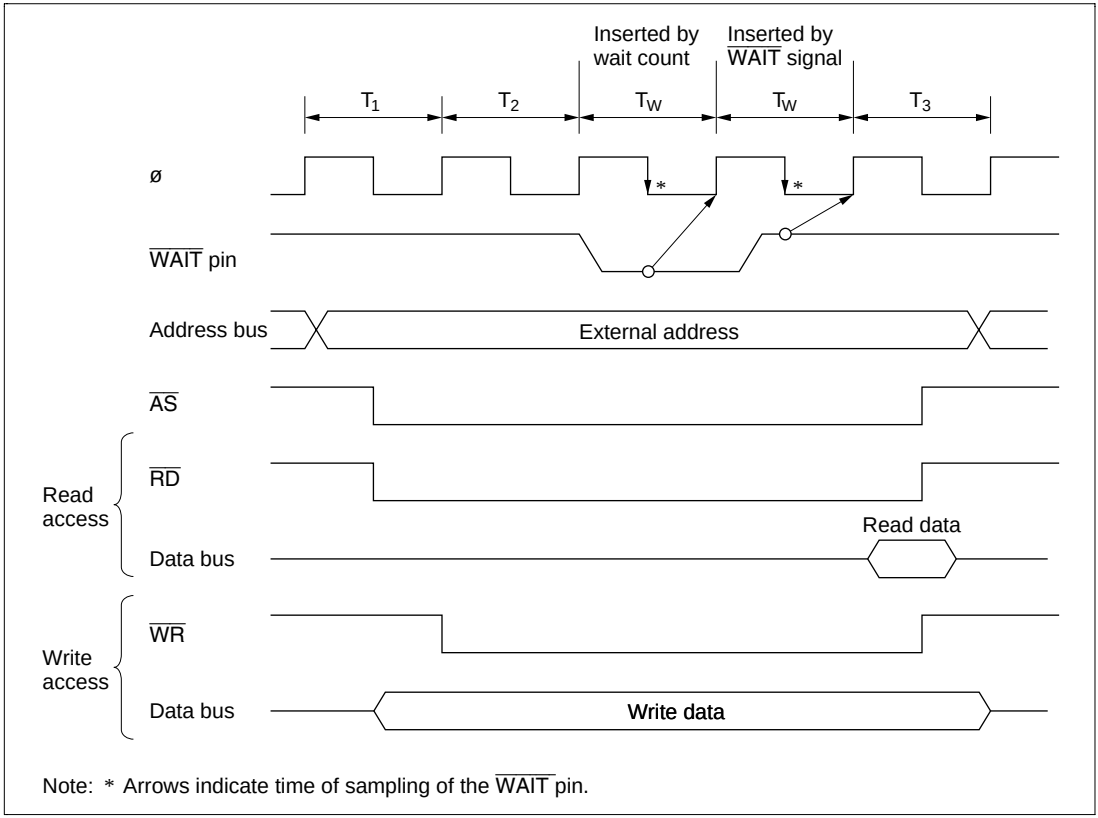


Figure 6.3 Pin Wait Mode

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

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

Figure 6.4 shows the timing when the wait count is 1.

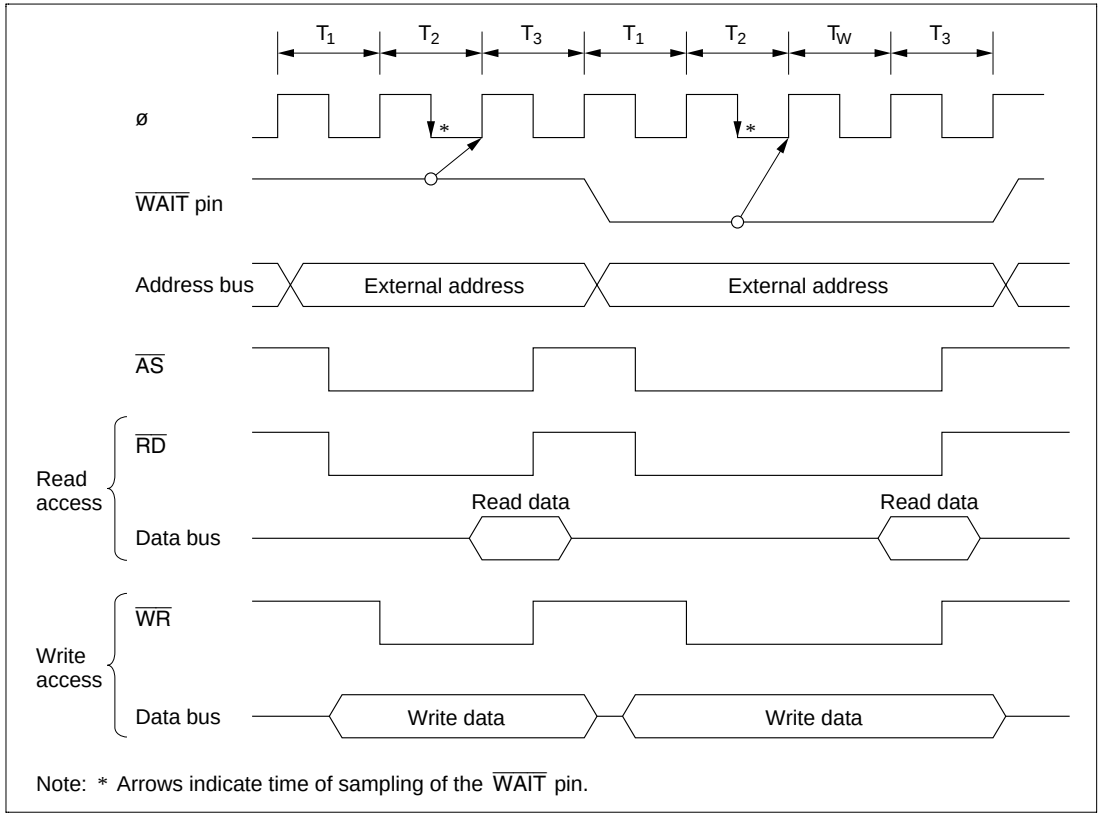


Figure 6.4 Pin Auto-Wait Mode

Section 7 Clock Pulse Generator

7.1 Overview

The H8/3318 has a built-in clock pulse generator (CPG) consisting of an oscillator circuit, a duty adjustment circuit, and a divider and a prescaler that generates clock signals for the on-chip supporting modules.

7.1.1 Block Diagram

Figure 7.1 shows a block diagram of the clock pulse generator.

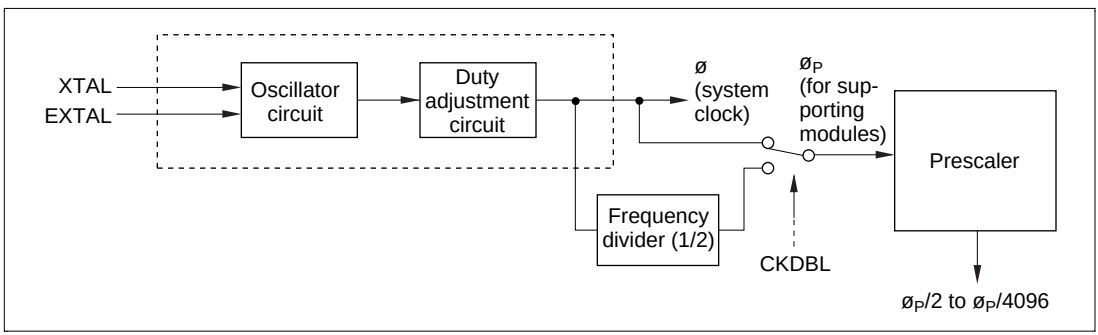


Figure 7.1 Block Diagram of Clock Pulse Generator

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

7.1.2 Wait-State Control Register (WSCR)

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

WSCR is initialized to H'C8 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit	7	6	5	4	3	2	1	0
	—	—	CKDBL	—	WMS1	WMS0	WC1	WC0
Initial value	1	1	0	1	1	0	0	0
Read/Write	—	—	R/W	—	R/W	R/W	R/W	R/W

Bits 7 and 6—Reserved: These bits cannot be modified and are always read as 1.

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

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

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

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

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

These bits control wait-state insertion. For details, see section 6, Wait-State Controller.

7.2 Oscillator Circuit

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

7.2.1 Connecting an External Crystal

Circuit Configuration: An external crystal can be connected as in the example in figure 7.2. Table 7.1 indicates the appropriate damping resistance Rd. An AT-cut parallel resonance crystal should be used.

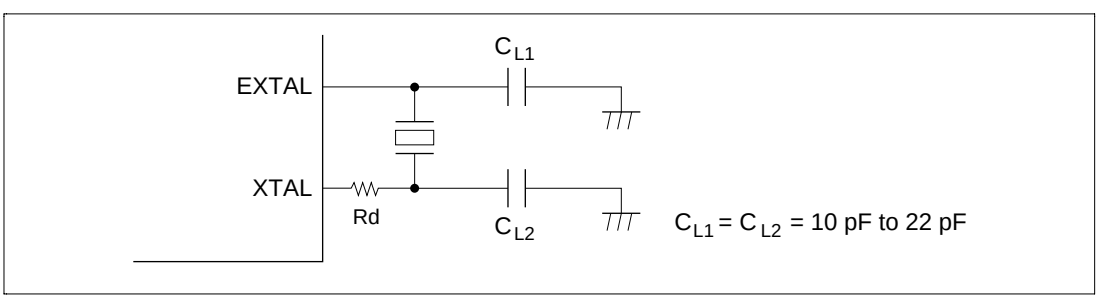


Figure 7.2 Connection of Crystal Oscillator (Example)

Table 7.1 Damping Resistance

Frequency (MHz)	2	4	8	10	12	16
Rd (Ω)	1 k	500	200	0	0	0

Crystal Oscillator: Figure 7.3 shows an equivalent circuit of the crystal resonator. The crystal resonator should have the characteristics listed in table 7.2.

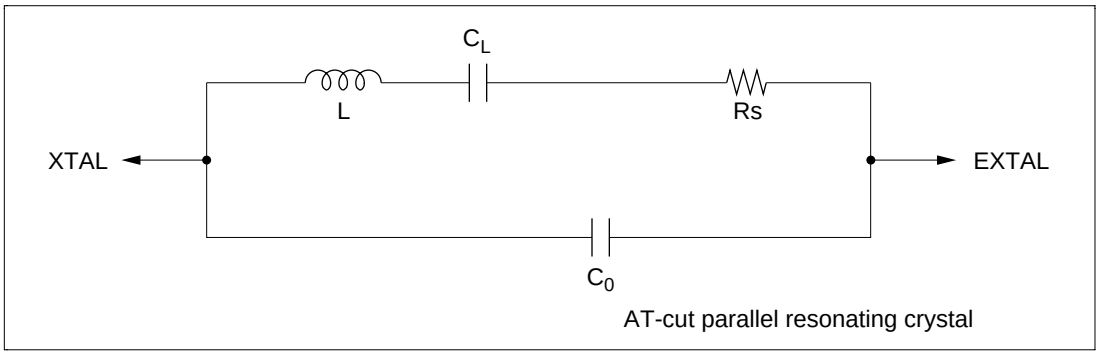


Figure 7.3 Equivalent Circuit of External Crystal

Table 7.2 External Crystal Parameters

Frequency (MHz)	2	4	8	10	12	16
Rd max (Ω)	500	120	80	70	60	50
C_0 (pF)	7 pF max	7 pF max	7 pF max	7 pF max	7 pF max	7 pF max

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

Note on Board Design: When an external crystal is connected, other signal lines should be kept away from the crystal circuit to prevent induction from interfering with correct oscillation. See figure 7.4. The crystal and its load capacitors should be placed as close as possible to the XTAL and EXTAL pins.

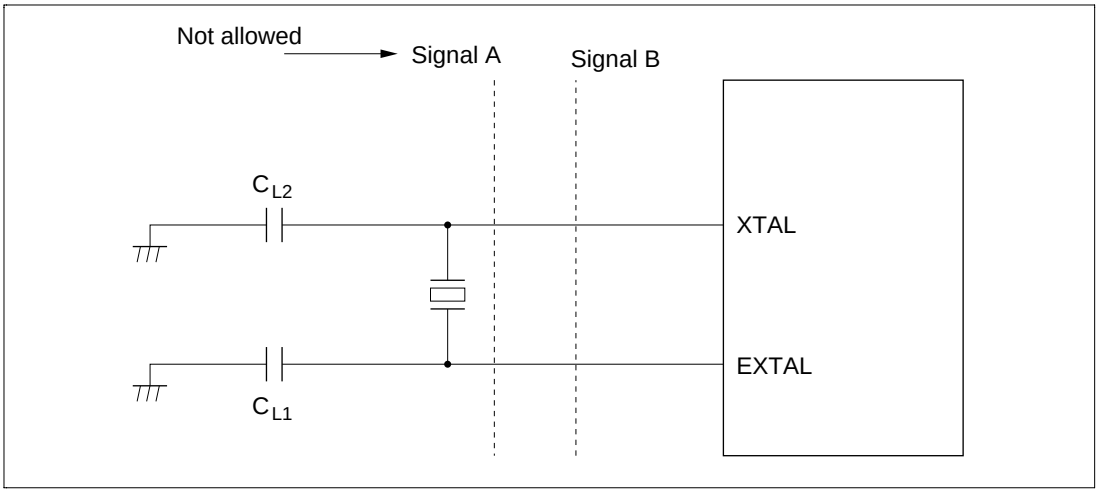


Figure 7.4 Notes on Board Design around External Crystal

7.2.2 Input of External Clock Signal

Circuit Configuration: An external clock signal can be input as shown in the examples in figure 7.5. In example (b) in figure 7.5, the external clock signal should be kept high during standby.

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

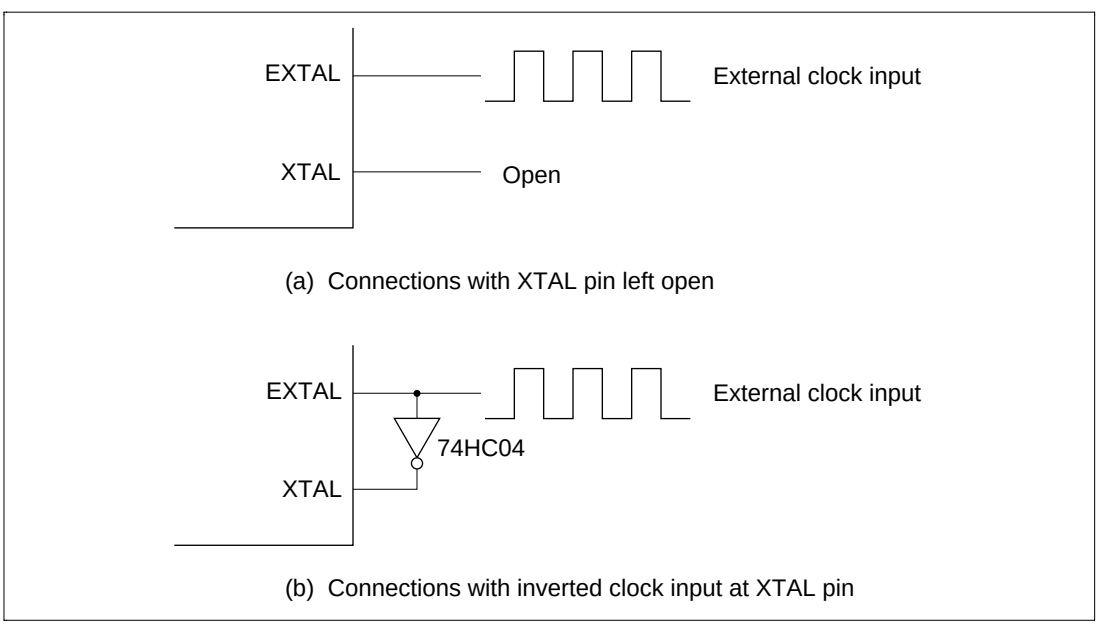


Figure 7.5 External Clock Input (Example)

External Clock Input: The external clock signal should have the same frequency as the desired system clock (ø). Clock timing parameters are given in table 7.3 and figure 7.6.

Table 7.3 Clock Timing

		V_{cc} = 2.7 to 5.5 V		V_{cc} = 4.0 to 5.5 V		V_{cc} = 5.0 V ±10%				
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Test Conditions**	
Low pulse width of external clock input	t_{EXL}	40	—	30	—	20	—	ns	Figure 7.6	
High pulse width of external clock input	t_{EXH}	40	—	30	—	20	—	ns		
External clock rise time	t_{EXr}	—	10	—	10	—	5	ns		
External clock fall time	t_{EXf}	—	10	—	10	—	5	ns		
Clock pulse width low	t_{CL}	0.3	0.7	0.3	0.7	0.3	0.7	t_{cyc}	ø ≥ 5 MHz	Figure 19.4
		0.4	0.6	0.4	0.6	0.4	0.6	t_{cyc}	ø < 5 MHz	
Clock pulse width high	t_{CH}	0.3	0.7	0.3	0.7	0.3	0.7	t_{cyc}	ø ≥ 5 MHz	
		0.4	0.6	0.4	0.6	0.4	0.6	t_{cyc}	ø < 5 MHz	

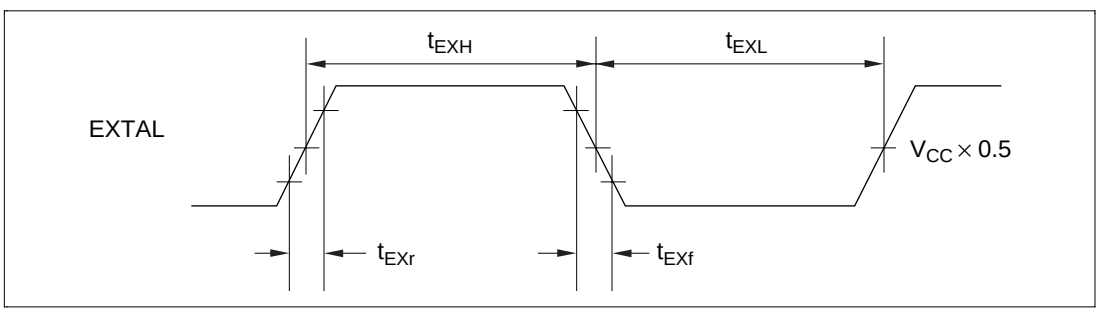


Figure 7.6 External Clock Input Timing

Table 7.4 lists the external clock output stabilization delay time. Figure 7.7 shows the timing for the external clock output stabilization delay time. The oscillator and duty correction circuit have the function of regulating the waveform of the external clock input to the EXTAL pin. When the specified clock signal is input to the EXTAL pin, internal clock signal output is confirmed after the elapse of the external clock output stabilization delay time (t_{DEXT}). As clock signal output is not confirmed during the t_{DEXT} period, the reset signal should be driven low and the reset state maintained during this time.

Table 7.4 External Clock Output Stabilization Delay Time

Conditions: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V

Item	Symbol	Min	Max	Unit	Notes
External clock output stabilization delay time	t_{DEXT}*	500	—	µs	Figure 7.7

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

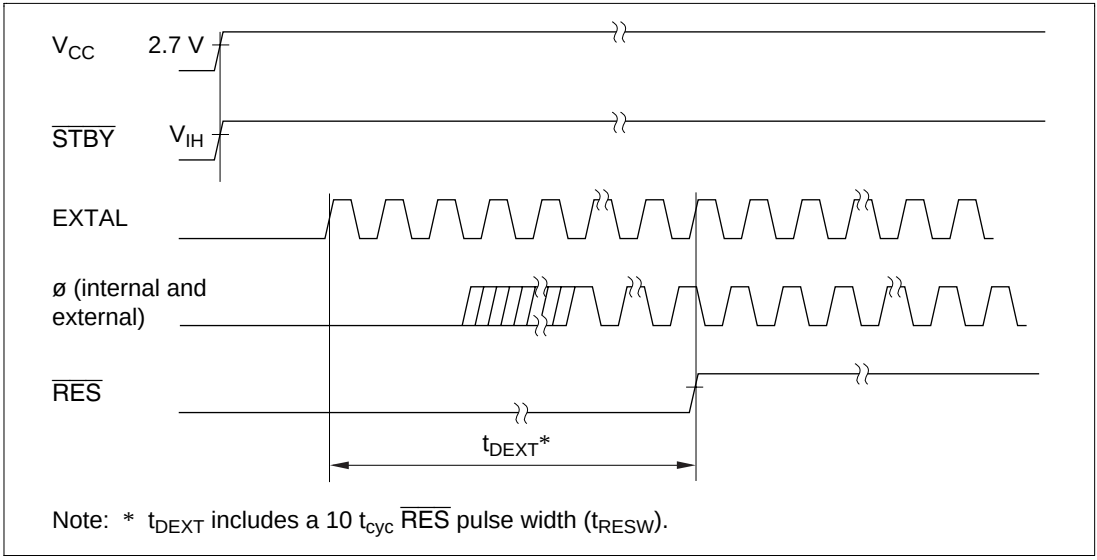


Figure 7.7 External Clock Output Stabilization Delay Time

7.3 Duty Adjustment Circuit

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

7.4 Prescaler

The 1/2 frequency divider generates an on-chip supporting module clock ($ø_P$) from the system clock (ø) according to the setting of the CKDBL bit. The prescaler divides the frequency of $ø_P$ to generate internal clock signals with frequencies from $ø_P/2$ to $ø_P/4096$.

Section 8 I/O Ports

8.1 Overview

The H8/3318 has six 8-bit input/output ports, one 7-bit input/output port, one 3-bit input/output port, and one 8-bit input port.

Table 8.1 lists the functions of each port in each operating mode. As table 8.1 indicates, the port functions are multiplexed with other functions, and the pin functions differ depending on the operating mode.

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

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

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

Table 8.1 Port Functions

Port	Description	Pins	Expanded Modes		Single-Chip Mode	
					Mode 3	
			Mode 1	Mode 2	PBI Disabled (DPME = 0)	PBI Enabled (DPME = 1)
Port 1	• 8-bit I/O port • Input pull-ups • Can drive LEDs	$P1_7/A_7/TP_7$ to $P1_0/A_0/TP_0$	A_7 to A_0 Address output (lower)	$P1_7/A_7$ to $P1_0/A_0$ General input when DDR = 0 (initial value) Address output (lower) when DDR = 1 and NDER = 0 TPC output when DDR = 1 and NDER = 1	$P1_7/TP_7$ to $P1_0/TP_0$ General input when DDR = 0 (initial value) General output when DDR = 1 and NDER = 0 TPC output when DDR = 1 and NDER = 1	
Port 2	• 8-bit I/O port • Input pull-ups • Can drive LEDs	$P2_7/A_{15}/TP_{15}$ to $P2_0/A_8/TP_8$	A_{15} to A_8 Address output (upper)	$P2_7/A_{15}$ to $P2_0/A_8$ General input when DDR = 0 and NDER = 0 (initial value) Address output (upper) when DDR = 1 and NDER = 1 TPC output when DDR = 1 and NDER = 1	$P2_7/TP_{15}$ to $P2_0/TP_8$ General input when DDR = 0 (initial value) General output when DDR = 1 and NDER = 0 TPC output when DDR = 1 and NDER = 1	
Port 3	• 8-bit I/O port • Input pull-ups	$P3_7/D_7/DDB_7$ to $P3_0/D_0/DDB_0$	D_7 to D_0 data bus		$P3_7$ to $P3_0$ General input/output	DDB_7 to DDB_0 DPRAM data bus
Port 4	• 8-bit I/O port	$P4_7/FTOB_1/XDDB_7$, $P4_6/FTOA_1/XDDB_6$	$P4_7/FTOB_1/XDDB_7$ to $P4_0/TMCI_0/XDDB_0$ General input/output or timer input/output (8-bit timer 0/1, input/output (8-bit timer 0/1, 16-bit free-running timer 1) ($TMCI_0$, TMO_0, $TMRI_0$, $TMCI_1$, TMO_1, $TMRI_1$, $FTOA_1$, $FTOB_1$) when PBI is disabled (DPME = 0) DPRAM data bus when PBI is enabled (DPME = 1)		$P4_7/FTOB_1$, $P4_6/FTOA_1$ Free-running timer 1 output ($FTOA_1$, $FTOB_1$) or general input/output	
		$P4_5/TMRI_1/XDDB_5$, $P4_4/TMO_1/XDDB_4$, $P4_3/TMCI_1/XDDB_3$, $P4_2/TMRI_0/XDDB_2$, $P4_1/TMO_0/XDDB_1$, $P4_0/TMCI_0/XDDB_0$			$P4_5/TMRI_1$, $P4_4/TMO_1$, $P4_3/TMCI_1$, $P4_2/TMRI_0$, $P4_1/TMO_0$, $P4_0/TMCI_0$ 8-bit timer 0/1 input/output ($TMCI_0$, TMO_0, $TMRI_0$, $TMCI_1$, TMO_1, $TMRI_1$) and general input/output	

Table 8.1 Port Functions (cont)

Port	Description	Pins	Expanded Modes		Single-Chip Mode	
					Mode 3	
			Mode 1	Mode 2	PBI Disabled (DPME = 0)	PBI Enabled (DPME = 1)
Port 5	• 3-bit I/O port	$P5_2/SCK_0$, $P5_1/RxD_0$, $P5_0/TxD_0$	General input/output or serial communication interface 0 input/output (TxD_0, RxD_0, SCK_0)			
Port 6	• 8-bit I/O port	$P6_7/\overline{IRQ_7}/ETMO_1$, $P6_6/FTOB_0/\overline{IRQ_6}/ETMRI_1$, $P6_5/FTID/ETMCI_1$, $P6_4/FTIC/ETMO_0$, $P6_3/FTIB/ETMRI_0$, $P6_2/FTIA/FTI$, $P6_1/FTOA_0$, $P6_0/FTCI/ETMCI_0$	16-bit free-running timer 0 input/output (FTCI, $FTOA_0$, $FTOB_0$, FTIA, FTIB, FTIC, FTID), 16-bit free-running timer 1 input (FTCI, FTI), interrupt input ($\overline{IRQ_6}$, $\overline{IRQ_7}$), and 8-bit general input/output when PBI is disabled (DPME = 0) The above and 8-bit timer 0/1 input/output ($ETMCI_0$, $ETMRI_0$, $ETMO_0$, $ETMCI_1$, $ETMRI_1$, $ETMO_1$) when PBI is enabled (DPME = 1)		16-bit free-running timer 0 input/output (FTCI, $FTOA_0$, $FTOB_0$, FTIA, FTIB, FTIC, FTID), 16-bit free-running timer 1 input (FTCI, FTI), interrupt input ($\overline{IRQ_6}$, $\overline{IRQ_7}$), and 8-bit general input/output	
Port 7	• 8-bit I/O port	$P7_7/AN_7$ to $P7_0/AN_0$	Analog input to A/D converter (AN_7 to AN_0) and general input			
Port 8	• 7-bit I/O port	$P8_6/SCK_1/\overline{IRQ_5}/\overline{XOE}$, $P8_5/RxD_1/\overline{IRQ_4}$, $P8_4/TxD_1/\overline{IRQ_3}/\overline{XWE}$	Serial communication interface 1 input/output (TxD_1, RxD_1, SCK_1), interrupt input ($\overline{IRQ_3}$, $\overline{IRQ_4}$, $\overline{IRQ_5}$), and general input/output when PBI is disabled (DPME = 0) $\overline{XOE}$ and $\overline{XWE}$ input, serial communication interface 1 input (RxD_1), $\overline{IRQ_4}$ input, and general input/output when PBI is enabled (DPME = 1)		Serial communication interface 1 input/output (TxD_1, RxD_1, SCK_1), interrupt input ($\overline{IRQ_3}$, $\overline{IRQ_4}$, $\overline{IRQ_5}$), and general input/output	
		$P8_3/\overline{WRQ}/\overline{XRDY}$	General input/output when PBI is disabled (DPME = 0) $\overline{WRQ}$ and $\overline{XRDY}$ output when PBI is enabled (DPME = 1)		General input/output	$\overline{WRQ}$ output
		$P8_2/RS2$ to $P8_0/RS_0$	General input/output when PBI is disabled (DPME = 0) RS_2 to RS_0 input when PBI is enabled (DPME = 1)		General input/output	RS_2 to RS_0 input

Table 8.1 Port Functions (cont)

Port	Description	Pins	Expanded Modes		Single-Chip Mode	
					Mode 3	
			Mode 1	Mode 2	PBI Disabled (DPME = 0)	PBI Enabled (DPME = 1)
Port 9	• 8-bit I/O port	$P9_7$/$\overline{WAIT}$/$\overline{WE}$	$\overline{WAIT}$ input		General input/ output	$\overline{WE}$ input
		$P9_6$/ø	System clock (ø) output		General input when DDR = 0 (initial value) System clock (ø) output when DDR = 1	
		$P9_5$/$\overline{AS}$/$\overline{RDY}$	$\overline{AS}$ output		General input/ output	$\overline{RDY}$ output
		$P9_4$/$\overline{WR}$/$\overline{OE}$	$\overline{WR}$ output			$\overline{OE}$ input
		$P9_3$/$\overline{RD}$/$\overline{CS}$	$\overline{RD}$ output			$\overline{CS}$ input
		$P9_2$/$\overline{IRQ_0}$, $P9_1$/$\overline{IRQ_1}$/$\overline{XCS}$	Interrupt input ($\overline{IRQ_0}$, $\overline{IRQ_1}$) or general input/output when PBI is disabled (DPME = 0) $\overline{IRQ_0}$ input, general input/output, and $\overline{XCS}$ input when PBI is enabled (DPME = 1)		$\overline{IRQ_0}$ and $\overline{IRQ_1}$ input, and general input/output	
		$P9_0$/$\overline{IRQ_2}$/$\overline{ADTRG}$	External trigger input to A/D converter ($\overline{ADTRG}$), $\overline{IRQ_2}$ input, and general input/output			

8.2 Port 1

8.2.1 Overview

Port 1 is an 8-bit input/output port that is multiplexed with the lower address output pins (A_7 to A_0) and TPC output pins (TP_7 to TP_0). Figure 8.1 shows the pin configuration of port 1. The pin functions differ depending on the operating mode.

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

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

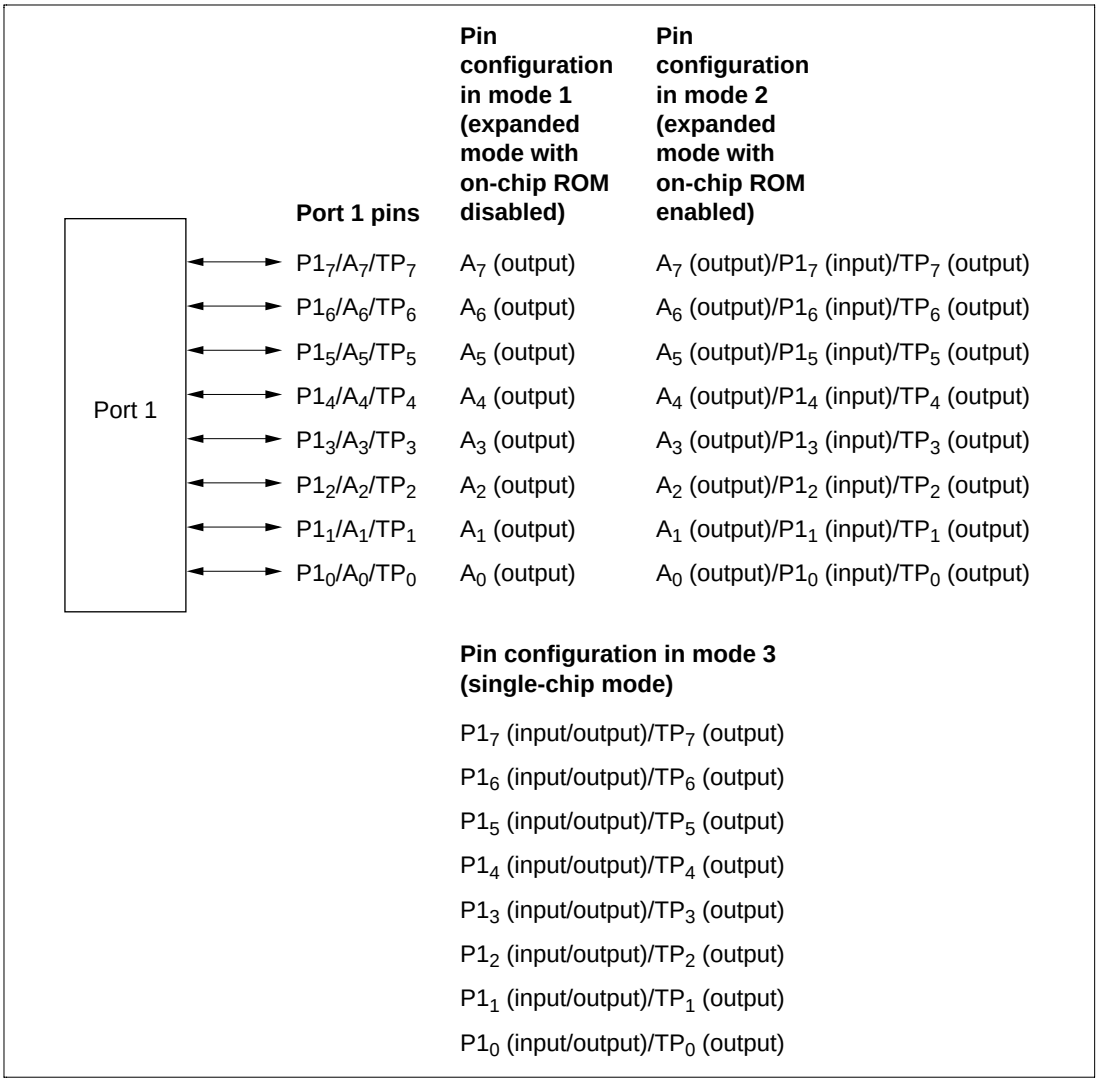


Figure 8.1 Port 1 Pin Configuration

8.2.2 Register Configuration and Descriptions

Table 8.2 summarizes the port 1 registers.

Table 8.2 Port 1 Registers

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

Port 1 Data Direction Register (P1DDR)

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

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

- Mode 1

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

  In hardware standby mode, the address bus is in the high-impedance state.
- Mode 2

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

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

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

Port 1 Data Register (P1DR)

Bit	7	6	5	4	3	2	1	0
	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

Port 1 Input Pull-Up Control Register (P1PCR)

Bit	7	6	5	4	3	2	1	0
	$P1_7$PCR	$P1_6$PCR	$P1_5$PCR	$P1_4$PCR	$P1_3$PCR	$P1_2$PCR	$P1_1$PCR	$P1_0$PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

8.2.3 Pin Functions in Each Mode

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

Pin Functions in Mode 1: In mode 1 (expanded mode with on-chip ROM disabled), port 1 is automatically used for lower address output (A_7 to A_0). Figure 8.2 shows the pin functions in mode 1.

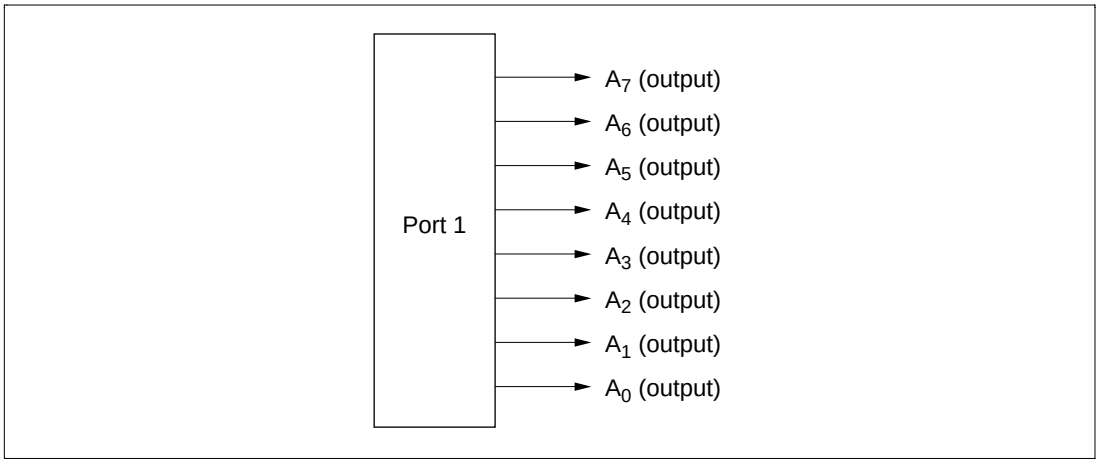


Figure 8.2 Pin Functions in Mode 1 (Port 1)

Mode 2: In mode 2 (expanded mode with on-chip ROM enabled), port 1 can provide lower address output pins, TPC output pins, and general input pins. Each pin becomes a lower address output pin or TPC output pin if its P1DDR bit is set to 1, and a general input pin if its P1DDR bit is cleared to 0. Following a reset, all pins are input pins. To be used for address output or TPC output, their P1DDR bits must be set to 1. Figure 8.3 shows the pin functions in mode 2.

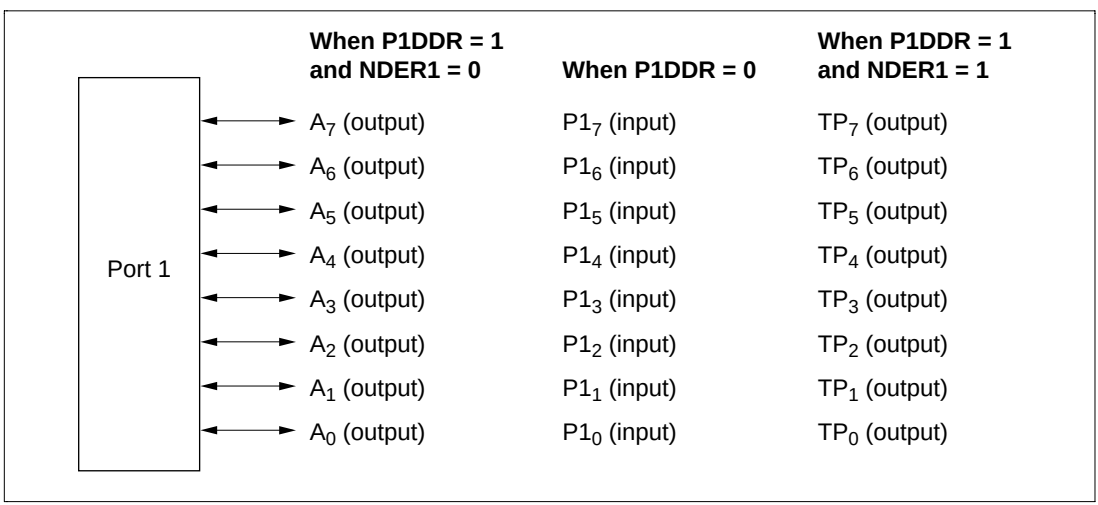


Figure 8.3 Pin Functions in Mode 2 (Port 1)

Mode 3: In mode 3 (single-chip mode), port 1 can provide TPC output pins and general input/output pins. For general input/output, the input or output direction of each pin can be selected individually. A pin becomes a general input pin when its P1DDR bit is cleared to 0. When its P1DDR bit is set to 1, the pin becomes a general output pin if its NDER1 bit is cleared to 0, or a TPC output pin if its NDER1 bit is set to 1. Figure 8.4 shows the pin functions in mode 2.

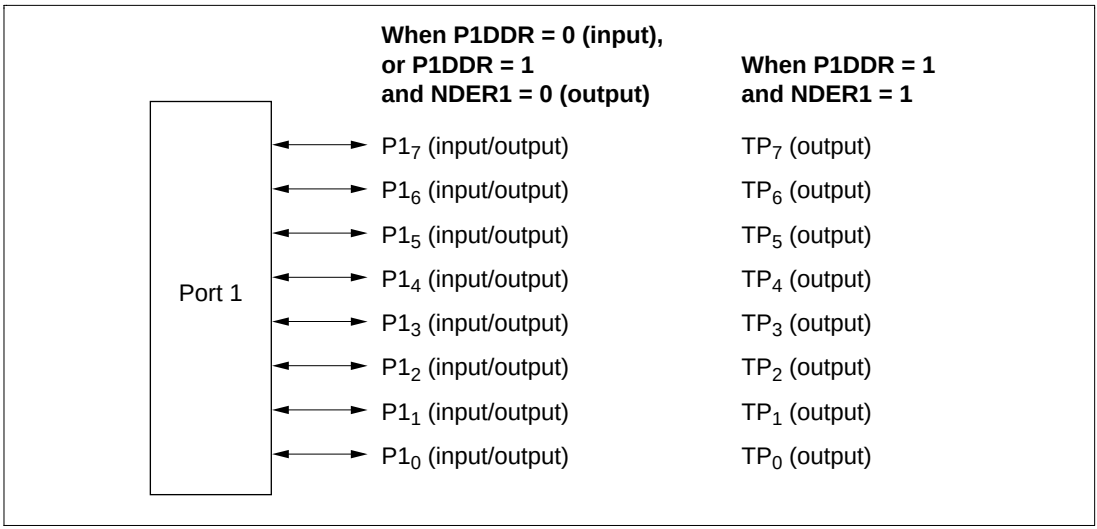


Figure 8.4 Pin Functions in Mode 3 (Port 1)

8.2.4 Input Pull-Up Transistors

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

Table 8.3 indicates the states of the input pull-up transistors in each operating mode.

Table 8.3 States of Input Pull-Up Transistors (Port 1)

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

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

8.3 Port 2

8.3.1 Overview

Port 2 is an 8-bit input/output port that is multiplexed with the upper address output pins (A_{15} to A_8) and TPC output pins (TP_{15} to TP_8). Figure 8.5 shows the pin configuration of port 2. The pin functions differ depending on the operating mode.

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

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

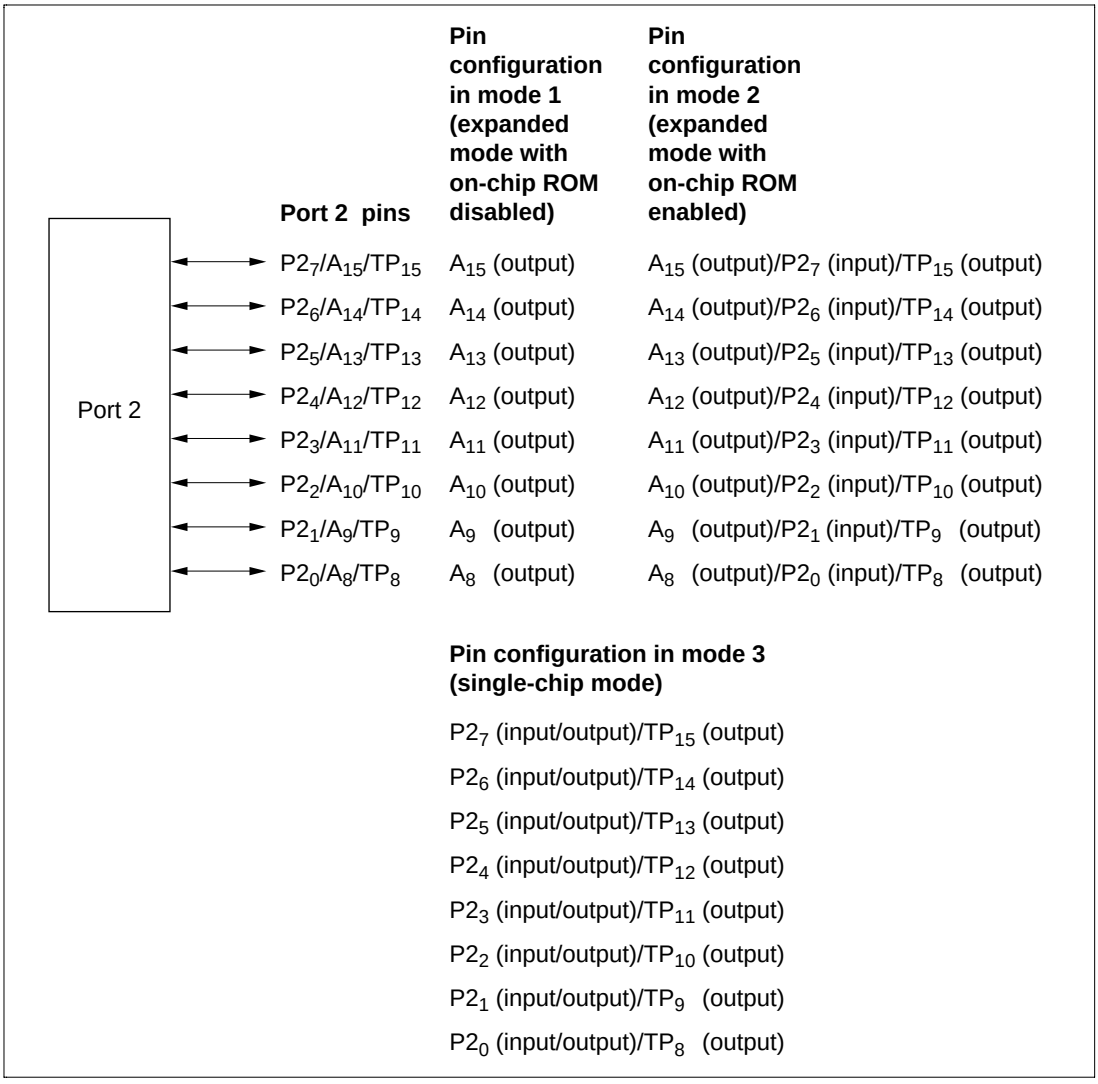


Figure 8.5 Port 2 Pin Configuration

8.3.2 Register Configuration and Descriptions

Table 8.4 summarizes the port 2 registers.

Table 8.4 Port 2 Registers

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

Port 2 Data Direction Register (P2DDR)

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

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

- Mode 1

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

  In hardware standby mode, the address bus is in the high-impedance state.
- Mode 2

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

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

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

Port 2 Data Register (P2DR)

Bit	7	6	5	4	3	2	1	0
	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

Port 2 Input Pull-Up Control Register (P2PCR)

Bit	7	6	5	4	3	2	1	0
	$P2_7$PCR	$P2_6$PCR	$P2_5$PCR	$P2_4$PCR	$P2_3$PCR	$P2_2$PCR	$P2_1$PCR	$P2_0$PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

8.3.3 Pin Functions in Each Mode

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

Pin Functions in Mode 1: In mode 1 (expanded mode with on-chip ROM disabled), port 2 is automatically used for upper address output (A_{15} to A_8). Figure 8.6 shows the pin functions in mode 1.

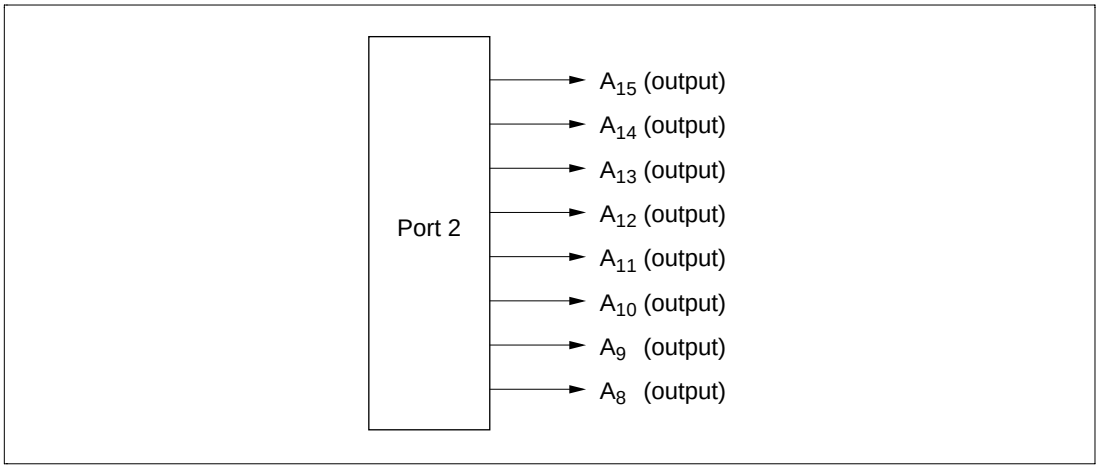


Figure 8.6 Pin Functions in Mode 1 (Port 2)

Mode 2: In mode 2 (expanded mode with on-chip ROM enabled), port 2 can provide upper address output pins, TPC output pins, and general input pins. Each pin becomes an upper address output pin or TPC output pin if its P2DDR bit is set to 1, and a general input pin if its P2DDR bit is cleared to 0. Following a reset, all pins are input pins. To be used for address output or TPC output, their P2DDR bits must be set to 1. Figure 8.7 shows the pin functions in mode 2.

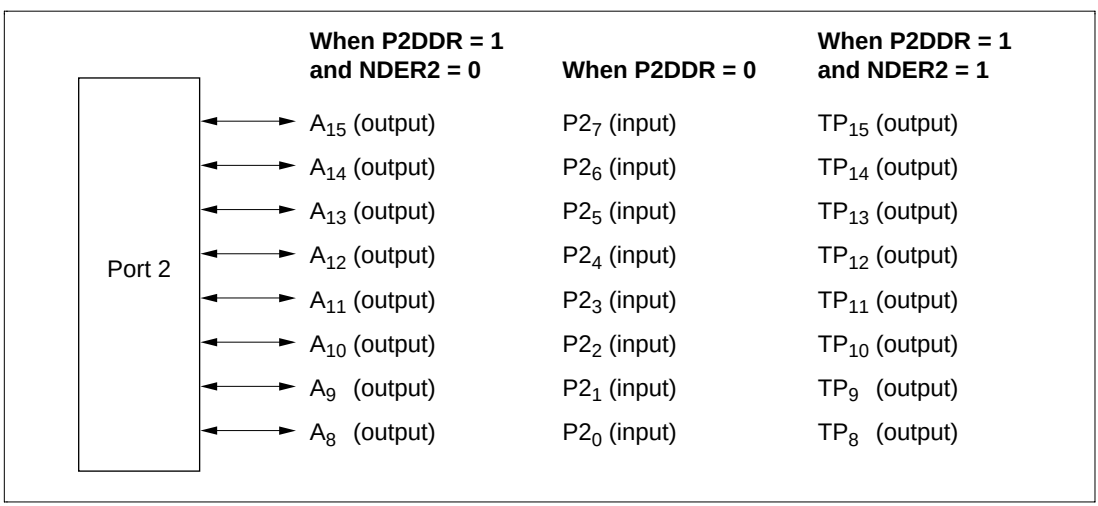


Figure 8.7 Pin Functions in Mode 2 (Port 2)

Mode 3: In mode 3 (single-chip mode), port 2 can provide TPC output pins and general input/output pins. For general input/output, the input or output direction of each pin can be selected individually. A pin becomes a general input pin when its P2DDR bit is cleared to 0. When its P2DDR bit is set to 1, the pin becomes a general output pin if its NDER2 bit is cleared to 0, or a TPC output pin if its NDER2 bit is set to 1. Figure 8.8 shows the pin functions in mode 2.

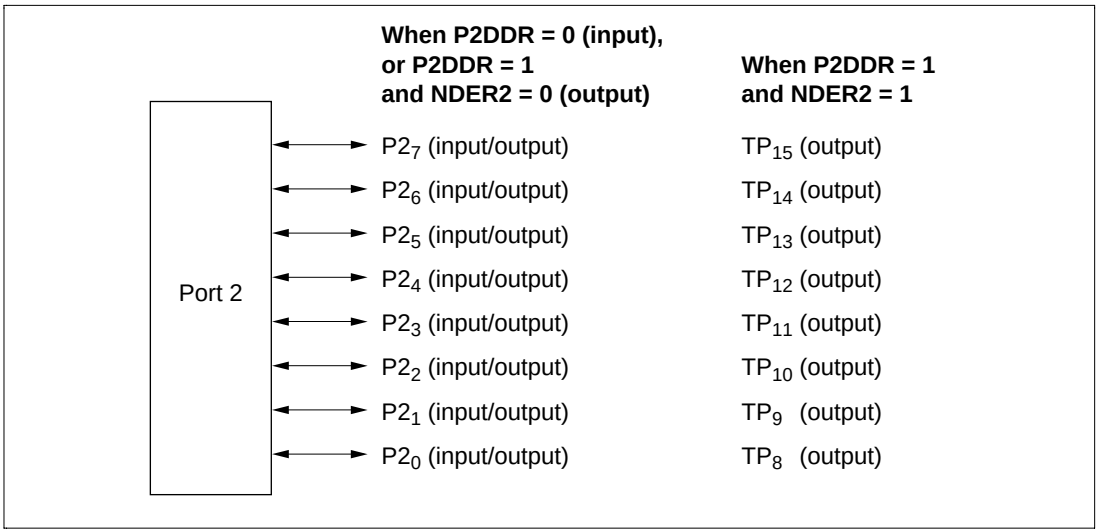


Figure 8.8 Pin Functions in Mode 3 (Port 2)

8.3.4 Input Pull-Up Transistors

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

Table 8.5 indicates the states of the input pull-up transistors in each operating mode.

Table 8.5 States of Input Pull-Up Transistors (Port 2)

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

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

8.4 Port 3

8.4.1 Overview

Port 3 is an 8-bit input/output port that is multiplexed with the data bus (D_7 to D_0) and DPRAM data bus (DDB_7 to DDB_0). Figure 8.9 shows the pin configuration of port 3. The pin functions differ depending on the operating mode.

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

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

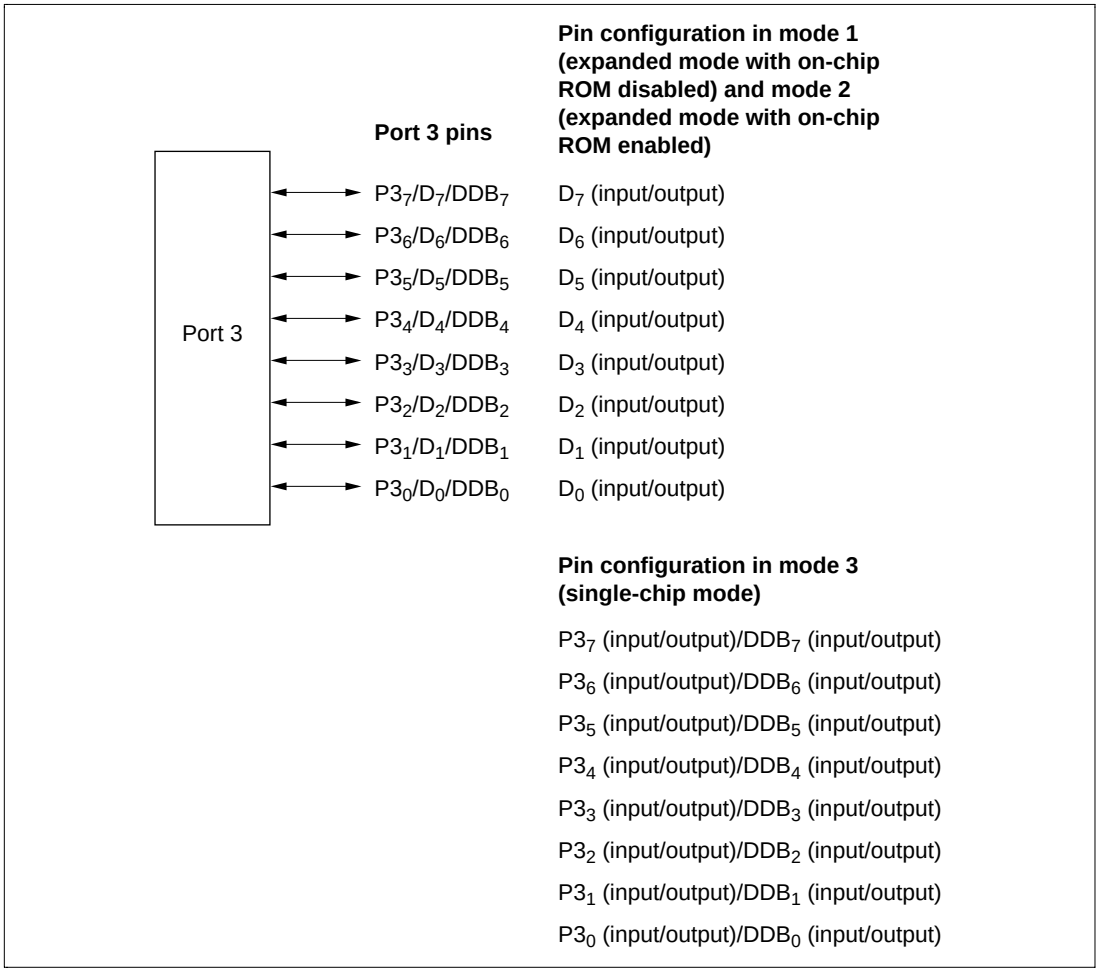


Figure 8.9 Port 3 Pin Configuration

8.4.2 Register Configuration and Descriptions

Table 8.6 summarizes the port 3 registers.

Table 8.6 Port 3 Registers

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

Port 3 Data Direction Register (P3DDR)—H'FFB4

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 register that controls the input/output direction of each pin in port 3. P3DDR is a write-only register. Read data is invalid. If read, all bits always read 1.

- Modes 1 and 2

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

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

- Mode 3

  When the DPME bit is cleared to 0 in SYSCR, a pin in port 3 is used for general output if the corresponding P3DDR bit is set to 1, and for general input if this bit is cleared to 0.

  P3DDR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values, so if a transition to software standby mode occurs while a P3DDR bit is set to 1, the corresponding pin remains in the output state.

  When the DPME bit is set to 1 (slave mode), P3DDR is cleared to H'00 and port 3 consists of input/output pins for the DPRAM data bus (DDB_7 to DDB_0). In software standby mode, the DPRAM data bus is in the high-impedance state. A reset or entry to hardware standby mode clears the DPME bit to 0 and initializes P3DDR to H'00.

Port 3 Data Register (P3DR)

Bit	7	6	5	4	3	2	1	0
	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

Port 3 Input Pull-Up Control Register (P3PCR)

Bit	7	6	5	4	3	2	1	0
	$P3_7$PCR	$P3_6$PCR	$P3_5$PCR	$P3_4$PCR	$P3_3$PCR	$P3_2$PCR	$P3_1$PCR	$P3_0$PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PP3PCR is an 8-bit readable/writable register that controls the input pull-up transistors in port 3. If a P3DDR bit is cleared to 0 (designating input) and the corresponding P3PCR bit is set to 1, the input pull-up transistor is turned on.

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

8.4.3 Pin Functions in Each Mode

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

Pin Functions in Modes 1 and 2: In mode 1 (expanded mode with on-chip ROM disabled) and mode 2 (expanded mode with on-chip ROM enabled), port 3 is automatically used for the input/output pins of the data bus (D_7 to D_0). Figure 8.10 shows the pin functions in modes 1 and 2.

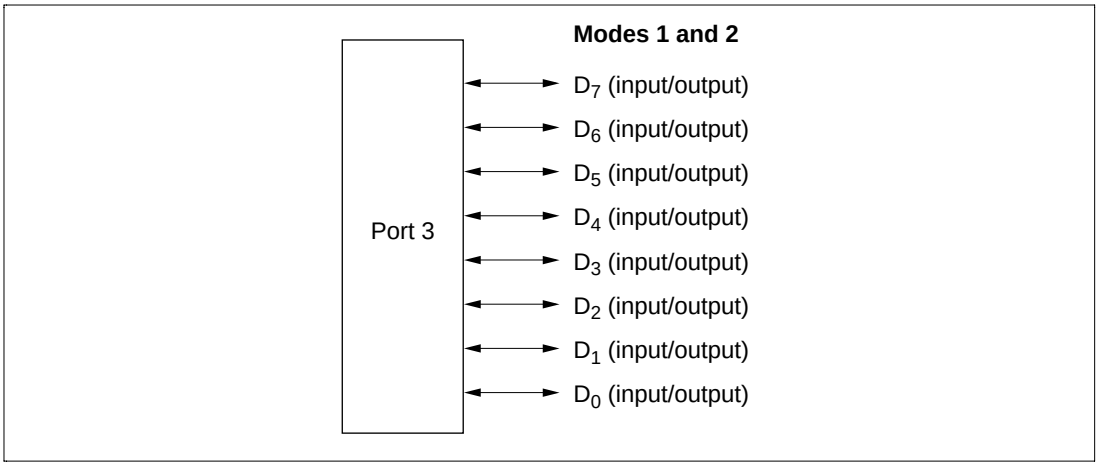


Figure 8.10 Pin Functions in Modes 1 and 2 (Port 3)

Mode 3: In mode 3 (single-chip mode), when the DPME bit is cleared to 0 in SYSCR, each pin can be designated for general input or output. A pin becomes an output pin when its P3DDR bit is set to 1, and an input pin when this bit is cleared to 0.

When the DPME bit is set to 1 (slave mode), port 3 becomes the DPRAM data bus (DDB_7 to DDB_0). P3DR should be cleared to H'00. P3DDR is automatically cleared to H'00.

Figure 8.11 shows the pin functions in mode 3.

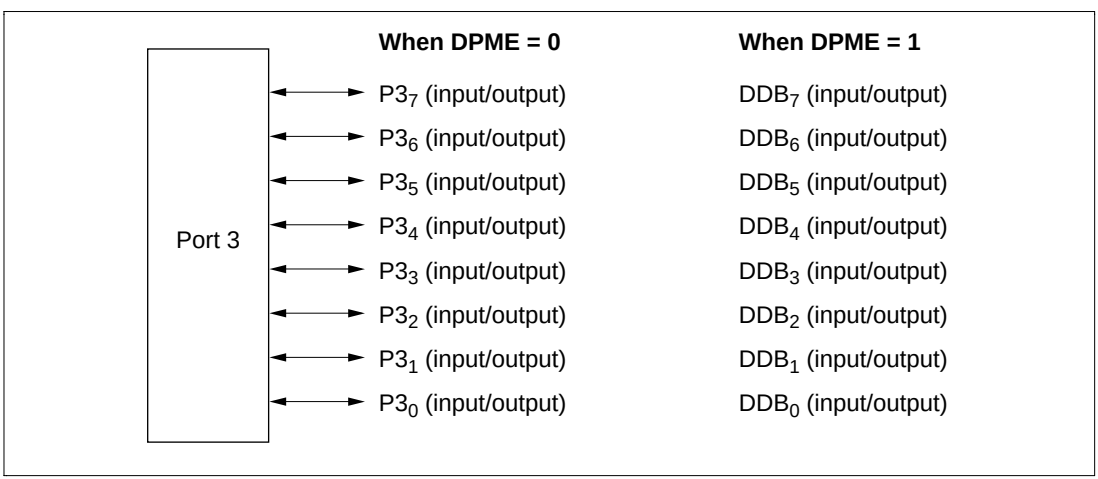


Figure 8.11 Pin Functions in Mode 3 (Port 3)

8.4.4 Input Pull-Up Transistors

Port 3 has built-in programmable input pull-up transistors that are available in mode 3. The pull-up for each bit can be turned on and off individually. To turn on an input pull-up in mode 3, clear bit DPME of SYSCR and bit P3DDR to 0, then write 1 to P3PCR. When DPME is set to 1 (slave mode), input pull-ups cannot be turned on. Input pull-ups are turned off after a reset and in hardware standby mode.

Table 8.7 indicates the states of the input pull-up transistors in each operating mode.

Table 8.7 States of Input Pull-Up Transistors (Port 3)

Mode		Reset	Hardware Standby	Other Operating Modes
1		Off	Off	Off
2		Off	Off	Off
3	DPME = 0	Off	Off	On/off
	DPME 1 (slave mode)			Off

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

8.5 Port 4

8.5.1 Overview

Port 4 is an 8-bit input/output port that is multiplexed with the DPRAM data bus ($XDDB_7$ to $XDDB_0$), with input/output pins ($TMRI_0$, $TMRI_1$, $TMCI_0$, $TMCI_1$, TMO_0, TMO_1) of 8-bit timers 0 and 1, and with output pins ($FTOA_1$, $FTOB_1$) of 16-bit free running timer 1 (FRT1). Figure 8.12 shows the pin configuration of port 4. The pin functions differ depending on the operating mode.

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

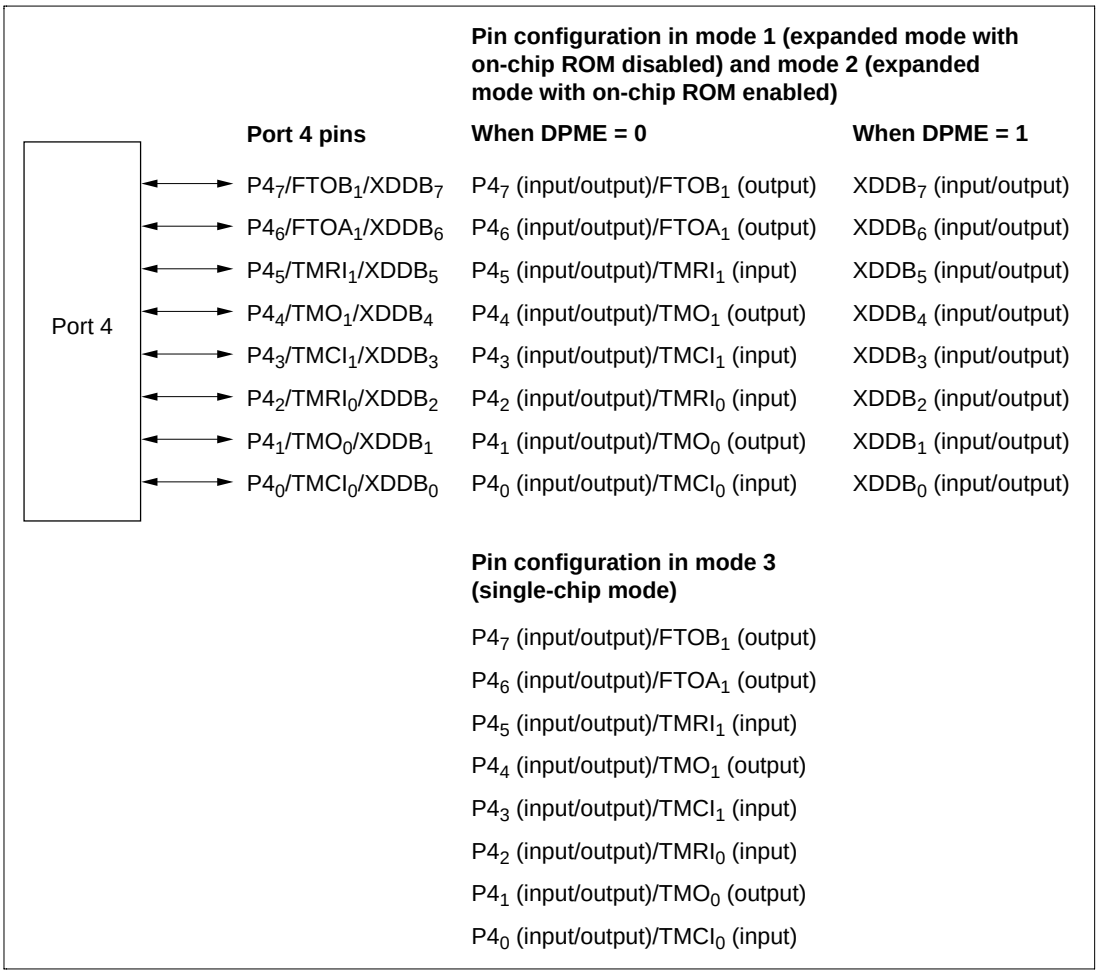


Figure 8.12 Port 4 Pin Configuration

8.5.2 Register Configuration and Descriptions

Table 8.8 summarizes the port 4 registers.

Table 8.8 Port 4 Registers

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

Port 4 Data Direction Register (P4DDR)

Bit	7	6	5	4	3	2	1	0
	$P4_7DDR$	$P4_6DDR$	$P4_5DDR$	$P4_4DDR$	$P4_3DDR$	$P4_2DDR$	$P4_1DDR$	$P4_0DDR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

P4DDR is an 8-bit register that controls the input/output direction of each pin in port 4. A pin functions as an output pin if the corresponding P4DDR bit is set to 1, and as an input pin if this bit is cleared to 0.

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

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

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

Port 4 Data Register (P4DR)

Bit	7	6	5	4	3	2	1	0
	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

8.5.3 Pin Functions in Each Mode

Port 4 has different pin functions in different modes. A separate description for each mode is given below.

Pin Functions in Modes 1 and 2: In mode 1 (expanded mode with on-chip ROM disabled) and mode 2 (expanded mode with on-chip ROM enabled), port 4 is multiplexed with the DPRAM data bus, ($XDDB_7$ to $XDDB_0$), with input/output pins ($TMRI_0$, $TMRI_1$, $TMCI_0$, $TMCI_1$, TMO_0, TMO_1) of 8-bit timers 0 and 1, and with output pins ($FTOA_1$, $FTOB_1$) of 16-bit free running timer 1 (FRT1). Table 8.9 indicates the pin functions in modes 1 and 2.

Table 8.9 Pin Functions in Modes 1 and 2

Pin	Pin Functions and Selection Method
$P4_7$/$FTOB_1$/ $XDDB_7$	Bit DPME in SYSCR, bit OEB in TCR of FRT1, and bit $P4_7$DDR select the pin function as follows

DPME	0			1
OEB	0		1	X
$P4_7$DDR	0	1	X	X
Pin function	$P4_7$ input	$P4_7$ output	$FTOB_1$ output	$XDDB_7$ input/output

$P4_6$/$FTOA_1$/ $XDDB_6$	Bit DPME in SYSCR, bit OEA in TCR of FRT1, and bit $P4_6$DDR select the pin function as follows

DPME	0			1
OEA	0		1	X
$P4_6$DDR	0	1	X	X
Pin function	$P4_6$ input	$P4_6$ output	$FTOA_1$ output	$XDDB_6$ input/output

$P4_5$/$TMRI_1$/ $XDDB_5$	Bit DPME in SYSCR and bit $P4_5$DDR select the pin function as follows

DPME	0		1
$P4_5$DDR	0	1	X
Pin function	$P4_5$ input	$P4_5$ output	$XDDB_5$ input/output
	$TMRI_1$ input		

$TMRI_1$ input is usable when bits CCLR1 and CCLR0 are both set to 1 in TCR of 8-bit timer 1

$P4_4$/TMO_1/ $XDDB_4$	Bit DPME in SYSCR, bits OS3 to OS0 in TCSR of 8-bit timer 1, and bit $P4_4$DDR select the pin function as follows

DPME	0			1
OS3 to OS0	All 0		Not all 0	X
P_4DDR	0	1	X	X
Pin function	$P4_4$ input	$P4_4$ output	TMO_1 output	$XDDB_4$ input/output

X: 0 and 1 settings both give the same pin function.

Table 8.9 Pin Functions in Modes 1 and 2 (cont)

Pin	Pin Functions and Selection Method
$P4_3$/$TMCI_1$/ $XDDB_3$	Bit DPME in SYSCR and bit $P4_3$DDR select the pin function as follows

DPME	0		1
$P4_3$DDR	0	1	X
Pin function	$P4_3$ input	$P4_3$ output	$XDDB_3$ input/output
	$TMCI_1$ output		

$TMCI_1$ input is usable when bits CKS2 to CKS0 in TCR of 8-bit timer 1 select an external clock source

$P4_2$/$TMRI_0$/ $XDDB_2$ — Bit DPME in SYSCR, bits CCLR1 and CCLR0 in TCR of 8-bit timer 0, and bit $P4_2$DDR select the pin function as follows

DPME	0		1
$P4_2$DDR	0	1	X
Pin function	$P4_2$ input	$P4_2$ output	$XDDB_2$ input/output
	$TMRI_0$ input		

$TMRI_0$ input is usable when bits CCLR1 and CCLR0 are both set to 1 in TCR of 8-bit timer 0

$P4_1$/TMO_0/ $XDDB_1$ — Bit DPME in SYSCR, bits OS3 to OS0 in TCSR of 8-bit timer 0, and bit $P4_1$DDR select the pin function as follows

DPME	0			1
OS3 to OS0	All 0		Not all 0	X
$P4_1$DDR	0	1	X	X
Pin function	$P4_1$ input	$P4_1$ output	TMO_0 output	$XDDB_1$ input/output

$P4_0$/$TMCI_0$/ $XDDB_0$ — Bit DPME in SYSCR and bit $P4_0$DDR select the pin function as follows

DPME	0		1
$P4_0$DDR	0	1	X
Pin function	$P4_0$ input	$P4_0$ output	$XDDB_0$ input/output
	$TMCI_0$ output		

$TMCI_0$ input is usable when bits CKS2 to CKS0 in TCR of 8-bit timer 0 select an external clock source

X: 0 and 1 settings both give the same pin function.

Pin Functions in Mode 3: In mode 3 (single-chip mode), port 4 is multiplexed with input/output pins ($TMRI_0$, $TMRI_1$, $TMCI_0$, $TMCI_1$, TMO_0, TMO_1) of 8-bit timers 0 and 1, and with output pins ($FTOA_1$, $FTOB_1$) of 16-bit free running timer 1 (FRT1). Table 8.10 indicates the pin functions in mode 3.

Table 8.10 Pin Functions in Mode 3

Pin	Pin Functions and Selection Method
$P4_7$/$FTOB_1$	Bit OEB in TCR of FRT1 and bit $P4_7$DDR select the pin function as follows

OEB	0		1
$P4_7$DDR	0	1	X
Pin function	$P4_7$ input	$P4_7$ output	$FTOB_1$ output

$P4_6$/$FTOA_1$	Bit OEA in TCR of FRT1 and bit $P4_6$DDR select the pin function as follows

OEA	0		1
$P4_6$DDR	0	1	X
Pin function	$P4_6$ input	$P4_6$ output	$FTOA_1$ output

$P4_5$/$TMRI_1$	

$P4_5$DDR	0	1
Pin function	$P4_5$ input	$P4_5$ output
	$TMRI_1$ input	

$TMRI_1$ input is usable when bits CCLR1 and CCLR0 are both set to 1 in TCR of 8-bit timer 1

$P4_4$/TMO_1	Bits OS3 to OS0 in TCSR of 8-bit timer 1 and bit $P4_4$DDR select the pin function as follows

OS3 to OS0	All 0		Not all 0
$P4_4$DDR	0	1	X
Pin function	$P4_4$ input	$P4_4$ output	TMO_1 output

X: 0 and 1 settings both give the same pin function.

Table 8.10 Pin Functions in Mode 3 (cont)

Pin	Pin Functions and Selection Method
$P4_3/TMCI_1$	

$P4_3$DDR	0	1
Pin function	$P4_3$ input	$P4_3$ output
	$TMCI_1$ input	

$TMCI_1$ input is usable when bits CKS2 to CKS0 in TCR of 8-bit timer 1 select an external clock source

$P4_2/TMRI_0$

$P4_2$DDR	0	1
Pin function	$P4_2$ input	$P4_2$ output
	$TMRI_0$ input	

$TMRI_0$ input is usable when bits CCLR1 and CCLR0 are both set to 1 in TCR of 8-bit timer 0

$P4_1/TMO_0$

Bits OS3 to OS0 in TCSR of 8-bit timer 0 and bit $P4_1$DDR select the pin function as follows

OS3 to OS0	All 0		Not all 0
$P4_1$DDR	0	1	X
Pin function	$P4_1$ input	$P4_1$ output	TMO_0 output

$P4_0/TMCI_0$

$P4_0$DDR	0	1
Pin function	$P4_0$ input	$P4_0$ output
	$TMCI_0$ input	

$TMCI_0$ input is usable when bits CKS2 to CKS0 in TCR of 8-bit timer 0 select an external clock source

X: 0 and 1 settings both give the same pin function.

8.6 Port 5

8.6.1 Overview

Port 5 is a 3-bit input/output port that is multiplexed with input/output pins (TxD_0, RxD_0, SCK_0) of serial communication interface 0. The port 5 pin functions are the same in all operating modes. Figure 8.13 shows the pin configuration of port 5.

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

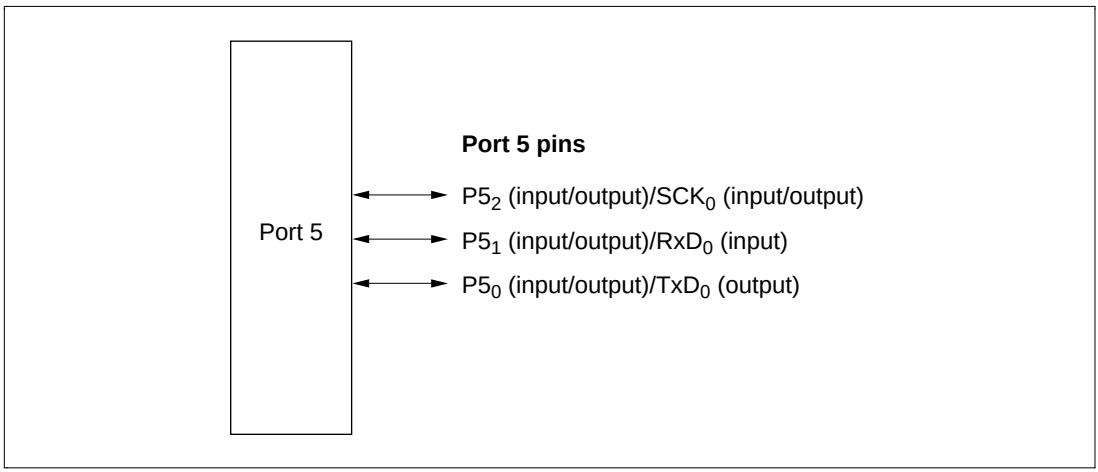


Figure 8.13 Port 5 Pin Configuration

8.6.2 Register Configuration and Descriptions

Table 8.11 summarizes the port 5 registers.

Table 8.11 Port 5 Registers

Name	Abbreviation	Read/Write	Initial Value	Address
Port 5 data direction register	P5DDR	W	H'F8	H'FFB8
Port 5 data register	P5DR	R/W	H'F8	H'FFBA

Port 5 Data Direction Register (P5DDR)

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

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

P5DDR is a write-only register. Read data is invalid. If read, all bits always read 1.

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

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

Port 5 Data Register (P5DR)

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

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

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

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

8.6.3 Pin Functions in Each Mode

Port 5 has the same pin functions in each operating mode. All pins can also be used as SCI0 input/output pins. Table 8.12 indicates the pin functions of port 5.

Table 8.12 Pin Functions of Port 5

<table>
<tr><th>Pin</th><th>Pin Functions and Selection Method</th></tr>
<tr><td>$P5_2/SCK_0$</td><td>Bit C/$\overline{A}$ in SMR of SCI0, bits CKE0 and CKE1 in SCR of SCI0, and bit $P5_2$DDR select the pin function as follows
<table>
<tr><td>CKE1</td><td colspan="3">0</td><td>1</td></tr>
<tr><td>C/$\overline{A}$</td><td colspan="2">0</td><td>1</td><td>—</td></tr>
<tr><td>CKE0</td><td colspan="2">0</td><td>1</td><td>—</td><td>—</td></tr>
<tr><td>$P5_2$DDR</td><td>0</td><td>1</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>Pin function</td><td>$P5_2$ input</td><td>$P5_2$ output</td><td>SCK_0 output</td><td>SCK_0 output</td><td>SCK_0 input</td></tr>
</table></td></tr>
<tr><td>$P5_1/RxD_0$</td><td>Bit RE in SCR of SCI0 and bit $P5_1$DDR select the pin function as follows
<table>
<tr><td>RE</td><td colspan="2">0</td><td>1</td></tr>
<tr><td>$P5_1$DDR</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td>Pin function</td><td>$P5_1$ input</td><td>$P5_1$ output</td><td>RxD_0 input</td></tr>
</table></td></tr>
<tr><td>$P5_0/TxD_0$</td><td>Bit TE in SCR of SCI0 and bit $P5_0$DDR select the pin function as follows
<table>
<tr><td>TE</td><td colspan="2">0</td><td>1</td></tr>
<tr><td>$P5_0$DDR</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td>Pin function</td><td>$P5_0$ input</td><td>$P5_0$ output</td><td>TxD_0 output</td></tr>
</table></td></tr>
</table>

8.7 Port 6

8.7.1 Overview

Port 6 is an 8-bit input/output port that is multiplexed with input/output pins ($FTOA_0$, $FTOB_0$, FTIA to FTID, FTCI) of 16-bit free-running timer 0 (FRT0), with input pins (FTCI, FTI) of 16-bit free running timer 1 (FRT1), with input/output pins ($ETMCI_0$, $ETMRI_0$. $ETMO_0$, $ETMCI_1$, $ETMRI_1$, $ETMO_1$) of 8-bit timers 0 and 1, and with $\overline{IRQ}_6$ and $\overline{IRQ}_7$ input pins. The pin functions of $P6_2$ and $P6_1$ are the same in all operating modes. The pin functions of $P6_7$ to $P6_3$ and $P6_0$ differ depending on the operating mode. Figure 8.14 shows the pin configuration of port 6.

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

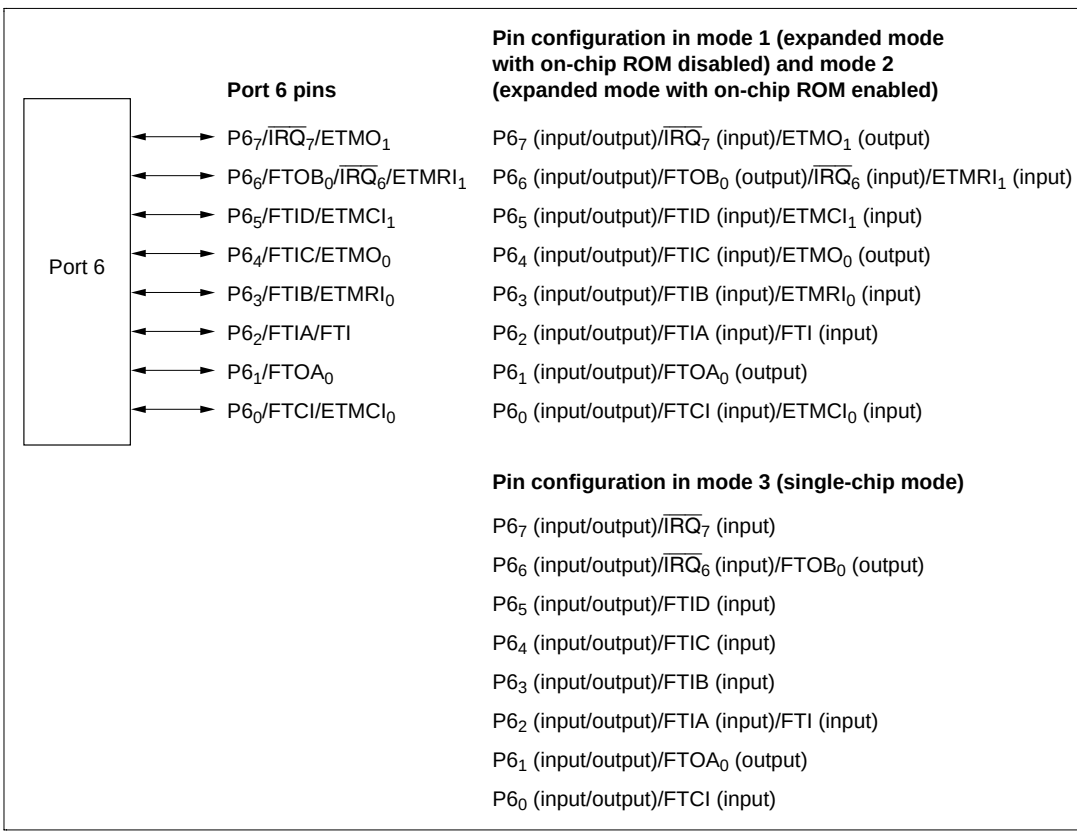


Figure 8.14 Port 6 Pin Configuration

8.7.2 Register Configuration and Descriptions

Table 8.13 summarizes the port 6 registers.

Table 8.13 Port 6 Registers

Name	Abbreviation	Read/Write	Initial Value	Address
Port 6 data direction register	P6DDR	W	H'00	H'FFB9
Port 6 data register	P6DR	R/W	H'00	H'FFBB

Port 6 Data Direction Register (P6DDR)—H'FFB9

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 register that controls the input/output direction of each pin in port 6. A pin functions as an output pin if the corresponding P6DDR bit is set to 1, and as an input pin if this bit is cleared to 0.

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

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

If a transition to software standby mode occurs while port 6 is being used by an on-chip supporting module (for example, for free-running timer output), the on-chip supporting module will be initialized, so the pin will revert to general-purpose input/output, controlled by P6DDR and P6DR.

Port 6 Data Register (P6DR)—H'FFBB

Bit	7	6	5	4	3	2	1	0
	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

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

8.7.3 Pin Functions in Each Mode

Port 6 has different pin functions in different modes. A separate description for each mode is given below.

Pin Functions in Modes 1 and 2: In mode 1 (expanded mode with on-chip ROM disabled) and mode 2 (expanded mode with on-chip ROM enabled), port 6 is multiplexed with input/output pins (FTCI, $FTOA_0$, $FTOB_0$, FTIA, FTIB, FTIC, FTID) of 16-bit free-running timer 0 (FRT0), with input pins (FTCI, FTI) of 16-bit free running timer 1 (FRT1), and with $\overline{IRQ}_6$ and $\overline{IRQ}_7$ input pins. When the PBI is enabled (when DPME is 1), port 6 is also multiplexed with input/output pins ($ETMRI_0$, $ETMRI_1$, $ETMCI_0$, $ETMCI_1$, $ETMO_0$, $ETMO_1$) of 8-bit timers 0 and 1. When the PBI is disabled (DPME is 0), the input/output pins of 8-bit timers 0 and 1 are multiplexed with port 4. Figure 8.15 and table 8.14 indicate the pin functions in modes 1 and 2.

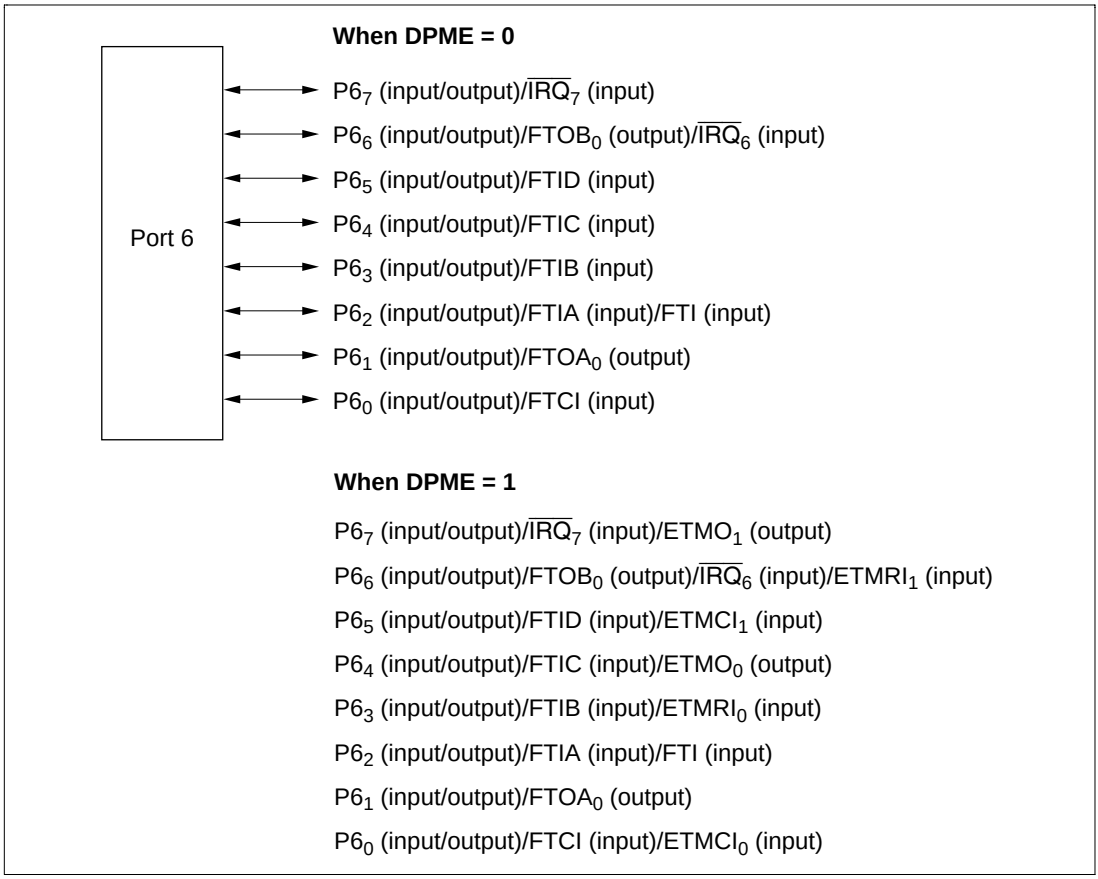


Figure 8.15 Pin Functions in Modes 1 and 2 (Port 6)

Table 8.14 Pin Functions in Modes 1 and 2

Pin	Pin Functions and Selection Method
$P6_7/\overline{IRQ}_7/ETMO_1$	Bit DPME in SYSCR, bits OS3 to OS0 in the timer control/status register (TCSR) of 8-bit timer 1, and bit $P6_7$DDR select the pin function as follows When bit DPME is 0

$P6_7$DDR	0	1
Pin function	$P6_7$ input	$P6_7$ output
	$\overline{IRQ}_7$ input	

When bit DPME is 1

OS3 to OS0	All 0		Not all 0
$P6_7$DDR	0	1	X
Pin function	$P6_7$ input	$P6_7$ output	$ETMO_1$ output
	$\overline{IRQ}_7$ input		

$\overline{IRQ}_7$ input is usable when bit IERQ7E is set to 1 in IER

$P6_6/FTOB_0/\overline{IRQ}_6/ETMRI_1$	Bit DPME in SYSCR, bit OEB in the timer output compare control register (TCR) of FRT0, and bit $P6_6$DDR select the pin function as follows When bit DPME is 0

OEB	0		1
$P6_6$DDR	0	1	X
Pin function	$P6_6$ input	$P6_6$ output	$FTOB_0$ output
	$\overline{IRQ}_6$ input		

When bit DPME is 1

OEB	0		1
$P6_6$DDR	0	1	X
Pin function	$P6_6$ input	$P6_6$ output	$FTOB_0$ output
	$\overline{IRQ}_6$ input and $ETMRI_1$ input		

$\overline{IRQ}_6$ input is usable when bit IERQ6E is set to 1 in IER.

$ETMRI_1$ input is usable when bits CCLR1 and CCLR0 are both set to 1 in TCR of 8-bit timer 1

X: 0 and 1 settings both give the same pin function.

Table 8.14 Pin Functions in Modes 1 and 2 (cont)

<table>
<tr><th>Pin</th><th>Pin Functions and Selection Method</th></tr>
<tr><td>P6₅/FTID/
ETMCI₁</td><td>Bit DPME in SYSCR and bit P6₅DDR select the pin function as follows
When bit DPME is 0
<table>
<tr><td>P6₅DDR</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>P6₅ input</td><td>P6₅ output</td></tr>
<tr><td></td><td colspan="2">FTID input</td></tr>
</table>
When bit DPME is 1
<table>
<tr><td>P6₅DDR</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>P6₅ input</td><td>P6₅ output</td></tr>
<tr><td></td><td colspan="2">FTID input and ETMCI₁ input</td></tr>
</table>
ETMCI₁ input is usable when bits CKS2 to CKS0 in TCR of 8-bit timer 1 select an external clock source</td></tr>
<tr><td>P6₄/FTIC/
ETMO₀</td><td>Bit DPME in SYSCR, bits OS3 to OS0 in TCSR of 8-bit timer 0, and bit P6₄DDR select the pin function as follows
When bit DPME is 0
<table>
<tr><td>P6₄DDR</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>P6₄ input</td><td>P6₄ output</td></tr>
<tr><td></td><td colspan="2">FTIC input</td></tr>
</table>
When bit DPME is 1
<table>
<tr><td>OS3 to OS0</td><td colspan="2">All 0</td><td>Not all 0</td></tr>
<tr><td>P6₄DDR</td><td>0</td><td>1</td><td>X</td></tr>
<tr><td>Pin function</td><td>P6₄ input</td><td>P6₄ output</td><td>ETMO₀ output</td></tr>
<tr><td></td><td colspan="3">FTIC input</td></tr>
</table></td></tr>
</table>

X: 0 and 1 settings both give the same pin function.

Table 8.14 Pin Functions in Modes 1 and 2 (cont)

<table>
<tr><th>Pin</th><th>Pin Functions and Selection Method</th></tr>
<tr><td>P6$_3$/FTIB/
ETMRI$_0$</td><td>Bit DPME in SYSCR and bit P6$_3$DDR select the pin function as follows
When bit DPME is 0
<table>
<tr><td>P6$_3$DDR</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>P6$_3$ input</td><td>P6$_3$ output</td></tr>
<tr><td></td><td colspan="2">FTIB input</td></tr>
</table>
When bit DPME is 1
<table>
<tr><td>P6$_3$DDR</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>P6$_3$ input</td><td>P6$_3$ output</td></tr>
<tr><td></td><td colspan="2">FTIB input and ETMRI$_0$ input</td></tr>
</table>
ETMRI$_0$ input is usable when bits CCLR1 and CCLR0 are both set to 1 in TCR of 8-bit timer 0</td></tr>
<tr><td>P6$_2$/FTIA/FTI</td><td>
<table>
<tr><td>P6$_2$DDR</td><td>0</td><td>1</td></tr>
<tr><td>Pin function</td><td>P6$_2$ input</td><td>P6$_2$ output</td></tr>
<tr><td></td><td colspan="2">FTIA input and FTI input</td></tr>
</table>
</td></tr>
<tr><td>P6$_1$/FTOA$_0$</td><td>Bit OEA in TOCR of FRT0 and bit P6$_1$DDR select the pin function as follows
<table>
<tr><td>OEA</td><td colspan="2">0</td><td>1</td></tr>
<tr><td>P6$_1$DDR</td><td>0</td><td>1</td><td>X</td></tr>
<tr><td>Pin function</td><td>P6$_1$ input</td><td>P6$_1$ output</td><td>FTOA$_0$ output</td></tr>
</table>
</td></tr>
<tr><td></td><td>X: 0 and 1 settings both give the same pin function.</td></tr>
</table>

Table 8.14 Pin Functions in Modes 1 and 2 (cont)

<table>
<tr><th>Pin</th><th>Pin Functions and Selection Method</th></tr>
<tr><td>$P6_0$/FTCI/
$ETMCI_0$</td><td>Bit DPME in SYSCR and bit $P6_0$DDR select the pin function as follows
When bit DPME is 0
<table>
<tr><td>$P6_0$DDR</td><td>0</td><td>1</td></tr>
<tr><td rowspan="2">Pin function</td><td>$P6_0$ input</td><td>$P6_0$ output</td></tr>
<tr><td colspan="2">FTCI input</td></tr>
</table>
When bit DPME is 1
<table>
<tr><td>$P6_0$DDR</td><td>0</td><td>1</td></tr>
<tr><td rowspan="2">Pin function</td><td>$P6_0$ input</td><td>$P6_0$ output</td></tr>
<tr><td colspan="2">FTCI input and $ETMCI_0$ input</td></tr>
</table>
FTCI input is usable by either FRT0 or FRT1 when bits CKS2 to CKS0 in its TCR select an external clock source
$ETMCI_0$ input is usable when bits CKS2 to CKS0 in TCR of 8-bit timer 0 select an external clock source</td></tr>
</table>

Pin Functions in Mode 3: In mode 3 (single-chip mode), port 6 is multiplexed with input/output pins (FTCI, $FTOA_0$, $FTOB_0$, FTIA, FTIB, FTIC, FTID) of FRT0, with input pins (FTCI, FTI) of FRT1, and with $\overline{IRQ}_6$ and $\overline{IRQ}_7$ input pins. Figure 8.16 and table 8.15 indicate the pin functions in mode 3.

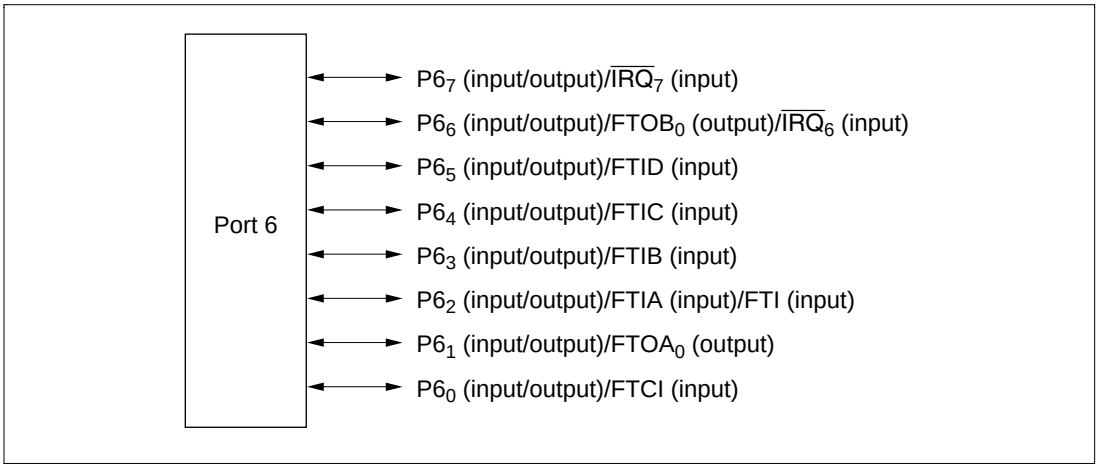


Figure 8.16 Pin Functions in Mode 3 (Port 6)

Table 8.15 Pin Functions in Mode 3

Pin	Pin Functions and Selection Method			
$P6_7/\overline{IRQ}_7$				
	$P6_7$DDR	0	1	
	Pin function	$P6_7$ input	$P6_7$ output	
		$\overline{IRQ}_7$ input		
	$\overline{IRQ}_7$ input is usable when bit IERQ7E is set to 1 in the IRQ enable register (IER)			
$P6_6/FTOB_0/\overline{IRQ}_6$	Bit OEB in the timer output compare control register (TOCR) of FRT0 and bit $P6_6$DDR select the pin function as follows			
	OEB	0		1
	$P6_6$DDR	0	1	X
	Pin function	$P6_6$ input	$P6_6$ output	$FTOB_0$ output
		$\overline{IRQ}_6$ input		
	$\overline{IRQ}_6$ input is usable when bit IERQ6E is set to 1 in IER			
	X: 0 and 1 settings both give the same pin function.			

Table 8.15 Pin Functions in Mode 3 (cont)

Pin	Pin Functions and Selection Method

$P6_5$/FTID

$P6_5$DDR	0	1
Pin function	$P6_5$ input	$P6_5$ output
	FTID input	

$P6_4$/FTIC

$P6_4$DDR	0	1
Pin function	$P6_4$ input	$P6_4$ output
	FTIC input	

$P6_3$/FTIB

$P6_3$DDR	0	1
Pin function	$P6_3$ input	$P6_3$ output
	FTIB input	

$P6_2$/FTIA/FTI

$P6_2$DDR	0	1
Pin function	$P6_2$ input	$P6_2$ output
	FTIA input and FTI input	

$P6_1$/$FTOA_0$

Bit OEA in TOCR of FRT0 and bit $P6_1$DDR select the pin function as follows

OEA	0		1
$P6_1$DDR	0	1	X
Pin function	$P6_1$ input	$P6_1$ output	$FTOA_0$ output

$P6_0$/FTCI

$P6_0$DDR	0	1
Pin function	$P6_0$ input	$P6_0$ output
	FTCI input	

FTCI input is usable by either FRT0 or FRT1 when bits CKS2 to CKS0 in its TCR select an external clock source

X: 0 and 1 settings both give the same pin function.

8.8 Port 7

8.8.1 Overview

Port 7 is an 8-bit input port that also provides the analog input pins for the A/D converter module. The pin functions are the same in expanded and single-chip modes. Port 7 is an 8-bit input port that also provides the analog input pins for the A/D converter module. The pin functions are the same in all modes. Figure 8.17 shows the pin configuration of port 7.

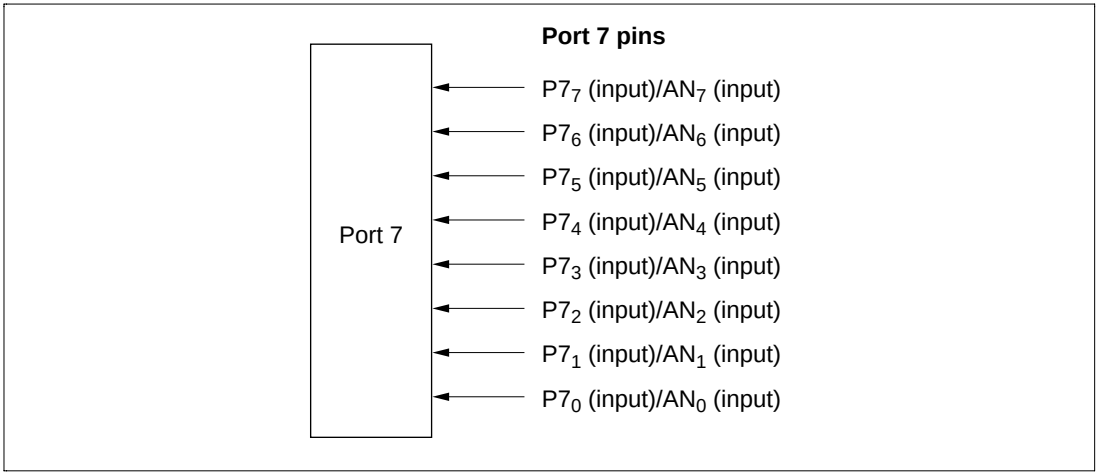


Figure 8.17 Port 7 Pin Configuration

8.8.2 Register Configuration and Descriptions

Table 8.16 summarizes the port 7 registers. Port 7 is an input port, so there is no data direction register.

Table 8.16 Port 7 Register

Name	Abbreviation	Read/Write	Initial Value	Address
Port 7 data register	P7DR	R	Undetermined	H'FFBE

Port 7 Data Register (P7DR)

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

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

When P7DR is read, the pin states are always read.

8.9 Port 8

8.9.1 Overview

Port 8 is a 7-bit input/output port that is multiplexed with dual-port RAM (DPRAM) input/output pins (RS_2 to RS_0, $\overline{XOE}$, $\overline{XWE}$, $\overline{WRQ}$, $\overline{XRDY}$), with input/output pins (TxD_1, RxD_1, SCK_1) of serial communication interface 1, and with interrupt input pins ($\overline{IRQ}_5$ to $\overline{IRQ}_3$). The functions of pins $P8_5$ and $P8_2$ to $P8_0$ are the same in all modes. The functions of pins $P8_6$, $P8_4$, and $P8_3$ differ depending on the operating mode. Figure 8.18 shows the pin configuration of port 8.

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

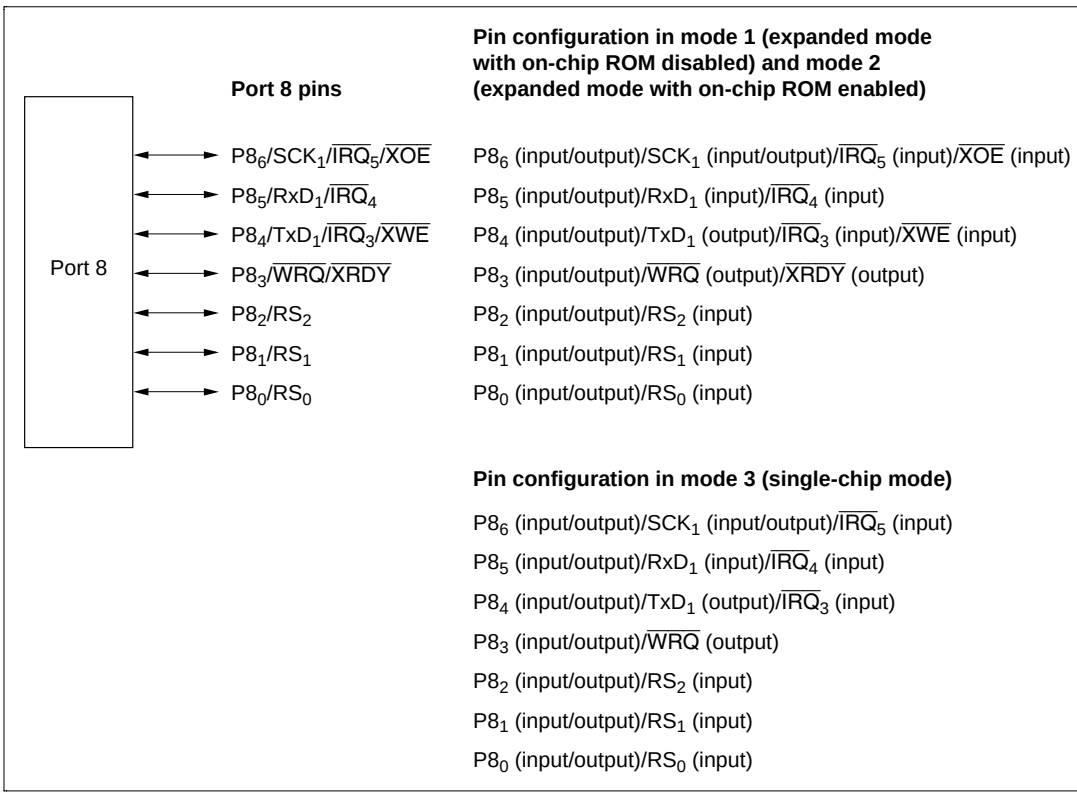


Figure 8.18 Port 8 Pin Configuration

8.9.2 Register Configuration and Descriptions

Table 8.17 summarizes the port 8 registers.

Table 8.17 Port 8 Registers

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

Port 8 Data Direction Register (P8DDR)

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

P8DDR is an 8-bit register that controls the input/output direction of each pin in port 8. A pin functions as an output pin if the corresponding P8DDR bit is set to 1, and as an input pin if this bit is cleared to 0.

P8DDR is a write-only register. Read data is invalid. If read, all bits always read 1. Bit 7 is a reserved bit that always reads 1.

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

If a transition to software standby mode occurs while port 8 is being used by an on-chip supporting module (for example, for SCI input/output), the on-chip supporting module will be initialized, so the pin will revert to general-purpose input/output, controlled by P8DDR and P8DR.

Port 8 Data Register (P8DR)

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

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

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

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

8.9.3 Pin Functions in Each Mode

Port 8 has different pin functions in different modes. A separate description for each mode is given below.

Pin Functions in Modes 1 and 2: In mode 1 (expanded mode with on-chip ROM disabled) and mode 2 (expanded mode with on-chip ROM enabled), port 8 is multiplexed with dual-port RAM (DPRAM) input/output pins (RS_2 to RS_0, $\overline{XOE}$, $\overline{XWE}$, $\overline{WRQ}$, $\overline{XRDY}$), with input/output pins (TxD_1, RxD_1, SCK_1) of serial communication interface 1 (SCI1), and with interrupt input pins ($\overline{IRQ}_5$ to $\overline{IRQ}_3$). Figure 8.19 and table 8.18 indicate the pin functions in modes 1 and 2.

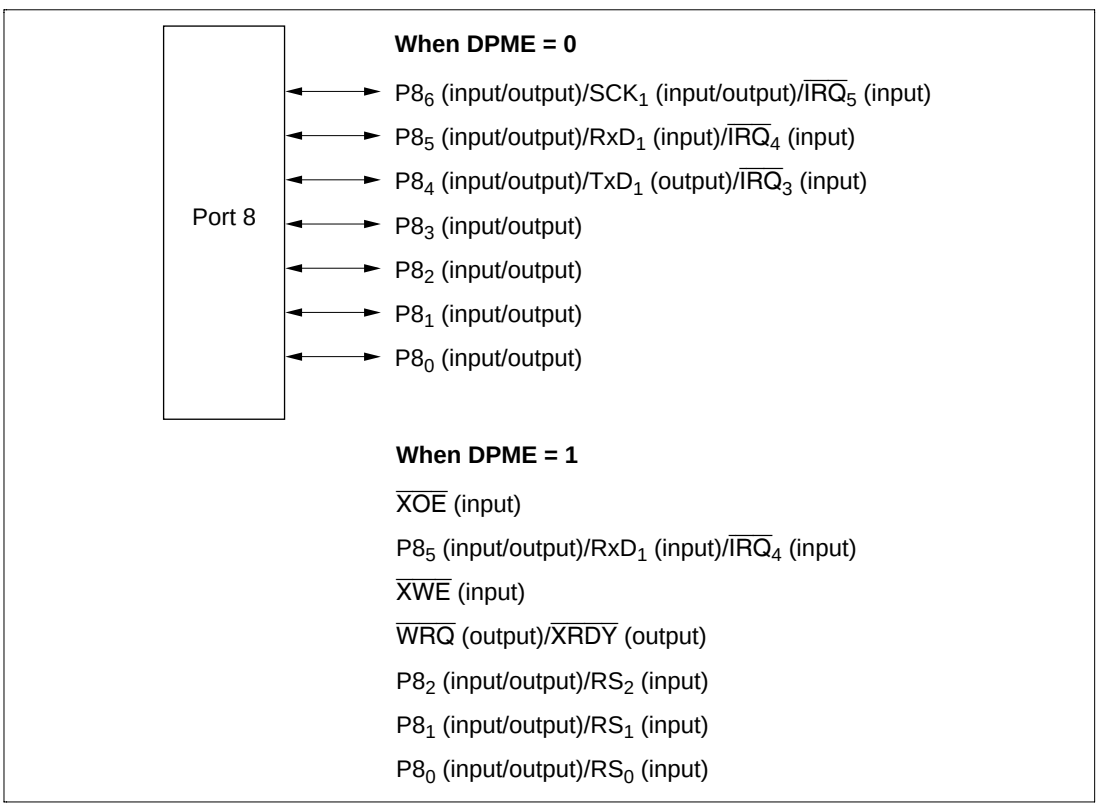


Figure 8.19 Pin Functions in Modes 1 and 2 (Port 8)

Table 8.18 Pin Functions in Modes 1 and 2

Pin	Pin Functions and Selection Method
$P8_6$/SCK_1/ $\overline{IRQ_5}$/$\overline{XOE}$	Bit DPME in SYSCR, bit C/$\overline{A}$ in SMR of SCI1, bits CKE0 and CKE1 in SCR of SCI1, and bit $P8_6$DDR select the pin function as follows

DPME	0					1
CKE1	0				1	X
C/$\overline{A}$	0			1	X	X
CKE0	0		1	X	X	X
$P8_6$DDR	0	1	X	X	X	X
Pin function	$P8_6$ input	$P8_6$ output	SCK_1 output	SCK_1 output	SCK_1 input	$\overline{XOE}$ input
	$\overline{IRQ_5}$ input					

$\overline{IRQ_5}$ input is usable when bit IRQ5E is set to 1 in IER.

$P8_5$/RxD_1/ $\overline{IRQ_4}$ — Bit RE in SCR of SCI1 and bit $P8_5$DDR select the pin function as follows

RE	0		1
$P8_5$DDR	0	1	X
Pin function	$P8_5$ input	$P8_5$ output	RxD_1 input
	$\overline{IRQ_4}$ input		

$\overline{IRQ_4}$ input is usable when bit IRQ4E is set to 1 in IER.

$P8_4$/TxD_1/ $\overline{IRQ_3}$/$\overline{XWE}$ — Bit DPME in SYSCR, bit TE in SCR of SCI1, and bit $P8_4$DDR select the pin function as follows

DPME	0			1
TE	0		1	X
$P8_4$DDR	0	1	X	X
Pin function	$P8_4$ input	$P8_4$ output	TxD_1 output	$\overline{XWE}$ input
	$\overline{IRQ_3}$ input			

$\overline{IRQ_3}$ input is usable when bit IRQ3E is set to 1 in IER.

X: 0 and 1 settings both give the same pin function.

Table 8.18 Pin Functions in Modes 1 and 2 (cont)

Pin	Pin Functions and Selection Method
$P8_3$/$\overline{WRQ}$/ $\overline{XRDY}$	Bit DPME in SYSCR, bit EWRQ in PCCSR of the PBI, and bit $P8_3$DDR select the pin function as follows

DPME	0		1	
EWRQ	X		0	1
$P8_3$DDR	0	1	X	X
Pin function	$P8_3$ input	$P8_3$ output	$\overline{XRDY}$ output	$\overline{WRQ}$ output

$P8_2$/RS_2	Bit DPME in SYSCR, bit HSCE in IOCR of the PBI, and bit $P8_2$DDR select the pin function as follows

DPME	0		1		
HSCE	X		1		0
$P8_2$DDR	0	1	0	1	X
Pin function	$P8_2$ input	$P8_2$ output	$P8_2$ input	$P8_2$ output	RS_2 input

$P8_1$/RS_1	Bit DPME in SYSCR, bit HSCE in IOCR of the PBI, and bit $P8_1$DDR select the pin function as follows

DPME	0		1		
HSCE	X		1		0
$P8_1$DDR	0	1	0	1	X
Pin function	$P8_1$ input	$P8_1$ output	$P8_1$ input	$P8_1$ output	RS_1 input

$P8_0$/RS_0	Bit DPME in SYSCR, bit HSCE in IOCR of the PBI, and bit $P8_0$DDR select the pin function as follows

DPME	0		1		
HSCE	X		1		0
$P8_0$DDR	0	1	0	1	X
Pin function	$P8_0$ input	$P8_0$ output	$P8_0$ input	$P8_0$ output	RS_0 input

X: 0 and 1 settings both give the same pin function.

Pin Functions in Mode 3: In mode 3 (single-chip mode), port 8 is multiplexed with dual-port RAM (DPRAM) input/output pins (RS_2 to RS_0, $\overline{WRQ}$), with input/output pins (TxD_1, RxD_1, SCK_1) of serial communication interface 1 (SCI1), and with interrupt input pins ($\overline{IRQ}_5$ to $\overline{IRQ}_3$). Figure 8.20 and table 8.19 indicate the pin functions in mode 3.

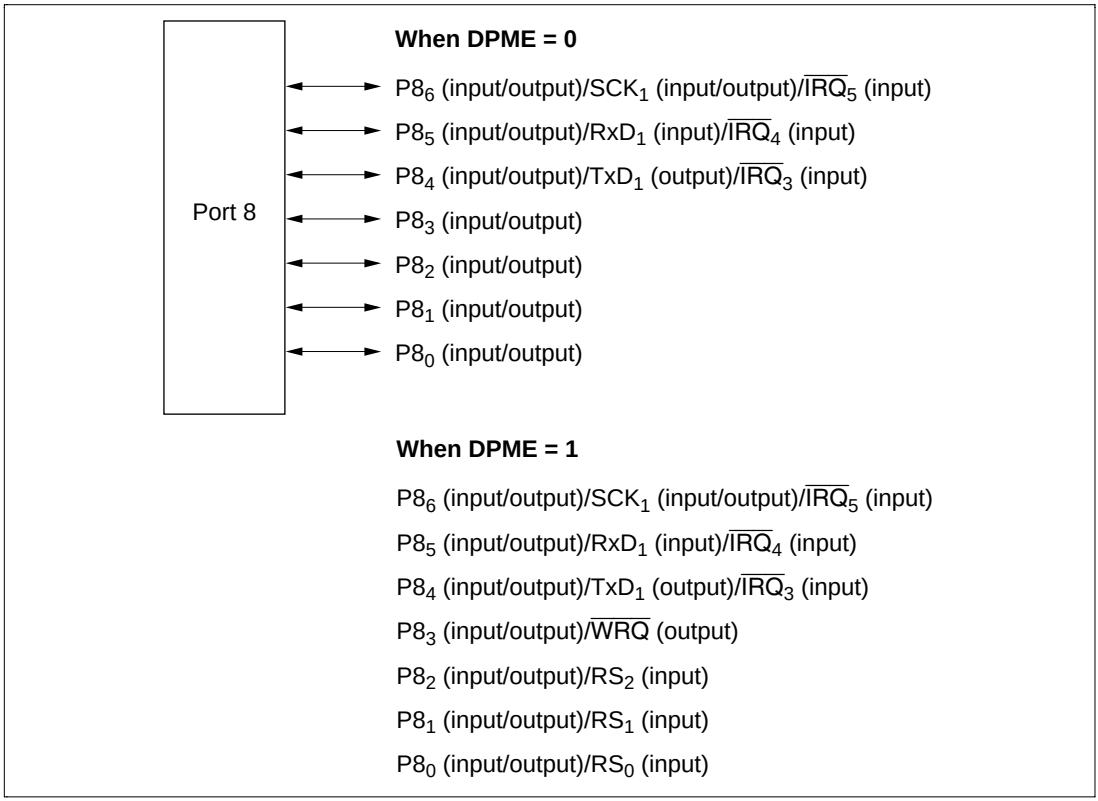


Figure 8.20 Pin Functions in Mode 3 (Port 8)

Table 8.19 Pin Functions in Mode 3

Pin	Pin Functions and Selection Method
$P8_6$/SCK_1/ $\overline{IRQ_5}$	Bit C/$\overline{A}$ in SMR of SCI1, bits CKE0 and CKE1 in SCR of SCI1, and bit $P8_6$DDR select the pin function as follows

CKE1	0				1
C/$\overline{A}$	0			1	X
CKE0	0		1	X	X
$P8_6$DDR	0	1	X	X	X
Pin function	$P8_6$ input	$P8_6$ output	SCK_1 output	SCK_1 output	SCK_1 input
	$\overline{IRQ_5}$ input				

$\overline{IRQ_5}$ input is usable when bit IRQ5E is set to 1 in IER.

$P8_5$/RxD_1/ $\overline{IRQ_4}$	Bit RE in SCR of SCI1 and bit $P8_5$DDR select the pin function as follows

RE	0		1
$P8_5$DDR	0	1	X
Pin function	$P8_5$ input	$P8_5$ output	RxD_1 input
	$\overline{IRQ_4}$ input		

$\overline{IRQ_4}$ input is usable when bit IRQ4E is set to 1 in IER.

$P8_4$/TxD_1/ $\overline{IRQ_3}$	Bit TE in SCR of SCI1 and bit $P8_4$DDR select the pin function as follows

TE	0		1
$P8_4$DDR	0	1	X
Pin function	$P8_4$ input	$P8_4$ output	TxD_1 output
	$\overline{IRQ_3}$ input		

$\overline{IRQ_3}$ input is usable when bit IRQ3E is set to 1 in IER.

X: 0 and 1 settings both give the same pin function.

Table 8.19 Pin Functions in Mode 3 (cont)

Pin	Pin Functions and Selection Method
$P8_3/\overline{WRQ}$	Bit DPME in SYSCR, bit EWRQ in PCCSR of the PBI, and bit $P8_3$DDR select the pin function as follows

DPME	0		1		
EWRQ	X		0		1
$P8_3$DDR	0	1	0	1	X
Pin function	$P8_3$ input	$P8_3$ output	$P8_3$ input	$P8_3$ output	$\overline{WRQ}$ output

$P8_2/RS_2$	Bit DPME in SYSCR, bit HSCE in IOCR of the PBI, and bit $P8_2$DDR select the pin function as follows

DPME	0		1		
HSCE	X		1		0
$P8_2$DDR	0	1	0	1	X
Pin function	$P8_2$ input	$P8_2$ output	$P8_2$ input	$P8_2$ output	RS_2 input

$P8_1/RS_1$	Bit DPME in SYSCR, bit HSCE in IOCR of the DTC, and bit $P8_1$DDR select the pin function as follows

DPME	0		1		
HSCE	X		1		0
$P8_1$DDR	0	1	0	1	X
Pin function	$P8_1$ input	$P8_1$ output	$P8_1$ input	$P8_1$ output	RS_1 input

$P8_0/RS_0$	Bit DPME in SYSCR, bit HSCE in IOCR of the PBI, and bit $P8_0$DDR select the pin function as follows

DPME	0		1		
HSCE	X		1		0
$P8_0$DDR	0	1	0	1	X
Pin function	$P8_0$ input	$P8_0$ output	$P8_0$ input	$P8_0$ output	RS_0 input

X: 0 and 1 settings both give the same pin function.

8.10 Port 9

8.10.1 Overview

Port 9 is an 8-bit input/output port that is multiplexed with interrupt input pins($\overline{IRQ_0}$ to $\overline{IRQ_2}$), input/output pins for bus control signals ($\overline{RD}$, $\overline{WR}$, $\overline{AS}$, $\overline{WAIT}$), input/output pins ($\overline{CS}$, $\overline{OE}$, $\overline{RDY}$, $\overline{WE}$, $\overline{XCS}$) for the dual-port RAM (DPRAM), an input pin ($\overline{ADTRG}$) for the A/D converter, and an output pin (ø) for the system clock. Pins $P9_2$ and $P9_0$ have the same functions in all modes. The functions of pins $P9_7$ to $P9_3$ and $P9_1$ differ depending on the operating mode. Figure 8.21 shows the pin configuration of port 9.

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

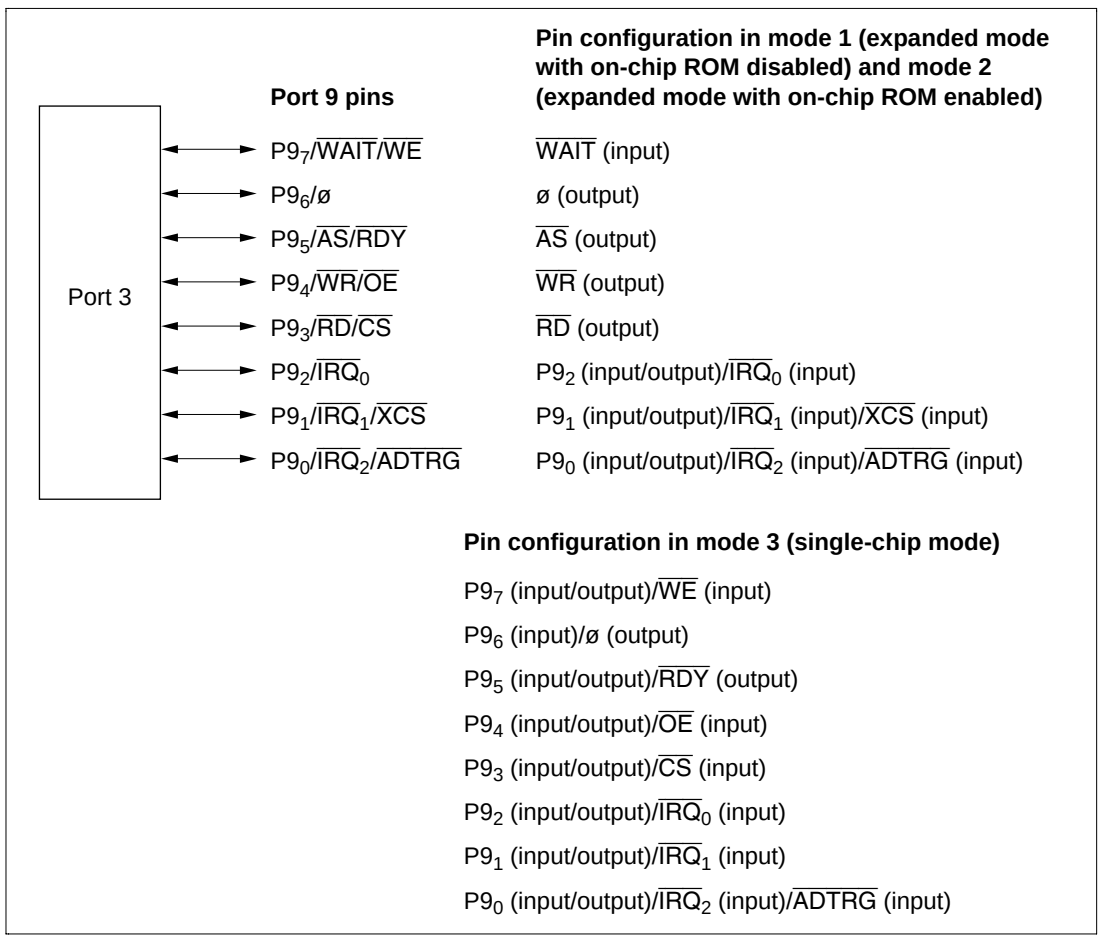


Figure 8.21 Port 9 Pin Configuration

8.10.2 Register Configuration and Descriptions

Table 8.20 summarizes the port 9 registers.

Table 8.20 Port 9 Registers

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

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

Port 9 Data Direction Register (P9DDR)

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

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

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

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

Port 9 Data Register (P9DR)

Bit	7	6	5	4	3	2	1	0
	$P9_7$	$P9_6$	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
Initial value	0	*	0	0	0	0	0	0
Read/Write	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the level at pin $P9_6$.

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

P9DR is initialized to H'00 by a reset and in hardware standby mode. In software standby mode it retains its existing values.

8.10.3 Pin Functions in Each Mode

Port 9 has one set of pin functions in modes 1 and 2, and a different set of pin functions in mode 3. The pins are multiplexed with $\overline{IRQ}_0$ to $\overline{IRQ}_2$ input, bus control signal input/output, DPRAM input/output, A/D converter input, and system clock (ø) output. Table 8.21 indicates the pin functions of port 9.

Table 8.21 Port 9 Pin Functions

Pin	Pin Functions and Selection Method
$P9_7$/$\overline{WE}$/$\overline{WAIT}$	Bit DPME in SYSCR, bit $P9_7$DDR, and the operating mode select the pin function as follows

Operating mode	Modes 1 and 2	Mode 3		
DPME	X	0		1
$P9_7$DDR	X	0	1	X
Pin function	$\overline{WAIT}$ input	$P9_7$ input	$P9_7$ output	$\overline{WE}$ input

$P9_6$/ø	Bit $P9_6$DDR and the operating mode select the pin function as follows

Operating mode	Modes 1 and 2	Mode 3	
$P9_6$DDR	Always 1	0	1
Pin function	ø output	$P9_6$ input	ø output

$P9_5$/$\overline{RDY}$/$\overline{AS}$	Bit DPME in SYSCR, bit $P9_5$DDR, and the operating mode select the pin function as follows

Operating mode	Modes 1 and 2	Mode 3		
DPME	X	0		1
$P9_5$DDR	X	0	1	X
Pin function	$\overline{AS}$ output	$P9_5$ input	$P9_5$ output	$\overline{RDY}$ output*

Note: * NMOS open-drain output. Connect an external pull-up resistor.

$P9_4$/$\overline{OE}$/$\overline{WR}$	Bit DPME in SYSCR, bit $P9_4$DDR, and the operating mode select the pin function as follows

Operating mode	Modes 1 and 2	Mode 3		
DPME	X	0		1
$P9_4$DDR	X	0	1	X
Pin function	$\overline{WR}$ output	$P9_4$ input	$P9_4$ output	$\overline{OE}$ input

X: 0 and 1 settings both give the same pin function.

Table 8.21 Port 9 Pin Functions (cont)

Pin	Pin Functions and Selection Method
$P9_3/\overline{CS}/\overline{RD}$	Bit DPME in SYSCR, bit HSCE in IOCR of the PBI, bit $P9_3$DDR, and the operating mode select the pin function as follows

Operating mode	Modes 1 and 2	Mode 3				
DPME	X	0		1		
HSCE	X	X		1		0
$P9_3$DDR	X	0	1	0	1	X
Pin function	$\overline{RD}$ output	$P9_3$ input	$P9_3$ output	$P9_3$ input	$P9_3$ output	$\overline{CS}$ input

$P9_2/\overline{IRQ}_0$

$P9_2$DDR	0	1
Pin function	$P9_2$ input	$P9_2$ output
	$\overline{IRQ}_0$ input	

$\overline{IRQ}_0$ input can be used when bit IRQ0E is set to 1 in IER.

$P9_1/\overline{IRQ}_1/\overline{XCS}$ Bit DPME in SYSCR, bit $P9_1$DDR, and the operating mode select the pin function as follows

Operating mode	Modes 1 and 2			Mode 3	
DPME	0		1	X	
$P9_1$DDR	0	1	X	0	1
Pin function	$P9_1$ input	$P9_1$ output	$\overline{XCS}$ input	$P9_1$ input	$P9_1$ output
	$\overline{IRQ}_1$ input			$\overline{IRQ}_1$ input	

$\overline{IRQ}_1$ input can be used when bit IRQ1E is set to 1 in IER.

$P9_0/\overline{IRQ}_2/\overline{ADTRG}$

$P9_0$DDR	0	1
Pin function	$P9_0$ input	$P9_0$ output
	$\overline{IRQ}_2$ input and ADTRG input	

$\overline{IRQ}_2$ input can be used when bit IRQ2E is set to 1 in IER.
$\overline{ADTRG}$ input can be used when bit TRGE is set to 1 in ADCR.

X: 0 and 1 settings both give the same pin function.

8.11 Application Notes

8.11.1 Processing when Ports are Not Used

When a port is not used, designate it as an input port and pull each pin in the port either up or down individually.

If a number of unused pins are pulled up or down with a single resistor, in the event of chip malfunction the pins will go to the output state, possibly resulting in collisions between outputs.

Section 9 16-Bit Free-Running Timer 0

9.1 Overview

The H8/3318 has an on-chip 16-bit free-running timer (FRT) with two channels: FRT0 and FRT1. Each channel is based on a 16-bit free-running counter (FRC) and can generate two independent output waveforms, measure input pulse widths, or measure external clock periods.

This section describes FRT0. For FRT1, see section 10, 16-Bit Free-Running Timer 1.

The differences between FRT0 and FRT1 are that FRT0 has four input-capture lines and four input-capture interrupt sources, and allows buffering to be designated. FRT1 has only one input-capture line and one input-capture interrupt source, and does not support buffering.

9.1.1 Features

The features of FRT0 are listed below.

- Selection of four clock sources
  The free-running counter can be driven by an internal clock source ($ø_p/2$, $ø_p/8$, or $ø_p/32$), or an external clock input (enabling use as an external event counter).
- Two independent comparators
  Each comparator can generate an independent waveform.
- Four input capture channels
  The current count can be captured on the rising or falling edge (selectable) of an input signal. The four input capture registers can be used separately, or in a buffer mode.
- Counter can be cleared under program control
  The free-running counters can be cleared on compare-match A.
- Seven independent interrupts
  Compare-match A and B, input capture A to D, and overflow interrupts are requested independently.

9.1.2 Block Diagram

Figure 9.1 shows a block diagram of free-running timer 0.

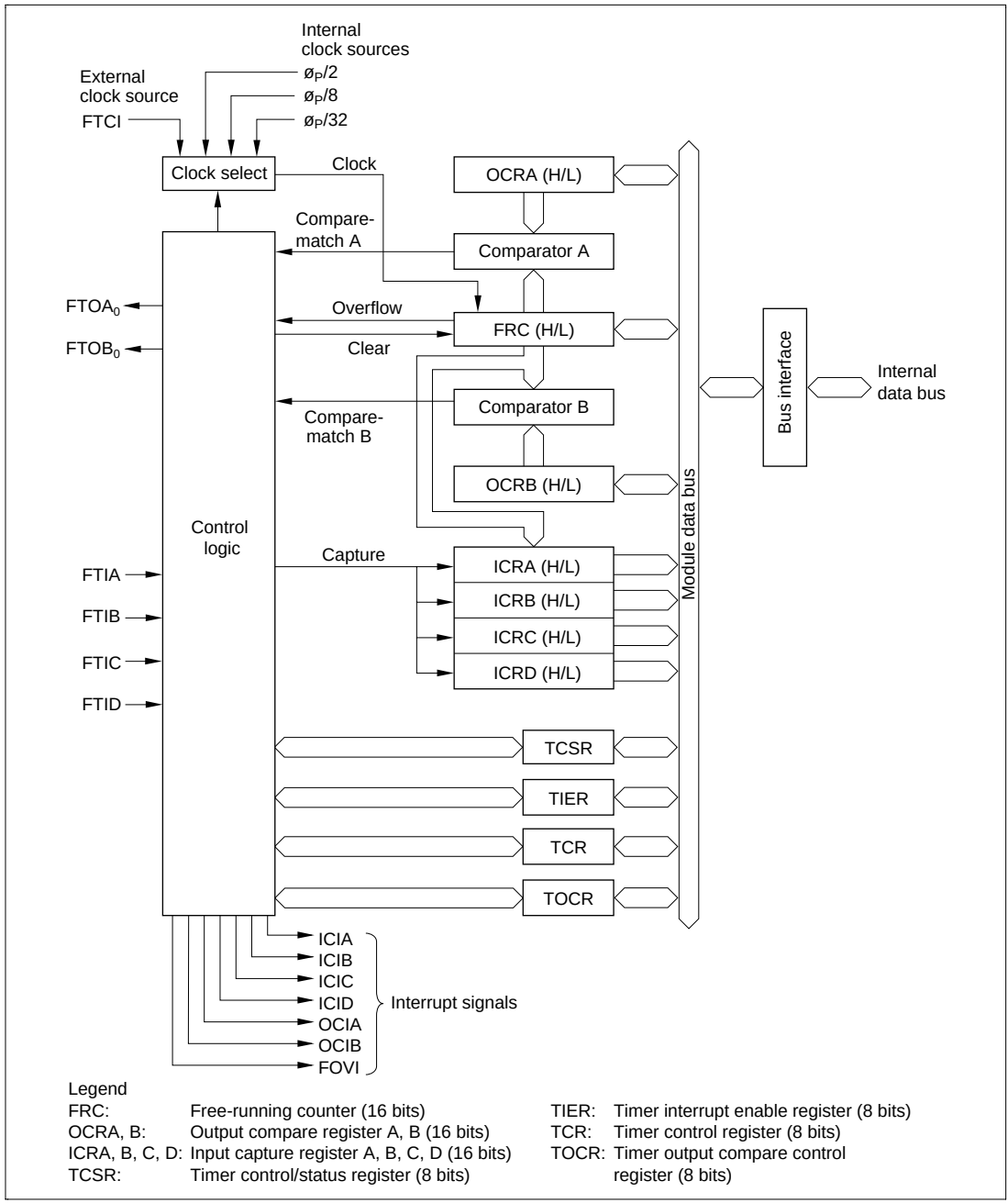


Figure 9.1 Block Diagram of 16-Bit Free-Running Timer 0

9.1.3 Input and Output Pins

Table 9.1 lists the input and output pins of free-running timer 0.

Table 9.1 FRT0 Input and Output Pins

Name	Abbreviation	I/O	Function
Counter clock input	FTCI	Input	Input of external free-running counter clock signal
Output compare A	$FTOA_0$*	Output	Output controlled by comparator A
Output compare B	$FTOB_0$*	Output	Output controlled by comparator B
Input capture A	FTIA	Input	Trigger for capturing current count into input capture register A
Input capture B	FTIB	Input	Trigger for capturing current count into input capture register B
Input capture C	FTIC	Input	Trigger for capturing current count into input capture register C
Input capture D	FTID	Input	Trigger for capturing current count into input capture register D

Note: * In this manual, the channel subscripts are normally omitted.

9.1.4 Register Configuration

Table 9.2 lists the registers of free-running timer 0.

Table 9.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address
Timer interrupt enable register	TIER	R/W	H'01	H'FF90
Timer control/status register	TCSR	R/(W)*1	H'00	H'FF91
Free-running counter (high)	FRC (H)	R/W	H'00	H'FF92
Free-running counter (low)	FRC (L)	R/W	H'00	H'FF93
Output compare register A/B (high)*2	OCRA/B (H)	R/W	H'FF	H'FF94*2
Output compare register A/B (low)*2	OCRA/B (L)	R/W	H'FF	H'FF95*2
Timer control register	TCR	R/W	H'00	H'FF96
Timer output compare control register	TOCR	R/W	H'E0	H'FF97
Input capture register A (high)	ICRA (H)	R	H'00	H'FF98
Input capture register A (low)	ICRA (L)	R	H'00	H'FF99
Input capture register B (high)	ICRB (H)	R	H'00	H'FF9A
Input capture register B (low)	ICRB (L)	R	H'00	H'FF9B
Input capture register C (high)	ICRC (H)	R	H'00	H'FF9C
Input capture register C (low)	ICRC (L)	R	H'00	H'FF9D
Input capture register D (high)	ICRD (H)	R	H'00	H'FF9E
Input capture register D (low)	ICRD (L)	R	H'00	H'FF9F

Notes: 1. Software can write a 0 to clear bits 7 to 1, but cannot write a 1 in these bits.
2. OCRA and OCRB share the same addresses. Access is controlled by the OCRS bit in TOCR.

9.2 Register Descriptions

9.2.1 Free-Running Counter (FRC)

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

FRC is a 16-bit readable/writable up-counter that increments on an internal pulse generated from a clock source. The clock source is selected by the clock select 1 and 0 bits (CKS1 and CKS0) of the timer control register (TCR).

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

Because FRC is a 16-bit register, a temporary register (TEMP) is used when FRC is written or read. See section 9.3, CPU Interface, for details.

FRC is initialized to H'0000 by a reset and in the standby modes. It can also be cleared by compare-match A.

9.2.2 Output Compare Registers A and B (OCRA and OCRB)

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

OCRA and OCRB are 16-bit readable/writable registers, the contents of which are continually compared with the value in the FRC. When a match is detected, the corresponding output compare flag (OCFA or OCFB) is set in the timer control/status register (TCSR).

In addition, if the output enable bit (OEA or OEB) in the timer output compare control register (TOCR) is set to 1, when the output compare register and FRC values match, the logic level selected by the output level bit (OLVLA or OLVLB) in TOCR is output at the output compare pin (FTOA or FTOB). Following a reset, the FTOA and FTOB output levels are 0 until the first compare-match.

Because OCRA and OCRB are 16-bit registers, a temporary register (TEMP) is used for write access, as explained in section 9.3, CPU Interface.

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

9.2.3 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

Each input capture register is a 16-bit read-only register.

When the rising or falling edge of the signal at an input capture pin (FTIA to FTID) is detected, the current FRC value is copied to the corresponding input capture register (ICRA to ICRD).* At the same time, the corresponding input capture flag (ICFA to ICFD) in the timer control/status register (TCSR) is set to 1. The input capture edge is selected by the input edge select bits (IEDGA to IEDGD) in the timer control register (TCR).

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

Input capture can be buffered by using the input capture registers in pairs. When the BUFEA bit in TCR is set to 1, ICRC is used as a buffer register for ICRA as shown in figure 9.2. When an FTIA input is received, the old ICRA contents are moved into ICRC, and the new FRC count is copied into ICRA.

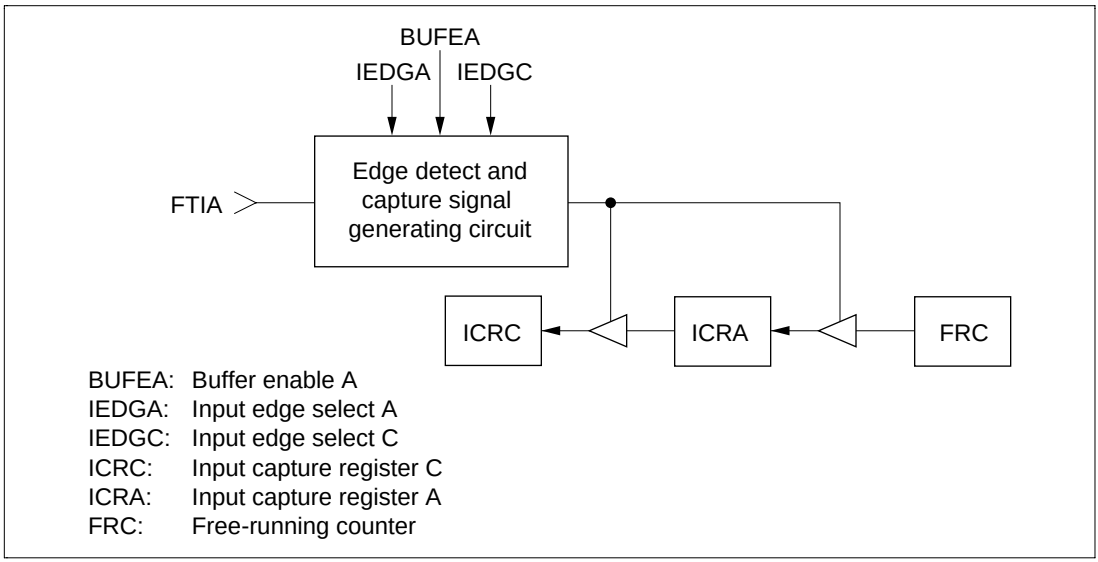


Figure 9.2 Input Capture Buffering (Example)

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

When input capture is buffered, if the two input edge bits are set to different values (IEDGA ≠ IEDGC or IEDGB ≠ IEDGD), then input capture is triggered on both the rising and falling edges of the FTIA or FTIB input signal. If the two input edge bits are set to the same value (IEDGA = IEDGC or IEDGB = IEDGD), then input capture is triggered on only one edge. See table 9.3.

Table 9.3 Buffered Input Capture Edge Selection (Example)

IEDGA	IEDGC	Input Capture Edge
0	0	Captured on falling edge of input capture A (FTIA) (Initial value)
	1	Captured on both rising and falling edges of input capture A (FTIA)
1	0	
	1	Captured on rising edge of input capture A (FTIA)

Because the input capture registers are 16-bit registers, a temporary register (TEMP) is used when they are read. See section 9.3, CPU Interface, for details.

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

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

9.2.4 Timer Interrupt Enable Register (TIER)

Bit	7	6	5	4	3	2	1	0
	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

TIER is an 8-bit readable/writable register that enables and disables interrupts. TIER is initialized to H'01 by a reset and in the standby modes.

Bit 7—Input Capture Interrupt A Enable (ICIAE): This bit selects whether to request input capture interrupt A (ICIA) when input capture flag A (ICFA) in the timer status/control register (TCSR) is set to 1.

Bit 7 ICIAE	Description	
0	Input capture interrupt request A (ICIA) is disabled	(Initial value)
1	Input capture interrupt request A (ICIA) is enabled	

Bit 6—Input Capture Interrupt B Enable (ICIBE): This bit selects whether to request input capture interrupt B (ICIB) when input capture flag B (ICFB) in TCSR is set to 1.

Bit 6 ICIBE	Description	
0	Input capture interrupt request B (ICIB) is disabled	(Initial value)
1	Input capture interrupt request B (ICIB) is enabled	

Bit 5—Input Capture Interrupt C Enable (ICICE): This bit selects whether to request input capture interrupt C (ICIC) when input capture flag C (ICFC) in TCSR is set to 1.

Bit 5 ICICE	Description	
0	Input capture interrupt request C (ICIC) is disabled	(Initial value)
1	Input capture interrupt request C (ICIC) is enabled	

Bit 4—Input Capture Interrupt D Enable (ICIDE): This bit selects whether to request input capture interrupt D (ICID) when input capture flag D (ICFD) in TCSR is set to 1.

Bit 4 ICIDE	Description	
0	Input capture interrupt request D (ICID) is disabled	(Initial value)
1	Input capture interrupt request D (ICID) is enabled	

Bit 3—Output Compare Interrupt A Enable (OCIAE): This bit selects whether to request output compare interrupt A (OCIA) when output compare flag A (OCFA) in TCSR is set to 1.

Bit 3 OCIAE	Description	
0	Output compare interrupt request A (OCIA) is disabled	(Initial value)
1	Output compare interrupt request A (OCIA) is enabled	

Bit 2—Output Compare Interrupt B Enable (OCIBE): This bit selects whether to request output compare interrupt B (OCIB) when output compare flag B (OCFB) in TCSR is set to 1.

Bit 2 OCIBE	Description	
0	Output compare interrupt request B (OCIB) is disabled	(Initial value)
1	Output compare interrupt request B (OCIB) is enabled	

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

Bit 1 OVIE	Description	
0	Timer overflow interrupt request (FOVI) is disabled	(Initial value)
1	Timer overflow interrupt request (FOVI) is enabled	

Bit 0—Reserved: This bit cannot be modified and is always read as 1.

9.2.5 Timer Control/Status Register (TCSR)

Bit	7	6	5	4	3	2	1	0
	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

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

TCSR is an 8-bit register that controls interrupt request signals and selects whether to clear the counter. TCSR is initialized to H'00 by a reset and in the standby modes. Timing is described in section 9.4, Operation.

Bit 7—Input Capture Flag A (ICFA): This status bit is set to 1 to flag an input capture A event. If BUFEA = 0, ICFA indicates that the FRC value has been copied to ICRA. If BUFEA = 1, ICFA indicates that the old ICRA value has been moved into ICRC and the new FRC value has been copied to ICRA.

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

Bit 7 ICFA	Description
0	To clear ICFA, the CPU must read ICFA after it has been set to 1, then write a 0 in this bit (Initial value)
1	This bit is set to 1 when an FTIA input signal causes the FRC value to be copied to ICRA

Bit 6—Input Capture Flag B (ICFB): This status bit is set to 1 to flag an input capture B event. If BUFEB = 0, ICFB indicates that the FRC value has been copied to ICRB. If BUFEB = 1, ICFB indicates that the old ICRB value has been moved into ICRD and the new FRC value has been copied to ICRB.

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

Bit 6 ICFB	Description
0	To clear ICFB, the CPU must read ICFB after it has been set to 1, then write a 0 in this bit (Initial value)
1	This bit is set to 1 when an FTIB input signal causes the FRC value to be copied to ICRB

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

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

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

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

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

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

Bit 3—Output Compare Flag A (OCFA): This status flag is set to 1 when the FRC value matches the OCRA value. This flag must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 3 OCFA	Description
0	To clear OCFA, the CPU must read OCFA after it has been set to 1, then write a 0 in this bit (Initial value)
1	This bit is set to 1 when FRC = OCRA

Bit 2—Output Compare Flag B (OCFB): This status flag is set to 1 when the FRC value matches the OCRB value. This flag must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 2 OCFB	Description
0	To clear OCFB, the CPU must read OCFB after it has been set to 1, then write a 0 in this bit (Initial value)
1	This bit is set to 1 when FRC = OCRB

Bit 1—Timer Overflow Flag (OVF): This status flag is set to 1 when FRC overflows (changes from H'FFFF to H'0000). This flag must be cleared by software. It is set by hardware, however, and cannot be set by software.

Bit 1 OVF	Description
0	To clear OVF, the CPU must read OVF after it has been set to 1, then write a 0 in this bit (Initial value)
1	This bit is set to 1 when FRC changes from H'FFFF to H'0000

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

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

9.2.6 Timer Control Register (TCR)

Bit	7	6	5	4	3	2	1	0
	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCR is an 8-bit readable/writable register that selects the rising or falling edge of the input capture signals, enables the input capture buffer mode, and selects the FRC clock source.

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

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

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

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

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

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

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

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

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

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

Bit 3 BUFEA	Description	
0	ICRC is used for input capture C	(Initial value)
1	ICRC is used as a buffer register for input capture A	

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

Bit 2 BUFEB	Description	
0	ICRD is used for input capture D	(Initial value)
1	ICRD is used as a buffer register for input capture B	

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

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

9.2.7 Timer Output Compare Control Register (TOCR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	OCRS	OEA	OEB	OLVLA	OLVLB
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

TOCR is an 8-bit readable/writable register that enables output from the output compare pins, selects the output levels, and switches access between output compare registers A and B.

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

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

Bit 4—Output Compare Register Select (OCRS): OCRA and OCRB share the same address. When this address is accessed, the OCRS bit selects which register is accessed. This bit does not affect the operation of OCRA or OCRB.

Bit 4 OCRS	Description	
0	OCRA is selected	(Initial value)
1	OCRB is selected	

Bit 3—Output Enable A (OEA): This bit enables or disables output of the output compare A signal (FTOA).

Bit 3 OEA	Description	
0	Output compare A output is disabled	(Initial value)
1	Output compare A output is enabled	

Bit 2—Output Enable B (OEB): This bit enables or disables output of the output compare B signal (FTOB).

Bit 2 OEB	Description	
0	Output compare B output is disabled	(Initial value)
1	Output compare B output is enabled	

Bit 1—Output Level A (OLVLA): This bit selects the logic level to be output at the FTOA pin when the FRC and OCRA values match.

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

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

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

9.3 CPU Interface

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

These registers are written and read as follows:

- Register Write

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

- Register Read

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

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

Figure 9.3 shows the data flow when FRC is accessed. The other registers are accessed in the same way. As an exception, when the CPU reads OCRA or OCRB, it reads both the upper and lower bytes directly, without using TEMP.

Coding Examples

To write the contents of general register R0 to OCRA: `MOV.W R0, @OCRA`
To transfer the contents of ICRA to general register R0: `MOV.W @ICRA, R0`

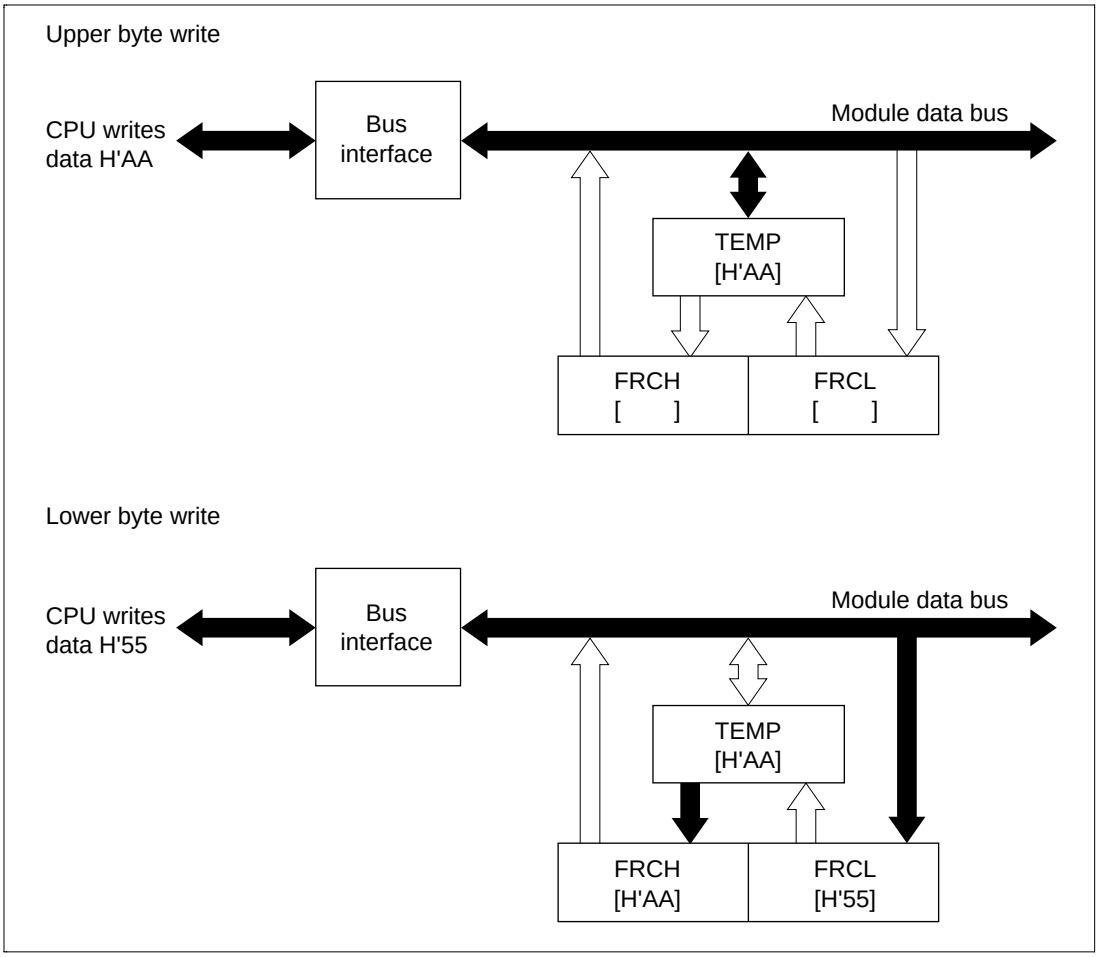


Figure 9.3a Write Access to FRC (when CPU Writes H'AA55)

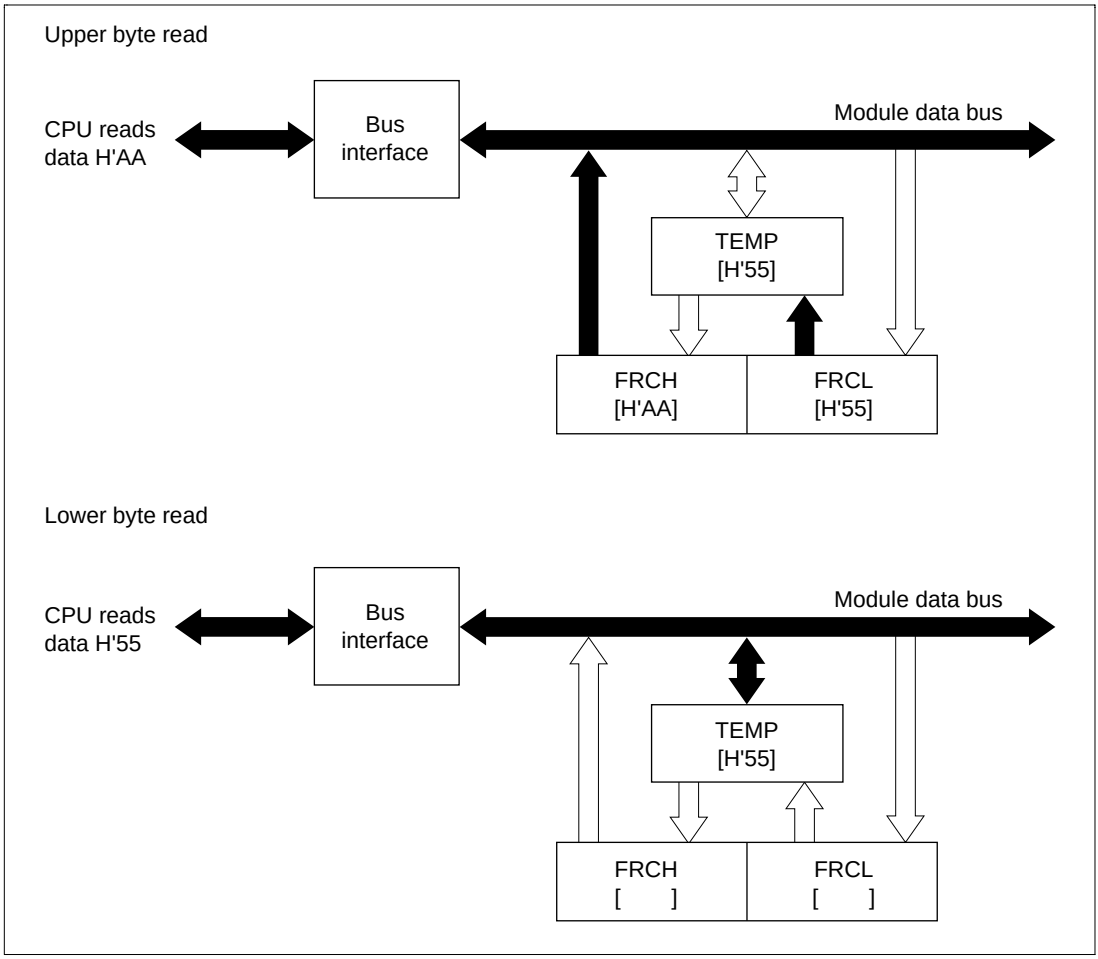


Figure 9.3b Read Access to FRC (when FRC Contains H'AA55)

9.4 Operation

9.4.1 FRC Incrementation Timing

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

Internal Clock: The internal clock sources ($ø_P/2$, $ø_P/8$, $ø_P/32$) are created from the system clock (ø) by a prescaler. FRC increments on a pulse generated from the falling edge of the prescaler output. See figure 9.4.

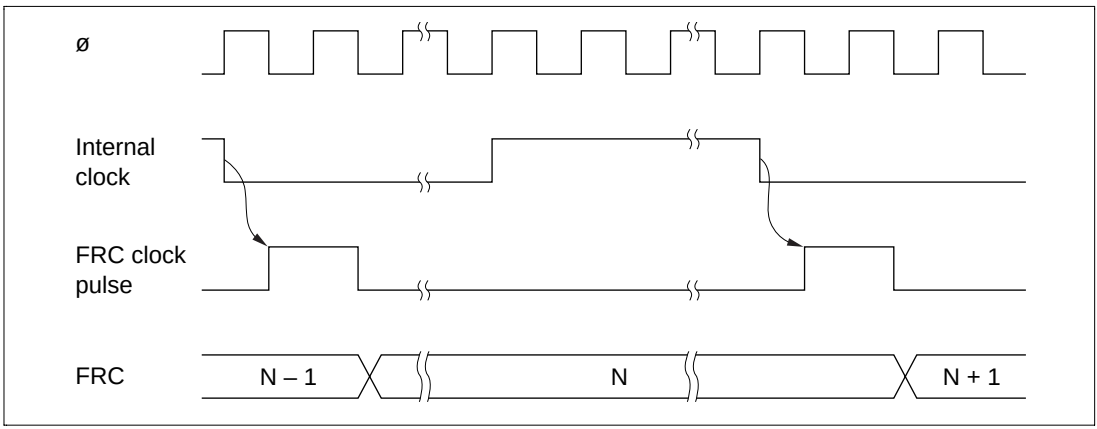


Figure 9.4 Increment Timing for Internal Clock Source

External Clock: If external clock input is selected, FRC increments on the rising edge of the FTCI clock signal. Figure 9.5 shows the increment timing.

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

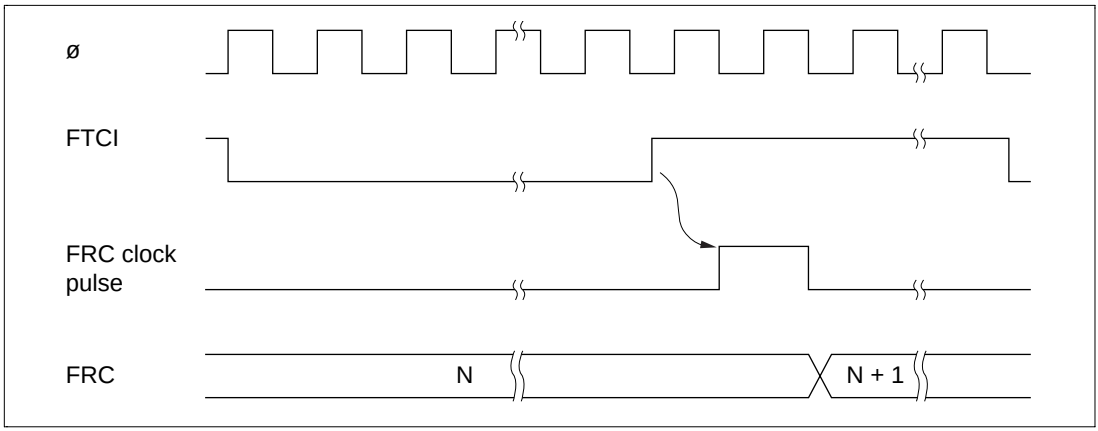


Figure 9.5 Increment Timing for External Clock Source

9.4.2 Output Compare Timing

When a compare-match occurs, the logic level selected by the output level bit (OLVLA or OLVLB) in TOCR is output at the output compare pin (FTOA or FTOB). Figure 9.6 shows the timing of this operation for compare-match A.

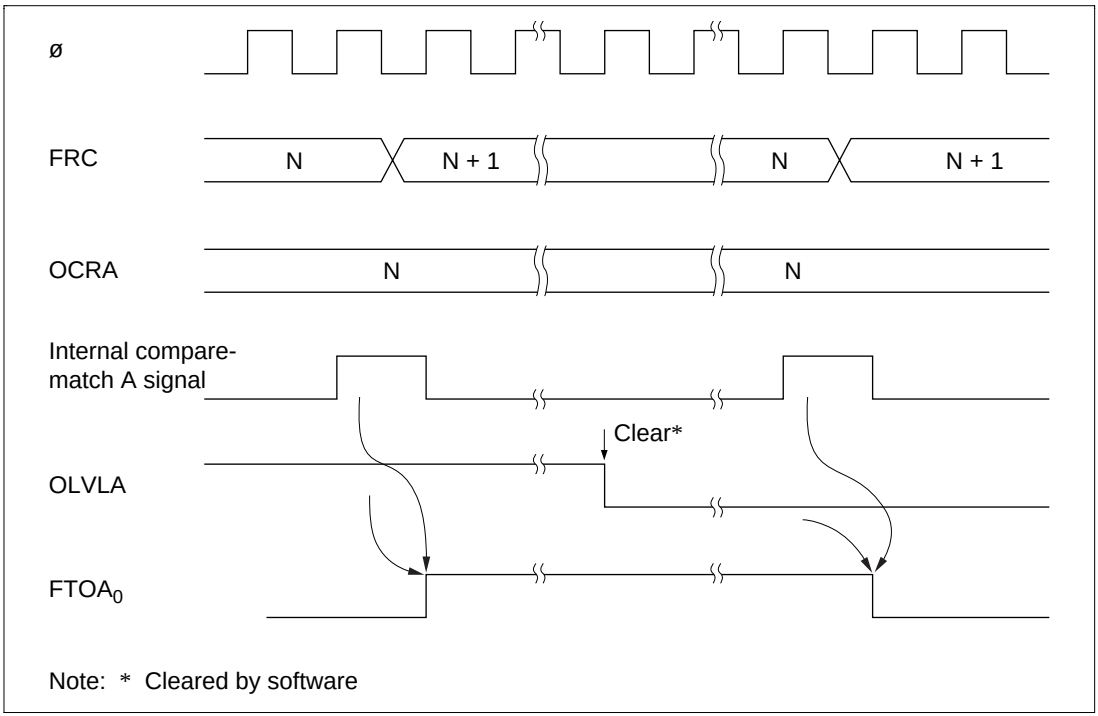


Figure 9.6 Timing of Output Compare A

9.4.3 FRC Clear Timing

If the CCLRA bit in TCSR is set to 1, FRC is cleared when compare-match A occurs. Figure 9.7 shows the timing of this operation.

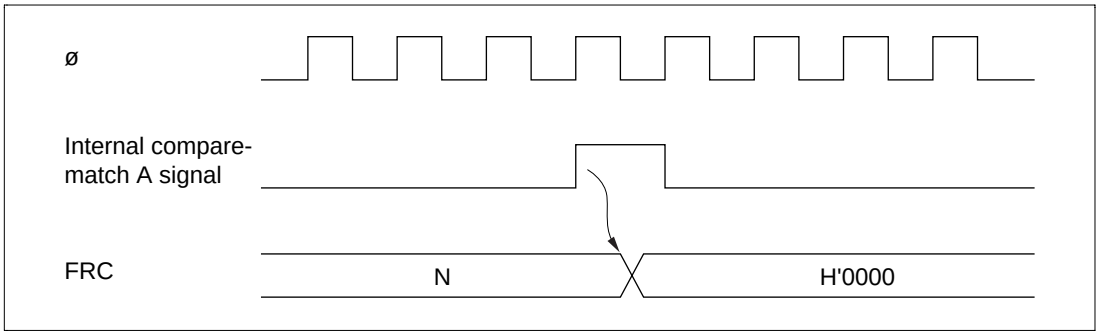


Figure 9.7 Clearing of FRC by Compare-Match A

9.4.4 Input Capture Timing

Input Capture Timing: An internal input capture signal is generated from the rising or falling edge of the signal at the input capture pin FTIx (x = A, B, C, D), as selected by the corresponding IEDGx bit in TCR. Figure 9.8 shows the usual input capture timing when the rising edge is selected (IEDGx = 1).

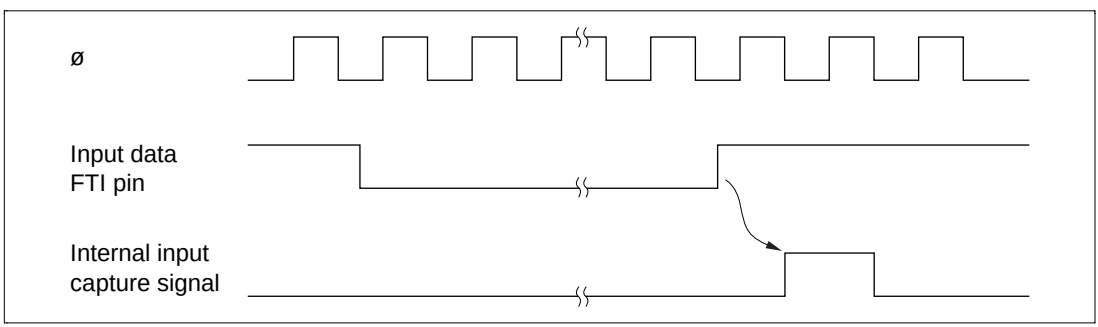


Figure 9.8 Input Capture Timing (Usual Case)

If the upper byte of ICRA/B/C/D is being read when the corresponding input capture signal arrives, the internal input capture signal is delayed by one state. Figure 9.9 shows the timing for this case.

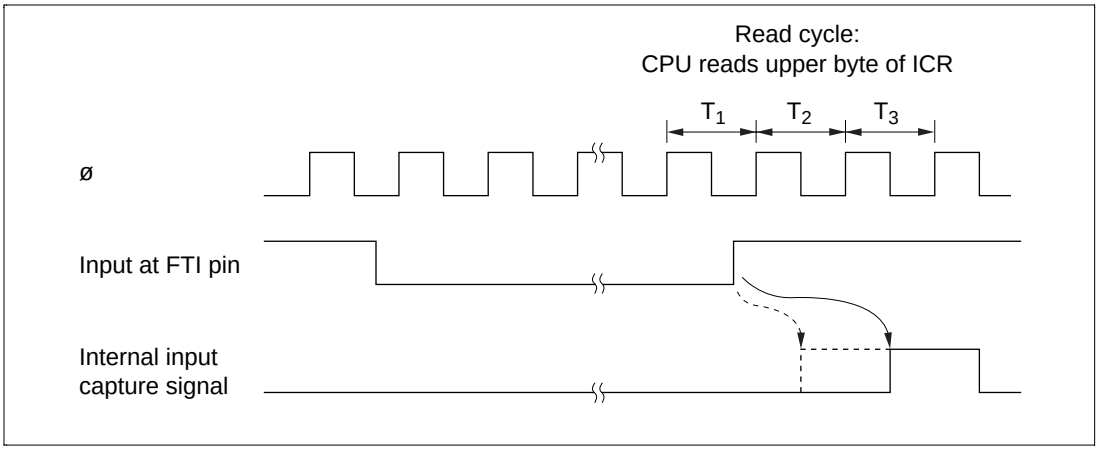


Figure 9.9 Input Capture Timing (1-State Delay due to ICRA/B/C/D Read)

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

Figure 9.10 shows how input capture operates when ICRA and ICRC are used in buffer mode and IEDGA and IEDGC are set to different values (IEDGA = 0 and IEDGC = 1, or IEDGA = 1 and IEDGC = 0), so that input capture is performed on both the rising and falling edges of FTIA.

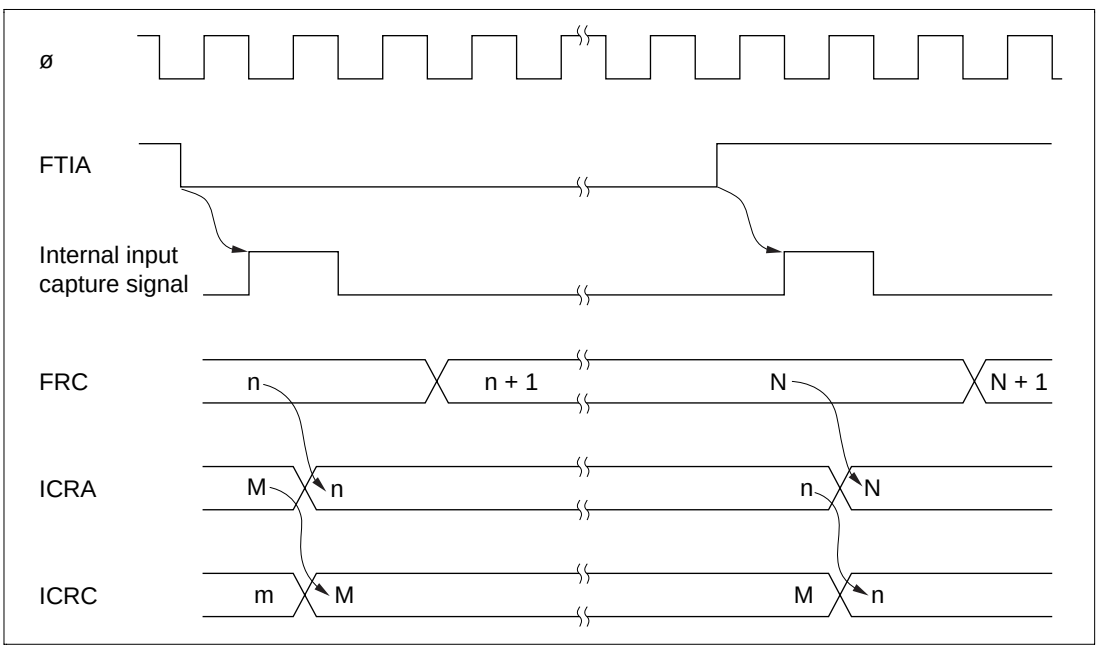


Figure 9.10 Buffered Input Capture with Both Edges Selected

When ICRC or ICRD is used as a buffer register, its input capture flag is set by the selected transition of its input capture signal. For example, if ICRC is used to buffer ICRA, when the edge transition selected by the IEDGC bit occurs on the FTIC input capture line, ICFC will be set, and if the ICIEC bit is set, an interrupt will be requested. The FRC value will not be transferred to ICRC, however.

In buffered input capture, if the upper byte of either of the two registers to which data will be transferred (ICRA and ICRC, or ICRB and ICRD) is being read when the input signal arrives, input capture is delayed by one system clock (ø). Figure 9.11 shows the timing when BUFEA = 1.

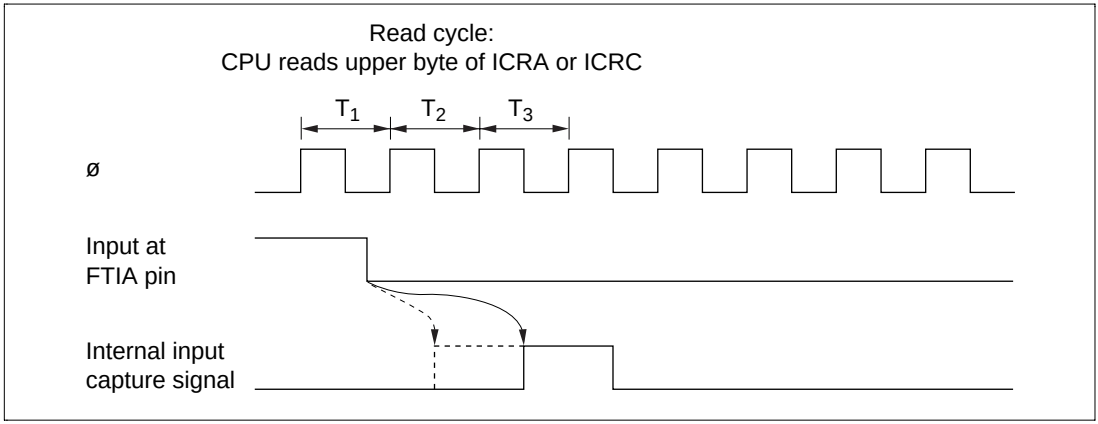


Figure 9.11 Input Capture Timing (1-State Delay, Buffer Mode)

9.4.5 Timing of Input Capture Flag (ICF) Setting

The input capture flag ICFx (x = A, B, C, D) is set to 1 by the internal input capture signal. Figure 9.12 shows the timing of this operation.

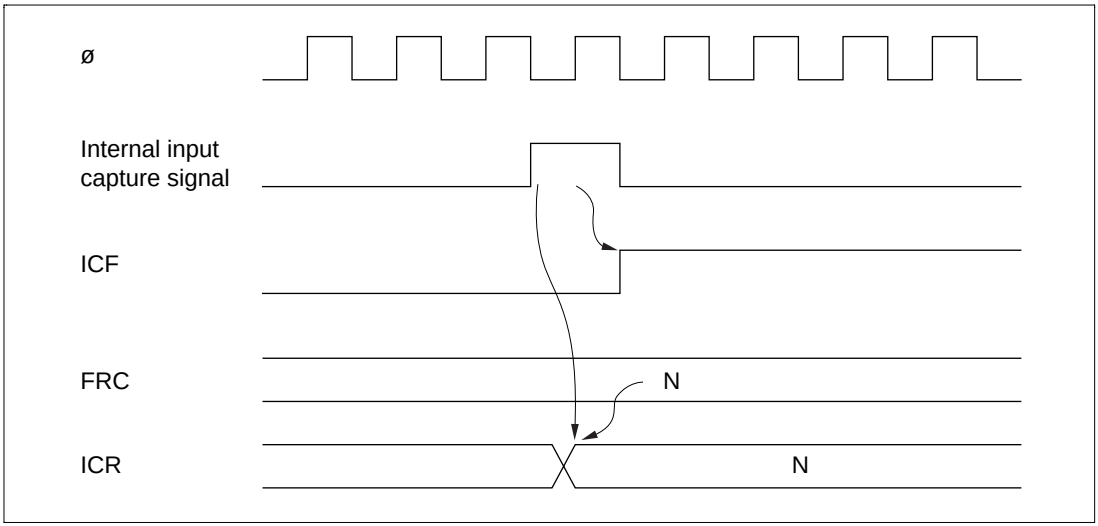


Figure 9.12 Setting of Input Capture Flag

9.4.6 Setting of Output Compare Flags A and B (OCFA and OCFB)

The output compare flags are set to “1” by an internal compare-match signal generated when the FRC value matches the OCRA or OCRB value. This compare-match signal is generated at the last state in which the two values match, just before FRC increments to a new value.

Accordingly, when the FRC and OCR values match, the compare-match signal is not generated until the next period of the clock source. Figure 9.13 shows the timing of the setting of the output compare flags.

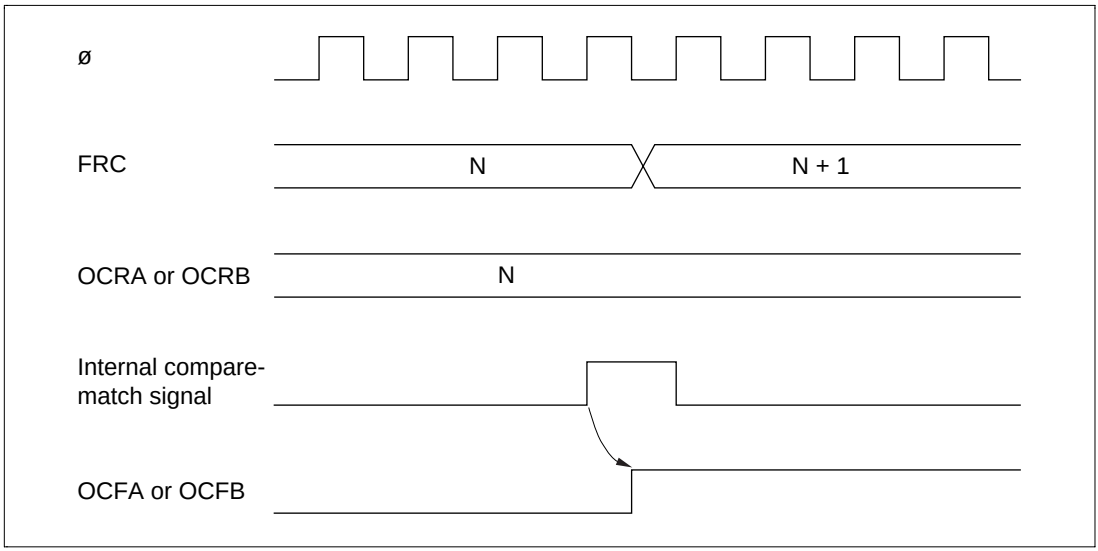


Figure 9.13 Setting of Output Compare Flags

9.4.7 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 9.14 shows the timing of this operation.

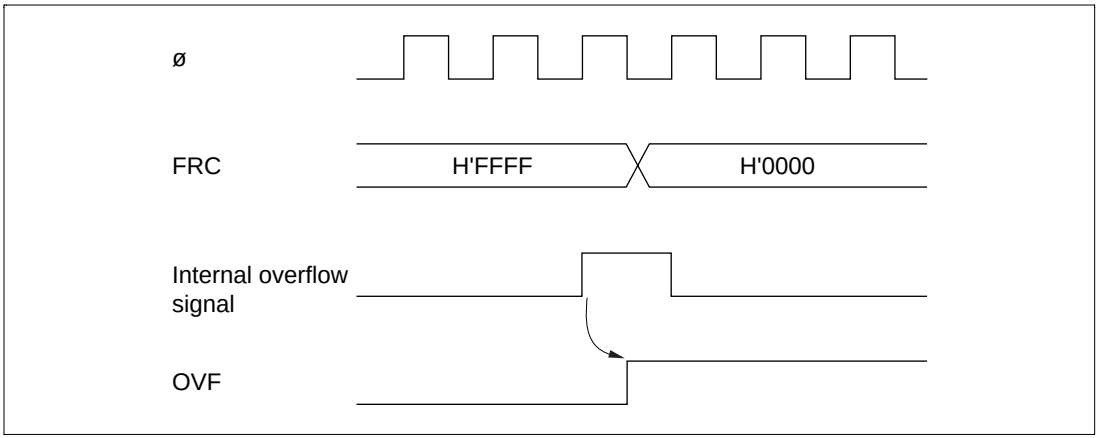


Figure 9.14 Setting of Overflow Flag (OVF)

9.5 Interrupts

FRT0 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 9.4 lists information about these interrupts.

Table 9.4 FRT0 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

9.6 Sample Application

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

1. The CCLRA bit in TCSR is set to 1.
2. Each time a compare-match interrupt occurs, software inverts the corresponding output level bit in TOCR (OLVLA or OLVLB).

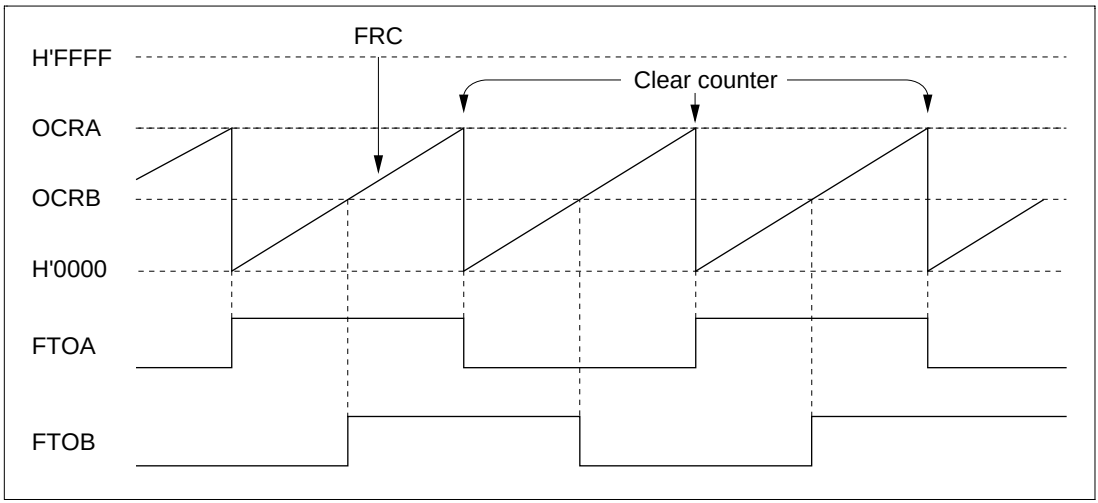


Figure 9.15 Square-Wave Output (Example)

9.7 Application Notes

Application programmers should note that the following types of contention can occur in free-running timer 0.

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

Figure 9.16 shows this type of contention.

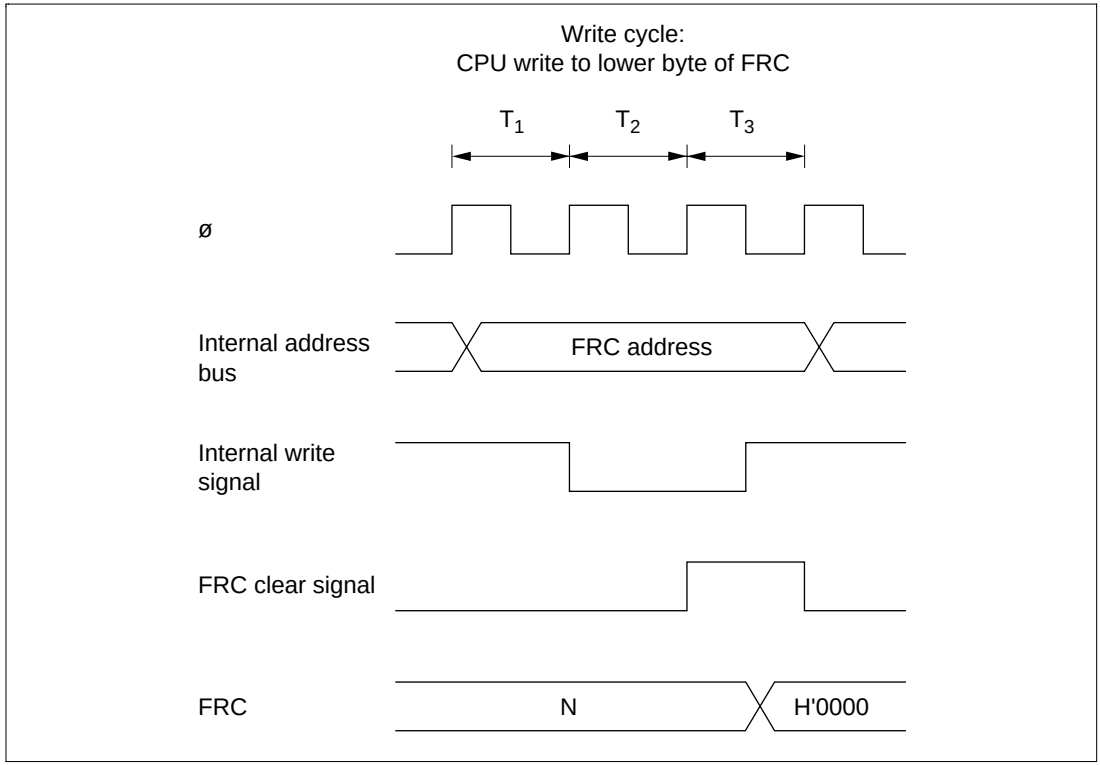


Figure 9.16 FRC Write-Clear Contention

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

Figure 9.17 shows this type of contention.

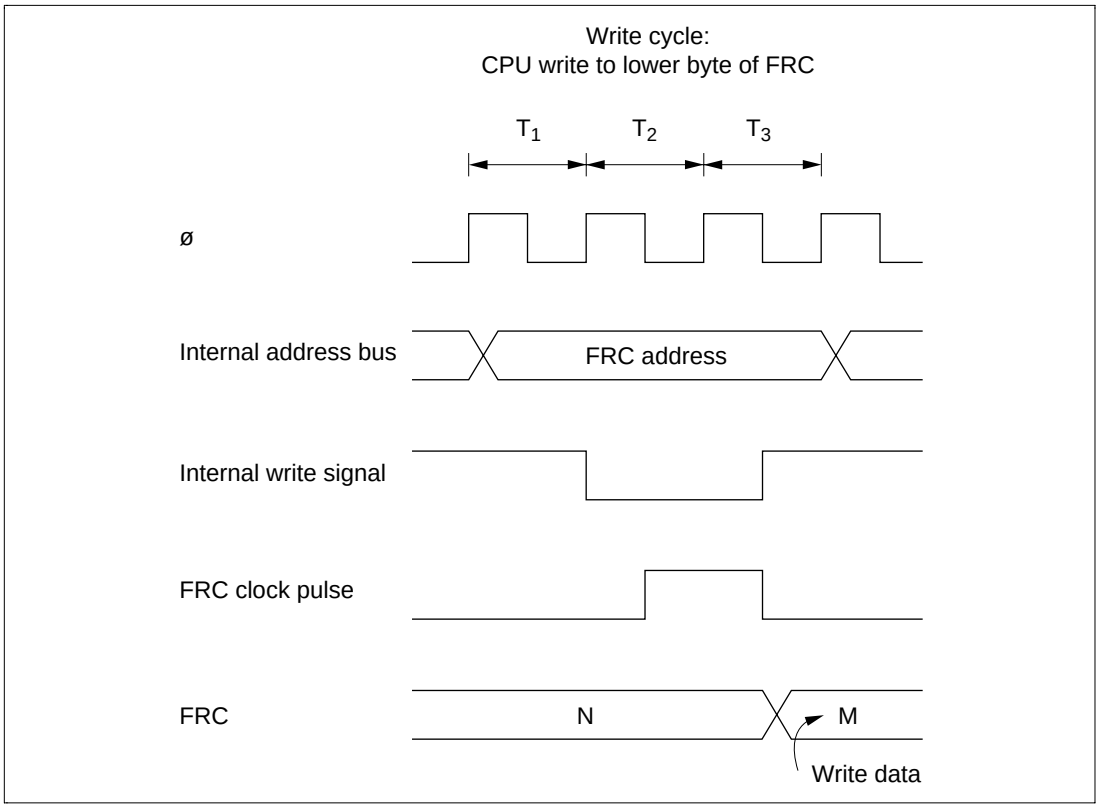


Figure 9.17 FRC Write-Increment Contention

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

Figure 9.18 shows this type of contention.

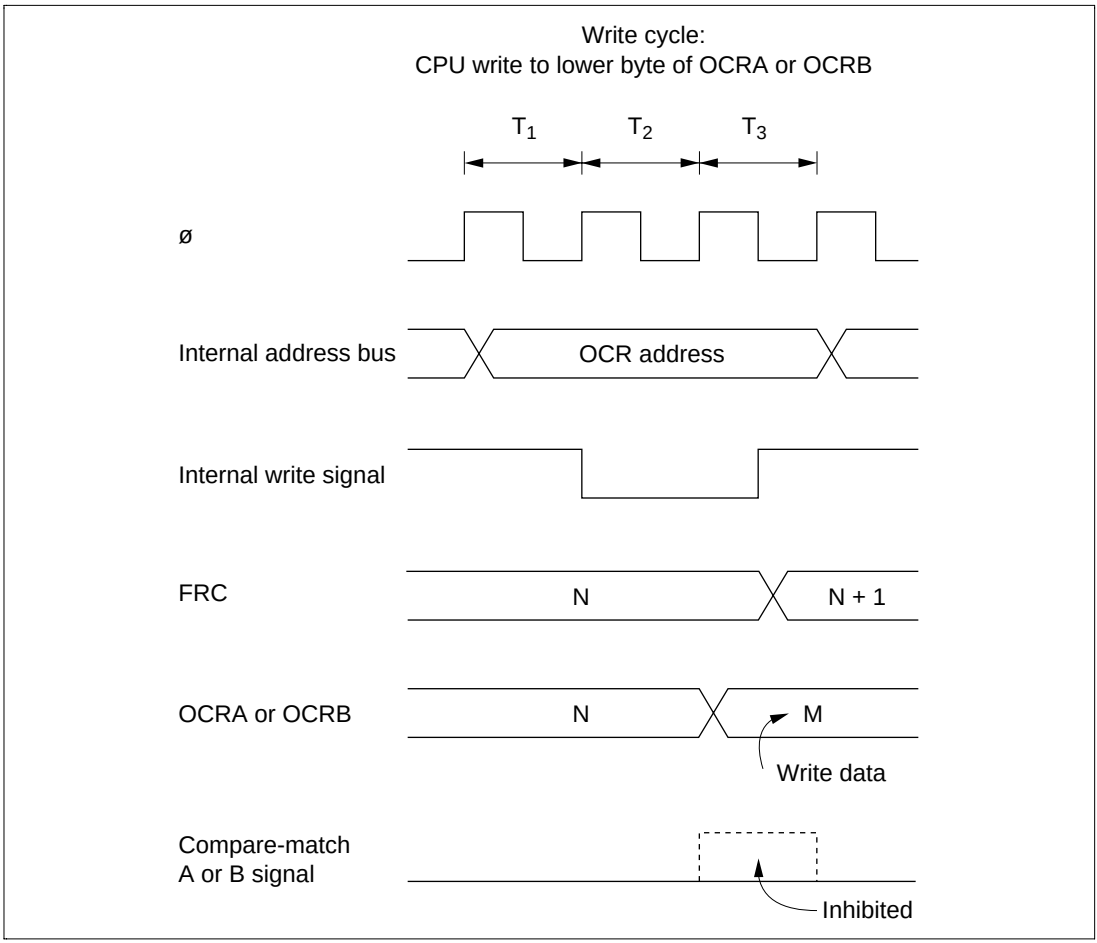


Figure 9.18 Contention between OCR Write and Compare-Match

Incrementation Caused by Changing of Internal Clock Source: When an internal clock source is changed, the changeover may cause FRC to increment. This depends on the time at which the clock select bits (CKS1 and CKS0) are rewritten, as shown in table 9.5.

The pulse that increments FRC is generated at the falling edge of the internal clock source. If clock sources are changed when the old source is high and the new source is low, as in case number 3 in table 9.5, the changeover generates a falling edge that triggers the FRC increment clock pulse.

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

Table 9.5 Effect of Changing Internal Clock Sources

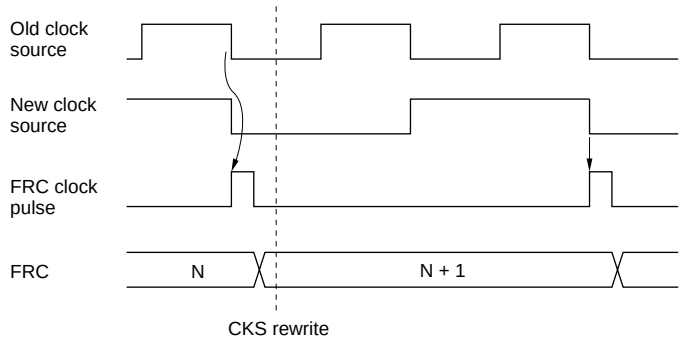
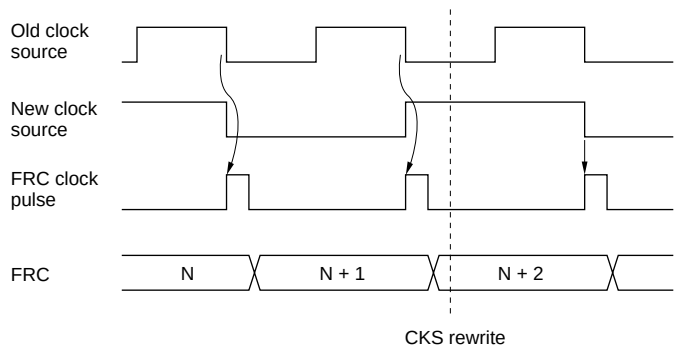
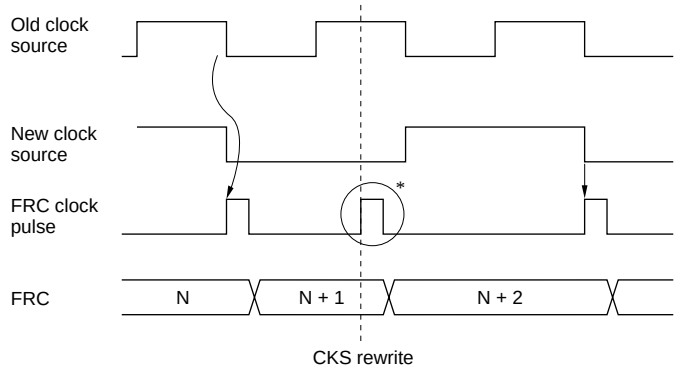
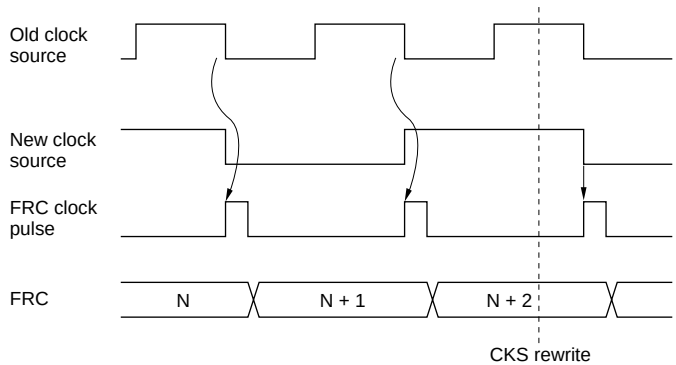
No.	Description	Timing
1	Low → low: CKS1 and CKS0 are rewritten while both clock sources are low.	
2	Low → high: CKS1 and CKS0 are rewritten while old clock source is low and new clock source is high.	

Table 9.5 Effect of Changing Internal Clock Sources (cont)

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

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

Section 10 16-Bit Free-Running Timer 1

10.1 Overview

The H8/3318 has an on-chip 16-bit free-running timer (FRT) with two channels: FRT0 and FRT1. Each channel is based on a 16-bit free-running counter (FRC) and can generate two independent output waveforms, measure input pulse widths, or measure external clock periods.

This section describes FRT1. For FRT0, see section 9, 16-Bit Free-Running Timer 0.

The differences between FRT0 and FRT1 are that FRT0 has four input-capture lines and four input-capture interrupt sources, and allows buffering to be designated. FRT1 has only one input-capture line and one input-capture interrupt source, and does not support buffering.

10.1.1 Features

The features of FRT1 are listed below.

- Selection of four clock sources
  The free-running counter can be driven by an internal clock source ($ø_p/2$, $ø_p/8$, or $ø_p/32$), or an external clock input (enabling use as an external event counter).
- Two independent comparators
  Each comparator can generate an independent waveform.
- Input capture
  The current count can be captured on the rising or falling edge (selectable) of an input signal.
- Counter can be cleared under program control
  The free-running counter can be cleared on compare-match A.
- Four interrupt sources
  Compare-match A and B, input capture, and overflow interrupts are requested independently.

10.1.2 Block Diagram

Figure 10.1 shows a block diagram of free-running timer 1.

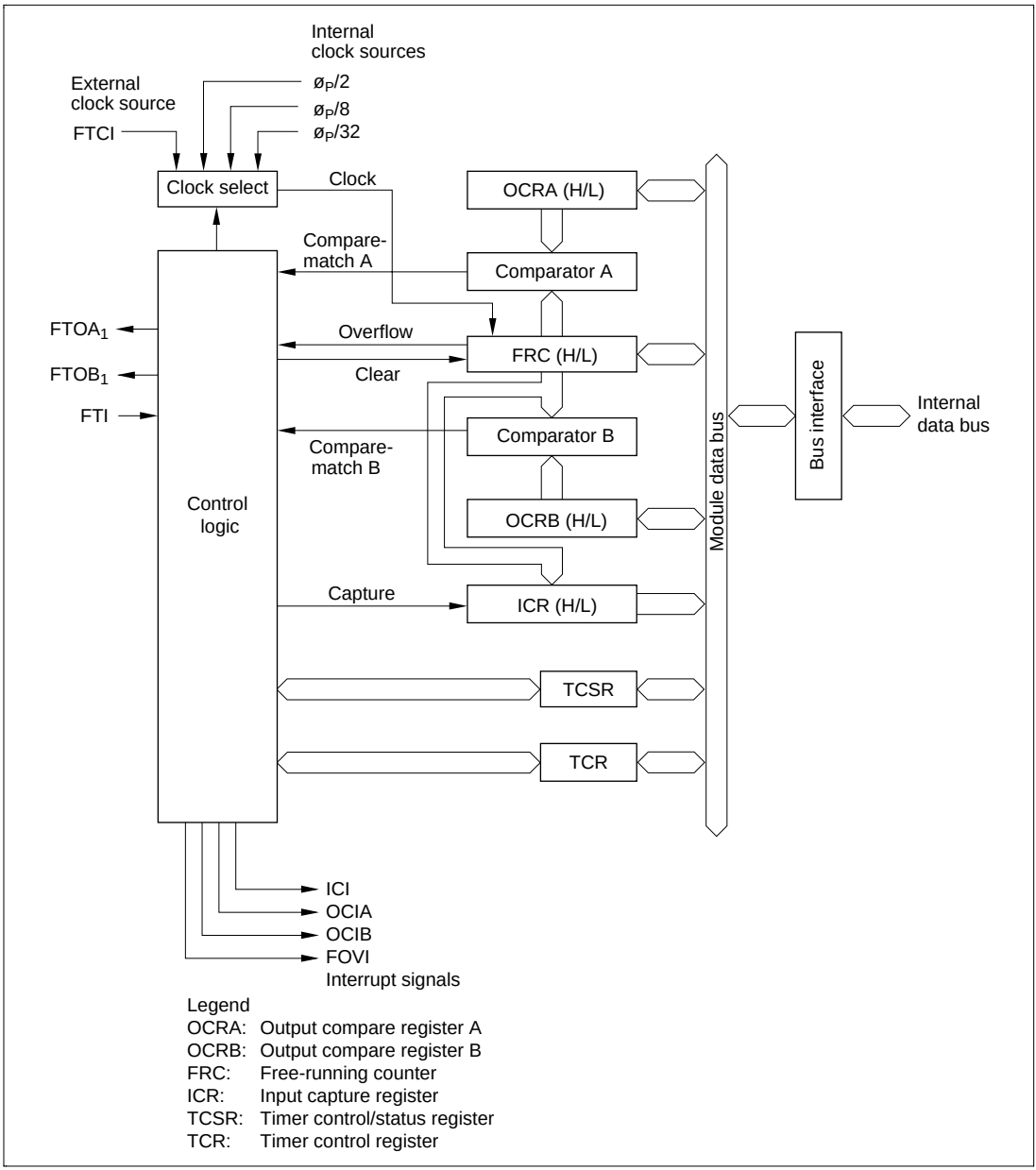


Figure 10.1 Block Diagram of 16-Bit Free-Running Timer 1

10.1.3 Input and Output Pins

Table 10.1 lists the input and output pins of free-running timer 1.

Table 10.1 FRT1 Input and Output Pins

Name	Abbreviation	I/O	Function
Counter clock input	FTCI	Input	Input of external free-running counter clock signal
Output compare A	$FTOA_1$*	Output	Output controlled by comparator A
Output compare B	$FTOB_1$*	Output	Output controlled by comparator B
Input capture	FTI	Input	Input capture trigger

Note: * In this manual, the channel subscripts are normally omitted.

10.1.4 Register Configuration

Table 10.2 lists the registers of free-running timer 1.

Table 10.2 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address
Timer control register	TCR	R/W	H'00	H'FFA0
Timer control/status register	TCSR	R/(W)*	H'00	H'FFA1
Free-running counter (high)	FRC (H)	R/W	H'00	H'FFA2
Free-running counter (low)	FRC (L)	R/W	H'00	H'FFA3
Output compare register A (high)	OCRA (H)	R/W	H'FF	H'FFA4
Output compare register A (low)	OCRA (L)	R/W	H'FF	H'FFA5
Output compare register B (high)	OCRB (H)	R/W	H'FF	H'FFA6
Output compare register B (low)	OCRB (L)	R/W	H'FF	H'FFA7
Input capture register (high)	ICR (H)	R	H'00	H'FFA8
Input capture register (low)	ICR (L)	R	H'00	H'FFA9

Note: * Software can write a 0 to clear bits 7 to 4, but cannot write a 1 in these bits.

10.2 Register Descriptions

10.2.1 Free-Running Counter (FRC)

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

FRC is a 16-bit readable/writable up-counter that increments on an internal pulse generated from a clock source. The clock source is selected by the clock select 1 and 0 bits (CKS1 and CKS0) of the timer control register (TCR).

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

Because FRC is a 16-bit register, a temporary register (TEMP) is used when the FRC is written or read. See section 10.3, CPU Interface, for details.

FRC is initialized to H'0000 by a reset and in the standby modes. It can also be cleared by compare-match A.

10.2.2 Output Compare Registers A and B (OCRA and OCRB)

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

OCRA and OCRB are 16-bit readable/writable registers, the contents of which are continually compared with the value in FRC. When a match is detected, the corresponding output compare flag (OCFA or OCFB) is set in the timer control/status register (TCSR).

In addition, if the output enable bit (OEA or OEB) in the timer control register (TCR) is set to 1, when the output compare register and FRC values match, the logic level selected by the output level bit (OLVLA or OLVLB) in TCSR is output at the output compare pin ($FTOA_1$ or $FTOB_1$). Following a reset, the $FTOA_1$ and $FTOB_1$ output levels are 0 until the first compare-match.

Because OCRA and OCRB are 16-bit registers, a temporary register (TEMP) is used for write access, as explained in section 10.3, CPU Interface.

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

10.2.3 Input Capture Register (ICR)

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

ICR is a 16-bit read-only register.

When the rising or falling edge of the signal at the input capture pin (FTI) is detected, the current FRC value is copied to ICR. At the same time, the input capture flag (ICF) in the timer control/status register (TCSR) is set to 1. The input capture edge is selected by the input edge select bit (IEDG) in TCSR.

Because ICR is a 16-bit register, a temporary register (TEMP) is used when it is read. See Section 10.3, CPU Interface, for details.

To ensure input capture, the width of the input capture pulse should be at least 1.5 system clock cycles (1.5·ø).

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

Note: When input capture is detected, the FRC value is transferred to ICR even if the input capture flag is already set.

10.2.4 Timer Control Register (TCR)

Bit	7	6	5	4	3	2	1	0
	ICIE	OCIBE	OCICE	OVIE	OEB	OEA	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 enables and disables output signals and interrupts, and selects the timer clock source.

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

Bit 7—Input Capture Interrupt Enable (ICIE): Selects whether to request an input capture interrupt (ICI) when the input capture flag (ICF) in the timer status/control register (TCSR) is set to 1.

Bit 7 ICIE	Description	
0	Input capture interrupt request (ICI) is disabled	(Initial value)
1	Input capture interrupt request (ICI) is enabled	

Bit 6—Output Compare Interrupt Enable B (OCIEB): Selects whether to request output compare interrupt B (OCIB) when output compare flag B (OCFB) in TCSR is set to 1.

Bit 6 OCIEB	Description	
0	Output compare interrupt request B (OCIB) is disabled	(Initial value)
1	Output compare interrupt request B (OCIB) is enabled	

Bit 5—Output Compare Interrupt Enable A (OCIEA): Selects whether to request output compare interrupt A (OCIA) when output compare flag A (OCFA) in TCSR is set to 1.

Bit 5 OCIEA	Description	
0	Output compare interrupt request A (OCIA) is disabled	(Initial value)
1	Output compare interrupt request A (OCIA) is enabled	

Bit 4—Timer Overflow Interrupt Enable (OVIE): Selects whether to request an overflow interrupt (FOVI) when the timer overflow flag (OVF) in TCSR is set to 1.

Bit 4 OVIE	Description	
0	Timer overflow interrupt request (FOVI) is disabled	(Initial value)
1	Timer overflow interrupt request (FOVI) is enabled	

Bit 3—Output Enable B (OEB): Enables or disables output of the output compare B signal (FTOB). If output compare B is enabled, the $FTOB_1$ pin is driven to the level selected by OLVLB in TCSR whenever the FRC value matches the value in output compare register B (OCRB).

Bit 3 OEB	Description	
0	Output compare B output is disabled	(Initial value)
1	Output compare B output is enabled	

Bit 2—Output Enable A (OEA): Enables or disables output of the output compare A signal (FTOA). If output compare A is enabled, the FTOA pin is driven to the level selected by OLVLA in the timer status/control register (TCSR) whenever the FRC value matches the value in output compare register A (OCRA).

Bit 2 OEA	Description	
0	Output compare A output is disabled	(Initial value)
1	Output compare A output is enabled	

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

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

10.2.5 Timer Control/Status Register (TCSR)

Bit	7	6	5	4	3	2	1	0
	ICA	OCFB	OCFA	OVF	OLVLB	OLVLA	IEDG	CCLRA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)	R/(W)	R/(W)	R/(W)

Note: * Software can write a 0 in bits 7 to 4 to clear the flags, but cannot write a 1 in these bits.

TCSR is an 8-bit readable and partially writable register that contains the four interrupt flags and selects the output compare levels, input capture edge, and whether to clear the counter on compare-match A.

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

Bit 7—Input Capture Flag (ICF): This status bit is set to 1 to flag an input capture event, indicating that the FRC value has been copied to ICR.

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

Bit 7 ICF	Description
0	To clear ICF, the CPU must read ICF after it has been set to 1, then write a 0 in this bit (Initial value)
1	This bit is set to 1 when an input capture signal causes the FRC value to be copied to the ICR

Bit 6—Output Compare Flag B (OCFB): This status flag is set to 1 when the FRC value matches the OCRB value.

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

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

Bit 5—Output Compare Flag A (OCFA): This status flag is set to 1 when the FRC value matches the OCRA value.

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

Bit 5 OCFA	Description
0	To clear OCFA, the CPU must read OCFA after it has been set to 1, then write a 0 in this bit (Initial value)
1	This bit is set to 1 when FRC = OCRA

Bit 4—Timer Overflow Flag (OVF): This status flag is set to 1 when the FRC overflows (changes from H'FFFF to H'0000).

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

Bit 4 OVF	Description	
0	To clear OVF, the CPU must read OVF after it has been set to 1, then write a 0 in this bit	(Initial value)
1	This bit is set to 1 when FRC changes from H'FFFF to H'0000	

Bit 3—Output Level B (OLVLB): Selects the logic level output at the FTOB pin when the FRC and OCRB values match.

Bit 3 OLVLB	Description	
0	A 0 logic level is output for compare-match B	(Initial value)
1	A 1 logic level is output for compare-match B	

Bit 2—Output Level A (OLVLA): Selects the logic level output at the FTOA pin when the FRC and OCRA values match.

Bit 2 OLVLA	Description	
0	A 0 logic level is output for compare-match A	(Initial value)
1	A 1 logic level is output for compare-match A	

Bit 1—Input Edge Select (IEDG): Selects the rising or falling edge of the input capture signal (FTI).

Bit 1 IEDG	Description	
0	FRC contents are transferred to ICR on the falling edge of FTI	(Initial value)
1	FRC contents are transferred to ICR on the rising edge of FTI	

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

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

10.3 CPU Interface

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

These registers are written and read as follows:

- Register Write

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

- Register Read

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

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

Coding Examples

To write the contents of general register R0 to OCRA: `MOV.W R0, @OCRA`
To transfer the ICR contents to general register R0: `MOV.W @ICR, R0`

Figure 10.2 shows the data flow when FRC is accessed. The other registers are accessed in the same way. As an exception, when the CPU reads OCRA or OCRB, it reads both the upper and lower bytes directly, without using TEMP.

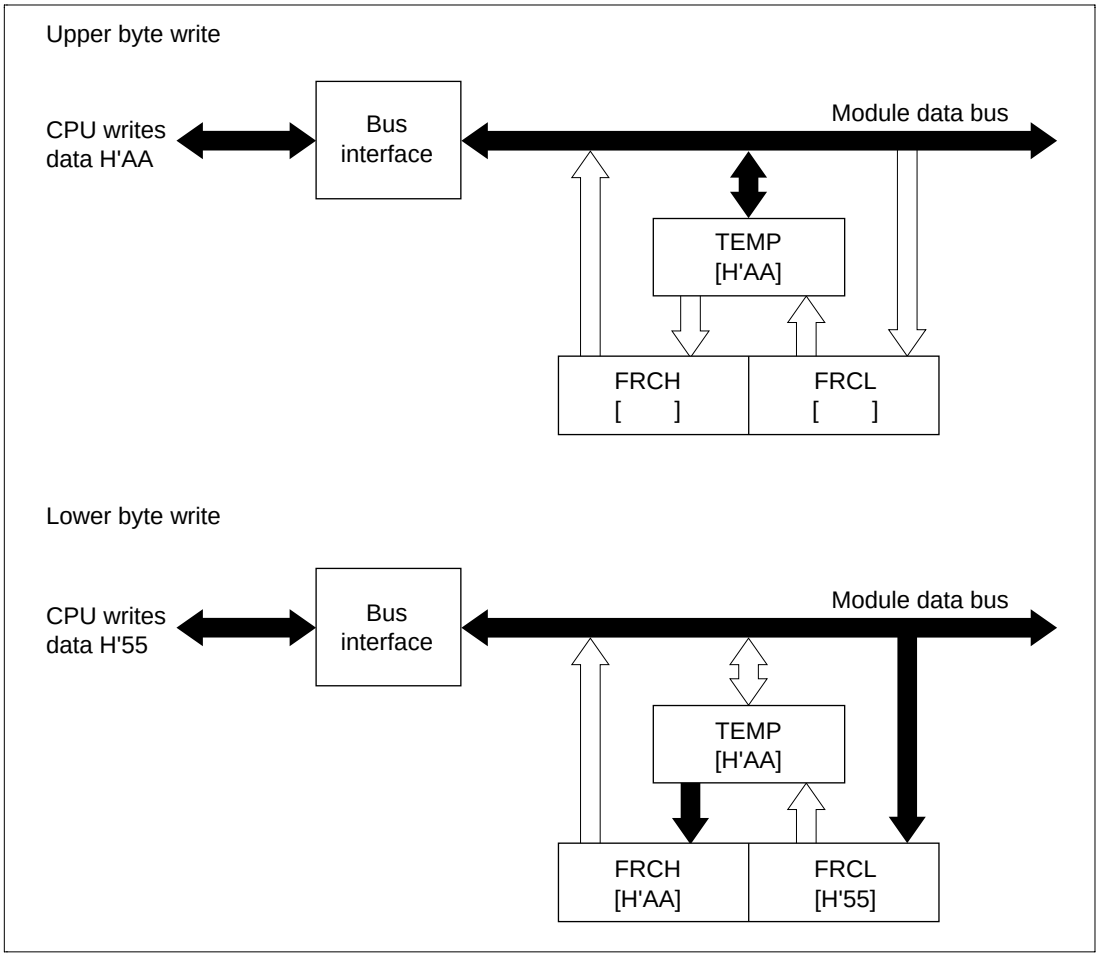


Figure 10.2a Write Access to FRC (When CPU Writes H'AA55)

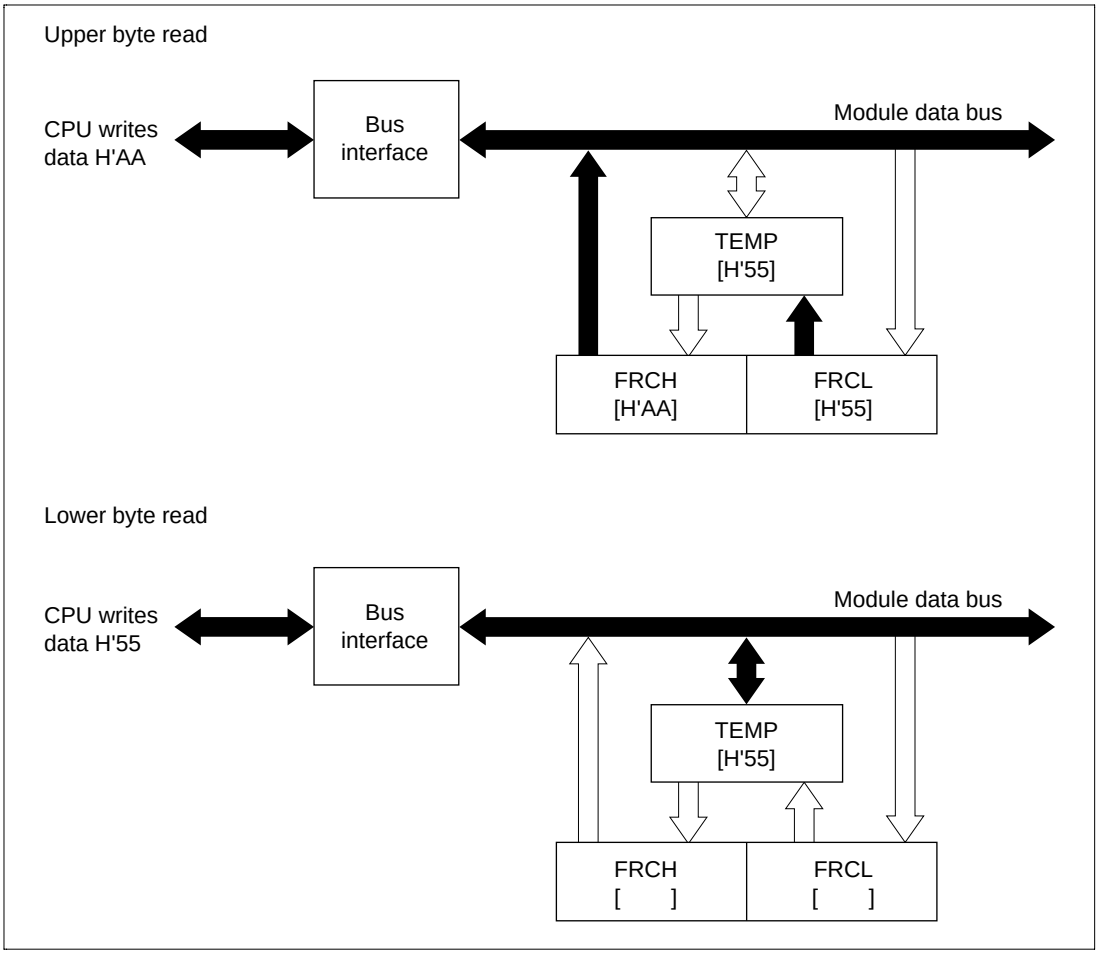


Figure 10.2b Read Access to FRC (When FRC Contains H'AA55)

10.4 Operation

10.4.1 FRC Incrementation Timing

FRC increments on a pulse generated once for each cycle of the selected (internal or external) clock source.

Internal Clock Sources: Can be selected by the CKS1 and CKS0 bits in the TCR. Internal clock sources are created by dividing the system clock (ø). Three internal clock sources are available: $ø_P/2$, $ø_P/8$, and $ø_P/32$. Figure 10.3 shows the increment timing.

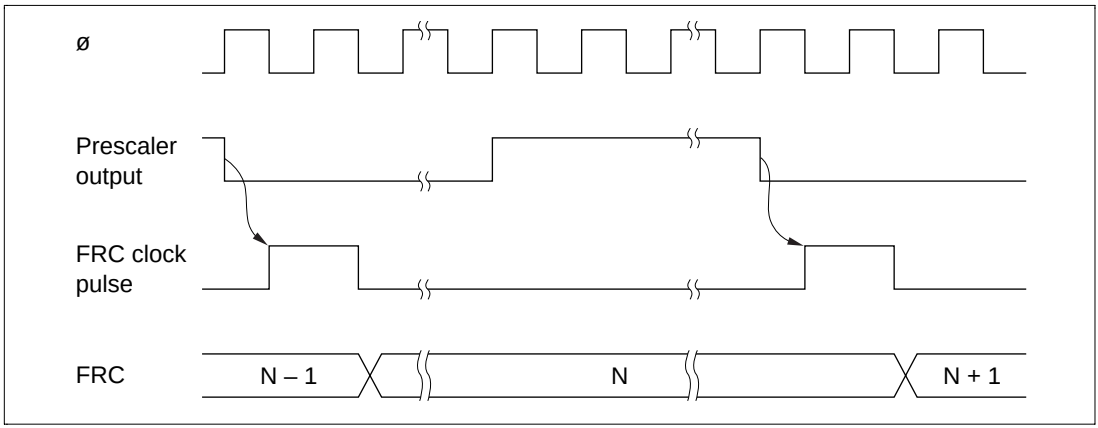


Figure 10.3 Increment Timing for Internal Clock Source

External Clock Input: Can be selected by the CKS1 and CKS0 bits in the TCR. The FRC increments on the rising edge of the FTCI clock signal. The pulse width of the external clock signal must be at least 1.5 system clock (ø) cycles. The counter will not increment correctly if the pulse width is shorter than this.

Figure 10.4 shows the increment timing.

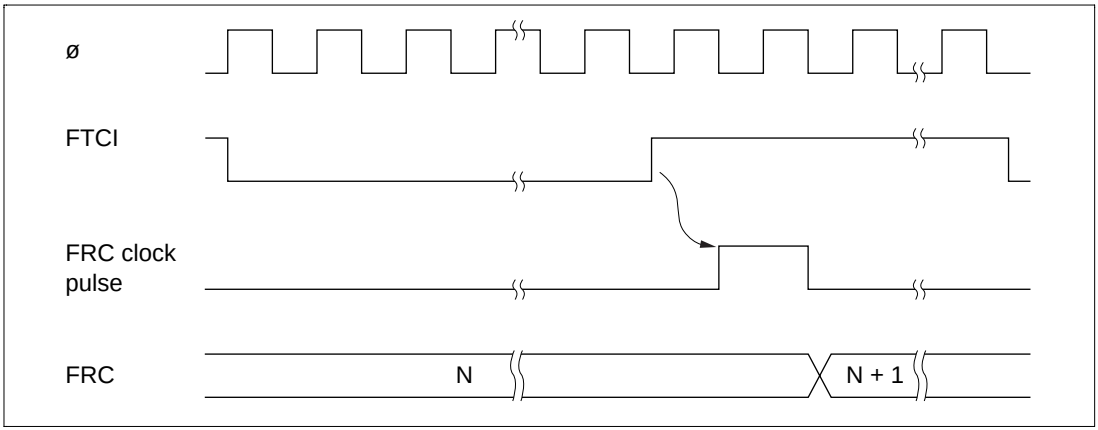


Figure 10.4 Increment Timing for External Clock Source

10.4.2 Output Compare Timing

When a compare-match occurs, the logic level selected by the output level bit (OLVLA or OLVLB) in the TCSR is output at the output compare pin (FTOA or FTOB). Figure 10.5 shows the timing of this operation for compare-match A.

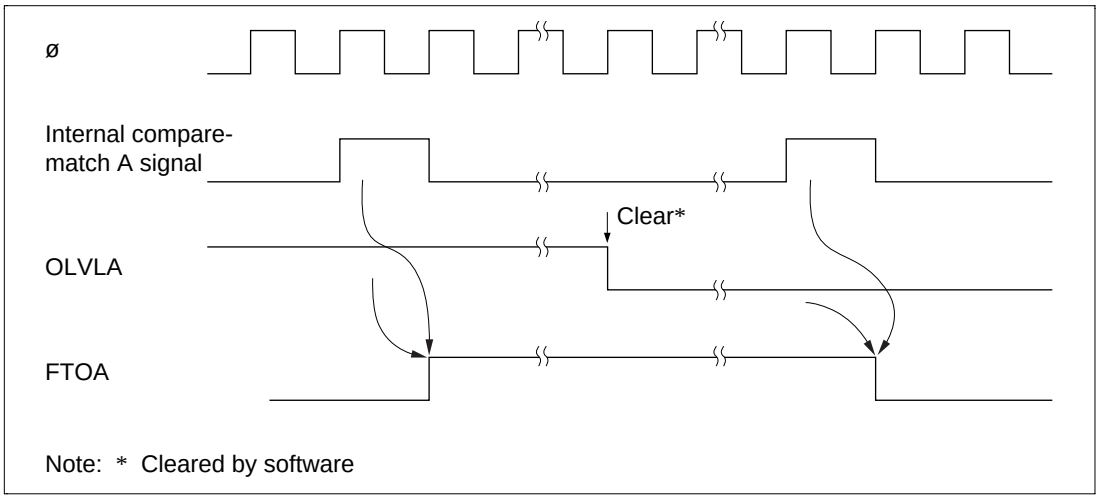


Figure 10.5 Timing of Output Compare A

10.4.3 FRC Clear Timing

If the CCLRA bit in the TCSR is set to 1, the FRC is cleared when compare-match A occurs. Figure 10.6 shows the timing of this operation.

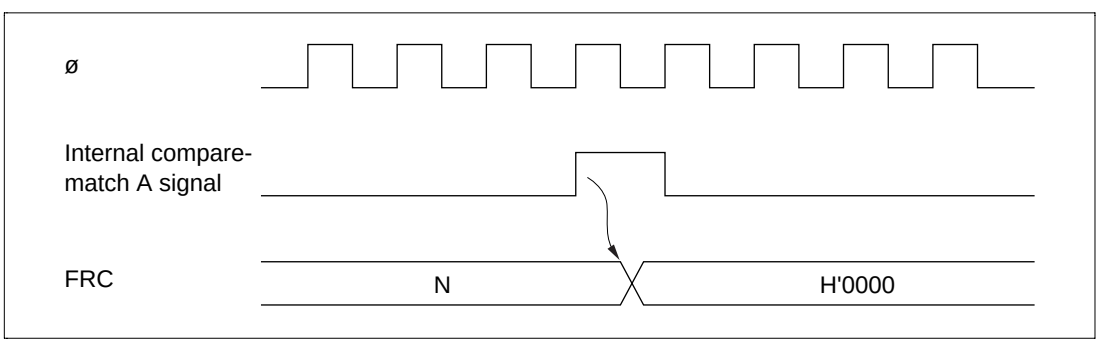


Figure 10.6 Clearing of FRC by Compare-Match A

10.4.4 Input Capture Timing

An internal input capture signal is generated from the rising or falling edge of the FTI input, as selected by the IEDG bit in TCSR. Figure 10.7 shows the usual input capture timing when the rising edge is selected (IEDG = 1).

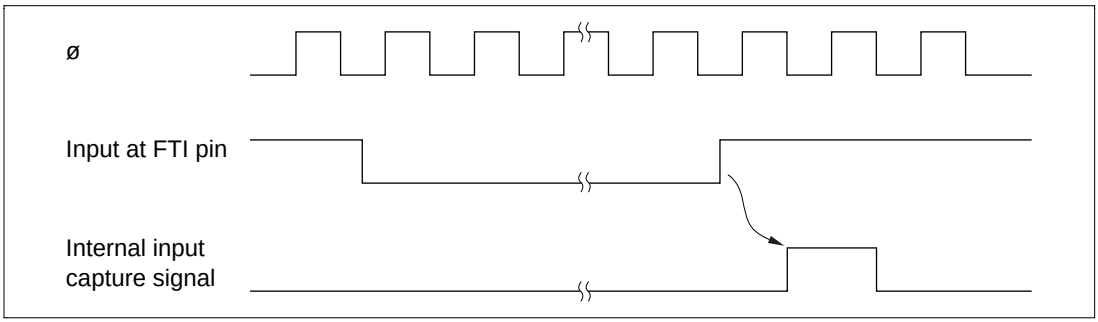


Figure 10.7 Input Capture Timing (Usual Case)

If the upper byte of ICR is being read when the internal input capture signal should be generated, the internal input capture signal is delayed by one state. Figure 10.8 shows the timing for this case.

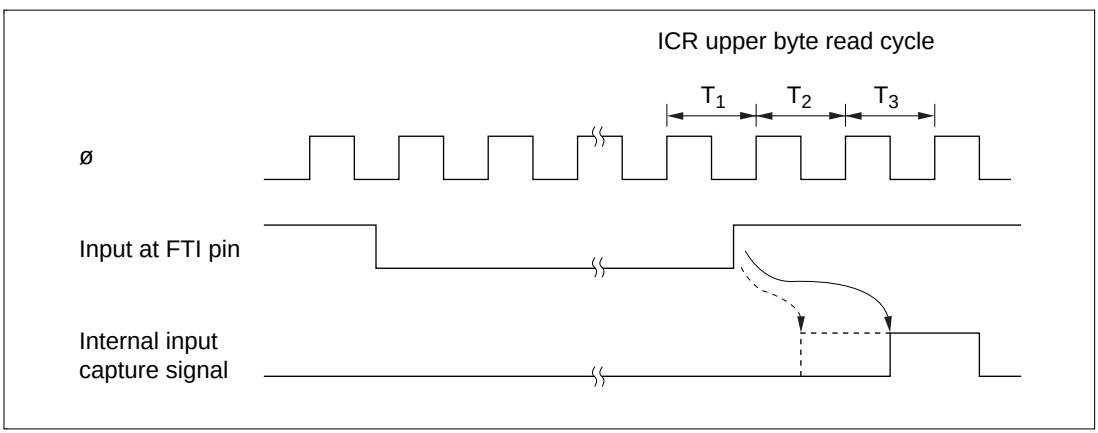


Figure 10.8 Input Capture Timing (1-State Delay Due to ICR Read)

10.4.5 Timing of Input Capture Flag (ICF) Setting

The input capture flag ICF is set to 1 by the internal input capture signal. The FRC contents are transferred to ICR at the same time. Figure 10.9 shows the timing of this operation.

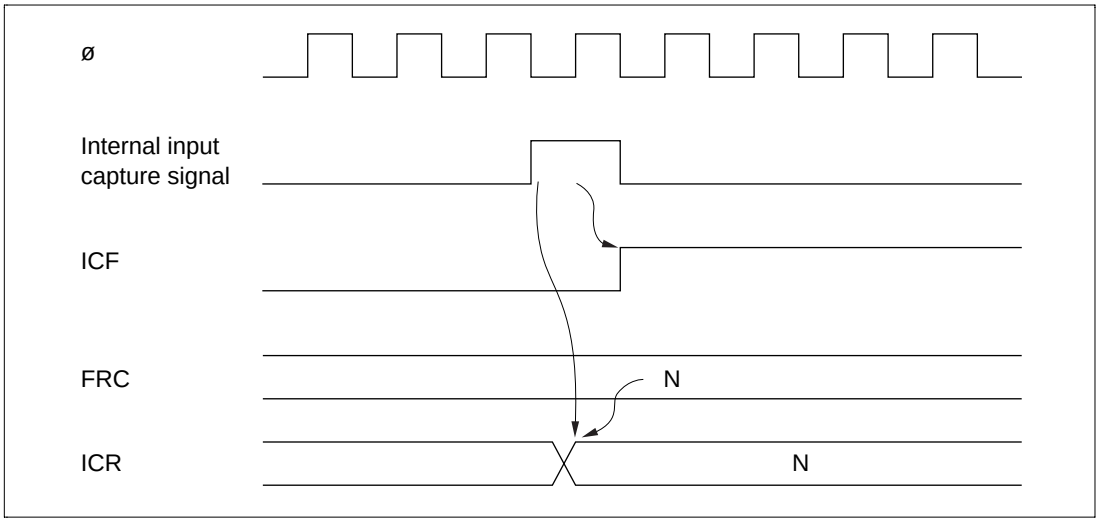


Figure 10.9 Setting of Input Capture Flag

10.4.6 Setting of FRC Overflow Flag (OVF)

The FRC overflow flag (OVF) is set to 1 when FRC changes from H'FFFF to H'0000. Figure 10.10 shows the timing of this operation.

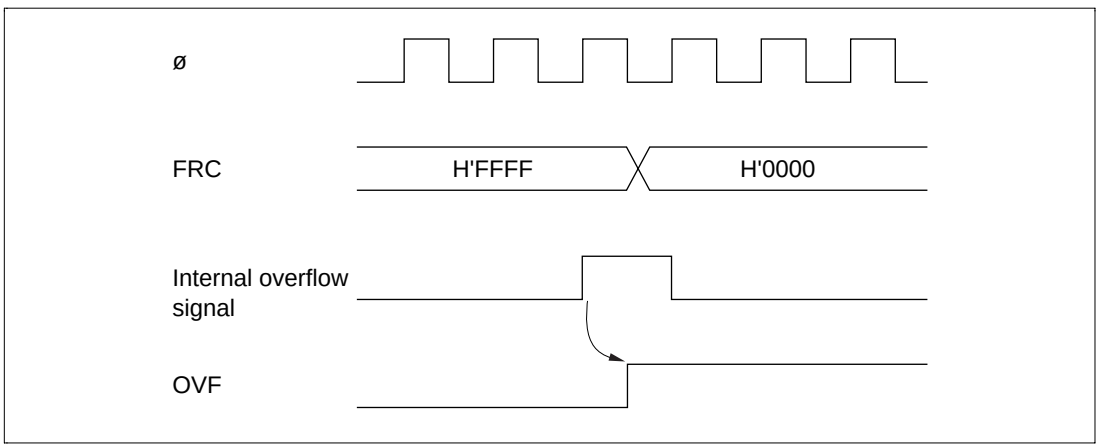


Figure 10.10 Setting of Overflow Flag (OVF)

10.5 Interrupts

FRT1 can request four interrupts: input capture (ICI), output compare A and B (OCIA and OCIB), and overflow (FOVI). Each interrupt can be enabled or disabled by an enable bit in TCR. Independent signals are sent to the interrupt controller for each interrupt. Table 10.3 lists information about these interrupts.

Table 10.3 FRT1 Interrupts

Interrupt	Description	Priority
ICI	Requested by ICF	High
OCIA	Requested by OCFA	
OCIB	Requested by OCFB	
FOVI	Requested by OVF	Low

10.6 Sample Application

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

1. The CCLRA bit in TCSR is set to 1.
2. Each time a compare-match interrupt occurs, software inverts the corresponding output level bit in TCSR (OLVLA or OLVLB).

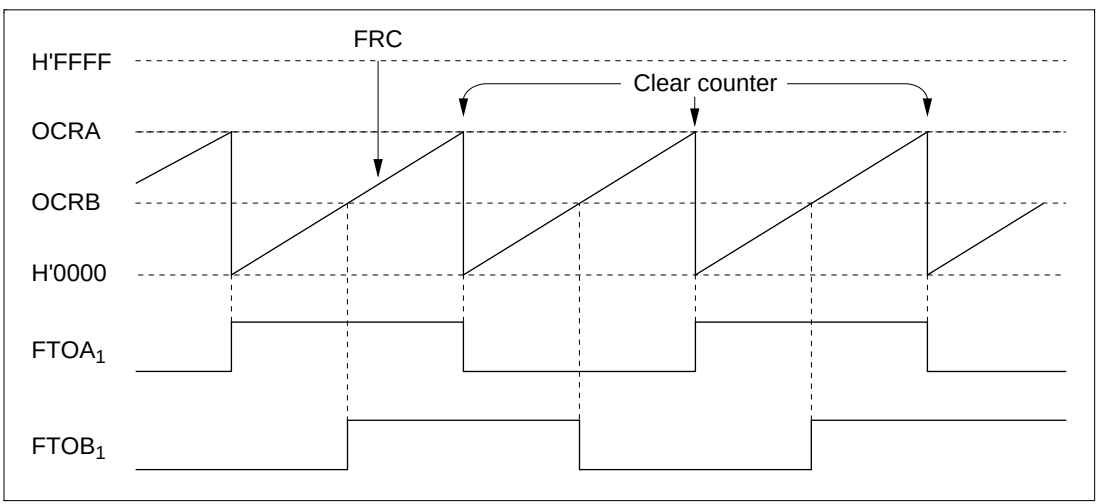


Figure 10.11 Square-Wave Output (Example)

10.7 Application Notes

Application programmers should note that the following types of contention can occur in free-running timer 1.

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

Figure 10.12 shows this type of contention.

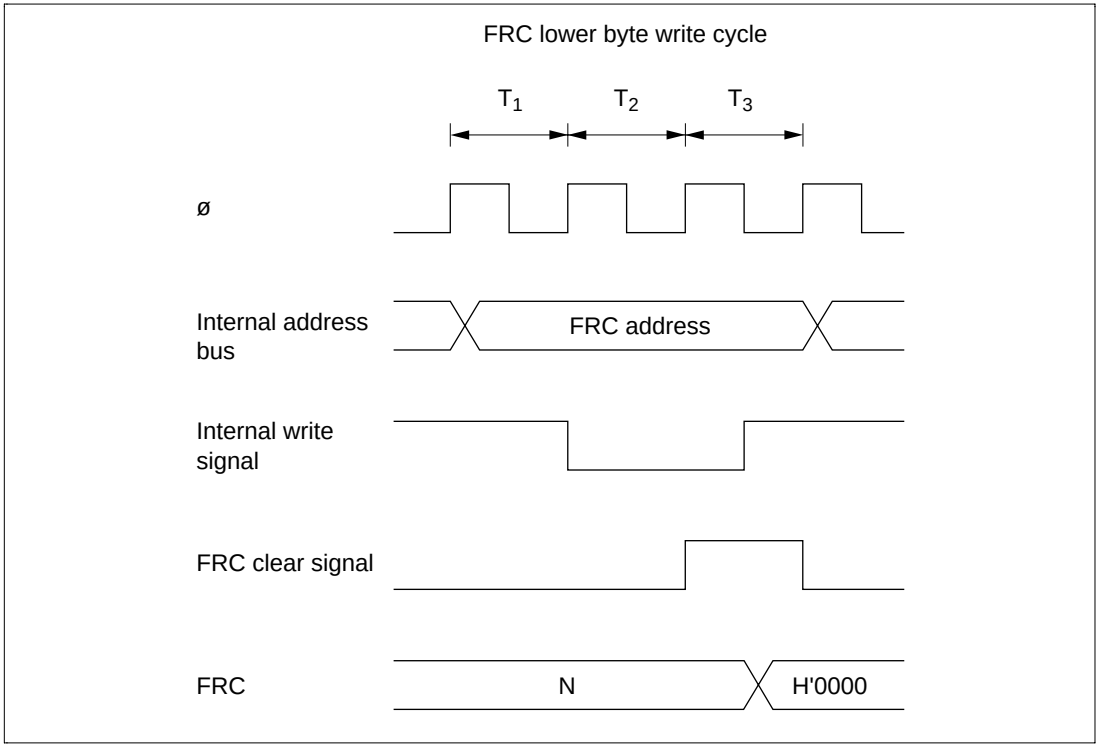


Figure 10.12 FRC Write-Clear Contention

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

Figure 10.13 shows this type of contention.

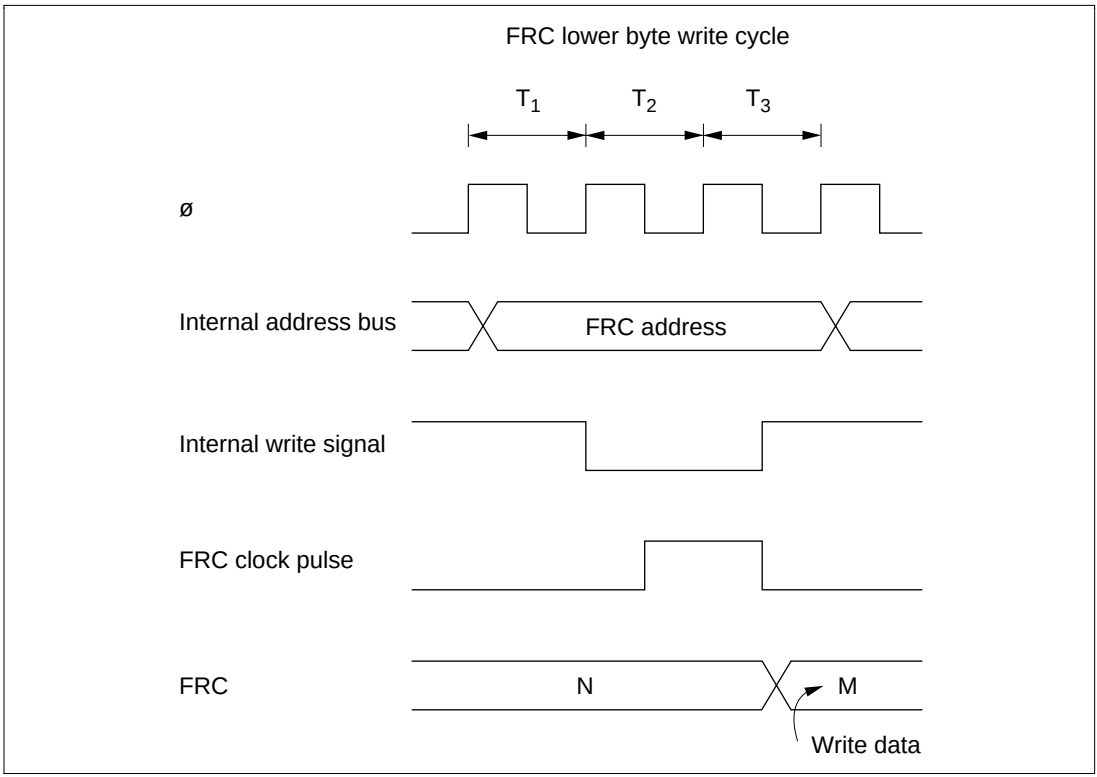


Figure 10.13 FRC Write-Increment Contention

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

Figure 10.14 shows this type of contention.

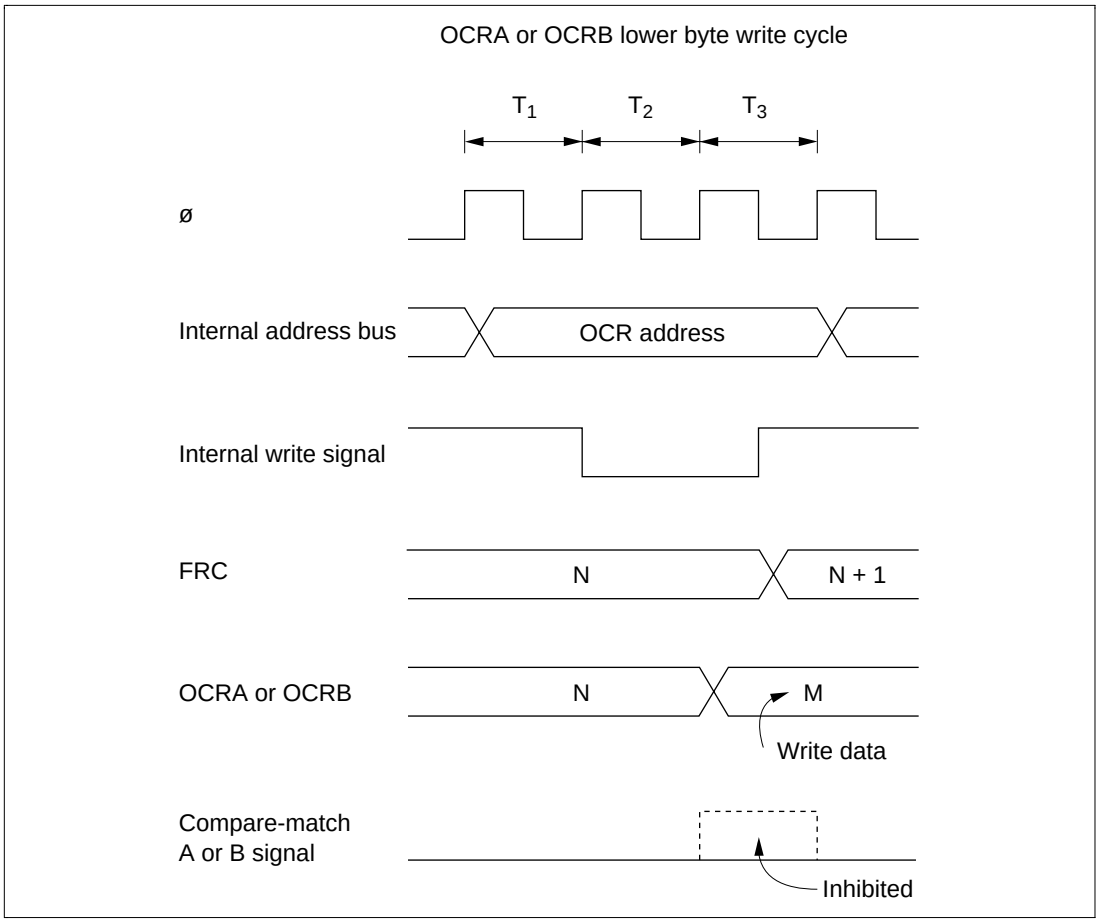


Figure 10.14 Contention between OCR Write and Compare-Match

Incrementation Caused by Changing of Internal Clock Source: When an internal clock source is changed, the changeover may cause FRC to increment. This depends on the time at which the clock select bits (CKS1 and CKS0) are rewritten, as shown in table 10.4.

The pulse that increments FRC is generated at the falling edge of the internal clock source. If clock sources are changed when the old source is high and the new source is low, as in case number 3 in table 10.4, the changeover generates a falling edge that triggers the FRC increment clock pulse.

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

Table 10.4 Effect of Changing Internal Clock Sources

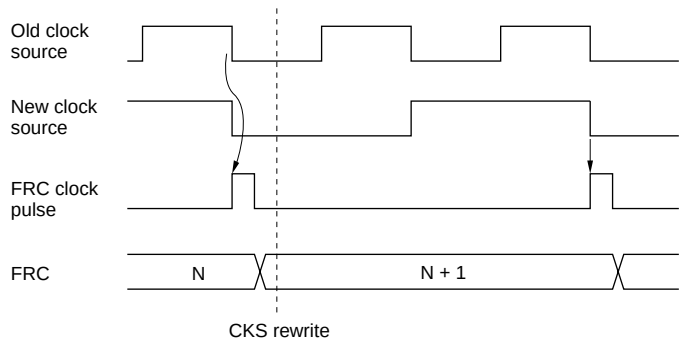
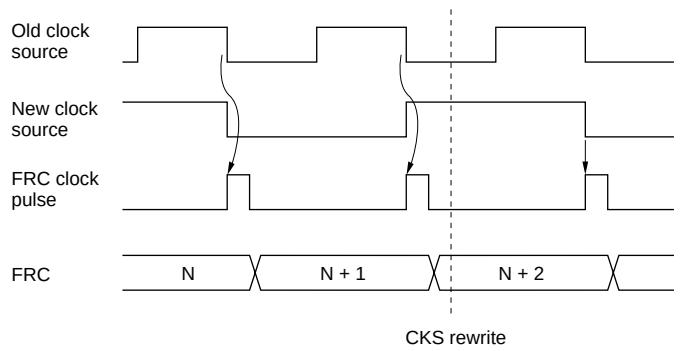
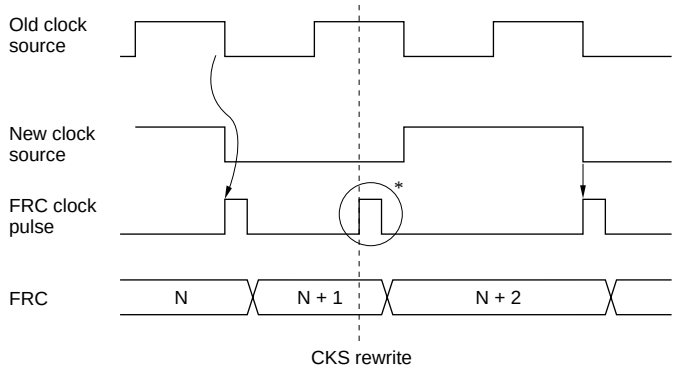
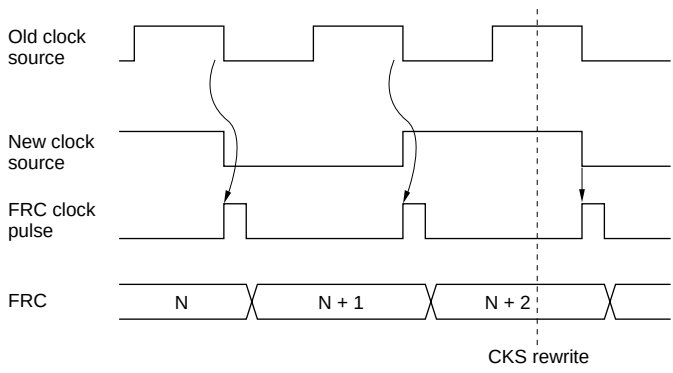
No.	Description	Timing
1	Low → low: CKS1 and CKS0 are rewritten while both clock sources are low.	
2	Low → high: CKS1 and CKS0 are rewritten while old clock source is low and new clock source is high.	

Table 10.4 Effect of Changing Internal Clock Sources (cont)

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

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

Section 11 8-Bit Timers

11.1 Overview

The H8/3318 includes an 8-bit timer module with two channels (numbered 0 and 1). Each channel has an 8-bit counter (TCNT) and two time constant registers (TCORA and TCORB) that are constantly compared with the TCNT value to detect compare-match events. One of the many applications of the 8-bit timer module is to generate a rectangular-wave output with an arbitrary duty cycle.

11.1.1 Features

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

- Selection of seven clock sources
  The counters can be driven by one of six internal clock signals or an external clock input (enabling use as an external event counter).
- Selection of three ways to clear the counters
  The counters can be cleared on compare-match A or B, or by an external reset signal.
- Timer output controlled by two time constants
  The timer output signal in each channel is controlled by two independent time constants, enabling the timer to generate output waveforms with an arbitrary duty cycle. PWM waveforms with a duty cycle of 0 to 100% can easily be output by selecting PWM mode.
- Three independent interrupts
  Compare-match A and B and overflow interrupts can be requested independently.

11.1.2 Block Diagram

Figure 11.1 shows a block diagram of one channel in the 8-bit timer module.

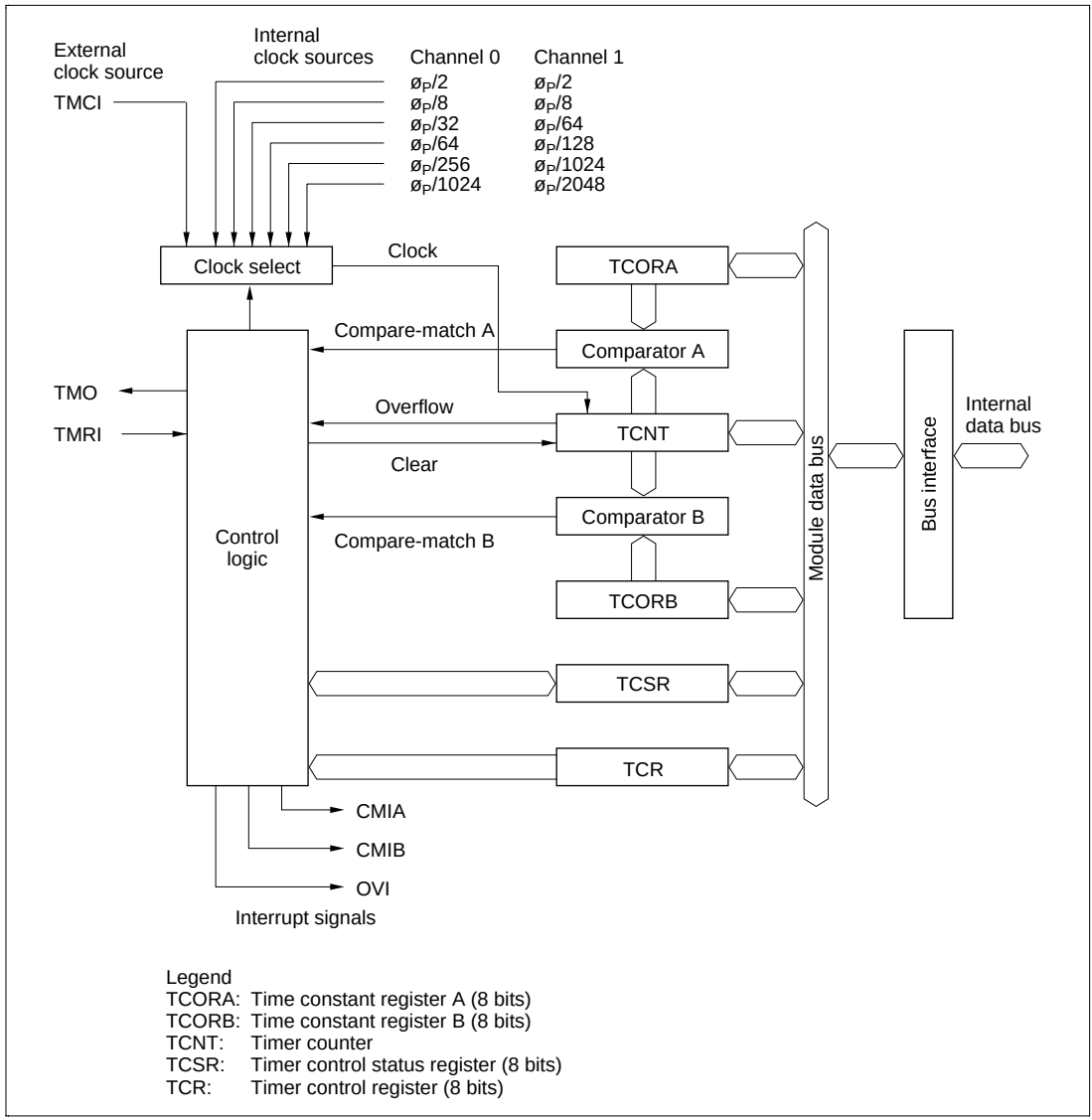


Figure 11.1 Block Diagram of 8-Bit Timer (1 Channel)

11.1.3 Input and Output Pins

Table 11.1 lists the input and output pins of the 8-bit timer.

Table 11.1 Input and Output Pins of 8-Bit Timer

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

Note: * In this manual, the channel subscripts are normally omitted.

11.1.4 Register Configuration

Table 11.2 lists the registers of the 8-bit timer module.

Table 11.2 8-Bit Timer Registers

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

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

11.2 Register Descriptions

11.2.1 Timer Counter (TCNT)

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

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

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

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

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

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

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

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

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

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

11.2.3 Timer Control Register (TCR)

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCR is an 8-bit readable/writable register that selects the clock source and the time at which the timer counter is cleared, and enables interrupts.

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

For timing diagrams, see section 11.3, Operation.

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

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

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

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

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

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

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

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

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

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

11.2.4 Timer Control/Status Register (TCSR)

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	PWME	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: * Software can write a 0 in bits 7 to 5 to clear the flags, but cannot write a 1 in these bits.

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

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

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

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

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

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

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

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

Bit 4—PWM Mode Enable (PWME): This bit selects PWM mode for the timer output.

Bit 4 PWME	Description
0	Normal timer mode (Initial value)
1	PWM mode

In PWM mode, CCLR1 to CCLR0 and OS3 to OS0 must be set so that the timer output cycle is determined by the contents of TCORA, and the timer output duty cycle by the contents of TCORB. In this case, the timer output pulse cycle, pulse width, and duty cycle conform to the following expressions. If TCORA < TCORB, the output is saturated at a 100% duty cycle.

When TCORB ≤ TCORA:

Timer output pulse cycle = selected internal clock cycle × (TCORA + 1)
Timer output pulse width = selected internal clock cycle × TCORB
Timer output duty cycle = TCORB / (TCORA + 1)

	TCR		TCSR			
PWM Output Mode	**CCLR1**	**CCLR0**	**OS3**	**OS2**	**OS1**	**OS0**
Direct output (when the above timer pulse width is high)	0	1	0	1	1	0
Inverted output (when the above timer pulse width is low)	0	1	1	0	0	1

In PWM mode, a buffer register is inserted between TCORB and the module data bus, and the data written to TCORB is held in the buffer register until a TCORA compare-match occurs. This makes it easy to obtain PWM output with no disruption of the waveform. As regards the timer output specification made by OS3 to OS0, the priority of a change due to compare-match B is higher. Care is required since the operation is different from that in the normal timer mode.

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

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

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

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

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

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

11.2.5 Serial/Timer Control Register (STCR)

Bit	7	6	5	4	3	2	1	0
	RING	CMPF	CMPIE	LOAD	MARK	—	ICKS1	ICKS0
Initial value	0	0	0	1	1	1	0	0
Read/Write	R/W	R/(W)*	R/W	(W)	(W)	—	R/W	R/W

Note: * Software can write a 0 in bit 6 to clear the flags, but cannot write a 1 in this bit.

STCR is an 8-bit readable/writable register that controls the serial communication interface, selects clock sources for the timer counters, and controls DTU channel B.

STCR is initialized to H'1C by a reset.

Bits 7 to 3—DTU Channel B Control: These bits control DTU channel B. For details, see section 5, Data Transfer Unit.

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

Bits 1 and 0—Internal Clock Source Select 1 and 0 (ICKS1 and ICKS0): These bits and bits CKS2 to CKS0 in the TCR select clock sources for the timer counters. For details, see section 11.2.3, Timer Control Register.

11.3 Operation

11.3.1 TCNT Incrementation Timing

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

Internal Clock: Internal clock sources are created from the system clock by a prescaler. The counter increments on an internal TCNT clock pulse generated from the falling edge of the prescaler output, as shown in figure 11.2. Bits CKS2 to CKS0 of TCR and bits ICKS1 and ICKS0 of STCR can select one of the six internal clocks.

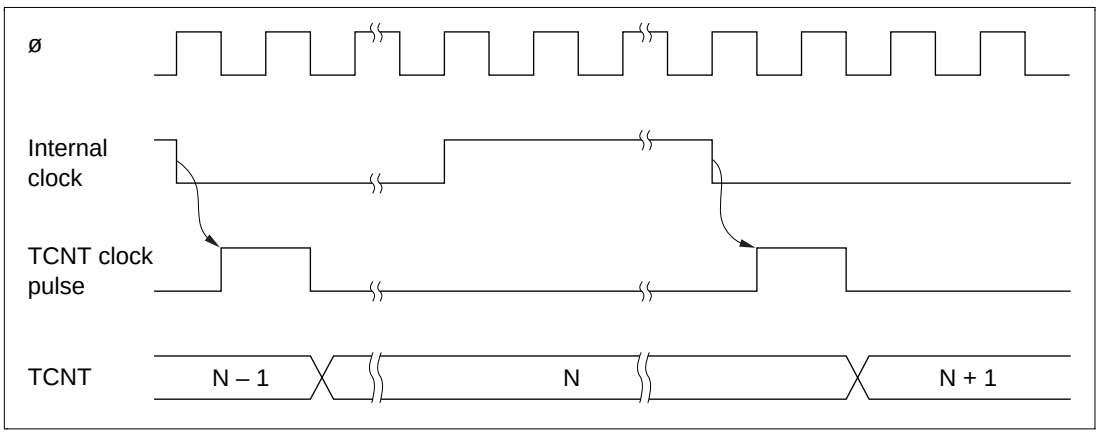


Figure 11.2 Count Timing for Internal Clock Input

External Clock: If external clock input (TMCI) is selected, the timer counter can increment on the rising edge, the falling edge, or both edges of the external clock signal. Figure 11.3 shows incrementation on both edges of the external clock signal.

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

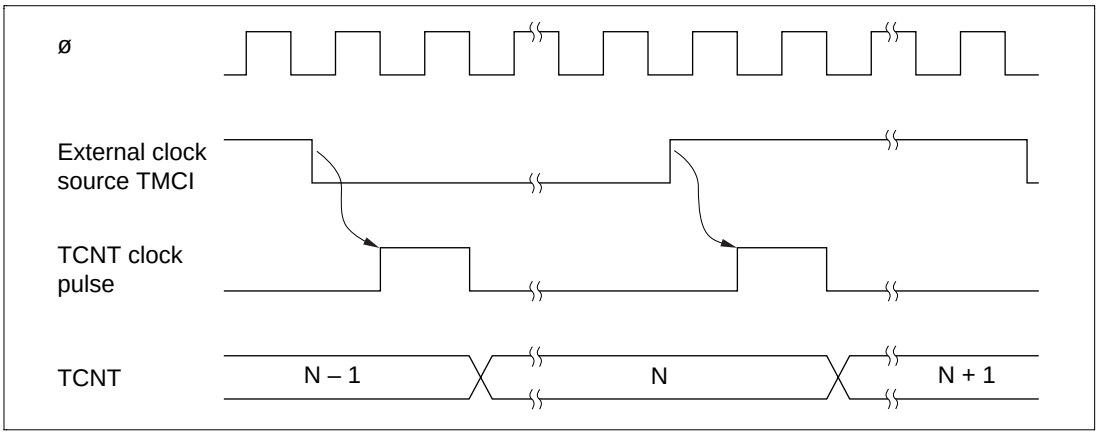


Figure 11.3 Count Timing for External Clock Input

11.3.2 Compare-Match Timing

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

Accordingly, when the timer count matches one of the time constants, the compare-match signal is not generated until the next period of the clock source. Figure 11.4 shows the timing of the setting of the compare-match flags.

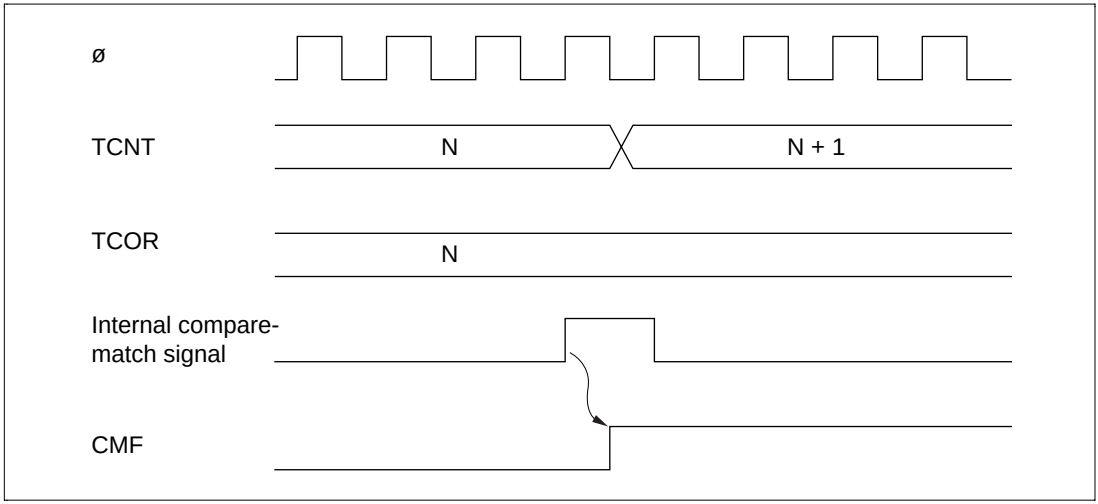


Figure 11.4 Setting of Compare-Match Flags

Output Timing: When a compare-match event 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, change to 0, change to 1, or toggle.

Figure 11.5 shows the timing when the output is set to toggle on compare-match A.

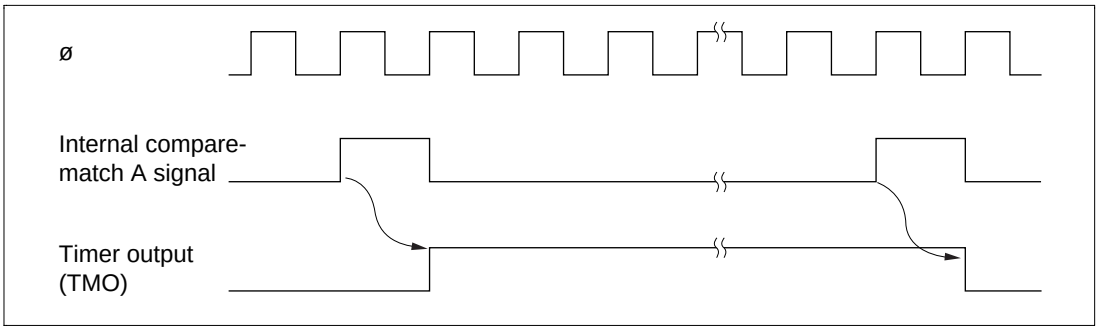


Figure 11.5 Timing of Timer Output

Timing of Compare-Match Clear: Depending on the CCLR1 and CCLR0 bits in TCR, the timer counter can be cleared when compare-match A or B occurs. Figure 11.6 shows the timing of this operation.

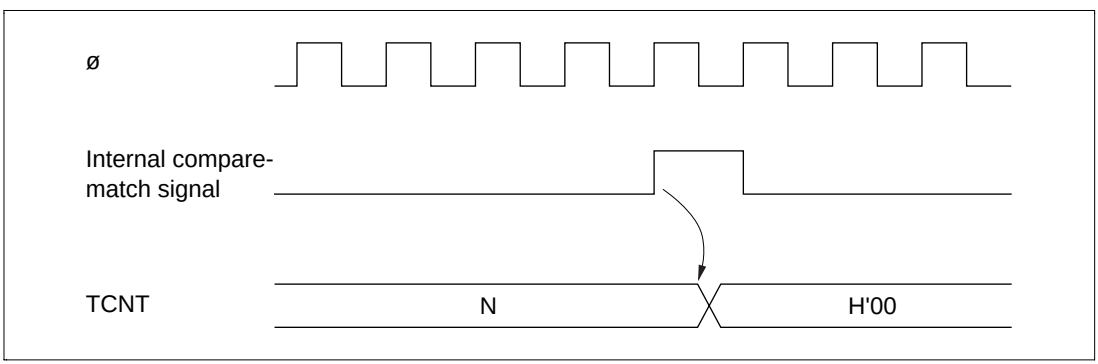


Figure 11.6 Timing of Compare-Match Clear

11.3.3 External Reset of TCNT

When the CCLR1 and CCLR0 bits in TCR are both set to 1, the timer counter is cleared on the rising edge of an external reset input. Figure 11.7 shows the timing of this operation. The timer reset pulse width must be at least 1.5 system clock periods.

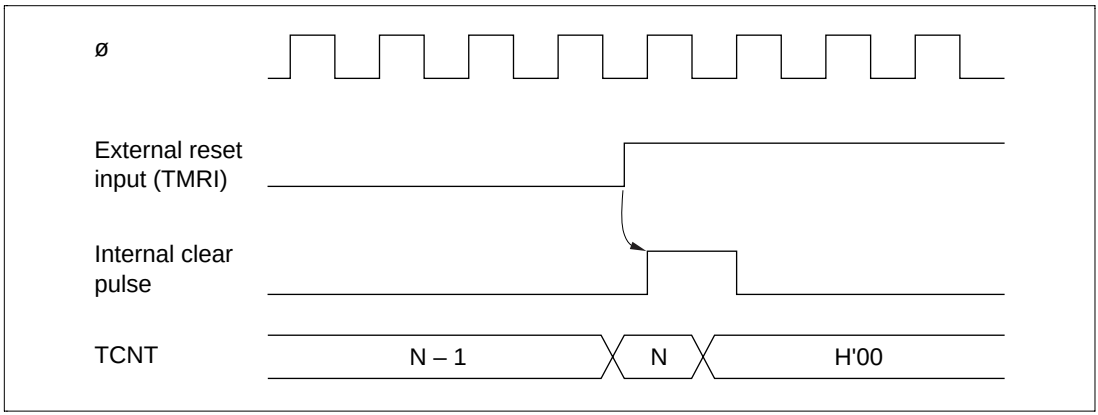


Figure 11.7 Timing of External Reset

11.3.4 Setting of TCSR Overflow Flag (OVF)

The overflow flag (OVF) is set to 1 when the timer count overflows (changes from H'FF to H'00). Figure 11.8 shows the timing of this operation.

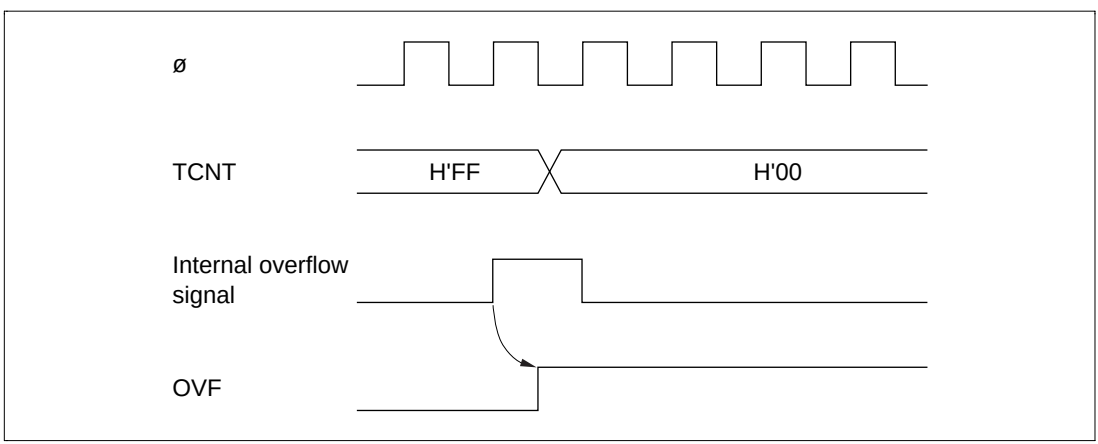


Figure 11.8 Setting of Overflow Flag (OVF)

11.4 Interrupts

Each channel in the 8-bit timer can generate three types of interrupts: compare-match A and B (CMIA and CMIB), and overflow (OVI). Each interrupt can be enabled or disabled by an enable bit in TCR. Independent signals are sent to the interrupt controller for each interrupt. Table 11.3 lists information about these interrupts.

Table 11.3 8-Bit Timer Interrupts

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

11.5 Sample Application

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

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

With these settings, the 8-bit timer provides output of pulses at a rate determined by TCORA with a pulse width determined by TCORB. No software intervention is required.

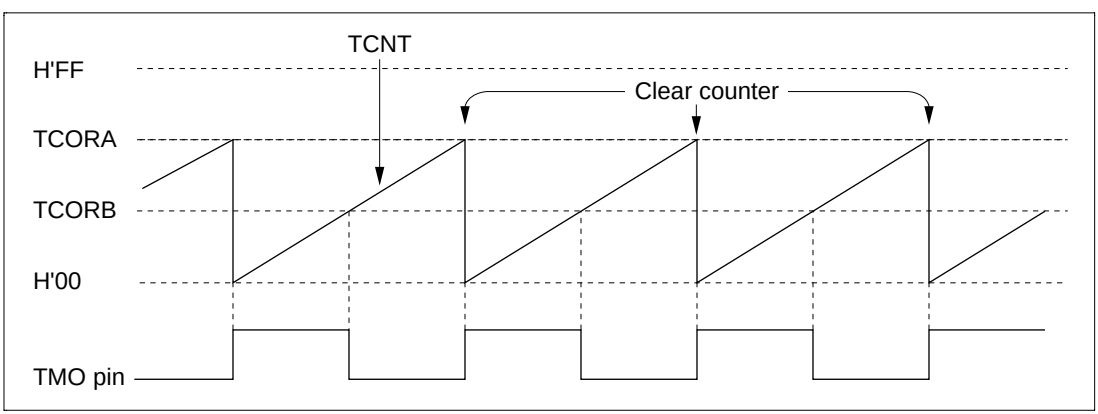


Figure 11.9 Example of Pulse Output

11.6 Application Notes

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

11.6.1 Contention between TCNT Write and Clear

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

Figure 11.10 shows this type of contention.

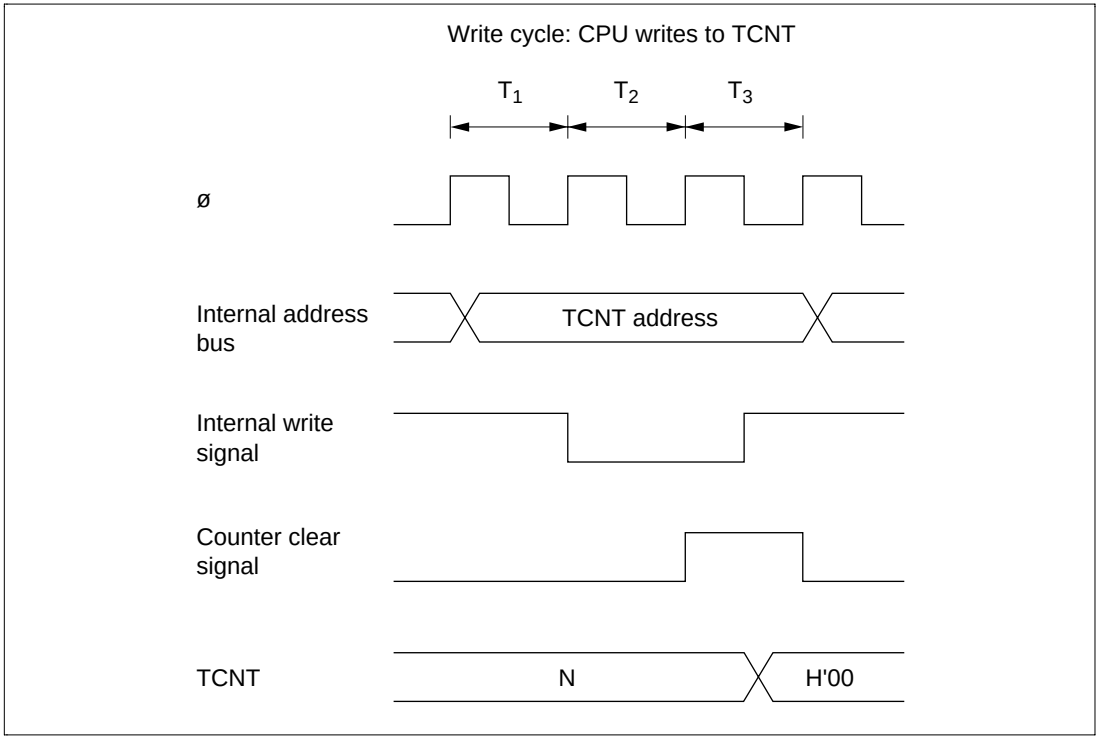


Figure 11.10 TCNT Write-Clear Contention

11.6.2 Contention between TCNT Write and Increment

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

Figure 11.11 shows this type of contention.

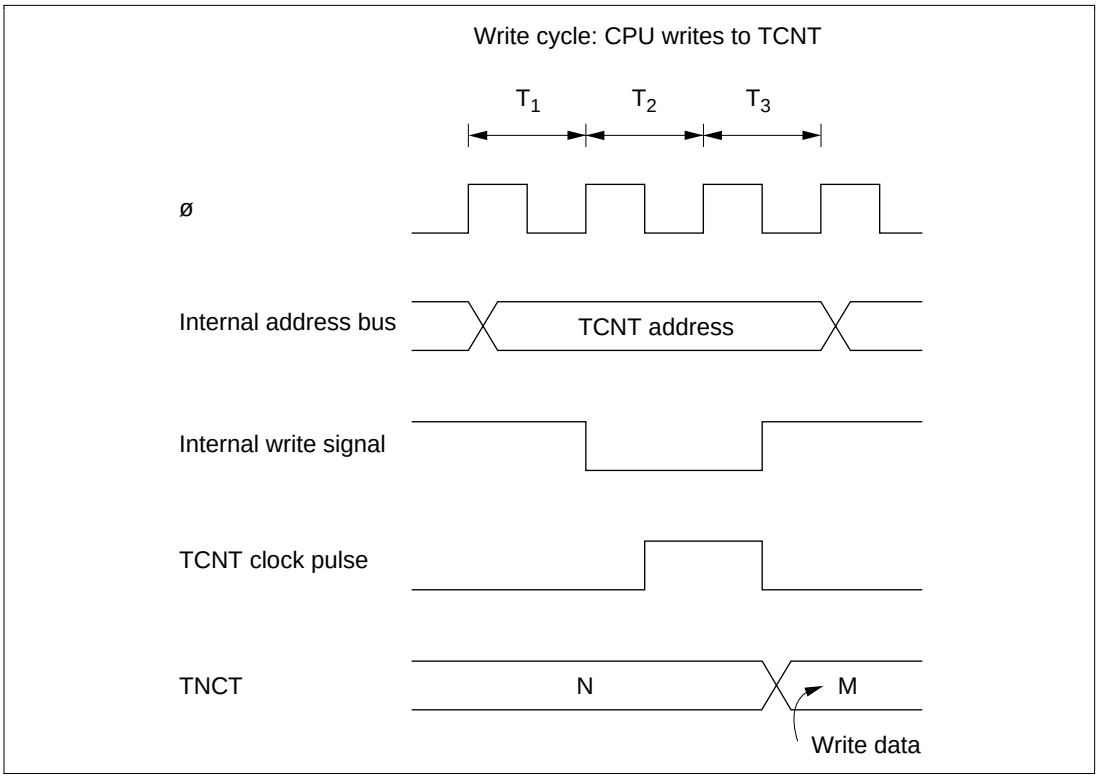


Figure 11.11 TCNT Write-Increment Contention

11.6.3 Contention between TCOR Write and Compare-Match

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

Figure 11.12 shows this type of contention.

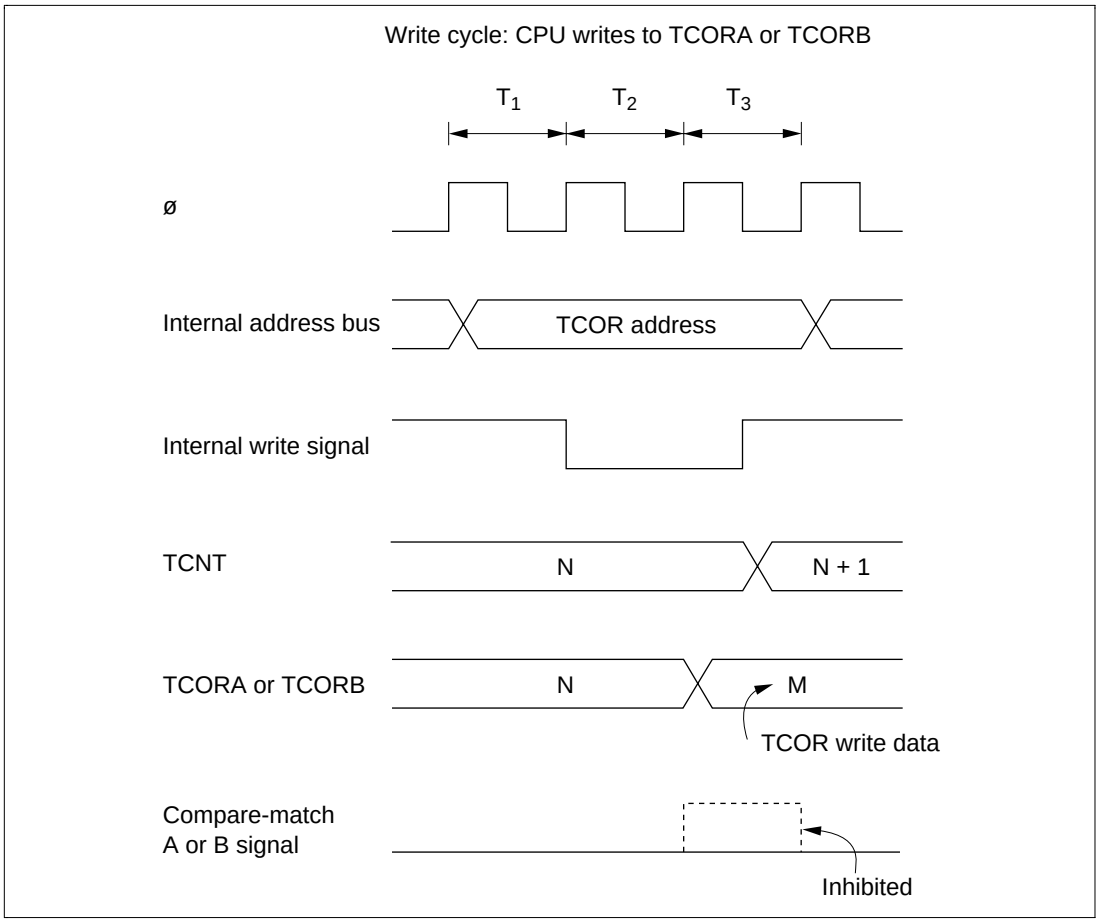


Figure 11.12 Contention between TCOR Write and Compare-Match

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

If identical time constants are written in TCORA and TCORB, causing compare-match A and B to occur simultaneously, any conflict between the output selections for compare-match A and B is resolved by following the priority order in table 11.4.

Table 11.4 Priority of Timer Output

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

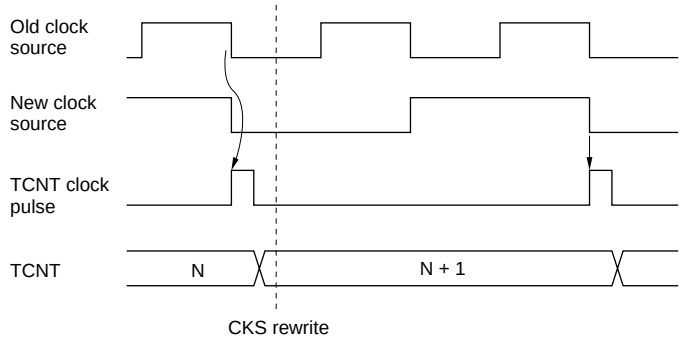
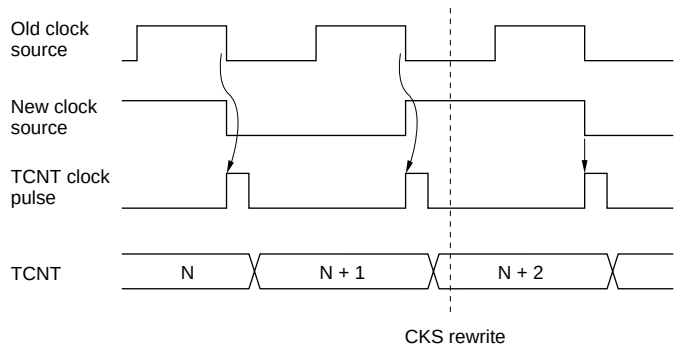
11.6.5 Incrementation Caused by Changing of Internal Clock Source

When an internal clock source is changed, the changeover may cause the timer counter to increment. This depends on the time at which the clock select bits (CKS1, CKS0) are rewritten, as shown in table 11.5.

The pulse that increments the timer counter is generated at the falling edge of the internal clock source signal. If clock sources are changed when the old source is high and the new source is low, as in case number 3 in table 11.5, the changeover generates a falling edge that triggers the TCNT clock pulse and increments the timer counter.

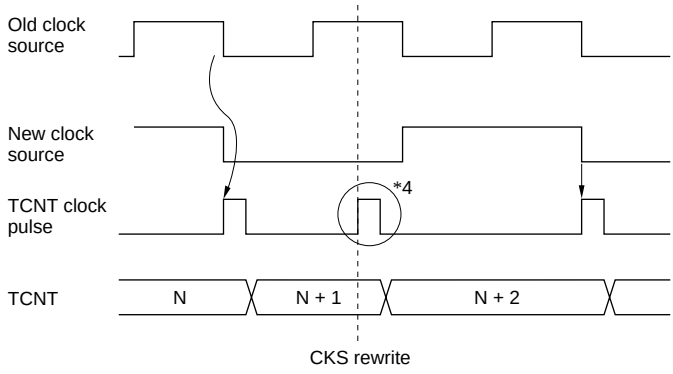
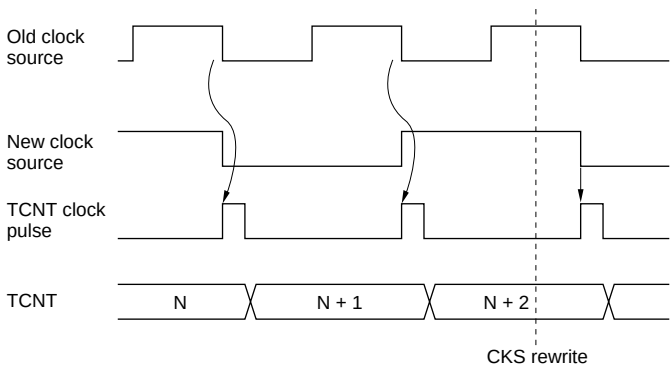
Switching between an internal and external clock source can also cause the timer counter to increment.

Table 11.5 Effect of Changing Internal Clock Sources

No.	Description	Timing
1	Low → low*[1]: Clock select bits are rewritten while both clock sources are low.	
2	Low → high*[2]: Clock select bits are rewritten while old clock source is low and new clock source is high.	

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

Table 11.5 Effect of Changing Internal Clock Sources (cont)

No.	Description	Timing
3	High → low*[3]: Clock select bits are rewritten while old clock source is high and new clock source is low.	
4	High → high: Clock select bits are rewritten while both clock sources are high.	

Notes: 3. Including a transition from high to the stopped state.
4. The switching of clock sources is regarded as a falling edge that increments TCNT.

Section 12 Programmable Timing Pattern Controller

12.1 Overview

The H8/3318 has a built-in programmable timing pattern controller (TPC) that provides pulse outputs by using free-running timer 0 or 1 (FRT0 or FRT1) as a time base. The TPC pulse outputs are divided into 4-bit groups (group 3 to group 0) that can operate simultaneously and independently.

12.1.1 Features

The TPC has the following features:

- 16-bit output data
  Maximum 16-bit data can be output. TPC output can be enabled on a bit-by-bit basis.
- Four output groups
  Output trigger signals can be selected in 4-bit groups to provide up to four different 4-bit outputs.
- Selectable output trigger signals
  Output trigger signals can be selected for each group from the compare-match signals of FRT0 and FRT1.
- Non-overlap mode
  A non-overlap margin can be provided between pulse outputs.
- Can operate together with the data transfer unit (DTU)
  The compare-match signals selected as trigger signals can activate the DTU for sequential output of data without CPU intervention.

12.1.2 Block Diagram

Figure 12.1 shows a block diagram of the TPC.

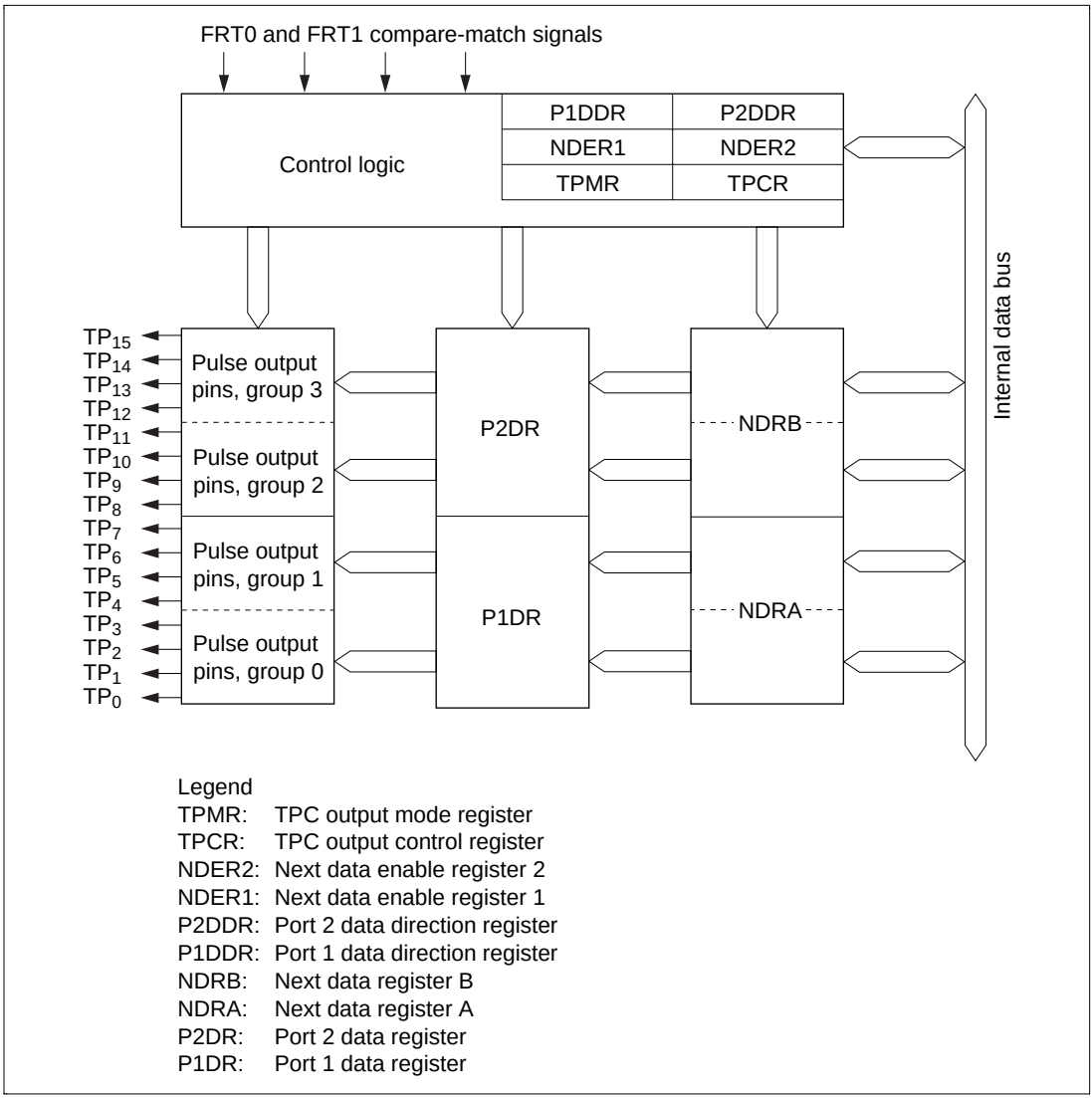


Figure 12.1 TPC Block Diagram

12.1.3 TPC Pins

Table 12.1 summarizes the TPC output pins.

Table 12.1 TPC Pins

Name	Symbol	I/O	Function
TPC output 0	TP_0	Output	Group 0 pulse output
TPC output 1	TP_1	Output	
TPC output 2	TP_2	Output	
TPC output 3	TP_3	Output	
TPC output 4	TP_4	Output	Group 1 pulse output
TPC output 5	TP_5	Output	
TPC output 6	TP_6	Output	
TPC output 7	TP_7	Output	
TPC output 8	TP_8	Output	Group 2 pulse output
TPC output 9	TP_9	Output	
TPC output 10	TP_{10}	Output	
TPC output 11	TP_{11}	Output	
TPC output 12	TP_{12}	Output	Group 3 pulse output
TPC output 13	TP_{13}	Output	
TPC output 14	TP_{14}	Output	
TPC output 15	TP_{15}	Output	

12.1.4 Registers

Table 12.2 summarizes the TPC registers.

Table 12.2 TPC 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)*[1]	H'00	H'FFB2
Port 2 data direction register	P2DDR	W	H'00	H'FFB1
Port 2 data register	P2DR	R/(W)*[1]	H'00	H'FFB3
TPC output mode register	TPMR	R/W	H'F0	H'FFEA
TPC output control register	TPCR	R/W	H'FF	H'FFEB
Next data enable register 2	NDER2	R/W	H'00	H'FFCD
Next data enable register 1	NDER1	R/W	H'00	H'FFD5
Next data register A	NDRA	R/W	H'00	H'FFCF/H'FFD7*[2]
Next data register B	NDRB	R/W	H'00	H'FFCE/H'FFD6*[2]

Notes: 1. Bits used for TPC output cannot be written to.

2. The NDRA address is H'FFCF when the same output trigger is selected for TPC output groups 0 and 1 by settings in TPCR. When the output triggers are different, the NDRA address is H'FFD7 for group 0 and H'FFA5 for group 1. Similarly, the address of NDRB is H'FFCE when the same output trigger is selected for TPC output groups 2 and 3 by settings in TPCR. When the output triggers are different, the NDRB address is H'FFD6 for group 2 and H'FFCE for group 3.

12.2 Register Descriptions

12.2.1 Port 1 Data Direction Register (P1DDR)

P1DDR is an 8-bit write-only register that selects input or output for each pin in port 1.

Bit	7	6	5	4	3	2	1	0
	$P1_7$DDR	$P1_6$DDR	$P1_5$DDR	$P1_4$DDR	$P1_3$DDR	$P1_2$DDR	$P1_1$DDR	$P1_0$DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 1 is multiplexed with pins TP_7 to TP_0. Bits corresponding to pins used for TPC output must be set to 1. For further information about P1DDR, see section 8.2, Port 1.

12.2.2 Port 1 Data Register (P1DR)

P1DR is an 8-bit readable/writable register that stores TPC output data for groups 0 and 1, when these TPC output groups are used.

Bit	7	6	5	4	3	2	1	0
	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Bits selected for TPC output by NDER1 settings become read-only bits.

For further information about P1DR, see section 8.2, Port 1.

12.2.3 Port 2 Data Direction Register (P2DDR)

P2DDR is an 8-bit write-only register that selects input or output for each pin in port 2.

Bit	7	6	5	4	3	2	1	0
	$P2_7$DDR	$P2_6$DDR	$P2_5$DDR	$P2_4$DDR	$P2_3$DDR	$P2_2$DDR	$P2_1$DDR	$P2_0$DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 2 is multiplexed with pins TP_{15} to TP_8. Bits corresponding to pins used for TPC output must be set to 1. For further information about P2DDR, see section 8.3, Port 2.

12.2.4 Port 2 Data Register (P2DR)

P2DR is an 8-bit readable/writable register that stores TPC output data for groups 2 and 3, when these TPC output groups are used.

Bit	7	6	5	4	3	2	1	0
	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Bits selected for TPC output by NDER2 settings become read-only bits.

For further information about P2DR, see section 8.3, Port 2.

12.2.5 Next Data Register A (NDRA)

NDRA is an 8-bit readable/writable register that stores the next output data for TPC output groups 1 and 0 (pins TP_7 to TP_0). During TPC output, when an FRT0 or FRT1 compare-match event specified in TPCR occurs, NDRA contents are transferred to the corresponding bits in P1DR. The address of NDRA differs depending on whether TPC output groups 0 and 1 have the same output trigger or different output triggers.

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

Same Trigger for TPC Output Groups 0 and 1: If TPC output groups 0 and 1 are triggered by the same compare-match event, the NDRA address is H'FFCF. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Address H'FFD7 consists entirely of reserved bits that cannot be modified and always read 1.

- **Address H'FFCF**

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
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

Bits 7 to 4—Next Data 7 to 4 (NDR7 to NDR4): These bits store the next output data for TPC output group 1.

Bits 3 to 0—Next Data 3 to 0 (NDR3 to NDR0): These bits store the next output data for TPC output group 0.

- **Address H'FFD7**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

Bits 7 to 0—Reserved: These bits cannot be modified and are always read as 1.

Different Triggers for TPC Output Groups 0 and 1: If TPC output groups 0 and 1 are triggered by different compare-match events, the address of the upper 4 bits of NDRA (group 1) is H'FFCF and the address of the lower 4 bits (group 0) is H'FFD7. Bits 3 to 0 of address H'FFCF and bits 7 to 4 of address H'FFD7 are reserved bits that cannot be modified and always read 1.

- **Address H'FFCF**

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Bits 7 to 4—Next Data 7 to 4 (NDR7 to NDR4): These bits store the next output data for TPC output group 1.

Bits 3 to 0—Reserved: These bits cannot be modified and are always read as 1.

- **Address H'FFD7**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bits 3 to 0—Next Data 3 to 0 (NDR3 to NDR0): These bits store the next output data for TPC output group 0.

12.2.6 Next Data Register B (NDRB)

NDRB is an 8-bit readable/writable register that stores the next output data for TPC output groups 3 and 2 (pins TP_{15} to TP_8). During TPC output, when an FRT compare-match event specified in TPCR occurs, NDRB contents are transferred to the corresponding bits in P2DR. The address of NDRB differs depending on whether TPC output groups 2 and 3 have the same output trigger or different output triggers.

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

Same Trigger for TPC Output Groups 2 and 3: If TPC output groups 2 and 3 are triggered by the same compare-match event, the NDRB address is H'FFCE. The upper 4 bits belong to group 3 and the lower 4 bits to group 2. Address H'FFD6 consists entirely of reserved bits that cannot be modified and always read 1.

- **Address H'FFCE**

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
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

Bits 7 to 4—Next Data 15 to 12 (NDR15 to NDR12): These bits store the next output data for TPC output group 3.

Bits 3 to 0—Next Data 11 to 8 (NDR11 to NDR8): These bits store the next output data for TPC output group 2.

- **Address H'FFD6**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

Bits 7 to 0—Reserved: These bits cannot be modified and are always read as 1.

Different Triggers for TPC Output Groups 2 and 3: If TPC output groups 2 and 3 are triggered by different compare-match events, the address of the upper 4 bits of NDRB (group 3) is H'FFCE and the address of the lower 4 bits (group 2) is H'FFD6. Bits 3 to 0 of address H'FFCE and bits 7 to 4 of address H'FFD6 are reserved bits that cannot be modified and always read 1.

- **Address H'FFCE**

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Bits 7 to 4—Next Data 15 to 12 (NDR15 to NDR12): These bits store the next output data for TPC output group 3.

Bits 3 to 0—Reserved: These bits cannot be modified and are always read as 1.

- **Address H'FFD6**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Bits 7 to 4—Reserved: These bits cannot be modified and are always read as 1.

Bits 3 to 0—Next Data 11 to 8 (NDR11 to NDR8): These bits store the next output data for TPC output group 2.

12.2.7 Next Data Enable Register 1 (NDER1)

NDER1 is an 8-bit readable/writable register that enables or disables TPC output groups 1 and 0 (TP_7 to TP_0) on a bit-by-bit basis.

Bit	7	6	5	4	3	2	1	0
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
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

If a bit is enabled for TPC output by NDER1, then when the FRT compare-match event selected in the TPC output control register (TPCR) occurs, the NDRA value is automatically transferred to the corresponding P1DR bit, updating the output value. If TPC output is disabled, the bit value is not transferred from NDRA to P1DR and the output value does not change.

NDER1 is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 0—Next Data Enable 7 to 0 (NDER7 to NDER0): These bits enable or disable TPC output groups 1 and 0 (TP_7 to TP_0) on a bit-by-bit basis.

Bits 7 to 0: NDER7 to NDER0	Description
0	TPC outputs TP_7 to TP_0 are disabled (NDR7 to NDR0 are not transferred to $P1_7$ to $P1_0$) (Initial value)
1	TPC outputs TP_7 to TP_0 are enabled (NDR7 to NDR0 are transferred to $P1_7$ to $P1_0$)

12.2.8 Next Data Enable Register 2 (NDER2)

NDER2 is an 8-bit readable/writable register that enables or disables TPC output groups 3 and 2 (TP_{15} to TP_8) on a bit-by-bit basis.

Bit	7	6	5	4	3	2	1	0
	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
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

If a bit is enabled for TPC output by NDER2, then when the FRT compare-match event selected in the TPC output control register (TPCR) occurs, the NDRB value is automatically transferred to the corresponding P2DR bit, updating the output value. If TPC output is disabled, the bit value is not transferred from NDRB to P2DR and the output value does not change.

NDER2 is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits enable or disable TPC output groups 3 and 2 (TP_{15} to TP_8) on a bit-by-bit basis.

Bits 7 to 0: NDER15 to NDER8	Description
0	TPC outputs TP_{15} to TP_8 are disabled (NDR15 to NDR8 are not transferred to $P2_7$ to $P2_0$) (Initial value)
1	TPC outputs TP_{15} to TP_8 are enabled (NDR15 to NDR8 are transferred to $P2_7$ to $P2_0$)

12.2.9 TPC Output Control Register (TPCR)

TPCR is an 8-bit readable/writable register that selects output trigger signals for TPC outputs on a group-by-group basis.

Bit	7	6	5	4	3	2	1	0
	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
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

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

Bits 7 and 6—Group 3 Compare Match Select 1 and 0 (G3CMS1, G3CMS0): These bits select the compare-match event that triggers TPC output group 3 (TP_{15} to TP_{12}).

Bit 7 G3CMS1	Bit 6 G3CMS0	Description
0	0	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare-match A in FRT1
	1	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare-match A in FRT0
1	0	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare-match B in FRT1 (compare-match A can be used for non-overlap operation)
	1	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare-match B in FRT0 (compare-match A can be used for non-overlap operation) (Initial value)

Bits 5 and 4—Group 2 Compare Match Select 1 and 0 (G2CMS1, G2CMS0): These bits select the compare-match event that triggers TPC output group 2 (TP_{11} to TP_{8}).

Bit 5 G2CMS1	Bit 4 G2CMS0	Description
0	0	TPC output group 2 (TP_{11} to TP_{8}) is triggered by compare-match A in FRT1
	1	TPC output group 2 (TP_{11} to TP_{8}) is triggered by compare-match A in FRT0
1	0	TPC output group 2 (TP_{11} to TP_{8}) is triggered by compare-match B in FRT1 (compare-match A can be used for non-overlap operation)
	1	TPC output group 2 (TP_{11} to TP_{8}) is triggered by compare-match B in FRT0 (compare-match A can be used for non-overlap operation) (Initial value)

Bits 3 and 2—Group 1 Compare Match Select 1 and 0 (G1CMS1, G1CMS0): These bits select the compare-match event that triggers TPC output group 1 (TP_7 to TP_4).

Bit 3 G1CMS1	Bit 2 G1CMS0	Description
0	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare-match A in FRT1
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare-match A in FRT0
1	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare-match B in FRT1 (compare-match A can be used for non-overlap operation)
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare-match B in FRT0 (compare-match A can be used for non-overlap operation) (Initial value)

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): These bits select the compare-match event that triggers TPC output group 0 (TP_3 to TP_0).

Bit 1 G0CMS1	Bit 0 G0CMS0	Description
0	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare-match A in FRT1
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare-match A in FRT0
1	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare-match B in FRT1 (compare-match A can be used for non-overlap operation)
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare-match B in FRT0 (compare-match A can be used for non-overlap operation) (Initial value)

12.2.10 TPC Output Mode Register (TPMR)

TPMR is an 8-bit readable/writable register that selects normal or non-overlapping TPC output for each group.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

The output trigger period of a non-overlapping TPC output waveform is set in output compare register A (OCRA) in the FRT0 or FRT1 channel selected for output triggering. The non-overlap margin is set in output compare register B (OCRB). The output values change at compare-match A and B. For details see section 12.3.4, Non-Overlapping TPC Output.

TPMR is initialized to H'F0 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 4—Reserved: These bits cannot be modified, and are always read as 1.

Bit 3—Group 3 Non-Overlap (G3NOV): Selects normal or non-overlapping TPC output for group 3 (TP_{15} to TP_{12}).

Bit 3 G3NOV	Description	
0	Normal TPC output in group 3 (output values change at the selected compare-match)	(Initial value)
1	Non-overlapping TPC output in group 3 (0 and 1 output at compare-match A and B, respectively)	

Bit 2—Group 2 Non-Overlap (G2NOV): Selects normal or non-overlapping TPC output for group 2 (TP_{11} to TP_{8}).

Bit 2 G2NOV	Description	
0	Normal TPC output in group 2 (output values change at the selected compare-match)	(Initial value)
1	Non-overlapping TPC output in group 2 (0 and 1 output at compare-match A and B, respectively)	

Bit 1—Group 1 Non-Overlap (G1NOV): Selects normal or non-overlapping TPC output for group 1 (TP_7 to TP_4).

Bit 1 G1NOV	Description	
0	Normal TPC output in group 1 (output values change at the selected compare-match)	(Initial value)
1	Non-overlapping TPC output in group 1 (0 and 1 output at compare-match A and B, respectively)	

Bit 0—Group 0 Non-Overlap (G0NOV): Selects normal or non-overlapping TPC output for group 0 (TP_3 to TP_0).

Bit 0 G0NOV	Description	
0	Normal TPC output in group 0 (output values change at the selected compare-match)	(Initial value)
1	Non-overlapping TPC output in group 0 (0 and 1 output at compare-match A and B, respectively)	

12.3 Operation

12.3.1 Overview

When corresponding bits in P1DDR or P2DDR and NDER1 or NDER2 are set to 1, TPC output is enabled. The TPC output initially consists of the corresponding P1DR or P2DR contents. When a compare-match event selected in TPCR occurs, the corresponding NDRA or NDRB bit contents are transferred to P1DR or P2DR to update the output values.

Figure 12.2 illustrates the TPC output operation. Table 12.3 summarizes the TPC operating conditions.

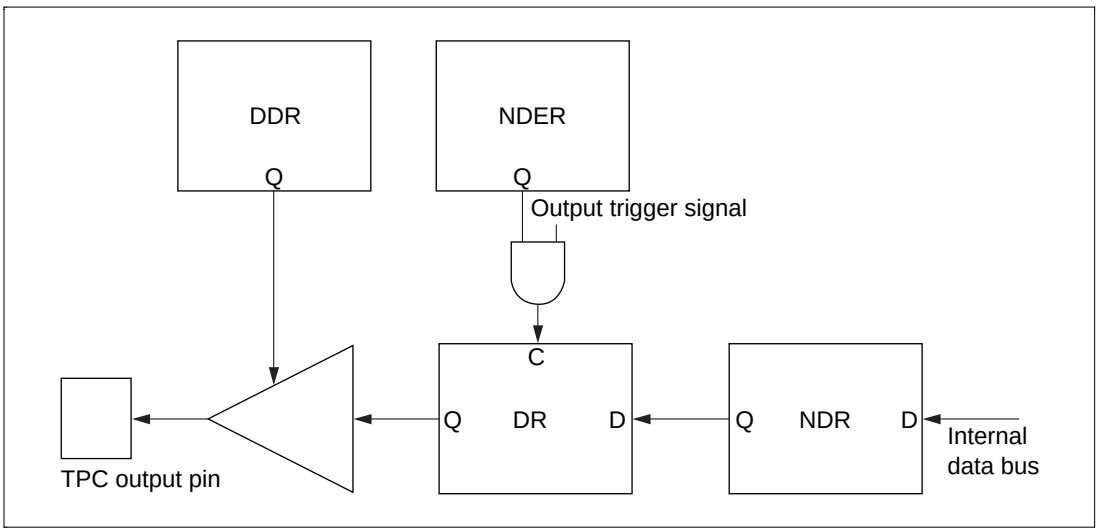


Figure 12.2 TPC Output Operation

Table 12.3 TPC Operating Conditions

NDER	DDR	Pin Function
0	0	General-purpose input port
	1	General-purpose output port
1	0	General-purpose input port (but software cannot write to the DR bit, and when compare-match occurs, the NDR bit value is transferred to the DR bit)
	1	TPC pulse output

Sequential output of up to 16-bit patterns is possible by writing new output data to NDRA and NDRB before the next compare-match. For information on non-overlapping operation, see section 12.3.4, Non-Overlapping TPC Output.

12.3.2 Output Timing

If TPC output is enabled, NDRA/NDRB contents are transferred to P1DR/P2DR and output when the selected compare-match event occurs. Figure 12.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare-match A.

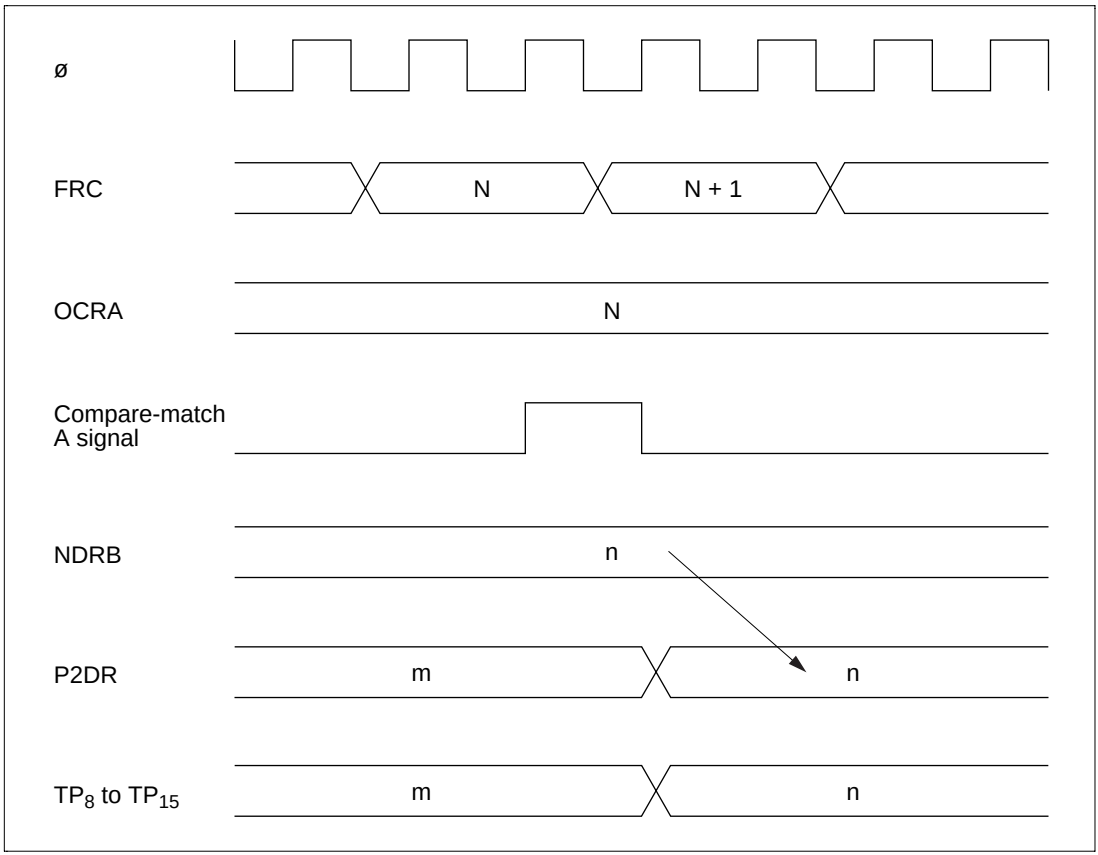
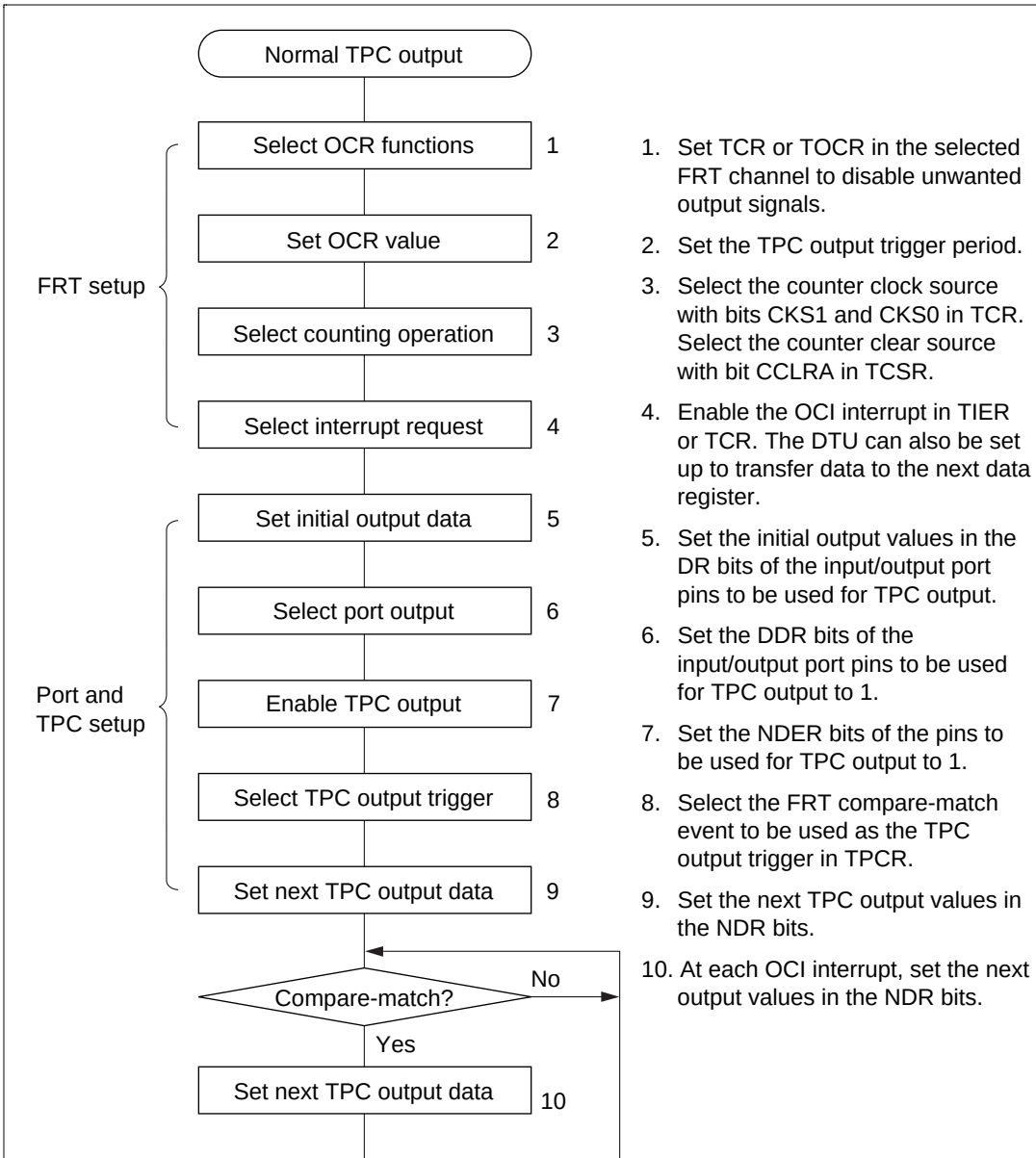


Figure 12.3 Timing of Transfer of Next Data Register Contents and Output (Example)

12.3.3 Normal TPC Output

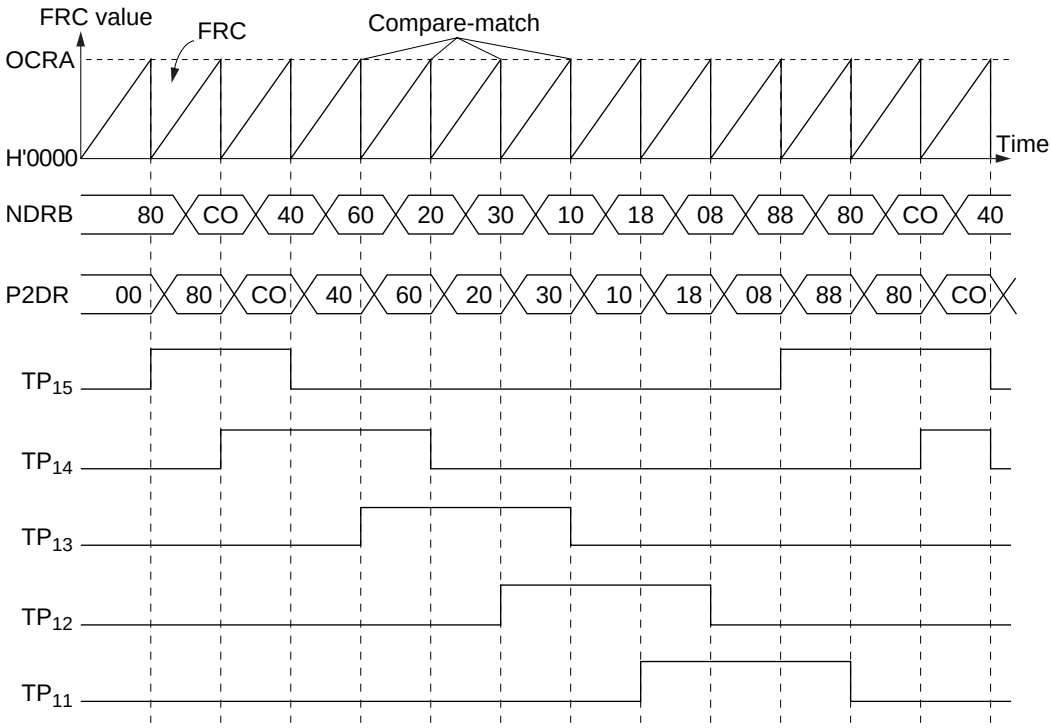
Sample Setup Procedure for Normal TPC Output: Figure 12.4 shows a sample procedure for setting up normal TPC output.



Note: Avoid allowing compare-match to occur while steps 4 to 9 are being carried out.

Figure 12.4 Setup Procedure for Normal TPC Output (Example)

Example of Normal TPC Output (Example of Five-Phase Pulse Output): Figure 12.5 shows an example in which the TPC is used for cyclic five-phase pulse output.

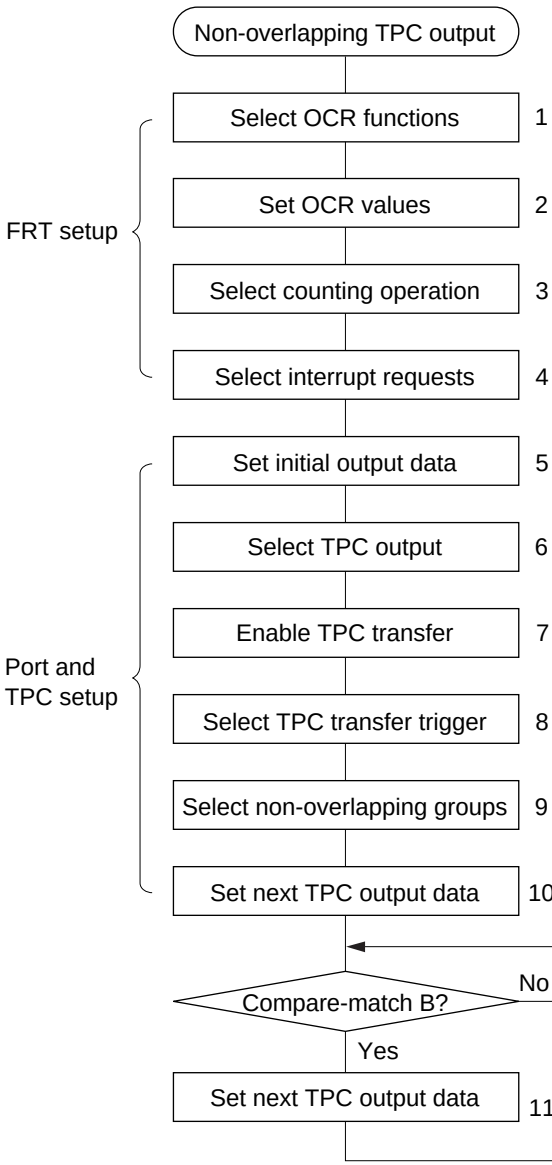


1. The FRT channel to be used as the output trigger channel is set up so that the counter will be cleared by compare-match A. The trigger period is set in OCRA. The OCIEA bit is set to 1 in TIER or TCR to enable the compare-match A interrupt.
2. H'F8 is written in P2DDR and NDER2, and bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 are set in TPCR to select compare-match in the FRT channel set up in step 1 as the output trigger. Output data H'80 is written in NDRB.
3. When compare-match A occurs, the NDRB contents are transferred to P2DR and output. The compare-match A (OCIA) interrupt service routine writes the next output data (H'C0) in NDRB.
4. Five-phase overlapping pulse output (one or two phases active at a time) can be obtained by writing H'40, H'60, H'20, H'30, H'10, H'18, H'08, H'88… at successive OCIA interrupts. If the DTU is set for activation by this interrupt, pulse output can be obtained without loading the CPU.

Figure 12.5 Normal TPC Output Example (Five-Phase Pulse Output)

12.3.4 Non-Overlapping TPC Output

Sample Setup Procedure for Non-Overlapping TPC Output: Figure 12.6 shows a sample procedure for setting up non-overlapping TPC output.

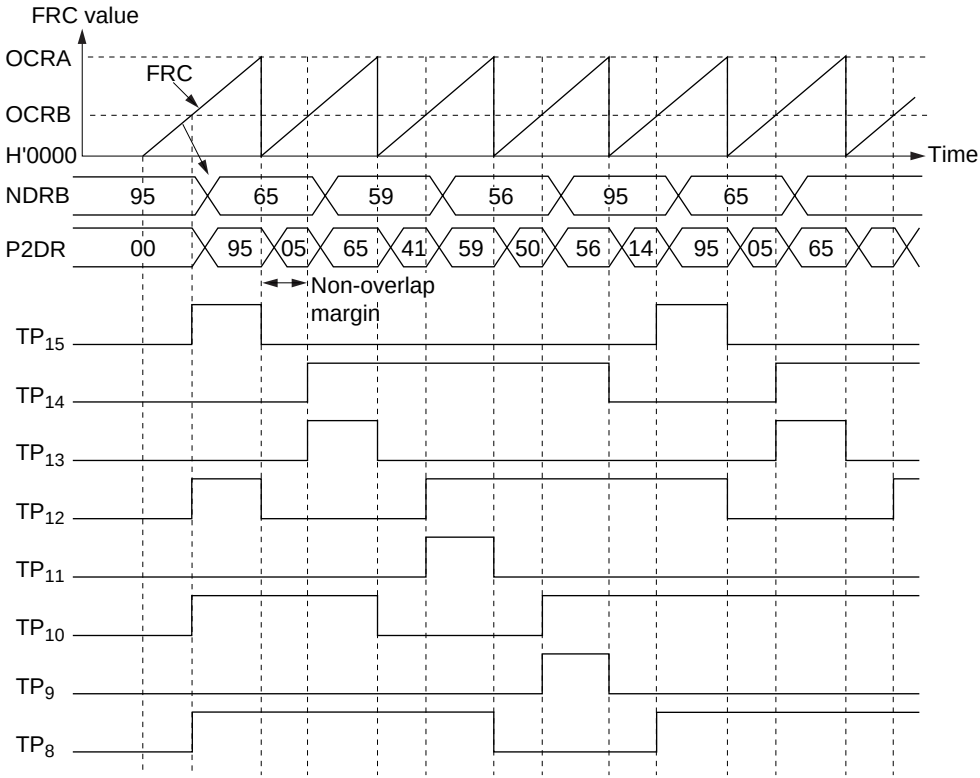


1. Set TCR or TOCR in the selected FRT channel to disable unwanted output signals.
2. Set the TPC output trigger period in OCRA and the non-overlap margin in OCRB.
3. Select the counter clock source with bits CKS1 and CKS0 in TCR. Select the counter clear source with bit CCLRA in TCSR.
4. Enable the OCIB interrupt in TIER or TCR. The DTU can also be set up to transfer data to the next data register.
5. Set the initial output values in the DR bits of the input/output port pins to be used for TPC output.
6. Set the DDR bits of the input/output port pins to be used for TPC output to 1.
7. Set the NDER bits of the pins to be used for TPC output to 1.
8. In TPCR, select the FRT compare-match event to be used as the TPC output trigger.
9. In TPMR, select the groups that will operate in non-overlap mode.
10. Set the next TPC output values in the NDR bits.
11. At each OCIB interrupt, write the next output value in the NDR bits.

Note: Avoid allowing compare-match to occur while steps 4 to 10 are being carried out.

Figure 12.6 Setup Procedure for Non-Overlapping TPC Output (Example)

Example of Non-Overlapping TPC Output (Example of Four-Phase Complementary Non-Overlapping Output): Figure 12.7 shows an example of the use of TPC output for four-phase complementary non-overlapping pulse output.



1. The FRT channel to be used as the output trigger channel is set up so that the counter will be cleared by compare-match A. The trigger period is set in OCRA. The non-overlap margin is set in OCRB. The OCIEB bit is set to 1 in TIER or TCR to enable OCIB interrupts.
2. H'FF is written in P2DDR and NDER2, and bits G3CMS1, G3CMS0, G2CMS1, and G2CMS0 are set in TPCR to select compare-match in the FRT channel set up in step 1 as the output trigger. Bits G3NOV and G2NOV are set to 1 in TPMR to select non-overlapping output. Output data H'95 is written in NDRB.
3. After FRT operation begins, when compare-match A occurs, outputs change from 1 to 0. When compare-match B occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value of OCRB). The OCIB interrupt service routine writes the next output data (H'65) in NDRB.
4. Four-phase complementary non-overlapping pulse output can be obtained by writing H'59, H'56, H'95… at successive OCIB interrupts. If the DTU is set for activation by this interrupt, pulse output can be obtained without loading the CPU.

Figure 12.7 Non-Overlapping TPC Output Example (Four-Phase Complementary Non-Overlapping Pulse Output)

12.4 Application Notes

12.4.1 Operation of TPC Output Pins

TP_0 to TP_{15} are multiplexed with address output in the expanded modes. When address output is enabled in mode 1, the corresponding pins cannot be used for TPC output. The data transfer from NDR bits to DR bits takes place, however, regardless of the usage of the pin.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

12.4.2 Note on Non-Overlapping Output

During non-overlapping operation, the transfer of NDR bit values to DR bits takes place as follows.

1. NDR bits are always transferred to DR bits at compare-match B.
2. At compare-match A, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 12.8 illustrates the non-overlapping TPC output operation.

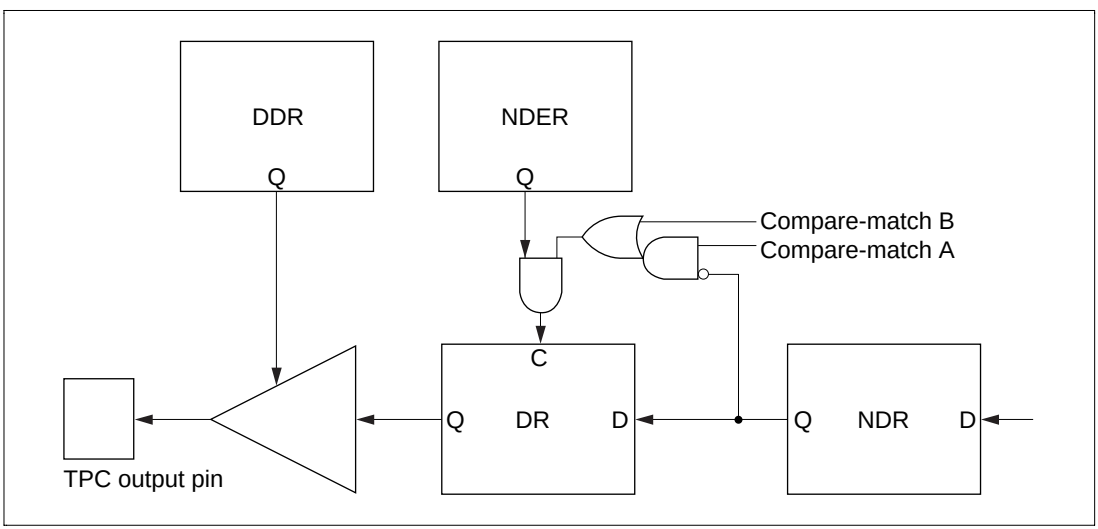


Figure 12.8 Non-Overlapping TPC Output

Therefore, 0 data can be transferred ahead of 1 data by making compare-match A occur before compare-match B. NDR contents should not be altered during the interval from compare-match A to compare-match B (the non-overlap margin).

This can be accomplished by having the OCIB interrupt service routine write the next data in NDR, or by having the OCIB interrupt activate the DTU. The next data must be written before the next compare-match A occurs.

Figure 12.9 shows the timing relationships.

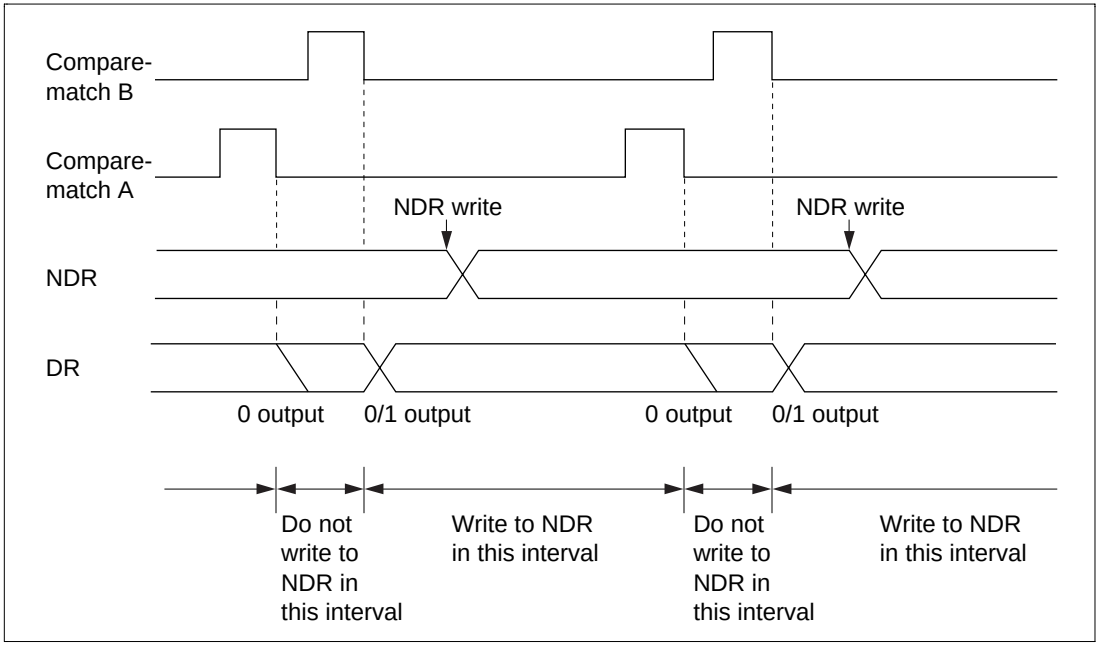


Figure 12.9 Non-Overlapping Operation and NDR Write Timing

Section 13 Watchdog Timer

13.1 Overview

The H8/3318 on-chip watchdog timer (WDT) module can monitor system operation by requesting a nonmaskable interrupt internally if a system crash allows the timer count to overflow. It can also generate an internal chip reset instead of a nonmaskable interrupt.

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

13.1.1 Features

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

13.1.2 Block Diagram

Figure 13.1 is a block diagram of the watchdog timer.

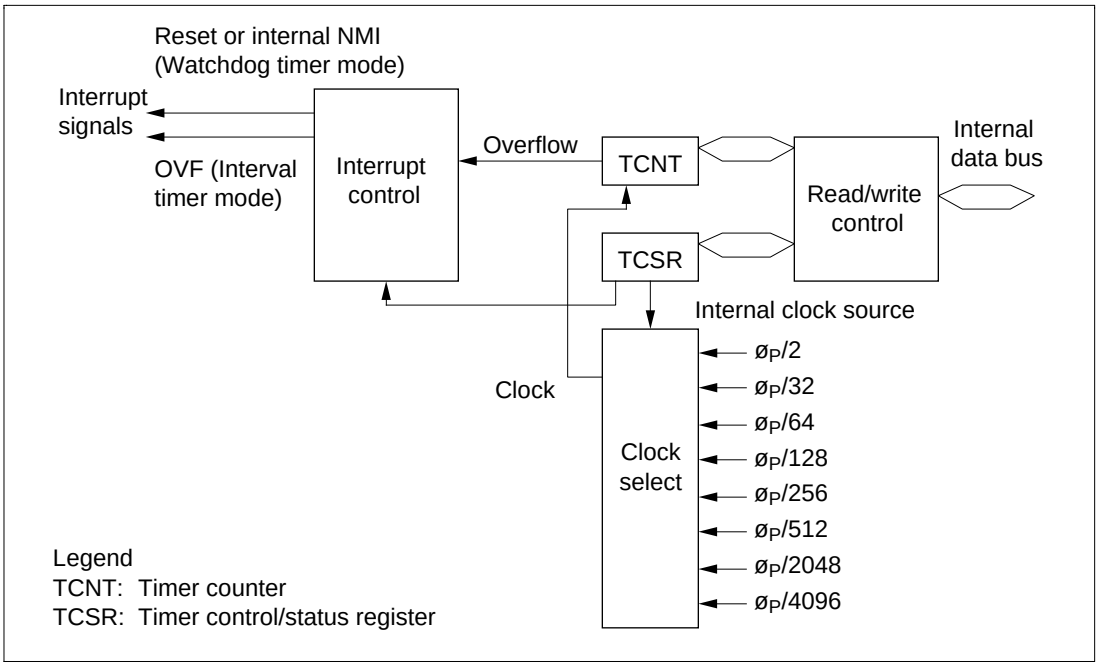


Figure 13.1 Block Diagram of Watchdog Timer

13.1.3 Register Configuration

Table 13.1 lists information on the watchdog timer registers.

Table 13.1 Register Configuration

Name	Abbreviation	R/W	Initial Value	Addresses	
				Write	Read
Timer counter	TCNT	R/W	H'00	H'FFAA (word transfer)	H'FFAB
Timer control/status register	TCSR	R/(W)*	H'10	H'FFAA (word transfer)	H'FFAA

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

13.2 Register Descriptions

13.2.1 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

The watchdog timer counter (TCNT) is a read/write 8-bit up-counter. (TCNT is write-protected by a password. See section 13.2.3, Register Access, for details.) When the timer enable bit (TME) in the timer control/status register (TCSR) is set to 1, the timer counter starts counting pulses of an internal clock source selected by clock select bits 2 to 0 (CKS2 to CKS0) in TCSR. When the count overflows (changes from H'FF to H'00), an overflow flag (OVF) in TCSR is set to 1.

The watchdog timer counter is initialized to H'00 by a reset and when the TME bit is cleared to 0.

Note: TCNT is more difficult to write to than other registers. See section 13.2.3, Register Access, for details.

13.2.2 Timer Control/Status Register (TCSR)

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{\text{IT}}$	TME	—	RST/$\overline{\text{NMI}}$	CKS2	CKS1	CKS0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	—	R/W	R/W	R/W	R/W

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

The watchdog timer control/status register (TCSR) is an 8-bit read/write register that selects the timer mode and clock source and performs other functions. (TCSR is write-protected by a password. See section 13.2.3, Register Access, for details.)

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

Note: TCSR is more difficult to write to than other registers. See section 13.2.3, Register Access, for details.

Bit 7—Overflow Flag (OVF): Indicates that the watchdog timer count has overflowed.

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

Bit 6—Timer Mode Select (WT/$\overline{\text{IT}}$): Selects whether to operate in watchdog timer mode or interval timer mode.

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

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

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

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

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

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

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

The overflow interval listed in the following table is the time from when the watchdog timer counter begins counting from H'00 until an overflow occurs.

In interval timer mode, OVF interrupts are requested at this interval.

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

13.2.3 Register Access

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

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

The TCNT and TCSR registers have the same write address. The write data must be contained in the lower byte of a word written at this address. The upper byte must contain H'5A (password for TCNT) or H'A5 (password for TCSR). See figure 13.2. The result of the access depicted in figure 13.2 is to transfer the write data from the lower byte to TCNT or TCSR.

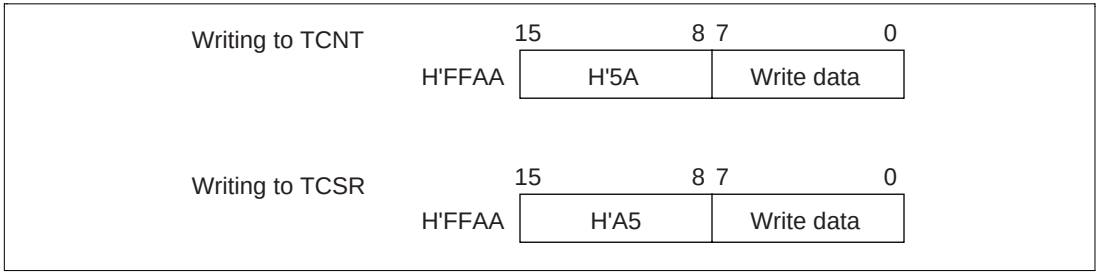


Figure 13.2 Writing to TCNT and TCSR

Reading TCNT and TCSR: The read addresses are H'FFAA for TCSR and H'FFAB for TCNT, as indicated in table 13.2.

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

Table 13.2 Read Addresses of TCNT and TCSR

Read Address	Register
H'FFAA	TCSR
H'FFAB	TCNT

13.3 Operation

13.3.1 Watchdog Timer Mode

The watchdog timer function begins operating when software sets the WT/$\overline{\text{IT}}$ and TME bits to 1 in TCSR. Thereafter, software should periodically rewrite the contents of the timer counter (normally by writing H'00) to prevent the count from overflowing. If a program crash allows the timer count to overflow, the watchdog timer either requests a nonmaskable interrupt (NMI) or resets the entire chip for 518 system clocks (ø), depending on the status of bit 3 of TCSR. Figure 13.3 shows the operation.

NMI requests from the watchdog timer have the same vector as NMI requests from the $\overline{\text{NMI}}$ pin. NMI requests from the watchdog timer and pin $\overline{\text{NMI}}$ should not be handled simultaneously.

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

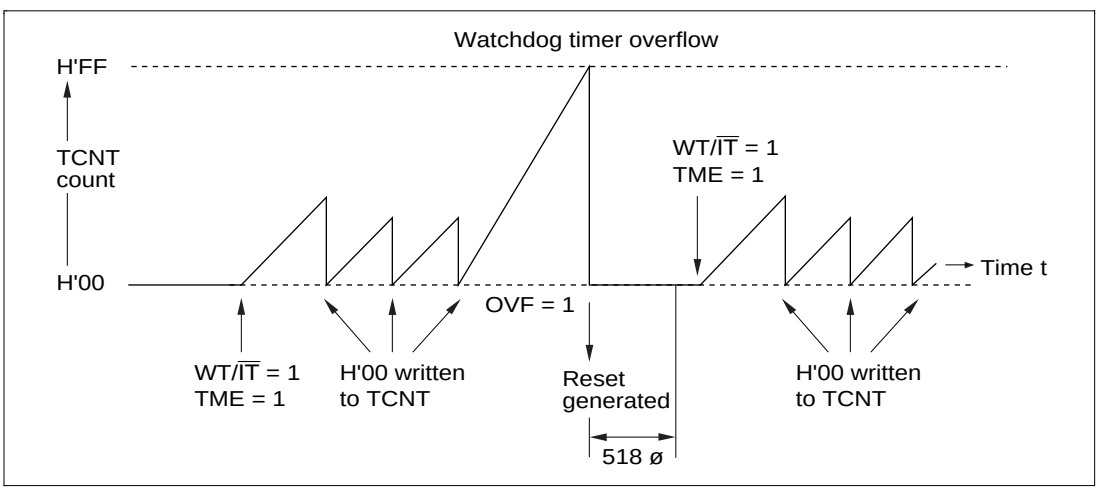


Figure 13.3 Operation in Watchdog Timer Mode

13.3.2 Interval Timer Mode

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

In interval timer mode, an OVF request is generated each time the timer count overflows. This function can be used to generate OVF requests at regular intervals. See figure 13.4.

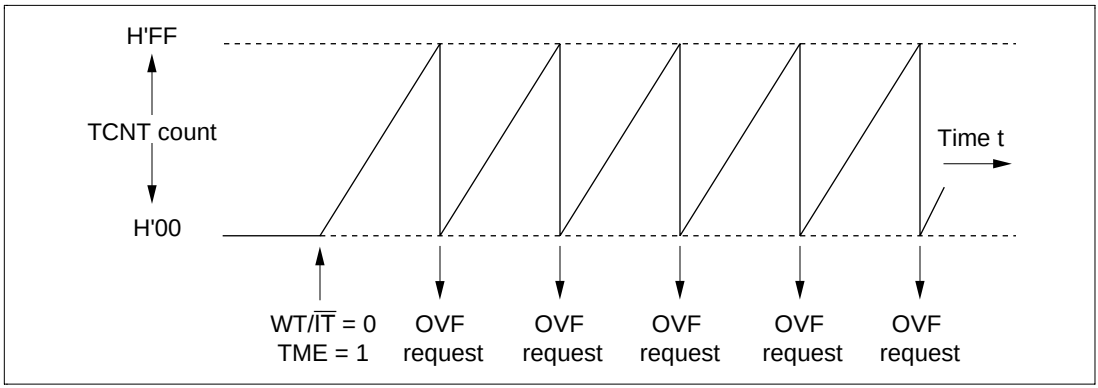


Figure 13.4 Operation in Interval Timer Mode

13.3.3 Setting the Overflow Flag

The OVF bit is set to 1 when the timer count overflows. Simultaneously, the WDT module requests an internal reset, NMI, or OVF interrupt. The timing is shown in figure 13.5.

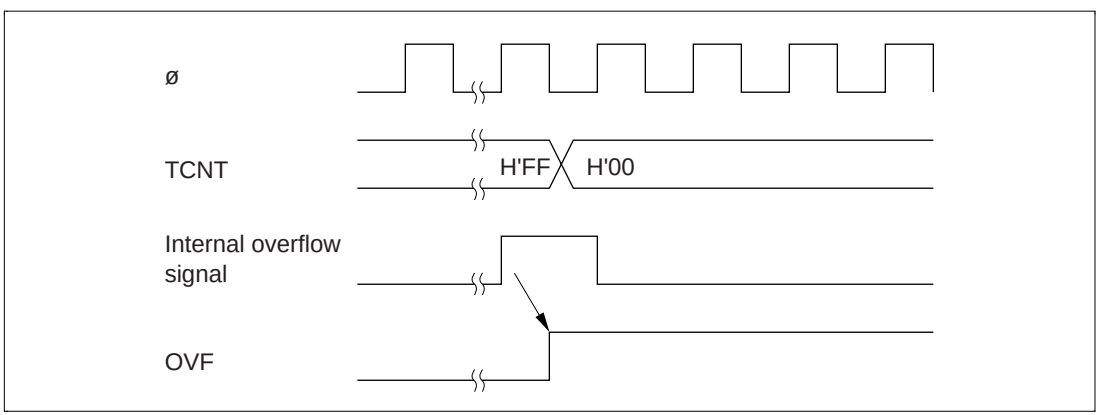


Figure 13.5 Setting the OVF Bit

13.4 Application Notes

13.4.1 Contention between TCNT Write and Increment

If a timer counter clock pulse is generated during the T_3 state of a write cycle to the timer counter, the write takes priority and the timer counter is not incremented. See figure 13.6.

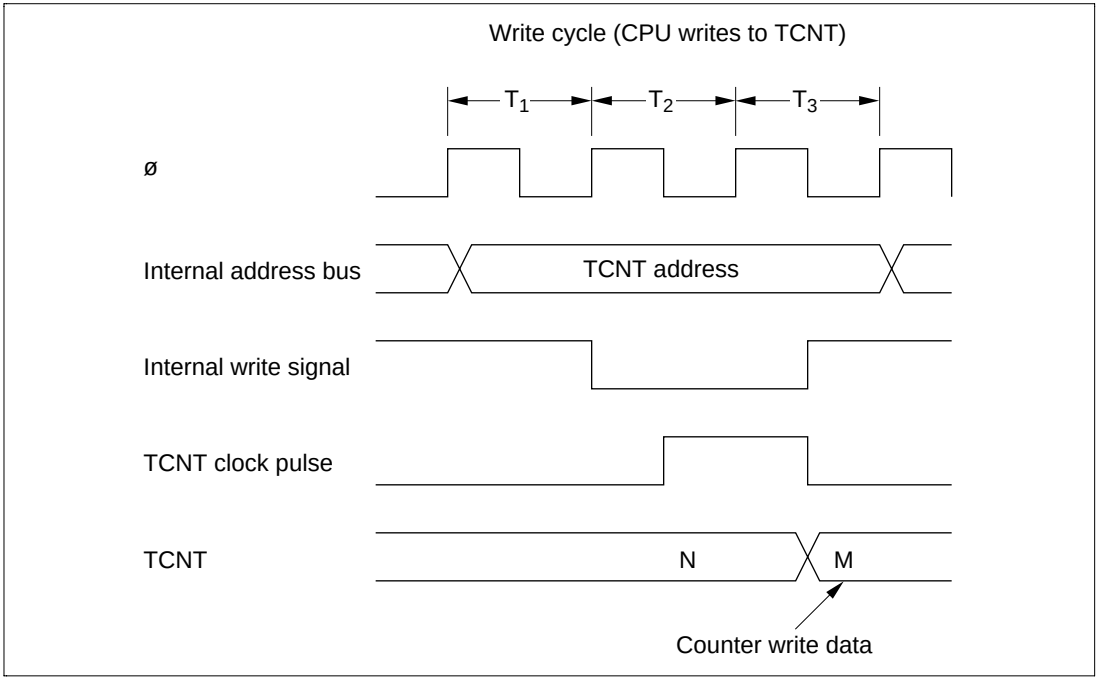


Figure 13.6 TCNT Write-Increment Contention

13.4.2 Changing the Clock Select Bits (CKS2 to CKS0)

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

13.4.3 Recovery from Software Standby Mode

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

Section 14 Serial Communication Interface

14.1 Overview

The H8/3318 includes two serial communication interface channels (SCI0 and SCI1) for transferring serial data to and from other chips. Either synchronous or asynchronous communication can be selected.

14.1.1 Features

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

- Asynchronous mode

  An H8/3318 microcontroller can communicate with a UART (Universal Asynchronous Receiver/Transmitter), ACIA (Asynchronous Communication Interface Adapter), or other chip that employs standard asynchronous serial communication. It also has a multiprocessor communication function for communication with other processors. Twelve data formats are available.

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

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

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

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

  Any specified bit rate can be generated.
- Internal or external clock source

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

- Four interrupts
  TDR-empty, TSR-empty, receive-end, and receive-error interrupts are requested independently.

14.1.2 Block Diagram

Figure 14.1 shows a block diagram of one serial communication interface channel.

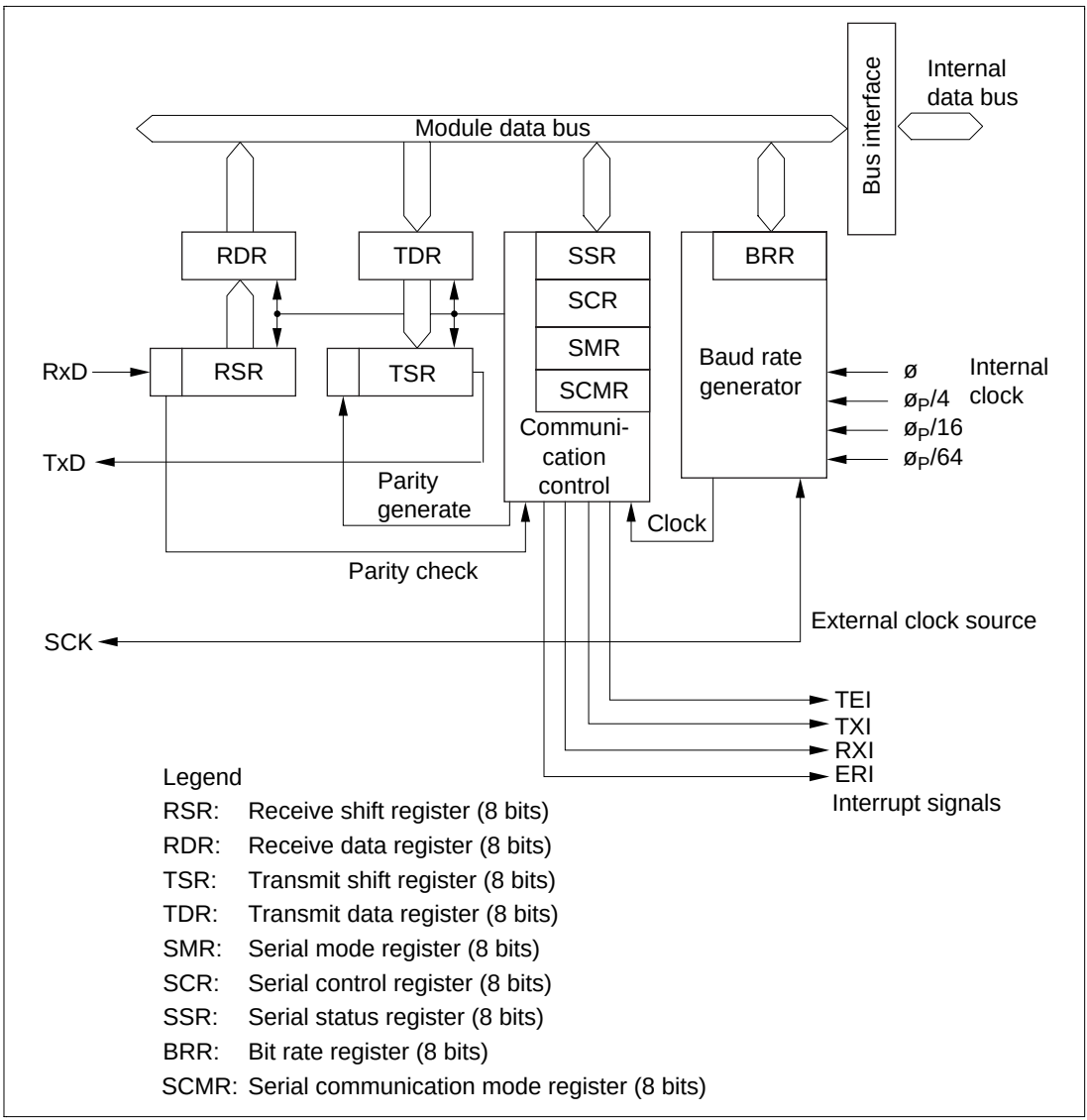


Figure 14.1 Block Diagram of Serial Communication Interface

14.1.3 Input and Output Pins

Table 14.1 lists the input and output pins used by the SCI module.

Table 14.1 SCI Input/Output Pins

Channel	Name	Abbr.*	I/O	Function
0	Serial clock	SCK_0	Input/output	Serial clock input and output
	Receive data	RxD_0	Input	Receive data input
	Transmit data	TxD_0	Output	Transmit data output
1	Serial clock	SCK_1	Input/output	Serial clock input and output
	Receive data	RxD_1	Input	Receive data input
	Transmit data	TxD_1	Output	Transmit data output

Note: * In this manual, the channel subscripts are normally omitted.

14.1.4 Register Configuration

Table 14.2 lists the SCI registers. These registers specify the operating mode (synchronous or asynchronous), data format and bit rate, and control the transmit and receive sections.

Table 14.2 SCI Registers

Channel	Name	Abbr.	R/W	Value	Address
0	Receive shift register	RSR	—	—	—
	Receive data register	RDR	R	H'00	H'FFDD
	Transmit shift register	TSR	—	—	—
	Transmit data register	TDR	R/W	H'FF	H'FFDB
	Serial mode register	SMR	R/W	H'00	H'FFD8
	Serial control register	SCR	R/W	H'00	H'FFDA
	Serial status register	SSR	R/(W)*	H'84	H'FFDC
	Bit rate register	BRR	R/W	H'FF	H'FFD9
	Serial communication mode register	SCMR	R/W	H'F2	H'FFDE
1	Receive shift register	RSR	—	—	—
	Receive data register	RDR	R	H'00	H'FF8D
	Transmit shift register	TSR	—	—	—
	Transmit data register	TDR	R/W	H'FF	H'FF8B
	Serial mode register	SMR	R/W	H'00	H'FF88
	Serial control register	SCR	R/W	H'00	H'FF8A
	Serial status register	SSR	R/(W)*	H'84	H'FF8C
	Bit rate register	BRR	R/W	H'FF	H'FF89
0 and 1	Serial/timer control register	STCR	R/W	H'1C	H'FFC3

Note: * Software can write a 0 to clear the flags in bits 7 to 3, but cannot write 1 in these bits.

14.2 Register Descriptions

14.2.1 Receive Shift Register (RSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

RSR is a shift register that converts incoming serial data to parallel data. When one data character has been received, it is transferred to the receive data register (RDR).

The CPU cannot read or write RSR directly.

14.2.2 Receive Data Register (RDR)

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

RDR stores received data. As each character is received, it is transferred from RSR to RDR, enabling RSR to receive the next character. This double-buffering allows the SCI to receive data continuously.

RDR is a read-only register. RDR is initialized to H'00 by a reset and in the standby modes.

14.2.3 Transmit Shift Register (TSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

TSR is a shift register that converts parallel data to serial transmit data. When transmission of one character is completed, the next character is moved from the transmit data register (TDR) to TSR and transmission of that character begins. If the TDRE bit is still set to 1, however, nothing is transferred to TSR.

The CPU cannot read or write TSR directly.

14.2.4 Transmit Data Register (TDR)

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

TDR is an 8-bit readable/writable register that holds the next data to be transmitted. When TSR becomes empty, the data written in TDR is transferred to TSR. Continuous data transmission is possible by writing the next data in TDR while the current data is being transmitted from TSR.

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

14.2.5 Serial Mode Register (SMR)

Bit	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SMR is an 8-bit readable/writable register that controls the communication format and selects the clock source of the on-chip baud rate generator. It is initialized to H'00 by a reset and in the standby modes. For further information on the SMR settings and communication formats, see tables 14.5 and 14.7 in section 14.3, Operation.

Bit 7—Communication Mode (C/A): This bit selects asynchronous or synchronous communication mode.

Bit 7 C/$\overline{A}$	Description	
0	Asynchronous communication	(Initial value)
1	Synchronous communication	

Bit 6—Character Length (CHR): This bit selects the character length in asynchronous mode. It is ignored in synchronous mode.

Bit 6 CHR	Description	
0	8 bits per character	(Initial value)
1	7 bits per character (Bits 0 to 6 of TDR and RDR are used for transmitting and receiving, respectively)	

Bit 5—Parity Enable (PE): This bit selects whether to add a parity bit in asynchronous mode. It is ignored in synchronous mode, and when a multiprocessor format is used.

Bit 5 PE	Description	
0	Transmit: No parity bit is added.	(Initial value)
	Receive: Parity is not checked.	
1	Transmit: A parity bit is added.	
	Receive: Parity is checked.	

Bit 4—Parity Mode (O/$\overline{E}$): In asynchronous mode, when parity is enabled (PE = 1), this bit selects even or odd parity.

Even parity means that a parity bit is added to the data bits for each character to make the total number of 1's even. Odd parity means that the total number of 1's is made odd.

This bit is ignored when PE = 0, or when a multiprocessor format is used. It is also ignored in synchronous mode.

Bit 4 O/$\overline{E}$	Description	
0	Even parity	(Initial value)
1	Odd parity	

Bit 3—Stop Bit Length (STOP): This bit selects the number of stop bits. It is ignored in synchronous mode.

Bit 3 STOP	Description	
0	One stop bit	(Initial value)
	Transmit: One stop bit is added.	
	Receive: One stop bit is checked to detect framing errors.	
1	Two stop bits	
	Transmit: Two stop bits are added.	
	Receive: The first stop bit is checked to detect framing errors. If the second stop bit is a space (0), it is regarded as the next start bit.	

Bit 2—Multiprocessor Mode (MP): This bit selects the multiprocessor format. When multiprocessor format is selected, the parity settings of the parity enable bit (PE) and parity mode bit (O/$\overline{E}$) are ignored. The MP bit setting is valid only in asynchronous communication, and is ignored in synchronous communication.

Bit 2 MP	Description	
0	Multiprocessor communication function is disabled.	(Initial value)
1	Multiprocessor communication function is enabled.	

Bits 1 and 0—Clock Select 1 and 0 (CKS1 and CKS0): These bits select the clock source of the on-chip baud rate generator.

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	ø clock	(Initial value)
	1	$ø_P$/4 clock	
1	0	$ø_P$/16 clock	
	1	$ø_P$/64 clock	

14.2.6 Serial Control Register (SCR)

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR is an 8-bit readable/writable register that enables or disables various SCI functions.
It is initialized to H'00 by a reset and in the standby modes.

Bit 7—Transmit Interrupt Enable (TIE): This bit enables or disables the TDR-empty interrupt (TXI) requested when the transmit data register empty (TDRE) bit in the serial status register (SSR) is set to 1.

Bit 7 TIE	Description	
0	The TDR-empty interrupt request (TXI) is disabled.	(Initial value)
1	The TDR-empty interrupt request (TXI) is enabled.	

Bit 6—Receive Interrupt Enable (RIE): This bit enables or disables the receive-end interrupt (RXI) requested when the receive data register full (RDRF) bit in the serial status register (SSR) is set to 1, and the receive error interrupt (ERI) requested when the overrun error (ORER), framing error (FER), or parity error (PER) bit in the serial status register (SSR) is set to 1.

Bit 6 RIE	Description
0	The receive-end interrupt (RXI) and receive-error (ERI) requests are disabled. (Initial value)
1	The receive-end interrupt (RXI) and receive-error (ERI) requests are enabled.

Bit 5—Transmit Enable (TE): This bit enables or disables the transmit function. When the transmit function is enabled, the TxD pin is automatically used for output. When the transmit function is disabled, the TxD pin can be used as a general-purpose I/O port.

Bit 5 TE	Description
0	The transmit function is disabled. (Initial value) The TxD pin can be used for general-purpose I/O.
1	The transmit function is enabled. The TxD pin is used for output.

Bit 4—Receive Enable (RE): This bit enables or disables the receive function. When the receive function is enabled, the RxD pin is automatically used for input. When the receive function is disabled, the RxD pin is available as a general-purpose I/O port.

Bit 4 RE	Description
0	The receive function is disabled. (Initial value) The RxD pin can be used for general-purpose I/O.
1	The receive function is enabled. The RxD pin is used for input.

Bit 3—Multiprocessor Interrupt Enable (MPIE): When serial data is received in a multiprocessor format, this bit enables or disables the receive-end interrupt (RXI) and receive-error interrupt (ERI) until data with the multiprocessor bit set to 1 is received. It also enables or disables the transfer of received data from RSR to RDR, and enables or disables setting of the RDRF, FER, PER, and ORER bits in the serial status register (SSR).

The MPIE bit is ignored when the MP bit is cleared to 0, and in synchronous mode.

Clearing the MPIE bit to 0 disables the multiprocessor receive interrupt function. In this condition data is received regardless of the value of the multiprocessor bit in the receive data.

Setting the MPIE bit to 1 enables the multiprocessor receive interrupt function. In this condition, if the multiprocessor bit in the receive data is 0, the receive-end interrupt (RXI) and receive-error interrupt (ERI) are disabled, the receive data is not transferred from RSR to RDR, and the RDRF, FER, PER, and ORER bits in the serial status register (SSR) are not set. If the multiprocessor bit is 1, however, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0, the receive data is transferred from RSR to RDR, the FER, PER, and ORER bits can be set, and the receive-end and receive-error interrupts are enabled.

Bit 3 MPIE	Description	
0	The multiprocessor receive interrupt function is disabled. (Normal receive operation)	(Initial value)
1	The multiprocessor receive interrupt function is enabled. During the interval before data with the multiprocessor bit set to 1 is received, the receive interrupt request (RXI) and receive-error interrupt request (ERI) are disabled, the RDRF, FER, PER, and ORER bits are not set in the serial status register (SSR), and no data is transferred from the RSR to the RDR. The MPIE bit is cleared at the following times: 1. When 0 is written in MPIE 2. When data with the multiprocessor bit set to 1 is received	

Bit 2—Transmit-End Interrupt Enable (TEIE): This bit enables or disables the TSR-empty interrupt (TEI) requested when the transmit-end bit (TEND) in the serial status register (SSR) is set to 1.

Bit 2 TEIE	Description	
0	The TSR-empty interrupt request (TEI) is disabled.	(Initial value)
1	The TSR-empty interrupt request (TEI) is enabled.	

Bit 1—Clock Enable 1 (CKE1): This bit selects the internal or external clock source for the baud rate generator. When the external clock source is selected, the SCK pin is automatically used for input of the external clock signal.

Bit 1 CKE1	Description	
0	Internal clock source When C/$\overline{A}$ = 1, the serial clock signal is output at the SCK pin. When C/$\overline{A}$ = 0, output depends on the CKE0 bit.	(Initial value)
1	External clock source The SCK pin is used for input.	

Bit 0—Clock Enable 0 (CKE0): When an internal clock source is used in asynchronous mode, this bit enables or disables serial clock output at the SCK pin.

This bit is ignored when the external clock is selected, or when synchronous mode is selected.

For further information on the communication format and clock source selection, see table 14.6 in section 14.3, Operation.

Bit 0 CKE0	Description
0	The SCK pin is not used by the SCI (and is available as a general-purpose I/O port). (Initial value)
1	The SCK pin is used for serial clock output.

14.2.7 Serial Status Register (SSR)

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

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

SSR is an 8-bit register that indicates transmit and receive status. It is initialized to H'84 by a reset and in the standby modes.

Bit 7—Transmit Data Register Empty (TDRE): This bit indicates when transmit data can safely be written in TDR.

Bit 7 TDRE	Description
0	To clear TDRE, the CPU must read TDRE after it has been set to 1, then write a 0 in this bit. Otherwise, when a data is written to TDR in the DTU bus cycle.
1	This bit is set to 1 at the following times: (Initial value) 1. When TDR contents are transferred to TSR 2. When the TE bit in SCR is cleared to 0

Bit 6—Receive Data Register Full (RDRF): This bit indicates when one character has been received and transferred to RDR.

Bit 6 RDRF	Description
0	To clear RDRF, the CPU must read RDRF after it has been set to 1, then write a 0 in this bit. (Initial value) Otherwise, when RDR is read in the DTU bus cycle.
1	This bit is set to 1 when one character is received without error and transferred from RSR to RDR.

Bit 5—Overrun Error (ORER): This bit indicates an overrun error during reception.

Bit 5 ORER	Description
0	To clear ORER, the CPU must read ORER after it has been set to 1, then write a 0 in this bit. (Initial value)
1	This bit is set to 1 if reception of the next character ends while the receive data register is still full (RDRF = 1).

Bit 4—Framing Error (FER): This bit indicates a framing error during data reception in asynchronous mode. It has no meaning in synchronous mode.

Bit 4 FER	Description
0	To clear FER, the CPU must read FER after it has been set to 1, then write a 0 in this bit. (Initial value)
1	This bit is set to 1 if a framing error occurs (stop bit = 0).

Bit 3—Parity Error (PER): This bit indicates a parity error during data reception in the asynchronous mode, when a communication format with parity bits is used.

This bit has no meaning in the synchronous mode, or when a communication format without parity bits is used.

Bit 3 PER	Description
0	To clear PER, the CPU must read PER after it has been set to 1, then write a 0 in this bit. (Initial value)
1	This bit is set to 1 when a parity error occurs (the parity of the received data does not match the parity selected by the O/$\overline{E}$ bit in SMR).

Bit 2—Transmit End (TEND): This bit indicates that the serial communication interface has stopped transmitting because there was no valid data in TDR when the last bit of the current character was transmitted. The TEND bit is also set to 1 when the TE bit in the serial control register (SCR) is cleared to 0.

The TEND bit is a read-only bit and cannot be modified directly. To use the TEI interrupt, first start transmitting data, which clears TEND to 0, then set TEIE to 1.

Bit 2 TEND	Description
0	To clear TEND, the CPU must read TDRE after TDRE has been set to 1, then write a 0 in TDRE. (Initial value)
1	This bit is set to 1 when: 1. TE = 0 2. TDRE = 1 at the end of transmission of a character

Bit 1—Multiprocessor Bit (MPB): Stores the value of the multiprocessor bit in data received in a multiprocessor format in asynchronous communication mode. This bit retains its previous value in synchronous mode, when a multiprocessor format is not used, or when the RE bit is cleared to 0 even if a multiprocessor format is used.

MPB can be read but not written.

Bit 1 MPB	Description
0	Multiprocessor bit = 0 in receive data. (Initial value)
1	Multiprocessor bit = 1 in receive data.

Bit 0—Multiprocessor Bit Transfer (MPBT): Stores the value of the multiprocessor bit inserted in transmit data when a multiprocessor format is used in asynchronous communication mode. The MPBT bit is double-buffered in the same way as TSR and TDR. The MPBT bit has no effect in synchronous mode, or when a multiprocessor format is not used.

Bit 0 MPBT	Description
0	Multiprocessor bit = 0 in transmit data. (Initial value)
1	Multiprocessor bit = 1 in transmit data.

14.2.8 Bit Rate Register (BRR)

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

BRR is an 8-bit register that, together with the CKS1 and CKS0 bits in SMR, determines the bit rate output by the baud rate generator.

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

Tables 14.3 to 14.6 show examples of BRR settings.

Table 14.3 Examples of BRR Settings in Asynchronous Mode (When $ø_P$ = ø)

	ø (MHz)											
	2			2.4576			3			3.6864		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	+0.03	1	174	–0.26	1	212	+0.03	2	64	+0.70
150	1	103	+0.16	1	127	0	1	155	+0.16	1	191	0
300	0	207	+0.16	0	255	0	1	77	+0.16	1	95	0
600	0	103	+0.16	0	127	0	0	155	+0.16	0	191	0
1200	0	51	+0.16	0	63	0	0	77	+0.16	0	95	0
2400	0	25	+0.16	0	31	0	0	38	+0.16	0	47	0
4800	0	12	+0.16	0	15	0	0	19	–2.34	0	23	0
9600	—	—	—	0	7	0	0	9	–2.34	0	11	0
19200	—	—	—	0	3	0	0	4	–2.34	0	5	0
31250	0	1	0	—	—	—	0	2	0	—	—	—
38400	—	—	—	0	1	0	—	—	—	0	2	0

Note: If possible, the error should be within 1%.

Table 14.3 Examples of BRR Settings in Asynchronous Mode (When $ø_P$ = ø) (cont)

	ø (MHz)											
	4			4.9152			5			6		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	70	+0.03	2	86	+0.31	2	88	–0.25	2	106	–0.44
150	1	207	+0.16	1	255	0	2	64	+0.16	2	77	+0.16
300	1	103	+0.16	1	127	0	1	129	+0.16	1	155	+0.16
600	0	207	+0.16	0	255	0	1	64	+0.16	1	77	+0.16
1200	0	103	+0.16	0	127	0	0	129	+0.16	0	155	+0.16
2400	0	51	+0.16	0	63	0	0	64	+0.16	0	77	+0.16
4800	0	25	+0.16	0	31	0	0	32	–1.36	0	38	+0.16
9600	0	12	+0.16	0	15	0	0	15	+1.73	0	19	–2.34
19200	—	—	—	0	7	0	0	7	+1.73	0	9	–2.34
31250	0	3	0	0	4	–1.70	0	4	0	0	5	0
38400	—	—	—	0	3	0	0	3	+1.73	0	4	–2.34

	ø (MHz)											
	6.144			7.3728			8			9.8304		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	108	+0.08	2	130	–0.07	2	141	+0.03	2	174	–0.26
150	2	79	0	2	95	0	2	103	+0.16	2	127	0
300	1	159	0	1	191	0	1	207	+0.16	1	255	0
600	1	79	0	1	95	0	1	103	+0.16	1	127	0
1200	0	159	0	0	191	0	0	207	+0.16	0	255	0
2400	0	79	0	0	95	0	0	103	+0.16	0	127	0
4800	0	39	0	0	47	0	0	51	+0.16	0	63	0
9600	0	19	0	0	23	0	0	25	+0.16	0	31	0
19200	0	4	0	0	11	0	0	12	+0.16	0	15	0
31250	0	5	+2.40	—	—	—	0	7	0	0	9	–1.70
38400	0	4	0	0	5	0	—	—	—	0	7	0

Note: If possible, the error should be within 1%.

Table 14.3 Examples of BRR Settings in Asynchronous Mode (When $ø_P$ = ø) (cont)

	ø (MHz)											
	10			12			12.288			14.7456		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	177	–0.25	2	212	+0.03	2	217	+0.08	3	64	+0.70
150	2	129	+0.16	2	155	+0.16	2	159	0	2	191	0
300	2	64	+0.16	2	77	+0.16	2	79	0	2	95	0
600	1	129	+0.16	1	155	+0.16	1	159	0	1	191	0
1200	1	64	+0.16	1	77	+0.16	1	79	0	1	95	0
2400	0	129	+0.16	0	155	+0.16	0	159	0	0	191	0
4800	0	64	+0.16	0	77	+0.16	0	79	0	0	95	0
9600	0	32	–1.36	0	38	+0.16	0	39	0	0	47	0
19200	0	15	+1.73	0	19	–2.34	0	19	0	0	23	0
31250	0	9	0	0	11	0	0	11	+2.4	0	14	–1.7
38400	0	7	+1.73	0	9	–2.34	0	9	0	0	11	0

	ø (MHz)		
	16		
Bit Rate (bits/s)	n	N	Error (%)
110	3	70	+0.03
150	2	207	+0.16
300	2	103	+0.16
600	1	207	+0.16
1200	1	103	+0.16
2400	0	207	+0.16
4800	0	103	+0.16
9600	0	51	+0.16
19200	0	25	+0.16
31250	0	15	0
38400	0	12	+0.16

Note: If possible, the error should be within 1%.

Table 14.4 Examples of BRR Settings in Asynchronous Mode (When $ø_P$ = ø/2)

	ø (MHz)											
	2			2.4576			3			3.6864		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	70	0.03	1	86	0.31	1	106	–0.44	1	130	–0.07
150	1	51	0.16	1	63	0	1	77	0.16	1	95	0
300	0	103	0.16	0	255	0	1	38	0.16	1	47	0
600	0	51	0.16	0	127	0	0	77	0.16	0	95	0
1200	0	25	0.16	0	63	0	0	38	0.16	0	47	0
2400	0	12	0.16	0	31	0	0	19	–2.34	0	23	0
4800	—	—	—	0	15	0	0	9	–2.34	0	11	0
9600	—	—	—	0	7	0	0	4	–2.34	0	5	0
19200	—	—	—	0	3	0	—	—	—	0	2	0
31250	0	0	0	—	—	—	—	—	—	—	—	—
38400	—	—	—	0	1	0	—	—	—	—	—	—

	ø (MHz)											
	4			4.9152			5			6		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	141	0.03	1	174	–0.26	1	177	–0.25	1	212	0.03
150	1	103	0.16	1	127	0	1	129	0.16	1	155	0.16
300	1	51	0.16	1	63	0	1	64	0.16	1	77	0.16
600	0	103	0.16	0	127	0	1	32	1.36	1	38	0.16
1200	0	51	0.16	0	63	0	0	64	0.16	0	77	0.16
2400	0	25	0.16	0	31	0	0	32	–1.36	0	38	0.16
4800	0	12	0.16	0	15	0	0	15	1.73	0	19	–2.34
9600	0	—	—	0	7	0	0	7	1.73	0	9	–2.34
19200	—	—	—	0	3	0	0	3	1.73	0	4	–2.34
31250	0	1	0	0	—	—	0	—	—	0	2	0
38400	—	—	—	0	1	0	0	1	1.73	—	—	—

Table 14.4 Examples of BRR Settings in Asynchronous Mode (When $ø_P$ = ø/2) (cont)

	ø (MHz)											
	6.144			7.3728			8			9.8304		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	217	0.08	2	64	0.70	2	70	0.03	2	86	0.31
150	1	159	0	1	191	0	1	207	0.16	1	255	0
300	1	79	0	1	95	0	1	103	0.16	1	127	0
600	1	39	0	1	47	0	1	51	0.16	1	63	0
1200	0	79	0	0	95	0	0	103	0.16	0	127	0
2400	0	39	0	0	47	0	0	51	0.16	0	63	0
4800	0	19	0	0	23	0	0	25	0.16	0	31	0
9600	0	9	0	0	11	0	0	12	0.16	0	31	0
19200	0	4	0	0	5	0	0	—	—	0	15	0
31250	0	2	2.40	—	—	—	0	3	0	0	9	–1.70
38400	—	—	—	0	2	0	—	—	—	0	3	0

	ø (MHz)											
	10			12			12.288			14.7456		
Bit Rate (bits/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	88	–0.25	2	106	–0.44	2	108	0.08	2	130	–0.07
150	2	64	0.16	2	77	0.16	2	79	0	2	95	0
300	1	129	0.16	1	155	0.16	1	159	0	1	191	0
600	1	64	0.16	1	77	0.16	1	79	0	1	95	0
1200	0	129	0.16	0	155	0.16	0	159	0	0	191	0
2400	0	64	0.16	0	77	0.16	0	79	0	0	95	0
4800	0	32	–1.36	0	38	0.16	0	39	0	0	47	0
9600	0	15	1.73	0	19	–2.34	0	19	0	0	23	0
19200	0	7	1.73	0	9	–2.34	0	9	0	0	11	0
31250	0	4	0	0	5	0	0	5	2.40	—	—	—
38400	0	3	1.73	0	4	–2.34	0	4	0	0	5	0

Table 14.4 Examples of BRR Settings in Asynchronous Mode (When $ø_P$ = ø/2) (cont)

	ø (MHz)		
	16		
Bit Rate (bits/s)	**n**	**N**	**Error (%)**
110	2	141	0.03
150	2	103	0.16
300	1	207	0.16
600	1	103	0.16
1200	0	207	0.16
2400	0	103	0.16
4800	0	51	0.16
9600	0	25	0.16
19200	0	12	0.16
31250	0	7	0
38400	—	—	—

Legend:

Blank: No setting is available

—: A setting is available, but error occurs.

Note: If possible, the error should be within 1%.

$$B = \frac{F}{64 \times 2^{2n-1} \times (N + 1)} \times 10^6 \quad \Rightarrow \quad N = \frac{F}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/second)
N: Baud rate generator BRR value ($0 \leq N \leq 255$)
F: $ø_P$ (MHz) when $n \neq 0$, or ø (MHz) when n = 0
n: Baud rate generator input clock (n = 0, 1, 2, 3)
The meaning of n is given below.

	SMR		WSCR	
n	**CKS1**	**CKS0**	**CKDBL**	**Clock**
0	0	0	0	ø
1	0	1	0	ø/4
2	1	0	0	ø/16
3	1	1	0	ø/64
0	0	0	1	ø
1	0	1	1	ø/8
2	1	0	1	ø/32
3	1	1	1	ø/128

The bit rate error can be calculated with the formula below.

$$\text{Error (\%)} = \left\{ \frac{F \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

Table 14.5 Examples of BRR Settings in Synchronous Mode (When $ø_P$ = ø)

Bit Rate (bits/s)	ø (MHz)											
	2		4		5		8		10		16	
	n	N	n	N	n	N	n	N	n	N	n	N
100	—	—	—	—	—	—	—	—	—	—	—	—
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	124	1	199	1	249	2	99
5 k	0	99	0	199	0	249	1	99	1	124	1	199
10 k	0	49	0	99	0	124	0	199	0	249	1	99
25 k	0	19	0	39	0	49	0	79	0	99	0	159
50 k	0	9	0	19	0	24	0	39	0	49	0	79
100 k	0	4	0	9	—	—	0	19	0	24	0	39
250 k	0	1	0	3	0	4	0	7	0	9	0	15
500 k	0	0*	0	1	—	—	0	3	0	4	0	7
1 M			0	0*	—	—	0	1	—	—	0	3
2.5 M									0	0*	—	—
4 M											0	0*

Legend

Blank: No setting is available

—: A setting is available, but error occurs.

*: Continuous transfer is not possible

Table 14.6 Examples of BRR Settings in Synchronous Mode (When $ø_P$ = ø/2)

Bit Rate (bits/s)	ø (MHz)											
	2		4		5		8		10		16	
	n	N	n	N	n	N	n	N	n	N	n	N
100	—	—	—	—	—	—	—	—	—	—	—	—
250	1	249	2	124	—	—	2	249	—	—	3	124
500	1	124	1	249	—	—	2	124	—	—	2	249
1 k	—	—	1	124	—	—	1	249	—	—	2	124
2.5 k	0	99	1	49	—	—	1	99	1	124	1	199
5 k	0	49	0	99	0	124	1	49	—	—	1	99
10 k	0	24	0	49	—	—	0	99	0	124	1	49
25 k	0	9	0	19	0	24	0	39	0	49	0	79
50 k	0	4	0	9	—	—	0	19	0	24	0	39
100 k	—	—	0	4	—	—	0	9	0	12	0	19
250 k	0	0*	0	1	—	—	0	3	0	4	0	7
500 k	—	—	—	0*	—	—	0	1	—	—	0	3
1 M			—	—			0	0	—	—	0	1
2.5 M									—	—	—	—
4 M									—	—		

Legend

Blank: No setting is available

—: A setting is available, but error occurs.

*: Continuous transfer is not possible

$$B = \frac{F}{8 \times 2^{2n-1} \times (N + 1)} \times 10^6 \quad \Rightarrow \quad N = \frac{F}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bits/second)

N: Baud rate generator BRR value ($0 \leq N \leq 255$)

F: $ø_P$ (MHz) when $n \neq 0$, or ø (MHz) when n = 0

n: Baud rate generator input clock (n = 0, 1, 2, 3)

The meaning of n is given below.

	SMR		WSCR	
n	**CKS1**	**CKS0**	**CKDBL**	**Clock**
0	0	0	0	ø
1	0	1	0	ø/4
2	1	0	0	ø/16
3	1	1	0	ø/64
0	0	0	1	ø
1	0	1	1	ø/8
2	1	0	1	ø/32
3	1	1	1	ø/128

14.2.9 Serial/Timer Control Register (STCR)

Bit	7	6	5	4	3	2	1	0
	RING	CMPF	CMPIE	LOAD	MARK	—	ICKS1	ICKS0
Initial value	0	0	0	1	1	1	0	0
Read/Write	R/W	R/(W)*	R/W	(W)	(W)	—	R/W	R/W

Note: * Software can write a 0 in bit 6 to clear the flag, but cannot write a 1 in this bit.

STCR is an 8-bit readable/writable register that controls the SCI operating mode, selects the TCNT clock source in the 8-bit timers, and controls DTU channel B. STCR is initialized to H'1C by a reset.

Bits 7 to 3—DTU Channel B Control: These bits control DTU channel B. For details, see section 5, Data Transfer Unit.

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

Bits 1 and 0—Internal Clock Source Select 1 and 0 (ICKS1, ICKS0): These bits select the clock input to the timer counters (TCNT) in the 8-bit timers. For details, see section 11.2.3, Timer Control Register.

14.2.10 Serial Communication 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 that selects the SCI0 interface function.

SCMR is initialized to H'F2 by a reset and in the standby modes.

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. Receive data is stored in RDR LSB-first.	(Initial value)
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. Inversion by the SINV bit applies only to data bits D_7 to D_0. To invert the parity bit, the O/$\overline{E}$ bit in SMR must be inverted.

Bit 2 SINV	Description	
0	TDR contents are transmitted as they are. Receive data is stored in RDR as it is.	(Initial value)
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 Mode Select (SMIF): This bit is reserved, and should not be written with 1.

Bit 0 SMIF	Description	
0	Normal SCI mode	(Initial value)
1	Reserved mode	

14.3 Operation

14.3.1 Overview

The SCI supports serial data transfer in two modes. In asynchronous mode each character is synchronized individually. In synchronous mode communication is synchronized with a clock signal.

The selection of asynchronous or synchronous mode and the communication format depend on SMR settings as indicated in table 14.7. The clock source depends on the settings of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR as indicated in table 14.8.

Asynchronous Mode

- Data length: 7 or 8 bits can be selected.
- A parity bit or multiprocessor bit can be added, and stop bit lengths of 1 or 2 bits can be selected. (These selections determine the communication format and character length.)
- Framing errors (FER), parity errors (PER), and overrun errors (ORER) can be detected in receive data, and the line-break condition can be detected.
- SCI clock source: An internal or external clock source can be selected.
  - Internal clock: The SCI is clocked by the on-chip baud rate generator and can output a clock signal at the bit-rate frequency.
  - External clock: The external clock frequency must be 16 times the bit rate. (The on-chip baud rate generator is not used.)

Synchronous Mode

- Communication format: The data length is 8 bits.
- Overrun errors (ORER) can be detected in receive data.
- SCI clock source: An internal or external clock source can be selected.
  - Internal clock: The SCI is clocked by the on-chip baud rate generator and outputs a serial clock signal to external devices.
  - External clock: The on-chip baud rate generator is not used. The SCI operates on the input serial clock.

Table 14.7 Communication Formats Used by SCI

SMR Settings						Communication Format			
Bit 7 C/$\overline{A}$	Bit 6 CHR	Bit 2 MP	Bit 5 PE	Bit 3 STOP	Mode	Data Length	Multipro-cessor Bit	Parity Bit	Stop Bit Length
0	0	0	0	0	Asynchronous mode	8 bits	None	None	1 bit
				1					2 bits
			1	0				Present	1 bit
				1					2 bits
	1		0	0		7 bits		None	1 bit
				1					2 bits
			1	0				Present	1 bit
				1					2 bits
	0	1	—	0	Asynchronous mode (multiprocessor format)	8 bits	Present	None	1 bit
				1					2 bits
	1			0		7 bits			1 bit
				1					2 bits
1	—	—	—	—	Synchronous mode	8 bits	None		None

Table 14.8 SCI Clock Source Selection

SMR	SCR			Serial Transmit/Receive Clock	
Bit 7 C/$\overline{A}$	Bit 1 CKE1	Bit 0 CKE0	Mode	Clock Source	SCK Pin Function
0	0	0	Async	Internal	Input/output port (not used by SCI)
		1			Serial clock output at bit rate
	1	0		External	Serial clock input at 16 × bit rate
		1			
1	0	0	Sync	Internal	Serial clock output
		1			
	1	0		External	Serial clock input
		1			

14.3.2 Asynchronous Mode

In asynchronous mode, each transmitted or received character is individually synchronized by framing it with a start bit and stop bit.

Full duplex data transfer is possible because the SCI has independent transmit and receive sections. Double buffering in both sections enables the SCI to be programmed for continuous data transfer.

Figure 14.2 shows the general format of one character sent or received in asynchronous mode. The communication channel is normally held in the mark state (high). Character transmission or reception starts with a transition to the space state (low).

The first bit transmitted or received is the start bit (low). It is followed by the data bits, in which the least significant bit (LSB) comes first. The data bits are followed by the parity or multiprocessor bit, if present, then the stop bit or bits (high) confirming the end of the frame.

In receiving, the SCI synchronizes on the falling edge of the start bit, and samples each bit at the center of the bit (at the 8th cycle of the internal serial clock, which runs at 16 times the bit rate).

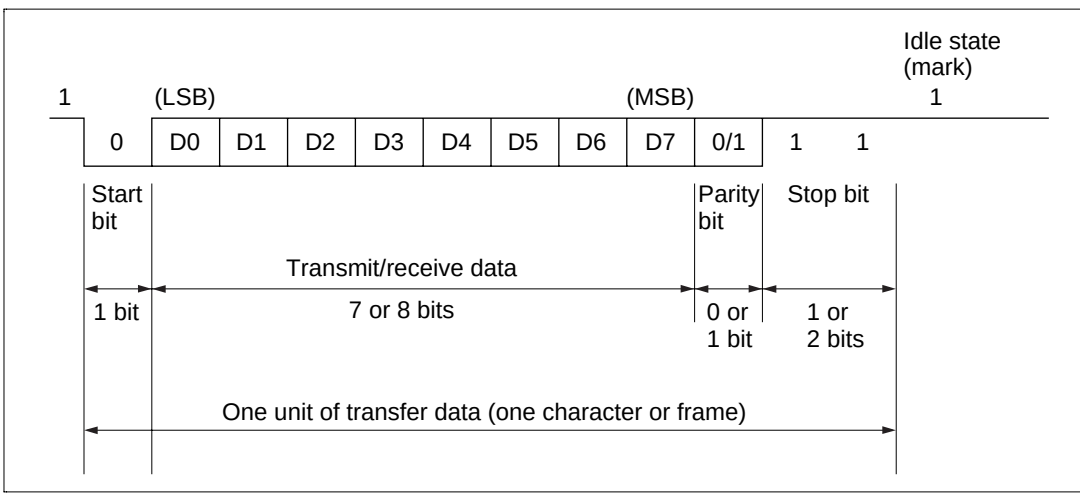


Figure 14.2 Data Format in Asynchronous Mode
(Example of 8-Bit Data with Parity Bit and Two Stop Bits)

Data Format: Table 14.9 lists the data formats that can be sent and received in asynchronous mode. Twelve formats can be selected by bits in the serial mode register.

Table 14.9 Data Formats in Asynchronous Mode

SMR Bits														
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	S	8-bit data								STOP		
0	0	0	1	S	8-bit data								STOP	STOP	
0	1	0	0	S	8-bit data								P	STOP	
0	1	0	1	S	8-bit data								P	STOP	STOP
1	0	0	0	S	7-bit data							STOP			
1	0	0	1	S	7-bit data							STOP	STOP		
1	1	0	0	S	7-bit data							P	STOP		
1	1	0	1	S	7-bit data							P	STOP	STOP	
0	—	1	0	S	8-bit data								MPB	STOP	
0	—	1	1	S	8-bit data								MPB	STOP	STOP
1	—	1	0	S	7-bit data							MPB	STOP		
1	—	1	1	S	7-bit data							MPB	STOP	STOP	

Notes: SMR: Serial mode register
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

Clock: In asynchronous mode it is possible to select either an internal clock created by the on-chip baud rate generator, or an external clock input at the SCK pin. The selection is made by the C/$\overline{A}$ bit in the serial mode register (SMR) and the CKE1 and CKE0 bits in the serial control register (SCR). Refer to table 14.8.

If an external clock is input at the SCK pin, its frequency should be 16 times the desired bit rate.

If the internal clock provided by the on-chip baud rate generator is selected and the SCK pin is used for clock output, the output clock frequency is equal to the bit rate, and the clock pulse rises at the center of the transmit data bits. Figure 14.3 shows the phase relationship between the output clock and transmit data.

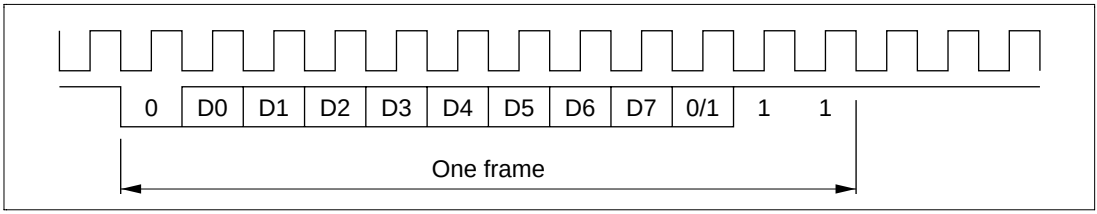


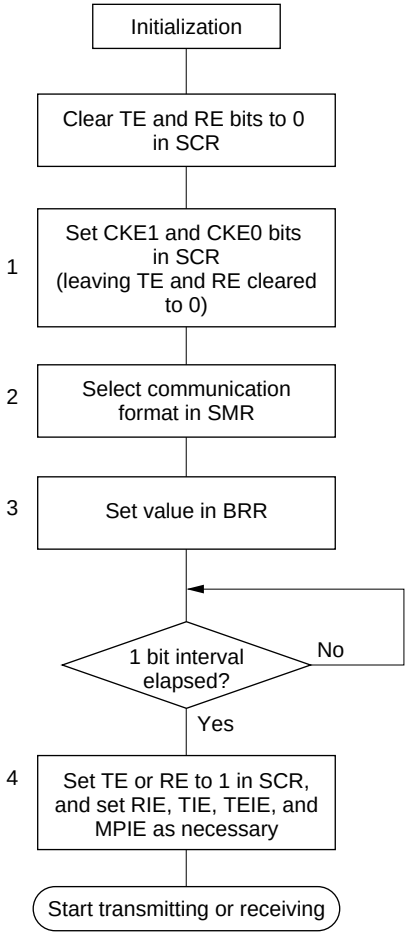
Figure 14.3 Phase Relationship between Clock Output and Transmit Data (Asynchronous Mode)

Transmitting and Receiving Data

• **SCI Initialization:** Before transmitting or receiving, software must clear the TE and RE bits to 0 in the serial control register (SCR), then initialize the SCI as follows.

Note: When changing the communication mode or format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets TDRE to 1 and initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDRF, PER, FER, and ORER flags and receive data register (RDR), which retain their previous contents.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCI operation becomes unreliable if the clock is stopped.



1. Select interrupts and the clock source in the serial control register (SCR). Leave TE and RE cleared to 0. If clock output is selected, in asynchronous mode, clock output starts immediately after the setting is made in SCR.

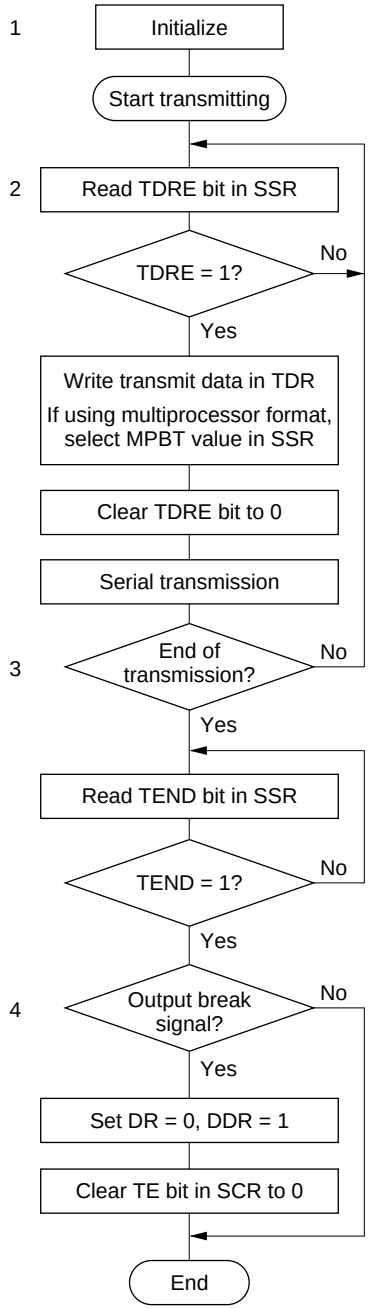
2. Select the communication format in the serial mode register (SMR).

3. Write the value corresponding to the bit rate in the bit rate register (BRR). This step is not necessary when an external clock is used.

4. Wait for at least the interval required to transmit or receive one bit, then set TE or RE in the serial control register (SCR). Setting TE or RE enables the SCI to use the TxD or RxD pin.
   Also set the RIE, TIE, TEIE, and MPIE bits as necessary to enable interrupts. The initial states are the mark transmit state, and the idle receive state (waiting for a start bit).

Figure 14.4 Sample Flowchart for SCI Initialization

- **Transmitting Serial Data:** Follow the procedure below for transmitting serial data.



1. SCI initialization: the transmit data output function of the TxD pin is selected automatically. Transmit become available 1 frame term after TE bit is set.

2. SCI status check and transmit data write: read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0. If a multiprocessor format is selected, after writing the transmit data write 0 or 1 in the multiprocessor bit transfer (MPBT) in SSR. Transition of the TDRE bit from 0 to 1 can be reported by an interrupt.

3. (a) To continue transmitting serial data: read the TDRE bit to check whether it is safe to write; if TDRE = 1, write data in TDR, then clear TDRE to 0.
   (b) To end serial transmission: end of transmission can be confirmed by checking transition of the TEND bit from 0 to 1. This can be reported by a TEI interrupt.

4. To output a break signal at the end of serial transmission: set the DDR bit to 1 and clear the DR bit to 0 (DDR and DR are I/O port registers), then clear TE to 0 in SCR.

Figure 14.5 Sample Flowchart for Transmitting Serial Data

In transmitting serial data, the SCI operates as follows.

1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).

2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the TIE bit (TDR-empty interrupt enable) is set to 1 in SCR, the SCI requests a TXI interrupt (TDR-empty interrupt) at this time.

   Serial transmit data is transmitted in the following order from the TxD pin:

   a. Start bit: One 0 bit is output.
   b. Transmit data: Seven or eight bits are output, LSB first.
   c. Parity bit or multiprocessor bit: One parity bit (even or odd parity) or one multiprocessor bit is output. Formats in which neither a parity bit nor a multiprocessor bit is output can also be selected.
   d. Stop bit: One or two 1 bits (stop bits) are output.
   e. Idle state: Output of 1 bits continues until the start bit of the next transmit data.

3. The SCI checks the TDRE bit when it outputs the stop bit. If TDRE is 0, after loading new data from TDR into TSR and transmitting the stop bit, the SCI begins serial transmission of the next frame. If TDRE is 1, after setting the TEND bit to 1 in SSR and transmitting the stop bit, the SCI continues 1-level output in the mark state, and if the TEIE bit (TSR-empty interrupt enable) in SCR is set to 1, the SCI generates a TEI interrupt request (TSR-empty interrupt).

Figure 14.6 shows an example of SCI transmit operation in asynchronous mode.

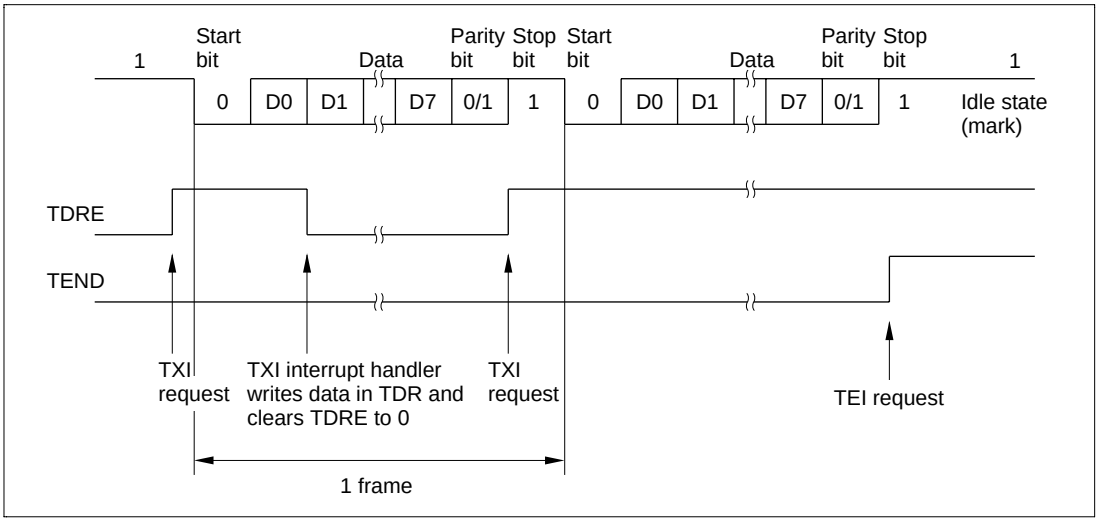
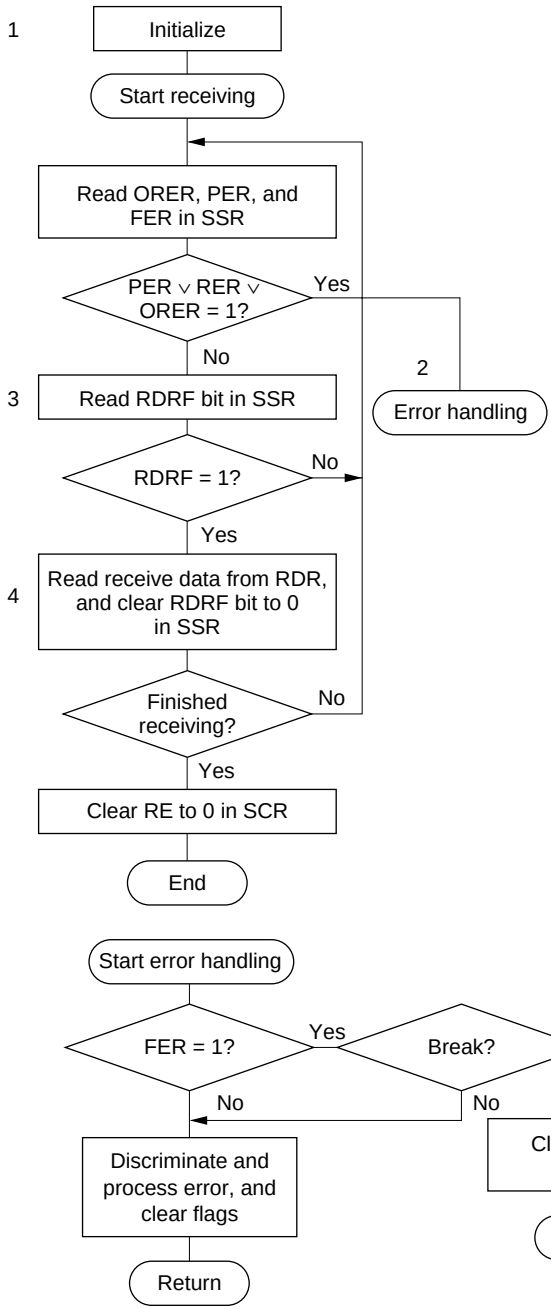


Figure 14.6 Example of SCI Transmit Operation (8-Bit Data with Parity and One Stop Bit)

- **Receiving Serial Data:** Follow the procedure below for receiving serial data.



1. SCI initialization: the receive data function of the RxD pin is selected automatically.
2. Receive error handling and break detection: if a receive error occurs, read the ORER, PER, and FER bits in SSR to identify the error. After executing the necessary error handling, clear ORER, PER, and FER all to 0. Transmitting and receiving cannot resume if ORER, PER, or FER remains set to 1. When a framing error occurs, the RxD pin can be read to detect the break state.
3. SCI status check and receive data read: read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. Transition of the RDRF bit from 0 to 1 can be reported by an RXI interrupt.
4. To continue receiving serial data: read RDR and clear RDRF to 0 before the stop bit of the current frame is received.

Figure 14.7 Sample Flowchart for Receiving Serial Data

In receiving, the SCI operates as follows.

1. The SCI monitors the receive data line and synchronizes internally when it detects a start bit.
2. Receive data is shifted into RSR in order from LSB to MSB.
3. The parity bit and stop bit are received.

   After receiving these bits, the SCI makes the following checks:

   a. Parity check: The number of 1s in the receive data must match the even or odd parity setting of the O/E bit in SMR.
   b. Stop bit check: The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.
   c. Status check: RDRF must be 0 so that receive data can be loaded from RSR into RDR.

   If these checks all pass, the SCI sets RDRF to 1 and stores the received data in RDR. If one of the checks fails (receive error), the SCI operates as indicated in table 14.10.

   Note: When a receive error flag is set, further receiving is disabled. The RDRF bit is not set to 1. Be sure to clear the error flags.

4. After setting RDRF to 1, if the RIE bit (receive-end interrupt enable) is set to 1 in SCR, the SCI requests an RXI (receive-end) interrupt. If one of the error flags (ORER, PER, or FER) is set to 1 and the RIE bit in SCR is also set to 1, the SCI requests an ERI (receive-error) interrupt.

Figure 14.8 shows an example of SCI receive operation in asynchronous mode.

Table 14.10 Receive Error Conditions and SCI Operation

Receive Error	Abbreviation	Condition	Data Transfer
Overrun error	ORER	Receiving of next data ends while RDRF is still set to 1 in SSR	Receive data not loaded from RSR into RDR
Framing error	FER	Stop bit is 0	Receive data loaded from RSR into RDR
Parity error	PER	Parity of receive data differs from even/odd parity setting in SMR	Receive data loaded from RSR into RDR

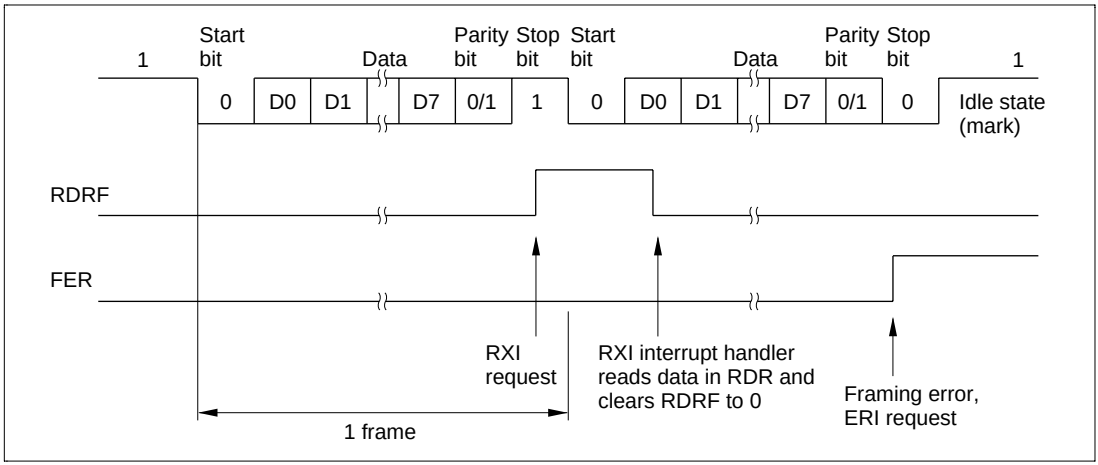


Figure 14.8 Example of SCI Receive Operation (8-Bit Data with Parity and One Stop Bit)

Multiprocessor Communication: The multiprocessor communication function enables several processors to share a single serial communication line. The processors communicate in asynchronous mode using a format with an additional multiprocessor bit (multiprocessor format).

In multiprocessor communication, each receiving processor is addressed by an ID.

A serial communication cycle consists of two cycles: an ID-sending cycle that identifies the receiving processor, and a data-sending cycle. The multiprocessor bit distinguishes ID-sending cycles from data-sending cycles.

The transmitting processor starts by sending the ID of the receiving processor with which it wants to communicate as data with the multiprocessor bit set to 1. Next the transmitting processor sends transmit data with the multiprocessor bit cleared to 0.

Receiving processors skip incoming data until they receive data with the multiprocessor bit set to 1.

After receiving data with the multiprocessor bit set to 1, the receiving processor with an ID matching the received data continues to receive further incoming data. Multiple processors can send and receive data in this way.

Four formats are available. Parity-bit settings are ignored when a multiprocessor format is selected. For details see table 14.9.

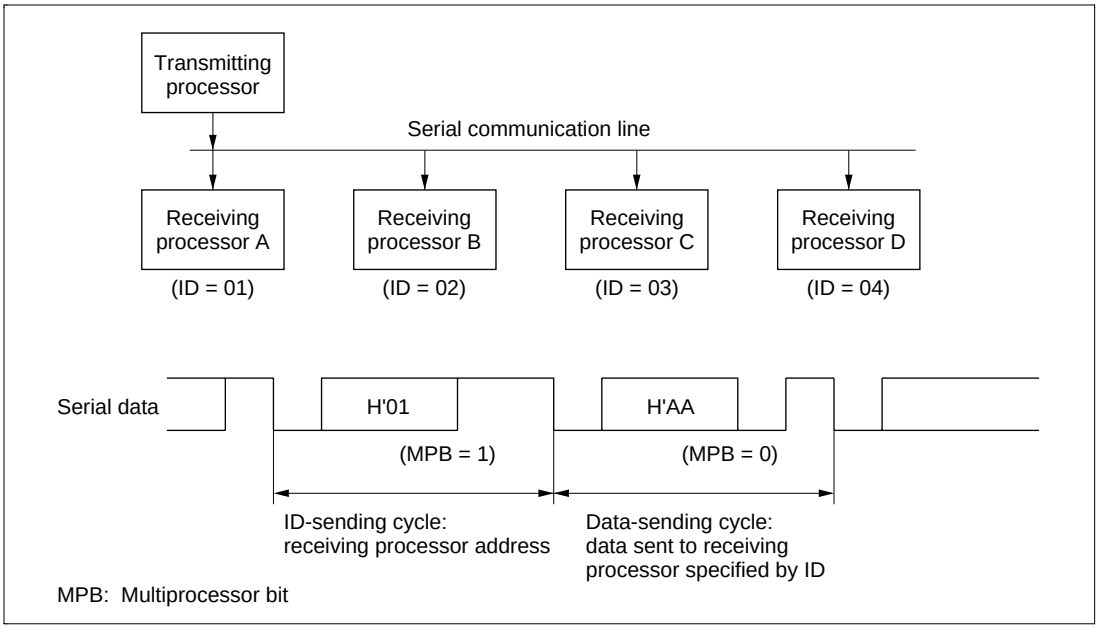


Figure 14.9 Example of Communication among Processors using Multiprocessor Format (Sending Data H'AA to Receiving Processor A)

- **Transmitting Multiprocessor Serial Data:** See figures 14.5 and 14.6.

• **Receiving Multiprocessor Serial Data:** Follow the procedure below for receiving multiprocessor serial data.

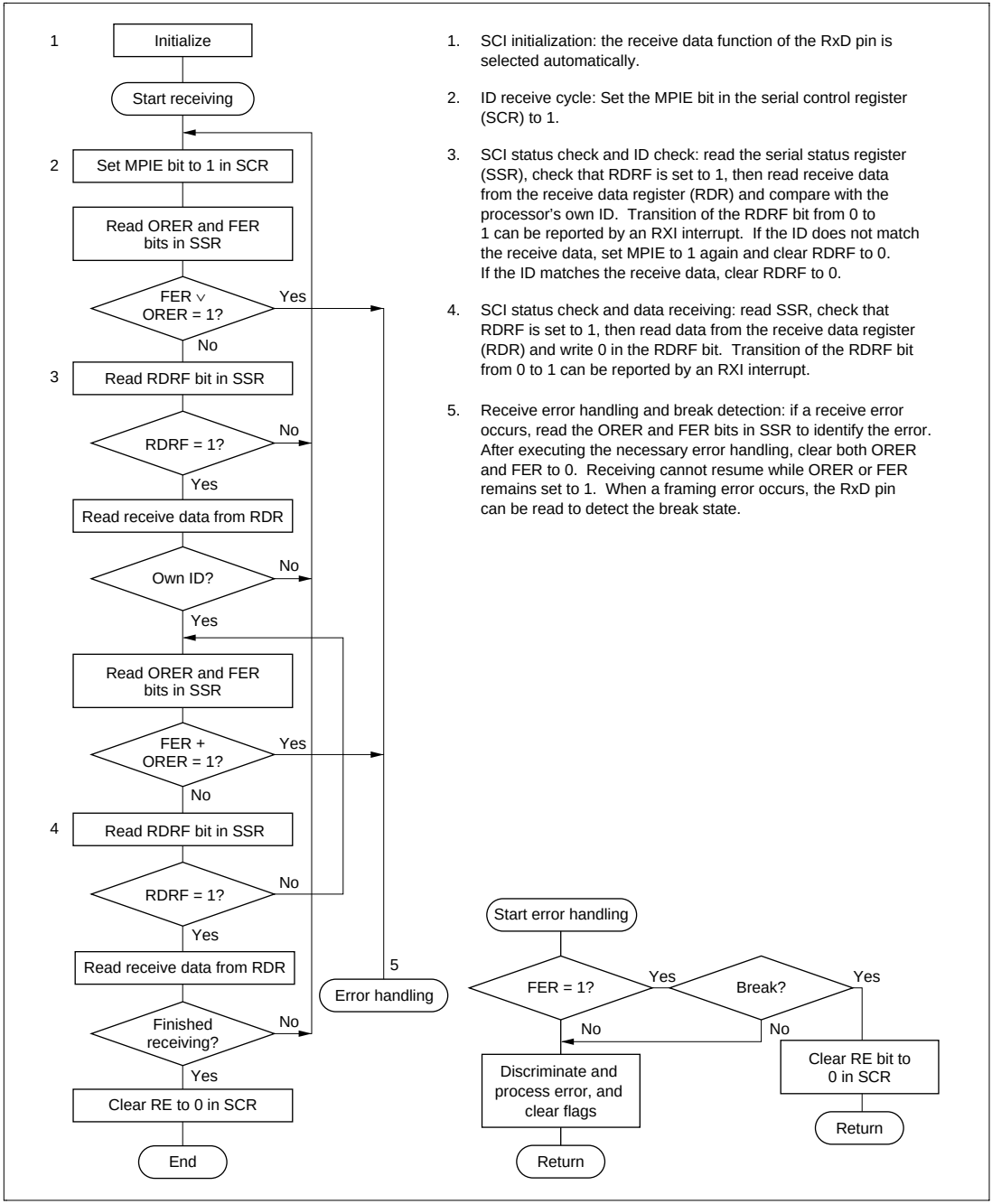


Figure 14.10 Sample Flowchart for Receiving Multiprocessor Serial Data

Figure 14.11 shows an example of an SCI receive operation using a multiprocessor format (8-bit data with multiprocessor bit and one stop bit).

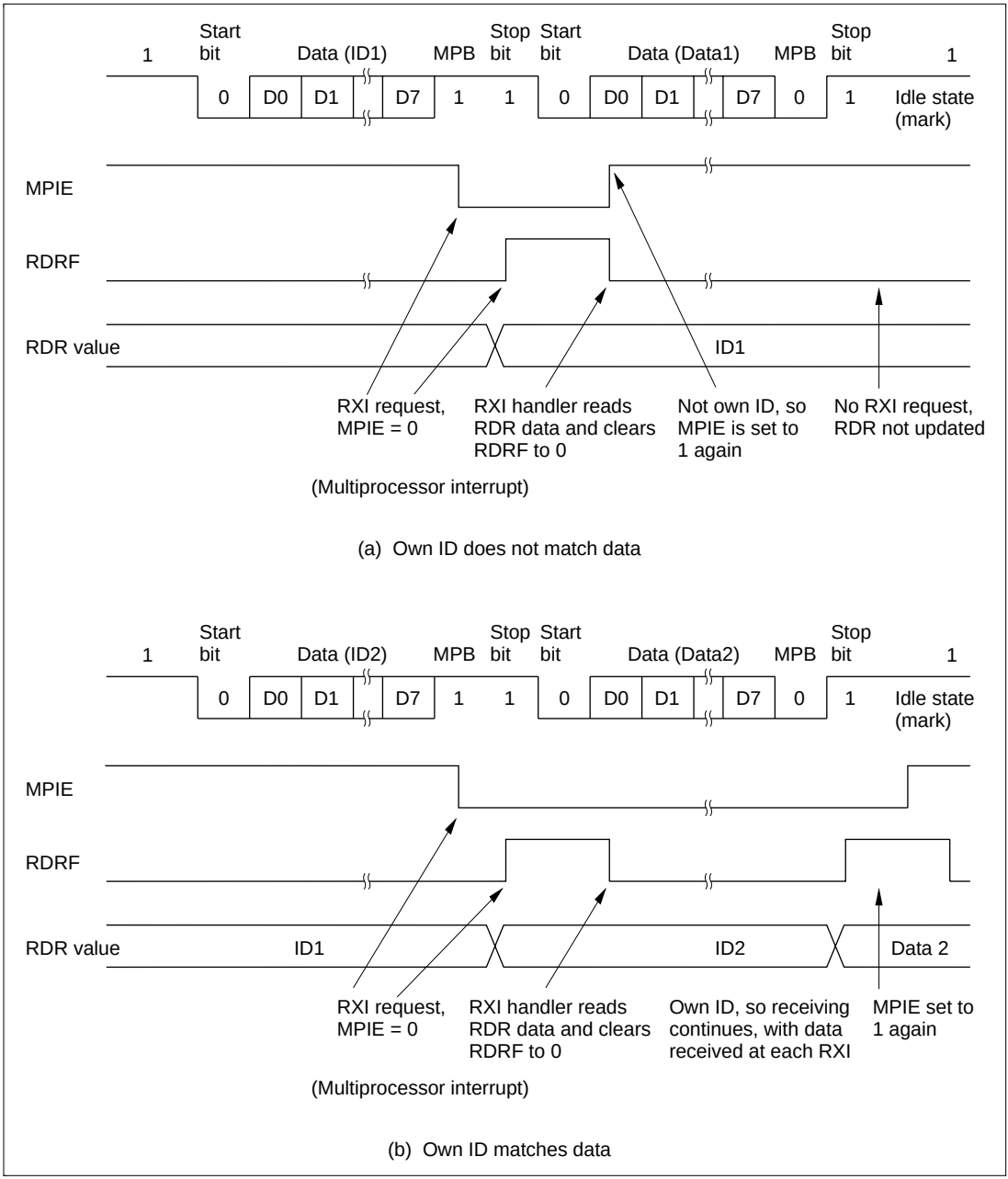


Figure 14.11 Example of SCI Receive Operation (Eight-Bit Data with Multiprocessor Bit and One Stop Bit)

14.3.3 Synchronous Mode

Overview: In synchronous mode, the SCI transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

The SCI transmitter and receiver share the same clock but are otherwise independent, so full duplex communication is possible. The transmitter and receiver are also double buffered, so continuous transmitting or receiving is possible by reading or writing data while transmitting or receiving is in progress.

Figure 14.12 shows the general format in synchronous serial communication.

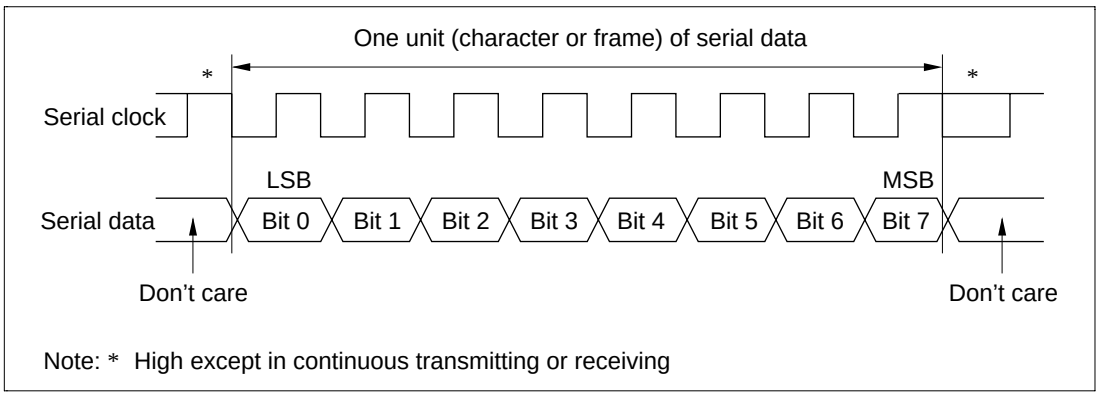


Figure 14.12 Data Format in Synchronous Communication

In synchronous serial communication, each data bit is sent on the communication line from one falling edge of the serial clock to the next. Data is received in synchronization with the rising edge of the serial clock.

In each character, the serial data bits are transmitted in order from LSB (first) to MSB (last). After output of the MSB, the communication line remains in the state of the MSB.

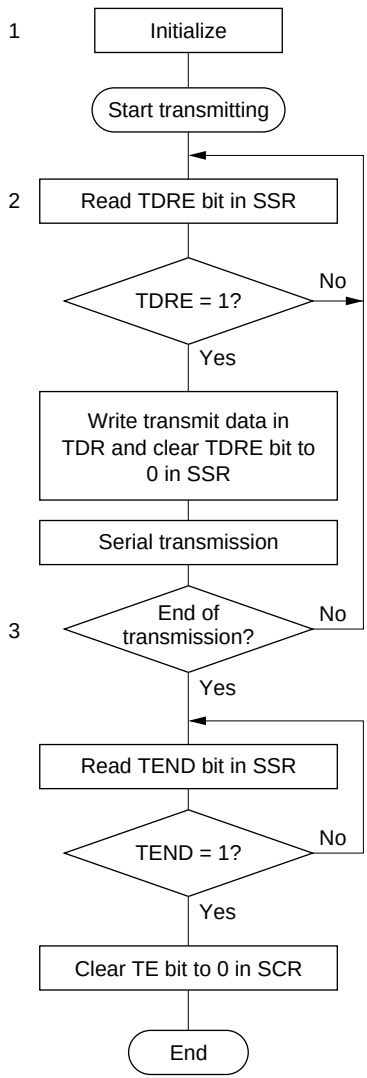
- **Communication Format:** The data length is fixed at eight bits. No parity bit or multiprocessor bit can be added.

- **Clock:** An internal clock generated by the on-chip baud rate generator or an external clock input from the SCK pin can be selected by clearing or setting the CKE1 bit in the serial control register (SCR). See table 14.8.

When the SCI operates on an internal clock, it outputs the clock signal at the SCK pin. Eight clock pulses are output per transmitted or received character. When the SCI is not transmitting or receiving, the clock signal remains at the high level.

• **SCI Initialization:** The SCI must be initialized in the same way as in asynchronous mode. See figure 14.4. When switching from asynchronous mode to synchronous mode, check that the ORER, FER, and PER bits are cleared to 0. Transmitting and receiving cannot begin if ORER, FER, or PER is set to 1.

• **Transmitting Serial Data:** Follow the procedure below for transmitting serial data.



1. SCI initialization: the transmit data output function of the TxD pin is selected automatically.

2. SCI status check and transmit data write: read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0. Transition of the TDRE bit from 0 to 1 can be reported by a TXI interrupt.

3. (a) To continue transmitting serial data: read the TDRE bit to check whether it is safe to write; if TDRE = 1, write data in TDR, then clear TDRE to 0.
   (b) To end serial transmission: end of transmission can be confirmed by checking transition of the TEND bit from 0 to 1. This can be reported by a TEI interrupt.

Figure 14.13 Sample Flowchart for Serial Transmitting

In transmitting serial data, the SCI operates as follows.

1. The SCI monitors the TDRE bit in SSR. When TDRE is cleared to 0 the SCI recognizes that the transmit data register (TDR) contains new data, and loads this data from TDR into the transmit shift register (TSR).

2. After loading the data from TDR into TSR, the SCI sets the TDRE bit to 1 and starts transmitting. If the TIE bit (TDR-empty interrupt enable) in SCR is set to 1, the SCI requests a TXI interrupt (TDR-empty interrupt) at this time.

   If clock output is selected the SCI outputs eight serial clock pulses, triggered by the clearing of the TDRE bit to 0. If an external clock source is selected, the SCI outputs data in synchronization with the input clock.

   Data is output from the TxD pin in order from LSB (bit 0) to MSB (bit 7).

3. The SCI checks the TDRE bit when it outputs the MSB (bit 7). If TDRE is 0, the SCI loads data from TDR into TSR, then begins serial transmission of the next frame. If TDRE is 1, the SCI sets the TEND bit in SSR to 1, transmits the MSB, then holds the output in the MSB state. If the TEIE bit (transmit-end interrupt enable) in SCR is set to 1, a TEI interrupt (TSR-empty interrupt) is requested at this time.

4. After the end of serial transmission, the SCK pin is held at the high level.

Figure 14.14 shows an example of SCI transmit operation.

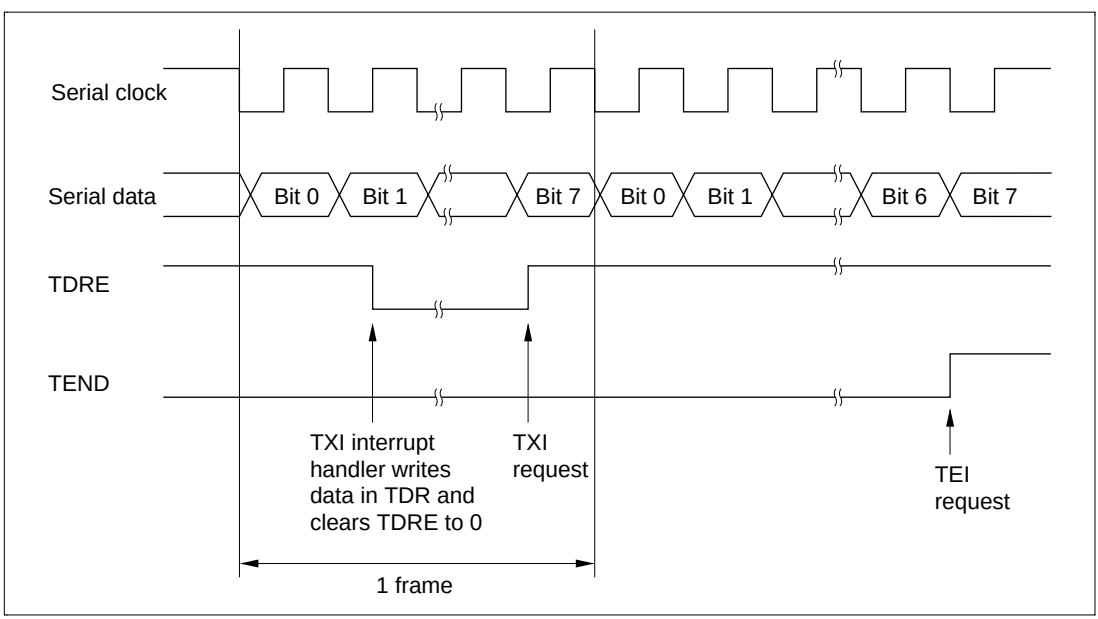
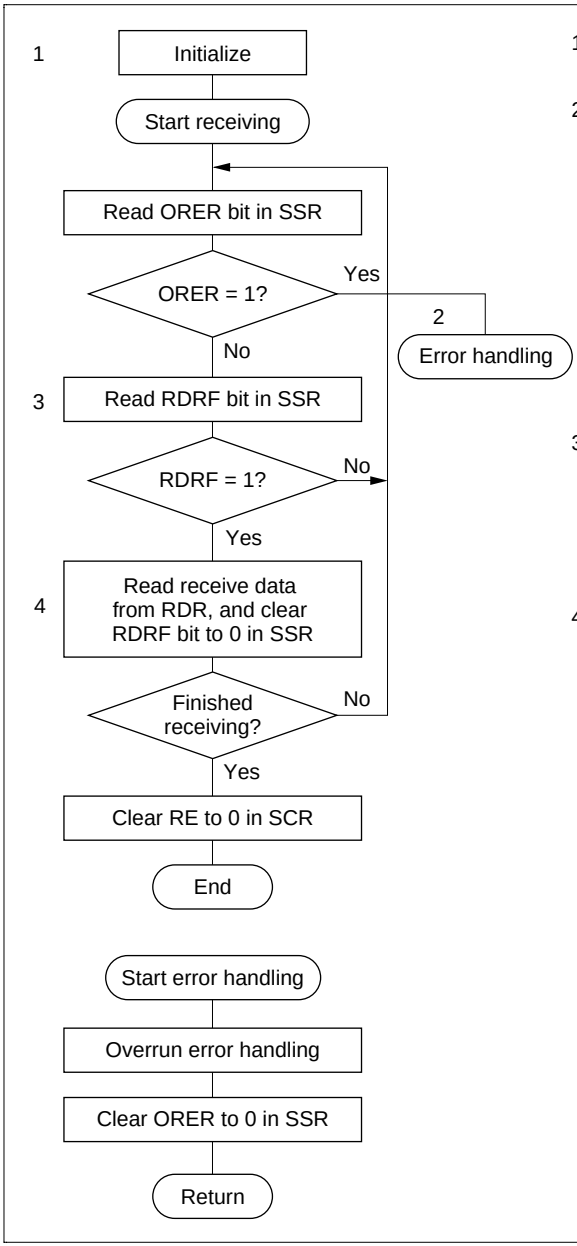


Figure 14.14 Example of SCI Transmit Operation

• **Receiving Serial Data:** Follow the procedure below for receiving serial data. When switching from asynchronous mode to synchronous mode, be sure to check that PER and FER are cleared to 0. If PER or FER is set to 1 the RDRF bit will not be set and both transmitting and receiving will be disabled.



1. SCI initialization: the receive data function of the RxD pin is selected automatically.
2. Receive error handling: if a receive error occurs, read the ORER bit in SSR then, after executing the necessary error handling, clear ORER to 0. Neither transmitting nor receiving can resume while ORER remains set to 1. When clock output mode is selected, receiving can be halted temporarily by receiving one dummy byte and causing an overrun error. When preparations to receive the next data are completed, clear the ORER bit to 0. This causes receiving to resume, so return to the step marked 2 in the flowchart.
3. SCI status check and receive data read: read the serial status register (SSR), check that RDRF is set to 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. Transition of the RDRF bit from 0 to 1 can be reported by an RXI interrupt.
4. To continue receiving serial data: read RDR and clear RDRF to 0 before the MSB (bit 7) of the current frame is received.

Figure 14.15 Sample Flowchart for Serial Receiving

In receiving, the SCI operates as follows.

1. If an external clock is selected, data is input in synchronization with the input clock. If clock output is selected, as soon as the RE bit is set to 1 the SCI begins outputting the serial clock and inputting data. If clock output is stopped because the ORER bit is set to 1, output of the serial clock and input of data resume as soon as the ORER bit is cleared to 0.

2. Receive data is shifted into RSR in order from LSB to MSB.

   After receiving the data, the SCI checks that RDRF is 0 so that receive data can be loaded from RSR into RDR. If this check passes, the SCI sets RDRF to 1 and stores the received data in RDR. If the check does not pass (receive error), the SCI operates as indicated in table 14.8.

   Note: Both transmitting and receiving are disabled while a receive error flag is set. The RDRF bit is not set to 1. Be sure to clear the error flag.

3. After setting RDRF to 1, if the RIE bit (receive-end interrupt enable) is set to 1 in SCR, the SCI requests an RXI (receive-end) interrupt. If the ORER bit is set to 1 and the RIE bit in SCR is set to 1, the SCI requests an ERI (receive-error) interrupt.

   When clock output mode is selected, clock output stops when the RE bit is cleared to 0 or the ORER bit is set to 1. To prevent clock count errors, it is safest to receive one dummy byte and generate an overrun error.

Figure 14.16 shows an example of SCI receive operation.

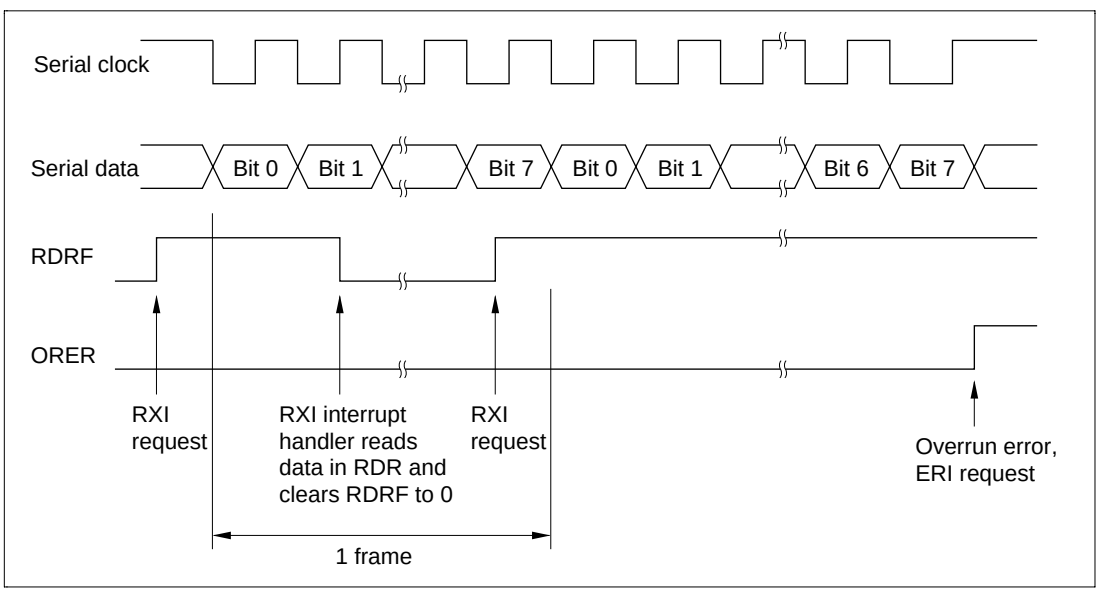
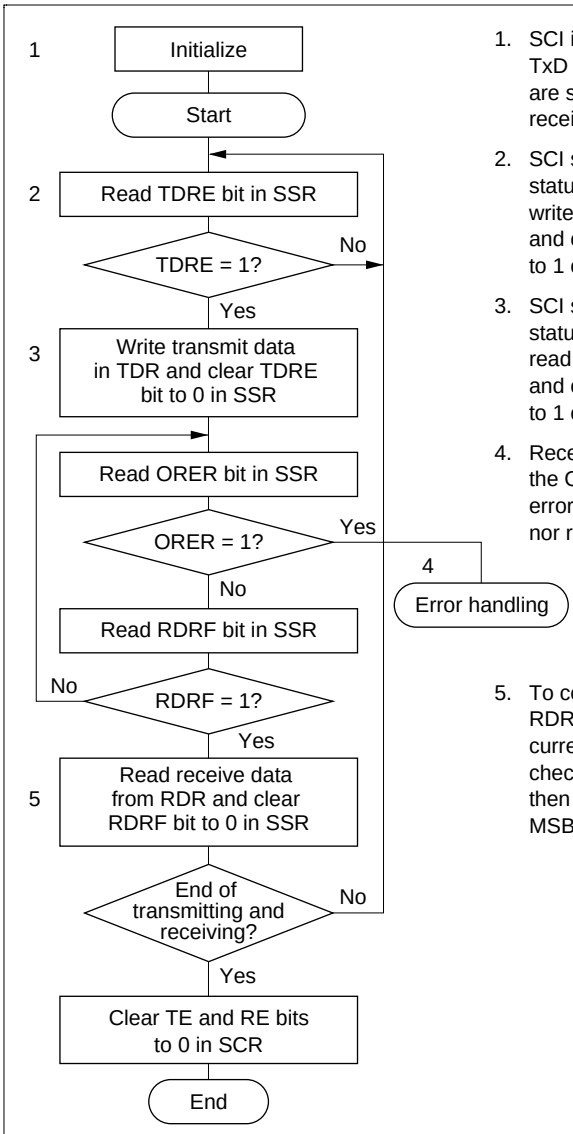


Figure 14.16 Example of SCI Receive Operation

• **Transmitting and Receiving Serial Data Simultaneously:** Follow the procedure below for transmitting and receiving serial data simultaneously. If clock output mode is selected, output of the serial clock begins simultaneously with serial transmission.



1. SCI initialization: the transmit data output function of the TxD pin and receive data input function of the RxD pin are selected, enabling simultaneous transmitting and receiving.
2. SCI status check and transmit data write: read the serial status register (SSR), check that the TDRE bit is 1, then write transmit data in the transmit data register (TDR) and clear TDRE to 0. Transition of the TDRE bit from 0 to 1 can be reported by a TXI interrupt.
3. SCI status check and receive data read: read the serial status register (SSR), check that the RDRF bit is 1, then read receive data from the receive data register (RDR) and clear RDRF to 0. Transition of the RDRF bit from 0 to 1 can be reported by an RXI interrupt.
4. Receive error handling: if a receive error occurs, read the ORER bit in SSR then, after executing the necessary error handling, clear ORER to 0. Neither transmitting nor receiving can resume while ORER remains set to 1.
5. To continue transmitting and receiving serial data: read RDR and clear RDRF to 0 before the MSB (bit 7) of the current frame is received. Also read the TDRE bit and check that it is set to 1, indicating that it is safe to write; then write data in TDR and clear TDRE to 0 before the MSB (bit 7) of the current frame is transmitted.

Figure 14.17 Sample Flowchart for Serial Transmitting and Receiving

Note: In switching from transmitting or receiving to simultaneous transmitting and receiving, clear both TE and RE to 0, then set both TE and RE to 1.
Do not use the BSET instruction for this purpose.

14.4 Interrupts

The SCI can request four types of interrupts: ERI, RXI, TXI, and TEI. Table 14.11 indicates the source and priority of these interrupts. The interrupt sources can be enabled or disabled by the TIE, RIE, and TEIE bits in the SCR. Independent signals are sent to the interrupt controller for each interrupt source, except that the receive-error interrupt (ERI) is the logical OR of three sources: overrun error, framing error, and parity error.

The TXI interrupt indicates that the next transmit data can be written. The TEI interrupt indicates that the SCI has stopped transmitting data.

Table 14.11 SCI Interrupt Sources

Interrupt	Description	Priority
ERI	Receive error interrupt (ORER, FER, or PER)	High
RXI	Receive end interrupt (RDRF)	
TXI	TDR-empty interrupt (TDRE)	
TEI	TSR-empty interrupt (TEND)	Low

14.5 Application Notes

Application programmers should note the following features of the SCI.

TDR Write: The TDRE bit in SSR is simply a flag that indicates that the TDR contents have been transferred to TSR. The TDR contents can be rewritten regardless of the TDRE value. If a new byte is written in TDR while the TDRE bit is 0, before the old TDR contents have been moved into TSR, the old byte will be lost. Software should check that the TDRE bit is set to 1 before writing to the TDR.

Multiple Receive Errors: Table 14.12 lists the values of flag bits in the SSR when multiple receive errors occur, and indicates whether the RSR contents are transferred to RDR.

Table 14.12 SSR Bit States and Data Transfer when Multiple Receive Errors Occur

	SSR Bits				RSR →
Receive error	**RDRF**	**ORER**	**FER**	**PER**	**RDR*[2]**
Overrun error	1*[1]	1	0	0	No
Framing error	0	0	1	0	Yes
Parity error	0	0	0	1	Yes
Overrun and framing errors	1*[1]	1	1	0	No
Overrun and parity errors	1*[1]	1	0	1	No
Framing and parity errors	0	0	1	1	Yes
Overrun, framing, and parity errors	1*[1]	1	1	1	No

Notes: 1. Set to 1 before the overrun error occurs.
2. Yes: The RSR contents are transferred to RDR.
No: The RSR contents are not transferred to RDR.

Line Break Detection: When the RxD pin receives a continuous stream of 0's in asynchronous mode (line-break state), a framing error occurs because the SCI detects a 0 stop bit. The value H'00 is transferred from RSR to RDR. Software can detect the line-break state as a framing error accompanied by H'00 data in RDR.

The SCI continues to receive data, so if the FER bit is cleared to 0 another framing error will occur.

Sampling Timing and Receive Margin in Asynchronous Mode: The serial clock used by the SCI in asynchronous mode runs at 16 times the bit rate. The falling edge of the start bit is detected by sampling the RxD input on the falling edge of this clock. After the start bit is detected, each bit of receive data in the frame (including the start bit, parity bit, and stop bit or bits) is sampled on the rising edge of the serial clock pulse at the center of the bit. See figure 14.18.

It follows that the receive margin can be calculated as in equation (1).

When the absolute frequency deviation of the clock signal is 0 and the clock duty cycle is 0.5, data can theoretically be received with distortion up to the margin given by equation (2). This is a theoretical limit, however. In practice, system designers should allow a margin of 20% to 30%.

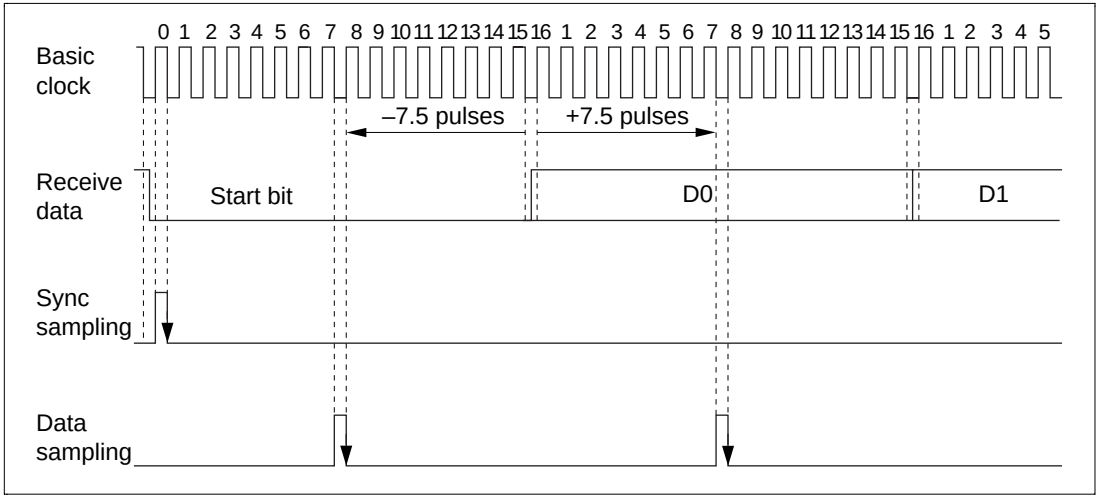


Figure 14.18 Sampling Timing (Asynchronous Mode)

$$M = \left\{ \left(0.5 - \frac{1}{2N} \right) - \frac{D - 0.5}{N} - (L - 0.5)\,F \right\} \times 100\ [\%] \quad \ldots\ldots\ldots\ldots\ (1)$$

M: Receive margin
N: Ratio of basic clock to bit rate (N=16)
D: Duty factor of clock—ratio of high pulse width to low width (0.5 to 1.0)
L: Frame length (9 to 12)
F: Absolute clock frequency deviation

When D = 0.5 and F = 0

$$M = (0.5 - 1/2 \times 16) \times 100\ [\%] = 46.875\% \quad \ldots\ldots\ldots\ldots\ (2)$$

Section 15 A/D Converter

15.1 Overview

The H8/3318 includes a 10-bit successive-approximations A/D converter with a selection of up to eight analog input channels.

15.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Eight input channels
- High-speed conversion
  Conversion time: minimum 8.4 µs per channel (with 16-MHz system clock)
- Two conversion modes
  Single mode: A/D conversion of one channel
  Scan mode: continuous conversion on one to four channels
- Four 16-bit data registers
  A/D conversion results are transferred for storage into data registers corresponding to the channels.
- Sample-and-hold function
- A/D conversion can be externally triggered
- A/D interrupt requested at end of conversion
  At the end of A/D conversion, an A/D end interrupt (ADI) can be requested.

15.1.2 Block Diagram

Figure 15.1 shows a block diagram of the A/D converter.

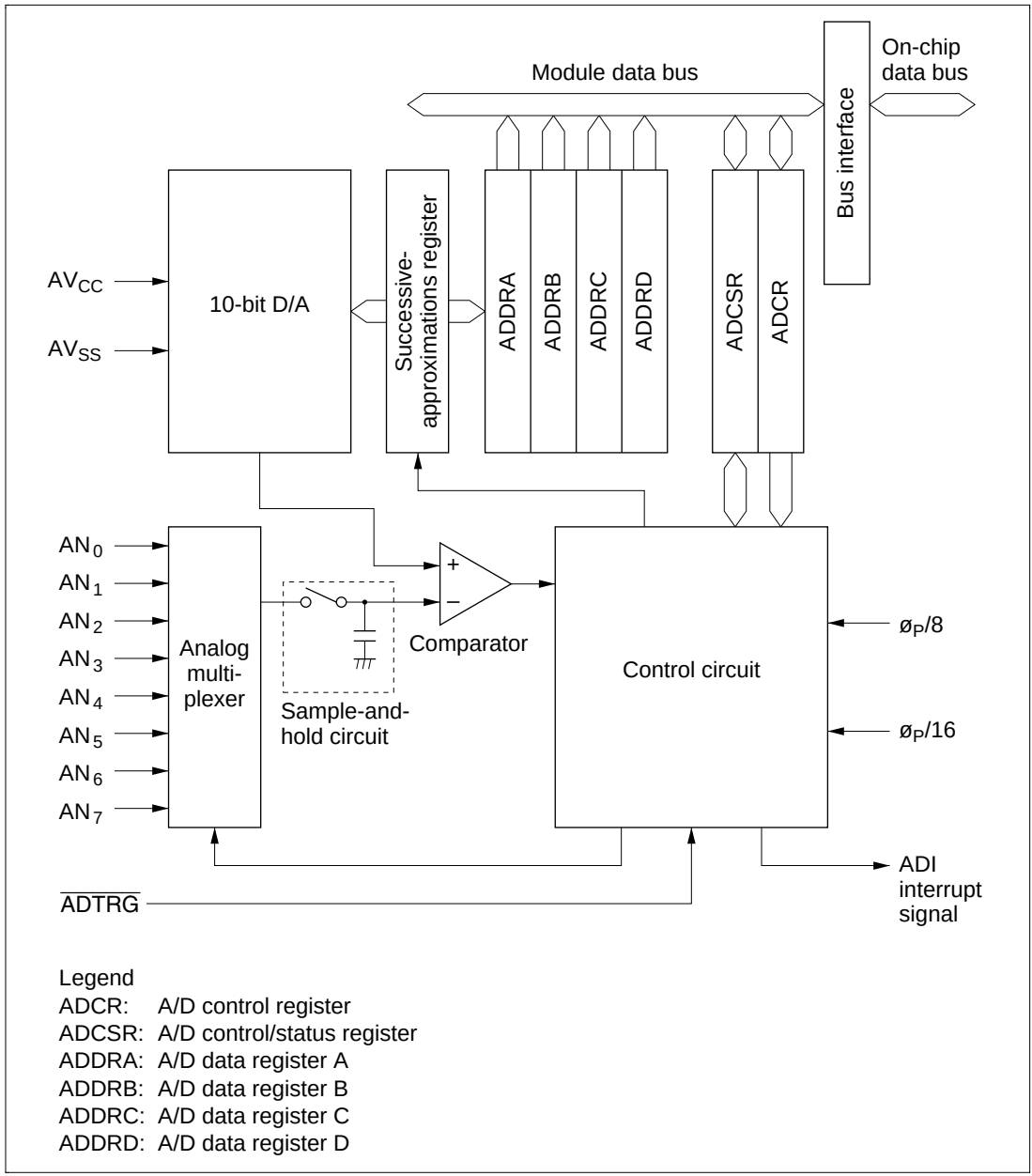


Figure 15.1 A/D Converter Block Diagram

15.1.3 Input Pins

Table 15.1 lists the A/D converter's input pins. The eight analog input pins are divided into two groups: group 0 (AN_0 to AN_3), and group 1 (AN_4 to AN_7). AV_{CC} and AV_{SS} are the power supply for the analog circuits in the A/D converter.

Table 15.1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AV_{CC}	Input	Analog power supply
Analog ground pin	AV_{SS}	Input	Analog ground and reference voltage
Analog input pin 0	AN_0	Input	Group 0 analog inputs
Analog input pin 1	AN_1	Input	
Analog input pin 2	AN_2	Input	
Analog input pin 3	AN_3	Input	
Analog input pin 4	AN_4	Input	Group 1 analog inputs
Analog input pin 5	AN_5	Input	
Analog input pin 6	AN_6	Input	
Analog input pin 7	AN_7	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

15.1.4 Register Configuration

Table 15.2 summarizes the A/D converter's registers.

Table 15.2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address
A/D data register A (high)	ADDRAH	R	H'00	H'FFE0
A/D data register A (low)	ADDRAL	R	H'00	H'FFE1
A/D data register B (high)	ADDRBH	R	H'00	H'FFE2
A/D data register B (low)	ADDRBL	R	H'00	H'FFE3
A/D data register C (high)	ADDRCH	R	H'00	H'FFE4
A/D data register C (low)	ADDRCL	R	H'00	H'FFE5
A/D data register D (high)	ADDRDH	R	H'00	H'FFE6
A/D data register D (low)	ADDRDL	R	H'00	H'FFE7
A/D control/status register	ADCSR	R/(W)*	H'00	H'FFE8
A/D control register	ADCR	R/W	H'7F	H'FFE9

Note: * Only 0 can be written in bit 7, to clear the flag.

15.2 Register Descriptions

15.2.1 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	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

Bits 15 to 6—A/D Conversion Data (AD9 to AD0): 10-bit data giving an A/D conversion result.

Bits 5 to 0—Reserved: These bits cannot be modified and are always read as 0.

The four A/D data registers (ADDRA to ADDRD) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register corresponding to the selected channel. The upper 8 bits of the result are stored in the upper byte of the A/D data register. The lower 2 bits are stored in the lower byte. Bits 5 to 0 of an A/D data register are reserved bits that always read 0. Table 15.3 indicates the pairings of analog input channels and A/D data registers.

The CPU can always read and write the A/D data registers. The upper byte can be read directly, but the lower byte is read through a temporary register (TEMP). For details see section 15.3, CPU Interface.

The A/D data registers are initialized to H'0000 by a reset and in standby mode.

Table 15.3 Analog Input Channels and A/D Data Registers

Analog Input Channel		
Group 0	**Group 1**	**A/D Data Register**
AN_0	AN_4	ADDRA
AN_1	AN_5	ADDRB
AN_2	AN_6	ADDRC
AN_3	AN_7	ADDRD

15.2.2 A/D Control/Status Register (ADCSR)

Bit	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

ADCSR is an 8-bit readable/writable register that selects the mode and controls the A/D converter. ADCSR is initialized to H'00 by a reset and in standby mode.

Bit 7—A/D End Flag (ADF): Indicates the end of A/D conversion.

Bit 7 ADF	Description	
0	[Clearing condition]	(Initial value)
	Cleared by reading ADF while ADF = 1, then writing 0 in ADF	
1	[Setting conditions]	
	1. Single mode: A/D conversion ends	
	2. Scan mode: A/D conversion ends in all selected channels	

Bit 6—A/D Interrupt Enable (ADIE): Enables or disables the interrupt (ADI) requested at the end of A/D conversion.

Bit 6 ADIE	Description	
0	A/D end interrupt request (ADI) is disabled	(Initial value)
1	A/D end interrupt request (ADI) is enabled	

Bit 5—A/D Start (ADST): Starts or stops A/D conversion. The ADST bit remains set to 1 during A/D conversion. It can also be set to 1 by external trigger input at the $\overline{\text{ADTRG}}$ pin.

Bit 5 ADST	Description	
0	A/D conversion is stopped	(Initial value)
1	1. Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends 2. Scan mode: A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode	

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode. For further information on operation in these modes, see section 15.4, Operation. Clear the ADST bit to 0 before switching the conversion mode.

Bit 4 SCAN	Description	
0	Single mode	(Initial value)
1	Scan mode	

Bit 3—Clock Select (CKS): Selects the A/D conversion time. Clear the ADST bit to 0 before switching the conversion time. If $ø_P = ø/2$, the conversion time is twice as long.

Bit 3 CKS	Description	
0	Conversion time = 266 states (maximum) (when $ø_P = ø$)	(Initial value)
1	Conversion time = 134 states (maximum) (when $ø_P = ø$)	

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): These bits and the SCAN bit select the analog input channels. Clear the ADST bit to 0 before changing the channel selection.

Group Selection	Channel Selection		Description	
CH2	**CH1**	**CH0**	**Single Mode**	**Scan Mode**
0	0	0	AN_0 (initial value)	AN_0
		1	AN_1	AN_0, AN_1
	1	0	AN_2	AN_0 to AN_2
		1	AN_3	AN_0 to AN_3
1	0	0	AN_4	AN_4
		1	AN_5	AN_4, AN_5
	1	0	AN_6	AN_4 to AN_6
		1	AN_7	AN_4 to AN_7

15.2.3 A/D Control Register (ADCR)

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

ADCR is an 8-bit readable/writable register that enables or disables external triggering of A/D conversion. ADCR is initialized to H'7F by a reset and in standby mode.

Bit 7—Trigger Enable (TRGE): Enables or disables external triggering of A/D conversion.

Bit 7 TRGE	Description
0	A/D conversion cannot be externally triggered (Initial value)
1	Enables start of A/D conversion at the falling edge of the external trigger signal ($\overline{\text{ADTRG}}$) (A/D conversion can be started by an external trigger or by software.)

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

15.3 CPU Interface

ADDRA to ADDRD are 16-bit registers, but they are connected to the CPU by an 8-bit data bus. Therefore, although the upper byte can be be accessed directly by the CPU, the lower byte is read through an 8-bit temporary register (TEMP).

An A/D data register is read as follows. When the upper byte is read, the upper-byte value is transferred directly to the CPU and the lower-byte value is transferred into TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading an A/D data register, always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 15.2 shows the data flow for access to an A/D data register.

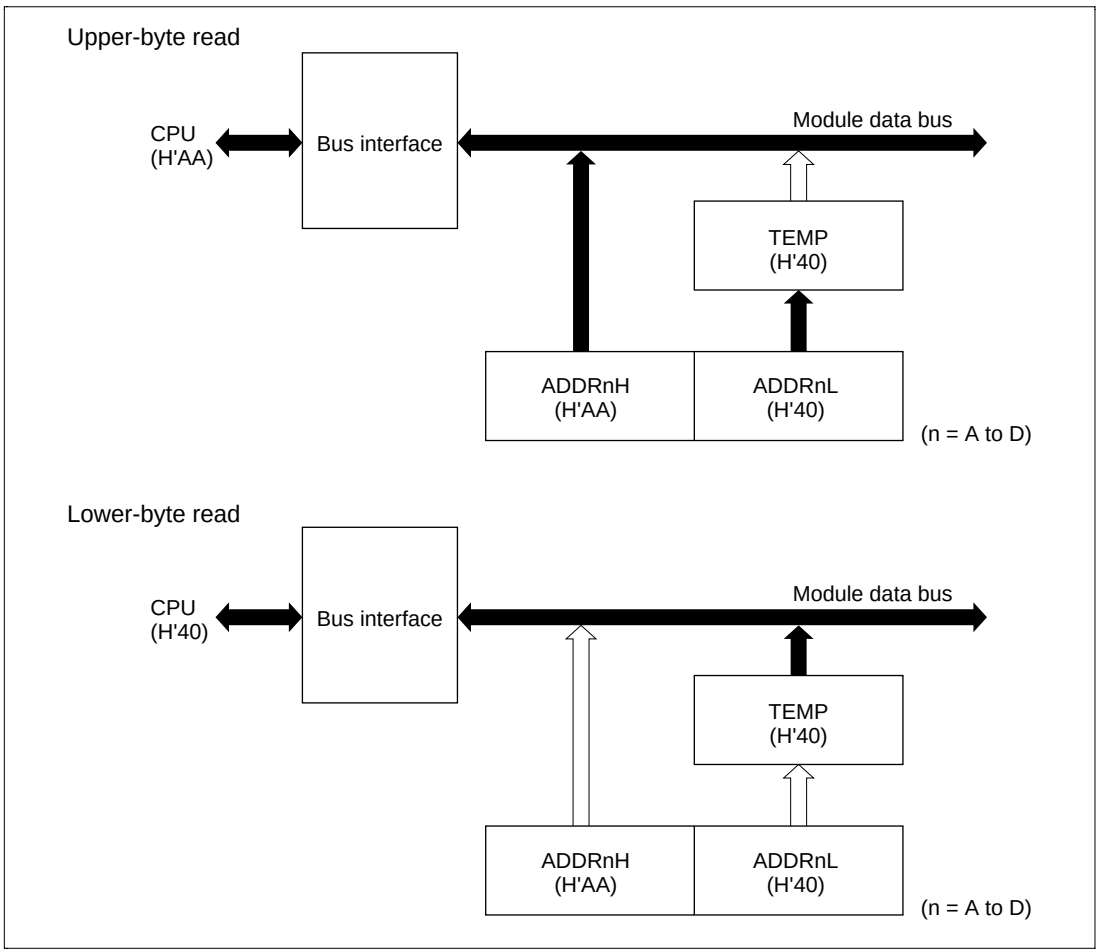


Figure 15.2 A/D Data Register Access Operation (Reading H'AA40)

15.4 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

15.4.1 Single Mode (SCAN = 0)

Single mode should be selected when only one A/D conversion on one channel is required. A/D conversion starts when the ADST bit is set to 1 by software, or by external trigger input. The ADST bit remains set to 1 during A/D conversion and is automatically cleared to 0 when conversion ends.

When conversion ends the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in ADF.

When the mode or analog input channel must be switched during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the mode or channel is changed.

Typical operations when channel 1 (AN_1) is selected in single mode are described next. Figure 15.3 shows a timing diagram for this example.

1. Single mode is selected (SCAN = 0), input channel AN_1 is selected (CH2 = CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 in the ADF flag.
6. The routine reads and processes the conversion result (ADDRB).
7. Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is set to 1, A/D conversion starts again and steps 2 to 7 are repeated.

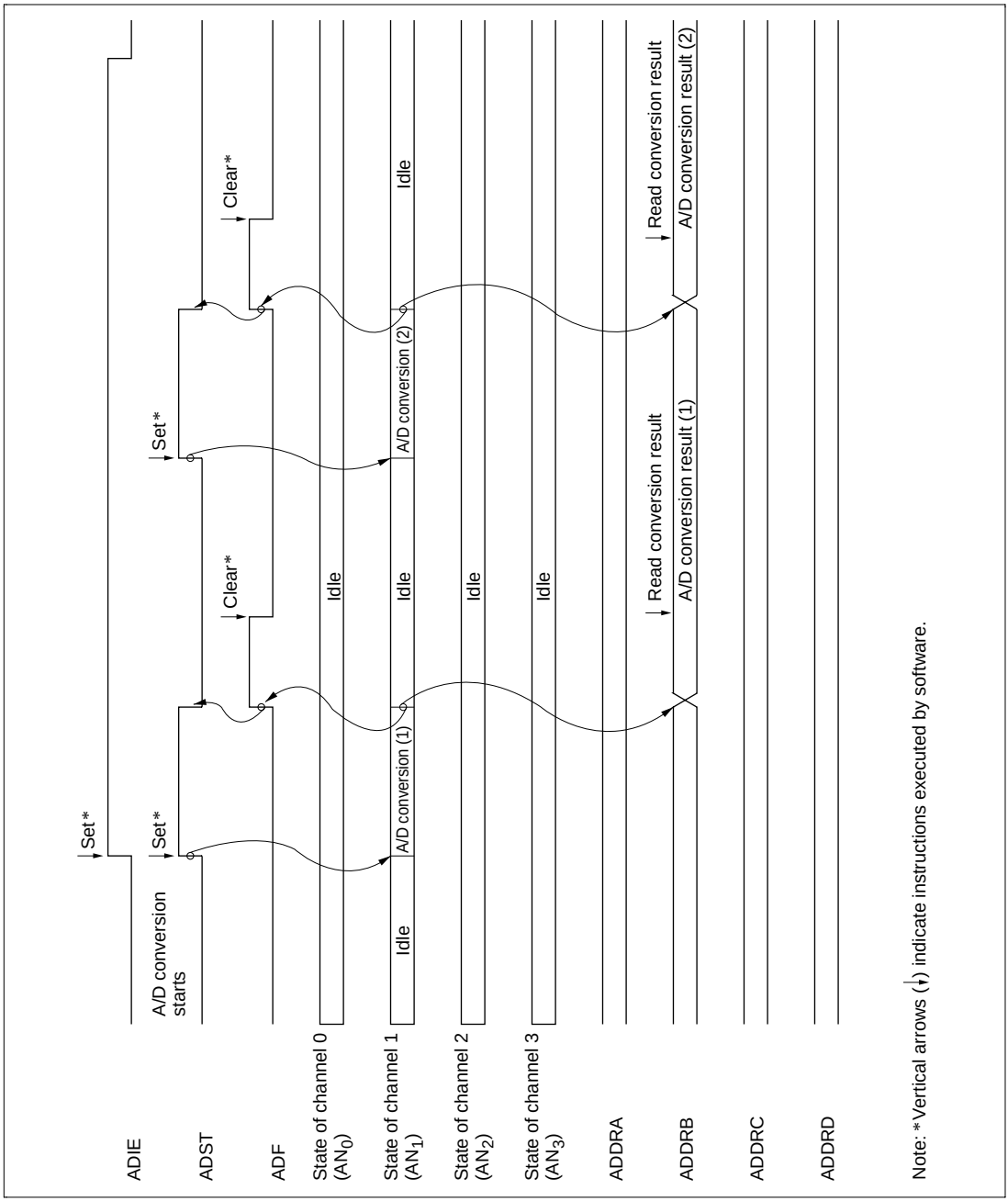


Figure 15.3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

15.4.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by software or external trigger input, A/D conversion starts on the first channel in the group (AN_0 when CH2 = 0, AN_4 when CH2 = 1). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN_1 or AN_5) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers corresponding to the channels.

When the mode or analog input channel selection must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. A/D conversion will start again from the first channel in the group. The ADST bit can be set at the same time as the mode or channel selection is changed.

Typical operations when three channels in group 0 (AN_0 to AN_2) are selected in scan mode are described next. Figure 15.4 shows a timing diagram for this example.

1. Scan mode is selected (SCAN = 1), scan group 0 is selected (CH2 = 0), analog input channels AN_0 to AN_2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).

2. When A/D conversion of the first channel (AN_0) is completed, the result is transferred into ADDRA. Next, conversion of the second channel (AN_1) starts automatically.

3. Conversion proceeds in the same way through the third channel (AN_2).

4. When conversion of all selected channels (AN_0 to AN_2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN_0) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.

5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN_0).

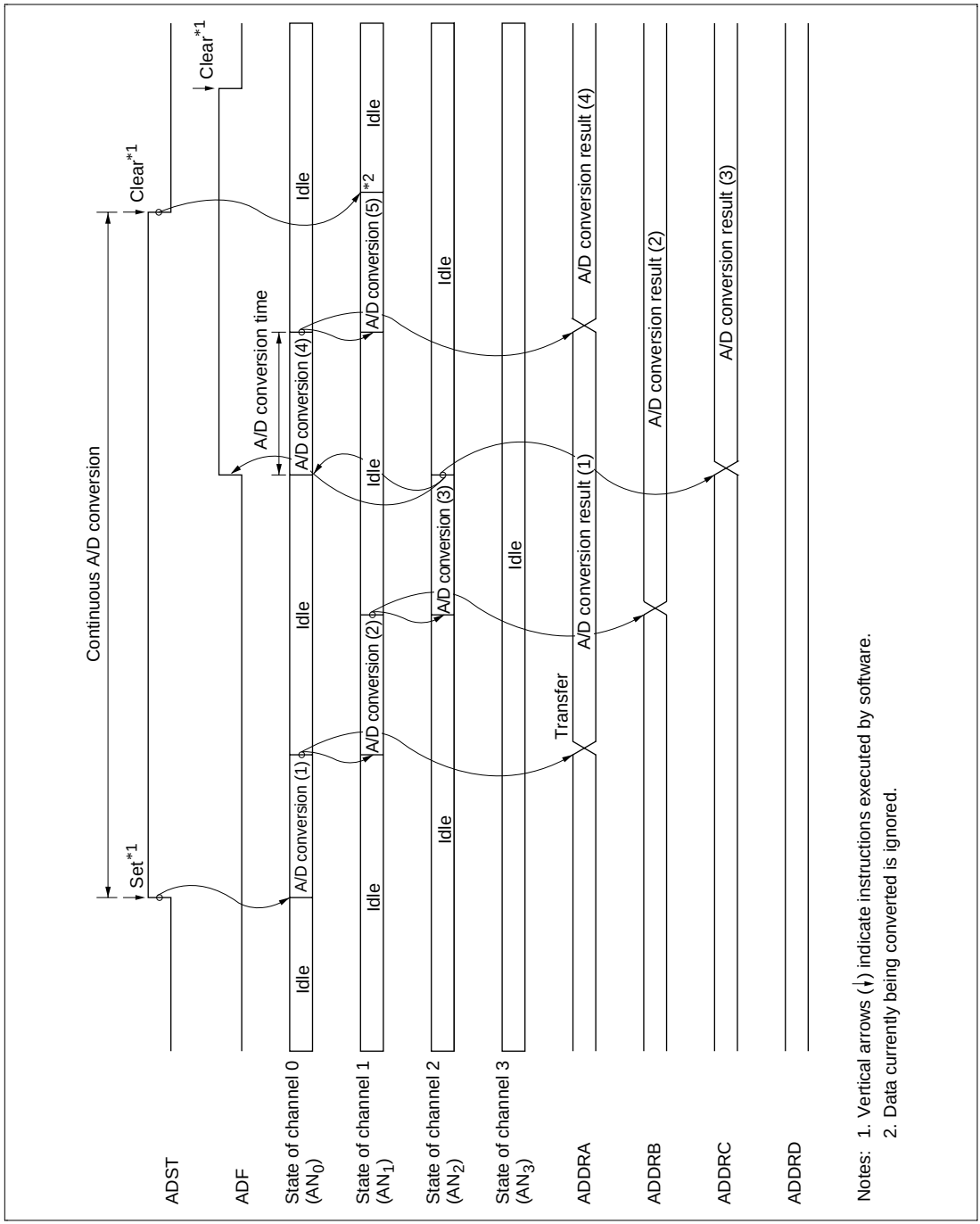


Figure 15.4 Example of A/D Converter Operation (Scan Mode, Channels AN_0 to AN_2 Selected)

15.4.3 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 15.5 shows the A/D conversion timing. Table 15.4 indicates the A/D conversion time.

As indicated in figure 15.5, 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 15.4.

In scan mode, the values given in table 15.4 apply to the first conversion. In the second and subsequent conversions the conversion time is fixed at 256 states when CKS = 0 or 128 states when CKS = 1 (when $ø_P = ø$).

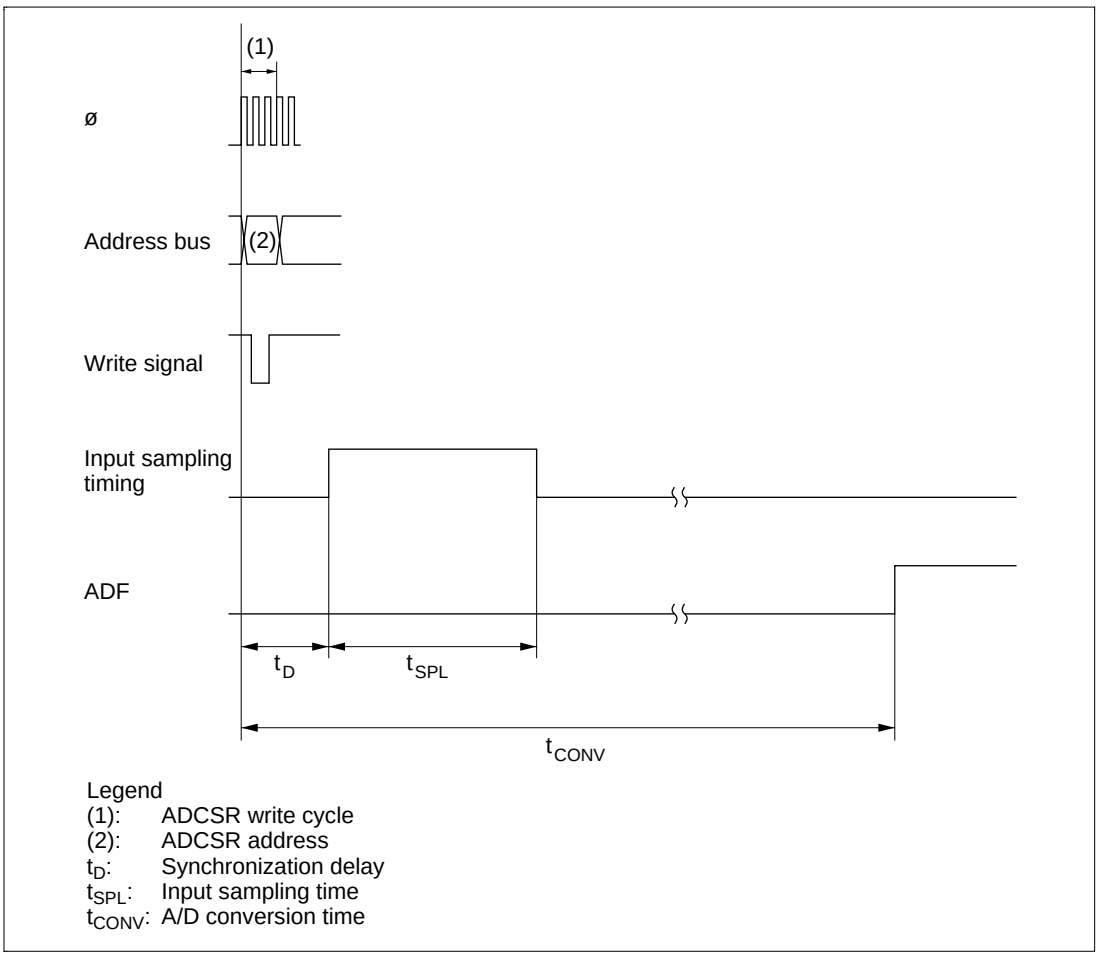


Figure 15.5 A/D Conversion Timing

Table 15.4 A/D Conversion Time (Single Mode)

		CKS = 0			CKS = 1		
	Symbol	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
Synchronization delay	t_D	10	—	17	6	—	9
Input sampling time*	t_{SPL}	—	80	—	—	40	—
A/D conversion time*	t_{CONV}	259	—	266	131	—	134

Notes: Values in the table are numbers of states.

* Values when $ø_P = ø$. If $ø_P = ø/2$, the values are twice those shown.

15.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE bit is set to 1 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A high-to-low transition at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as if the ADST bit had been set to 1 by software. Figure 15.6 shows the timing.

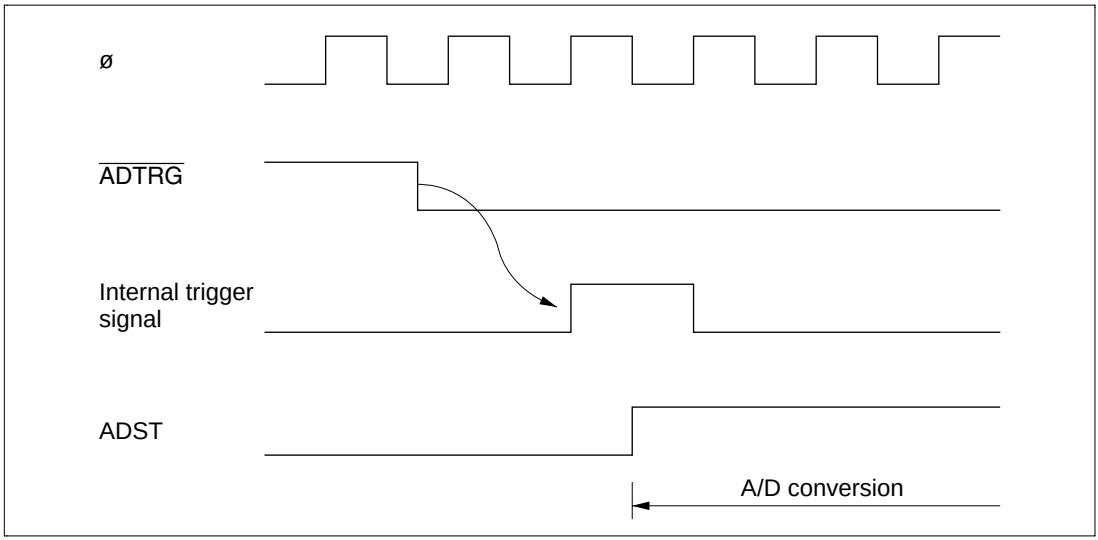


Figure 15.6 External Trigger Input Timing

15.5 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.

15.6 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 AN_n analog input pins during A/D conversion should be in the range $AV_{SS} \leq AN_n \leq AV_{CC}$ (n = 0 to 7).

2. AV_{CC} and AV_{SS} input voltages
   AV_{CC} should have the following value: $AV_{SS} = V_{SS}$. If the A/D converter is not used, the values should be $AV_{CC} = V_{CC}$ and $AV_{SS} = V_{SS}$.

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), analog reference power supply (AV_{ref}), 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 15.7.

Also, the bypass capacitors connected to AV_{CC}, 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 15.7, 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 (Rin), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.

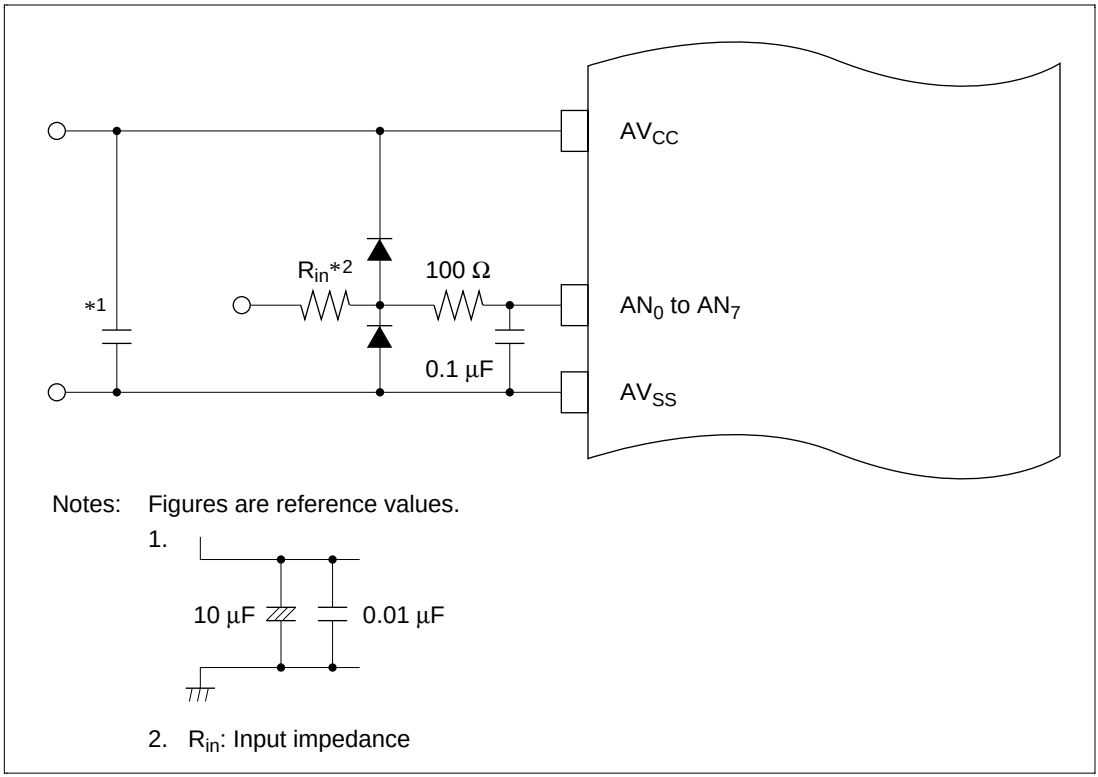


Figure 15.7 Example of Analog Input Protection Circuit

A/D Conversion Precision Definitions: H8/3318 A/D conversion precision definitions are given below.

- Resolution

  The number of A/D converter digital output codes

- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'000) to B'0000000001 (H'001) (see figure 15.9).

- 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 15.10).

- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 15.8).

- 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.

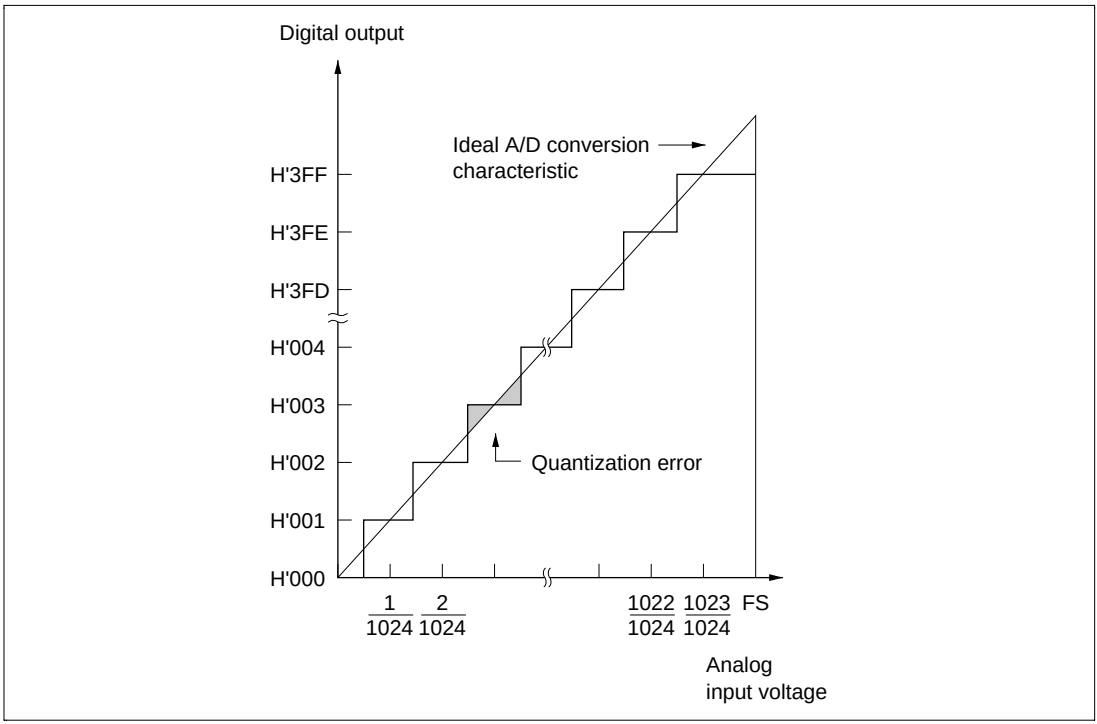


Figure 15.8 A/D Conversion Precision Definitions (1)

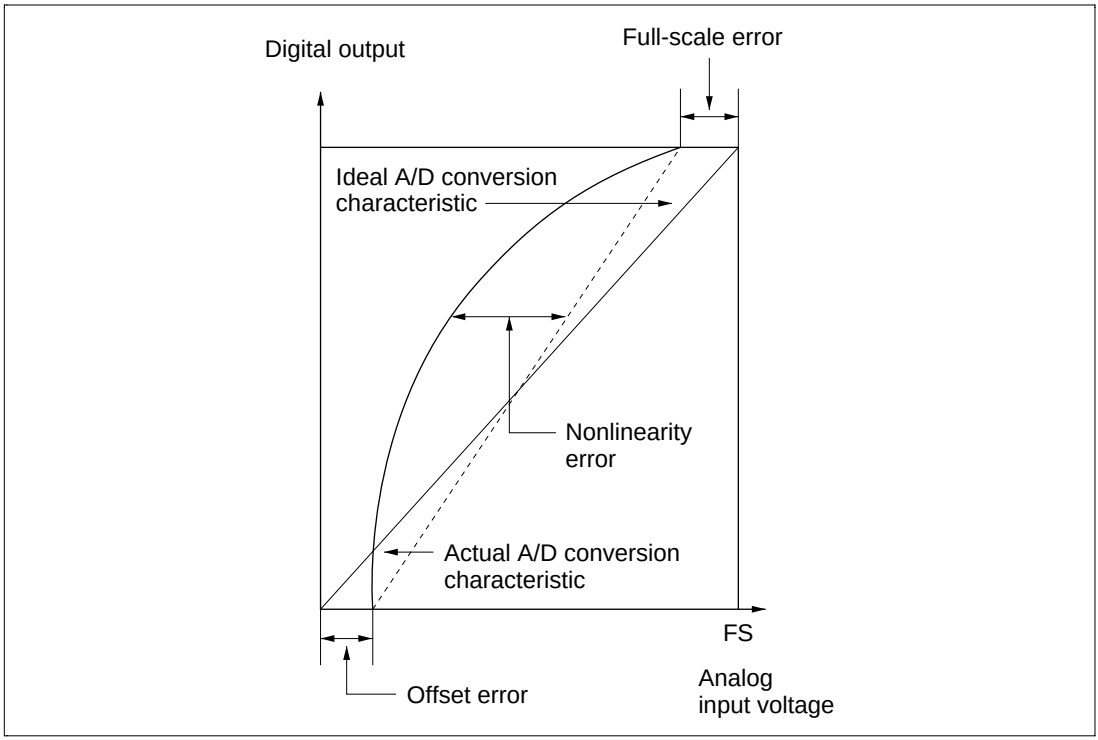


Figure 15.9 A/D Conversion Precision Definitions (2)

Permissible Signal Source Impedance: H8/3318 analog input is designed so that conversion precision is guaranteed for an input signal for which the signal source impedance is 10 kΩ 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, 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 AVSS.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, so acting as antennas.

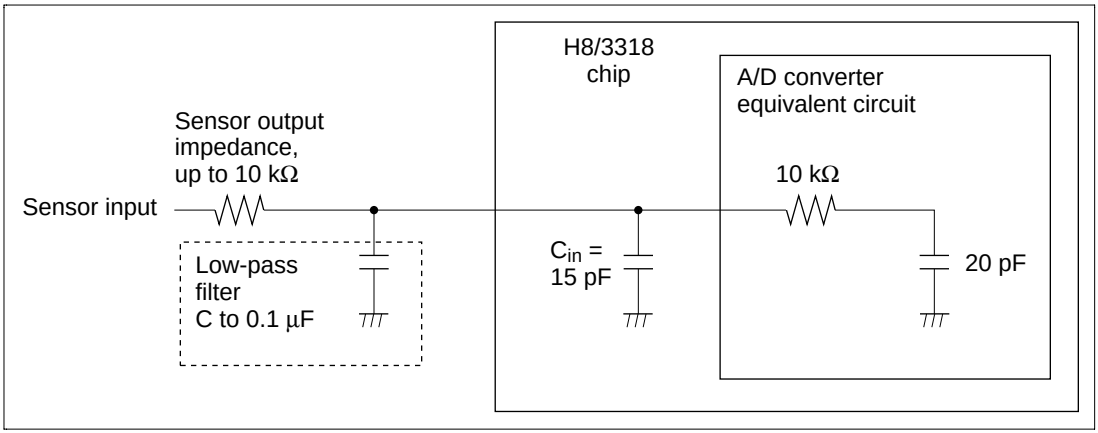


Figure 15.10 Example of Analog Input Circuit

Section 16 RAM

16.1 Overview

The H8/3318 includes 4 kbytes, of on-chip static RAM. The RAM is connected to the CPU via 16-bit data bus. Both byte and word access to the on-chip RAM are performed in two states, enabling rapid data transfer and instruction execution.

The on-chip RAM is assigned to addresses H'EF80 to H'FF7F in the H8/3318. The RAME bit in the system control register (SYSCR) can enable or disable the on-chip RAM.

16.1.1 Block Diagram

Figure 16.1 is a block diagram of the on-chip RAM.

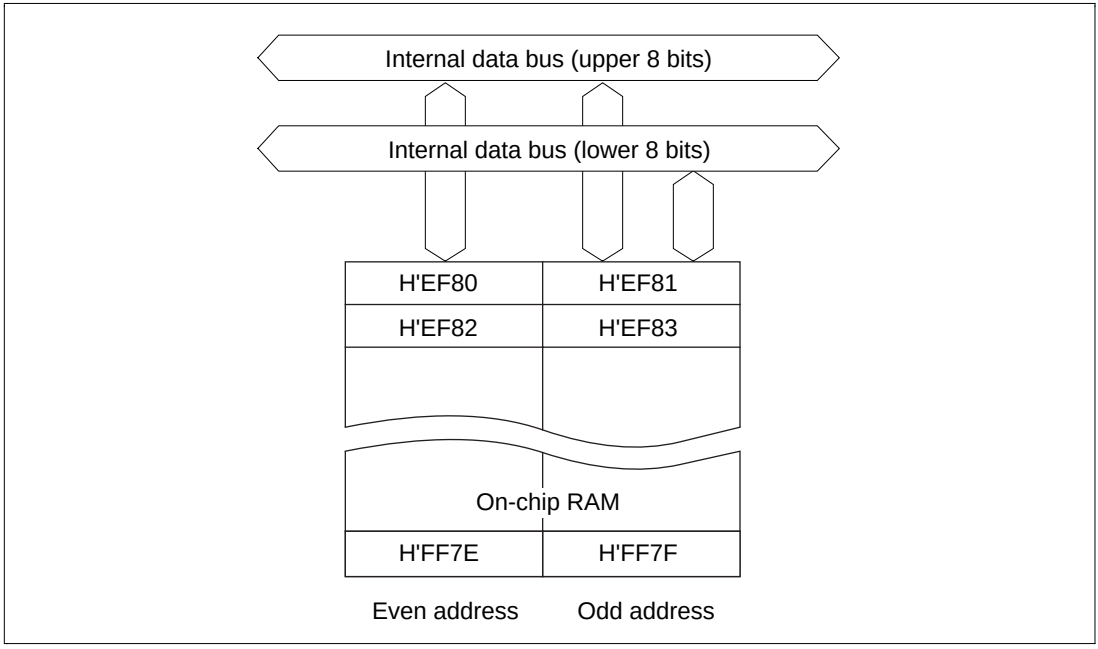


Figure 16.1 Block Diagram of On-Chip RAM

16.1.2 RAM Enable Bit (RAME) in System Control Register (SYSCR)

The on-chip RAM is enabled or disabled by the RAME (RAM Enable) bit in the system control register (SYSCR).

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

The only bit in the system control register that concerns the on-chip RAM is the RAME bit. See section 3.2, System Control Register, for the other bits.

Bit 0—RAM Enable (RAME): This bit enables or disables the on-chip RAM.

The RAME bit is initialized to 1 on the rising edge of the $\overline{\text{RES}}$ signal, so a reset enables the on-chip RAM. The RAME bit is not initialized in software standby mode.

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

16.2 Operation

16.2.1 Expanded Modes (Modes 1 and 2)

If the RAME bit is set to 1, accesses to addresses H'EF80 to H'FF7F in the H8/3318, are directed to the on-chip RAM. If the RAME bit is cleared to 0, accesses to these addresses are directed to the external data bus.

16.2.2 Single-Chip Mode (Mode 3)

If the RAME bit is set to 1, accesses to addresses H'EF80 to H'FF7F in the H8/3318, are directed to the on-chip RAM.

If the RAME bit is cleared to 0, the on-chip RAM data cannot be accessed. Attempted write access has no effect. Attempted read access always results in H'FF data being read.

16.3 Application Notes

16.3.1 Note on Initial Values

The initial value of RAM is undetermined after a power-on reset. Areas to be used must be initialized.

Section 17 ROM

17.1 Overview

The H8/3318 includes 60 kbytes, of high-speed, on-chip ROM. The on-chip ROM is connected to the CPU via a 16-bit data bus. Both byte data and word data are accessed in two states, enabling rapid data transfer and instruction fetching.

The on-chip ROM is enabled or disabled depending on the MCU operating mode, which is determined by the inputs at the mode pins (MD_1 and MD_0). See table 17.1.

Table 17.1 On-Chip ROM Usage in Each MCU Mode

	Mode Pins		
Mode	**MD_1**	**MD_0**	**On-Chip ROM**
Mode 1 (expanded mode)	0	1	Disabled (external addresses)
Mode 2 (expanded mode)	1	0	Enabled
Mode 3 (single-chip mode)	1	1	Enabled

In PROM mode the PROM version (H8/3318 ZTAT) can be programmed with a standard PROM programmer.

In the H8/3318, the accessible ROM addresses are H'0000 to H'E77F (59,264 bytes) in mode 2, and H'0000 to H'EF7F (61,312 bytes) in mode 3. For details, see section 3, MCU Operating Modes and Address Space.

17.1.1 Block Diagram

Figure 17.1 is a block diagram of the on-chip ROM.

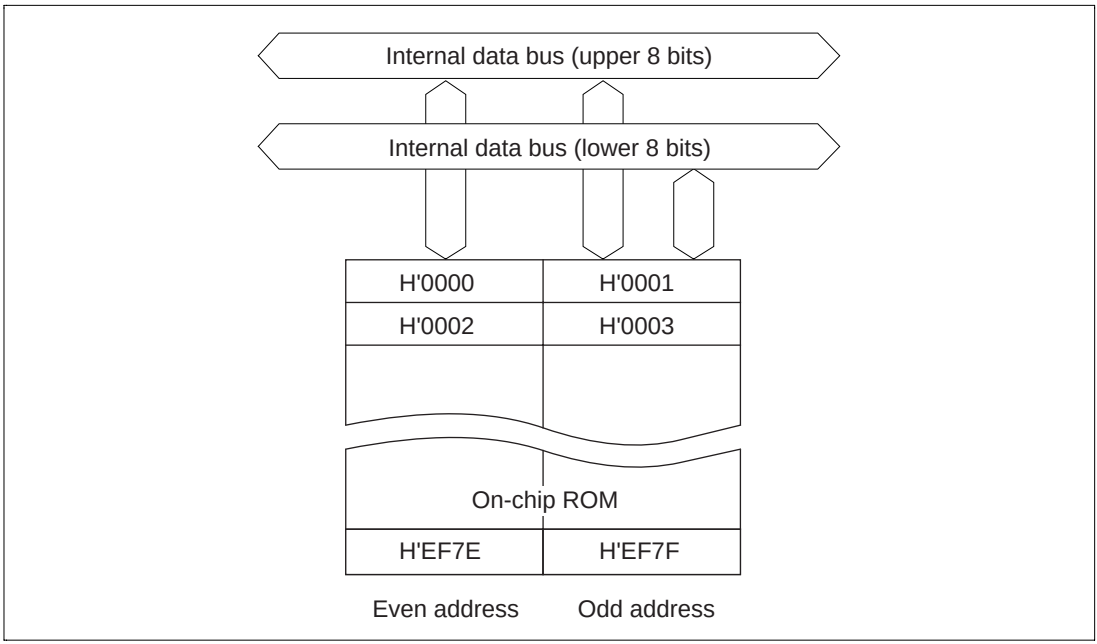


Figure 17.1 Block Diagram of On-Chip ROM (H8/3318 Single-Chip Mode)

17.2 PROM Mode (H8/3318)

17.2.1 PROM Mode Setup

In PROM mode, the H8/3318 PROM version suspends its microcomputer functions to allow the on-chip PROM to be programmed. The programming method is the same as for the HN27C101.

To select PROM mode, apply the signal inputs listed in table 17.2.

Table 17.2 Selection of PROM Mode

Pin	Input
Mode pin MD_1	Low
Mode pin MD_0	Low
$\overline{STBY}$ pin	Low
Pins $P6_3$ and $P6_4$	High

17.2.2 Socket Adapter Pin Assignments and Memory Map

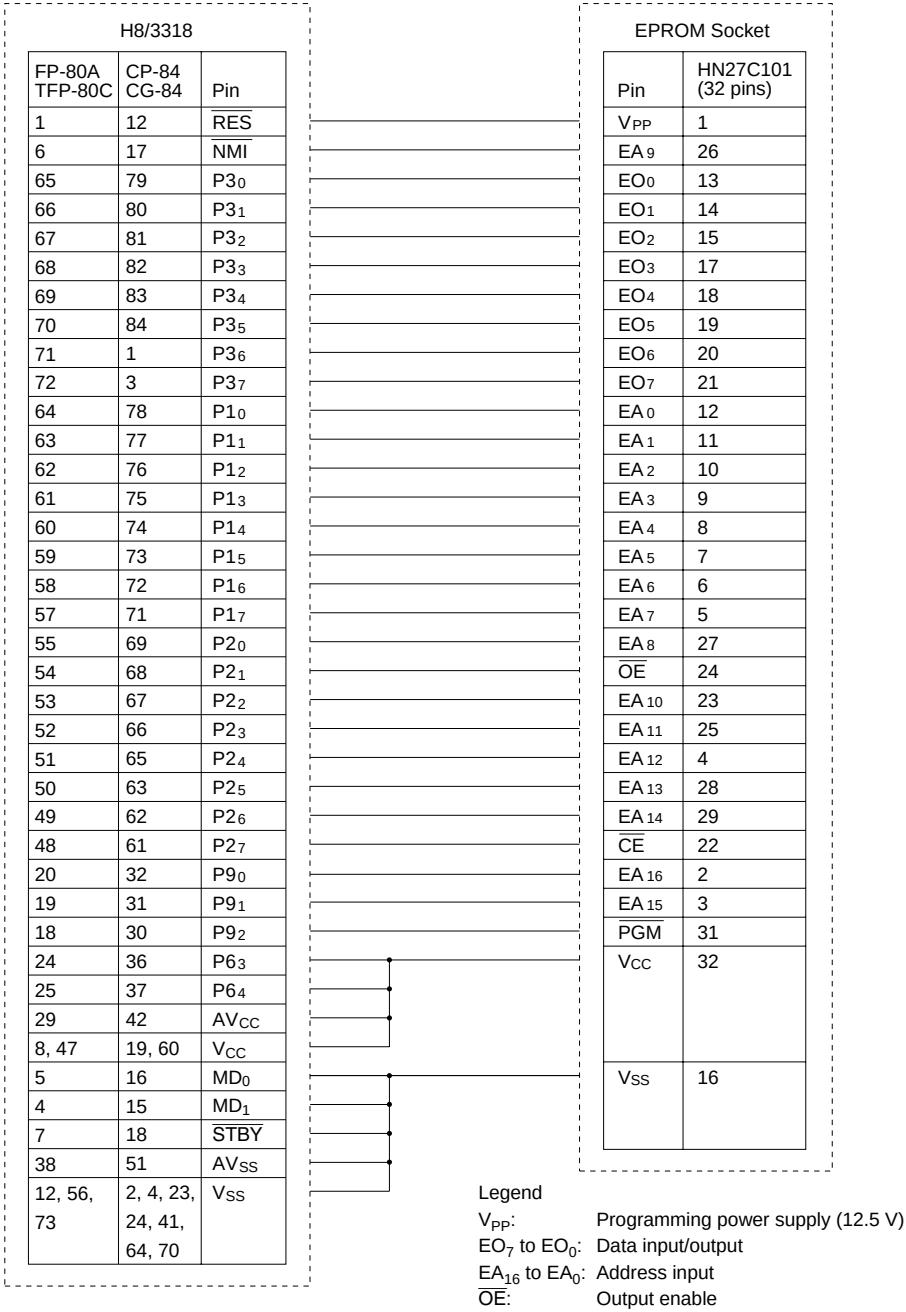
The H8/3318 PROM version can be programmed with a general-purpose PROM programmer by using a socket adapter to change the pin-out to 32 pins. There are different socket adapters for different packages as listed in table 17.3. Figure 17.2 shows the socket adapter pin assignments.

Table 17.3 Socket Adapters

Package	Socket Adapter	
84-pin PLCC	HS338ESC02H	HS3337ESCS1H
80-pin QFP	HS338ESH02H	HS3337ESHS1H
80-pin TQFP	—	HS3337ESNS1H
84-pin windowed LCC	HS338ESG02H	HS3337ESGS1H

The PROM size is 60 kbytes in the H8/3318. Figure 17.3 shows the memory map of the H8/3318 in PROM 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'EF7F. Specify H'FF data for addresses H'EF80 and above. If these addresses are programmed by mistake, it may become impossible to program or verify the PROM data. The same is true if page programming is attempted. Particular care is required with a plastic package, since the programmed data cannot be erased.



Note: All pins not listed in this figure should be left open.

Figure 17.2 Socket Adapter Pin Assignments

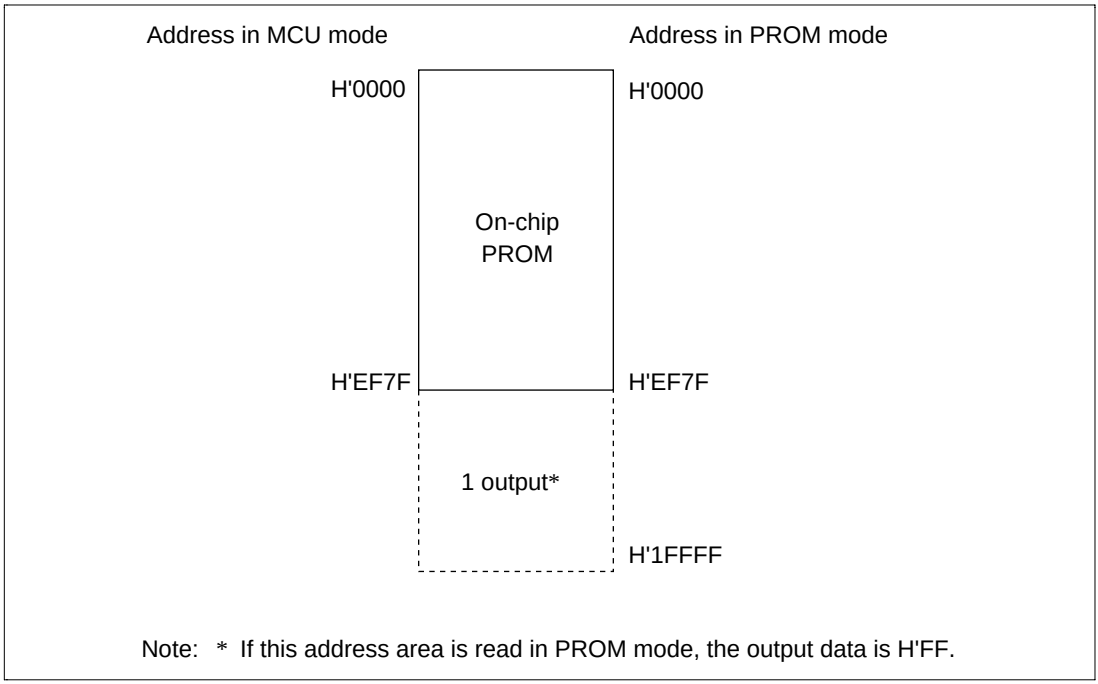


Figure 17.3 H8/3318 Memory Map in PROM Mode

17.3 Programming

The write, verify, and other sub-modes of the PROM mode are selected as shown in table 17.4.

Table 17.4 Selection of Sub-Modes in PROM 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 H8/3318 PROM has 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'EF7F.

17.3.1 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 H'FF in unused addresses.

Figure 17.4 shows the basic high-speed programming flowchart.

Tables 17.5 and 17.6 list the electrical characteristics of the chip in PROM mode. Figure 17.5 shows a program/verify timing chart.

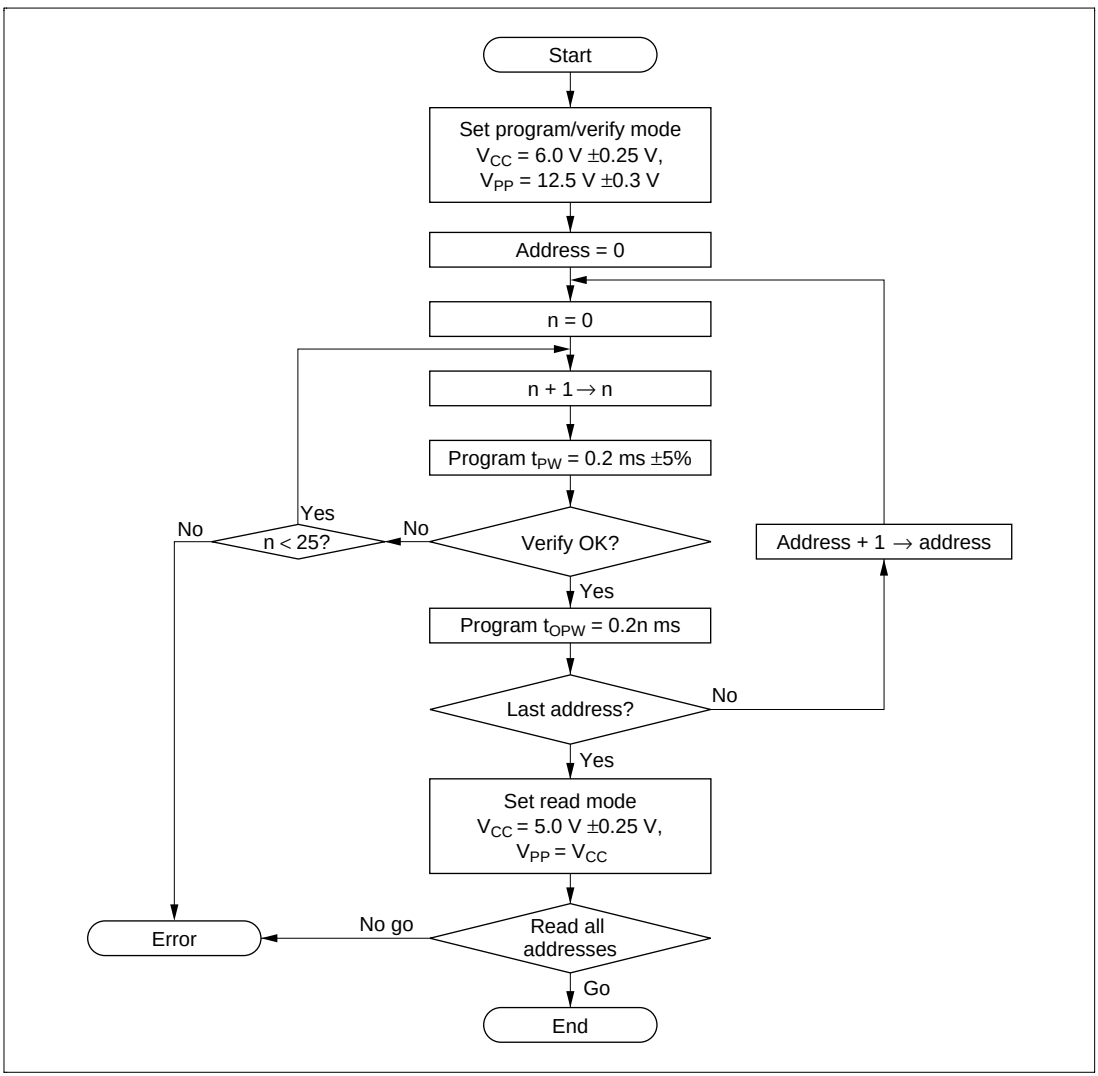


Figure 17.4 High-Speed Programming Flowchart

Table 17.5 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 17.6 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 17.5*
$\overline{OE}$ 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

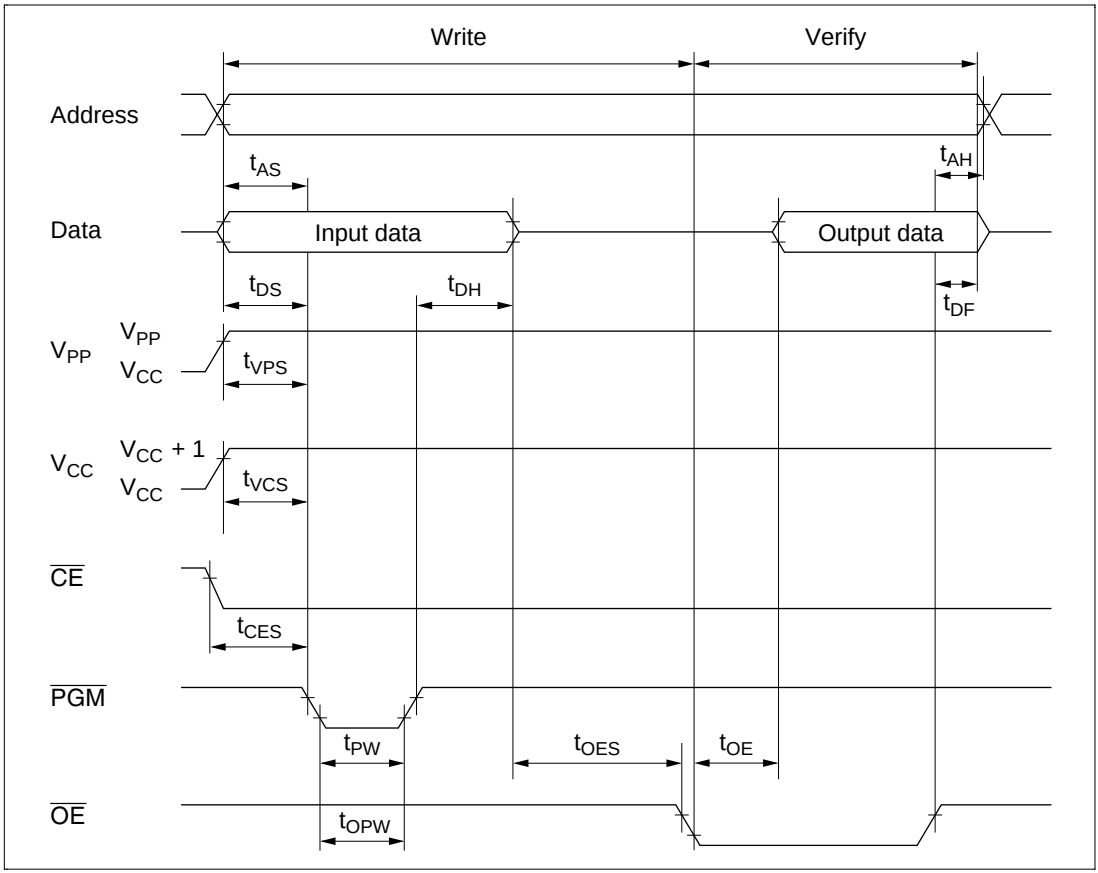


Figure 17.5 PROM Program/Verify Timing

17.3.2 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.

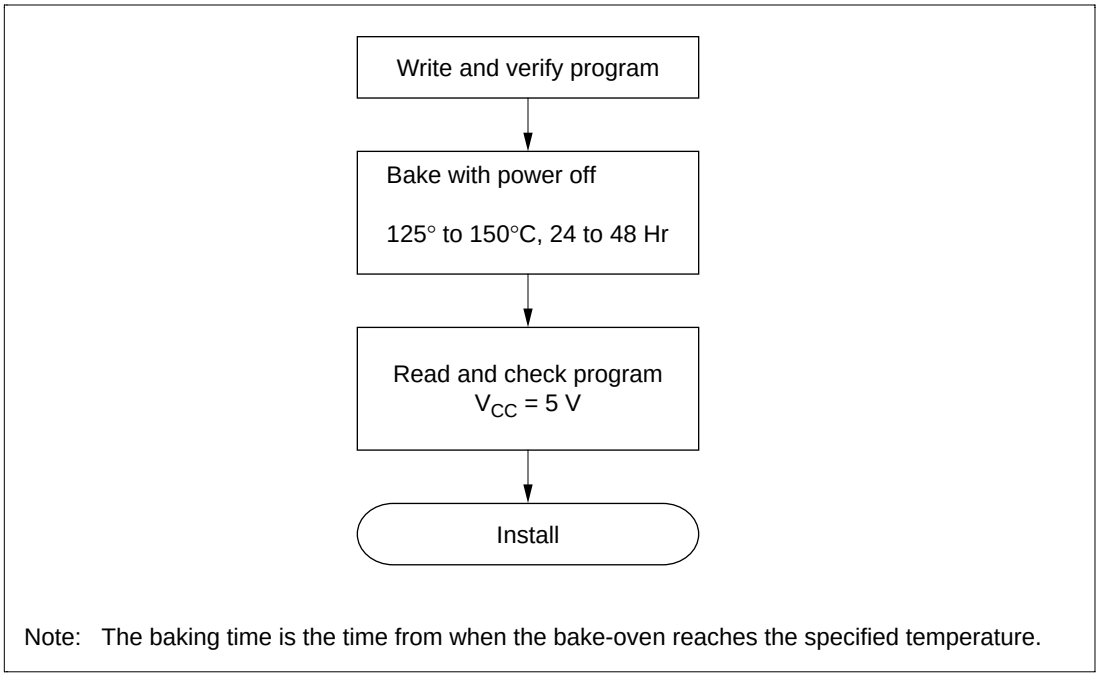
(4) Page programming is not supported. Do not select page programming mode.

(5) The PROM size is 60 kbytes. Set the address range to H'0000 to H'EF7F. When programming, specify H'FF data for unused address areas (H'EF80 to H'1FFFF).

17.3.3 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 17.6 shows the recommended screening procedure.



Note: The baking time is the time from when the bake-oven reaches the specified temperature.

Figure 17.6 Recommended Screening Procedure

If a series 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 Hitachi of any abnormal conditions noted during programming or in screening of program data after high-temperature baking.

17.3.4 Erasing Data

Data is erased by exposing the transparent window in the package to ultraviolet light. The erase conditions are shown in table 17.7.

Table 17.7 Erase Conditions

Item	Value
Ultraviolet wavelength	253.7nm
Minimum irradiation	15W · s/cm^2

The erase conditions in table 17.7 can be met by exposure to a 12000 µW/cm^2 ultraviolet lamp positioned 2 to 3 cm directly above the chip for approximately 20 minutes.

17.4 Handling of Windowed Packages

17.4.1 Glass Erasing Window

Rubbing the glass erasing window of a windowed package with a plastic material or touching it with an electrically charged object can create a static charge on the window surface which may cause the chip to malfunction.

If the erasing window becomes charged, the charge can be neutralized by a short exposure to ultraviolet light. This returns the chip to its normal condition, but it also reduces the charge stored in the floating gates of the PROM, so it is recommended that the chip be reprogrammed afterward.

Accumulation of static charge on the window surface can be prevented by the following precautions:

1. When handling the package, ground yourself. Don't wear gloves. Avoid other possible sources of static charge.
2. Avoid friction between the glass window and plastic or other materials that tend to accumulate static charge.
3. Be careful when using cooling sprays, since they may have slight ion content.
4. Cover the window with an ultraviolet-shield label, preferably a label including a conductive material. Besides protecting the PROM contents from ultraviolet light, the label protects the chip by distributing static charge uniformly.

17.4.2 Handling after Programming

Fluorescent light and sunlight contain small amounts of ultraviolet, so prolonged exposure to these types of light can cause programmed data to invert. In addition, exposure to any type of intense light can induce photoelectric effects that may lead to chip malfunction. It is recommended that after programming the chip, you cover the erasing window with a light-proof label (such as an ultraviolet-shield label).

17.4.3 84-Pin LCC Package

A socket should always be used when the 84-pin LCC package is mounted on a printed-circuit board. Table 17.8 shows the recommended socket.

Table 17.8 Recommended Socket for Mounting 84-Pin LCC Package

Manufacturer	Code
Sumitomo 3-M	284-1273-00-1102J

Section 18 Power-Down State

18.1 Overview

The H8/3318 has a power-down state that greatly reduces power consumption by stopping some or all of the chip functions. The power-down state includes three modes:

1. Sleep mode—a software-triggered mode in which the CPU halts but the rest of the chip remains active
2. Software standby mode—a software-triggered mode in which the entire chip is inactive
3. Hardware standby mode—a hardware-triggered mode in which the entire chip is inactive

Table 18.1 lists the conditions for entering and leaving the power-down modes. It also indicates the status of the CPU, on-chip supporting modules, etc. in each power-down mode.

Table 18.1 Power-Down State

Mode	Entering Procedure	Clock	CPU	DTU	CPU Reg's.	Sup. Mod.	RAM	I/O Ports	Exiting Methods
Sleep mode	Execute SLEEP instruction	Run	Halt	Run	Held	Run	Held	Held	• Interrupt • $\overline{\text{RES}}$ • $\overline{\text{STBY}}$
Software standby mode	Set SSBY bit in SYSCR to 1, then execute SLEEP instruction	Halt	Halt	Halt	Held	Halt and initialized	Held	Held	• $\overline{\text{NMI}}$ • $\overline{\text{IRQ}_0}$–$\overline{\text{IRQ}_2}$, $\overline{\text{IRQ}_6}$ • $\overline{\text{RES}}$ • $\overline{\text{STBY}}$
Hardware standby mode	Set $\overline{\text{STBY}}$ pin to low level	Halt	Halt	Halt	Not held	Halt and initialized	Held	High impedance state	• $\overline{\text{STBY}}$ and $\overline{\text{RES}}$

Legend

SYSCR: System control register

SSBY: Software standby bit

18.1.1 System Control Register (SYSCR)

Four of the eight bits in the system control register (SYSCR) control the power-down state. These are bit 7 (SSBY) and bits 6 to 4 (STS2 to STS0). See table 18.2.

Table 18.2 System Control Register

Name	Abbreviation	R/W	Initial Value	Address
System control register	SYSCR	R/W	H'09	H'FFC4

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	XRST	NMIEG	DPME	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit 7—Software Standby (SSBY): This bit enables or disables the transition to software standby mode.

On recovery from software standby mode by an external interrupt, SSBY remains set to 1. To clear this bit, software must write a 0.

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

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the clock settling time when the chip recovers from software standby mode by an external interrupt. During the selected time, the clock oscillator runs but the CPU and on-chip supporting modules remain in standby. Set bits STS2 to STS0 according to the clock frequency to obtain a settling time of at least 8 ms. See table 18.3.

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

18.2 Sleep Mode

18.2.1 Transition to Sleep Mode

When the SSBY bit in the system control register is cleared to 0, execution of the SLEEP instruction causes a transition from the program execution state to sleep mode. After executing the SLEEP instruction, the CPU halts, but the contents of its internal registers remain unchanged. The on-chip supporting modules continue to operate normally.

18.2.2 Exit from Sleep Mode

The chip exits sleep mode when it receives an internal or external interrupt request, or a low input at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pin.

Exit by Interrupt: An interrupt releases sleep mode and starts the CPU's interrupt-handling sequence.

If an interrupt from an on-chip supporting module is disabled by the corresponding enable/disable bit in the module's control register, the interrupt cannot be requested, so it cannot wake the chip up. Similarly, the CPU cannot be awakened by an interrupt other than NMI if the I (interrupt mask) bit is set when the SLEEP instruction is executed.

Exit by $\overline{\text{RES}}$ pin: When the $\overline{\text{RES}}$ pin goes low, the chip exits from sleep mode to the reset state.

Exit by $\overline{\text{STBY}}$ pin: When the $\overline{\text{STBY}}$ pin goes low, the chip exits from sleep mode to hardware standby mode.

18.3 Software Standby Mode

18.3.1 Transition to Software Standby Mode

To enter software standby mode, set the standby bit (SSBY) in the system control register (SYSCR) to 1, then execute the SLEEP instruction.

In software standby mode, the system clock stops and chip functions halt, including both CPU functions and the functions of the on-chip supporting modules. Power consumption is reduced to an extremely low level. The on-chip supporting modules and their registers are reset to their initial states, but as long as a minimum necessary voltage supply is maintained, the contents of the CPU registers and on-chip RAM remain unchanged.

18.3.2 Exit from Software Standby Mode

The chip can be brought out of software standby mode by an input at one of six pins: $\overline{\text{NMI}}$, $\overline{\text{IRQ}_0}$, $\overline{\text{IRQ}_1}$, $\overline{\text{IRQ}_2}$, $\overline{\text{IRQ}_6}$, $\overline{\text{RES}}$, or $\overline{\text{STBY}}$.

Exit by Interrupt: When an NMI, IRQ_0, IRQ_1, IRQ_2, or IRQ_6 interrupt request signal is input, the clock oscillator begins operating. After the waiting time set in bits STS2 to STS0 of SYSCR, a stable clock is supplied to the entire chip, software standby mode is released, and interrupt exception-handling begins. IRQ_3 to IRQ_5, IRQ_7 interrupts should be disabled before the transition to software standby (clear IRQ3E to IRQ5E, IRQ7E to 0).

Exit by $\overline{\text{RES}}$ Pin: When the $\overline{\text{RES}}$ input goes low, the clock oscillator begins operating. When $\overline{\text{RES}}$ is brought to the high level (after allowing time for the clock oscillator to settle), the CPU starts reset exception handling. Be sure to hold $\overline{\text{RES}}$ low long enough for clock oscillation to stabilize.

Exit by $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ input goes low, the chip exits from software standby mode to hardware standby mode.

18.3.3 Clock Settling Time for Exit from Software Standby Mode

Set bits STS2 to STS0 in SYSCR as follows:

- Crystal oscillator

  Set STS2 to STS0 for a settling time of at least 8 ms. Table 18.3 lists the settling times selected by these bits at several clock frequencies.

- External clock

  The STS bits can be set to any value. Normally, the minimum value (STS2 = STS1 = STS0 = 0) is recommended.

Table 18.3 Times Set by Standby Timer Select Bits (Unit: ms)

				System Clock Frequency (MHz)								
STS2	**STS1**	**STS0**	**Settling Time (States)**	**16**	**12**	**10**	**8**	**6**	**4**	**2**	**1**	**0.5**
0	0	0	8,192	0.51	0.65	0.8	1.0	1.3	2.0	4.1	**8.2**	**16.4**
0	0	1	16,384	1.0	1.3	1.6	2.0	2.7	4.1	**8.2**	16.4	32.8
0	1	0	32,768	2.0	2.7	3.3	4.1	5.5	**8.2**	16.4	32.8	65.5
0	1	1	65,536	4.1	5.5	6.6	**8.2**	**10.9**	16.4	32.8	65.5	131.1
1	0	—	131,072	**8.2**	**10.9**	**13.1**	16.4	21.8	32.8	65.5	131.1	262.1

Note: Recommended values are printed in boldface.

18.3.4 Sample Application of Software Standby Mode

In this example the chip enters the software standby mode when $\overline{\text{NMI}}$ goes low and exits when $\overline{\text{NMI}}$ goes high, as shown in figure 18.1.

The NMI edge bit (NMIEG) in the system control register is originally cleared to 0, selecting the falling edge. When $\overline{\text{NMI}}$ goes low, the $\overline{\text{NMI}}$ interrupt handling routine sets NMIEG to 1 (selecting the rising edge), sets SSBY to 1, then executes the SLEEP instruction. The chip enters software standby mode. It recovers from software standby mode on the next rising edge of $\overline{\text{NMI}}$.

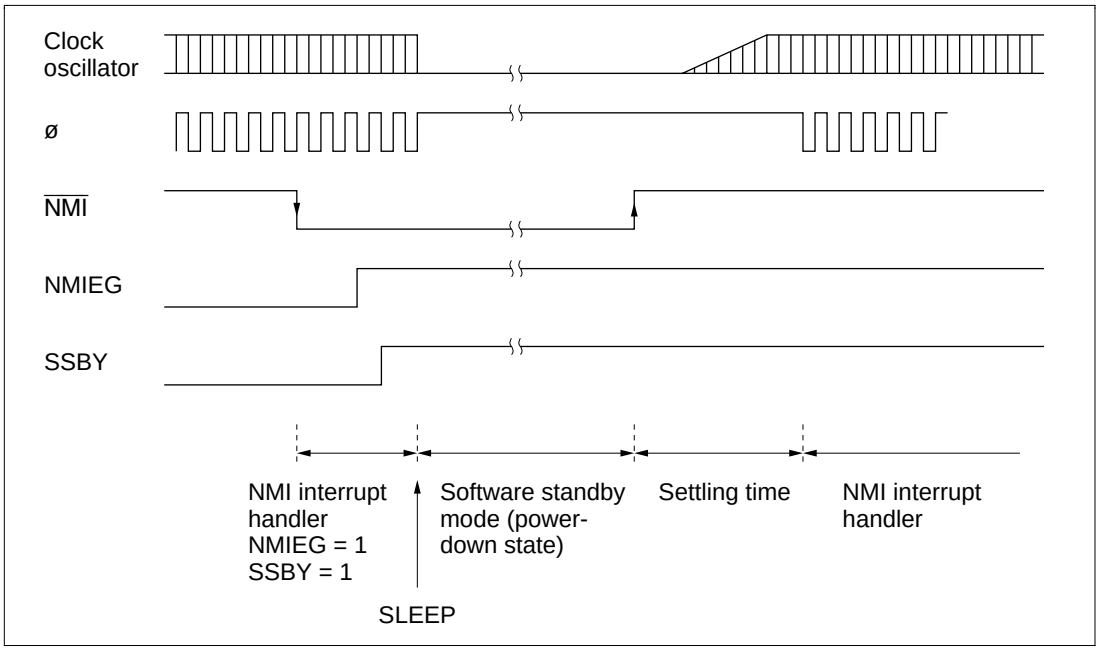


Figure 18.1 NMI Timing in Software Standby Mode

18.3.5 Application Note

The I/O ports retain their current states in software standby mode. If a port is in the high output state, the current dissipation caused by the high output current is not reduced.

18.4 Hardware Standby Mode

18.4.1 Transition to Hardware Standby Mode

Regardless of its current state, the chip enters hardware standby mode whenever the $\overline{\text{STBY}}$ pin goes low.

Hardware standby mode reduces power consumption drastically by halting the CPU, stopping all the functions of the on-chip supporting modules, and placing I/O ports in the high-impedance state. The registers of the on-chip supporting modules are reset to their initial values. Only the on-chip RAM is held unchanged, provided the minimum necessary voltage supply is maintained.

Notes: 1. The RAME bit in the system control register should be cleared to 0 before the $\overline{\text{STBY}}$ pin goes low.
2. Do not change the inputs at the mode pins (MD_1, MD_0) during hardware standby mode. Be particularly careful not to let both mode pins go low in hardware standby mode, since that places the chip in PROM mode and increases current dissipation.

18.4.2 Recovery from Hardware Standby Mode

Recovery from hardware standby mode requires inputs at both the $\overline{\text{STBY}}$ and $\overline{\text{RES}}$ pins. When the $\overline{\text{STBY}}$ pin goes high, the clock oscillator begins running. The $\overline{\text{RES}}$ pin should be low at this time and should be held low long enough for the clock to stabilize. When the $\overline{\text{RES}}$ pin changes from low to high, the reset sequence is executed and the chip returns to the program execution state.

18.4.3 Timing Relationships

Figure 18.2 shows the timing relationships in hardware standby mode.

In the sequence shown, first $\overline{\text{RES}}$ goes low, then $\overline{\text{STBY}}$ goes low, at which point the chip enters hardware standby mode. To recover, first $\overline{\text{STBY}}$ goes high, then after the clock settling time, $\overline{\text{RES}}$ goes high.

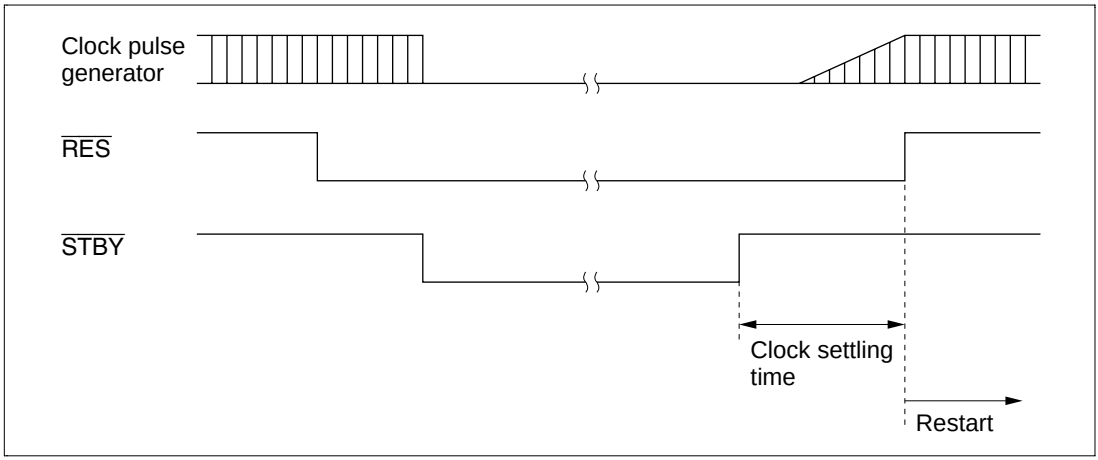


Figure 18.2 Hardware Standby Mode Timing

Section 19 Electrical Specifications

19.1 Absolute Maximum Ratings

Table 19.1 lists the absolute maximum ratings.

Table 19.1 Absolute Maximum Ratings

Item		Symbol	Rating	Unit
Supply voltage		V_{CC}	–0.3 to +7.0	V
Programming voltage		V_{PP}	–0.3 to +13.5	V
Input voltage	Ports 1–6, 8, 9	V_{in}	–0.3 to V_{CC} + 0.3	V
	Port 7	V_{in}	–0.3 to AV_{CC} + 0.3	V
Analog supply voltage		AV_{CC}	–0.3 to +7.0	V
Analog input voltage		V_{AN}	–0.3 to AV_{CC} + 0.3	V
Operating temperature		T_{opr}	Regular specifications: –20 to +75	°C
			Wide-range specifications: –40 to +85	°C
Storage temperature		T_{stg}	–55 to +125	°C

Note: Exceeding the absolute maximum ratings shown in table 19.1 can permanently destroy the chip.

19.2 Electrical Characteristics

19.2.1 DC Characteristics

Tables 19.2, 19.3, and 19.4 list the DC characteristics of the 5 V, 4 V, and 3 V versions, respectively, and tables 19.5 and 19.6 give the allowable current output values of the 5 V and 4 V, and 3 V versions, respectively.

Table 19.2 DC Characteristics (5 V Version)

Conditions: V_{CC} = 5.0 V ±10%, AV_{CC} = 5.0 V ±10%*[1], V_{SS} = AV_{SS} = 0 V, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage (1)	$P8_6$–$P8_0$*[2], $P9_7$, $P9_5$–$P9_0$*[2], FTCI, FTIA, FTIB, FTIC, FTID, FTI, $TMCI_0$, $TMCI_1$, $TMRI_0$, $TMRI_1$, $\overline{IRQ}_6$, $\overline{IRQ}_7$, $ETMCI_0$, $ETMCI_1$, $ETMRI_0$, $ETMRI_1$	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
Input high voltage (2)	$\overline{RES}$, $\overline{STBY}$, $\overline{NMI}$, MD_1, MD_0, EXTAL	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	$P7_7$ – $P7_0$		2.0	—	AV_{CC} + 0.3	V	
Input high voltage	Input pins other than (1) and (2)	V_{IH}	2.0	—	V_{CC} + 0.3	V	
Input low voltage (3)	$\overline{RES}$, $\overline{STBY}$, MD_1, MD_0	V_{IL}	–0.3	—	0.5	V	
Input low voltage	Input pins other than (1) and (3)	V_{IL}	–0.3	—	0.8	V	
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1.0 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	$P1_7$–$P1_0$, $P2_7$–$P2_0$		—	—	1.0	V	I_{OL} = 10.0 mA
Input leakage current	$\overline{RES}$	$\|I_{in}\|$	—	—	10.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
	$\overline{STBY}$, $\overline{NMI}$, MD_1, MD_0		—	—	1.0	µA	
	$P7_7$ – $P7_0$		—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} – 0.5 V

Table 19.2 DC Characteristics (5 V Version) (cont)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Leakage current in 3-state (off state)	Ports 1, 2, 3, 4, 5, 6, 8, 9	$\|I_{TSI}\|$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1, 2, 3	$-I_P$	30	—	250	µA	V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	60	pF	V_{in} = 0 V, f = 1 MHz, Ta = 25°C
	$\overline{NMI}$		—	—	50	pF	
	All input pins except $\overline{RES}$ and $\overline{NMI}$		—	—	15	pF	
Current dissipation*[3]	Normal operation	I_{CC}	—	27	45	mA	f = 12 MHz
			—	36	60	mA	f = 16 MHz
	Sleep mode		—	18	30	mA	f = 12 MHz
			—	24	40	mA	f = 16 MHz
	Standby modes*[4]		—	0.01	5.0	µA	Ta ≤ 50°C
			—	—	20.0	µA	50°C < Ta
Analog supply current	During A/D conversion	AI_{CC}	—	2.0	5.0	mA	
	Waiting		—	0.01	5.0	µA	AV_{CC} = 2.0 V to 5.5 V
Analog supply voltage*[1]		AV_{CC}	4.5	—	5.5	V	During operation
			2.0	—	5.5	V	During wait state or when not in use
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. Connect AV_{CC} to the power supply (V_{CC}) and apply between 2.0 V and 5.5 V even when the A/D converter is not used.

2. $P8_6$ to $P8_0$, $P9_7$, and $P9_5$ to $P9_0$ include peripheral function inputs multiplexed with these pins.

3. Current dissipation values assume that $V_{IH\ min} = V_{CC} – 0.5$ V, $V_{IL\ max} = 0.5$ V, all output pins are in the no-load state, and all input pull-up transistors are off.

4. For these values it is assumed that $V_{RAM} \leq V_{CC} < 4.5$ V and $V_{IH\ min} = V_{CC} \times 0.9$, $V_{IL\ max} = 0.3$ V.

Table 19.3 DC Characteristics (4 V Version)

Conditions: V_{CC} = 4.0 V to 5.5 V, AV_{CC} = 4.0 V to 5.5 V*[1], V_{SS} = AV_{SS} = 0 V, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage (1)	$P8_6$ – $P8_0$*[2], $P9_7$, $P9_5$ – $P9_0$*[2], FTCI, FTIA, FTIB, FTIC, FTID, FTI, $TMCI_0$, $TMCI_1$, $TMRI_0$, $TMRI_1$, $\overline{IRQ}_6$, $\overline{IRQ}_7$, $ETMCI_0$, $ETMCI_1$, $ETMRI_0$, $ETMRI_1$	V_T^-	1.0	—	—	V	V_{CC} = 4.5 V to 5.5 V
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.4	—	—	V	
		V_T^-	0.8	—	—	V	V_{CC} = 4.0 V to 4.5 V
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.3	—	—	V	
Input high voltage (2)	$\overline{RES}$, $\overline{STBY}$, $\overline{NMI}$, MD_1, MD_0, EXTAL	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	$P7_7$ – $P7_0$		2.0	—	AV_{CC} + 0.3	V	
Input high voltage	Input pins other than (1) and (2)	V_{IH}	2.0	—	V_{CC} + 0.3	V	
Input low voltage (3)	$\overline{RES}$, $\overline{STBY}$, MD_1, MD_0	V_{IL}	–0.3	—	0.5	V	
Input low voltage	Input pins other than (1) and (3)	V_{IL}	–0.3	—	0.8	V	V_{CC} = 4.5 V to 5.5 V
			–0.3	—	0.6	V	V_{CC} = 4.0 V to 4.5 V
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—	V	I_{OH} = –1.0 mA, V_{CC} = 4.5 V to 5.5 V
			2.8	—	—	V	I_{OH} = –1.0 mA, V_{CC} = 4.0 V to 4.5 V
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	$P1_7$ – $P1_0$, $P2_7$ – $P2_0$		—	—	1.0	V	I_{OL} = 10.0 mA

Table 19.3 DC Characteristics (4 V Version) (cont)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input leakage current	$\overline{RES}$	$\|I_{in}\|$	—	—	10.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
	$\overline{STBY}$, $\overline{NMI}$, MD_1, MD_0		—	—	1.0	µA	
	$P7_7$ – $P7_0$		—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} – 0.5 V
Leakage current in 3-state (off state)	Ports 1, 2, 3, 4, 5, 6, 8, 9	$\|I_{TSI}\|$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1, 2, 3	$-I_P$	30	—	250	µA	V_{in} = 0 V, V_{CC} = 4.5 V to 5.5 V
			20	—	200	µA	V_{in} = 0 V, V_{CC} = 4.0 V to 4.5 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	60	pF	V_{in} = 0 V, f = 1 MHz, Ta = 25°C
	$\overline{NMI}$		—	—	50	pF	
	All input pins except $\overline{RES}$ and $\overline{NMI}$		—	—	15	pF	
Current dissipation*[3]	Normal operation	I_{CC}	—	27	45	mA	f = 12 MHz
			—	36	60	mA	f = 16 MHz, V_{CC} = 4.5 V to 5.5 V
	Sleep mode		—	18	30	mA	f = 12 MHz
			—	24	40	mA	f = 16 MHz, V_{CC} = 4.5 V to 5.5 V
	Standby modes*[4]		—	0.01	5.0	µA	Ta ≤ 50°C
			—	—	20.0	µA	50°C < Ta

Table 19.3 DC Characteristics (4 V Version) (cont)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Analog supply current	During A/D conversion	AI_{CC}	—	2.0	5.0	mA	
	Waiting		—	0.01	5.0	µA	AV_{CC} = 2.0 V to 5.5 V
Analog supply voltage*[1]		AV_{CC}	4.0	—	5.5	V	During operation
			2.0	—	5.5	V	During wait state or when not in use
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: 1. Connect AV_{CC} to the power supply (V_{CC}) and apply between 2.0 V and 5.5 V even when the A/D converter is not used.

2. $P8_6$ to $P8_0$, $P9_7$, and $P9_5$ to $P9_0$ include peripheral function inputs multiplexed with these pins.

3. Current dissipation values assume that $V_{IH\ min} = V_{CC} - 0.5$ V, $V_{IL\ max} = 0.5$ V, all output pins are in the no-load state, and all input pull-up transistors are off.

4. For these values it is assumed that $V_{RAM} \leq V_{CC} < 4.0$ V and $V_{IH\ min} = V_{CC} \times 0.9$, $V_{IL\ max} = 0.3$ V.

Table 19.4 DC Characteristics (3 V Version)

Conditions: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V*[1], V_{SS} = AV_{SS} = 0 V, Ta = –20 to 75°C

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage (1)	$P8_6$–$P8_0$*[2], $P9_7$, $P9_5$–$P9_0$*[2], FTCI, FTIA, FTIB, FTIC, FTID, FTI, $TMCI_0$, $TMCI_1$, $TMRI_0$, $TMRI_1$, $\overline{IRQ}_6$, $\overline{IRQ}_7$, $ETMCI_0$, $ETMCI_1$, $ETMRI_0$, $ETMRI_1$	V_T^-	$V_{CC} \times 0.15$	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$	V	
		$V_T^+ - V_T^-$	0.2	—	—	V	
Input high voltage (2)	$\overline{RES}$, $\overline{STBY}$, $\overline{NMI}$, MD_1, MD_0, EXTAL	V_{IH}	$V_{CC} \times 0.9$	—	V_{CC} + 0.3	V	
	$P7_7$ – $P7_0$		$V_{CC} \times 0.7$	—	AV_{CC} + 0.3	V	
Input high voltage	Input pins other than (1) and (2)	V_{IH}	$V_{CC} \times 0.7$	—	V_{CC} + 0.3	V	
Input low voltage (3)	$\overline{RES}$, $\overline{STBY}$, MD_1, MD_0	V_{IL}	–0.3	—	$V_{CC} \times 0.1$	V	
Input low voltage	Input pins other than (1) and (3)	V_{IL}	–0.3	—	$V_{CC} \times 0.15$	V	
Output high voltage	All output pins	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			V_{CC} – 1.0	—	—	V	I_{OH} = –1.0 mA
Output low voltage	All output pins	V_{OL}	—	—	0.4	V	I_{OL} = 0.8 mA
	$P1_7$ – $P1_0$, $P2_7$ – $P2_0$		—	—	0.4	V	I_{OL} = 1.6 mA
Input leakage current	$\overline{RES}$	$\|I_{in}\|$	—	—	10.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
	$\overline{STBY}$, $\overline{NMI}$, MD_1, MD_0		—	—	1.0	µA	
	$P7_7$ – $P7_0$		—	—	1.0	µA	V_{in} = 0.5 V to AV_{CC} – 0.5 V

Table 19.4 DC Characteristics (3 V Version) (cont)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Leakage current in 3-state (off state)	Ports 1, 2, 3, 4, 5, 6, 8, 9	$\|I_{TSI}\|$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 0.5 V
Input pull-up MOS current	Ports 1, 2, 3	$-I_P$	3	—	120	µA	V_{in} = 0 V, V_{CC} = 2.7 V to 4.0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	60	pF	V_{in} = 0 V, f = 1 MHz, Ta = 25°C
	$\overline{NMI}$		—	—	50	pF	
	All input pins except $\overline{RES}$ and $\overline{NMI}$		—	—	15	pF	
Current dissipation*[3]	Normal operation	I_{CC}	—	7	—	mA	f = 6 MHz, V_{CC} = 2.7 V to 3.6 V
			—	12	22	mA	f = 10 MHz, V_{CC} = 2.7 V to 3.6 V
			—	25	—	mA	f = 10 MHz, V_{CC} = 4.0 V to 5.5 V
	Sleep mode		—	5	—	mA	f = 6 MHz, V_{CC} = 2.7 V to 3.6 V
			—	9	16	mA	f = 10 MHz, V_{CC} = 2.7 V to 3.6 V
			—	18	—	mA	f = 10 MHz, V_{CC} = 4.0 V to 5.5 V
	Standby modes*[4]		—	0.01	5.0	µA	Ta ≤ 50°C
			—	—	20.0	µA	50°C < Ta
Analog supply current	During A/D conversion	AI_{CC}	—	2.0	5.0	mA	
	Waiting		—	0.01	5.0	µA	AV_{CC} = 2.0 V to 5.5 V

Table 19.4 DC Characteristics (3 V Version) (cont)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Analog supply voltage*[1]	AV_{CC}	2.7	—	5.5	V	During operation
		2.0	—	5.5	V	During wait state or when not in use
RAM standby voltage	V_{RAM}	2.0	—	—	V	

Notes: 1. Connect AV_{CC} to the power supply (V_{CC}) and apply between 2.0 V and 5.5 V even when the A/D converter is not used.

2. $P8_6$ to $P8_0$, $P9_7$, and $P9_5$ to $P9_0$ include peripheral function inputs multiplexed with these pins.

3. Current dissipation values assume that $V_{IH\ min} = V_{CC} - 0.5$ V, $V_{IL\ max} = 0.5$ V, all output pins are in the no-load state, and all input pull-up transistors are off.

4. For these values it is assumed that $V_{RAM} \leq V_{CC} < 2.7$ V and $V_{IH\ min} = V_{CC} \times 0.9$, $V_{IL\ max} = 0.3$ V.

Table 19.5 Allowable Output Current Values (5 V and 4 V Versions)

Conditions: V_{CC} = 4.0 V to 5.5 V, AV_{CC} = 4.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit
Allowable output low current (per pin)	Ports 1 and 2	I_{OL}	—	—	10	mA
	Other output pins		—	—	2	mA
Allowable output low current (total)	Ports 1 and 2, total	ΣI_{OL}	—	—	80	mA
	Total of all output		—	—	120	mA
Allowable output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	mA
Allowable output high current (total)	Total of all output	$\Sigma -I_{OH}$	—	—	40	mA

Table 19.6 Allowable Output Current Values (3 V Version)

Conditions: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, Ta = –20 to 75°C

Item		Symbol	Min	Typ	Max	Unit
Allowable output low current (per pin)	Ports 1 and 2	I_{OL}	—	—	2	mA
	Other output pins		—	—	1	mA
Allowable output low current (total)	Ports 1 and 2, total	ΣI_{OL}	—	—	40	mA
	Total of all output		—	—	60	mA
Allowable output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	mA
Allowable output high current (total)	Total of all output	$\Sigma -I_{OH}$	—	—	30	mA

Notes on Use: To ensure the reliability of the chip, the output current values shown in tables 19.5 and 19.6 should not be exceeded. When directly driving a Darlington transistor or LED, in particular, a current-limiting resistance must be inserted (see figures 19.1 and 19.2).

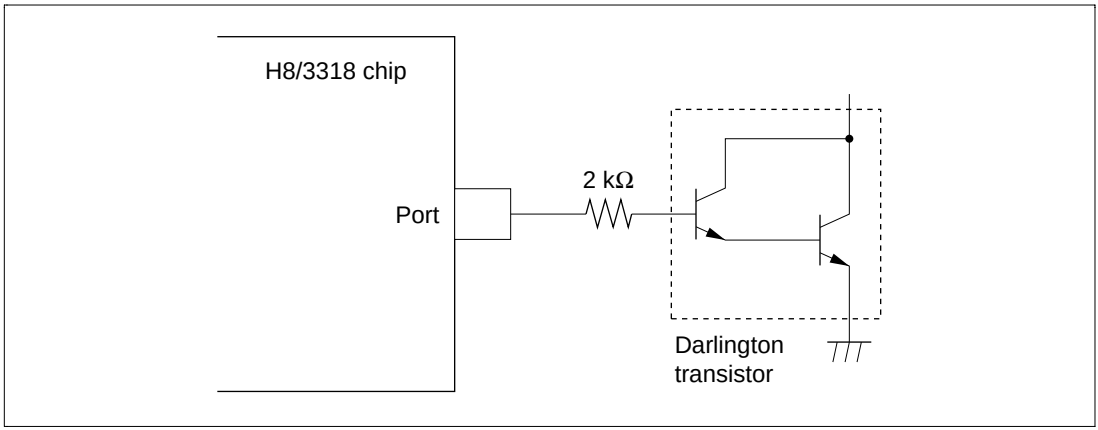


Figure 19.1 Example of Circuit for Driving a Darlington Transistor

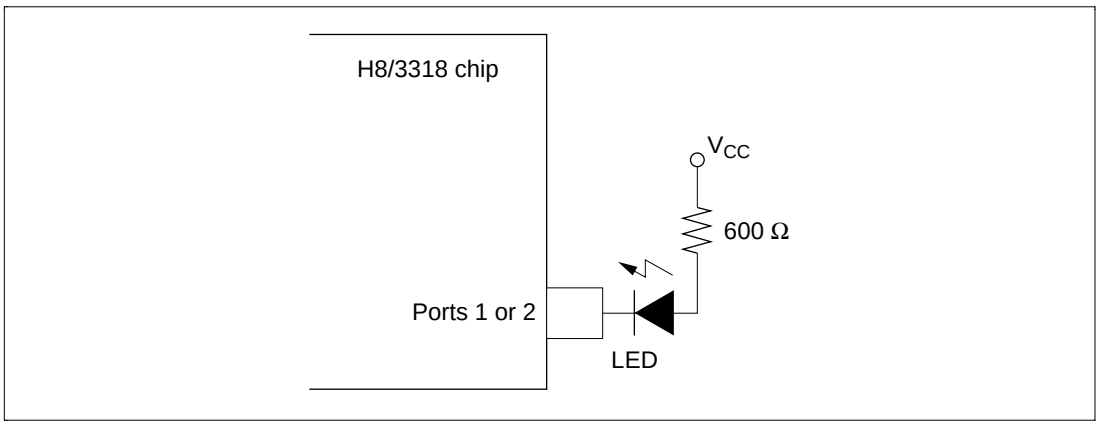


Figure 19.2 Example of Circuit for Driving an LED

19.2.2 AC Characteristics

The AC characteristics are listed in three tables. Bus timing parameters are given in table 19.7, control signal timing parameters in table 19.8, timing parameters of the on-chip supporting modules in table 19.9, and external clock output delay time in table 19.10.

Table 19.7 Bus Timing

Conditions A: V_{CC} = 4.5 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Conditions B: V_{CC} = 4.0 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Conditions C: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C

		Conditions C		Conditions B		Conditions A			
		10 MHz		12 MHz		16 MHz			Test
Item	Symbol	Min	Max	Min	Max	Min	Max	Unit	Conditions
Clock cycle time	t_{cyc}	100	500	83.3	500	62.5	500	ns	Fig. 19.4
Clock pulse width low	t_{CL}	30	—	30	—	20	—	ns	
Clock pulse width high	t_{CH}	30	—	30	—	20	—	ns	
Clock rise time	t_{Cr}	—	20	—	10	—	10	ns	
Clock fall time	t_{Cf}	—	20	—	10	—	10	ns	
Address delay time	t_{AD}	—	50	—	35	—	30	ns	
Address hold time	t_{AH}	20	—	15	—	10	—	ns	
Address strobe delay time	t_{ASD}	—	50	—	35	—	30	ns	
Write strobe delay time	t_{WSD}	—	50	—	35	—	30	ns	
Strobe delay time	t_{SD}	—	50	—	35	—	30	ns	
Write strobe pulse width*	t_{WSW}	110	—	90	—	60	—	ns	
Address setup time 1*	t_{AS1}	15	—	10	—	10	—	ns	
Address setup time 2*	t_{AS2}	65	—	50	—	40	—	ns	
Read data setup time	t_{RDS}	35	—	20	—	20	—	ns	
Read data hold time*	t_{RDH}	0	—	0	—	0	—	ns	
Read data access time*	t_{ACC}	—	170	—	160	—	110	ns	
Write data delay time	t_{WDD}	—	75	—	60	—	60	ns	
Write data setup time	t_{WDS}	5	—	5	—	5	—	ns	
Write data hold time	t_{WDH}	20	—	20	—	20	—	ns	
Wait setup time	t_{WTS}	40	—	35	—	30	—	ns	Fig. 19.5
Wait hold time	t_{WTH}	10	—	10	—	10	—	ns	

Note: * Values at maximum operating frequency

Table 19.8 Control Signal Timing

Conditions A: V_{CC} = 4.5 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Conditions B: V_{CC} = 4.0 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Conditions C: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C

		Conditions C		Conditions B		Conditions A			
		10 MHz		12 MHz		16 MHz			Test
Item	Symbol	Min	Max	Min	Max	Min	Max	Unit	Conditions
$\overline{RES}$ setup time	t_{RESS}	300	—	200	—	200	—	ns	Fig. 19.6
$\overline{RES}$ pulse width	t_{RESW}	10	—	10	—	10	—	t_{cyc}	
$\overline{NMI}$ setup time ($\overline{NMI}$, $\overline{IRQ_0}$ to $\overline{IRQ_7}$)	t_{NMIS}	300	—	150	—	150	—	ns	Fig. 19.7
$\overline{NMI}$ hold time ($\overline{NMI}$, $\overline{IRQ_0}$ to $\overline{IRQ_7}$)	t_{NMIH}	10	—	10	—	10	—	ns	
Interrupt pulse width for recovery from software standby mode ($\overline{NMI}$, $\overline{IRQ_0}$ to $\overline{IRQ_2}$, $\overline{IRQ_6}$)	t_{NMIW}	300	—	200	—	200	—	ns	
Crystal oscillator settling time (power-on reset)	t_{OSC1}	20	—	20	—	20	—	ms	Fig. 19.8
Crystal oscillator settling time (software standby)	t_{OSC2}	8	—	8	—	8	—	ms	Fig. 19.9

- **Test Conditions for AC Characteristics**

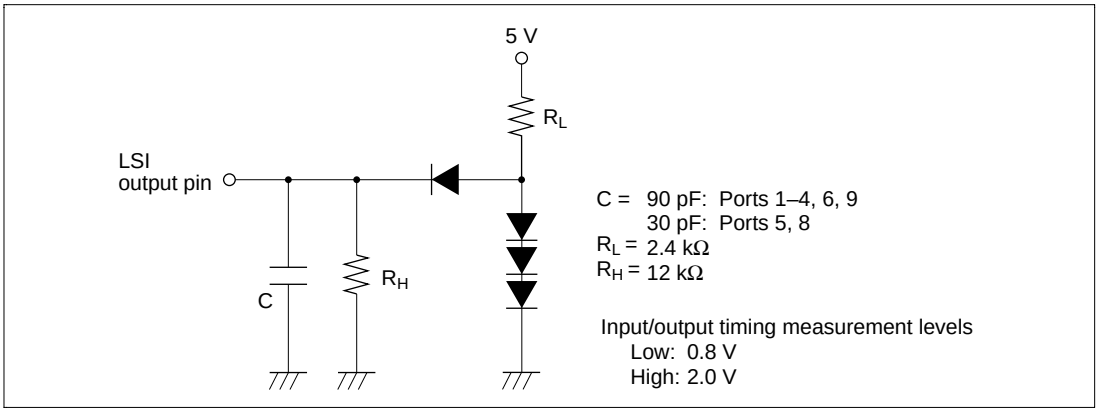


Figure 19.3 Output Load Circuit

Table 19.9 Timing Conditions of On-Chip Supporting Modules

Conditions A: V_{CC} = 4.5 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Conditions B: V_{CC} = 4.0 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Conditions C: V_{CC} = 2.7 V to 5.5 V, V_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C

			Conditions C		Conditions B		Conditions A			
			10 MHz		12 MHz		16 MHz			Test
Item		Symbol	Min	Max	Min	Max	Min	Max	Unit	Conditions
FRT	Timer output delay time	t_{FTOD}	—	150	—	100	—	100	ns	Fig. 19.10
	Timer input setup time	t_{FTIS}	80	—	50	—	50	—	ns	
	Timer clock input setup time	t_{FTCS}	80	—	50	—	50	—	ns	Fig. 19.11
	Timer clock pulse width	t_{FTCWH} t_{FTCWL}	1.5	—	1.5	—	1.5	—	t_{cyc}	
TMR	Timer output delay time	t_{TMOD}	—	150	—	100	—	100	ns	Fig. 19.12
	Timer reset input setup time	t_{TMRS}	80	—	50	—	50	—	ns	Fig. 19.14
	Timer clock input setup time	t_{TMCS}	80	—	50	—	50	—	ns	Fig. 19.13
	Timer clock pulse width (single edge)	t_{TMCWH}	1.5	—	1.5	—	1.5	—	t_{cyc}	
	Timer clock pulse width (both edges)	t_{TMCWL}	2.5	—	2.5	—	2.5	—	t_{cyc}	

Table 19.9 Timing Conditions of On-Chip Supporting Modules (cont)

Item		Symbol	Conditions C 10 MHz Min	Conditions C 10 MHz Max	Conditions B 12 MHz Min	Conditions B 12 MHz Max	Conditions A 16 MHz Min	Conditions A 16 MHz Max	Unit	Test Conditions
SCI	Input clock cycle (Async)	t_{scyc}	4	—	4	—	4	—	t_{cyc}	Fig. 19.15
	Input clock cycle (Sync)	t_{scyc}	6	—	6	—	6	—	t_{cyc}	
	Transmit data delay time (Sync)	t_{TXD}	—	200	—	100	—	100	ns	
	Receive data setup time (Sync)	t_{RXS}	150	—	100	—	100	—	ns	
	Receive data hold time (Sync)	t_{RXH}	150	—	100	—	100	—	ns	
	Input clock pulse width	t_{SCKW}	0.4	0.6	0.4	0.6	0.4	0.6	t_{scyc}	Fig. 19.16
Ports	Output data delay time	t_{PWD}	—	150	—	100	—	100	ns	Fig. 19.17
	Input data setup time	t_{PRS}	80	—	50	—	50	—	ns	
	Input data hold time	t_{PRH}	80	—	50	—	50	—	ns	
TPC	Output data delay time	t_{TPD}	—	100	—	100	—	100	ns	Fig. 19.18
DPRAM read/write cycle	Address hold time	t_{RSH}	10	—	10	—	10	—	ns	Fig. 19.19, Fig. 19.20
	Chip select hold time	t_{CSH}	10	—	10	—	10	—	ns	
	Address setup time	t_{RSS}	10	—	10	—	10	—	ns	
	Chip select setup time	t_{CSS}	10	—	10	—	10	—	ns	
DPRAM write cycle	Write pulse width	t_{DWP}	65	—	65	—	65	—	ns	Fig. 19.19
	Write data setup time	t_{DDW}	35	—	35	—	35	—	ns	
	Write data hold time	t_{DDH}	20	—	20	—	20	—	ns	

Table 19.9 Timing Conditions of On-Chip Supporting Modules (cont)

Item		Symbol	Conditions C 10 MHz Min	Conditions C 10 MHz Max	Conditions B 12 MHz Min	Conditions B 12 MHz Max	Conditions A 16 MHz Min	Conditions A 16 MHz Max	Unit	Test Conditions
DPRAM read cycle	Access time	t_{DAA}	—	100	—	85	—	85	ns	Fig. 19.20
	Read data delay time	t_{DOE}	—	85	—	85	—	85	ns	
	Chip select access time	t_{DACS}	—	100	—	85	—	85	ns	
	$\overline{CS}$ output floating time	t_{DCHZ}	0	50	0	50	0	50	ns	
	$\overline{OE}$ output floating time	t_{DOHZ}	0	50	0	50	0	50	ns	
DPRAM $\overline{WRQ}$ output	Wait request delay time 1	t_{WTD1}	—	80	—	50	—	50	ns	Fig. 19.21
	Wait request delay time 2	t_{WTD2}	20	—	20	—	20	—	ns	
Byte access interval in DPRAM bound buffer mode	Byte access interval 0	t_{BYTE0}	2	—	2	—	2	—	t_{cyc}	Fig. 19.22
	Byte access interval 1	t_{BYTE1}	20	—	20	—	20	—	ns	
	Byte access interval 2	t_{BYTE2}	9.5	—	9.5	—	9.5	—	t_{cyc}	
Byte access interval in DPRAM buffer query mode	Byte access interval 3	t_{BYTE3}	11	—	11	—	11	—	t_{cyc}	
Parallel handshake interface receive	$\overline{RDY}$ output delay time 1	t_{HRDYD1}	—	80	—	80	—	80	ns	Fig. 19.23
	$\overline{RDY}$ output delay time 2	t_{HRDYD2}	—	80	—	80	—	80	ns	
	Receive data setup time	t_{TDS}	20	—	20	—	20	—	ns	
	Receive data hold time	t_{TDH}	25	—	25	—	25	—	ns	
	Handshake pulse width	t_{HDWP}	300	—	300	—	300	—	ns	

Table 19.9 Timing Conditions of On-Chip Supporting Modules (cont)

Item		Symbol	Conditions C 10 MHz Min	Conditions C 10 MHz Max	Conditions B 12 MHz Min	Conditions B 12 MHz Max	Conditions A 16 MHz Min	Conditions A 16 MHz Max	Unit	Test Conditions
Parallel handshake interface transmit	$\overline{WRQ}$ output delay time 1	t_{HWRQD1}	—	80	—	80	—	80	ns	Fig. 19.24
	$\overline{WRQ}$ output delay time 2	t_{HWRQD2}	—	80	—	80	—	80	ns	

Note: Values at maximum operating frequency

Table 19.10 External Clock Output Stabilization Delay Time

Conditions: V_{CC} = 2.7 to 5.5 V, AV_{CC} = 2.7 to 5.5 V, V_{SS} = AV_{SS} = 0 V

Item	Symbol	Min	Max	Unit	Notes
External clock output stabilization delay time	t_{DEXT}*	500	—	µs	Fig. 19.25

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

19.2.3 A/D Converter Characteristics

Table 19.11 lists the characteristics of the on-chip A/D converter.

Table 19.11 A/D Converter Characteristics

Conditions A: V_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Conditions B: V_{CC} = 4.0 V to 5.5 V, AV_{CC} = 4.0 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C (regular specifications), Ta = –40 to 85°C (wide-range specifications)

Conditions C: V_{CC} = 2.7 V to 5.5 V, AV_{CC} = 2.7 V to 5.5 V, V_{SS} = AV_{SS} = 0 V, ø = 2 MHz to maximum operating frequency, Ta = –20 to 75°C

	Conditions C			Conditions B			Conditions A			
	10 MHz			12 MHz			16 MHz			
Item	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	10	10	10	10	10	10	Bits
Conversion time (single mode)*	—	—	13.4	—	—	11.2	—	—	8.4	µs
Analog input capacitance	—	—	20	—	—	20	—	—	20	pF
Allowable signal source impedance	—	—	5	—	—	10	—	—	10	kΩ
Nonlinearity error	—	—	±6.0	—	—	±3.0	—	—	±3.0	LSB
Offset error	—	—	±4.0	—	—	±3.5	—	—	±3.5	LSB
Full-scale error	—	—	±4.0	—	—	±3.5	—	—	±3.5	LSB
Quantizing error	—	—	±0.5	—	—	±0.5	—	—	±0.5	LSB
Absolute accuracy	—	—	±8.0	—	—	±4.0	—	—	±4.0	LSB

Note: * Values at maximum operating frequency

19.3 MCU Operational Timing

This section provides the following timing charts:

19.3.1 Bus Timing

(1) Basic Bus Cycle (without Wait States) in Expanded Modes

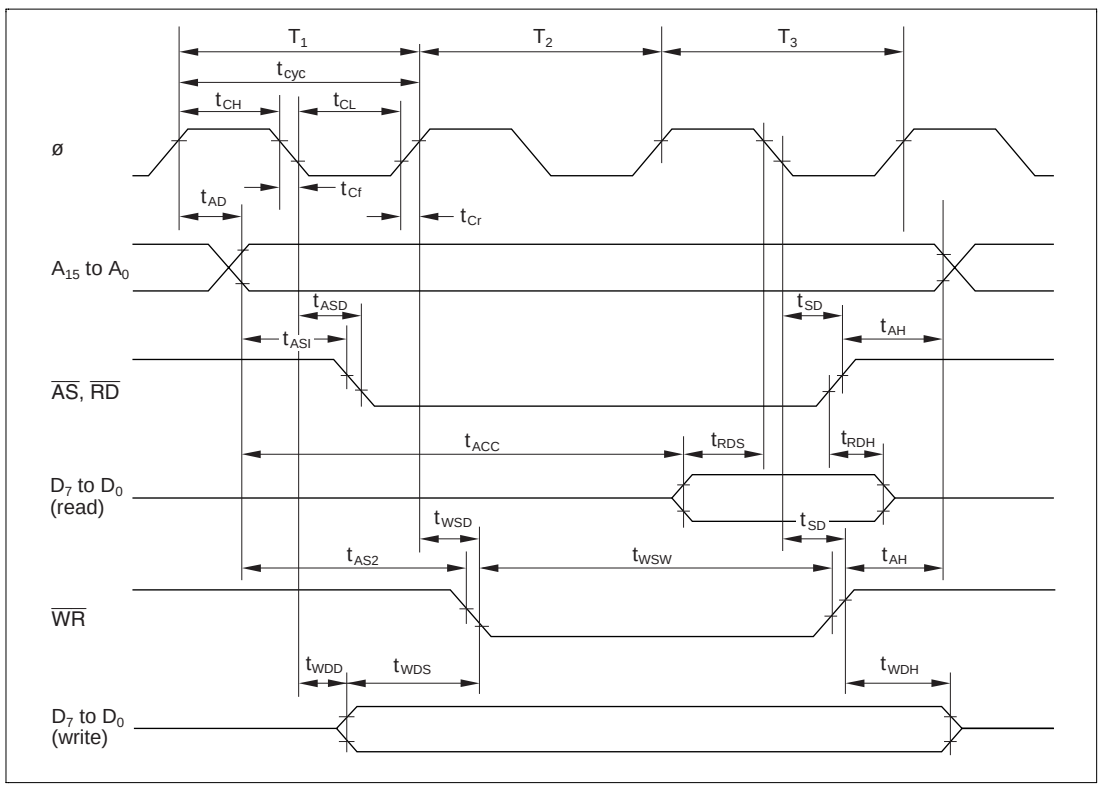


Figure 19.4 Basic Bus Cycle (without Wait States) in Expanded Modes

(2) Basic Bus Cycle (with 1 Wait State) in Expanded Modes

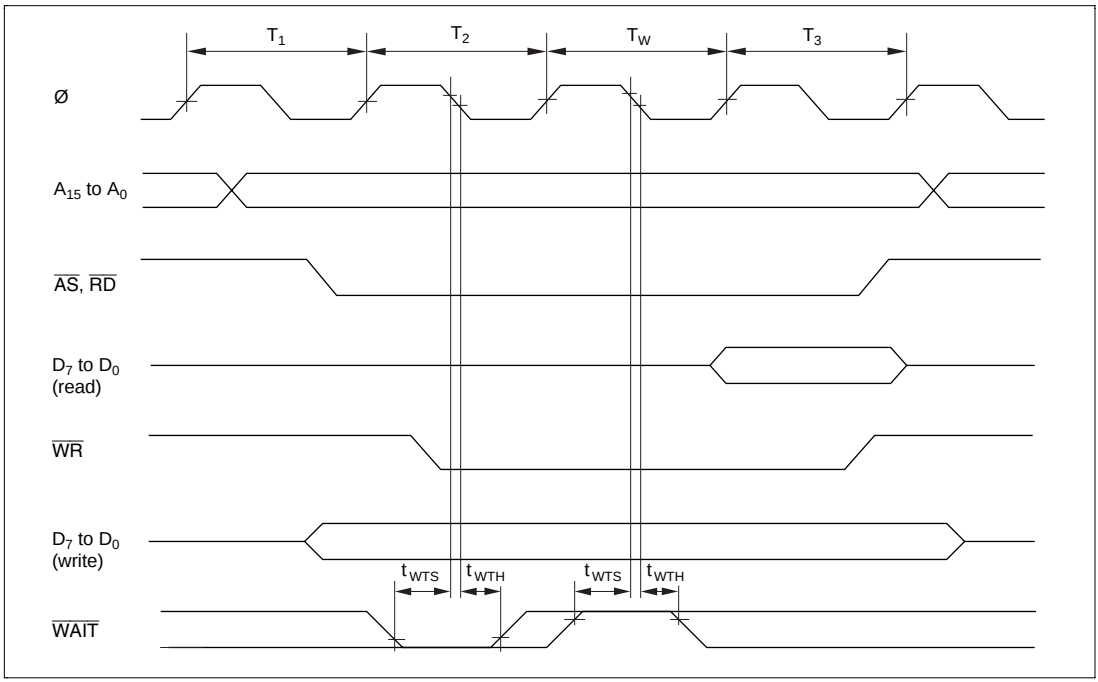


Figure 19.5 Basic Bus Cycle (with 1 Wait State) in Expanded Modes (Modes 1, 2)

19.3.2 Control Signal Timing

(1) Reset Input Timing

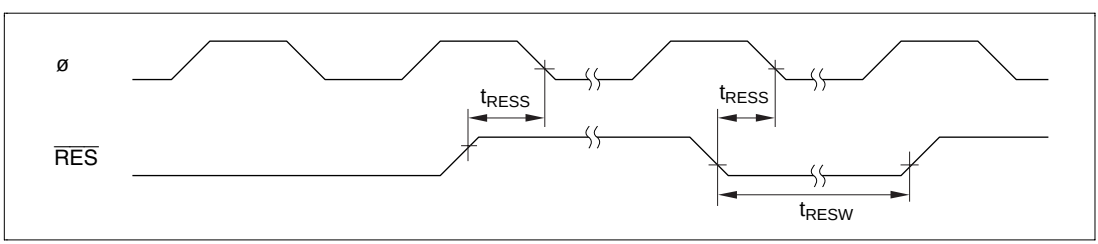


Figure 19.6 Reset Input Timing

(2) Interrupt Input Timing

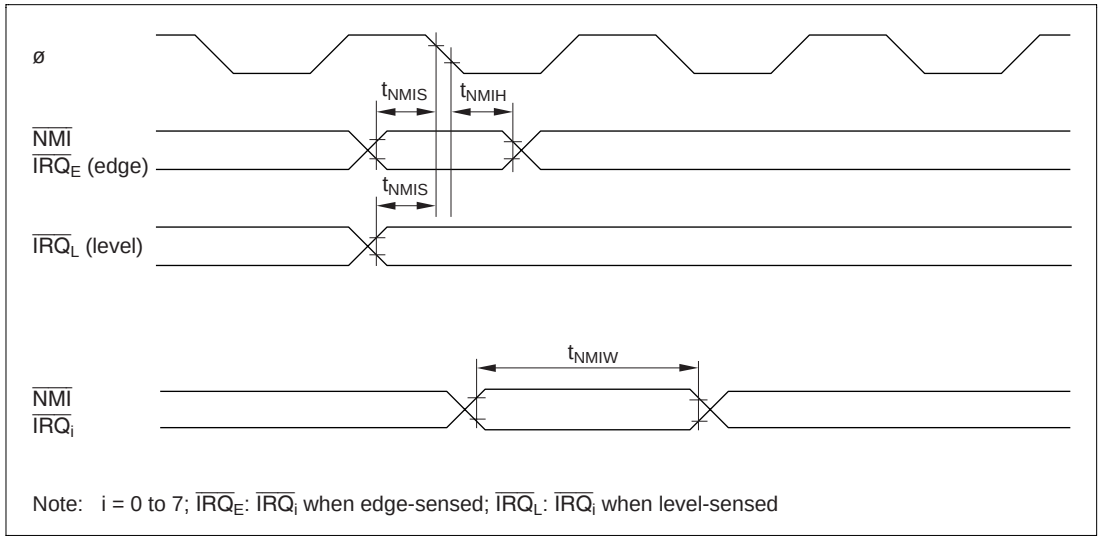


Figure 19.7 Interrupt Input Timing

(3) Clock Settling Timing

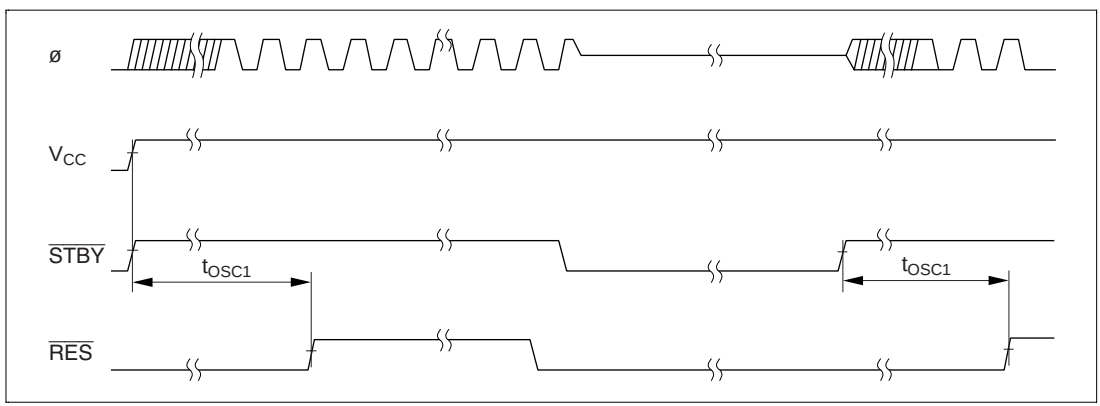


Figure 19.8 Clock Settling Timing

(4) Clock Settling Timing for Recovery from Software Standby Mode

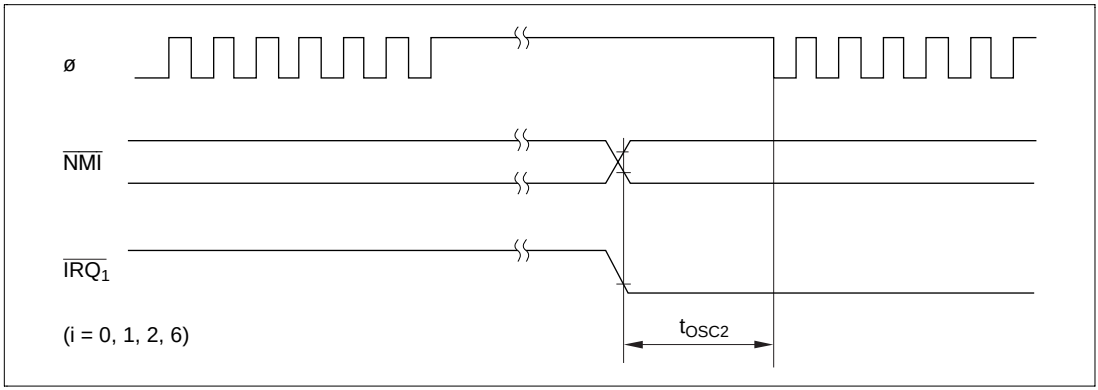


Figure 19.9 Clock Settling Timing for Recovery from Software Standby Mode

19.3.3 16-Bit Free-Running Timer Timing

(1) Free-Running Timer Input/Output Timing

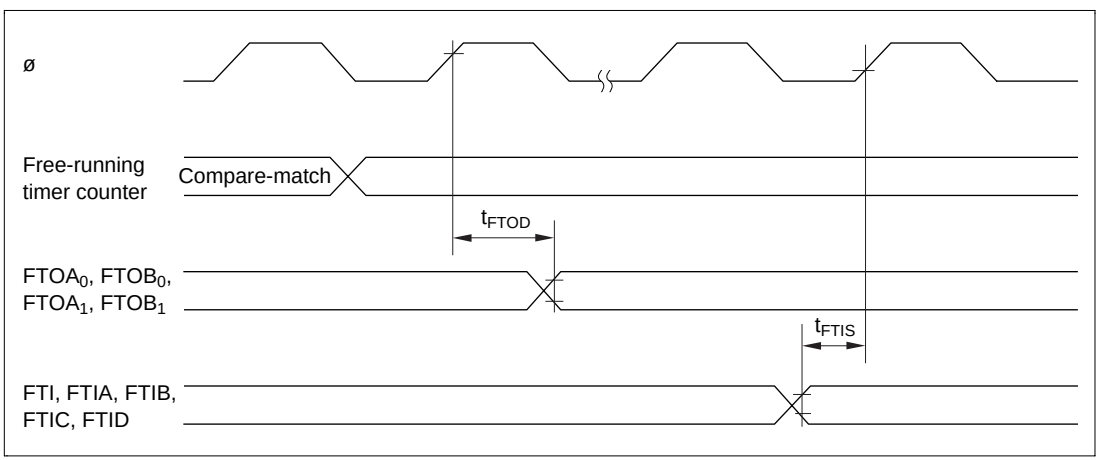


Figure 19.10 Free-Running Timer Input/Output Timing

(2) External Clock Input Timing for Free-Running Timer

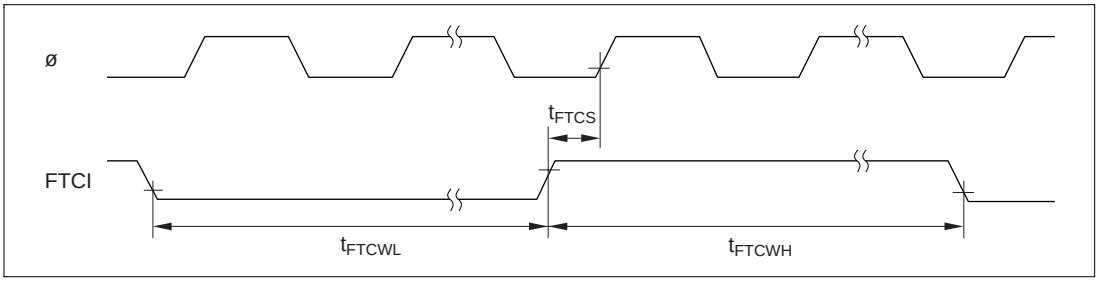


Figure 19.11 External Clock Input Timing for Free-Running Timer

19.3.4 8-Bit Timer Timing

(1) 8-Bit Timer Output Timing

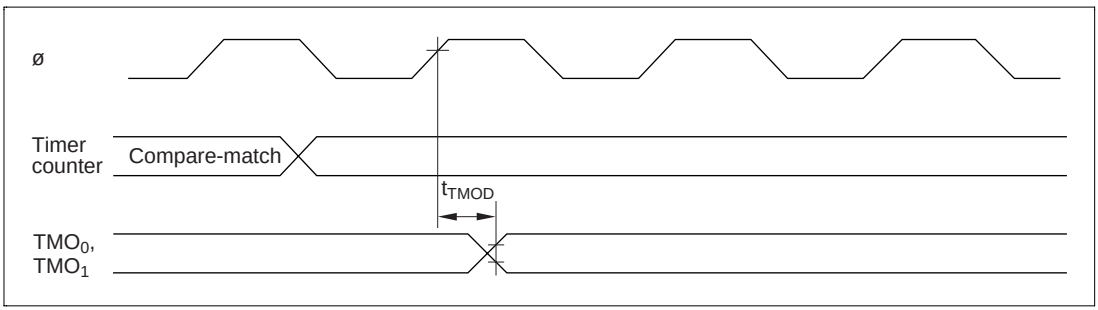


Figure 19.12 8-Bit Timer Output Timing

(2) 8-Bit Timer Clock Input Timing

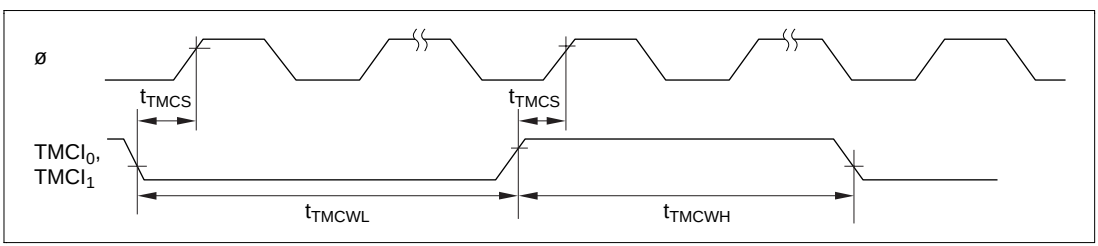


Figure 19.13 8-Bit Timer Clock Input Timing

(3) 8-Bit Timer Reset Input Timing

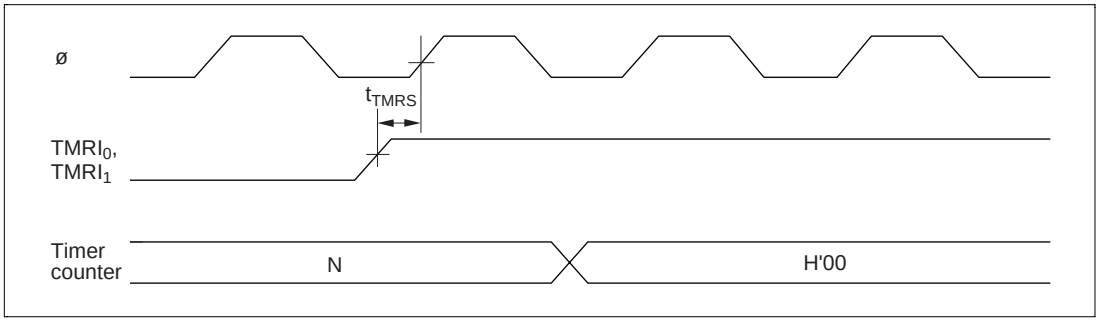


Figure 19.14 8-Bit Timer Reset Input Timing

19.3.5 Serial Communication Interface Timing

(1) SCI Input/Output Timing

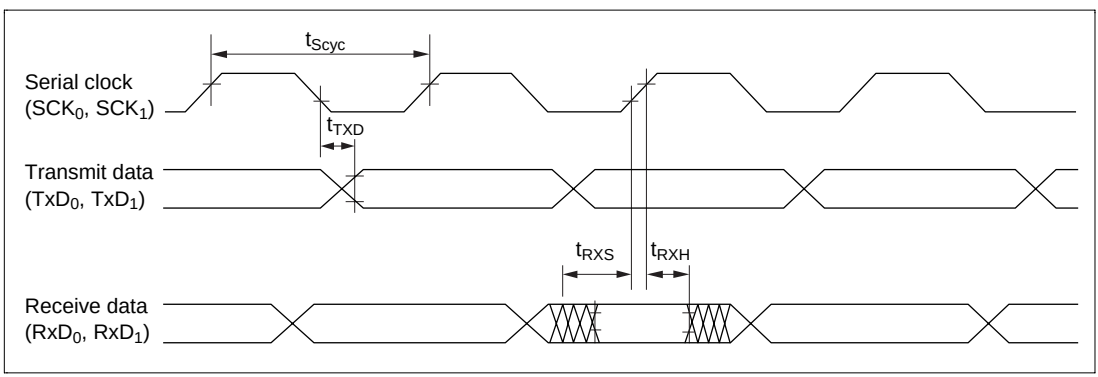


Figure 19.15 SCI Input/Output Timing (Synchronous Mode)

(2) SCI Input Clock Timing

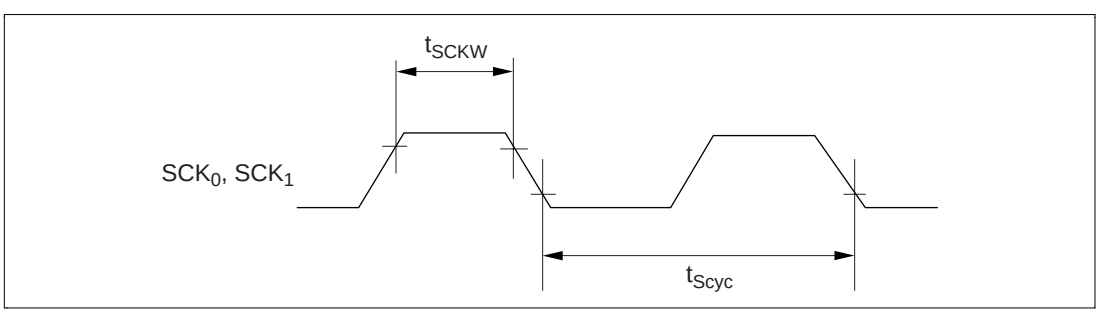


Figure 19.16 SCI Input Clock Timing

19.3.6 I/O Port Timing

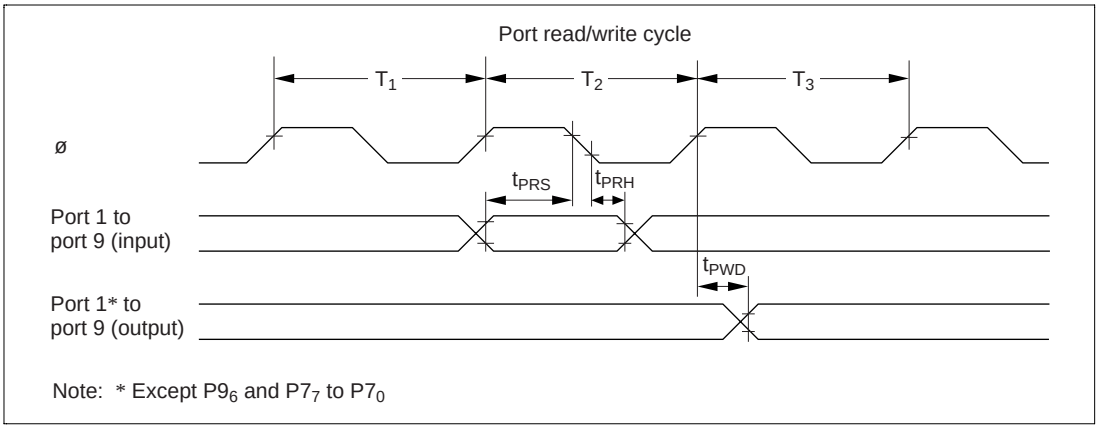


Figure 19.17 I/O Port Input/Output Timing

19.3.7 Timing Pattern Controller Timing

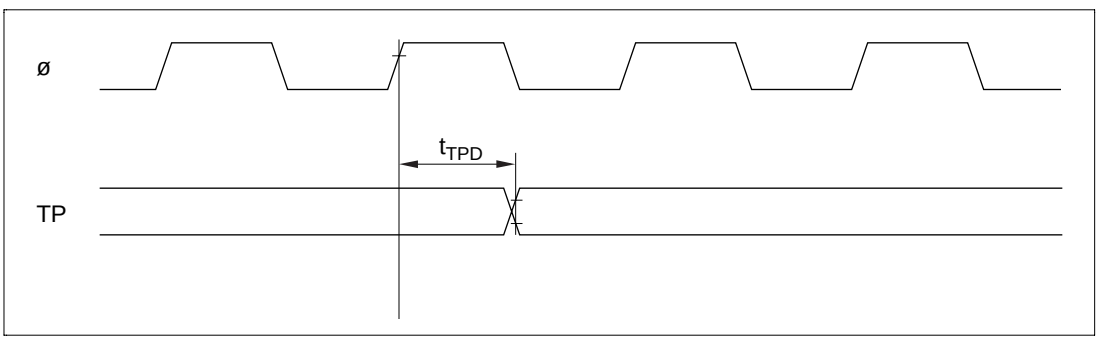


Figure 19.18 Timing Pattern Controller Timing

19.3.8 DPRAM Timing

Explanation of Notation in Figure 19.21 and 19.22: The numbers 0, 1, and 2 used to describe a data register state indicate 0, 1, and 2 bytes are available, respectively. They also indicate that the data register is being accessed.

(1) DPRAM Write Cycle

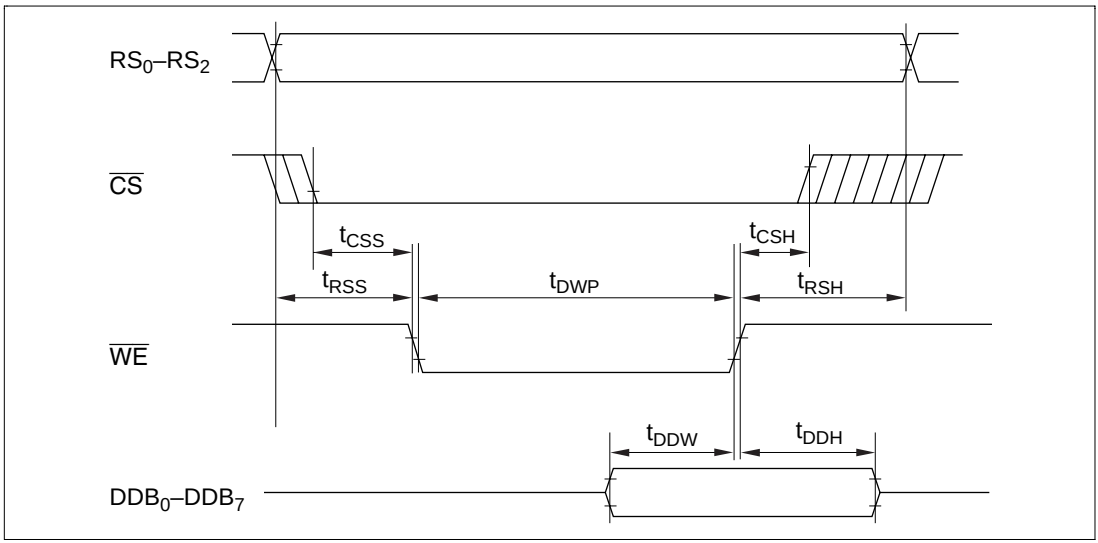


Figure 19.19 DPRAM Write Timing

(2) DPRAM Read Cycle

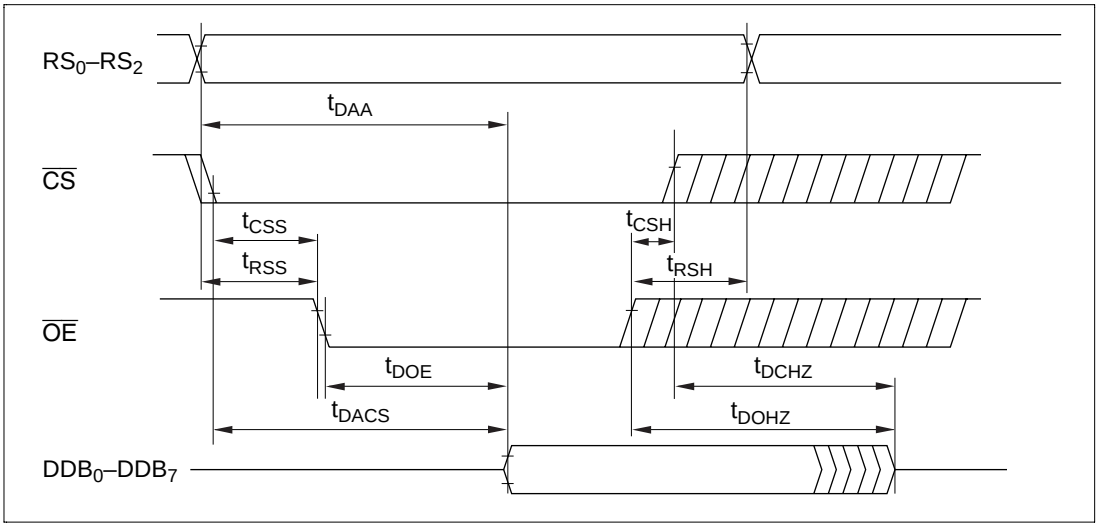


Figure 19.20 DPRAM Read Timing

(3) $\overline{\text{WRQ}}$ Output Timing

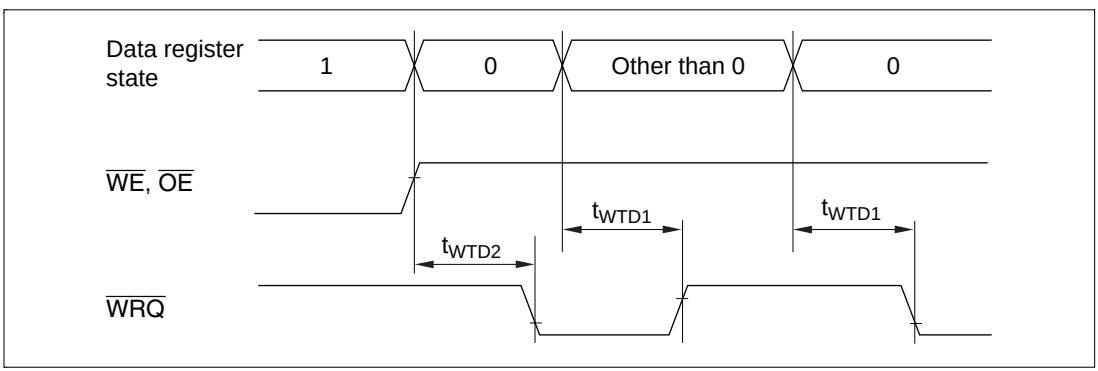


Figure 19.21 $\overline{\text{WRQ}}$ Output Timing

(4) Strobe Interval, Access Interval

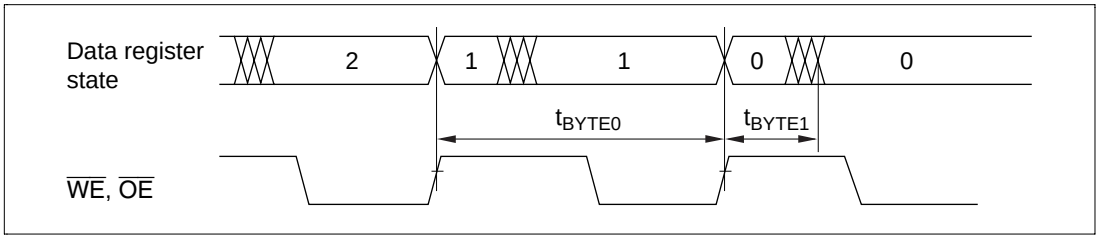


Figure 19.22 (a) Strobe Interval and Access Interval (with $\overline{WRQ}$)

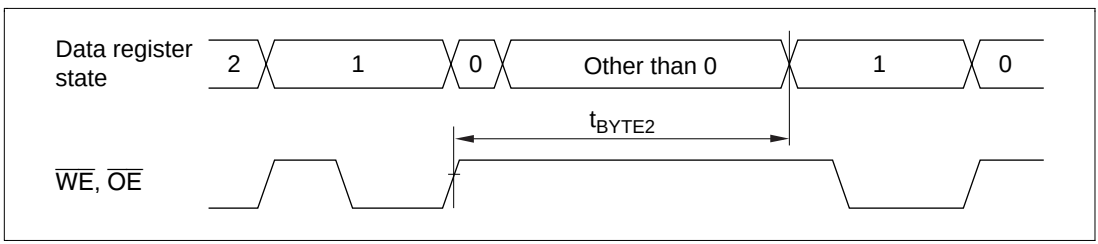


Figure 19.22 (b) Access Interval in Bound Buffer Mode (without $\overline{WRQ}$)

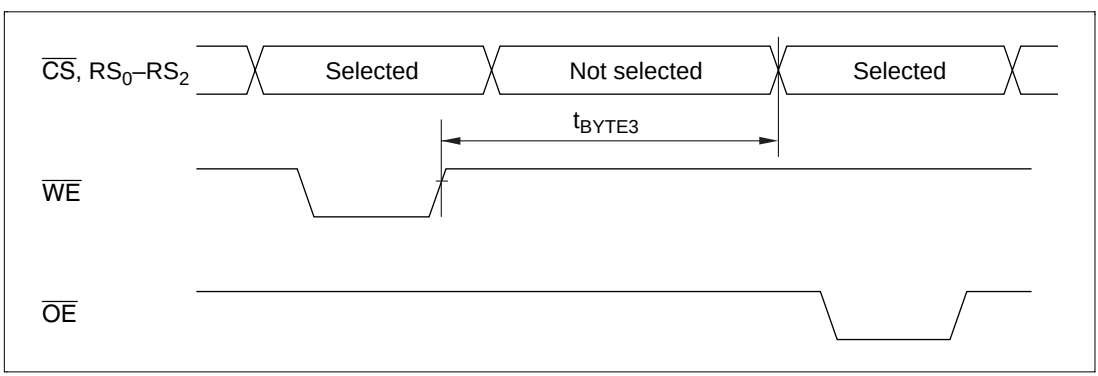


Figure 19.22 (c) Access Interval during Buffer Query Mode (without $\overline{WRQ}$)

(5) Receive Timing of Parallel Handshake Interface

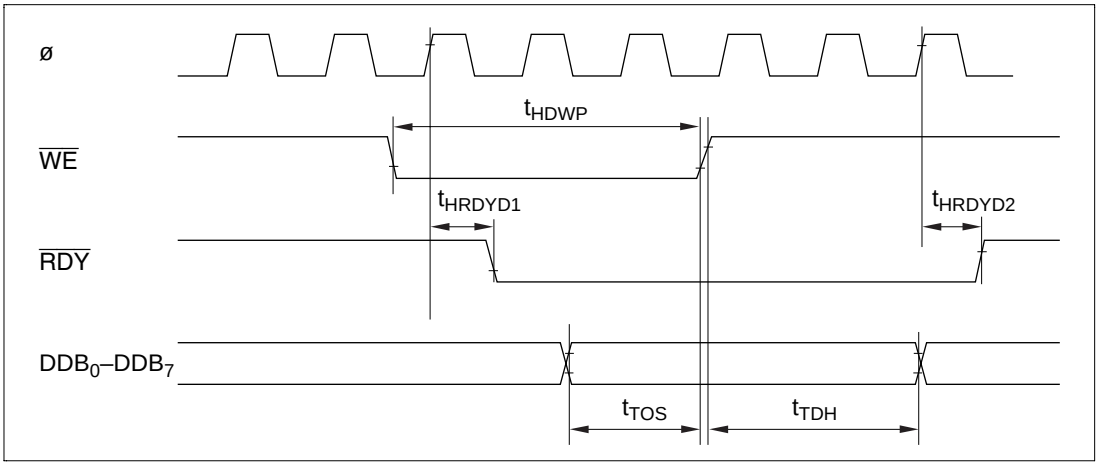


Figure 19.23 Receive Timing in Handshake Mode

(6) Transmit Timing of Parallel Handshake Interface

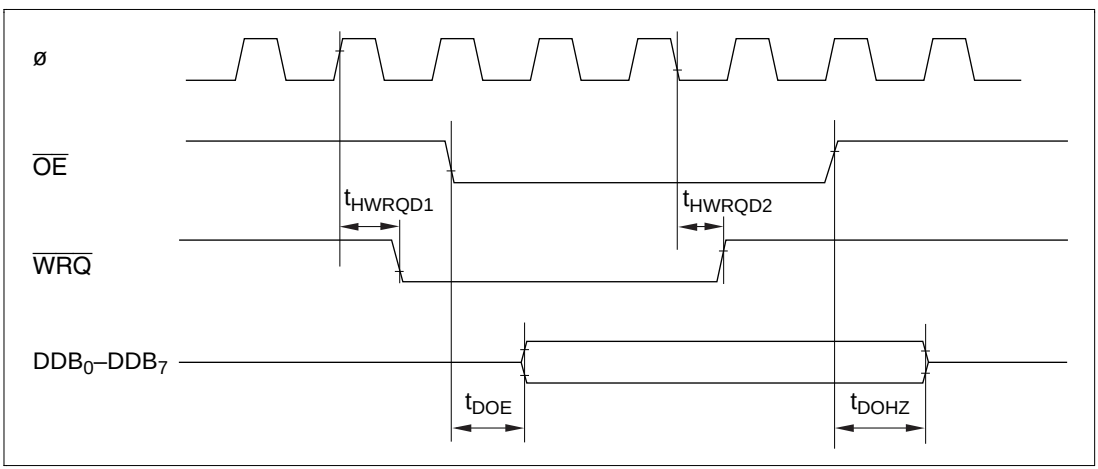


Figure 19.24 Transmit Timing in Handshake Mode

19.3.9 External Clock Output Timing

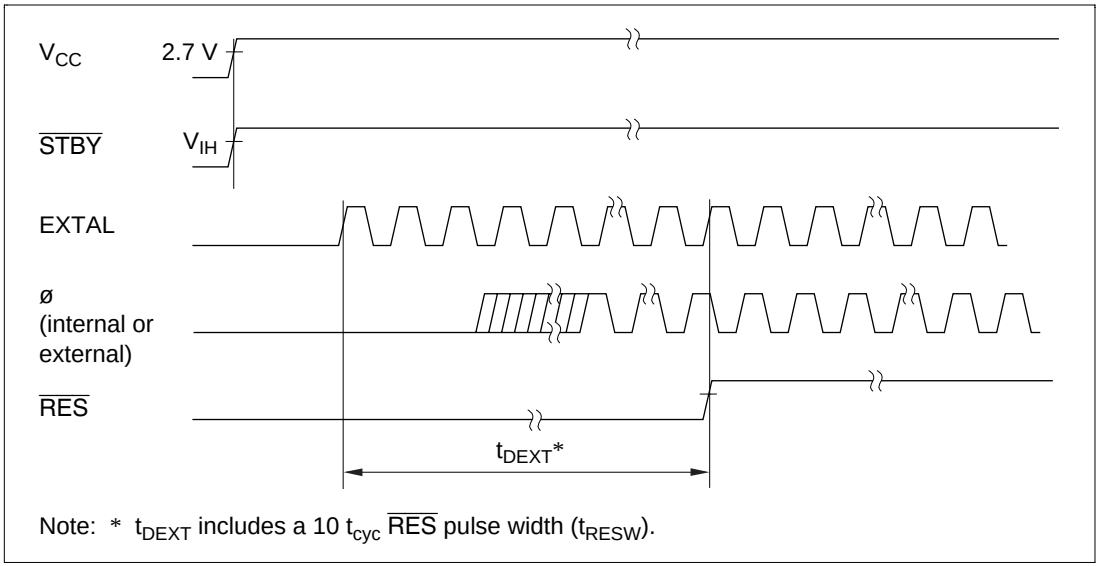


Figure 19.25 External Clock Output Stabilization Delay Time

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
∧	AND logical
∨	OR logical
⊕	Exclusive OR logical
→	Move
¬	Not

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

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
MOVFPE @aa:16, Rd	B	Not supported																
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

Table A.1 Instruction Set (cont)

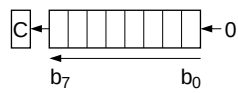
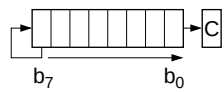
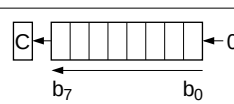
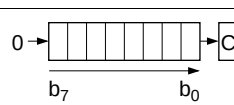
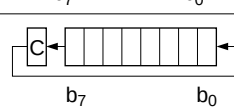
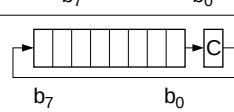
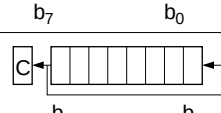
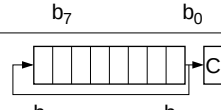
Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
OR.B #xx:8, Rd	B	Rd8∨#xx:8 → Rd8	2									—	—	↕	↕	0	—	2
OR.B Rs, Rd	B	Rd8∨Rs8 → Rd8		2								—	—	↕	↕	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	$\overline{Rd}$ → Rd		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	↕	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

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
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

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
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

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Branching Condition	Addressing Mode/ Instruction Length									Condition Code						No. of States
				#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
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

Table A.1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
JSR @@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	—	if R4L≠0 then Repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L Until R4L=0 else next									4	—	—	—	—	—	—	(4)

Notes: The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory.

(1) Set to 1 when there is a carry or borrow from bit 11; otherwise cleared to 0.

(2) If the result is zero, the previous value of the flag is retained; otherwise the flag is cleared to 0.

(3) Set to 1 if decimal adjustment produces a carry; otherwise cleared to 0.

(4) The number of states required for execution is 4n+8 (n = value of R4L).

(5) These instructions are not supported by the H8/3318 Series.

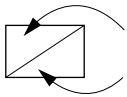
(6) Set to 1 if the divisor is negative; otherwise cleared to 0.

(7) Set to 1 if the divisor is zero; otherwise cleared to 0.

A.2 Operation Code Map

Table A.2 is a map of the operation codes contained in the first byte of the instruction code (bits 15 to 8 of the first instruction word).

Some pairs of instructions have identical first bytes. These instructions are differentiated by the first bit of the second byte (bit 7 of the first instruction word).



Instruction when first bit of byte 2 (bit 7 of first instruction word) is 0.
Instruction when first bit of byte 2 (bit 7 of first instruction word) is 1.

High \ Low	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SLEEP	STC	LDC	ORC	XORC	ANDC	LDC	ADD		INC	ADDS	MOV		ADDX	DAA
1	SHLL / SHAL	SHLR / SHAR	ROTXL / ROTL	ROTXR / ROTR	OR	XOR	AND	NOT / NEG	SUB		DEC	SUBS	CMP		SUBX	DAS
2	MOV															
3																
4	BRA*2	BRN*2	BHI	BLS	BCC*2	BCS*2	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU			RTS	BSR	RTE			JMP				JSR		
6	BSET	BNOT	BCLR	BTS				BST / BIST	MOV*1							
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD		MOV		EEPMOV	Bit manipulation instructions			
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

Notes: 1. The MOVFPE and MOVTPE instructions are identical to MOV instructions in the first byte and first bit of the second byte (bits 15 to 7 of the instruction word). The PUSH and POP instructions are identical in machine language to MOV instructions.
2. The BT, BF, BHS, and BLO instructions are identical in machine language to BRA, BRN, BCC, and BCS, respectively.

A.3 Number of States Required for Execution

The tables below can be used to calculate the number of states required for instruction execution. Table A.3 indicates the number of states required for each cycle (instruction fetch, branch address read, stack operation, byte data access, word data access, internal operation). Table A.4 indicates the number of cycles of each type occurring in each instruction. The total number of states required for execution of an instruction can be calculated from these two tables as follows:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples: Mode 1 (on-chip ROM disabled), stack located in external memory, 1 wait state inserted in external memory access.

1. BSET #0, @FFC7
   From table A.4: $I = L = 2$, $J = K = M = N = 0$
   From table A.3: $S_I = 8$, $S_L = 3$
   Number of states required for execution: $2 \times 8 + 2 \times 3 = 22$

2. JSR @@30
   From table A.4: $I = 2$, $J = K = 1$, $L = M = N = 0$
   From table A.3: $S_I = S_J = S_K = 8$
   Number of states required for execution: $2 \times 8 + 1 \times 8 + 1 \times 8 = 32$

Table A.3 Number of States Taken by Each Cycle in Instruction Execution

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		

Note: All values left blank are zero.

Table A.4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BIAND	BIAND #xx:3, Rd	1					
	BIAND #xx:3, @Rd	2			1		
	BIAND #xx:3, @aa:8	2			1		
BILD	BILD #xx:3, Rd	1					
	BILD #xx:3, @Rd	2			1		
	BILD #xx:3, @aa:8	2			1		
BIOR	BIOR #xx:3, Rd	1					
	BIOR #xx:3, @Rd	2			1		
	BIOR #xx:3, @aa:8	2			1		
BIST	BIST #xx:3, Rd	1					
	BIST #xx:3, @Rd	2			2		
	BIST #xx:3, @aa:8	2			2		
BIXOR	BIXOR #xx:3, Rd	1					
	BIXOR #xx:3, @Rd	2			1		
	BIXOR #xx:3, @aa:8	2			1		
BLD	BLD #xx:3, Rd	1					
	BLD #xx:3, @Rd	2			1		
	BLD #xx:3, @aa:8	2			1		
BNOT	BNOT #xx:3, Rd	1					
	BNOT #xx:3, @Rd	2			2		
	BNOT #xx:3, @aa:8	2			2		
	BNOT Rn, Rd	1					
	BNOT Rn, @Rd	2			2		
	BNOT Rn, @aa:8	2			2		
BOR	BOR #xx:3, Rd	1					
	BOR #xx:3, @Rd	2			1		
	BOR #xx:3, @aa:8	2			1		
BSET	BSET #xx:3, Rd	1					
	BSET #xx:3, @Rd	2			2		
	BSET #xx:3, @aa:8	2			2		
	BSET Rn, Rd	1					
	BSET Rn, @Rd	2			2		
	BSET Rn, @aa:8	2			2		

Note: All values left blank are zero.

Table A.4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BSR	BSR d:8	2		1			
BST	BST #xx:3, Rd	1					
	BST #xx:3, @Rd	2			2		
	BST #xx:3, @aa:8	2			2		
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @Rd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @Rd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	BXOR #xx:3, @Rd	2			1		
	BXOR #xx:3, @aa:8	2			1		
CMP	CMP.B #xx:8, Rd	1					
	CMP.B Rs, Rd	1					
	CMP.W Rs, Rd	1					
DAA	DAA.B Rd	1					
DAS	DAS.B Rd	1					
DEC	DEC.B Rd	1					
DIVXU	DIVXU.B Rs, Rd	1					12
EEPMOV	EEPMOV	2			2n+2*		1
INC	INC.B Rd	1					
JMP	JMP @Rn	2					
	JMP @aa:16	2					2
	JMP @@aa:8	2	1				2
JSR	JSR @Rn	2		1			
	JSR @aa:16	2		1			2
	JSR @@aa:8	2	1	1			
LDC	LDC #xx:8, CCR	1					
	LDC Rs, CCR	1					
MOV	MOV.B #xx:8, Rd	1					
	MOV.B Rs, Rd	1					

Notes: All values left blank are zero.

* n: Initial value in R4L. Source and destination are accessed n + 1 times each.

Table A.4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	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.Rs, Rd	1					12
NEG	NEG.B Rd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
OR	OR.B #xx:8, Rd	1					
	OR.B Rs, Rd	1					
ORC	ORC #xx:8, CCR	1					

Note: All values left blank are zero.

Table A.4 Number of Cycles in Each Instruction (cont)

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
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 Register Field

B.1 Register Addresses and Bit Names

Addr. (Last Byte)	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'80										External
H'81										addresses
H'82										(in expanded
H'83										modes)
H'84										
H'85										
H'86										
H'87										
H'88	SMR	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI1
H'89	BRR									
H'8A	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'8B	TDR									
H'8C	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'8D	RDR									
H'8E	—	—	—	—	—	—	—	—	—	
H'8F	—	—	—	—	—	—	—	—	—	
H'90	TIER	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	FRT0
H'91	TCSR	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA	
H'92	FRCH									
H'93	FRCL									
H'94	OCRAH									
	OCRBH									
H'95	OCRAL									
	OCRBL									
H'96	TCR	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0	
H'97	TOCR	—	—	—	OCRS	OEA	OEB	OLVLA	OLVLB	
H'98	ICRAH									
H'99	ICRAL									

Notes: FRT0: Free-running timer 0
SCI1: Serial communication interface 1

Addr. (Last Byte)	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'9A	ICRBH									FRT0
H'9B	ICRBL									
H'9C	ICRCH									
H'9D	ICRCL									
H'9E	ICRDH									
H'9F	ICRDL									
H'A0	TCR	ICIE	OCIEB	OCIEA	OVIE	OEB	OEA	CKS1	CKS0	FRT1
H'A1	TCSR	ICF	OCFB	OCFA	OVF	OLVLB	OLVLA	IEDG	CCLRA	
H'A2	FRCH									
H'A3	FRCL									
H'A4	OCRAH									
H'A5	OCRAL									
H'A6	OCRBH									
H'A7	OCRBL									
H'A8	ICRH									
H'A9	ICRL									
H'AA	TCSR/ TCNT									WDT
H'AB	TCNT									
H'AC	P1PCR	$P1_7$PCR	$P1_6$PCR	$P1_5$PCR	$P1_4$PCR	$P1_3$PCR	$P1_2$PCR	$P1_1$PCR	$P1_0$PCR	Port 1
H'AD	P2PCR	$P2_7$PCR	$P2_6$PCR	$P2_5$PCR	$P2_4$PCR	$P2_3$PCR	$P2_2$PCR	$P2_1$PCR	$P2_0$PCR	Port 2
H'AE	P3PCR	$P3_7$PCR	$P3_6$PCR	$P3_5$PCR	$P3_4$PCR	$P3_3$PCR	$P3_2$PCR	$P3_1$PCR	$P3_0$PCR	Port 3
H'AF	—	—	—	—	—	—	—	—	—	—
H'B0	P1DDR	$P1_7$DDR	$P1_6$DDR	$P1_5$DDR	$P1_4$DDR	$P1_3$DDR	$P1_2$DDR	$P1_1$DDR	$P1_0$DDR	Port 1
H'B1	P2DDR	$P2_7$DDR	$P2_6$DDR	$P2_5$DDR	$P2_4$DDR	$P2_3$DDR	$P2_2$DDR	$P2_1$DDR	$P2_0$DDR	Port 2
H'B2	P1DR	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$	Port 1
H'B3	P2DR	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$	Port 2
H'B4	P3DDR	$P3_7$DDR	$P3_6$DDR	$P3_5$DDR	$P3_4$DDR	$P3_3$DDR	$P3_2$DDR	$P3_1$DDR	$P3_0$DDR	Port 3
H'B5	P4DDR	$P4_7$DDR	$P4_6$DDR	$P4_5$DDR	$P4_4$DDR	$P4_3$DDR	$P4_2$DDR	$P4_1$DDR	$P4_0$DDR	Port 4
H'B6	P3DR	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$	Port 3
H'B7	P4DR	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$	Port 4
H'B8	P5DDR	—	—	—	—	—	$P5_2$DDR	$P5_1$DDR	$P5_0$DDR	Port 5

Notes: FRT0: Free-running timer 0
FRT1: Free-running timer 1
WDT: Watchdog timer

Addr. (Last Byte)	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'B9	P6DDR	$P6_7$DDR	$P6_6$DDR	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	$P6_2$DDR	$P6_1$DDR	$P6_0$DDR	Port 6
H'BA	P5DR	—	—	—	—	—	$P5_2$	$P5_1$	$P5_0$	Port 5
H'BB	P6DR	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$	Port 6
H'BC	—	—	—	—	—	—	—	—	—	—
H'BD	P8DDR	—	$P8_6$DDR	$P8_5$DDR	$P8_4$DDR	$P8_3$DDR	$P8_2$DDR	$P8_1$DDR	$P8_0$DDR	Port 8
H'BE	P7DR	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$	Port 7
H'BF	P8DR	—	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$	Port 8
H'C0	P9DDR	$P9_7$DDR	$P9_6$DDR	$P9_5$DDR	$P9_4$DDR	$P9_3$DDR	$P9_2$DDR	$P9_1$DDR	$P9_0$DDR	Port 9
H'C1	P9DR	$P9_7$	$P9_6$	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$	
H'C2	WSCR	—	—	CKDBL	—	WMS1	WMS0	WC1	WC0	System control
H'C3	STCR	RING	CMPF	CMPIE	LOAD	MARK	—	ICKS1	ICKS0	
H'C4	SYSCR	SSBY	STS2	STS1	STS0	XRST	NMIEG	DPME	RAME	
H'C5	MDCR	—	—	—	—	—	—	MDS1	MDS0	
H'C6	ISCR	IRQ7SC	IRQ6SC	IRQ5SC	IRQ4SC	IRQ3SC	IRQ2SC	IRQ1SC	IRQ0SC	
H'C7	IER	IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E	
H'C8	TCR	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR0
H'C9	TCSR	CMFB	CMFA	OVF	PWME	OS3	OS2	OS1	OS0	
H'CA	TCORA									
H'CB	TCORB									
H'CC	TCNT									
H'CD	NDER2	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8	TPC
H'CE	NDRB*	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8	
		NDR15	NDR14	NDR13	NDR12	—	—	—	—	
H'CF	NDRA*	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0	
		NDR7	NDR6	NDR5	NDR4	—	—	—	—	
H'D0	TCR	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	TMR1
H'D1	TCSR	CMFB	CMFA	OVF	PWME	OS3	OS2	OS1	OS0	
H'D2	TCORA									
H'D3	TCORB									
H'D4	TCNT									

Notes: TMR0: 8-bit timer channel 0
TMR1: 8-bit timer channel 1
TPC: Programmable timing pattern controller
* The address changes depending on the trigger setting

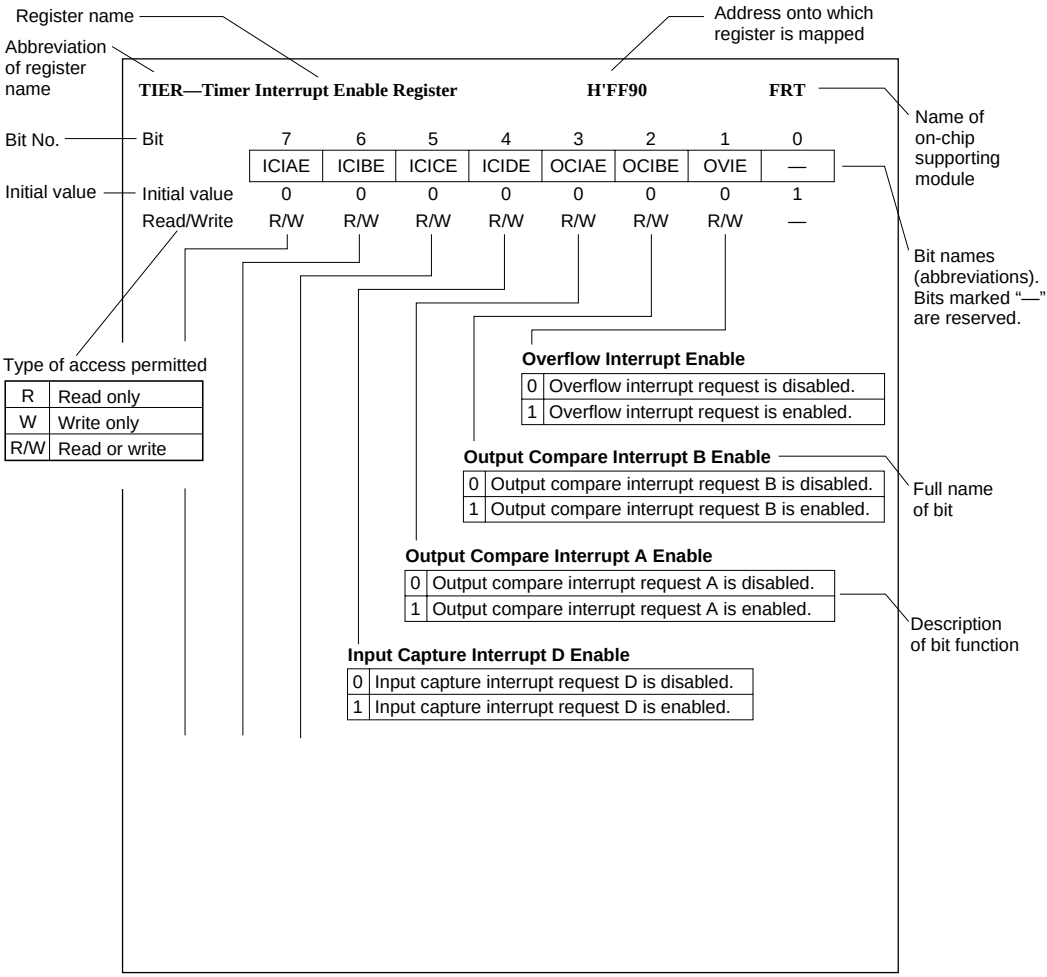
Addr. (Last Byte)	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'D5	NDER1	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0	TPC
H'D6	NDRB*	—	—	—	—	—	—	—	—	
		—	—	—	—	NDR11	NDR10	NDR9	NDR8	
H'D7	NDRA*	—	—	—	—	—	—	—	—	
		—	—	—	—	NDR3	NDR2	NDR1	NDR0	
H'D8	SMR	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI0
H'D9	BRR									
H'DA	SCR	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'DB	TDR									
H'DC	SSR	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT	
H'DD	RDR									
H'DE	SCMR	—	—	—	—	SDIR	SINV	—	SMIF	
H'DF	—	—	—	—	—	—	—	—	—	
H'E0	ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D
H'E1	ADDRAL	AD1	AD0	—	—	—	—	—	—	
H'E2	ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'E3	ADDRBL	AD1	AD0	—	—	—	—	—	—	
H'E4	ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'E5	ADDRCL	AD1	AD0	—	—	—	—	—	—	
H'E6	ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
H'E7	ADDRDL	AD1	AD0	—	—	—	—	—	—	
H'E8	ADCSR	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0	
H'E9	ADCR	TRGE	—	—	—	—	—	—	—	
H'EA	TPMR	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV	TPC
H'EB	TPCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	
H'EC	—	—	—	—	—	—	—	—	—	—
H'ED	—	—	—	—	—	—	—	—	—	
H'EE	—	—	—	—	—	—	—	—	—	
H'EF	—	—	—	—	—	—	—	—	—	

Notes: A/D: Analog-to-Digital converter
SCI0: Serial communication interface 0
TPC: Programmable timing pattern controller
* The address changes depending on the trigger setting

Addr. (Last Byte)	Register Name	Bit Names								Module
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'F0	PCCSR	QREF	EWRQ	EWAKAR	ERAKAR	MWEF	MREF	EMWI	EMRI	DTU
H'F1	IOCR	HSCE	DPEA	DPEB	RPEA	RPEB	RPEC	—	—	
H'F2	RLARA									
H'F3	RLARB									
H'F4	CPARB									
H'F5	DTARH									
H'F6	DTCRA	DTE	DTIE	BUD2	BUD1	BUD0	SOS2	SOS1	SOS0	
H'F7	DTARA									
H'F8	DTCRB	DTE	DTIE	BUD2	BUD1	BUD0	SOS2	SOS1	SOS0	
H'F9	DTARB									
H'FA	DTCRC	DTE	DTIE	BUD2	BUD1	BUD0	SOS2	SOS1	SOS0	
H'FB	DTARC									
H'FC	DPDRWH									
H'FD	DPDRWL									
H'FE	DPDRRH									
H'FF	DPDRRL									

Note: DTU: Data transfer controller

B.2 Register Descriptions



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

0	0	ø clock
0	1	$ø_P$/4 clock
1	0	$ø_P$/16 clock
1	1	$ø_P$/64 clock

Multiprocessor Mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop Bit Length

0	One stop bit
1	Two stop bits

Parity Mode

0	Even parity
1	Odd parity

Parity Enable

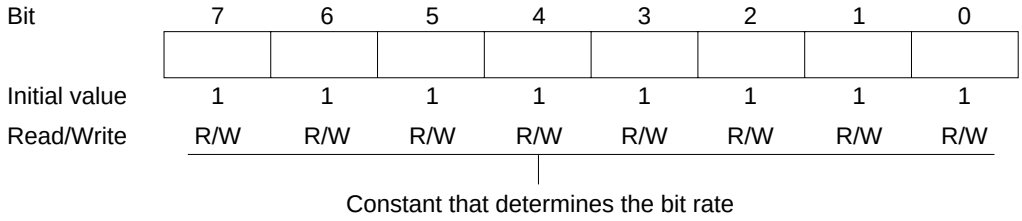
0	Transmit: No parity bit added. Receive: Parity bit not checked.
1	Transmit: Parity bit added. Receive: Parity bit checked.

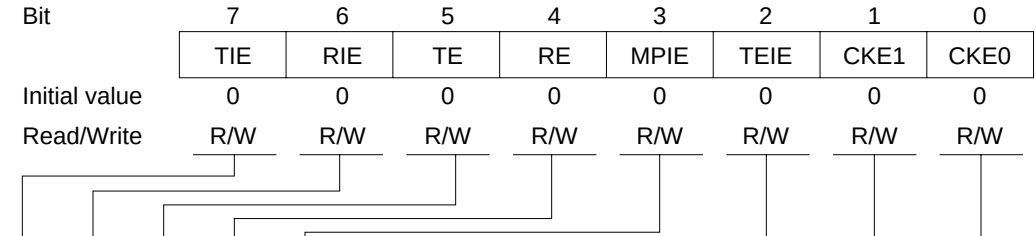
Character Length

0	8-bit data length
1	7-bit data length

Communication Mode

0	Asynchronous
1	Synchronous


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
Constant that determines the bit rate



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 0

0	The SCK pin is not used by the SCI.
1	The SCK pin is used for serial clock output.

Clock Enable 1

0	Internal clock
1	External clock

Transmit End Interrupt Enable

0	TSR-empty interrupt request is disabled.
1	TSR-empty interrupt request is enabled.

Multiprocessor Interrupt Enable

0	Multiprocessor receive interrupt function is disabled.
1	Multiprocessor receive interrupt function is enabled.

Receive Enable

0	Receive disabled
1	Receive enabled

Transmit Enable

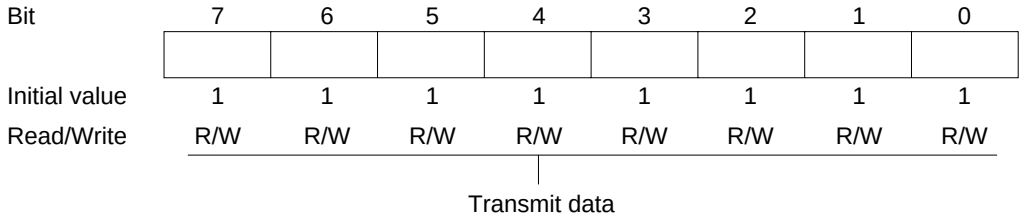
0	Transmit disabled
1	Transmit enabled

Receive Interrupt Enable

0	Receive interrupt and receive error interrupt requests are disabled.
1	Receive interrupt and receive error interrupt requests are enabled.

Transmit Interrupt Enable

0	TDR-empty interrupt request is disabled.
1	TDR-empty interrupt request is enabled.


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
Transmit data

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor Bit Transfer

0	Multiprocessor bit = 0 in transmit data.
1	Multiprocessor bit = 1 in transmit data.

Multiprocessor Bit

0	Multiprocessor bit = 0 in receive data.
1	Multiprocessor bit = 1 in receive data.

Transmit End

0	Cleared by reading TDRE = 1, then writing 0 in TDRE.
1	Set to 1 when TE = 0, or when TDRE = 1 at the end of character transmission.

Parity Error

0	Cleared by reading PER = 1, then writing 0 in PER.
1	Set when a parity error occurs (parity of receive data does not match parity selected by O/$\overline{E}$ bit in SMR).

Framing Error

0	Cleared by reading FER = 1, then writing 0 in FER.
1	Set when a framing error occurs (stop bit is 0).

Overrun Error

0	Cleared by reading ORER = 1, then writing 0 in ORER.
1	Set when an overrun error occurs (next data is completely received while RDRF bit is set to 1).

Receive Data Register Full

0	Cleared by reading RDRF = 1, then writing 0 in RDRF. Or when RDR is read in the DTU bus cycle.
1	Set when one character is received normally and transferred from RSR to RDR.

Transmit Data Register Empty

0	Cleared by reading TDRE = 1, then writing 0 in TDRE. Or when a data is written to TDR in the DTU bus cycle.
1	Set when: 1. Data is transferred from TDR to TSR. 2. TE is cleared while TDRE = 0.

Note: * Software can write a 0 in bits 7 to 3 to clear the flags, but cannot write a 1 in these bits.

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Receive data

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	—

Overflow Interrupt Enable

0	Overflow interrupt request is disabled.
1	Overflow interrupt request is enabled.

Output Compare Interrupt B Enable

0	Output compare interrupt request B is disabled.
1	Output compare interrupt request B is enabled.

Output Compare Interrupt A Enable

0	Output compare interrupt request A is disabled.
1	Output compare interrupt request A is enabled.

Input Capture Interrupt D Enable

0	Input capture interrupt request D is disabled.
1	Input capture interrupt request D is enabled.

Input Capture Interrupt C Enable

0	Input capture interrupt request C is disabled.
1	Input capture interrupt request C is enabled.

Input Capture Interrupt B Enable

0	Input capture interrupt request B is disabled.
1	Input capture interrupt request B is enabled.

Input Capture Interrupt A Enable

0	Input capture interrupt request A is disabled.
1	Input capture interrupt request A is enabled.

Bit	7	6	5	4	3	2	1	0
	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

Counter Clear A

0	FRC count is not cleared.
1	FRC count is cleared by compare-match A.

Timer Overflow Flag

0	Cleared by reading OVF = 1, then writing 0 in OVF.
1	Set when FRC changes from H'FFFF to H'0000.

Output Compare Flag B

0	Cleared by reading OCFB = 1, then writing 0 in OCFB.
1	Set when FRC = OCRB.

Output Compare Flag A

0	Cleared by reading OCFA = 1, then writing 0 in OCFA.
1	Set when FRC = OCRA.

Input Capture Flag D

0	Cleared by reading ICFD = 1, then writing 0 in ICFD.
1	Set by FTID input.

Input Capture Flag C

0	Cleared by reading ICFC = 1, then writing 0 in ICFC.
1	Set by FTIC input.

Input Capture Flag B

0	Cleared by reading ICFB = 1, then writing 0 in ICFB.
1	Set when FTIB input causes FRC to be copied to ICRB.

Input Capture Flag A

0	Cleared by reading ICFA = 1, then writing 0 in ICFA.
1	Set when FTIA input causes FRC to be copied to ICRA.

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

FRC (H and L)—Free-Running Counter **H'FF92, H'FF93** **FRT0**

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

Count value

OCRA (H and L)—Output Compare Register A **H'FF94, H'FF95** **FRT0**

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

Continually compared with FRC. OCFA is set to 1 when OCRA = FRC.

OCRB (H and L)—Output Compare Register B **H'FF94, H'FF95** **FRT0**

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

Continually compared with FRC. OCFB is set to 1 when OCRB = FRC.

Bit	7	6	5	4	3	2	1	0
	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	Internal clock source: $\phi_P/2$
0	1	Internal clock source: $\phi_P/8$
1	0	Internal clock source: $\phi_P/32$
1	1	External clock source: counted on rising edge

Buffer Enable B

0	ICRD is used for input capture D.
1	ICRD is buffer register for input capture B.

Buffer Enable A

0	ICRC is used for input capture C.
1	ICRC is buffer register for input capture A.

Input Edge Select D

0	Falling edge of FTID is valid.
1	Rising edge of FTID is valid.

Input Edge Select C

0	Falling edge of FTIC is valid.
1	Rising edge of FTIC is valid.

Input Edge Select B

0	Falling edge of FTIB is valid.
1	Rising edge of FTIB is valid.

Input Edge Select A

0	Falling edge of FTIA is valid.
1	Rising edge of FTIA is valid.

Bit	7	6	5	4	3	2	1	0
	—	—	—	OCRS	OEA	OEB	OLVLA	OLVLB
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Output Level B

0	Compare-match B causes 0 output.
1	Compare-match B causes 1 output.

Output Level A

0	Compare-match A causes 0 output.
1	Compare-match A causes 1 output.

Output Enable B

0	Output compare B output is disabled.
1	Output compare B output is enabled.

Output Enable A

0	Output compare A output is disabled.
1	Output compare A output is enabled.

Output Compare Register Select

0	The CPU can access OCRA.
1	The CPU can access OCRB.

ICRA (H and L)—Input Capture Register A **H'FF98, H'FF99** **FRT0**

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

Contains FRC count captured on FTIA input.

ICRB (H and L)—Input Capture Register B **H'FF9A, H'FF9B** **FRT0**

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

Contains FRC count captured on FTIB input.

ICRC (H and L)—Input Capture Register C **H'FF9C, H'FF9D** **FRT0**

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

Contains FRC count captured on FTIC input, or old ICRA value in buffer mode.

ICRD (H and L)—Input Capture Register D **H'FF9E, H'FF9F** **FRT0**

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

Contains FRC count captured on FTID input, or old ICRB value in buffer mode.

Bit	7	6	5	4	3	2	1	0
	ICIE	OCIEB	OCIEA	OVIE	OEB	OEA	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

0	0	Internal clock source: $\phi_P/2$
0	1	Internal clock source: $\phi_P/8$
1	0	Internal clock source: $\phi_P/32$
1	1	External clock source: counted on rising edge

Output Enable A

0	Output compare A output is disabled.
1	Output compare A output is enabled.

Output Enable B

0	Output compare B output is disabled.
1	Output compare B output is enabled.

Timer Overflow Interrupt Enable

0	Overflow interrupt request is disabled.
1	Overflow interrupt request is enabled.

Output Compare Interrupt Enable A

0	Output compare interrupt request A is disabled.
1	Output compare interrupt request A is enabled.

Output Compare Interrupt Enable B

0	Output compare interrupt request B is disabled.
1	Output compare interrupt request B is enabled.

Input Capture Interrupt Enable

0	Input capture interrupt request is disabled.
1	Input capture interrupt request is enabled.

Bit	7	6	5	4	3	2	1	0
	ICF	OCFB	OCFA	OVF	OLVLB	OLVLA	IEDG	CCLRA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W	R/W	R/W	R/W

Counter Clear A

0	FRC count is not cleared.
1	FRC count is cleared by compare-match A.

Input Edge Select

0	FRC value is transferred to ICR at falling edge of FTI
1	FRC value is transferred to ICR at rising edge of FTI

Output Level A

0	Compare-match A causes 0 output
1	Compare-match A causes 1 output

Output Level B

0	Compare-match B causes 0 output
1	Compare-match B causes 1 output

Timer Overflow Flag

0	Cleared by reading OVF = 1, then writing 0 in OVF
1	Set when FRC changes from H'FFFF to H'0000

Output Compare Flag A

0	Cleared by reading OCFA = 1, then writing 0 in OCFA
1	Set when FRC = OCRA

Output Compare Flag B

0	Cleared by reading OCFB = 1, then writing 0 in OCFB
1	Set when FRC = OCRB

Input Capture Flag

0	Cleared by reading ICF = 1, then writing 0 in ICF
1	Set when FRC value is copied to ICR by input capture signal

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

FRC (H and L)—Free Running Counter H'FFA2, FFA3 FRT1

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

Count value

OCRA (H and L)—Output Compare Register A H'FFA4, FFA5 FRT1

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

Continually compared with FRC. OCFA is set to 1 when OCRA = FRC.

OCRB (H and L)—Output Compare Register B H'FFA6, FFA7 FRT1

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

Continually compared with FRC. OCFB is set to 1 when OCRB = FRC.

ICR (H and L)—Input Capture Register H'FFA8, FFA9 FRT1

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

Contains FRC count captured by internal input capture signal generated from external input signal transition

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	—	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/(W)*3	R/W	R/W	—	R/W	R/W	R/W	R/W

Clock Select

0	0	0	øP/2
0	0	1	øP/32
0	1	0	øP/64
0	1	1	øP/128
1	0	0	øP/256
1	0	1	øP/512
1	1	0	øP/2048
1	1	1	øP/4096

Reset or NMI Select

0	NMI function enabled
1	Reset function enabled

Timer Enable

0	Timer disabled: TCNT is initialized to H'00 and stopped
1	Timer enabled: TCNT runs; CPU interrupts can be requested

Timer Mode Select

0	Interval timer mode (OVF interrupt request)
1	Watchdog timer mode (reset or NMI request)

Overflow flag

0	Cleared by reading OVF = 1, then writing 1 in OVF
1	Set when TCNT changes from H'FF to H'00

Notes: 1. Read access address.
2. Write address, but write access requires word transfer to H'FFAA.
3. Only 0 can be written, to clear the flag.

TCNT—Timer Counter **H'FFAB*** **WDT**

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

Count value

Note: * Write access requires word transfer to H'FFAA.

P1PCR—Port 1 Input 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

Port 1 Input Pull-Up Control

0	Input pull-up transistor is off.
1	Input pull-up transistor is on.

P2PCR—Port 2 Input 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

Port 2 Input Pull-Up Control

0	Input pull-up transistor is off.
1	Input pull-up transistor is on.

P3PCR—Port 3 Input Pull-Up Control Register H'FFAE Port 3

Bit	7	6	5	4	3	2	1	0
	$P3_7PCR$	$P3_6PCR$	$P3_5PCR$	$P3_4PCR$	$P3_3PCR$	$P3_2PCR$	$P3_1PCR$	$P3_0PCR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Port 3 Input Pull-Up Control

0	Input pull-up transistor is off.
1	Input pull-up transistor is on.

P1DDR—Port 1 Data Direction Register H'FFB0 Port 1

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

Port 1 Input/Output Control

0	Input port
1	Output port

P1DR—Port 1 Data Register H'FFB2 Port 1

Bit	7	6	5	4	3	2	1	0
	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2DDR—Port 2 Data Direction Register **H'FFB1** **Port 2**

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

Port 2 Input/Output Control

0	Input port
1	Output port

P2DR—Port 2 Data Register **H'FFB3** **Port 2**

Bit	7	6	5	4	3	2	1	0
	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3DDR—Port 3 Data Direction Register **H'FFB4** **Port 3**

Bit	7	6	5	4	3	2	1	0
	$P3_7DDR$	$P3_6DDR$	$P3_5DDR$	$P3_4DDR$	$P3_3DDR$	$P3_2DDR$	$P3_1DDR$	$P3_0DDR$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 3 Input/Output Control

0	Input port
1	Output port

P3DR—Port 3 Data Register **H'FFB6** **Port 3**

Bit	7	6	5	4	3	2	1	0
	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P4DDR—Port 4 Data Direction Register **H'FFB5** **Port 4**

Bit	7	6	5	4	3	2	1	0
	$P4_7$DDR	$P4_6$DDR	$P4_5$DDR	$P4_4$DDR	$P4_3$DDR	$P4_2$DDR	$P4_1$DDR	$P4_0$DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 4 Input/Output Control

0	Input port
1	Output port

P4DR—Port 4 Data Register **H'FFB7** **Port 4**

Bit	7	6	5	4	3	2	1	0
	$P4_7$	$P4_6$	$P4_5$	$P4_4$	$P4_3$	$P4_2$	$P4_1$	$P4_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P5DDR—Port 5 Data Direction Register **H'FFB8** **Port 5**

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

Port 5 Input/Output Control

0	Input port
1	Output port

P5DR—Port 5 Data Register **H'FFBA** **Port 5**

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

P6DDR—Port 6 Data Direction Register **H'FFB9** **Port 6**

Bit	7	6	5	4	3	2	1	0
	$P6_7$DDR	$P6_6$DDR	$P6_5$DDR	$P6_4$DDR	$P6_3$DDR	$P6_2$DDR	$P6_1$DDR	$P6_0$DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 6 Input/Output Control

0	Input port
1	Output port

P6DR—Port 6 Data Register **H'FFBB** **Port 6**

Bit	7	6	5	4	3	2	1	0
	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P7DR—Port 7 Data Register **H'FFBE** **Port 7**

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

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

P8DDR—Port 8 Data Direction Register **H'FFBD** **Port 8**

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

Port 8 Input/Output Control

0	Input port
1	Output port

P8DR—Port 8 Data Register **H'FFBF** **Port 8**

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

P9DDR—Port 9 Data Direction Register **H'FFC0** **Port 9**

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

Port 9 Input/Output Control

0	Input port
1	Output port

Bit	7	6	5	4	3	2	1	0
	$P9_7$	$P9_6$	$P9_5$	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
Initial value	0	*	0	0	0	0	0	0
Read/Write	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Depends on the level of pin $P9_6$.

WSCR—Wait-State Control Register H'FFC2

Bit	7	6	5	4	3	2	1	0
	—	—	CKDBL	—	WMS1	WMS0	WC1	WC0
Initial value	1	1	0	1	0	0	0	0
Read/Write	—	—	R/W	—	R/W	R/W	R/W	R/W

Wait Count

0	0	No wait states inserted by wait-state controller
0	1	1 state inserted
1	0	2 states inserted
1	1	3 states inserted

Wait Mode Select

0	0	Programmable wait mode
0	1	No wait states inserted by wait-state controller
1	0	Pin wait mode
1	1	Pin auto-wait mode

Clock Double

0	Clock for supporting modules is not divided ($ø_P$ = ø)
1	Clock for supporting modules is divided by 2 ($ø_P$ = ø/2)

Bit	7	6	5	4	3	2	1	0
	RING	CMPF	CMPIE	LOAD	MARK	—	ICKS1	ICKS0
Initial value	0	0	0	1	1	1	0	0
Read/Write	R/W	R/(W)*	R/W	(W)	(W)	—	R/W	R/W

Internal Clock Source Select
See TCR under TMR0 and TMR1.

Pointer Mark

0	DTARB contents are copied to RLARB
1	No operation

Pointer Load

0	RLARB contents are copied to DTARB
1	No operation

Compare Interrupt Enable

0	Interrupt request (CMPI) by CMPF is disabled
1	Interrupt request (CMPI) by CMPF is enabled

Compare Interrupt Flag

0	Cleared by reading CMPF = 1, then writing 0 in CMPF
1	Ring buffer overrun error Set when DTARB contents match CPARB contents after being incremented by DTU cycle occurrence

Ring Buffer Mode

0	DTU channel B does not operate in ring buffer mode
1	DTU channel B operates in ring buffer mode

Note: * Software can write a 0 in bit 6 to clear the flag, but cannot write a 1 in this bit.

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	XRST	NMIEG	DPME	RAME
Initial value	0	0	0	0	1	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

RAM Enable

0	On-chip RAM is disabled.
1	On-chip RAM is enabled.

Dual-Port RAM Mode Enable

0	Not being set to slave mode.
1	Being set to slave mode.

NMI Edge

0	Falling edge of $\overline{\text{NMI}}$ is detected.
1	Rising edge of $\overline{\text{NMI}}$ is detected.

External Reset

0	The chip is reset by a watchdog timer overflow.
1	The chip is reset by an external reset signal.

Standby Timer Select

0	0	0	Clock settling time = 8,192 states
0	0	1	Clock settling time = 16,384 states
0	1	0	Clock settling time = 32,768 states
0	1	1	Clock settling time = 65,536 states
1	0	—	Clock settling time = 131,072 states
1	1	—	Prohibited

Software Standby

0	SLEEP instruction causes transition to sleep mode.
1	SLEEP instruction causes transition to software standby mode.

MDCR—Mode Control Register **H'FFC5**

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

Mode Select Bits

Value at mode pins.

Note: * Determined by inputs at pins MD_1 and MD_0.

ISCR—IRQ Sense Control Register **H'FFC6**

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

IRQ_0 to IRQ_7 Sense Control

0	$\overline{IRQ}_0$ to $\overline{IRQ}_7$ are level-sensed (active low).
1	$\overline{IRQ}_0$ to $\overline{IRQ}_7$ are edge-sensed (falling edge).

IER—IRQ Enable Register **H'FFC7**

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

IRQ_0 to IRQ_7 Enable

0	$\overline{IRQ}_0$ to $\overline{IRQ}_7$ are disabled.
1	$\overline{IRQ}_0$ to $\overline{IRQ}_7$ are enabled.

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

TCR			STCR		Description
CKS2	CKS1	CKS0	ICKS1	ICKS0	
0	0	0	—	—	Timer stopped
0	0	1	—	0	øP/8 internal clock, falling edge
0	0	1	—	1	øP/2 internal clock, falling edge
0	1	0	—	0	øP/64 internal clock, falling edge
0	1	0	—	1	øP/32 internal clock, falling edge
0	1	1	—	0	øP/1024 internal clock, falling edge
0	1	1	—	1	øP/256 internal clock, falling edge
1	0	0	—	—	Timer stopped
1	0	1	—	—	External clock, rising edge
1	1	0	—	—	External clock, falling edge
1	1	1	—	—	External clock, rising and falling edges

Counter Clear

0	0	Counter is not cleared.
0	1	Cleared by compare-match A.
1	0	Cleared by compare-match B.
1	1	Cleared on rising edge of external reset input.

Timer Overflow Interrupt Enable

0	Overflow interrupt request is disabled.
1	Overflow interrupt request is enabled.

Compare-Match Interrupt Enable A

0	Compare-match A interrupt request is disabled.
1	Compare-match A interrupt request is enabled.

Compare-Match Interrupt Enable B

0	Compare-match B interrupt request is disabled.
1	Compare-match B interrupt request is enabled.

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	PWME	OS3*2	OS2*2	OS1*2	OS0*2
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*1	R/(W)*1	R/(W)*1	R/W	R/W	R/W	R/W	R/W

Output Select

0	0	No change on compare-match A.
0	1	Output 0 on compare-match A.
1	0	Output 1 on compare-match A.
1	1	Invert (toggle) output on compare-match A.

Output Select

0	0	No change on compare-match B.
0	1	Output 0 on compare-match B.
1	0	Output 1 on compare-match B.
1	1	Invert (toggle) output on compare-match B.

PWM Mode Enable

0	Normal timer mode
1	PWM mode

Timer Overflow Flag

0	Cleared by reading OVF = 1, then writing 0 in OVF.
1	Set when TCNT changes from H'FF to H'00.

Compare-Match Flag A

0	Cleared by reading CMFA = 1, then writing 0 in CMFA.
1	Set when TCNT = TCORA.

Compare-Match Flag B

0	Cleared by reading CMFB = 1, then writing 0 in CMFB.
1	Set when TCNT = TCORB.

Notes: 1. Software can write a 0 in bits 7 to 5 to clear the flags, but cannot write a 1 in these bits.
2. When all four bits (OS3 to OS0) are cleared to 0, output is disabled.

TCORA—Time Constant Register A **H'FFCA** **TMR0**

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

The CMFA bit is set to 1 when TCORA = TCNT.

TCORB—Time Constant Register B **H'FFCB** **TMR0**

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

The CMFB bit is set to 1 when TCORB = TCNT.

TCNT—Timer Counter **H'FFCC** **TMR0**

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

Count value

Bit	7	6	5	4	3	2	1	0
	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
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

Next Data Enable 15 to 8

Bits 7 to 0 NDER15 to 8	Description
0	TPC outputs TP_{15} to TP_8 are disabled (NDR15 to NDR8 cannot be transferred to $P2_7$ to $P2_0$)
1	TPC outputs TP_{15} to TP_8 are enabled (NDR15 to NDR8 can be transferred to $P2_7$ to $P2_0$)

- Same Trigger for TPC Output Groups 2 and 3

  Address H'FFCE

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
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
	Next output data for TPC output group 3				Next output data for TPC output group 2			

  Address H'FFD6

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

- Different Triggers for TPC Output Groups 2 and 3

  Address H'FFCE

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—
	Next output data for TPC output group 3							

  Address H'FFD6

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W
					Next output data for TPC output group 2			

- Same Trigger for TPC Output Groups 0 and 1

  Address H'FFCF

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
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

Next output data for TPC output group 1

Next output data for TPC output group 0

  Address H'FFD7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

- Different Triggers for TPC Output Groups 0 and 1

  Address H'FFCF

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Next output data for TPC output group 1

  Address H'FFD7

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Next output data for TPC output group 0

Bit	7	6	5	4	3	2	1	0
	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select

TCR			STCR		Description
CKS2	CKS1	CKS0	ICKS1	ICKS0	
0	0	0	—	—	Timer stopped
0	0	1	0	—	øP/8 internal clock, falling edge
0	0	1	1	—	øP/2 internal clock, falling edge
0	1	0	0	—	øP/64 internal clock, falling edge
0	1	0	1	—	øP/128 internal clock, falling edge
0	1	1	0	—	øP/1024 internal clock, falling edge
0	1	1	1	—	øP/2048 internal clock, falling edge
1	0	0	—	—	Timer stopped
1	0	1	—	—	External clock, rising edge
1	1	0	—	—	External clock, falling edge
1	1	1	—	—	External clock, rising and falling edges

Counter Clear

0	0	Counter is not cleared.
0	1	Cleared by compare-match A.
1	0	Cleared by compare-match B.
1	1	Cleared on rising edge of external reset input.

Timer Overflow Interrupt Enable

0	Overflow interrupt request is disabled.
1	Overflow interrupt request is enabled.

Compare-Match Interrupt Enable A

0	Compare-match A interrupt request is disabled.
1	Compare-match A interrupt request is enabled.

Compare-Match Interrupt Enable B

0	Compare-match B interrupt request is disabled.
1	Compare-match B interrupt request is enabled.

TCSR—Timer Control/Status Register **H'FFD1** **TMR1**

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	PWME	OS3*2	OS2*2	OS1*2	OS0*2
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*1	R/(W)*1	R/(W)*1	R/W	R/W	R/W	R/W	R/W

Notes: Bit functions are the same as for TMR0.
1. Software can write a 0 in bits 7 to 5 to clear the flags, but cannot write a 1 in these bits.
2. When all four bits (OS3 to OS0) are cleared to 0, output is disabled.

TCORA—Time Constant Register A **H'FFD2** **TMR1**

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

Note: Bit functions are the same as for TMR0.

TCORB—Time Constant Register B **H'FFD3** **TMR1**

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

Note: Bit functions are the same as for TMR0.

TCNT—Timer Counter **H'FFD4** **TMR1**

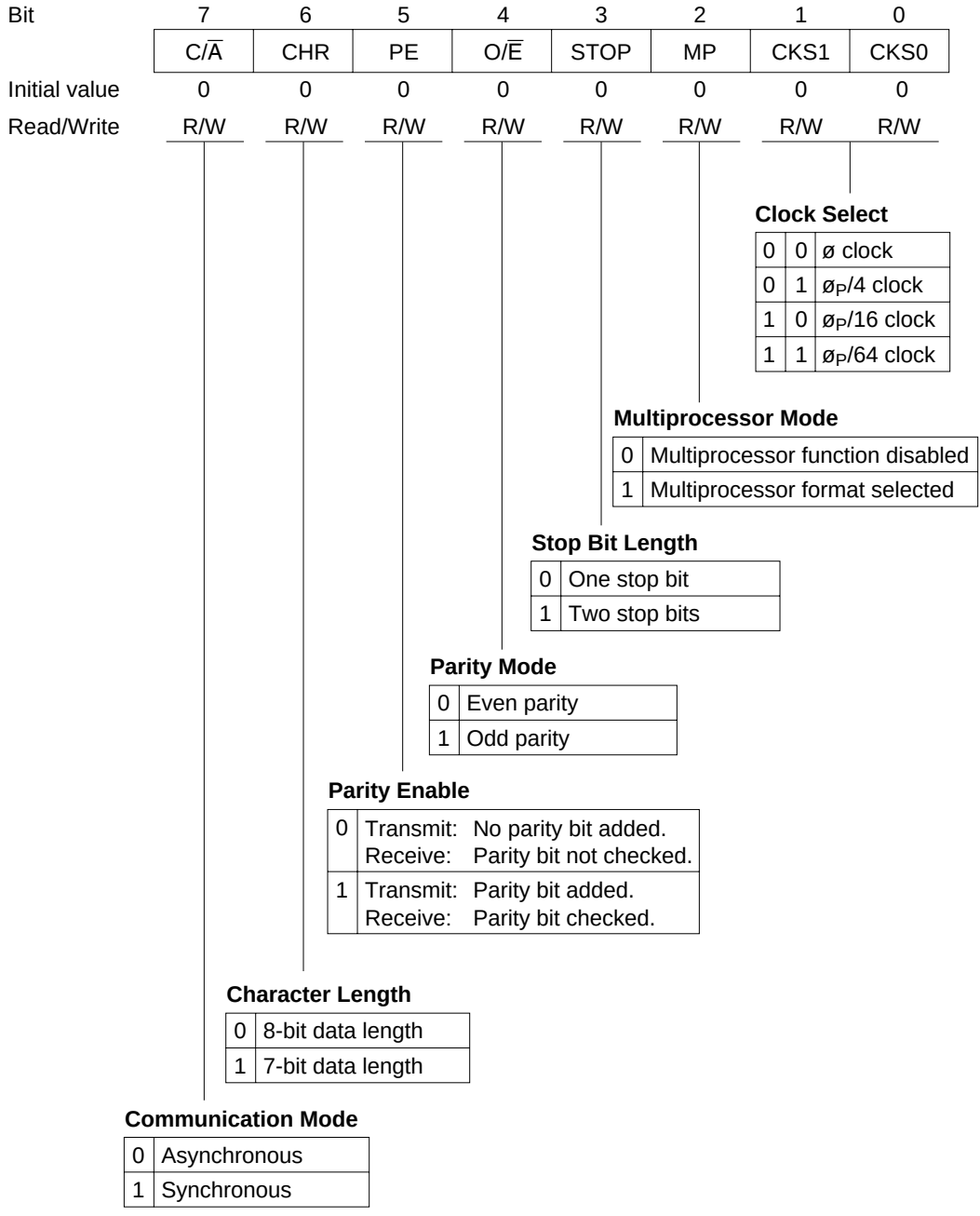
Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Bit functions are the same as for TMR0.

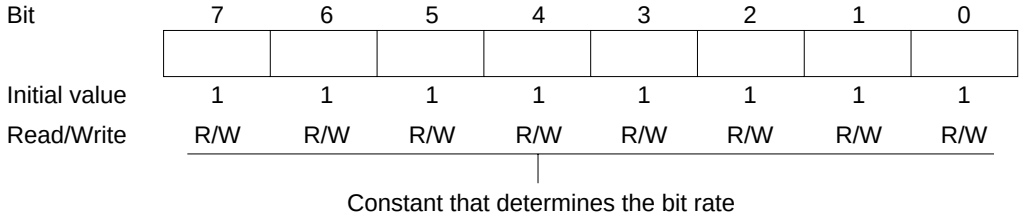
Bit	7	6	5	4	3	2	1	0
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next Data Enable 7 to 0

Bits 7 to 0 NDER7 to 0	Description
0	TPC outputs TP_7 to TP_0 are disabled (NDR7 to NDR0 cannot be transferred to $P1_7$ to $P1_0$)
1	TPC outputs TP_7 to TP_0 are enabled (NDR7 to NDR0 can be transferred to $P1_7$ to $P1_0$)



Note: Bit functions are the same as for SCI1.



Note: Bit functions are the same as for SCI1.

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 0

0	The SCK pin is not used by the SCI.
1	The SCK pin is used for serial clock output.

Clock Enable 1

0	Internal clock
1	External clock

Transmit End Interrupt Enable

0	TSR-empty interrupt request is disabled.
1	TSR-empty interrupt request is enabled.

Multiprocessor Interrupt Enable

0	Multiprocessor receive interrupt function is disabled.
1	Multiprocessor receive interrupt function is enabled.

Receive Enable

0	Receive disabled
1	Receive enabled

Transmit Enable

0	Transmit disabled
1	Transmit enabled

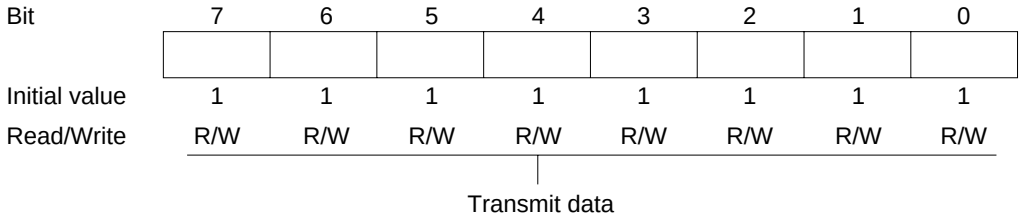
Receive Interrupt Enable

0	Receive interrupt and receive error interrupt requests are disabled.
1	Receive interrupt and receive error interrupt requests are enabled.

Transmit Interrupt Enable

0	TDR-empty interrupt request is disabled.
1	TDR-empty interrupt request is enabled.

Note: Bit functions are the same as for SCI1.



Note: Bit functions are the same as for SCI1.

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Multiprocessor Bit Transfer

0	Multiprocessor bit = 0 in transmit data.
1	Multiprocessor bit = 1 in transmit data.

Multiprocessor Bit

0	Multiprocessor bit = 0 in receive data.
1	Multiprocessor bit = 1 in receive data.

Transmit End

0	Cleared by reading TDRE = 1, then writing 0 in TDRE.
1	Set to 1 when TE = 0, or when TDRE = 1 at the end of character transmission.

Parity Error

0	Cleared by reading PER = 1, then writing 0 in PER.
1	Set when a parity error occurs (parity of receive data does not match parity selected by O/$\overline{E}$ bit in SMR).

Framing Error

0	Cleared by reading FER = 1, then writing 0 in FER.
1	Set when a framing error occurs (stop bit is 0).

Overrun Error

0	Cleared by reading ORER = 1, then writing 0 in ORER.
1	Set when an overrun error occurs (next data is completely received while RDRF bit is set to 1).

Receive Data Register Full

0	Cleared by reading RDRF = 1, then writing 0 in RDRF.
1	Set when one character is received normally and transferred from RSR to RDR.

Transmit Data Register Empty

0	Cleared by reading TDRE = 1, then writing 0 in TDRE.
1	Set when: 1. Data is transferred from TDR to TSR. 2. TE is cleared while TDRE = 0.

Note: * Software can write a 0 in bits 7 to 3 to clear the flags, but cannot write a 1 in these bits. Bit functions are the same as for SCI1.

RDR—Receive Data Register **H'FFDD** **SCI0**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Receive data

Note: Bit functions are the same as for SCI1.

SCMR—Serial Communication Mode Register **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 Mode Select

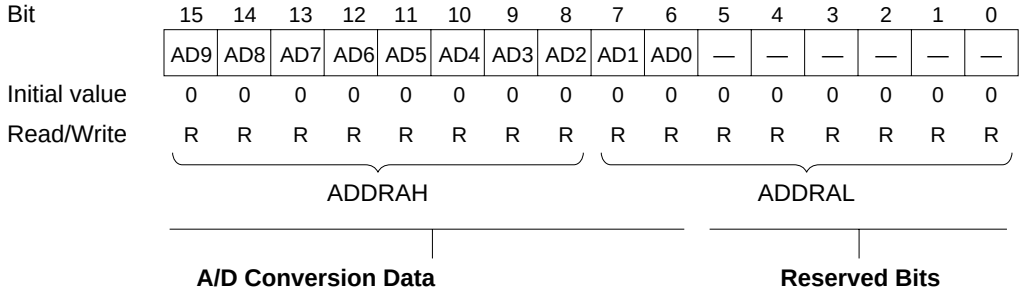
0	Normal SCI mode
1	Reserved mode

Data Invert

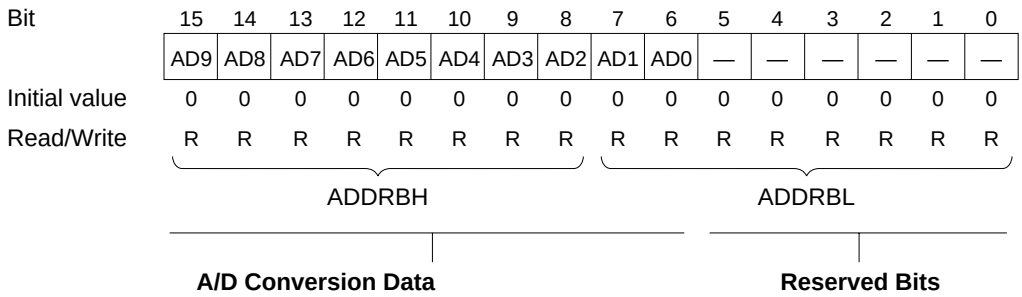
0	TDR contents are transmitted as they are. Receive data is stored in RDR as it is.
1	TDR contents are inverted before being transmitted. Receive data is stored in RDR in inverted form.

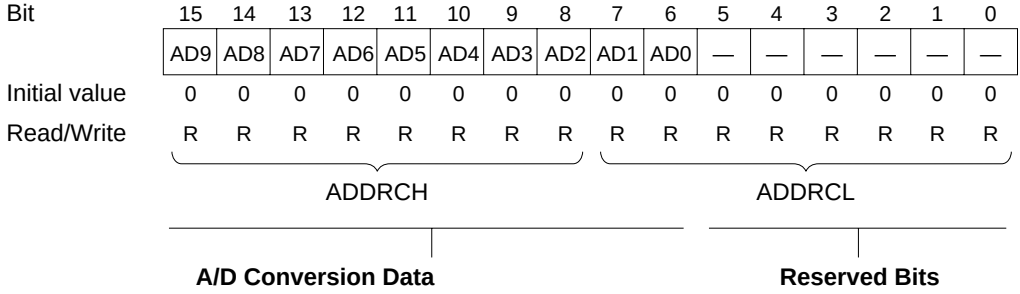
Data Transfer Direction

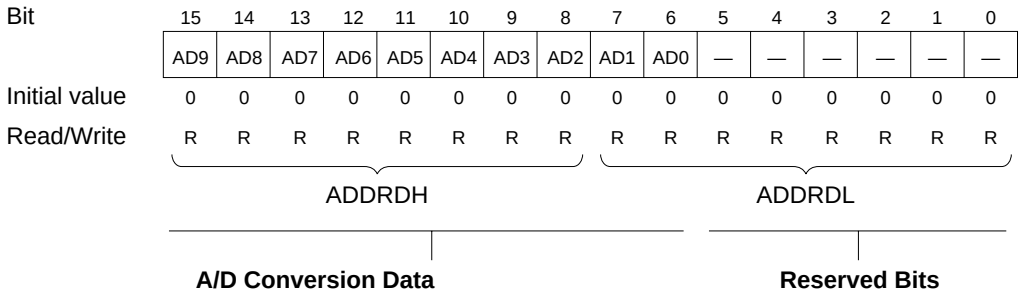
0	TDR contents are transmitted LSB-first. Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first. Receive data is stored in RDR MSB-first.



ADDRB (H and L)—A/D Data Register B **H'FFE2, H'FFE3** **A/D**




Bit
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
AD9 AD8 AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0 — — — — — —
Initial value
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Read/Write
R R R R R R R R R R R R R R R R
ADDRCH
ADDRCL
A/D Conversion Data
10-bit data giving an A/D conversion result
Reserved Bits


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
ADDRDH
ADDRDL
A/D Conversion Data
10-bit data giving an A/D conversion result
Reserved Bits

Bit	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	SCAN	CKS	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel Select

Group Selection	Channel Selection		Description	
CH2	CH1	CH0	Single Mode	Scan Mode
0	0	0	AN_0	AN_0
		1	AN_1	AN_0, AN_1
	1	0	AN_2	AN_0 to AN_2
		1	AN_3	AN_0 to AN_3
1	0	0	AN_4	AN_4
		1	AN_5	AN_4 to AN_5
	1	0	AN_6	AN_4 to AN_6
		1	AN_7	AN_4 to AN_7

Clock Select

0	Conversion time = 266 states (max)
1	Conversion time = 134 states (max)

Scan Mode

0	Single mode
1	Scan mode

A/D Start

0	A/D conversion is stopped
1	1. Single mode: A/D conversion starts; ADST is automatically cleared to 0 when conversion ends 2. Scan mode: A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, by a reset, or by a transition to standby mode

A/D Interrupt Enable

0	A/D end interrupt request is disabled
1	A/D end interrupt request is enabled

A/D End Flag

0	Cleared by reading ADF = 1, then writing 0 in ADF
1	Set when A/D conversion ends in single mode. Set when A/D conversion ends on all selected channels in scan mode.

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

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

Trigger Enable

0	A/D conversion cannot be externally triggered
1	A/D conversion starts at the fall of the external trigger signal ($\overline{\text{ADTRG}}$)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Group 0 Non-Overlap

0	Normal operation in TPC group 0, output values changing at selected FRT compare match
1	Non-overlapping operation in TPC group 0, using selected FRT compare match A and B

Group 1 Non-Overlap

0	Normal operation in TPC group 1, output values changing at selected FRT compare match
1	Non-overlapping operation in TPC group 1, using selected FRT compare match A and B

Group 2 Non-Overlap

0	Normal operation in TPC group 2, output values changing at selected FRT compare match
1	Non-overlapping operation in TPC group 2, using selected FRT compare match A and B

Group 3 Non-Overlap

0	Normal operation in TPC group 3, output values changing at selected FRT compare match
1	Non-overlapping operation in TPC group 3, using selected FRT compare match A and B

Bit	7	6	5	4	3	2	1	0
	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
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

Group 0 Compare Match Select 1 and 0

Bit 1	Bit 0	Output Trigger Selection
G0CMS1	G0CMS0	
0	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare match A in FRT1
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match A in FRT0
1	0	TPC output group 0 (TP_3 to TP_0) is triggered by compare match B in FRT1 (and can be non-overlapped)
	1	TPC output group 0 (TP_3 to TP_0) is triggered by compare match B in FRT0 (and can be non-overlapped)

Group 1 Compare Match Select 1 and 0

Bit 3	Bit 2	Output Trigger Selection
G1CMS1	G1CMS0	
0	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare match A in FRT1
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match A in FRT0
1	0	TPC output group 1 (TP_7 to TP_4) is triggered by compare match B in FRT1 (and can be non-overlapped)
	1	TPC output group 1 (TP_7 to TP_4) is triggered by compare match B in FRT0 (and can be non-overlapped)

Group 2 Compare Match Select 1 and 0

Bit 5	Bit 4	Output Trigger Selection
G2CMS1	G2CMS0	
0	0	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match A in FRT1
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match A in FRT0
1	0	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match B in FRT1 (and can be non-overlapped)
	1	TPC output group 2 (TP_{11} to TP_8) is triggered by compare match B in FRT0 (and can be non-overlapped)

Group 3 Compare Match Select 1 and 0

Bit 7	Bit 6	Output Trigger Selection
G3CMS1	G3CMS0	
0	0	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare match A in FRT1
	1	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare match A in FRT0
1	0	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare match B in FRT1 (and can be non-overlapped)
	1	TPC output group 3 (TP_{15} to TP_{12}) is triggered by compare match B in FRT0 (and can be non-overlapped)

Bit	7	6	5	4	3	2	1	0
	QREF	EWRQ	EWAKAR	ERAKAR	MWEF	MREF	EMWI	EMRI
Initial value	0	0	0	0	0	1	0	0
Read/Write								
Internal CPU	R	R	R	R	R/(W)	R/(W)	R/W	R/W
Master CPU	R	R/W	R/W	R/W	R	R	R	R

Enable Master Read Interrupt

0	Disables interrupt request (MREI) by MREF
1	Enables interrupt request (MREI) by MREF

Enable Master Write Interrupt

0	Disables interrupt request (MWEI) by MWEF
1	Enables interrupt request (MWEI) by MWEF

Master Read End Flag[*2]

Master Write End Flag[*2]

Enable Write Acknowledge and Request[*2]
(Enables operation of the $\overline{\text{RDY}}$ pin by master write access)

Enable Read Acknowledge and Request[*2]
(Enables operation of the $\overline{\text{RDY}}$ pin by master read access)

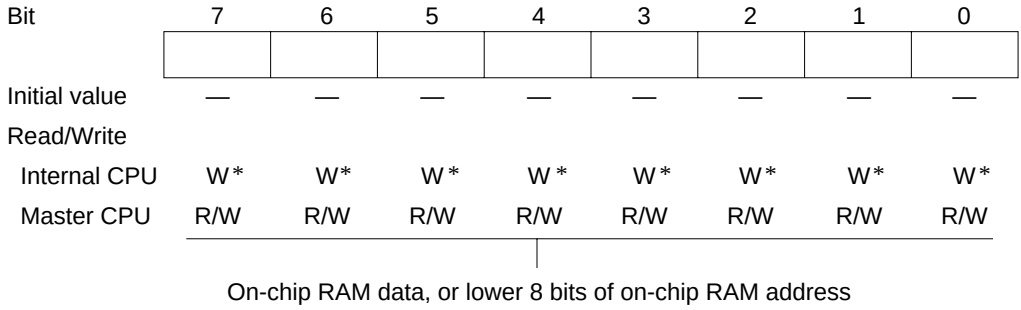
Enable Wait Request[*2]
(Enables operation of the $\overline{\text{WRQ}}$ pin)

Query Read End Flag

0	DPDRRQ contains data Cleared when DTC writes on-chip RAM data in DPDRRQ
1	DPDRRQ contains the lower byte of an on-chip RAM address Set when master CPU writes lower byte of an on-chip RAM address in DPDRRQ

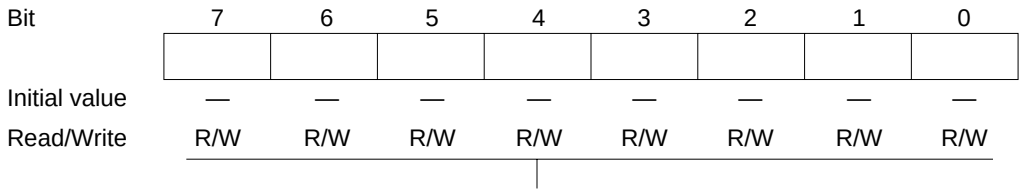
Notes: 1. Value of register select inputs (RS2 to RS0) by which the master CPU selects a PBI register.
2. For details see section 5, DTU.

DPDRRQ—DPRAM Data Register Query Read — (001 from Master) DTU

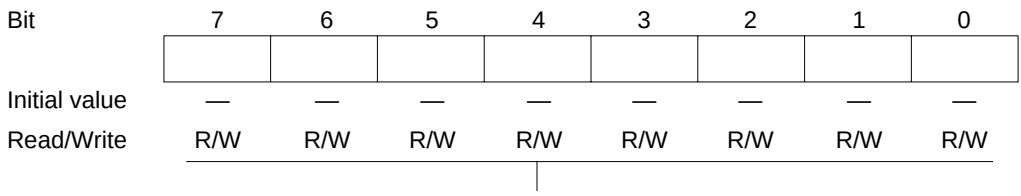


Note: * Transferred automatically from on-chip RAM by DTU.

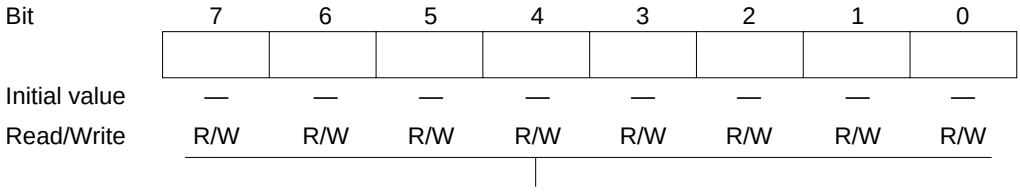
RLARA—Reload Address Register A H'FFF2 (— from Master) DTU



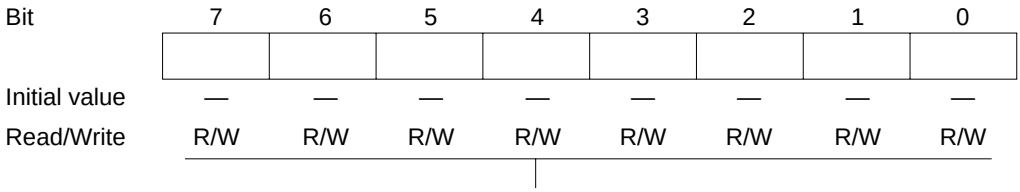
RLARB—Reload Address Register B H'FFF3 (— from Master) DTU



CPARB—Compare Address Register B **H'FFF4 (— from Master)** **DTU**



DTARH—Data Transfer Address Register H **H'FFF5 (— from Master)** **DTU**



Bit	7	6	5	4	3	2	1	0
	HSCE	DPEA	DPEB	RPEA	RPEB	RPEC	—	—
Initial value	0	0	0	0	0	0	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	—	—

Repeat Enable

0	Transfer in normal mode
1	Transfer in repeat mode

DPRAM Enable

0	I/O transfer or direct word mode
1	Bound buffer mode

Parallel Handshake Enable

0	Handshake mode
1	DPRAM mode

Note: The DPME, HSCE, DPEA, and DPEB bits combine to specify the following modes.

DTC Operating Mode Settings

DPME	Bit 7 HSCE	Bit 6 DPEA	Bit 5 DPEB	Hand-shake Mode	Query Buffer Operation (DTC Channel R)	Bound Buffer Mode (DTC Channel A)	Bound Buffer Mode (DTC Channel B)	Direct Word Mode (DPDRRH/L)	Direct Word Mode (DPDRWH/L)
1	1	—	—	○	×	I/O transfer	I/O transfer	Handshake	Handshake
	0	0	0	×	○ (read)	I/O transfer	I/O transfer	○ (read)	○ (write)
		1	0	×	○ (read)	○ (read)	I/O transfer	Bound buffer	○ (write)
		0	1	×	○ (read)	I/O transfer	○ (write)	○ (read)	Bound buffer
		1	1	×	○ (read)	○ (read)	○ (write)	Bound buffer	Bound buffer
0	—	—	—	×	×	I/O transfer	I/O transfer	×	×

Note: For I/O transfer, set the DTE bit to 1 in DTCRA, DTCRB, or DTCRC.
Handshake mode operation is only supported in single-chip mode.
Do not set bit HSCE to 1 in expanded mode.

Bit	7	6	5	4	3	2	1	0
	DTE	DTIE	BUD2	BUD1	BUD0	SOS2	SOS1	SOS0
Initial value	0	0	0	0	0	0	0	0
Read/Write								
I/O transfer	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bound buffer								
Internal CPU	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Master CPU	0	R	R	R	R	R	R	R

Source Select

SOS2	SOS1	SOS0	Interrupt Source	Module	Transfer	Clearing of Source
0	0	1	RXI0	SCI0	RDR → RAM (byte)	○
0	1	0	TXI0	SCI0	RAM → TDR (byte)	○
0	1	1	ADI	ADC	ADDRA → RAM (word)	○
1	0	0	OCIA1	FRT1	RAM → OCRA (word)	○
1	0	1	OCIA1	FRT1	RAM → OCRA (word)	×
1	1	0	OCIB1	FRT1	RAM → OCRB (word)	×
1	1	1	—	—	Reserved	—

Boundary

BUD2	BUD1	BUD0	DTAR Overflow Timing	Max. Bytes Transferred
0	0	0	End of each byte transfer	1
0	0	1	Carry from bit 0 to bit 1	2
0	1	0	Carry from bit 1 to bit 2	4
0	1	1	Carry from bit 2 to bit 3	8
1	0	0	Carry from bit 3 to bit 4	16
1	0	1	Carry from bit 4 to bit 5	32
1	1	0	Carry from bit 5 to bit 6	64
1	1	1	Carry from bit 6 to bit 7	128

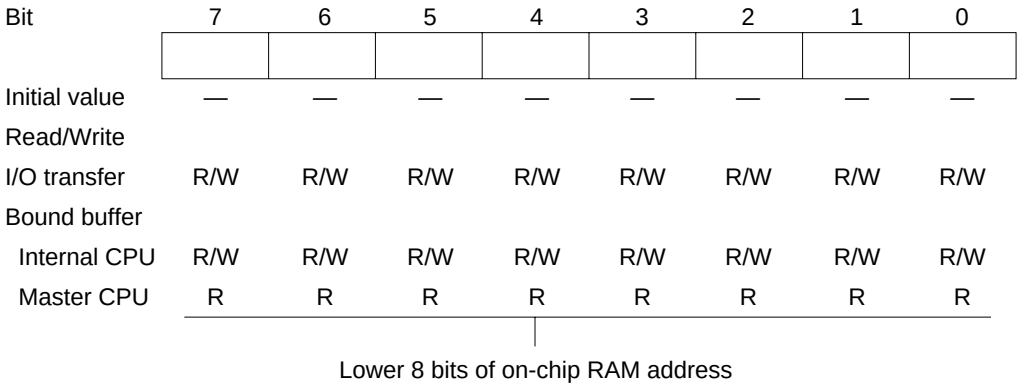
Data Transfer Interrupt Enable

0	Disables the interrupt requested when the DTE bit is cleared to 0 (DTIA)
1	Enables the interrupt requested when the DTE bit is cleared to 0 (DTIA)

Data Transfer Enable

0	Indicates that I/O transfer is halted Cleared when 0 is written in DTE, or when a transfer terminates at the boundary in normal mode
1	Indicates that I/O transfer is in progress Set by reading DTCR while DTE = 0, then writing 1 in DTE

Note: 0: Always read as 0.


Bit
7
6
5
4
3
2
1
0
Initial value
—
—
—
—
—
—
—
—
Read/Write
I/O transfer
R/W
R/W
R/W
R/W
R/W
R/W
R/W
R/W
Bound buffer
Internal CPU
R/W
R/W
R/W
R/W
R/W
R/W
R/W
R/W
Master CPU
R
R
R
R
R
R
R
R
Lower 8 bits of on-chip RAM address

Bit	7	6	5	4	3	2	1	0
	DTE	DTIE	BUD2	BUD1	BUD0	SOS2	SOS1	SOS0
Initial value	0	0	0	0	0	0	0	0
Read/Write								
I/O transfer	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bound buffer								
Internal CPU	0	0	R	R	R	0	0	0
Master CPU	—	—	W	W	W	—	—	—

Source Select

SOS2	SOS1	SOS0	Interrupt Source	Module	Transfer	Clearing of Source
0	0	1	RXI0	SCI0	RDR → RAM (byte)	○
0	1	0	TXI0	SCI0	RAM → TDR (byte)	○
0	1	1	ADI	ADC	ADDRA → RAM (word)	○
1	0	0	OCIA1	TPC	RAM → NDRB (word)	○
1	0	1	OCIA1	TPC	RAM → NDRA (byte)	○
1	1	0	OCIA1	FRT1	RAM → OCRA (word)	○
1	1	1	OCIA1	FRT1	RAM → OCRA (word)	×

Boundary

BUD2	BUD1	BUD0	DTAR Overflow Timing	Max. Bytes Transferred
0	0	0	End of each byte transfer	1
0	0	1	Carry from bit 0 to bit 1	2
0	1	0	Carry from bit 1 to bit 2	4
0	1	1	Carry from bit 2 to bit 3	8
1	0	0	Carry from bit 3 to bit 4	16
1	0	1	Carry from bit 4 to bit 5	32
1	1	0	Carry from bit 5 to bit 6	64
1	1	1	Carry from bit 6 to bit 7	128

Data Transfer Interrupt Enable

0	Disables the interrupt requested when the DTE bit is cleared to 0 (DTIB)
1	Enables the interrupt requested when the DTE bit is cleared to 0 (DTIB)

Data Transfer Enable

0	Indicates that I/O transfer is halted Cleared when 0 is written in DTE, or when a transfer terminates at the boundary in normal mode
1	Indicates that I/O transfer is in progress Set by reading DTCR while DTE = 0, then writing 1 in DTE

Notes: — : Cannot be used. Cannot be modified.
0 : Always read as 0.

Bit	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—
Read/Write								
I/O transfer	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bound buffer								
Internal CPU	R	R	R	R	R	R	R	R
Master CPU	W	W	W	W	W	W	W	W

Lower 8 bits of on-chip RAM address

Bit	7	6	5	4	3	2	1	0
	DTE	DTIE	BUD2	BUD1	BUD0	SOS2	SOS1	SOS0
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

Source Select

SOS2	SOS1	SOS0	Interrupt Source	Module	Transfer	Clearing of Source
0	0	1	RXI0	SCI0	RDR → RAM (byte)	○
0	1	0	TXI0	SCI0	RAM → TDR (byte)	○
0	1	1	ADI	ADC	ADDRA → RAM (word)	○
1	0	0	OCIB1	TPC	RAM → NDRB (word)	○
1	0	1	OCIA1	TPC	RAM → NDRB (word)	○
1	1	0	OCIB1	TPC	RAM → NDRA (byte)	○
1	1	1	OCIA1	TPC	RAM → NDRA (byte)	○

Boundary

BUD2	BUD1	BUD0	DTAR Overflow Timing	Max. Bytes Transferred
0	0	0	End of each byte transfer	1
0	0	1	Carry from bit 0 to bit 1	2
0	1	0	Carry from bit 1 to bit 2	4
0	1	1	Carry from bit 2 to bit 3	8
1	0	0	Carry from bit 3 to bit 4	16
1	0	1	Carry from bit 4 to bit 5	32
1	1	0	Carry from bit 5 to bit 6	64
1	1	1	Carry from bit 6 to bit 7	128

Data Transfer Interrupt Enable

0	Disables the interrupt requested when the DTE bit is cleared to 0 (DTIC)
1	Enables the interrupt requested when the DTE bit is cleared to 0 (DTIC)

Data Transfer Enable

0	Indicates that I/O transfer is halted Cleared when 0 is written in DTE, or when a transfer terminates at the boundary in normal mode
1	Indicates that I/O transfer is in progress Set by reading DTCR while DTE = 0, then writing 1 in DTE

DTARC—Data Transfer Address Register C **H'FFFB (— from Master)** **DTU**

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

Lower 8 bits of on-chip RAM address

DPDRWH—DPRAM Data Register Write H **H'FFFC (100 from Master)** **DTU**

Bit	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—
Read/Write								
Bound buffer mode								
On-chip CPU	R *	R *	R *	R *	R *	R *	R *	R *
Master CPU	W	W	W	W	W	W	W	W
Direct word mode								
Internal CPU	R	R	R	R	R	R	R	R
Master CPU	W	W	W	W	W	W	W	W

Data register reserved for write access by master CPU.

Note: * Transferred to on-chip RAM automatically by the DTU.

DPDRWL—DPRAM Data Register Write L **H'FFFD (101 from Master)** **DTU**

Bit	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—
Read/Write								
Bound buffer mode								
On-chip CPU	R*1	R*1	R*1	R*1	R*1	R*1	R*1	R*1
Master CPU	W	W	W	W	W	W	W	W
Direct word mode								
Internal CPU	R	R	R	R	R	R	R	R
Master CPU	W	W	W	W	W	W	W	W
Handshake mode								
Internal CPU	R	R	R	R	R	R	R	R
Master CPU	W*2	W*2	W*2	W*2	W*2	W*2	W*2	W*2

Data register reserved for write access by master CPU.

Notes: 1. Transferred to on-chip RAM automatically by the DTU.
2. Data on the DDB lines is latched at the falling edge of the $\overline{WE}$ input.

DPDRRH—DPRAM Data Register Read H **H'FFFE (110 from Master)** **DTU**

Bit	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—
Read/Write								
Bound buffer mode								
On-chip CPU	W*	W*	W*	W*	W*	W*	W*	W*
Master CPU	R	R	R	R	R	R	R	R
Direct word mode								
Internal CPU	W	W	W	W	W	W	W	W
Master CPU	R	R	R	R	R	R	R	R

Data register reserved for read access by the master CPU.

Note: * Transferred from on-chip RAM automatically by the DTU.

Bit	7	6	5	4	3	2	1	0
Initial value	—	—	—	—	—	—	—	—
Read/Write								
Bound buffer mode								
On-chip CPU	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1
Master CPU	R	R	R	R	R	R	R	R
Direct word mode								
Internal CPU	W	W	W	W	W	W	W	W
Master CPU	R	R	R	R	R	R	R	R
Handshake mode								
Internal CPU	W	W	W	W	W	W	W	W
Master CPU	R*2	R*2	R*2	R*2	R*2	R*2	R*2	R*2

Data register reserved for read access by the master CPU.

Notes: 1. Transferred from on-chip RAM automatically by the DTU.
2. Data is output on the DDB lines when the $\overline{OE}$ input is low.

Appendix C I/O Port Block Diagrams

C.1 Port 1 Block Diagram

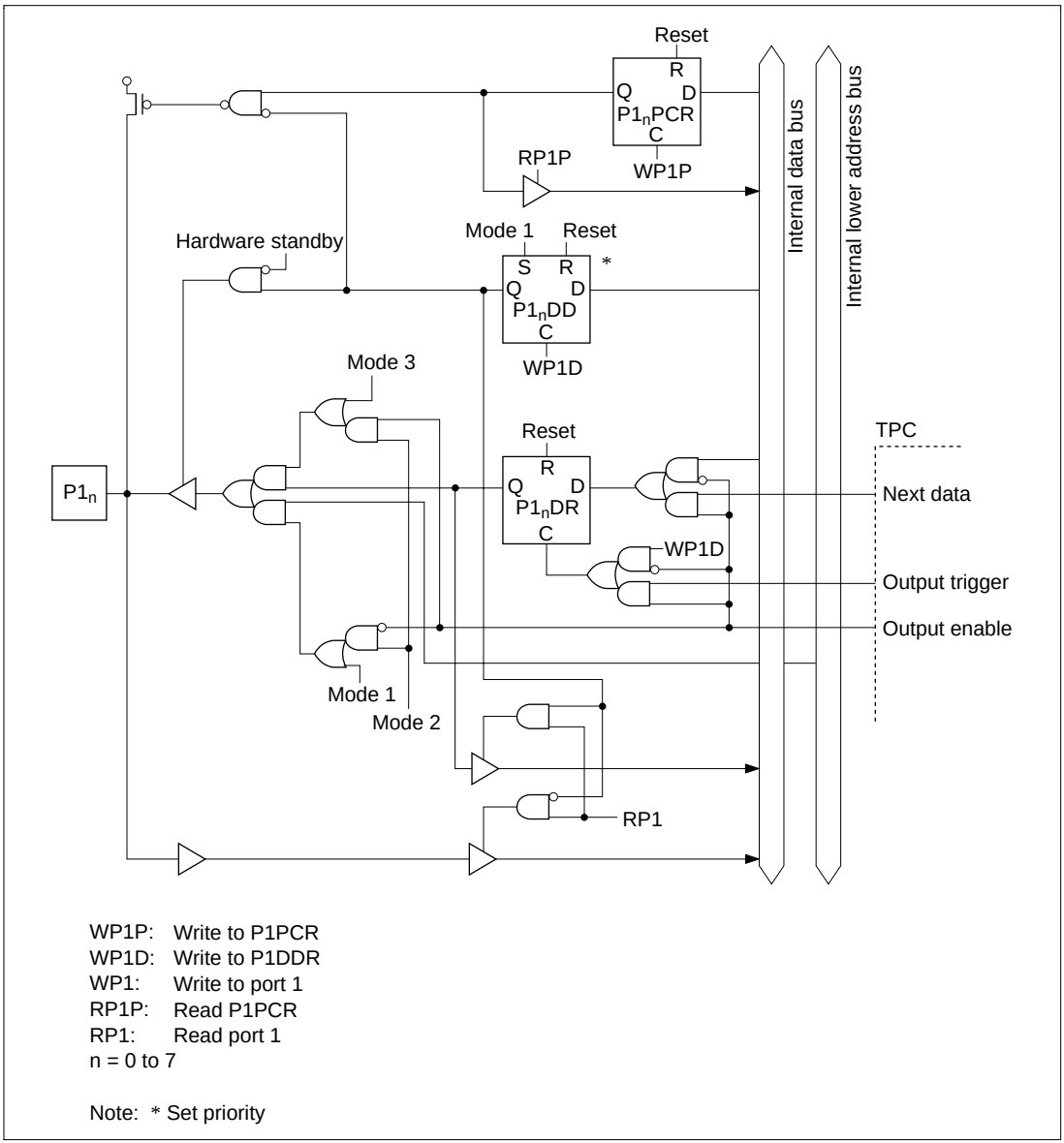
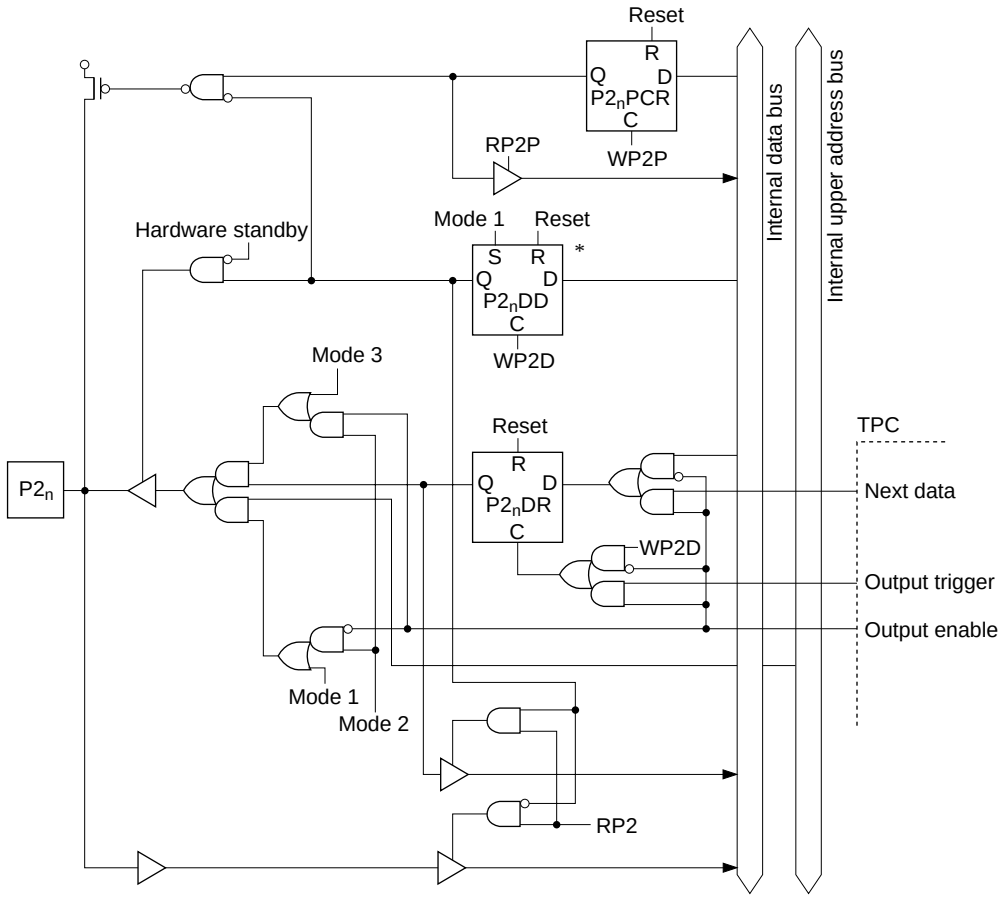


Figure C.1 Port 1 Block Diagram

C.2 Port 2 Block Diagram

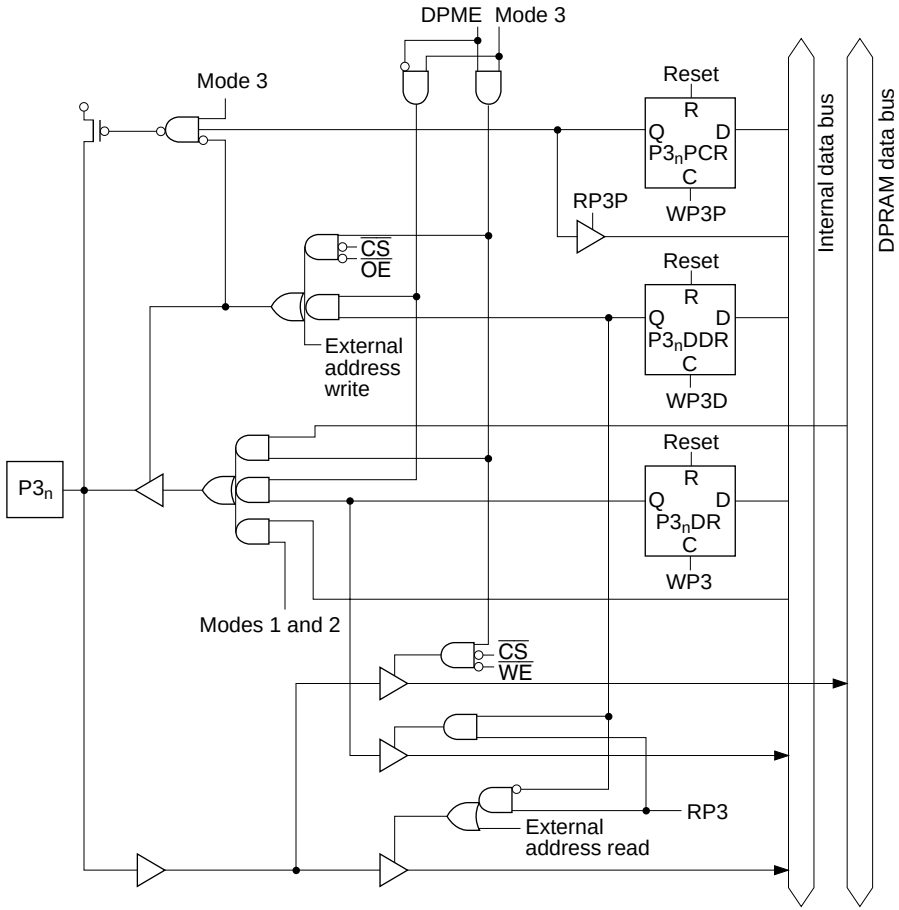


WP2P: Write to P2PCR
WP2D: Write to P2DDR
WP2: Write to port 2
RP2P: Read P2PCR
RP2: Read port 2
n = 0 to 7

Note: * Set priority

Figure C.2 Port 2 Block Diagram

C.3 Port 3 Block Diagram



WP3P: Write to P3PCR
WP3D: Write to P3DDR
WP3: Write to port 3
RP3P: Read P3PCR
RP3: Read port 3
n = 0 to 7

Note: When HSCE = 1, $\overline{CS}$ is internally fixed to 0.

Figure C.3 Port 3 Block Diagram

C.4 Port 4 Block Diagrams

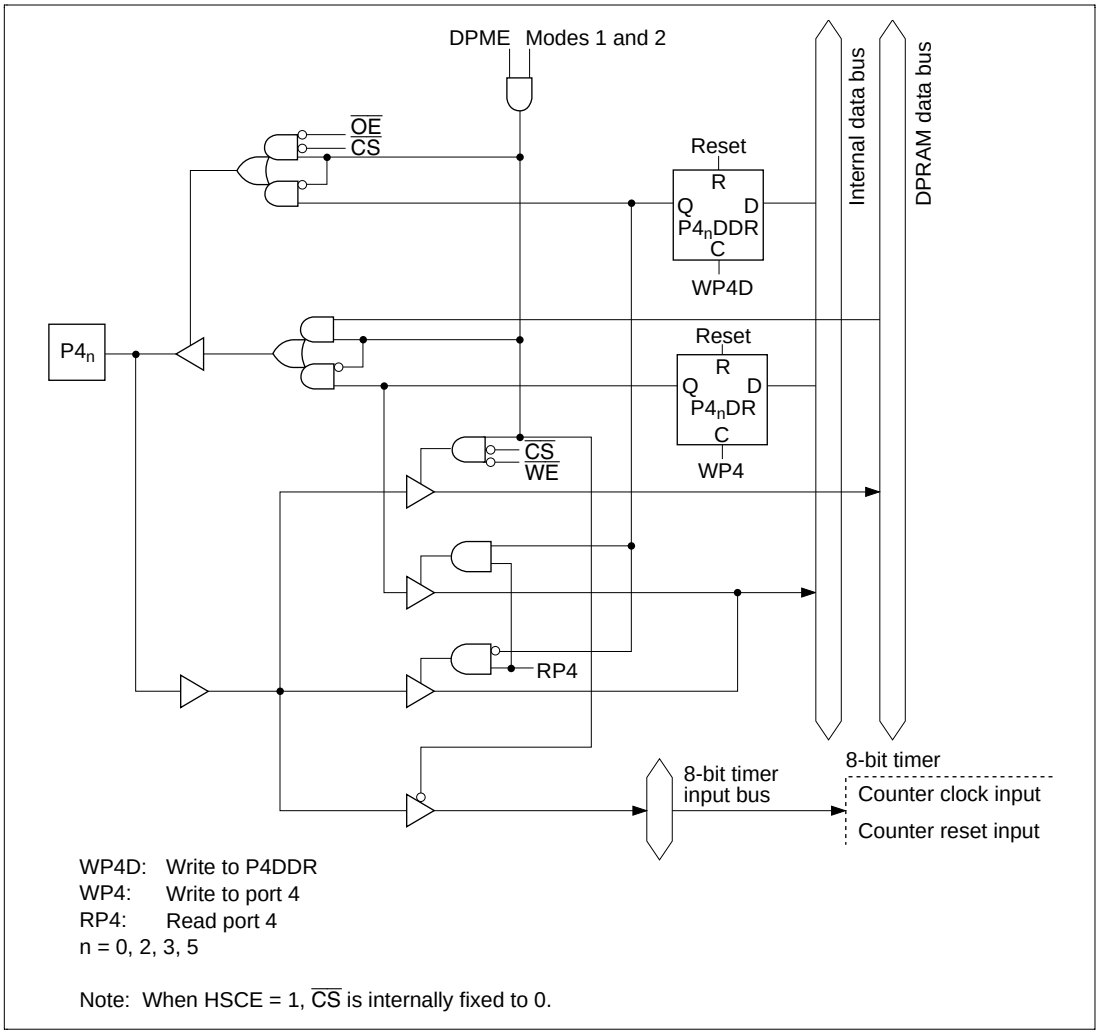
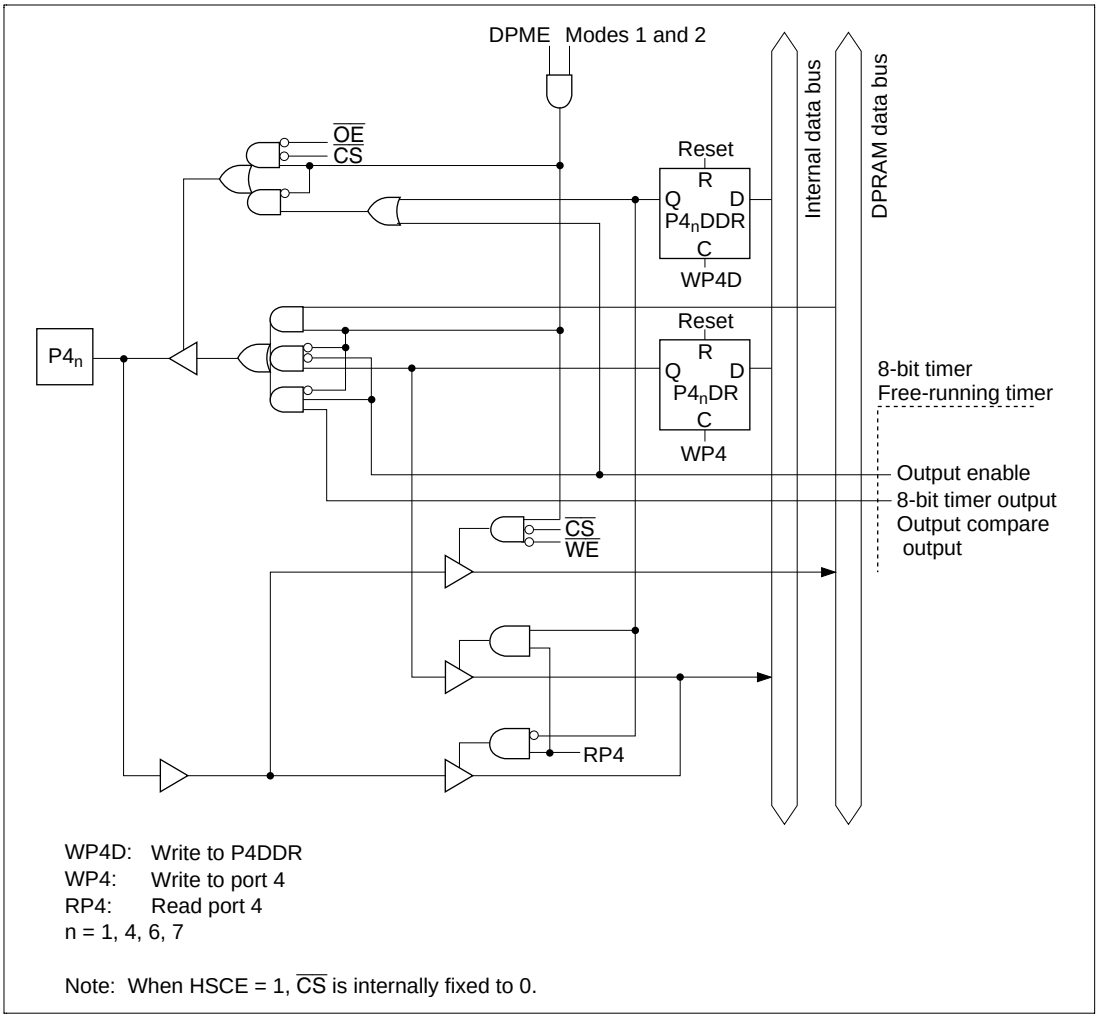


Figure C.4 (a) Port 4 Block Diagram (Pins $P4_0$, $P4_2$, $P4_3$, $P4_5$)



WP4D: Write to P4DDR
WP4: Write to port 4
RP4: Read port 4
n = 1, 4, 6, 7

Note: When HSCE = 1, $\overline{CS}$ is internally fixed to 0.

Figure C.4 (b) Port 4 Block Diagram (Pins $P4_1$, $P4_4$, $P4_6$, $P4_7$)

C.5 Port 5 Block Diagrams

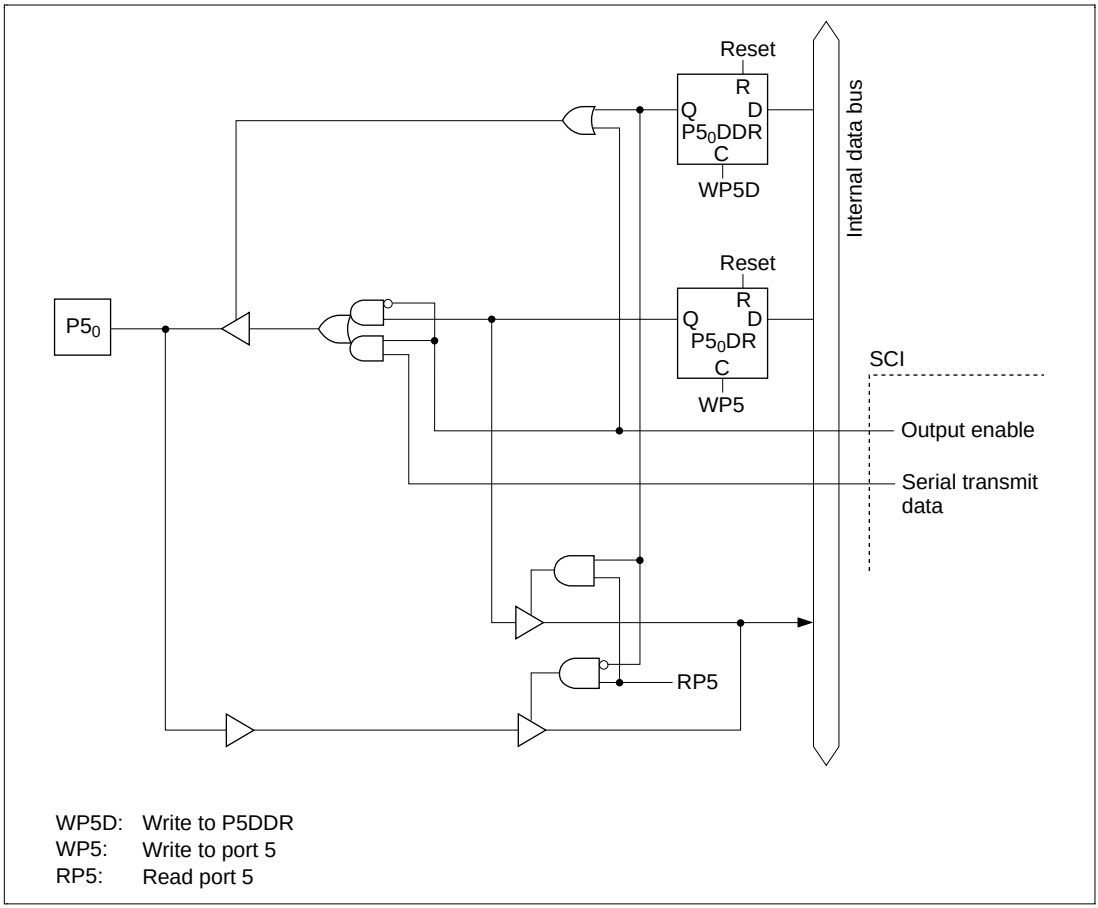


Figure C.5 (a) Port 5 Block Diagram (Pin $P5_0$)

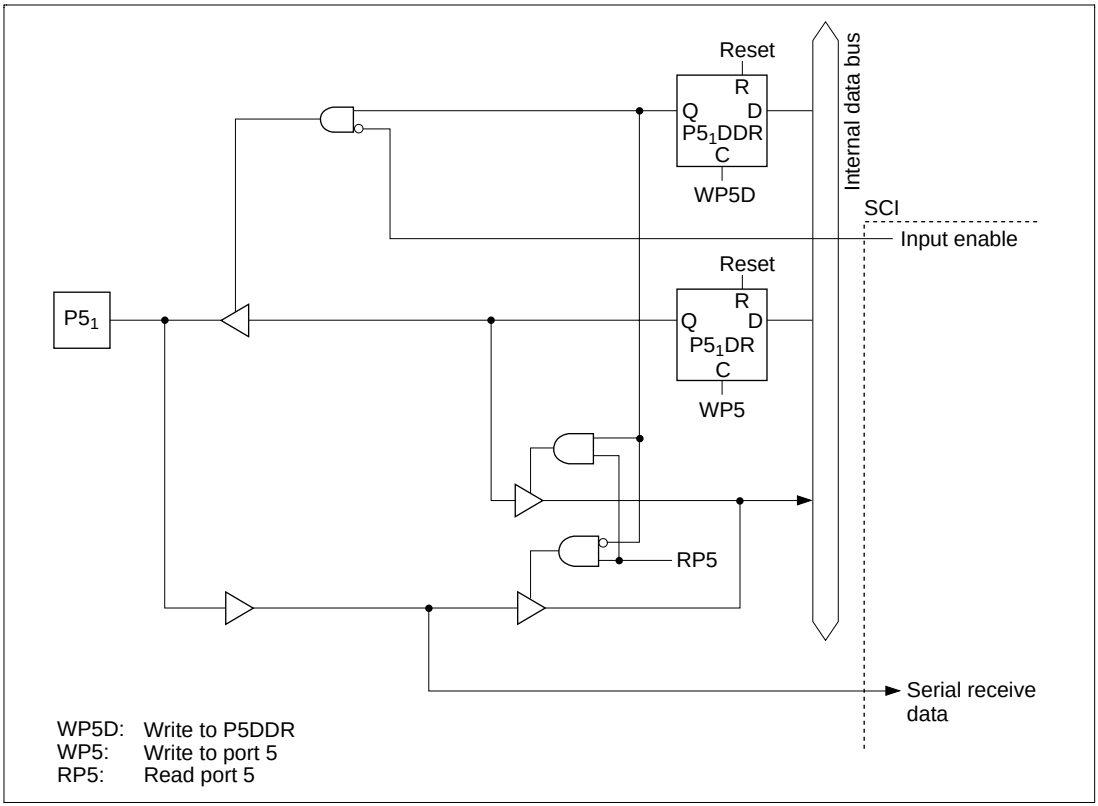


Figure C.5 (b) Port 5 Block Diagram (Pin $P5_1$)

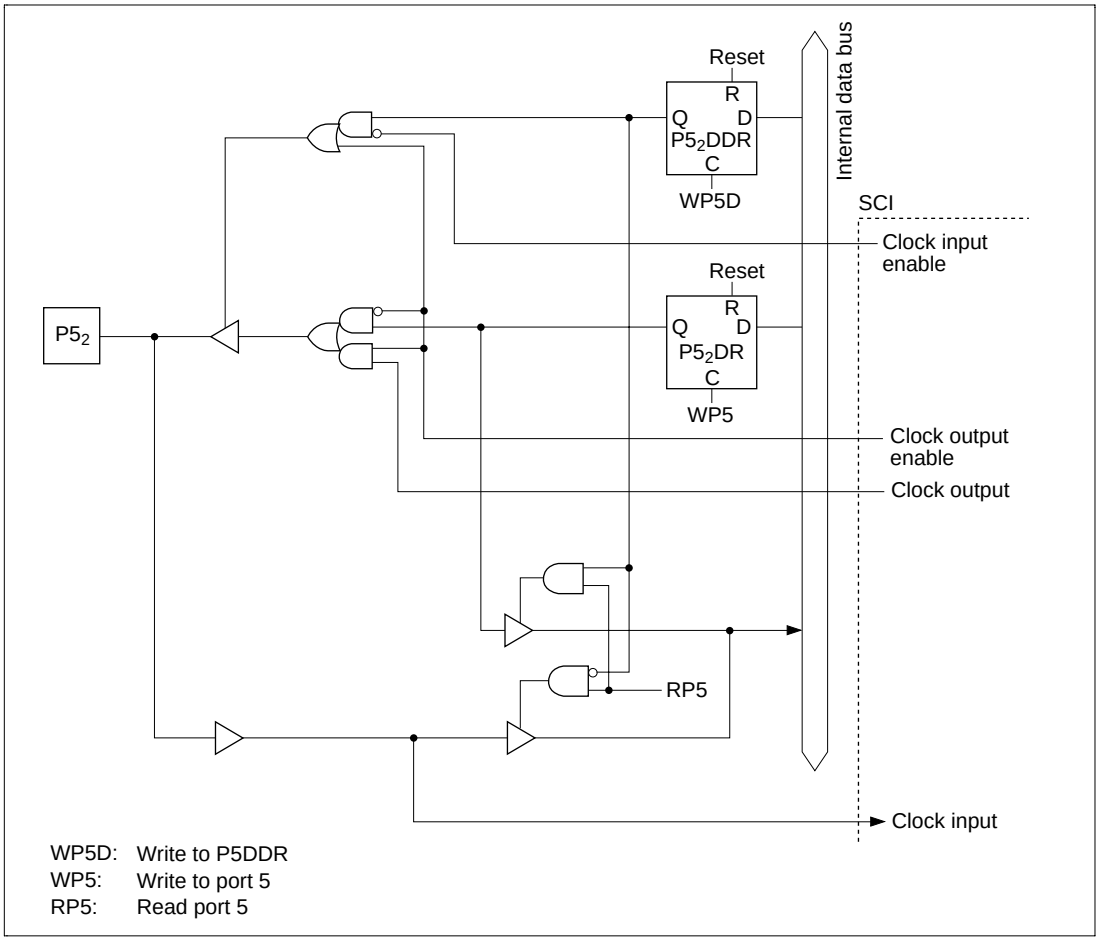


Figure C.5 (c) Port 5 Block Diagram (Pin $P5_2$)

C.6 Port 6 Block Diagrams

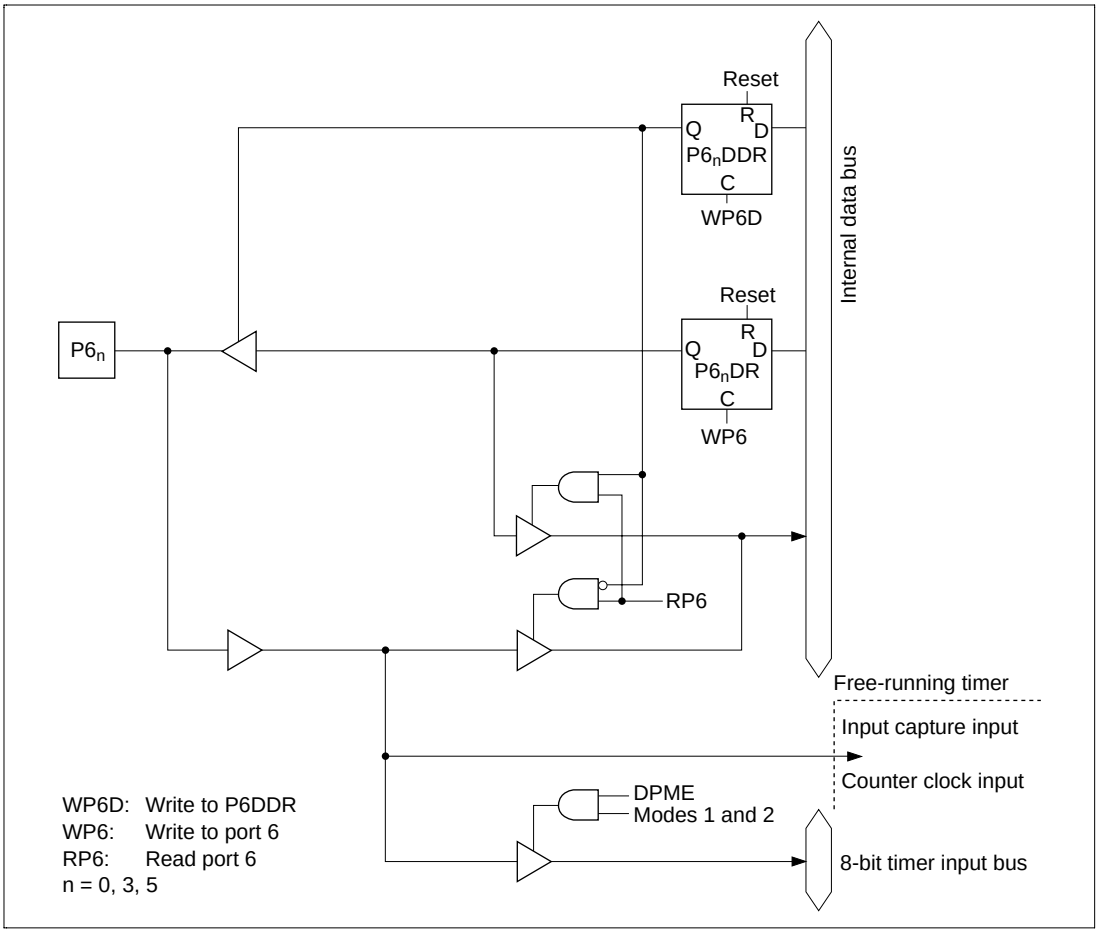


Figure C.6 (a) Port 6 Block Diagram (Pins $P6_0$, $P6_3$, $P6_5$)

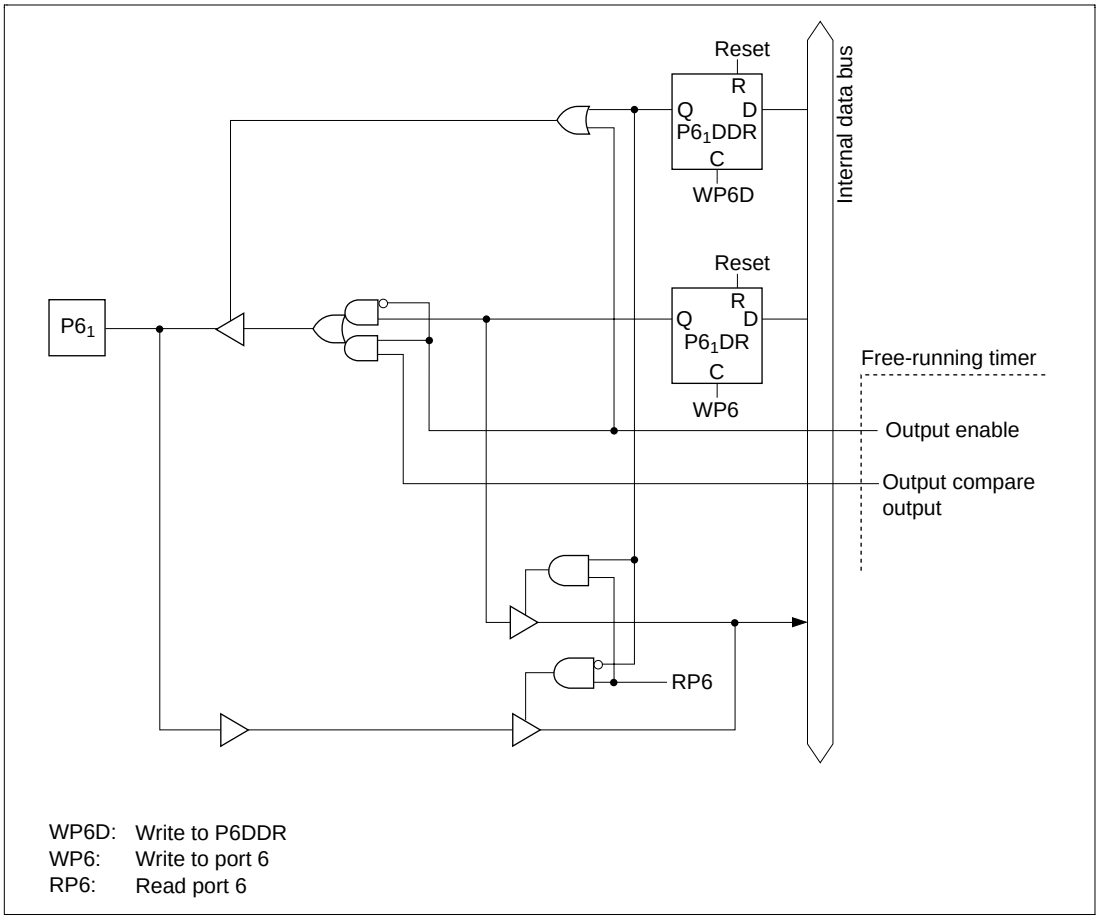


Figure C.6 (b) Port 6 Block Diagram (Pin P6$_1$)

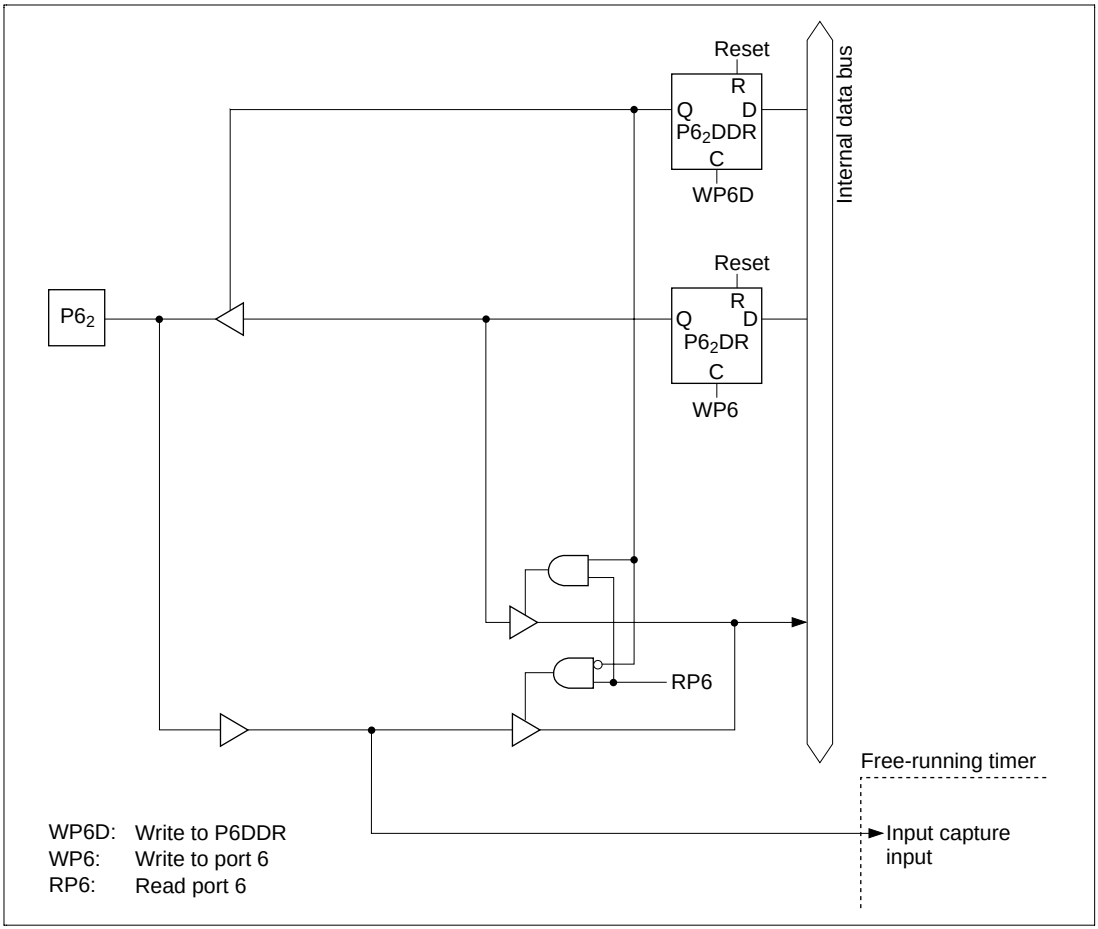


Figure C.6 (c) Port 6 Block Diagram (Pin P6$_2$)

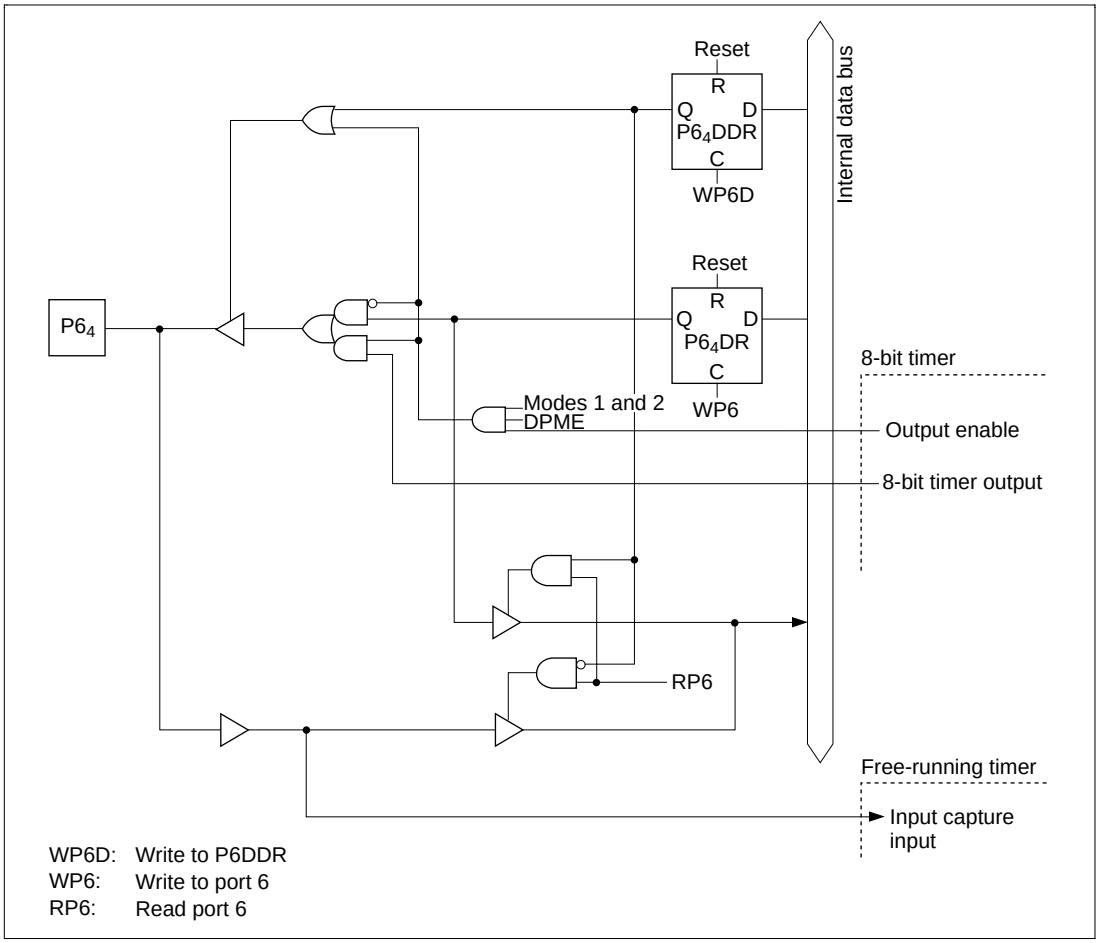


Figure C.6 (d) Port 6 Block Diagram (Pin $P6_4$)

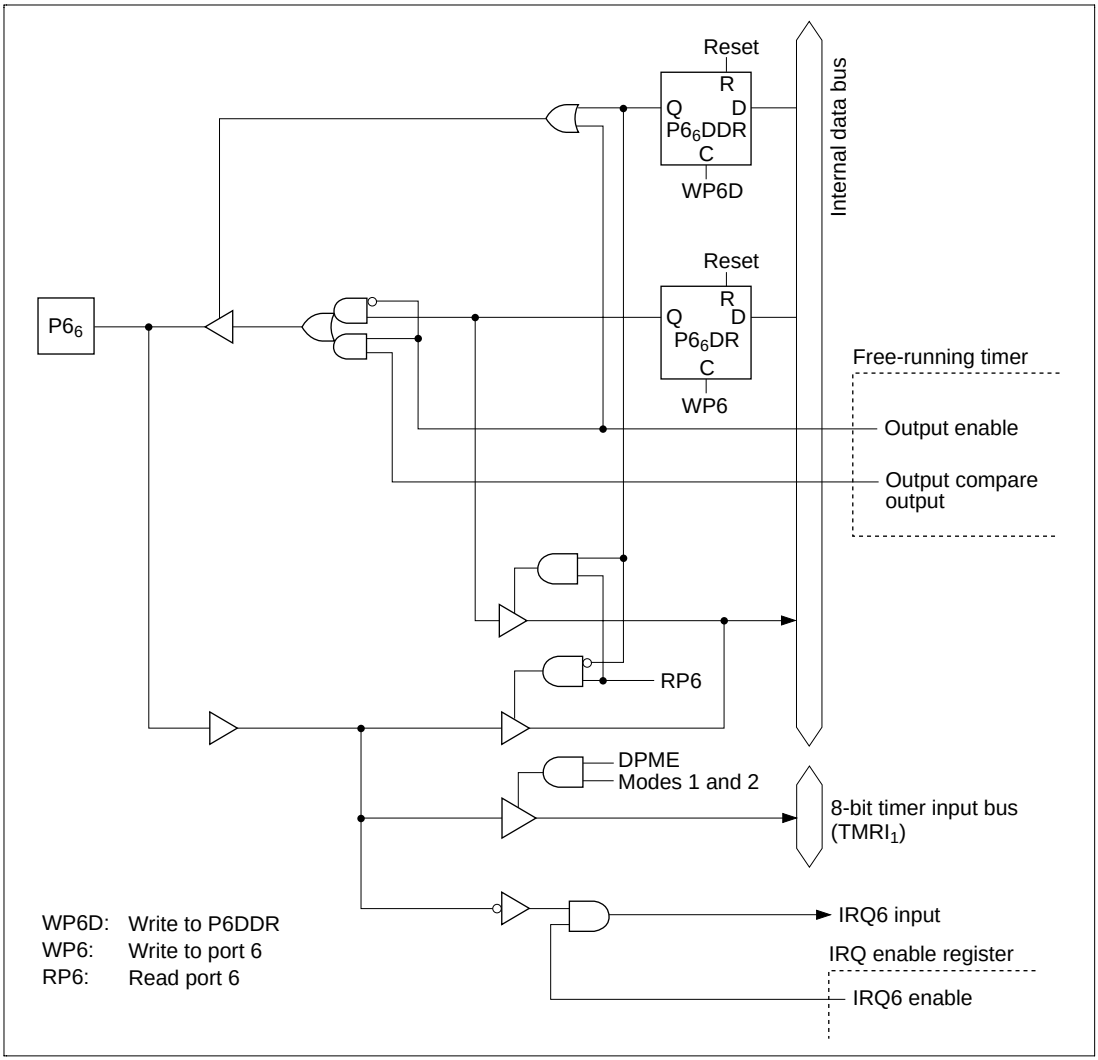


Figure C.6 (e) Port 6 Block Diagram (Pin $P6_6$)

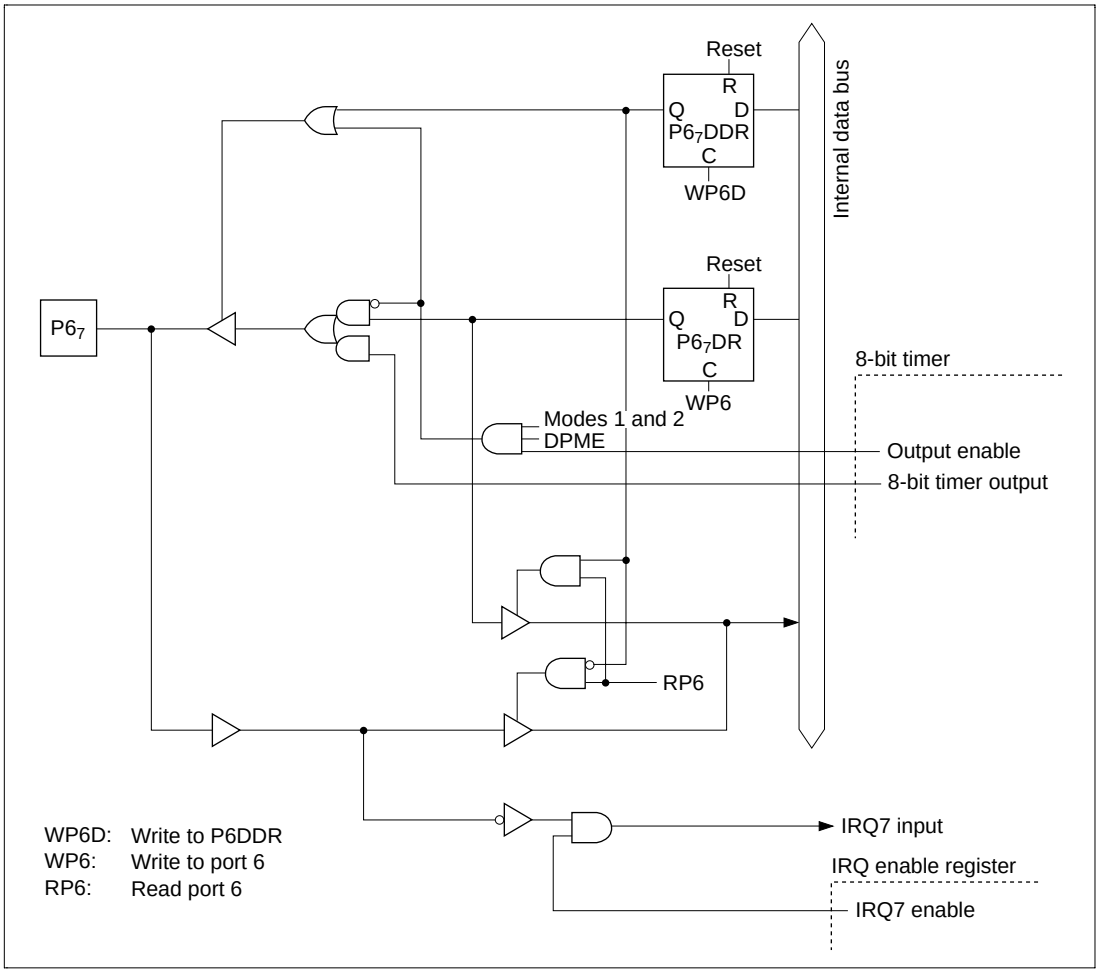


Figure C.6 (f) Port 6 Block Diagram (Pin $P6_7$)

C.7 Port 7 Block Diagram

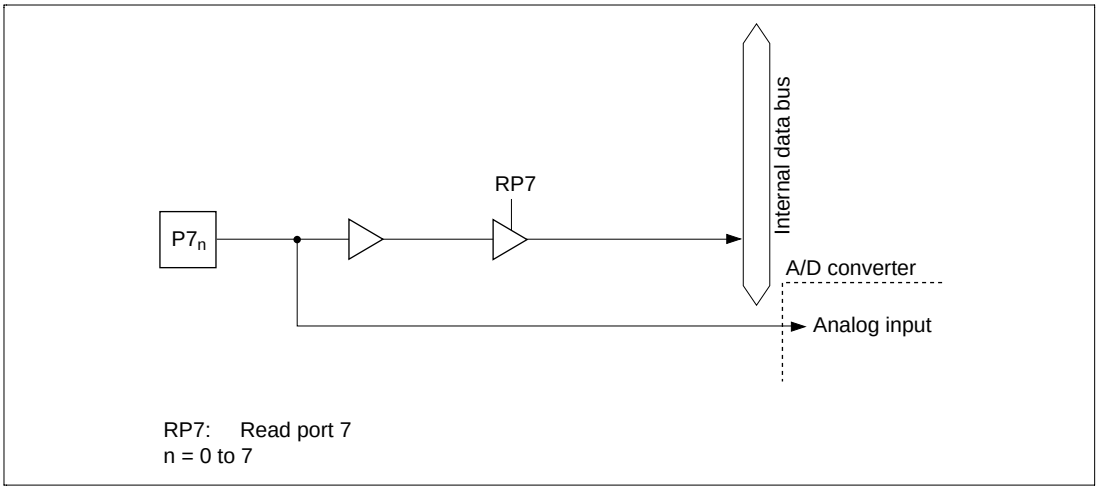


Figure C.7 Port 7 Block Diagram (Pins $P7_0$ to $P7_7$)

C.8 Port 8 Block Diagrams

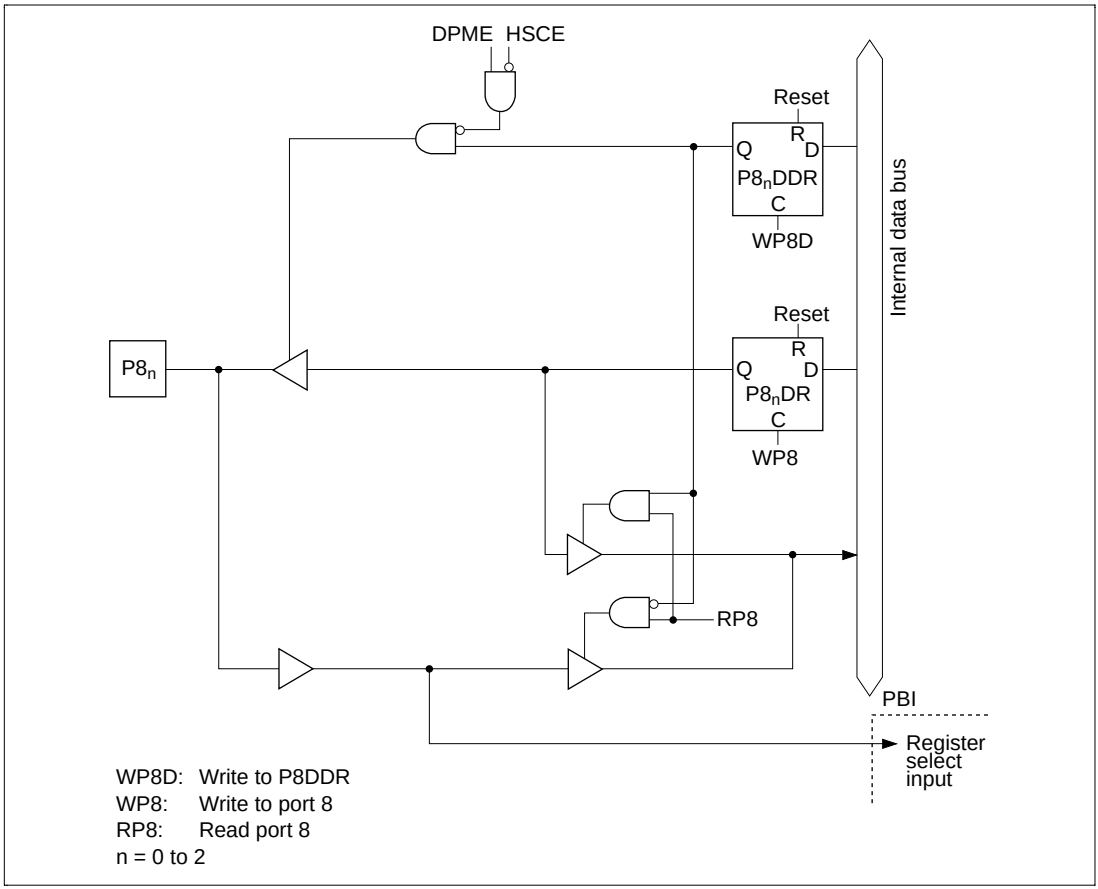


Figure C.8 (a) Port 8 Block Diagram (Pins $P8_0$ to $P8_2$)

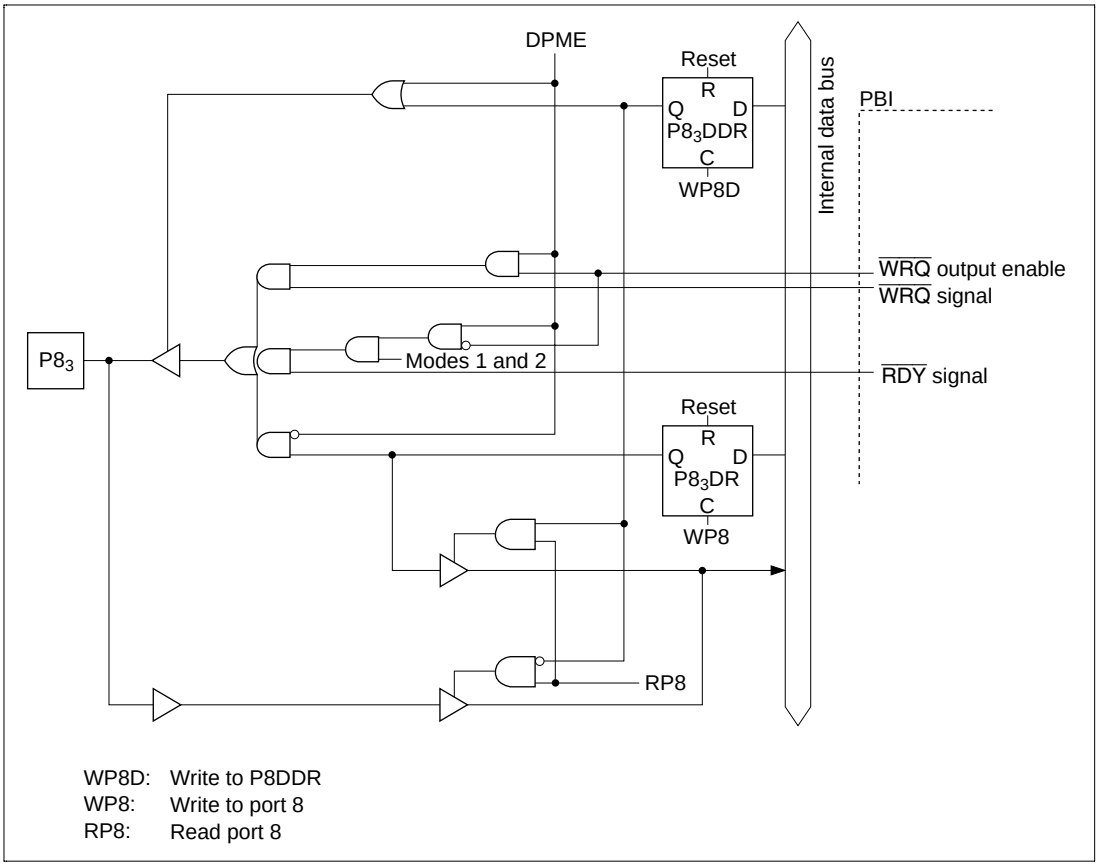


Figure C.8 (b) Port 8 Block Diagram (Pin P8$_3$)

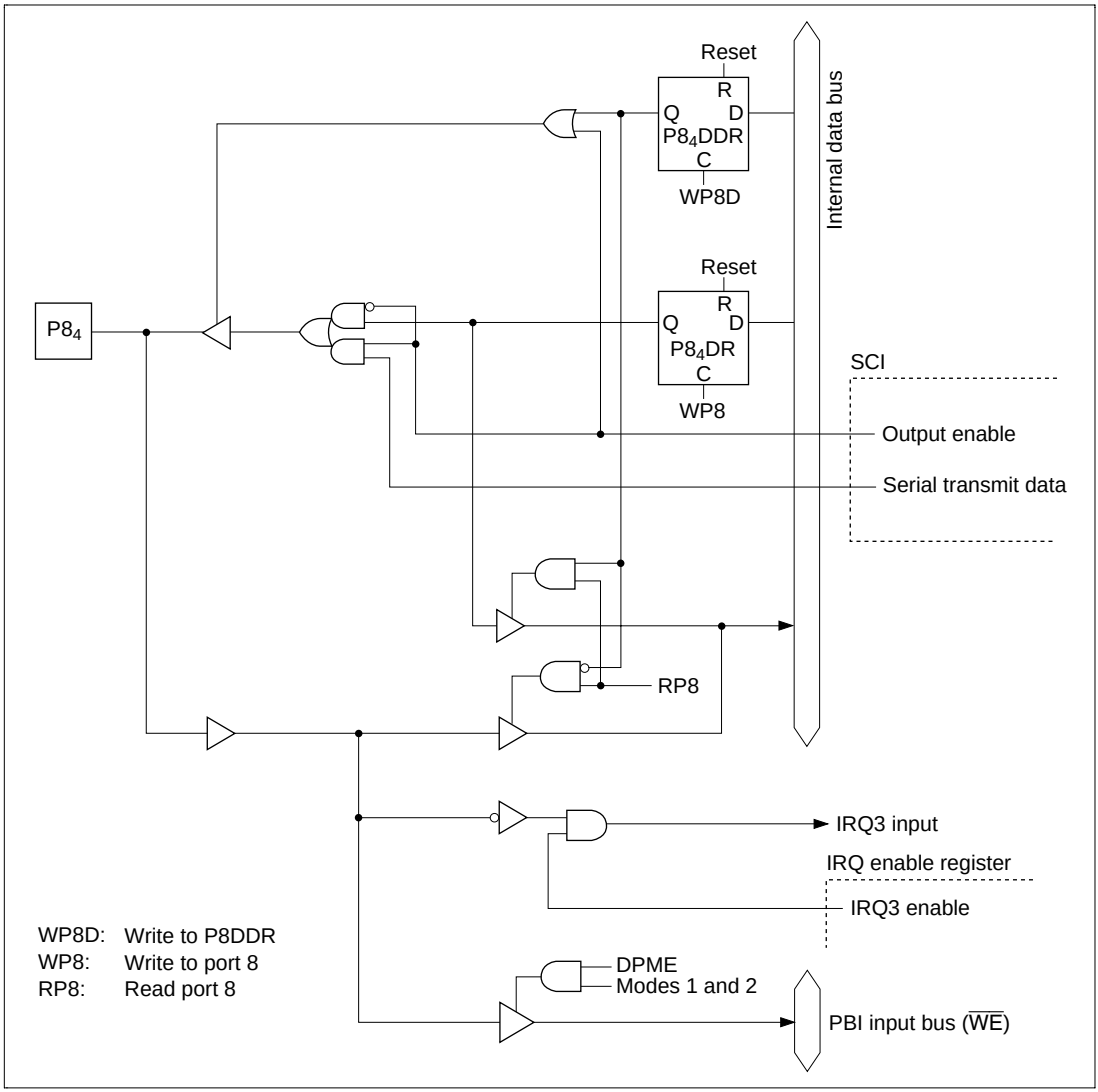


Figure C.8 (c) Port 8 Block Diagram (Pin $P8_4$)

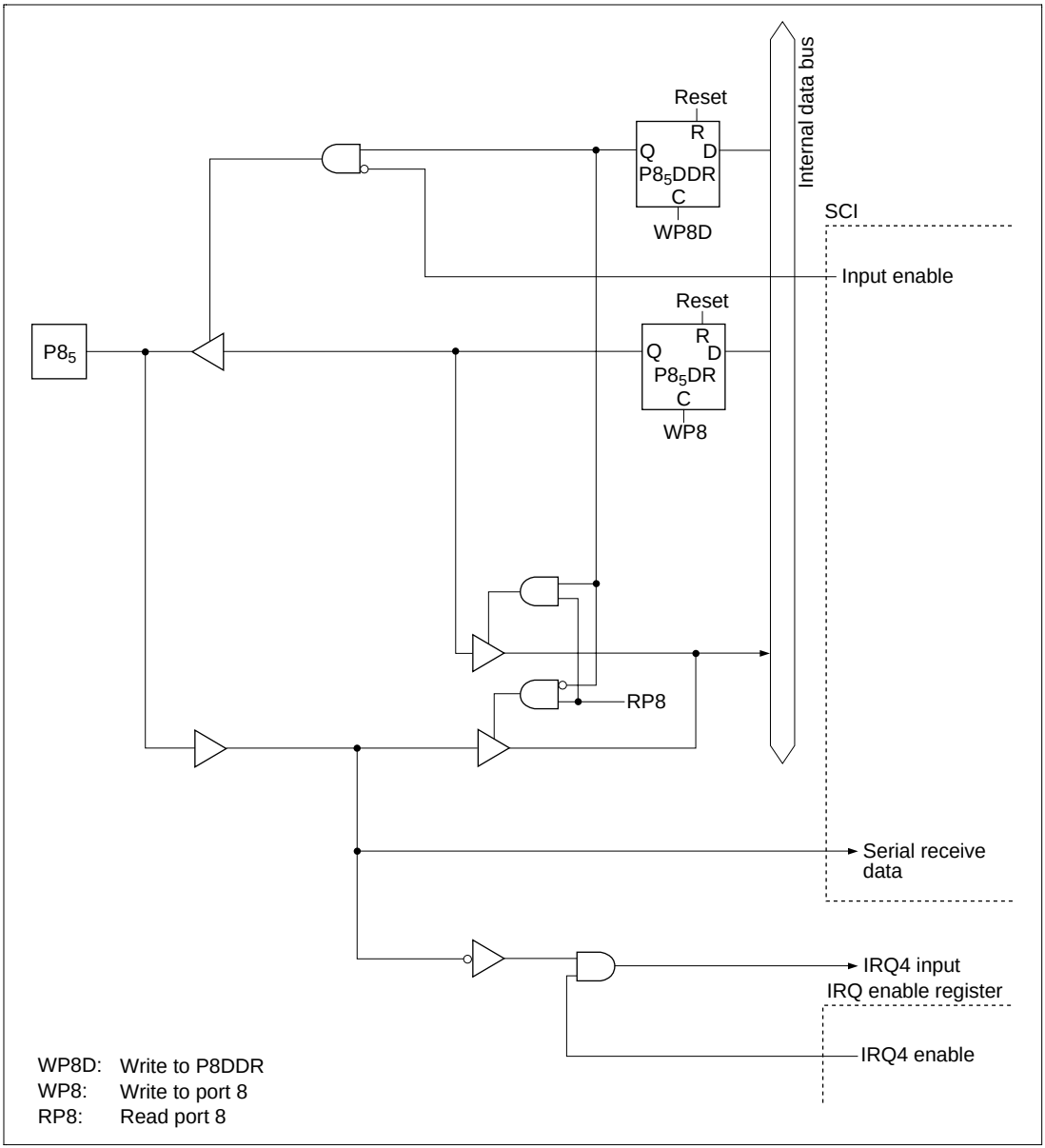


Figure C.8 (d) Port 8 Block Diagram (Pin $P8_5$)

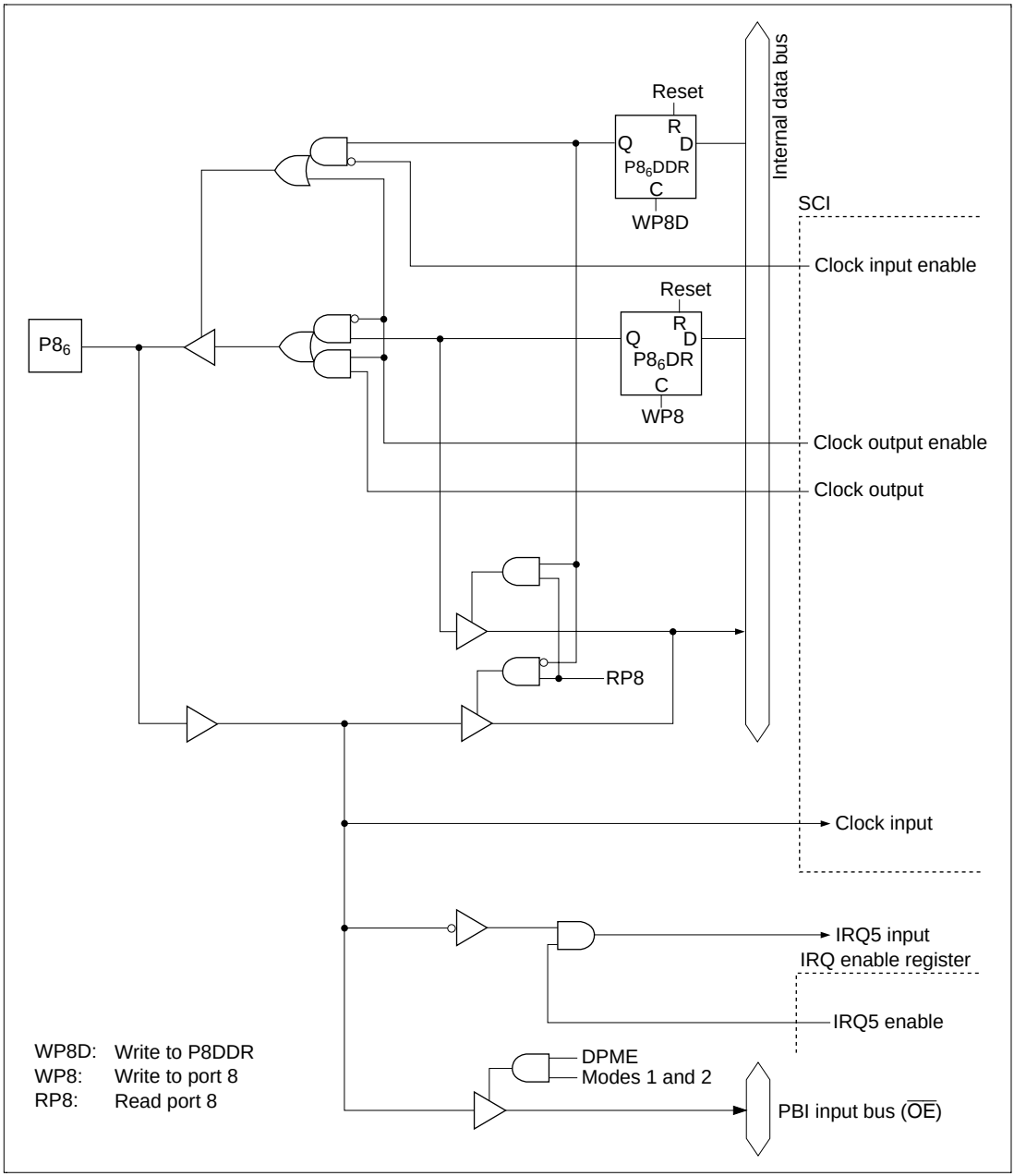


Figure C.8 (e) Port 8 Block Diagram (Pin $P8_6$)

C.9 Port 9 Block Diagrams

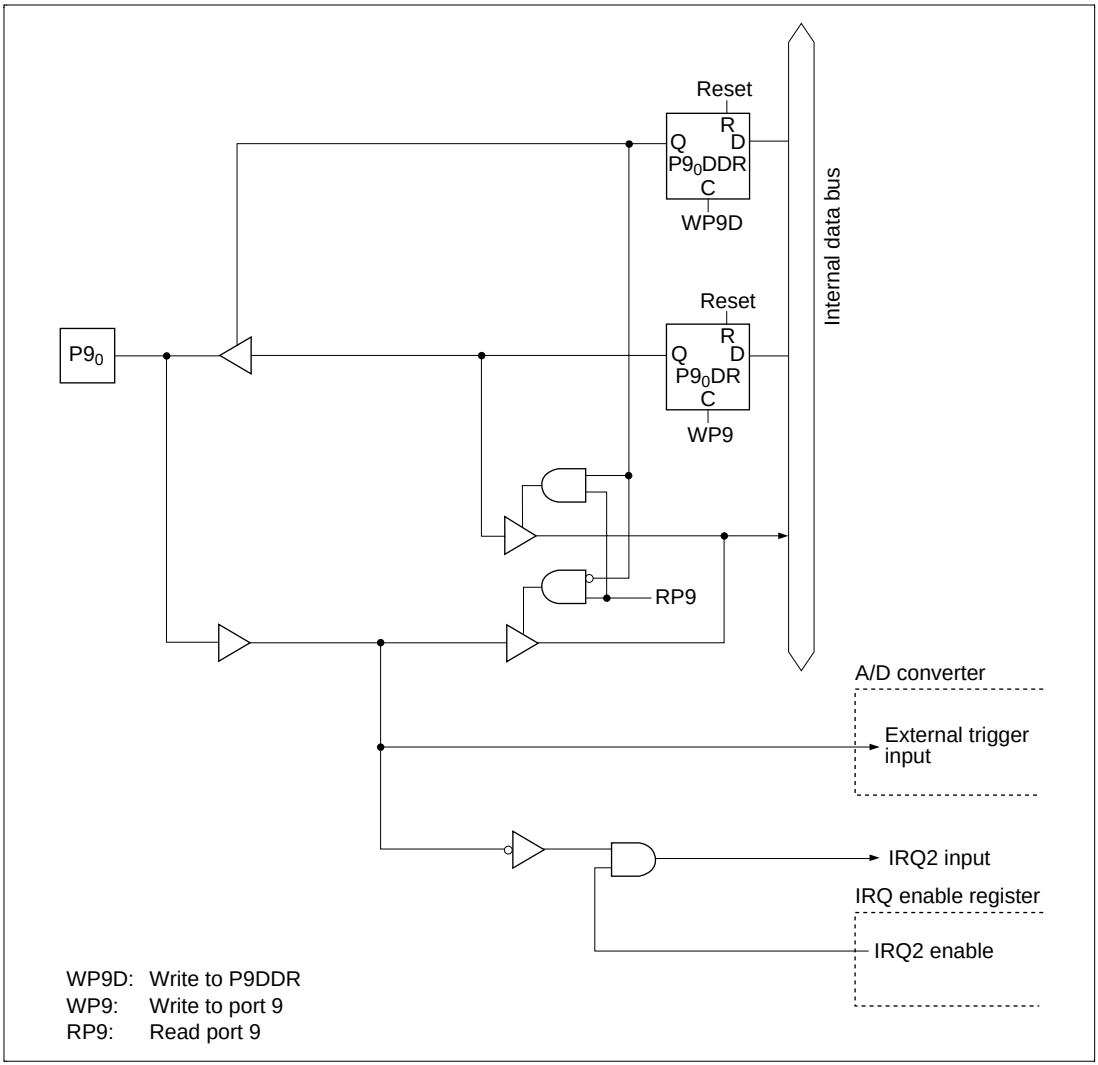


Figure C.9 (a) Port 9 Block Diagram (Pin P9$_0$)

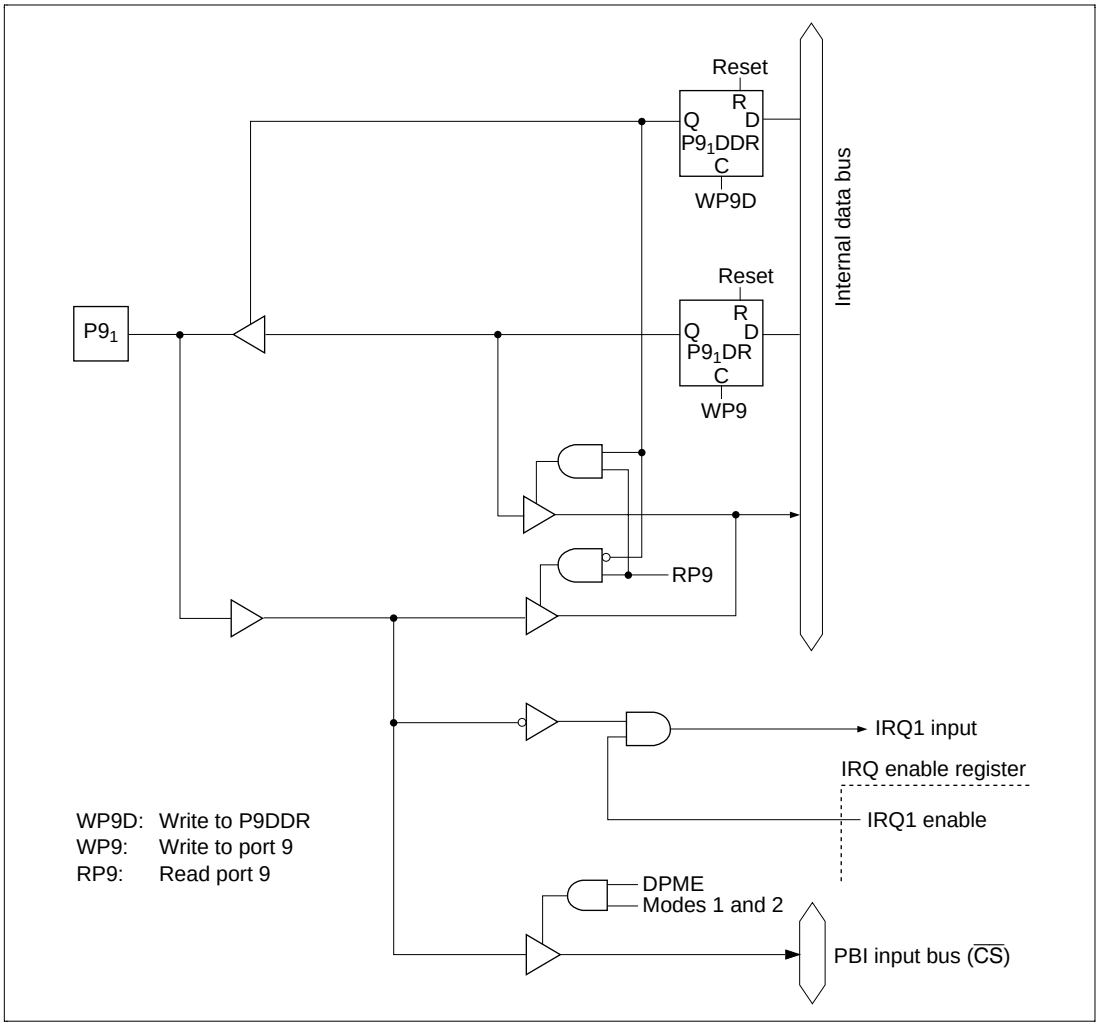


Figure C.9 (b) Port 9 Block Diagram (Pin P9$_1$)

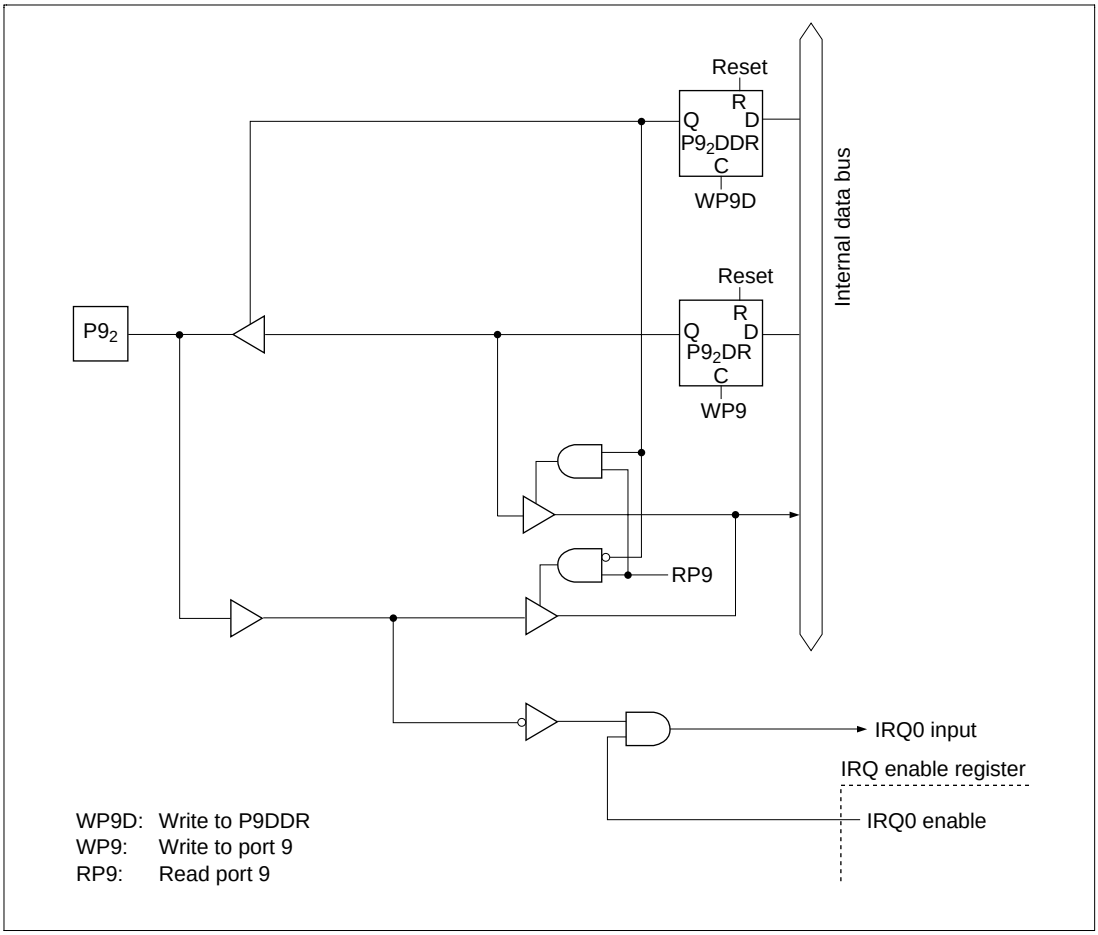


Figure C.9 (c) Port 9 Block Diagram (Pin P9$_2$)

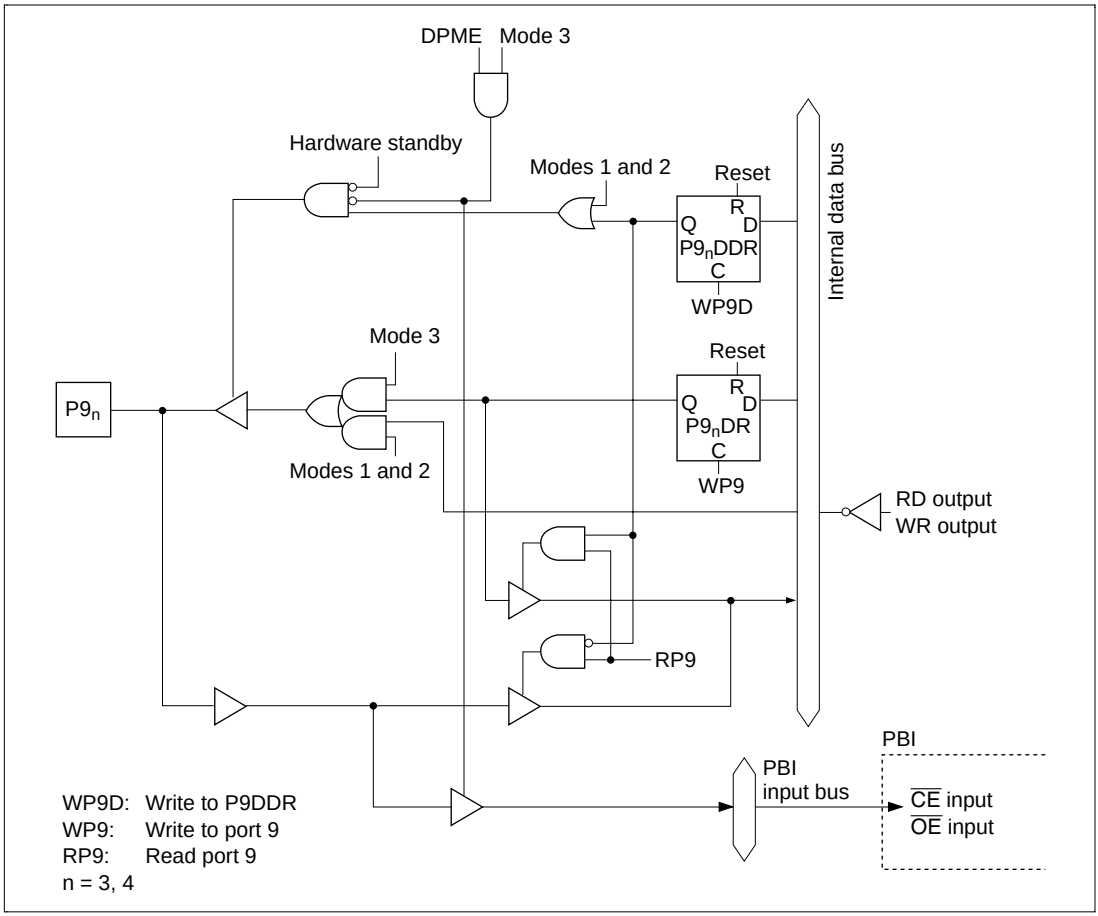
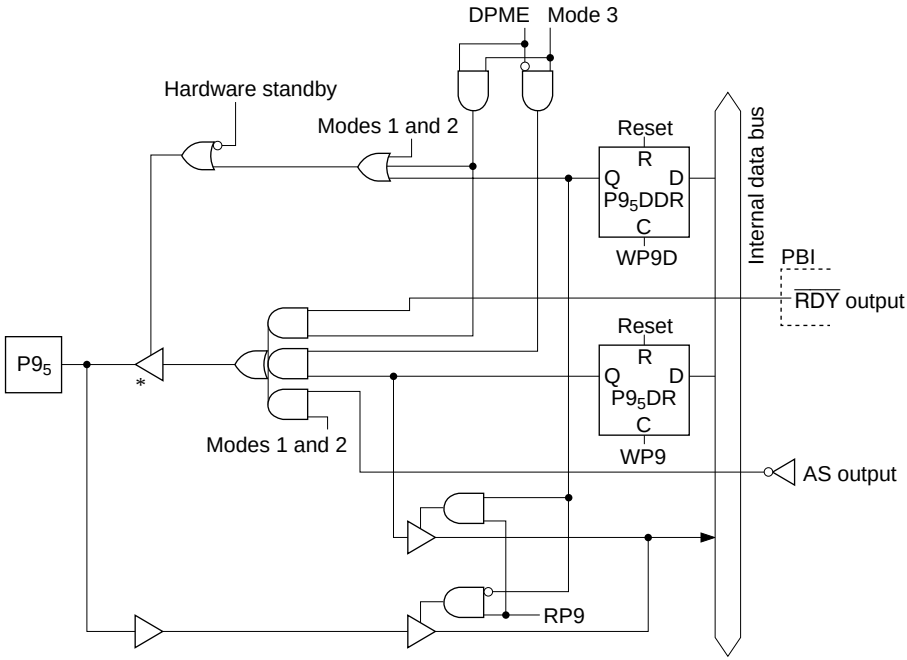


Figure C.9 (d) Port 9 Block Diagram (Pins $P9_3$, $P9_4$)



WP9D: Write to P9DDR
WP9: Write to port 9
RP9: Read port 9

Note: * NMOS open drain when DPME = 1 in mode 3.

Figure C.9 (e) Port 9 Block Diagram (Pin P9$_5$)

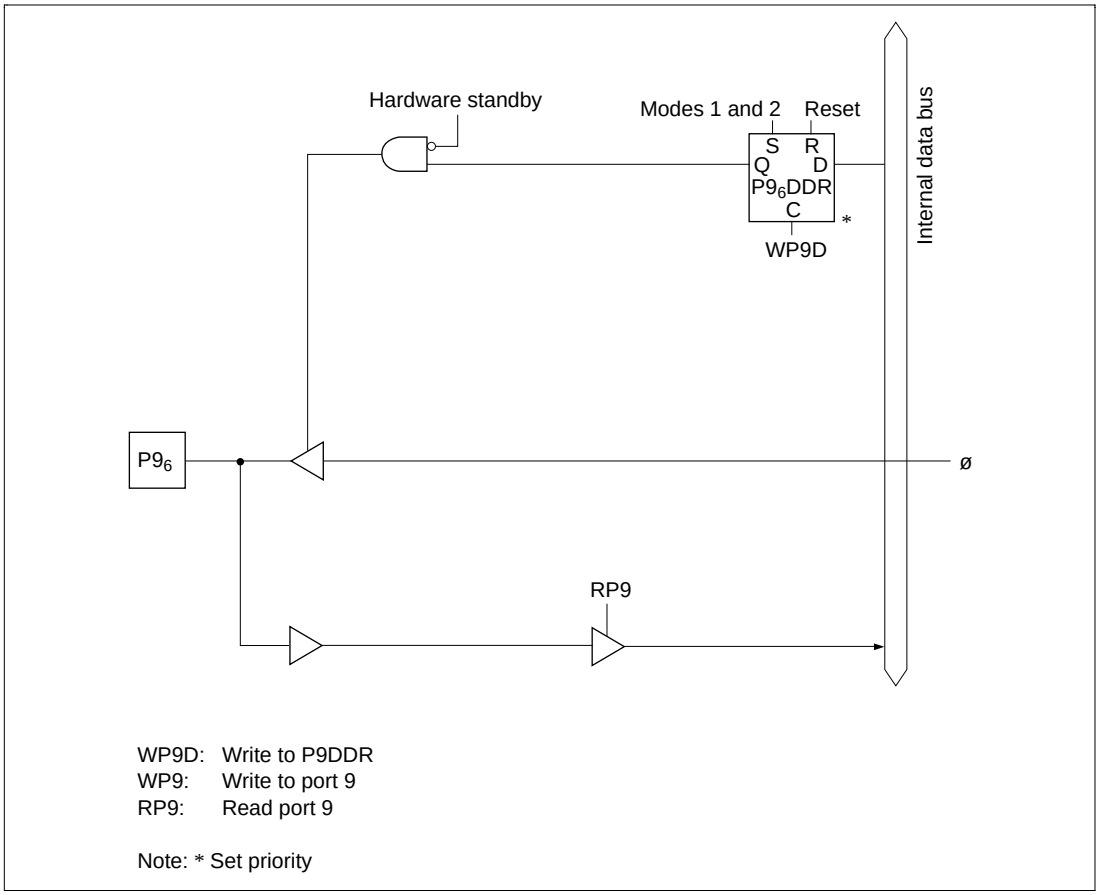


Figure C.9 (f) Port 9 Block Diagram (Pin $P9_6$)

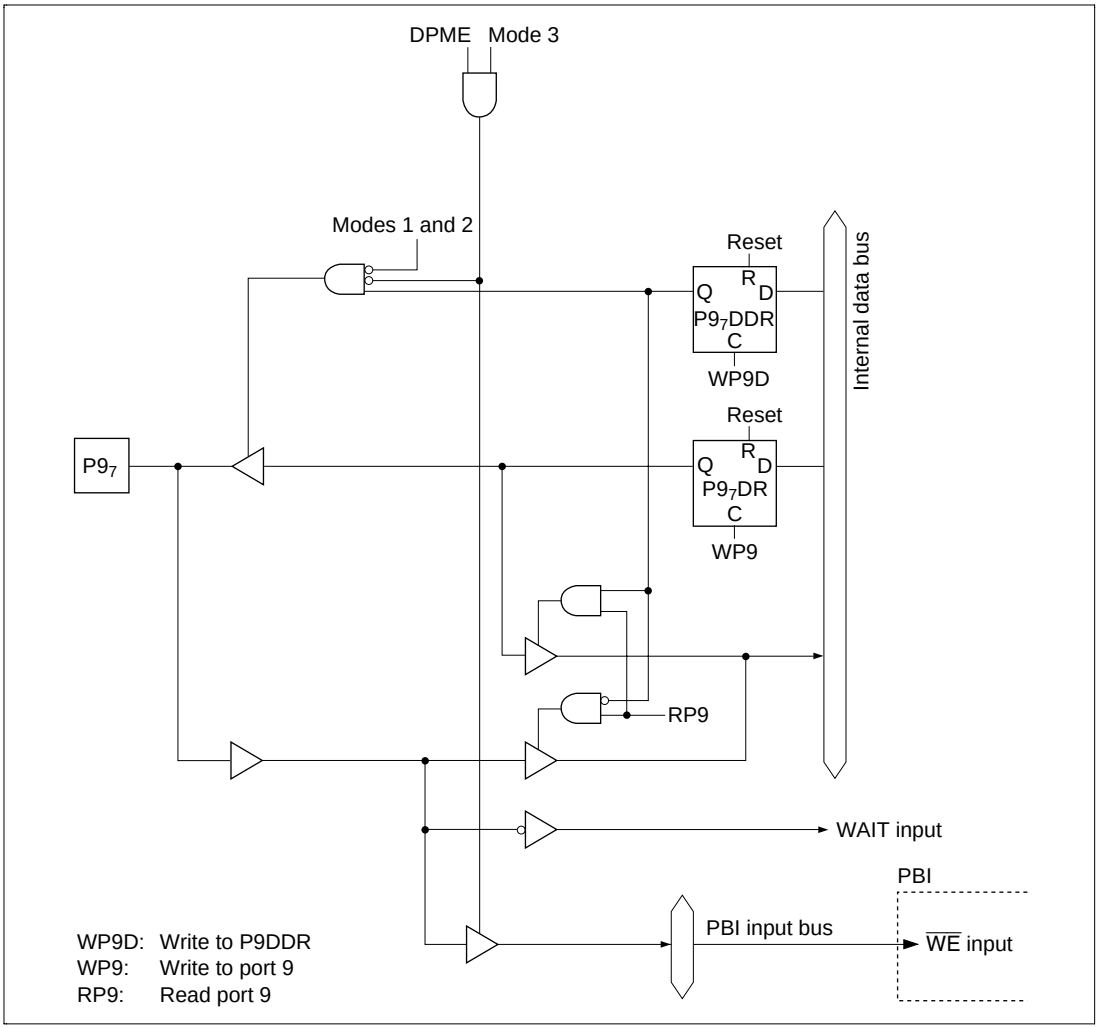


Figure C.9 (g) Port 9 Block Diagram (Pin $P9_7$)

Appendix D Pin States

D.1 Port States in Each Mode

Table D.1 Port States

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Sleep Mode	Program Execution Mode
$P1_7$ to $P1_0$	1	L	T	L	keep*[1]	A_7 to A_0
A_7 to A_0	2	T		(DDR = 1) L/keep		Address output or input port
TP_7 to TP_0				(DDR = 0) keep		
	3			keep		I/O port
$P2_7$ to $P2_0$	1	L	T	L	keep*[1]	A_{15} to A_8
A_{15} to A_8	2	T		(DDR = 1) L/keep		Address output or input port
TP_{15} to TP_8				(DDR = 0) keep		
	3			keep		I/O port
$P3_7$ to $P3_0$	1	T	T	T	T	D_7 to D_0
D_7 to D_0	2					
DDB_7 to DDB_0	3			keep	keep	I/O port
$P4_7$ to $P4_0$	1	T	T	keep*[2]	keep	I/O port
	2					
	3					
$P5_2$ to $P5_0$	1	T	T	keep*[2]	keep	I/O port
	2					
	3					
$P6_7$ to $P6_0$	1	T	T	keep*[2]	keep	I/O port
	2					
	3					
$P7_7$ to $P7_0$	1	T	T	T	T	Input port
	2					
	3					
$P8_6$ to $P8_0$	1	T	T	keep*[2]	keep	I/O port
	2					
	3					

Table D.1 Port States (cont)

Pin Name	Mode	Reset	Hardware Standby Mode	Software Standby Mode	Sleep Mode	Program Execution Mode
$P9_7$/$\overline{WAIT}$	1	T	T	T	T	$\overline{WAIT}$ or I/O port
	2					
	3			keep	keep	I/O port
$P9_6$/ø	1	Clock output	T	H	Clock output	Clock output
	2					
	3	T		(DDR = 1) H (DDR = 0) T	Clock output (DDR = 1) or T (DDR = 0)	Clock output (DDR = 1) or input port (DDR = 0)
$P9_5$ to $P9_3$ $\overline{AS}$, $\overline{WR}$, $\overline{RD}$	1	H	T	H	H	$\overline{AS}$, $\overline{WR}$, $\overline{RD}$
	2					
	3	T		keep	keep	I/O port
$P9_2$ to $P9_0$	1	T	T	keep	keep	I/O port
	2					
	3					

Legend

H: High

L: Low

T: High-impedance state

keep: Input pins are in the high-impedance state (with input pull-up on if PCR = 1) ; output pins maintain their previous state.

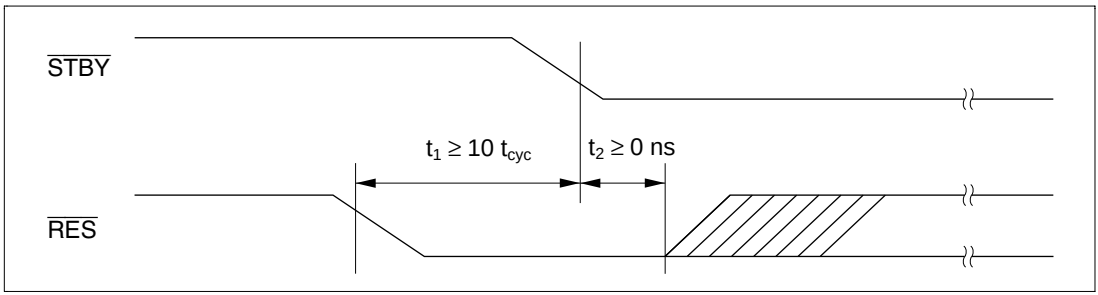
Notes: 1. Address output pins hold the last address accessed.

2. On-chip supporting modules are initialized, so these pins revert to I/O ports controlled by their DDR and DR bits.

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

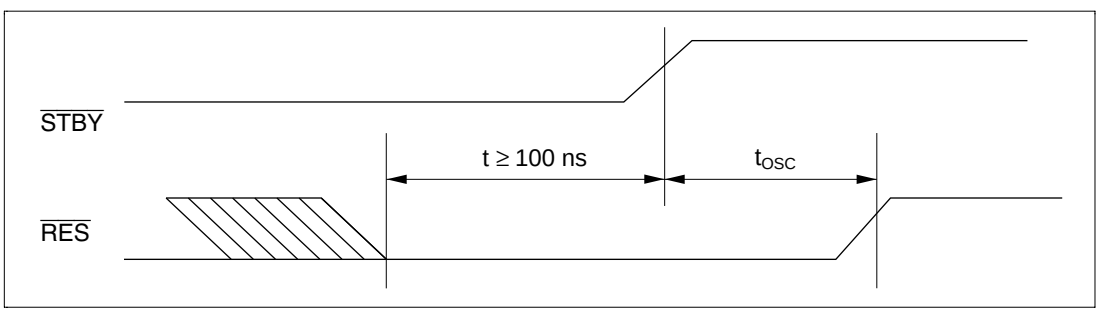
Timing of Transition to Hardware Standby Mode

(1) To retain RAM contents when the RAME bit in SYSCR is cleared to 0, drive the $\overline{\text{RES}}$ signal low 10 system clock cycles before the $\overline{\text{STBY}}$ signal goes low, as shown below. $\overline{\text{RES}}$ must remain low until $\overline{\text{STBY}}$ goes low (minimum delay from $\overline{\text{STBY}}$ low to $\overline{\text{RES}}$ high: 0 ns).



(2) When the RAME bit in SYSCR is set to 1 or when it is not necessary to retain RAM contents, $\overline{\text{RES}}$ does not have to be driven low as in (1).

Timing of Recovery From Hardware Standby Mode: Drive the $\overline{\text{RES}}$ signal low approximately 100 ns before $\overline{\text{STBY}}$ goes high.



Appendix F Product Code Lineup

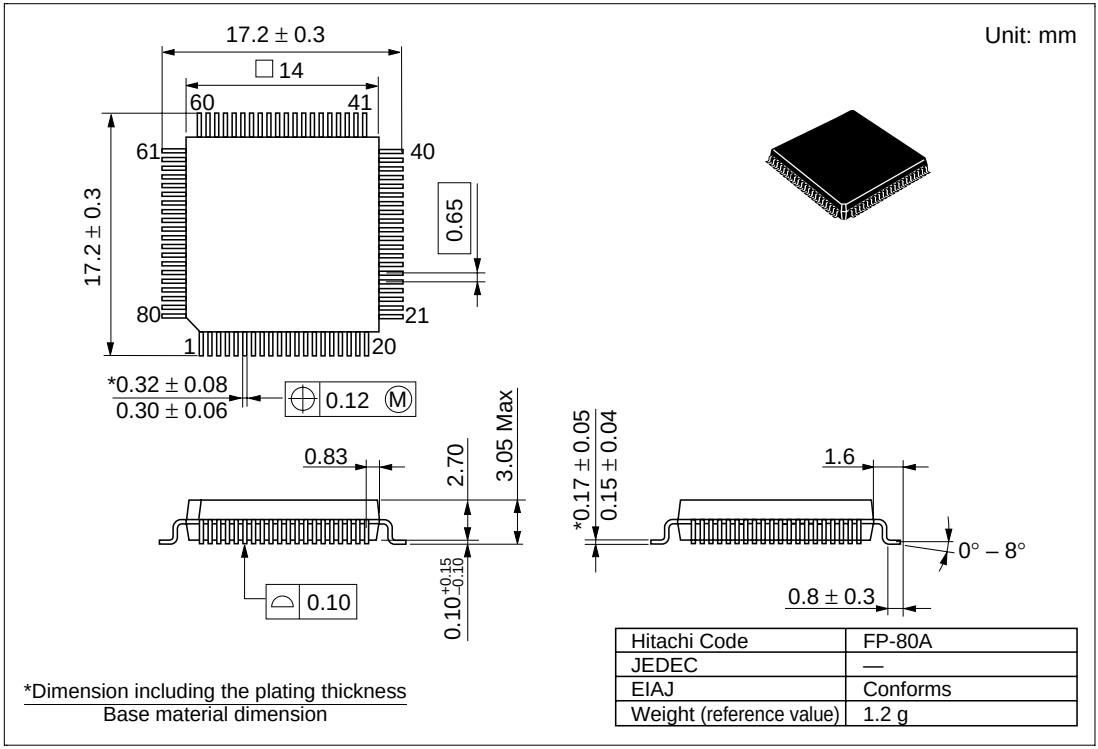
Table F.1 H8/3318 Product Code Lineup

Product Type		Product Code	Mark Code	Package (Hitachi Package Code)
H8/3318	ZTAT version	HD6473318CG16	HD6473318CG16	84-pin windowed LCC (CG-84)
		HD6473318CP16	HD6473318CP16	84-pin PLCC (CP-84)
		HD6473318F16	HD6473318F16	80-pin QFP (FP-80A)
		HD6473318TF16	HD6473318TF16	80-pin TQFP (TFP-80C)
	Mask ROM version	HD6433318CP	HD6433318(***)CP	84-pin PLCC (CP-84)
		HD6433318F	HD6433318(***)F	80-pin QFP (FP-80A)
		HD6433318TF	HD6433318(***)TF	80-pin TQFP (TFP-80C)

Note: (***) in mask ROM version is the ROM code.

Appendix G Package Dimensions

Figure G.1 shows the dimensions of the FP-80A package. Figure G.2 shows the dimensions of the TFP-80C package. Figure G.3 shows the dimensions of the CP-84 package. Figure G.4 shows the dimensions of the CE-84 package



Hitachi Code	FP-80A
JEDEC	—
EIAJ	Conforms
Weight (reference value)	1.2 g

Figure G.1 Package Dimensions (FP-80A)

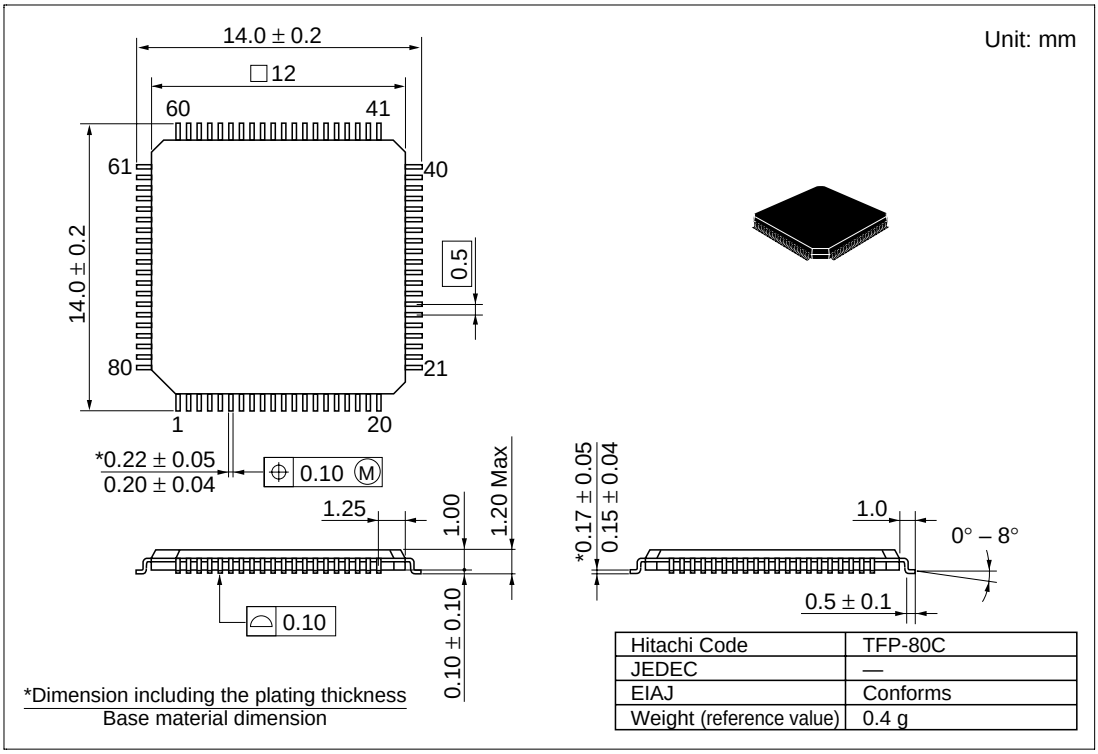
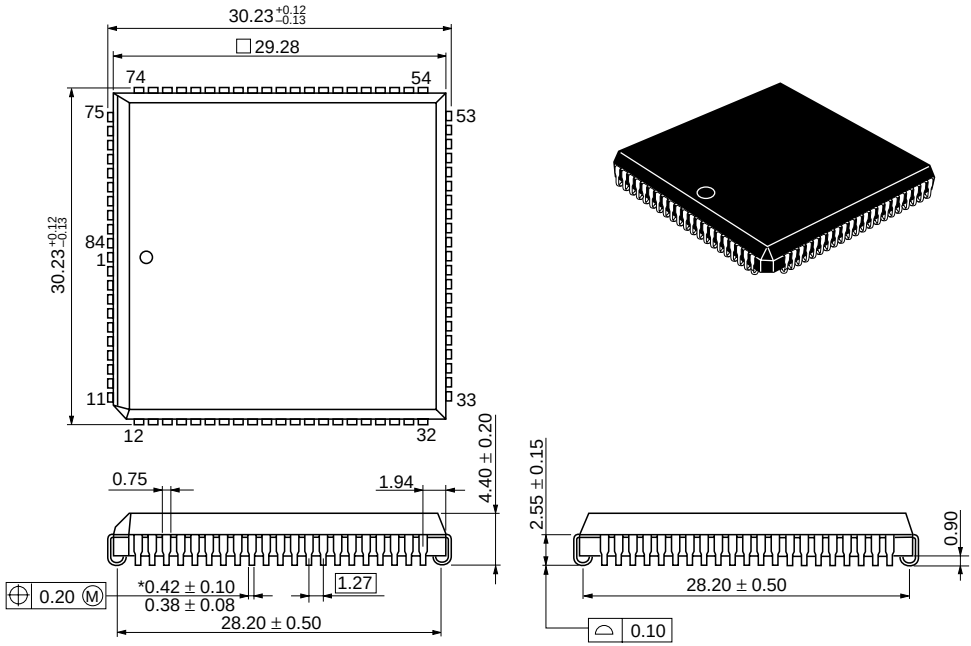


Figure G.2 Package Dimensions (TFP-80C)

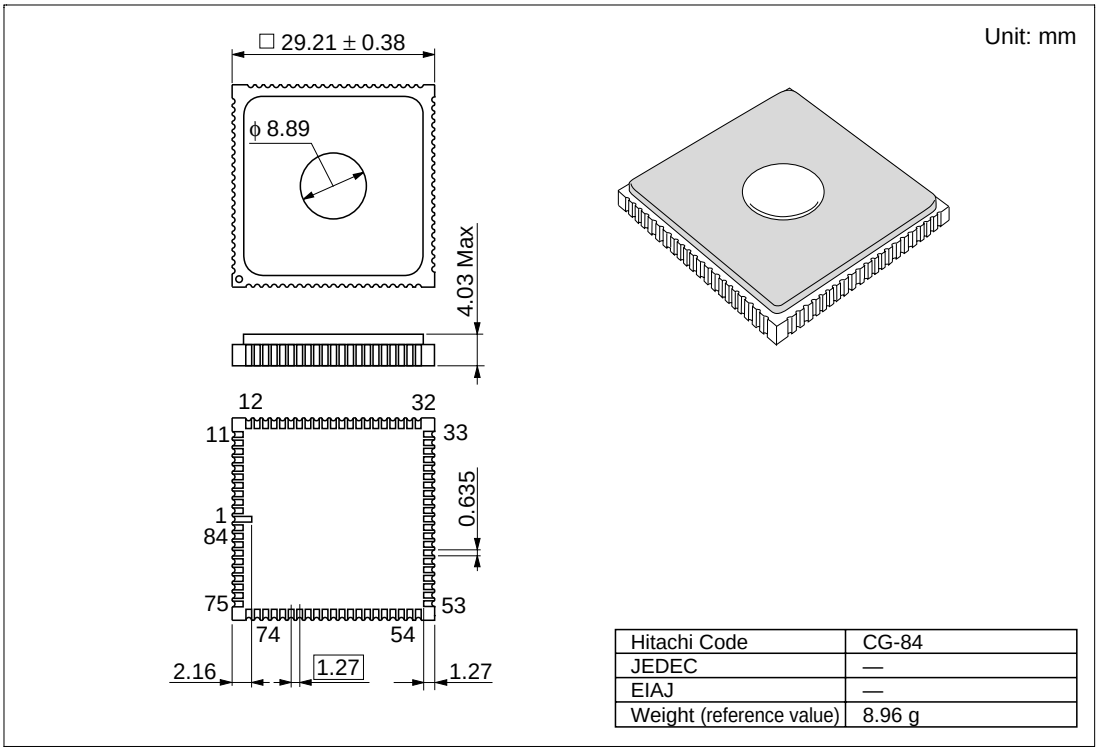
Unit: mm



Hitachi Code	CP-84
JEDEC	Conforms
EIAJ	Conforms
Weight (reference value)	6.4 g

*Dimension including the plating thickness
Base material dimension

Figure G.3 Package Dimensions (CP-84)



Hitachi Code	CG-84
JEDEC	—
EIAJ	—
Weight (reference value)	8.96 g

Figure G.4 Package Dimensions (CG-84)

H8/3318 Hardware Manual

Publication Date: 1st Edition, April 1996
2nd Edition, September 1999
Published by: Electronic Devices Sales & Marketing Group
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
UL Media Co., Ltd.