To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

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.

User's Manual

H8/3657 Series
HD6473657, HD6433657

Hardware Manual

H8/3656
HD6433656
H8/3655
HD6433655
H8/3654
HD6433654
H8/3653
HD6433653
H8/3652
HD6433652

Renesas Electronics
www.renesas.com

Preface

The H8/300L Series of single-chip microcomputers has the high-speed H8/300L CPU at its core, with many necessary peripheral functions on-chip. The H8/300L CPU instruction set is compatible with the H8/300 CPU.

The H8/3657 Series has a system-on-a-chip architecture that includes such peripheral functions as a five timers, a 14-bit PWM, a two-channel serial communication interface, and an A/D converter. This makes it ideal for use in advanced control systems.

This manual describes the hardware of the H8/3657 Series. For details on the H8/3657 Series instruction set, refer to the H8/300L Series Programming Manual.

Contents

Section 1 Overview

1.1 Overview

The H8/300L Series is a series of single-chip microcomputers (MCU: microcomputer unit), built around the high-speed H8/300L CPU and equipped with peripheral system functions on-chip.

Within the H8/300L Series, the H8/3657 Series of microcomputers are equipped with a UART (Universal Asynchronous Receiver/Transmitter). Other on-chip peripheral functions include five timers, a 14-bit pulse width modulator (PWM), two serial communication interface channels, and an A/D converter. Together, these functions make the H8/3657 Series ideally suited for embedded applications in advanced control systems. The ZTAT™* versions of the H8/3657 come with user-programmable PROM. Table 1 summarizes the features of the H8/3657 Series..

Table 1 summarizes the features of the H8/3657 Series.

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

Table 1-1 Features

Item	Description
CPU	High-speed H8/300L CPU
	• General-register architecture
	General registers: Sixteen 8-bit registers (can be used as eight 16-bit registers)
	• Operating speed
	— Max. operating speed: 5 MHz
	— Add/subtract: 0.4 µs (operating at 5 MHz)
	— Multiply/divide: 2.8 µs (operating at 5 MHz)
	— Can run on 32.768 kHz subclock
	• Instruction set compatible with H8/300 CPU
	— Instruction length of 2 bytes or 4 bytes
	— Basic arithmetic operations between registers
	— MOV instruction for data transfer between memory and registers
	• Typical instructions
	— Multiply (8 bits × 8 bits)
	— Divide (16 bits ÷ 8 bits)
	— Bit accumulator
	— Register-indirect designation of bit position

Table 1-1 Features (cont)

Item	Description
Interrupts	33 interrupt sources • 12 external interrupt sources (IRQ_3 to IRQ_0, INT_7 to INT_0) • 21 internal interrupt sources
Clock pulse generators	Two on-chip clock pulse generators • System clock pulse generator: 1 to 10 MHz • Subclock pulse generator: 32.768 kHz
Power-down modes	Seven power-down modes • Sleep (high-speed) mode • Sleep (medium-speed) mode • Standby mode • Watch mode • Subsleep mode • Subactive mode • Active (medium-speed) mode
Memory	Large on-chip memory • H8/3657: 60-kbyte ROM, 2-kbyte RAM • H8/3656: 48-kbyte ROM, 2-kbyte RAM • H8/3655: 40-kbyte ROM, 2-kbyte RAM • H8/3654: 32-kbyte ROM, 1 kbyte RAM • H8/3653: 24-kbyte ROM, 1 kbyte RAM • H8/3652: 16-kbyte ROM, 1 kbyte RAM
I/O ports	67 pins • 59 I/O pins • 8 input pins
Timers	Five on-chip timers • Timer A: 8-bit timer Count-up timer with selection of eight internal clock signals divided from the system clock (ø)* and four clock signals divided from the watch clock ($ø_w$)* • Timer B1: 8-bit timer — Count-up timer with selection of seven internal clock signals or event input from external pin — Auto-reloading Note: * ø and $ø_W$ are defined in section 4, Clock Pulse Generators.

Table 1-1 Features (cont)

Item	Description
Timers	• Timer V: 8-bit timer — Count-up timer with selection of six internal clock signals or event input from external pin — Compare-match waveform output — Incrementing specifiable by external trigger input • Timer X: 16-bit timer — Count-up timer with selection of three internal clock signals or event input from external pin — Output compare (2 output pins) — Input capture (4 input pins) • Watchdog timer — Reset signal generated by 8-bit counter overflow
Serial communication interface	Two channels on chip • SCI1: synchronous serial interface Choice of 8-bit or 16-bit data transfer • SCI3: 8-bit synchronous/asynchronous serial interface Incorporates multiprocessor communication function
14-bit PWM	Pulse-division PWM output for reduced ripple • Can be used as a 14-bit D/A converter by connecting to an external low-pass filter.
A/D converter	Successive approximations using a resistance ladder • 8-channel analog input pins • Conversion time: 31/ø or 62/ø per channel

Table 1-1 Features (cont)

Item	Specification				
Product lineup	Product Code				
	Mask ROM Version	ZTAT™ Version	Package		ROM/RAM Size
	HD6433657W	HD6473657W	80-pin	TQFP (TFP-80C)	ROM 60 kbytes RAM 2 kbytes
	HD6433657X	HD6473657X	80-pin	TQFP (TFP-80F)	
	HD6433657H	HD6473657H	80-pin	QFP (FP-80A)	
	HD6433657F	HD6473657F	80-pin	QFP (FP-80B)	
	HD6433656W	—	80-pin	TQFP (TFP-80C)	ROM 48 kbytes RAM 2 kbytes
	HD6433656X	—	80-pin	TQFP (TFP-80F)	
	HD6433656H	—	80-pin	QFP (FP-80A)	
	HD6433656F	—	80-pin	QFP (FP-80B)	
	HD6433655W	—	80-pin	TQFP (TFP-80C)	ROM 40 kbytes RAM 2 kbytes
	HD6433655X	—	80-pin	TQFP (TFP-80F)	
	HD6433655H	—	80-pin	QFP (FP-80A)	
	HD6433655F	—	80-pin	QFP (FP-80B)	
	HD6433654W	—	80-pin	TQFP (TFP-80C)	ROM 32 kbytes RAM 1 kbyte
	HD6433654X	—	80-pin	TQFP (TFP-80F)	
	HD6433654H	—	80-pin	QFP (FP-80A)	
	HD6433654F	—	80-pin	QFP (FP-80B)	
	HD6433653W	—	80-pin	TQFP (TFP-80C)	ROM 24 kbytes RAM 1 kbyte
	HD6433653X	—	80-pin	TQFP (TFP-80F)	
	HD6433653H	—	80-pin	QFP (FP-80A)	
	HD6433653F	—	80-pin	QFP (FP-80B)	
	HD6433652W	—	80-pin	TQFP (TFP-80C)	ROM 16 kbytes RAM 1 kbyte
	HD6433652X	—	80-pin	TQFP (TFP-80F)	
	HD6433652H	—	80-pin	QFP (FP-80A)	
	HD6433652F	—	80-pin	QFP (FP-80B)	

1.2 Internal Block Diagram

Figure 1-1 shows a block diagram of the H8/3657 Series.

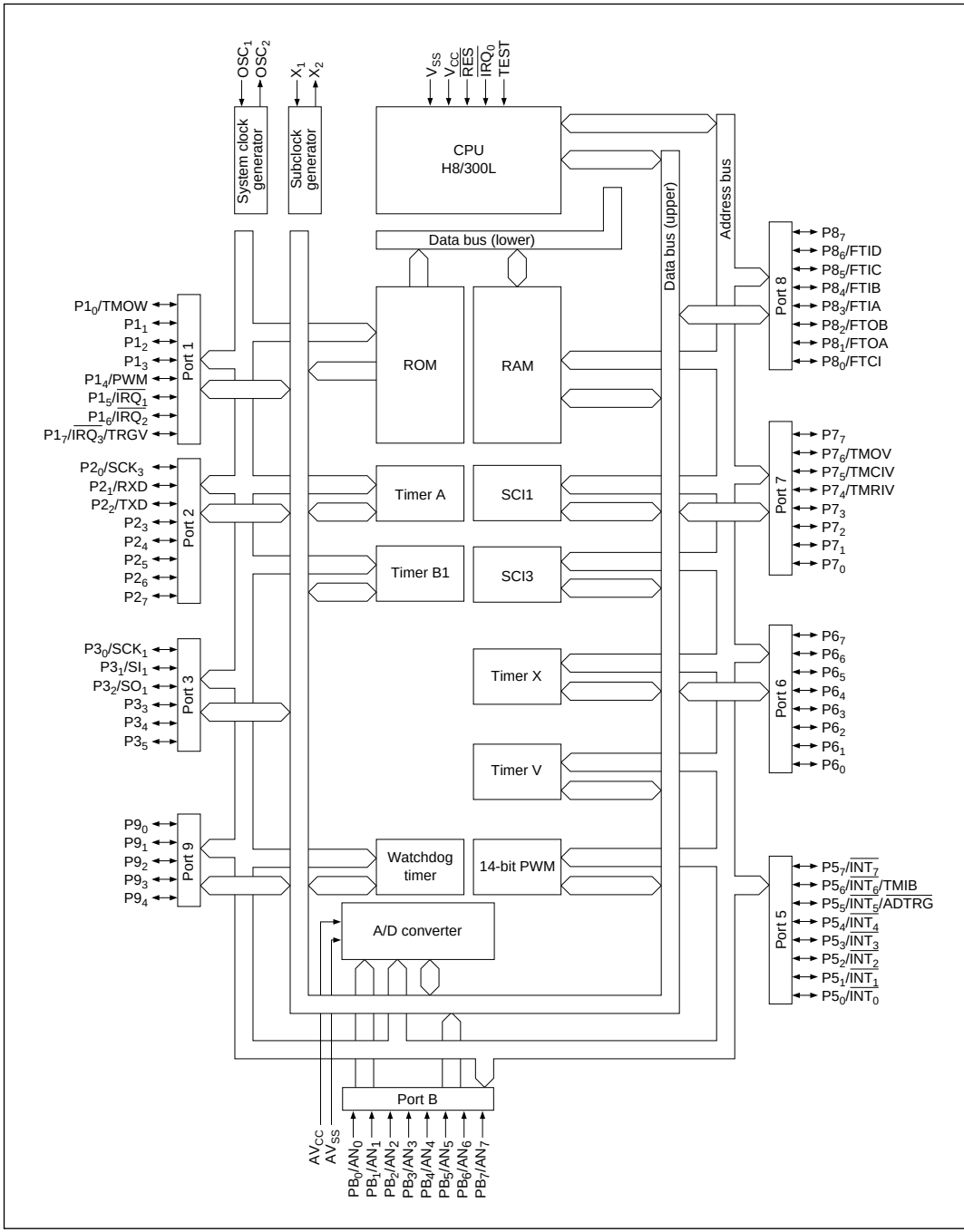


Figure 1-1 Block Diagram

1.3 Pin Arrangement and Functions

1.3.1 Pin Arrangement

The H8/3657 Series pin arrangement is shown in figures 1-2 (TFP-80C, TFP-80F, FP-80A), and in figures 1-3 (FP-80B).

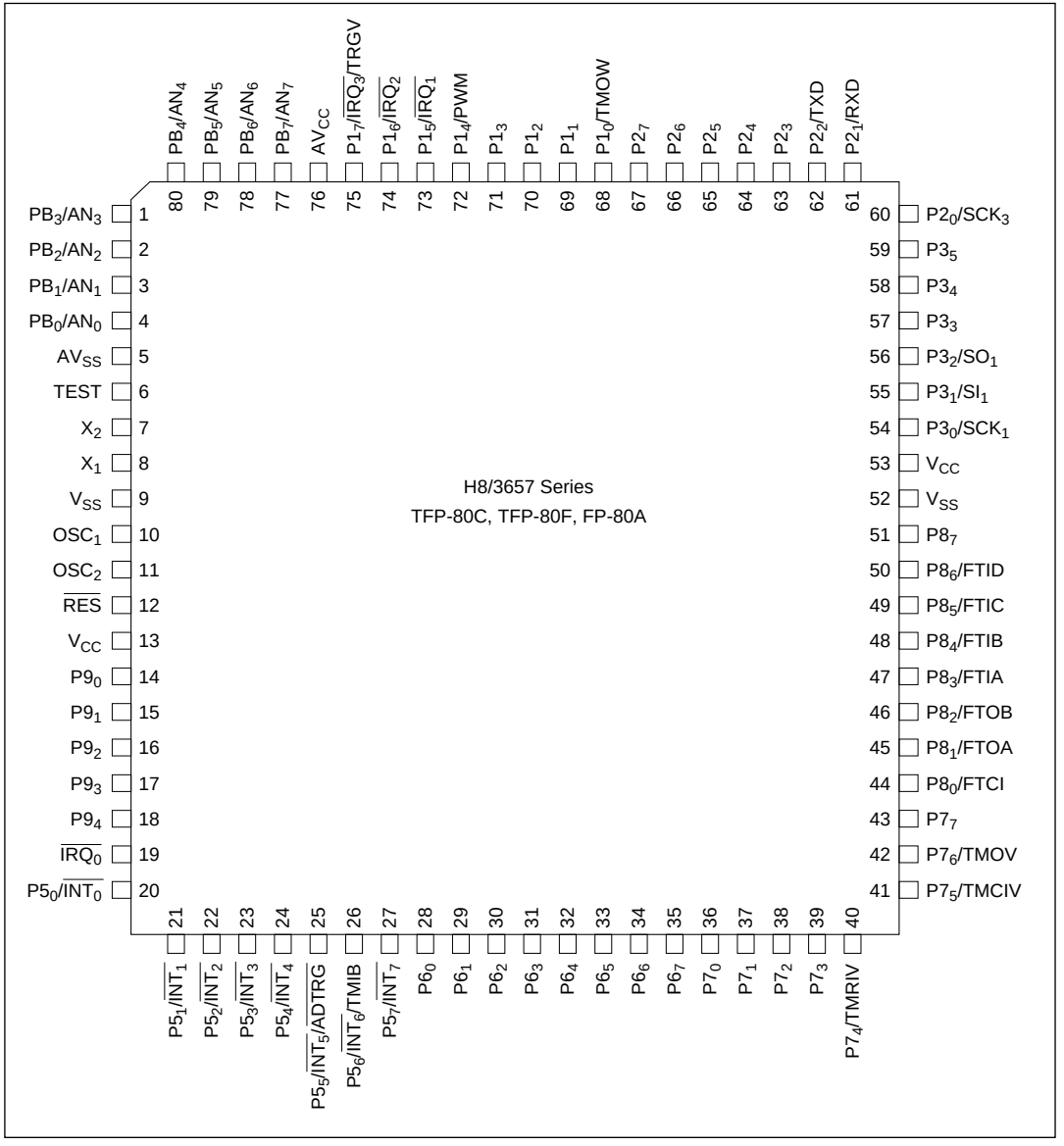


Figure 1-2 Pin Arrangement (TFP-80C, TFP-80F, FP-80A: Top View)

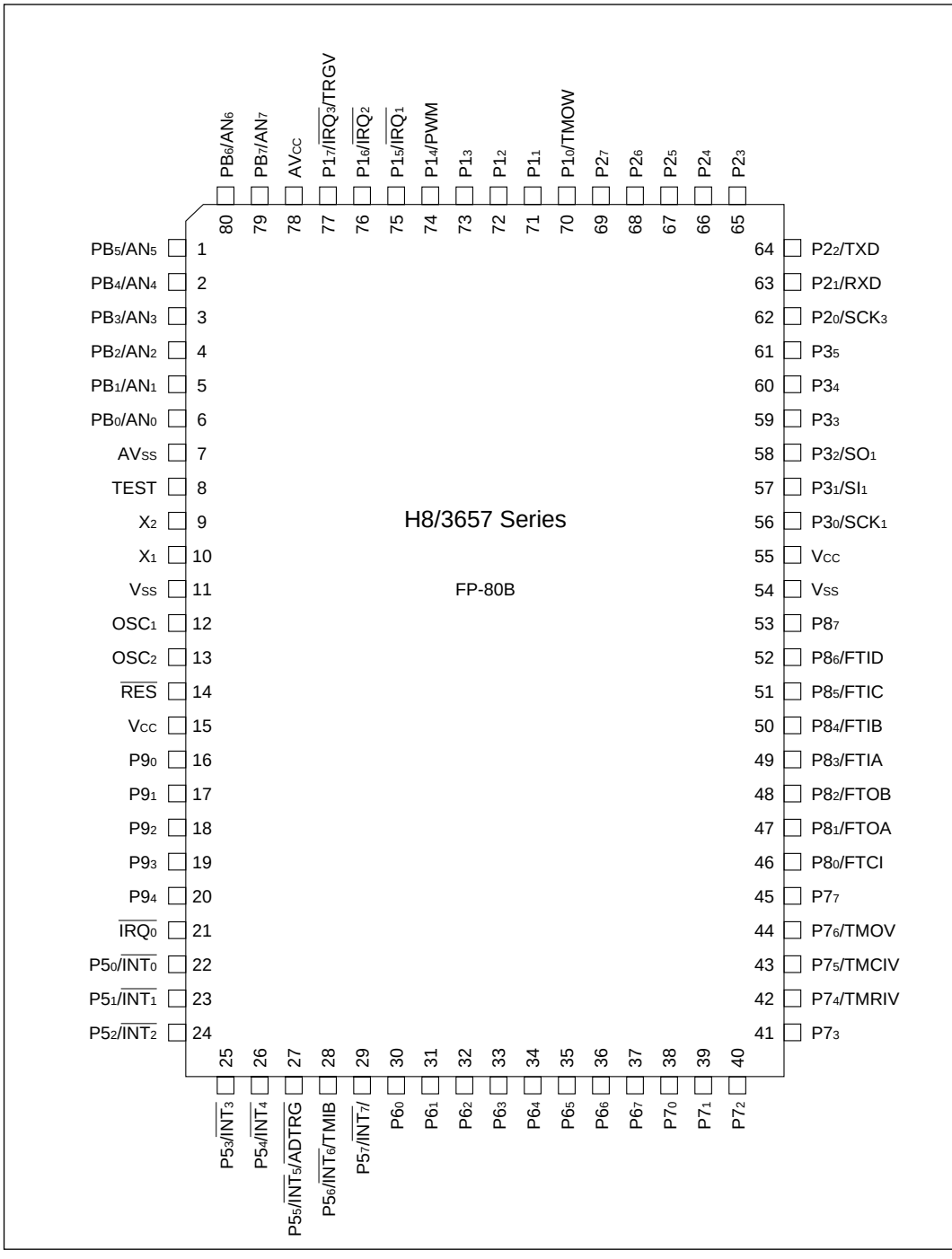


Figure 1-3 Pin Arrangement (FP-80B: Top View)

1.3.2 Pin Functions

Table 1-2 outlines the pin functions of the H8/3657 Series.

Table 1-2 Pin Functions

Type	Symbol	Pin No. TFP-80C, TFP-80F, FP-80A	Pin No. FP-80B	I/O	Name and Functions
Power source pins	V_{CC}	13, 53	15, 55	Input	**Power supply:** All V_{CC} pins should be connectedto the system power supply (+5 V)
	V_{SS}	9, 52	11, 54	Input	**Ground:** All V_{SS} pins should be connected to the system power supply (0 V)
	AV_{CC}	76	78	Input	**Analog power supply:** This is the power supply pin for the A/D converter. When the A/D converter is not used, connect this pin to the system power supply (+5 V).
	AV_{SS}	5	7	Input	**Analog ground:** This is the A/D converter ground pin. It should be connected to the system power supply (0 V).
Clock pins	OSC_1	10	12	Input	**System clock:** These pins connect to a crystal or ceramic oscillator, or can be used to input an external clock. See section 4, Clock Pulse Generators, for a typical connection diagram.
	OSC_2	11	13	Output	
	X_1	8	10	Input	**Subclock:** These pins connect to a 32.768-kHz crystal oscillator. See section 4, Clock Pulse Generators, for a typical connection diagram.
	X_2	7	9	Output	
System control	$\overline{RES}$	12	14	Input	**Reset:** When this pin is driven low, the chip is reset
	TEST	6	8	Input	**Test:** This is a test pin, not for use in application systems. It should be connected to V_{SS}.
Interrupt pins	$\overline{IRQ_0}$ $\overline{IRQ_1}$ $\overline{IRQ_2}$ $\overline{IRQ_3}$	19 73 74 75	21 75 76 77	Input	**IRQ interrupt request 0 to 3:** These are input pins for edge-sensitive external interrupts, with a selection of rising or falling edge
	$\overline{INT_7}$ to $\overline{INT_0}$	27 to 20	29 to 22	Input	**INT interrupt request 0 to 7:** These are input pins for edge-sensitive external interrupts, with a selection of rising or falling edge
Timer pins	TMOW	68	70	Output	**Clock output:** This is an output pin for waveforms generated by the timer A output circuit

Table 1-2 Pin Functions (cont)

Type	Symbol	Pin No. TFP-80C, TFP-80F, FP-80A	Pin No. FP-80B	I/O	Name and Functions
Timer pins	TMIB	26	28	Input	**Timer B1 event counter input:** This is an event input pin for input to the timer B1 counter
	TMOV	42	44	Output	**Timer V output:** This is an output pin for waveforms generated by the timer V output compare function
	TMCIV	41	43	Input	**Timer V event input:** This is an event input pin for input to the timer V counter
	TMRIV	40	42	Input	**Timer V counter reset:** This is a counter reset input pin for timer V
	TRGV	75	77	Input	**Timer V counter trigger input:** This is a trigger input pin for the timer V counter and realtime output port
	FTCI	44	46	Input	**Timer X clock input:** This is an external clock input pin for input to the timer X counter
	FTOA	45	47	Output	**Timer X output compare A output:** This is an output pin for timer X output compare A
	FTOB	46	48	Output	**Timer X output compare B output:** This is an output pin for timer X output compare B
	FTIA	47	49	Input	**Timer X input capture A input:** This is an input pin for timer X input capture A
	FTIB	48	50	Input	**Timer X input capture B input:** This is an input pin for timer X input capture B
	FTIC	49	51	Input	**Timer X input capture C input:** This isan input pin for timer X input capture C
	FTID	50	52	Input	**Timer X input capture D input:** This is an input pin for timer X input capture D
14-bit PWM pin	PWM	72	74	Output	**14-bit PWM output:** This is an output pin for waveforms generated by the 14-bit PWM
I/O ports	PB_7 to PB_0	77 to 80, 1 to 4	79 to 80 1 to 6	Input	**Port B:** This is an 8-bit input port
	$P1_7$ to $P1_0$	75 to 68	77 to 70	I/O	**Port 1:** This is a 8-bit I/O port. Input or output can be designated for each bit by means of port control register 1 (PCR1)

Table 1-2 Pin Functions (cont)

Type	Symbol	Pin No. TFP-80C, TFP-80F, FP-80A	Pin No. FP-80B	I/O	Name and Functions
I/O ports	$P2_7$ to $P2_0$	67 to 60	69 to 62	I/O	**Port 2:** This is a 8-bit I/O port. Input or output can be designated for each bit by means of port control register 2 (PCR2)
	$P3_5$ to $P3_0$	59 to 54	61 to 56	I/O	**Port 3:** This is a 6-bit I/O port. Input or output can be designated for each bit by means of port control register 3 (PCR3)
	$P5_7$ to $P5_0$	27 to 20	29 to 22	I/O	**Port 5:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 5 (PCR5)
	$P6_7$ to $P6_0$	35 to 28	37 to 30	I/O	**Port 6:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 6 (PCR6)
	$P7_7$ to $P7_0$	43 to 36	45 to 38	I/O	**Port 7:** This is a 8-bit I/O port. Input or output can be designated for each bit by means of port control register 7 (PCR7)
	$P8_7$ to $P8_0$	51 to 44	53 to 46	I/O	**Port 8:** This is an 8-bit I/O port. Input or output can be designated for each bit by means of port control register 8 (PCR8)
	$P9_4$ to $P9_0$	18 to 14	20 to 16	I/O	**Port 9:** This is a 5-bit I/O port. Input or output can be designated for each bit by means of port control register 9 (PCR9)
Serial communi-cation interface (SCI)	SI_1	55	57	Input	**SCI1 receive data input:** This is the SCI1 data input pin
	SO_1	56	58	Output	**SCI1 transmit data output:** This is the SCI1 data output pin
	SCK_1	54	56	I/O	**SCI1 clock I/O :** This is the SCI1 clock I/O pin
	RXD	61	63	Input	**SCI3 receive data input:** This is the SCI3 data input pin
	TXD	62	64	Output	**SCI3 transmit data output:** TThis is the SCI3 data output pin
	SCK_3	60	62	I/O	**SCI3 clock I/O:** This is the SCI3 clock I/O pin
A/D converter	AN_7 to An_0	77 to 80, 1 to 4	79, 80 1 to 6	Input	**Analog input channels 7 to 0:** These are analog data input channels to the A/D converter
	$\overline{ADTRG}$	25	27	Input	**A/D converter trigger input:** This is the external trigger input pin to the A/D converter

Section 2 CPU

2.1 Overview

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

2.1.1 Features

Features of the H8/300L CPU are listed below.

- General-register architecture
  Sixteen 8-bit general registers, also usable as eight 16-bit general registers

- Instruction set with 55 basic instructions, including:
  — Multiply and divide instructions
  — Powerful bit-manipulation instructions

- Eight addressing modes
  — Register direct
  — Register indirect
  — Register indirect with displacement
  — Register indirect with post-increment or pre-decrement
  — Absolute address
  — Immediate
  — Program-counter relative
  — Memory indirect

- 64-kbyte address space

- High-speed operation
  — All frequently used instructions are executed in two to four states
  — High-speed arithmetic and logic operations
  — 8- or 16-bit register-register add or subtract: 0.4 µs*
  — 8 × 8-bit multiply: 2.8 µs*
  — 16 ÷ 8-bit divide: 2.8 µs*

- Low-power operation modes
  SLEEP instruction for transfer to low-power operation

Note: * These values are at ø = 5 MHz.

2.1.2 Address Space

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

See 2.8, Memory Map, for details of the memory map.

2.1.3 Register Configuration

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

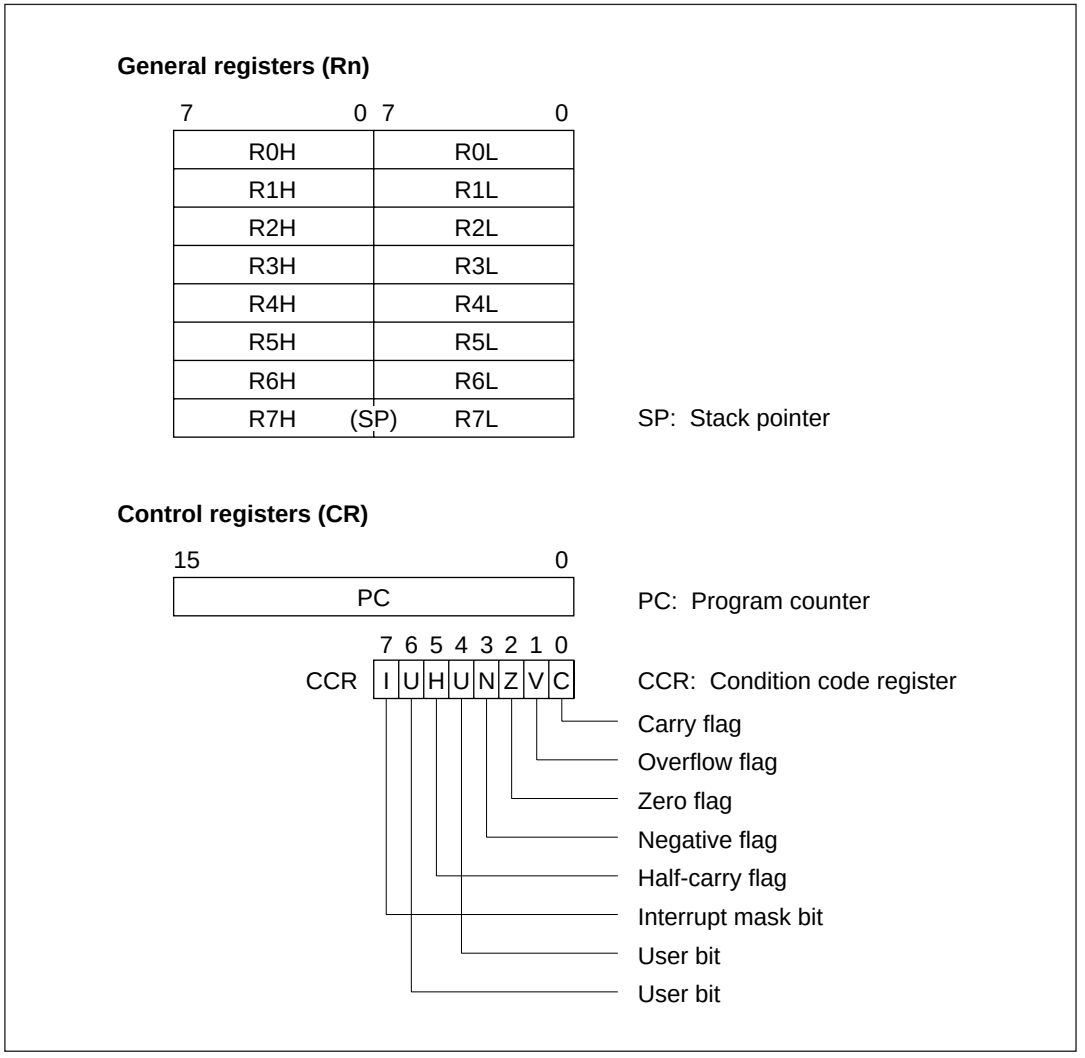


Figure 2-1 CPU Registers

2.2 Register Descriptions

2.2.1 General Registers

All the general registers can be used as both data registers and address registers.

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

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

R7 also functions as the stack pointer (SP), used implicitly by hardware in exception processing and subroutine calls. When it functions as the stack pointer, as indicated in figure 2-2, SP (R7) points to the top of the stack.

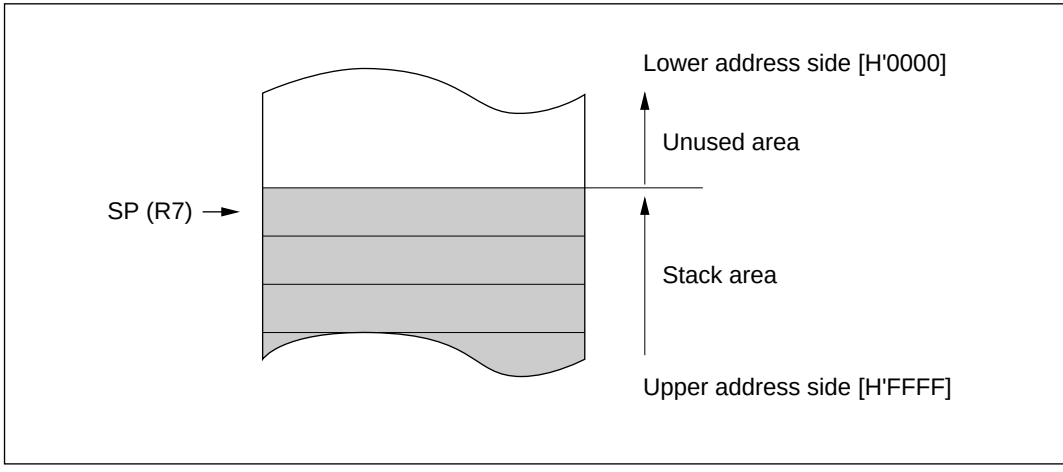


Figure 2-2 Stack Pointer

2.2.2 Control Registers

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

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

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

Bit 7—Interrupt Mask Bit (I): When this bit is set to 1, interrupts are masked. This bit is set to 1 automatically at the start of exception handling. The interrupt mask bit may be read and written by software. For further details, see section 3.3, Interrupts.

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

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

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

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

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

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

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

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

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

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

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

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

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

2.2.3 Initial Register Values

When the CPU is reset, the program counter (PC) is initialized to the value stored at address H'0000 in the vector table, and the I bit in the CCR is set to 1. The other CCR bits and the general registers are not initialized. In particular, the stack pointer (R7) is not initialized. The stack pointer should be initialized by software, by the first instruction executed after a reset.

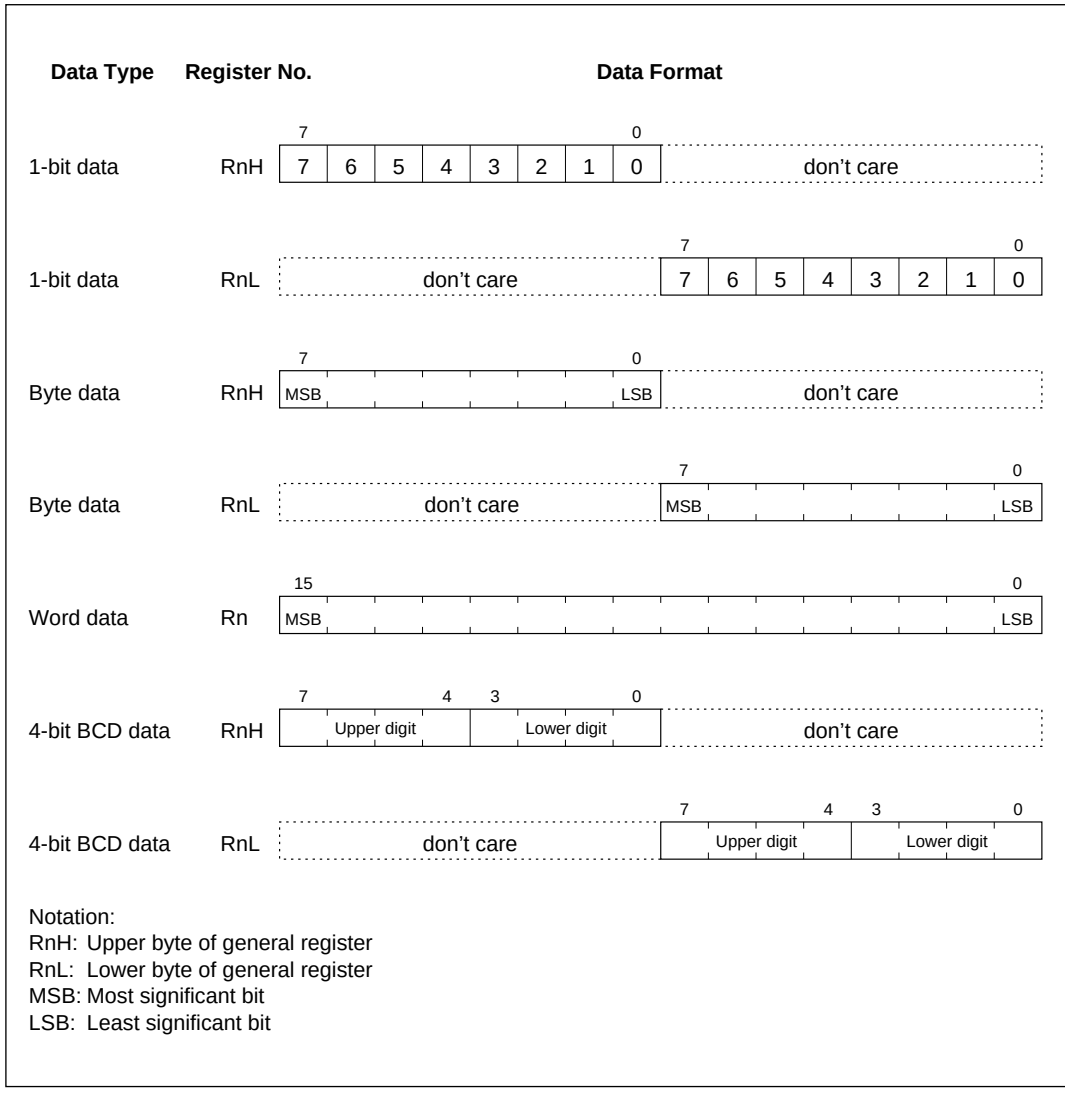
2.3 Data Formats

The H8/300L 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 in a byte operand (n = 0, 1, 2, ..., 7).
- All arithmetic and logic instructions except ADDS and SUBS can operate on byte data.
- 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.
- 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.

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.



Notation:
RnH: Upper byte of general register
RnL: Lower byte of general register
MSB: Most significant bit
LSB: Least significant bit

Figure 2-3 Register Data Formats

2.3.2 Memory Data Formats

Figure 2-4 indicates the data formats in memory. For access by the H8/300L CPU, word data stored in memory must always begin at an even address. In word access the least significant bit of the address is regarded as 0. If an odd address is specified, the access is performed at the preceding even address. This rule affects the MOV.W instruction, and also applies to instruction fetching.

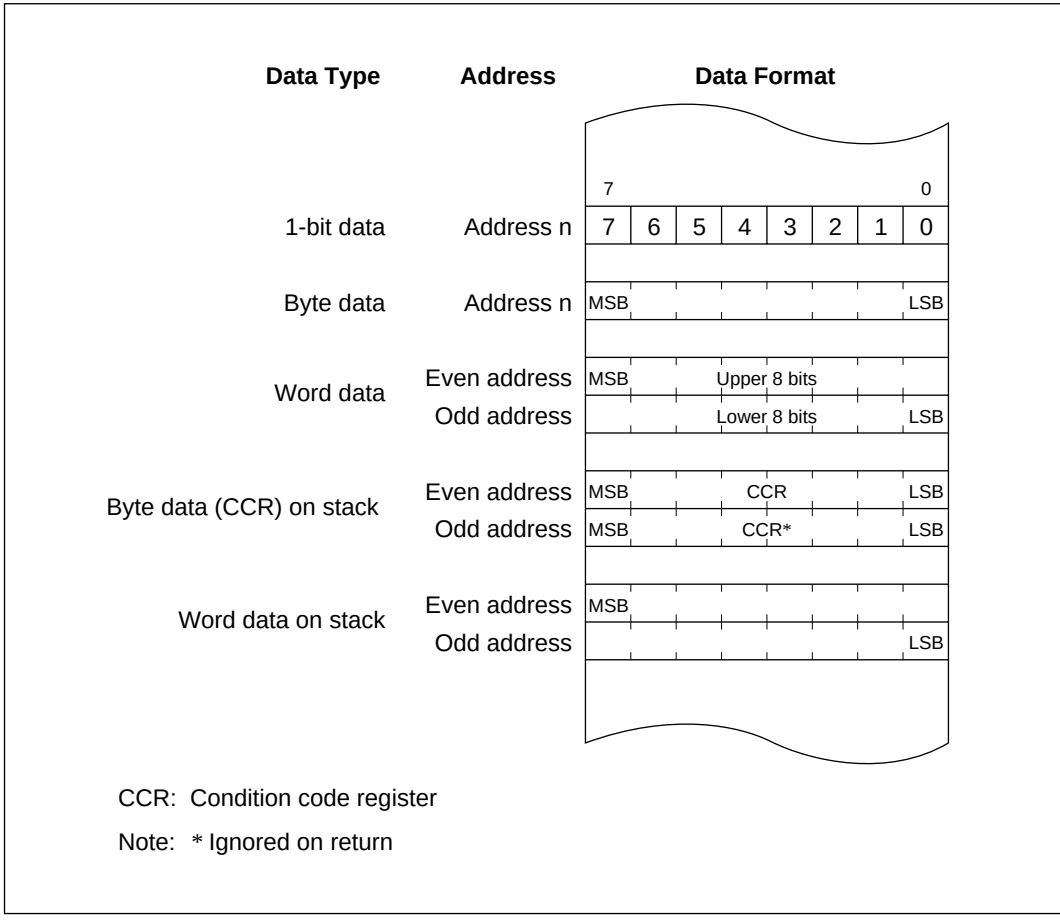


Figure 2-4 Memory Data Formats

When the stack is accessed using R7 as an address register, word access should always be performed. 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 Modes

The H8/300L CPU supports the eight addressing modes listed in table 2-1. Each instruction uses a subset of these addressing modes.

Table 2-1 Addressing Modes

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

1. **Register Direct—Rn:** The register field of the instruction specifies an 8- or 16-bit general register containing the operand.

   Only the MOV.W, ADD.W, SUB.W, CMP.W, ADDS, SUBS, MULXU (8 bits × 8 bits), and DIVXU (16 bits ÷ 8 bits) instructions have 16-bit operands.

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

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

   This mode is used only in MOV instructions. For the MOV.W instruction, the resulting address must be even.

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

   - Register indirect with post-increment—@Rn+

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

     The register field of the instruction specifies a 16-bit general register containing the address of the operand. After the operand is accessed, the register is incremented by 1 for MOV.B or 2 for MOV.W. For MOV.W, the original contents of the 16-bit general register must be even.

   - Register indirect with pre-decrement—@–Rn

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

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

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

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

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

6. **Immediate—#xx:8 or #xx:16:** The instruction contains an 8-bit operand (#xx:8) in its second byte, or a 16-bit operand (#xx:16) 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 in the second or fourth byte of the instruction, specifying a bit number.

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

8. **Memory Indirect—@@aa:8:** This mode can be used by the JMP and JSR instructions. The second byte of the instruction code specifies an 8-bit absolute address. The word located at this address contains the branch destination address.

   The upper 8 bits of the absolute address are assumed to be 0 (H'00), so the address range is from H'0000 to H'00FF (0 to 255). Note that with the H8/300L Series, the lower end of the address area is also used as a vector area. See 3.3, Interrupts, for details on the vector area.

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

2.4.2 Effective Address Calculation

Table 2-2 shows how effective addresses are calculated in each of the addressing modes.

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

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

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

Table 2-2 Effective Address Calculation

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

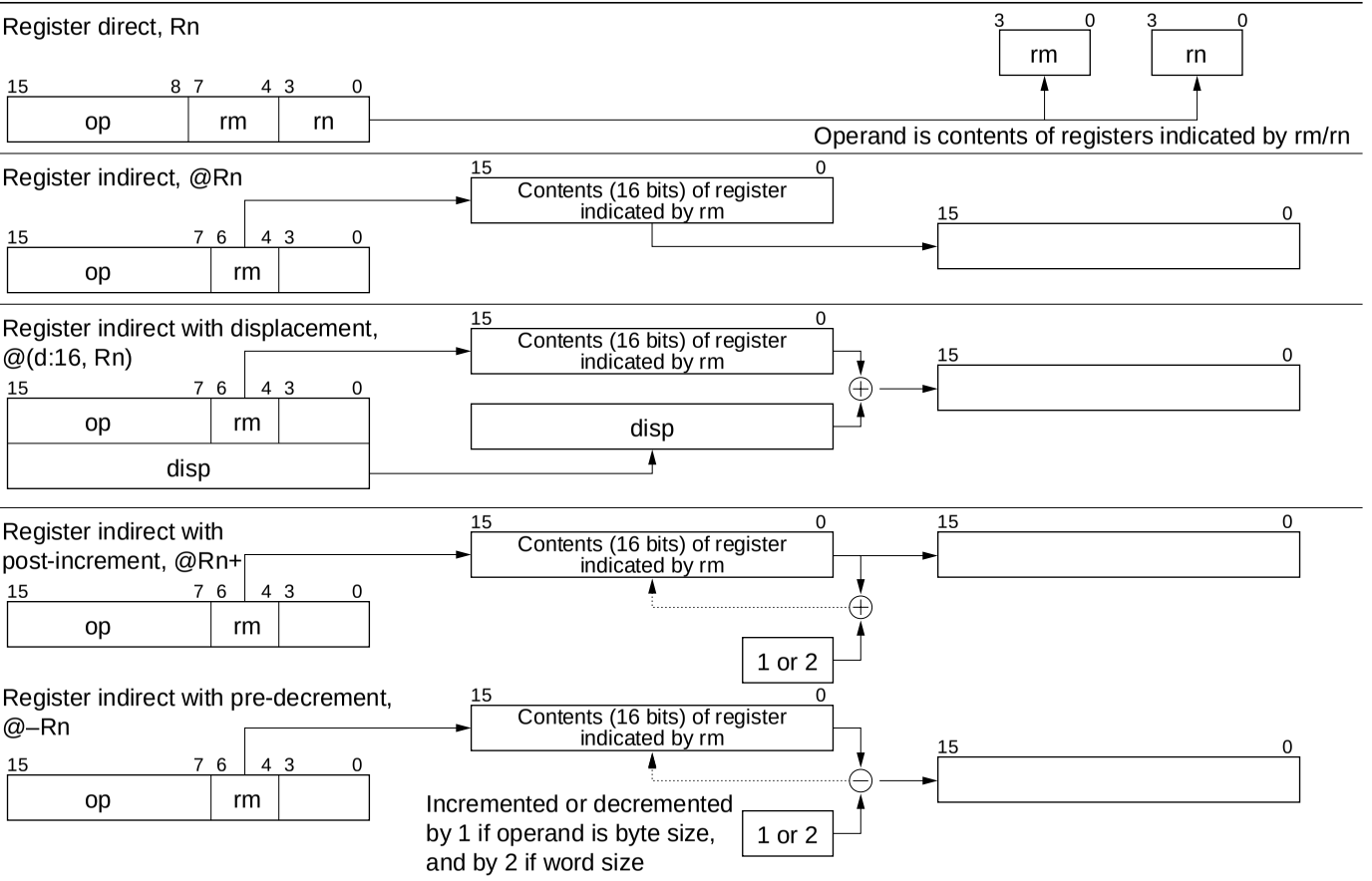


Table 2-2 Effective Address Calculation (cont)

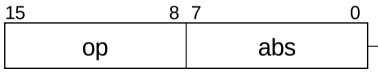
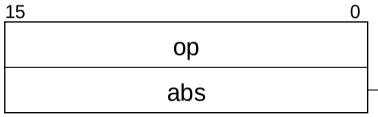
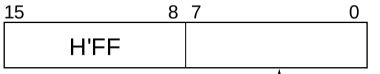
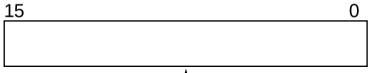
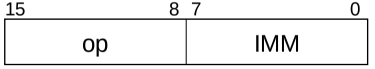
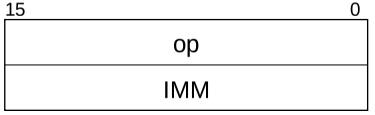
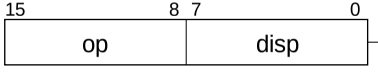
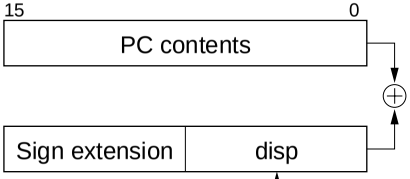
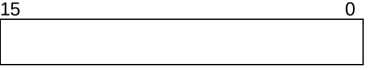
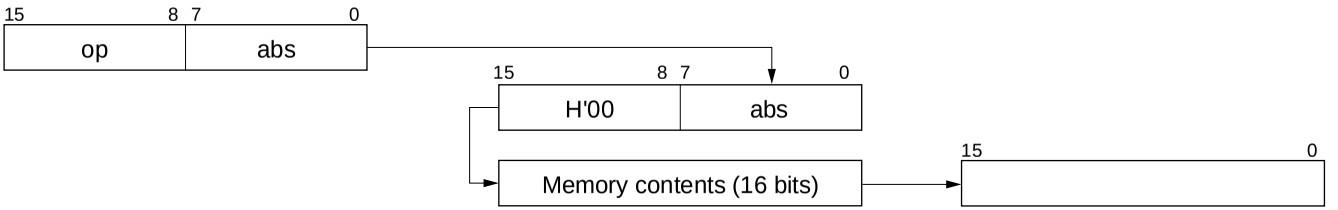
No.	Addressing Mode and Instruction Format	Effective Address Calculation Method	Effective Address (EA)
5	Absolute address @aa:8  @aa:16  		 
6	Immediate #xx:8  #xx:16  	Operand is 1- or 2-byte immediate data	
7	Program-counter relative @(d:8, PC)  	 	

Table 2-2 Effective Address Calculation (cont)

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

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

2.5 Instruction Set

The H8/300L Series can use a total of 55 instructions, which are grouped by function in table 2-3.

Table 2-3 Instruction Set

Function	Instructions	Number
Data transfer	MOV, PUSH*1, POP*1	1
Arithmetic operations	ADD, SUB, ADDX, SUBX, INC, DEC, ADDS, SUBS, DAA, DAS, MULXU, DIVXU, CMP, NEG	14
Logic operations	AND, OR, XOR, NOT	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR, BLD, BILD, BST, BIST	14
Branch	Bcc*2, JMP, BSR, JSR, RTS	5
System control	RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	8
Block data transfer	EEPMOV	1
		Total: 55

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

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.

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

2.5.1 Data Transfer Instructions

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

Table 2-4 Data Transfer Instructions

Instruction	Size*	Function
MOV	B/W	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register. The Rn, @Rn, @(d:16, Rn), @aa:16, #xx:16, @–Rn, and @Rn+ addressing modes are available for 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.
POP	W	@SP+ → Rn Pops a 16-bit general register from the stack. Equivalent to MOV.W @SP+, Rn.
PUSH	W	Rn → @–SP Pushes a 16-bit general register onto the stack. Equivalent to MOV.W Rn, @–SP.

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

Certain precautions are required in data access. See 2.9.1, Notes on Data Access, for details.

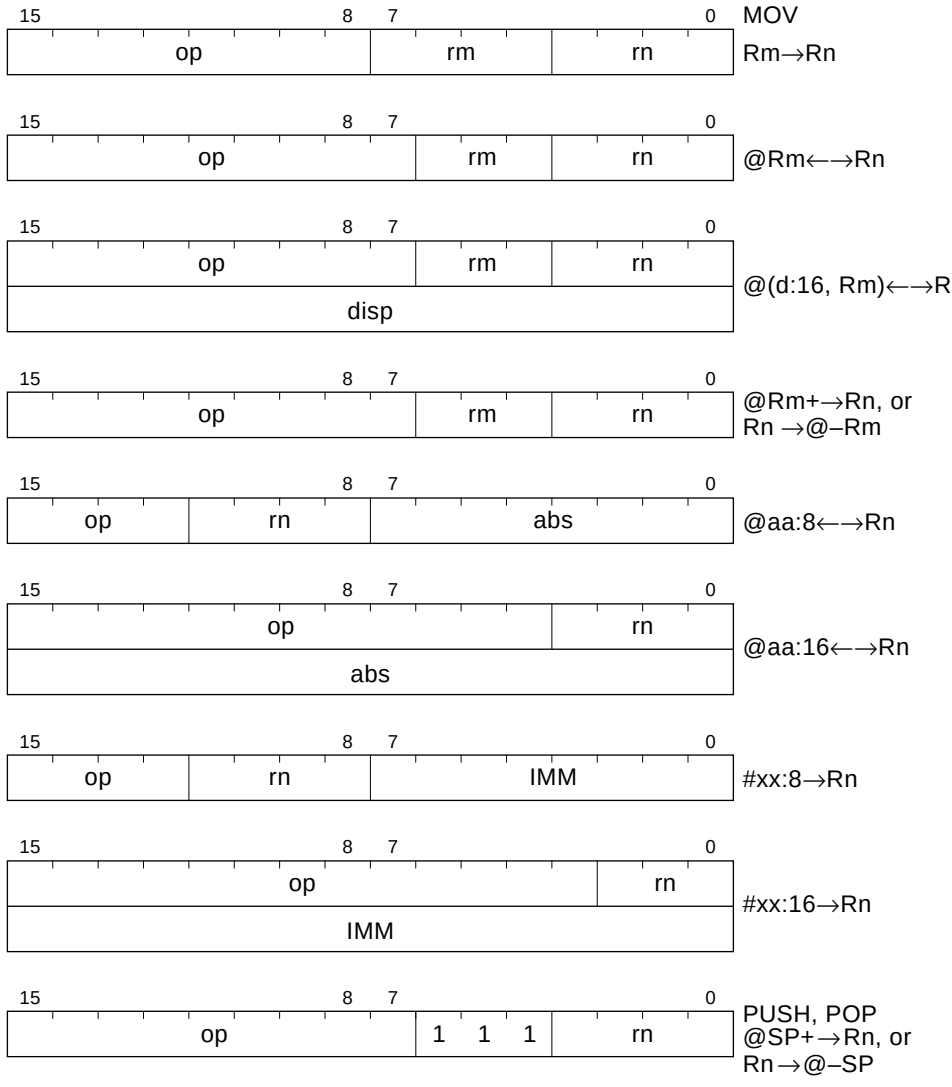


Figure 2-5 Data Transfer Instruction Codes

2.5.2 Arithmetic Operations

Table 2-5 describes the arithmetic instructions.

Table 2-5 Arithmetic Instructions

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

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

2.5.3 Logic Operations

Table 2-6 describes the four instructions that perform logic operations.

Table 2-6 Logic Operation Instructions

Instruction	Size*	Function
AND	B	Rd $\wedge$ Rs $\rightarrow$ Rd, Rd $\wedge$ #IMM $\rightarrow$ Rd Performs a logical AND operation on a general register and another general register or immediate data
OR	B	Rd $\vee$ Rs $\rightarrow$ Rd, Rd $\vee$ #IMM $\rightarrow$ Rd Performs a logical OR operation on a general register and another general register or immediate data
XOR	B	Rd $\oplus$ Rs $\rightarrow$ Rd, Rd $\oplus$ #IMM $\rightarrow$ Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data
NOT	B	~ Rd $\rightarrow$ Rd Obtains the one's complement (logical complement) of general register contents

Notes: * Size: Operand size
B: Byte

2.5.4 Shift Operations

Table 2-7 describes the eight shift instructions.

Table 2-7 Shift Instructions

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

Notes: * Size: Operand size
B: Byte

Figure 2-6 shows the instruction code format of arithmetic, logic, and shift instructions.

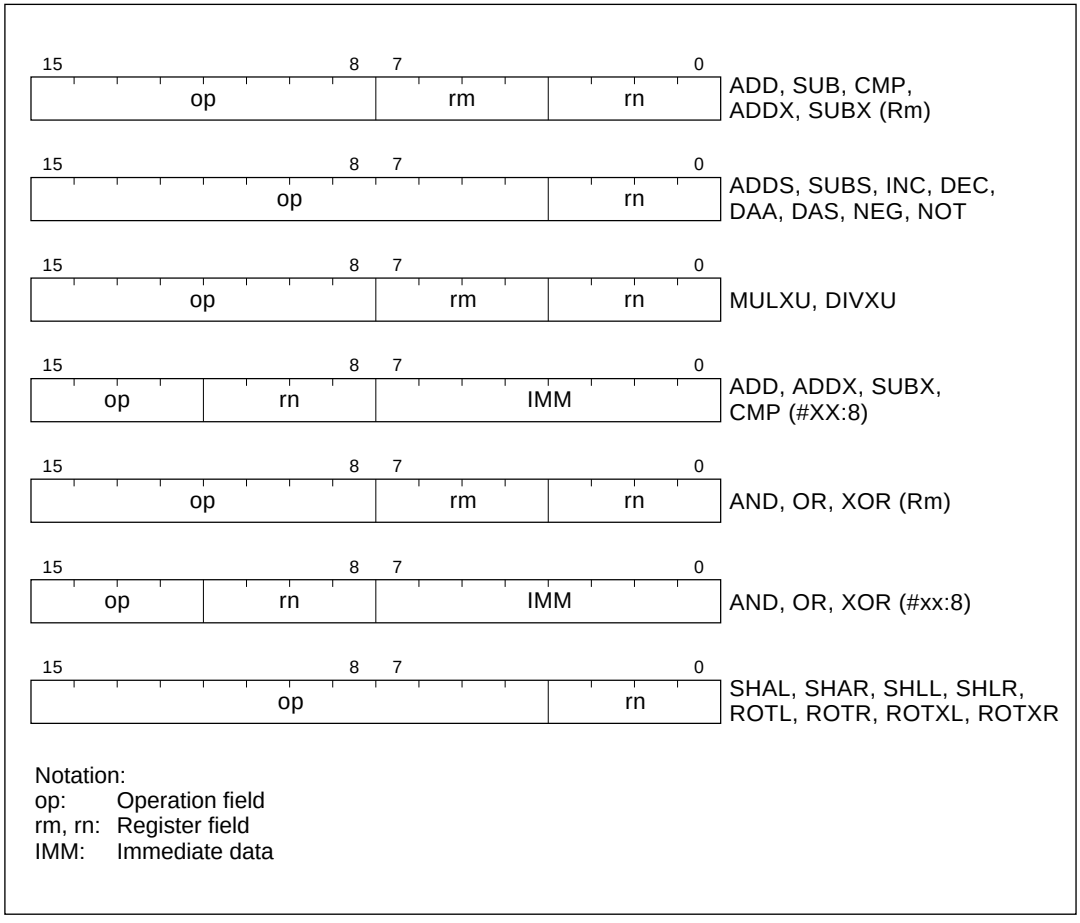


Figure 2-6 Arithmetic, Logic, and Shift Instruction Codes

2.5.5 Bit Manipulations

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

Table 2-8 Bit-Manipulation Instructions

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

Notes: * Size: Operand size
B: Byte

Table 2-8 Bit-Manipulation Instructions (cont)

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

Notes: * Size: Operand size
B: Byte

Certain precautions are required in bit manipulation. See 2.9.2, Notes on Bit Manipulation, for details.

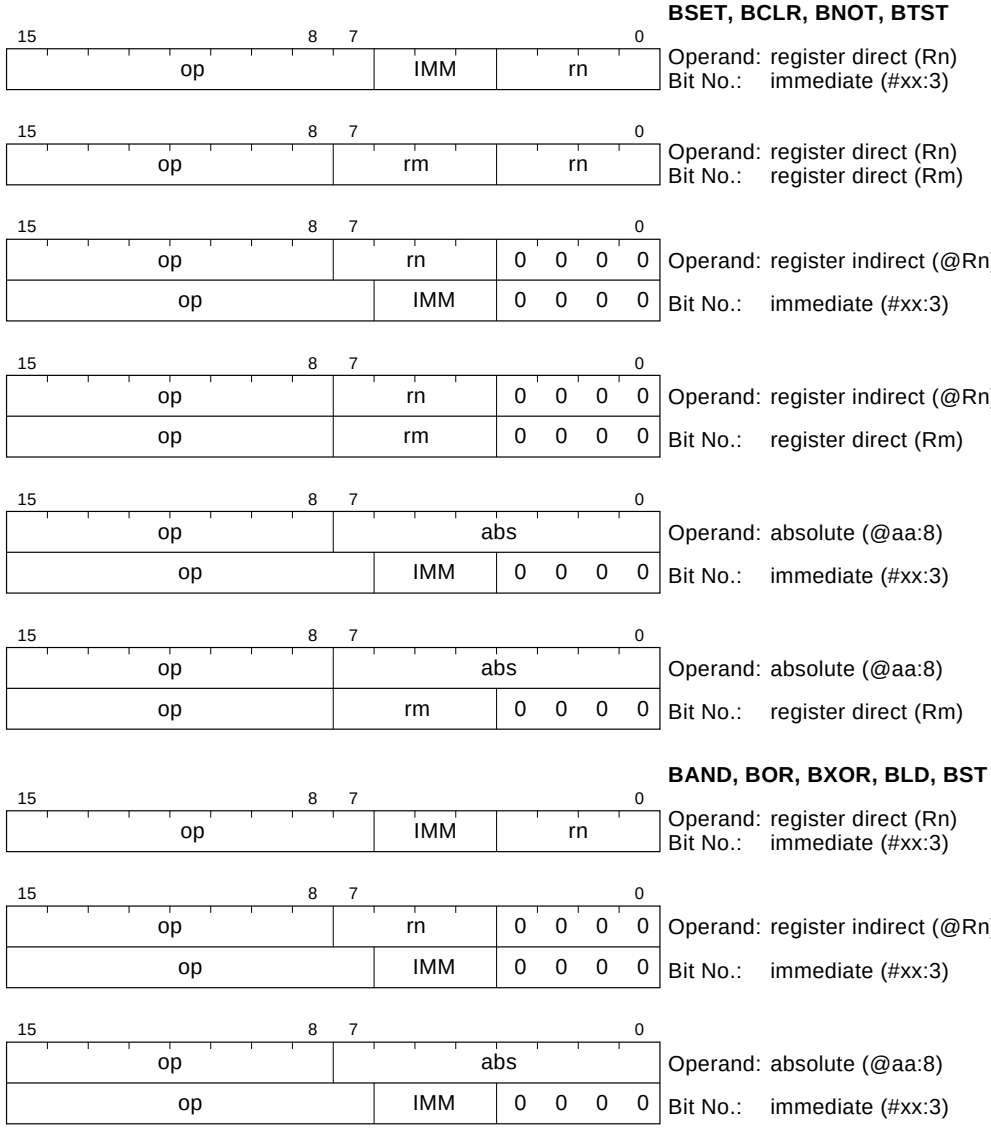
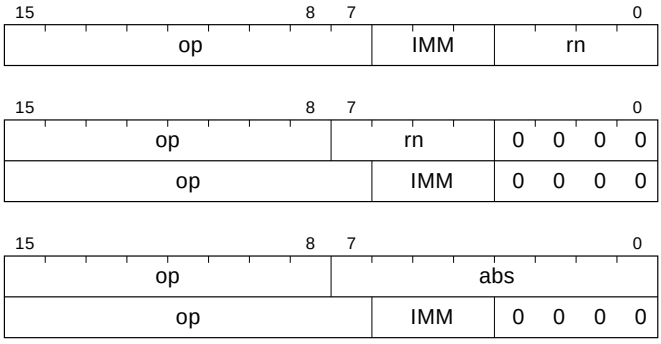


Figure 2-7 Bit Manipulation Instruction Codes

BIAND, BIOR, BIXOR, BILD, BIST



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

Figure 2-7 Bit Manipulation Instruction Codes (cont)

2.5.6 Branching Instructions

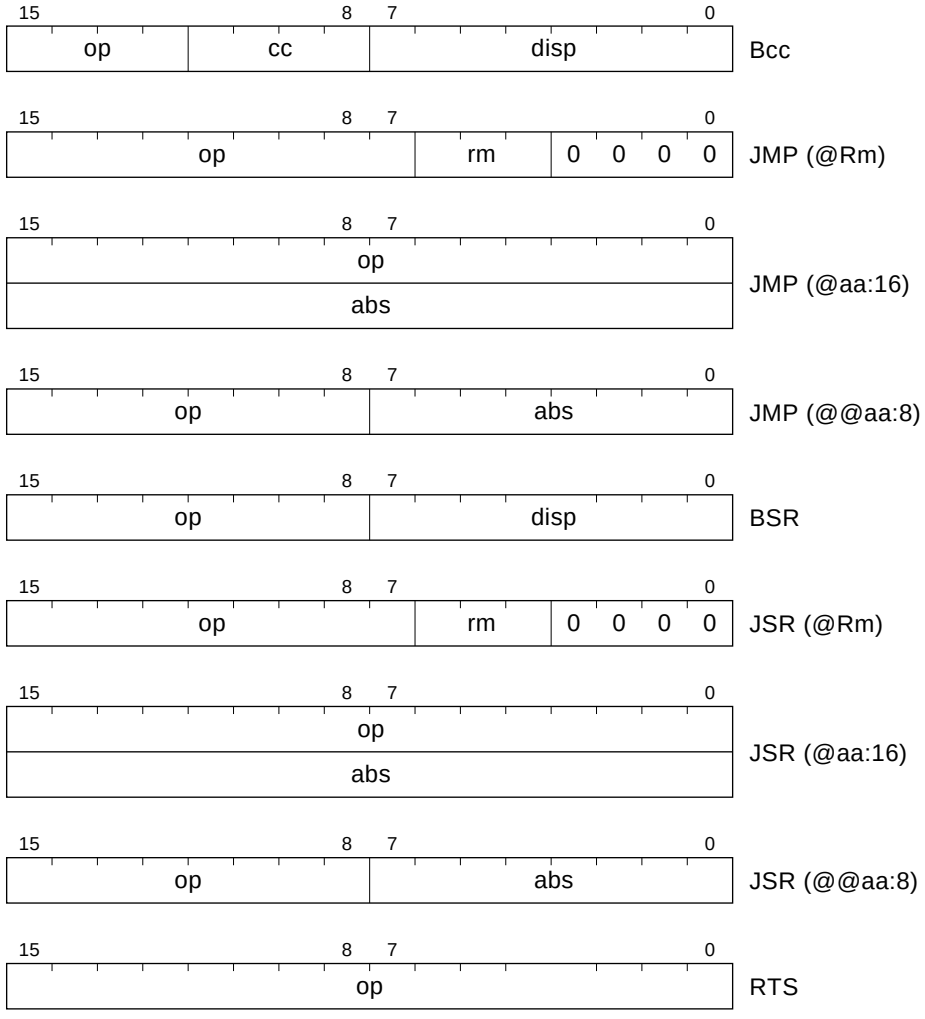
Table 2-9 describes the branching instructions. Figure 2-8 shows their object code formats.

Table 2-9 Branching Instructions

Instruction	Size	Function
Bcc	—	Branches to the designated address if condition cc is true. The branching conditions are given below.

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

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



Notation:
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 from active mode to a power-down mode. See section 5, Power-Down Modes, for details.
LDC	B	Rs → CCR, #IMM → CCR Moves immediate data or general register contents to the condition code register
STC	B	CCR → Rd Copies the condition code register to a specified general register
ANDC	B	CCR ∧ #IMM → CCR Logically ANDs the condition code register with immediate data
ORC	B	CCR ∨ #IMM → CCR Logically ORs the condition code register with immediate data
XORC	B	CCR ⊕ #IMM → CCR Logically exclusive-ORs the condition code register with immediate data
NOP	—	PC + 2 → PC Only increments the program counter

Notes: * Size: Operand size
B: Byte

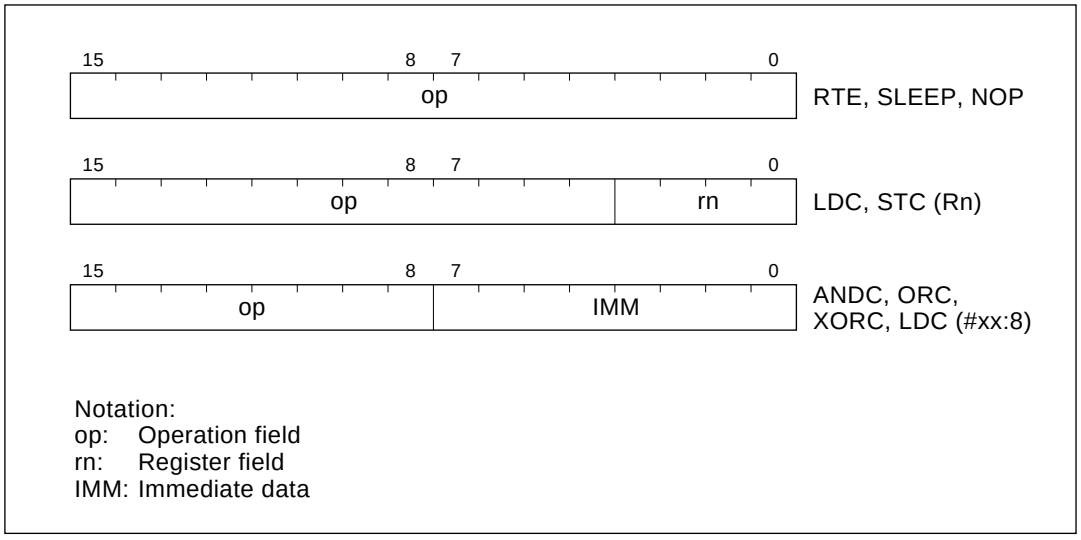


Figure 2-9 System Control Instruction Codes

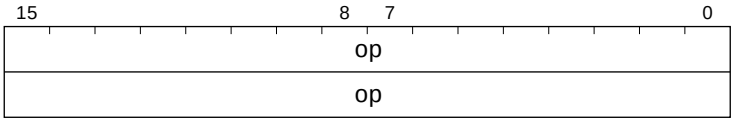
2.5.8 Block Data Transfer Instruction

Table 2-11 describes the block data transfer instruction. Figure 2-10 shows its object code format.

Table 2-11 Block Data Transfer Instruction

Instruction	Size	Function
EEPMOV	—	If R4L ≠ 0 then repeat @R5+ → @R6+ R4L – 1 → R4L until R4L = 0 else next; Block transfer instruction. Transfers the number of data bytes specified by R4L from locations starting at the address indicated by R5 to locations starting at the address indicated by R6. After the transfer, the next instruction is executed.

Certain precautions are required in using the EEPMOV instruction. See 2.9.3, Notes on Use of the EEPMOV Instruction, for details.



Notation:
op: Operation field

Figure 2-10 Block Data Transfer Instruction Code

2.6 Basic Operational Timing

CPU operation is synchronized by a system clock (ø) or a subclock ($ø_{SUB}$). For details on these clock signals see section 4, Clock Pulse Generators. The period from a rising edge of ø or $ø_{SUB}$ to the next rising edge is called one state. A bus cycle consists of two states or three states. The cycle differs depending on whether access is to on-chip memory or to on-chip peripheral modules.

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

Access to on-chip memory takes place in two states. The data bus width is 16 bits, allowing access in byte or word size. Figure 2-11 shows the on-chip memory access cycle.

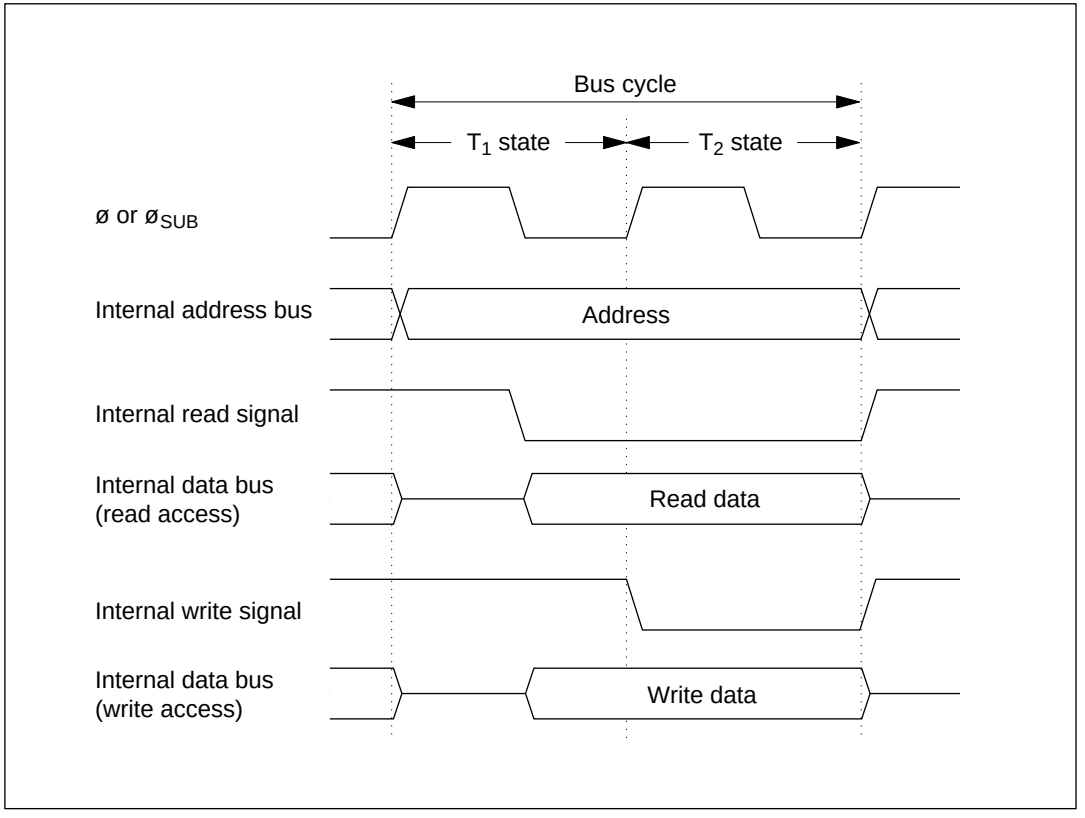


Figure 2-11 On-Chip Memory Access Cycle

2.6.2 Access to On-Chip Peripheral Modules

On-chip peripheral modules are accessed in two states or three states. The data bus width is 8 bits, so access is by byte size only. This means that for accessing word data, two instructions must be used. Figures 2-12 and 2-13 show the on-chip peripheral module access cycle.

Two-state access to on-chip peripheral modules

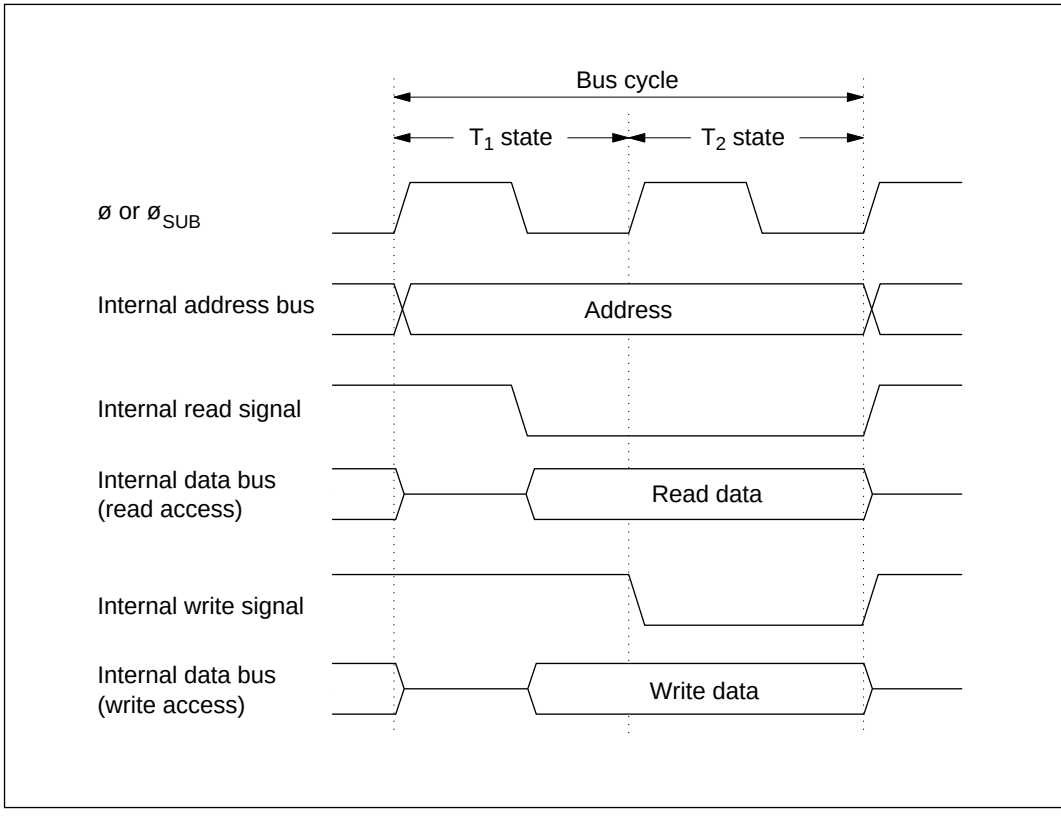


Figure 2-12 On-Chip Peripheral Module Access Cycle (2-State Access)

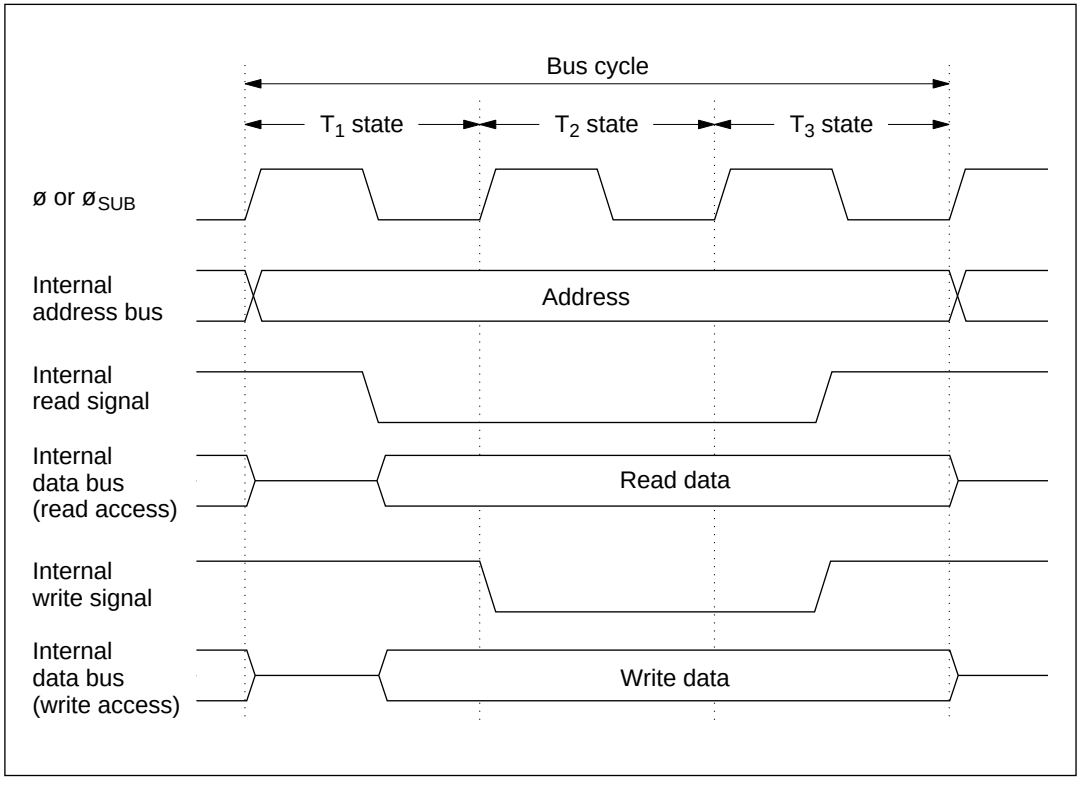


Figure 2-13 On-Chip Peripheral Module Access Cycle (3-State Access)

2.7 CPU States

2.7.1 Overview

There are four CPU states: the reset state, program execution state, program halt state, and exception-handling state. The program execution state includes active (high-speed or medium-speed) mode and subactive mode. In the program halt state there are a sleep (high-speed or medium-speed) mode, standby mode, watch mode, and sub-sleep mode. These states are shown in figure 2-14. Figure 2-15 shows the state transitions.

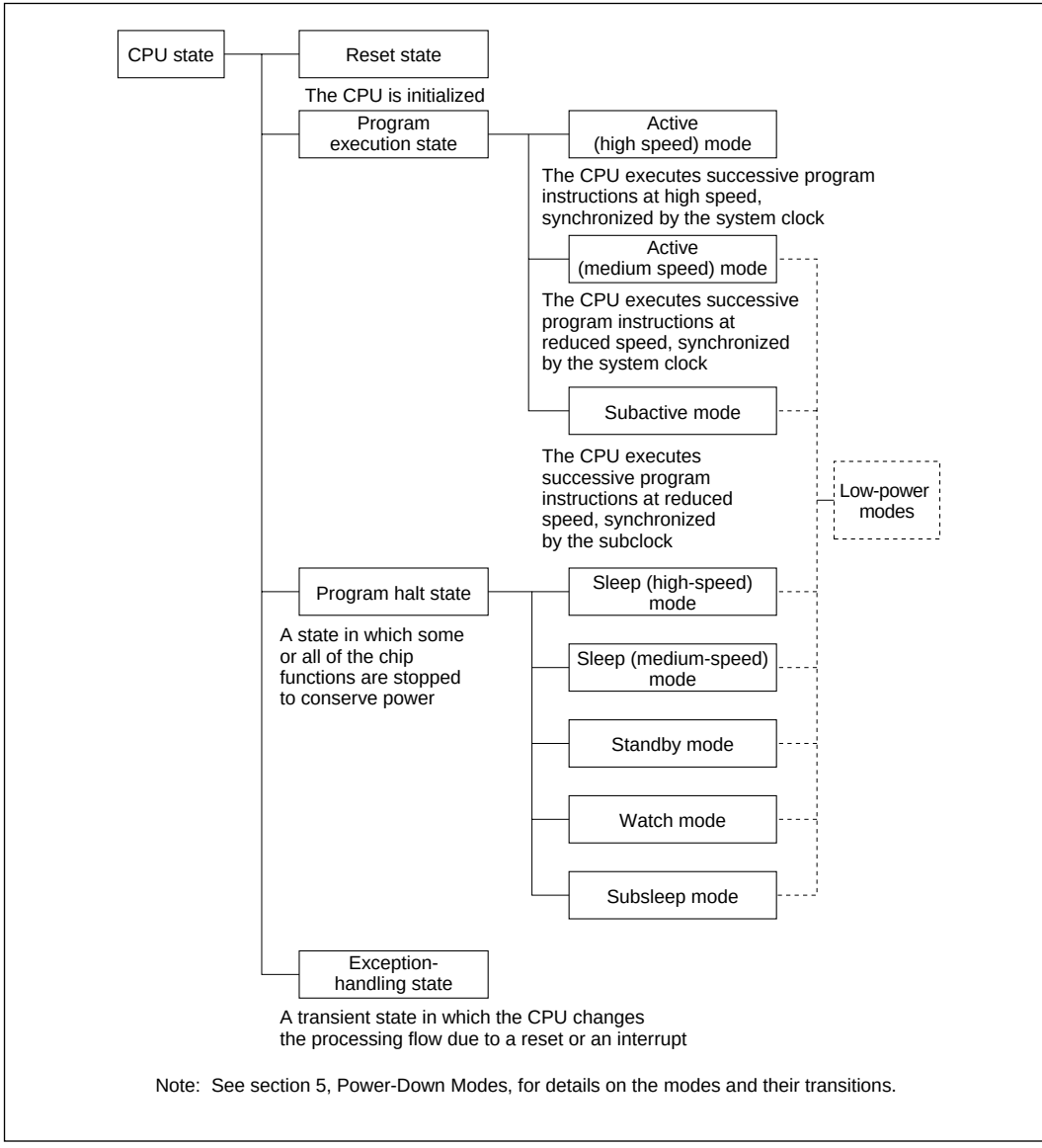


Figure 2-14 CPU Operation States

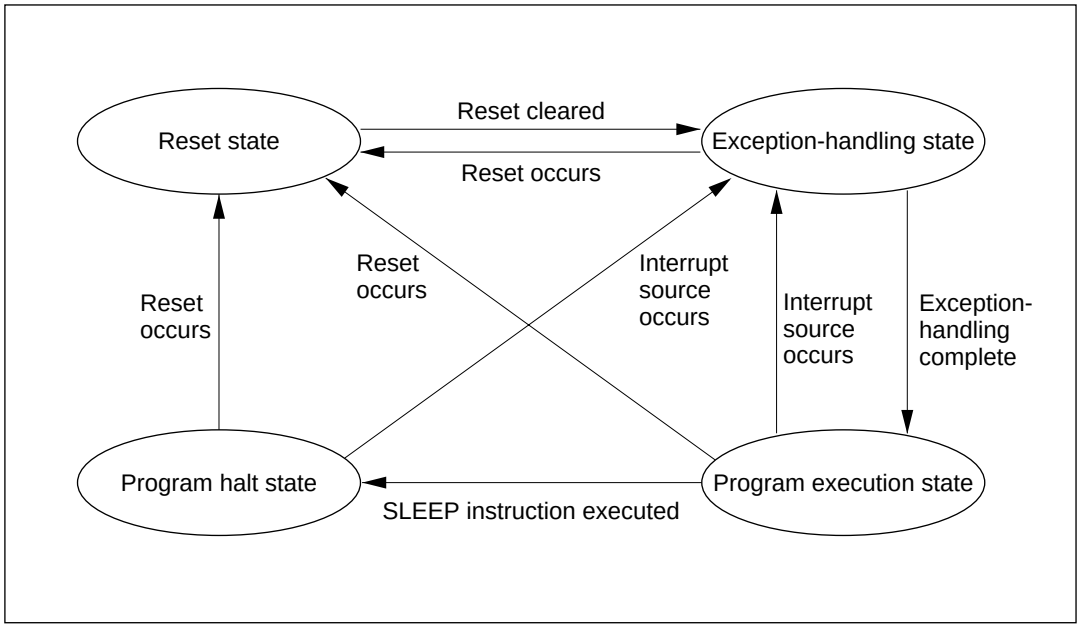


Figure 2-15 State Transitions

2.7.2 Program Execution State

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

There are three modes in this state, two active modes (high speed and medium speed) and one subactive mode. Operation is synchronized with the system clock in active mode (high speed and medium speed), and with the subclock in subactive mode. See section 5, Power-Down Modes for details on these modes.

2.7.3 Program Halt State

In the program halt state there are four modes: two sleep modes (high speed and medium speed), standby mode, watch mode, and subsleep mode. See section 5, Power-Down Modes for details on these modes.

2.7.4 Exception-Handling State

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

For details on interrupt handling, see section 3.3, Interrupts.

2.8 Memory Map

Figure 2-16 shows a memory map of the H8/3657 Series.

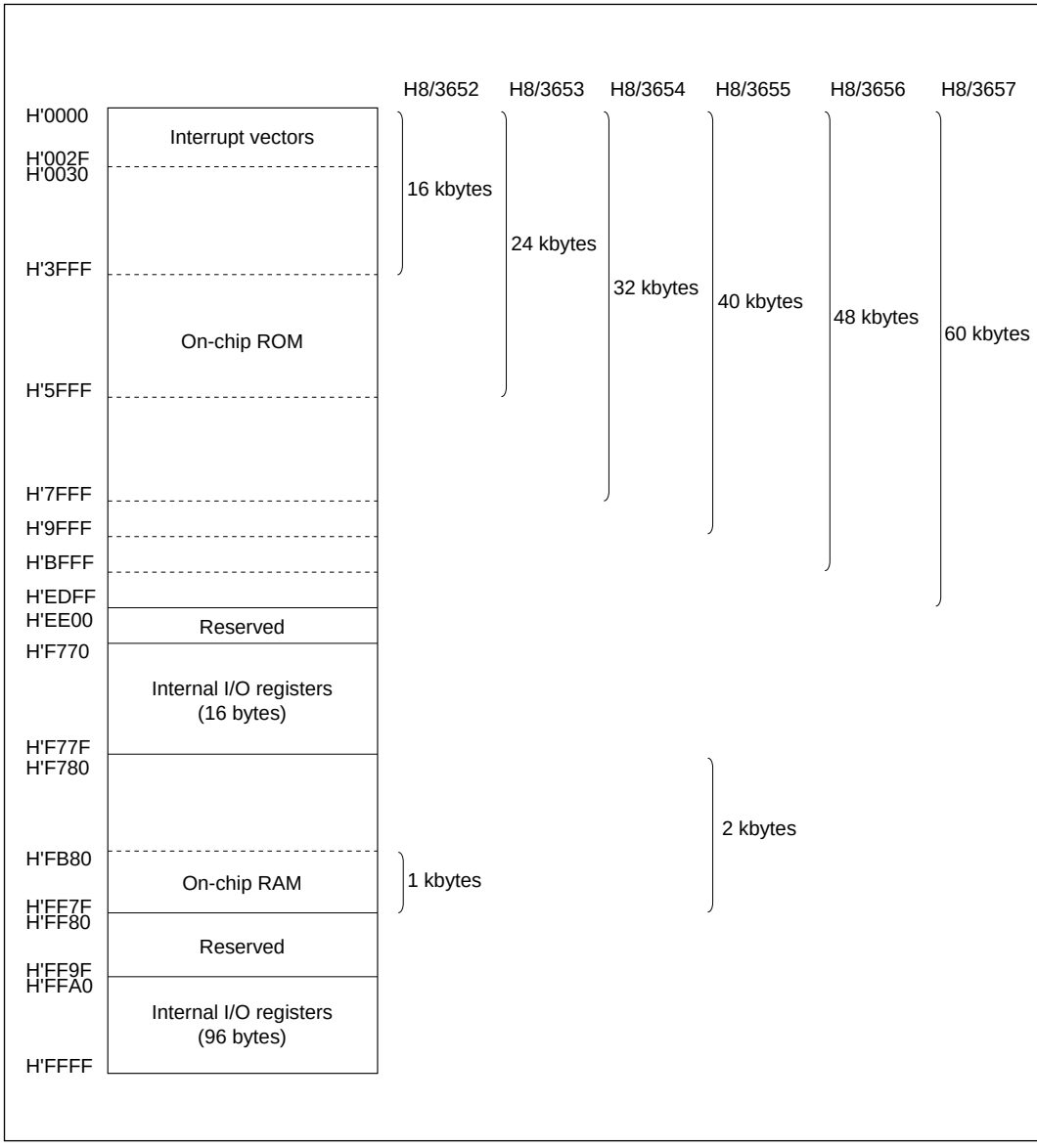


Figure 2-16 H8/3657 Series Memory Map

2.9 Application Notes

2.9.1 Notes on Data Access

1. The address space of the H8/300L CPU includes empty areas in addition to the RAM, registers, and ROM areas available to the user. If these empty areas are mistakenly accessed by an application program, the following results will occur.

   Data transfer from CPU to empty area:

   The transferred data will be lost. This action may also cause the CPU to misoperate.

   Data transfer from empty area to CPU:

   Unpredictable data is transferred.

2. Internal data transfer to or from on-chip modules other than the ROM and RAM areas makes use of an 8-bit data width. If word access is attempted to these areas, the following results will occur.

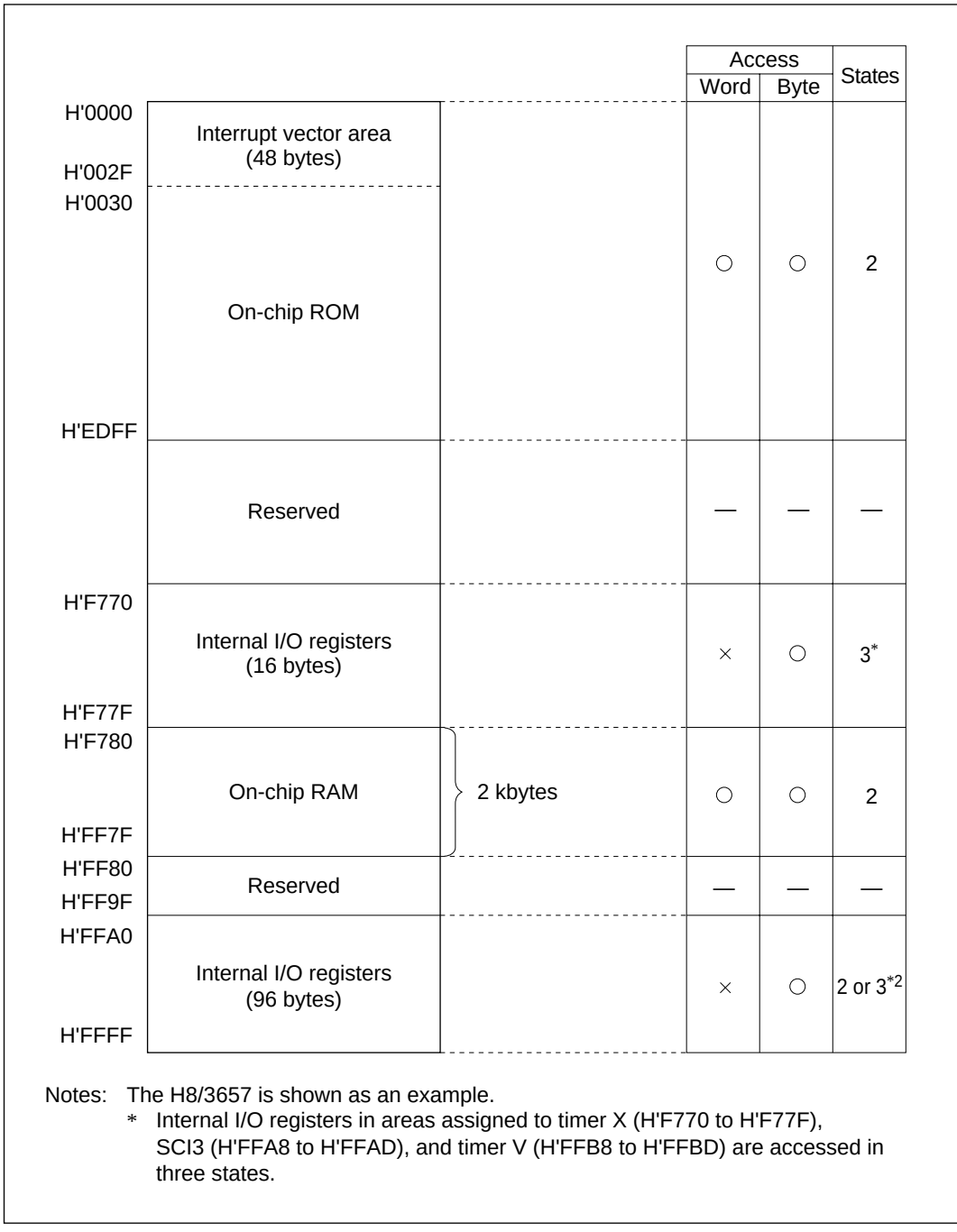
   Word access from CPU to I/O register area:

   Upper byte: Will be written to I/O register.
   Lower byte: Transferred data will be lost.

   Word access from I/O register to CPU:

   Upper byte: Will be written to upper part of CPU register.
   Lower byte: Unpredictable data will be written to lower part of CPU register.

Byte size instructions should therefore be used when transferring data to or from I/O registers other than the on-chip ROM and RAM areas. Figure 2-17 shows the data size and number of states in which on-chip peripheral modules can be accessed.



Notes: The H8/3657 is shown as an example.

* Internal I/O registers in areas assigned to timer X (H'F770 to H'F77F), SCI3 (H'FFA8 to H'FFAD), and timer V (H'FFB8 to H'FFBD) are accessed in three states.

Figure 2-17 Data Size and Number of States for Access to and from On-Chip Peripheral Modules

2.9.2 Notes on Bit Manipulation

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

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

1. Bit manipulation in two registers assigned to the same address

Example 1: timer load register and timer counter

Figure 2-18 shows an example in which two timer registers share the same address. When a bit manipulation instruction accesses the timer load register and timer counter of a reloadable timer, since these two registers share the same address, the following operations take place.

Order of Operation		Operation
1	Read	Timer counter data is read (one byte)
2	Modify	The CPU modifies (sets or resets) the bit designated in the instruction
3	Write	The altered byte data is written to the timer load register

The timer counter is counting, so the value read is not necessarily the same as the value in the timer load register. As a result, bits other than the intended bit in the timer load register may be modified to the timer counter value.

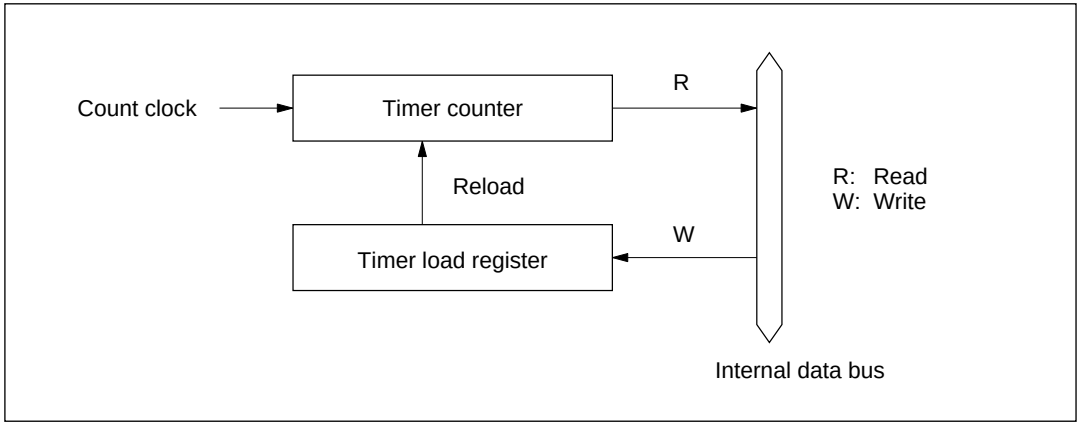


Figure 2-18 Timer Configuration Example

Example 2: BSET instruction executed designating port 3

$P3_7$ and $P3_6$ are designated as input pins, with a low-level signal input at $P3_7$ and a high-level signal at $P3_6$. The remaining pins, $P3_5$ to $P3_0$, are output pins and output low-level signals. In this example, the BSET instruction is used to change pin $P3_0$ to high-level output.

[A: Prior to executing BSET]

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0

[B: BSET instruction executed]

```
BSET   #0  ,  @PDR3
```

The BSET instruction is executed designating port 3.

[C: After executing BSET]

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	0	0	1	1	1	1	1	1
PDR3	0	1	0	0	0	0	0	1

[D: Explanation of how BSET operates]

When the BSET instruction is executed, first the CPU reads port 3.

Since $P3_7$ and $P3_6$ are input pins, the CPU reads the pin states (low-level and high-level input). $P3_5$ to $P3_0$ are output pins, so the CPU reads the value in PDR3. In this example PDR3 has a value of H'80, but the value read by the CPU is H'40.

Next, the CPU sets bit 0 of the read data to 1, changing the PDR3 data to H'41. Finally, the CPU writes this value (H'41) to PDR3, completing execution of BSET.

As a result of this operation, bit 0 in PDR3 becomes 1, and $P3_0$ outputs a high-level signal. However, bits 7 and 6 of PDR3 end up with different values.

To avoid this problem, store a copy of the PDR3 data in a work area in memory. Perform the bit manipulation on the data in the work area, then write this data to PDR3.

[A: Prior to executing BSET]

```
MOV. B  #80,  R0L
MOV. B  R0L,  @RAM0
MOV. B  R0L,  @PDR3
```

The PDR3 value (H'80) is written to a work area in memory (RAM0) as well as to PDR3.

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0
RAM0	1	0	0	0	0	0	0	0

[B: BSET instruction executed]

```
BSET    #0  ,  @RAM0
```

The BSET instruction is executed designating the PDR3 work area (RAM0).

[C: After executing BSET]

```
MOV. B  @RAM0, R0L
MOV. B  R0L,  @PDR3
```

The work area (RAM0) value is written to PDR3.

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	1
RAM0	1	0	0	0	0	0	0	1

2. Bit manipulation in a register containing a write-only bit

Example 3: BCLR instruction executed designating port 3 control register PCR3

As in the examples above, $P3_7$ and $P3_6$ are input pins, with a low-level signal input at $P3_7$ and a high-level signal at $P3_6$. The remaining pins, $P3_5$ to $P3_0$, are output pins that output low-level signals. In this example, the BCLR instruction is used to change pin $P3_0$ to an input port. It is assumed that a high-level signal will be input to this input pin.

[A: Prior to executing BCLR]

	$P3_7$	**$P3_6$**	**$P3_5$**	**$P3_4$**	**$P3_3$**	**$P3_2$**	**$P3_1$**	**$P3_0$**
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0

[B: BCLR instruction executed]

```
BCLR   #0  ,  @PCR3
```

The BCLR instruction is executed designating PCR3.

[C: After executing BCLR]

	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
Input/output	Output	Output	Output	Output	Output	Output	Output	Input
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	1	1	1	1	1	1	1	0
PDR3	1	0	0	0	0	0	0	0

[D: Explanation of how BCLR operates]

When the BCLR instruction is executed, first the CPU reads PCR3. Since PCR3 is a write-only register, the CPU reads a value of H'FF, even though the PCR3 value is actually H'3F.

Next, the CPU clears bit 0 in the read data to 0, changing the data to H'FE. Finally, this value (H'FE) is written to PCR3 and BCLR instruction execution ends.

As a result of this operation, bit 0 in PCR3 becomes 0, making $P3_0$ an input port. However, bits 7 and 6 in PCR3 change to 1, so that $P3_7$ and $P3_6$ change from input pins to output pins.

To avoid this problem, store a copy of the PCR3 data in a work area in memory. Perform the bit manipulation on the data in the work area, then write this data to PCR3.

[A: Prior to executing BCLR]

```
MOV. B  #3F,  R0L
MOV. B  R0L,  @RAM0
MOV. B  R0L,  @PCR3
```

The PCR3 value (H'3F) is written to a work area in memory (RAM0) as well as to PCR3.

	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	Low level
PCR3	0	0	1	1	1	1	1	1
PDR3	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	1

[B: BCLR instruction executed]

```
BCLR   #0  ,  @RAM0
```

The BCLR instruction is executed designating the PCR3 work area (RAM0).

[C: After executing BCLR]

```
MOV. B  @RAM0, R0L
MOV. B  R0L,  @PCR3
```

The work area (RAM0) value is written to PCR3.

	$P3_7$	$P3_6$	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$
Input/output	Input	Input	Output	Output	Output	Output	Output	Output
Pin state	Low level	High level	Low level	Low level	Low level	Low level	Low level	High level
PCR3	0	0	1	1	1	1	1	0
PDR3	1	0	0	0	0	0	0	0
RAM0	0	0	1	1	1	1	1	0

Table 2-12 lists the pairs of registers that share identical addresses. Table 2-13 lists the registers that contain write-only bits.

Table 2-12 Registers with Shared Addresses

Register Name		Abbreviation	Address
Output compare register AH and output compare register BH	(timer X)	OCRAH/OCRBH	H'F774
Output compare register AL and output compare register BL	(timer X)	OCRAL/OCRBL	H'F775
Timer counter B1 and timer load register B1	(timer B1)	TCB1/TLB1	H'FFB3
Port data register 1*		PDR1	H'FFD4
Port data register 2*		PDR2	H'FFD5
Port data register 3*		PDR3	H'FFD6
Port data register 5*		PDR5	H'FFD8
Port data register 6*		PDR6	H'FFD9
Port data register 7*		PDR7	H'FFDA
Port data register 8*		PDR8	H'FFDB
Port data register 9*		PDR9	H'FFDC

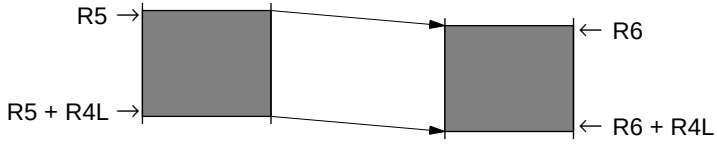
Note: * Port data registers have the same addresses as input pins.

Table 2-13 Registers with Write-Only Bits

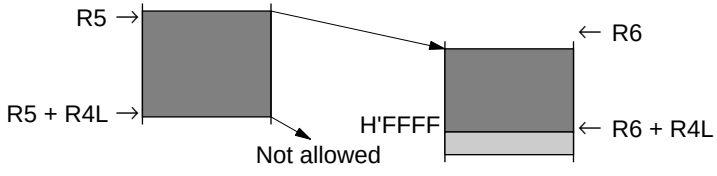
Register Name	Abbreviation	Address
Port control register 1	PCR1	H'FFE4
Port control register 2	PCR2	H'FFE5
Port control register 3	PCR3	H'FFE6
Port control register 5	PCR5	H'FFE8
Port control register 6	PCR6	H'FFE9
Port control register 7	PCR7	H'FFEA
Port control register 8	PCR8	H'FFEB
Port control register 9	PCR9	H'FFEC
PWM control register	PWCR	H'FFD0
PWM data register U	PWDRU	H'FFD1
PWM data register L	PWDRL	H'FFD2

2.9.3 Notes on Use of the EEPMOV Instruction

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



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



Section 3 Exception Handling

3.1 Overview

Exception handling is performed in the H8/3657 Series when a reset or interrupt occurs. Table 3-1 shows the priorities of these two types of exception handling.

Table 3-1 Exception Handling Types and Priorities

Priority	Exception Source	Time of Start of Exception Handling
High	Reset	Exception handling starts as soon as the reset state is cleared
↑	Interrupt	When an interrupt is requested, exception handling starts after execution of the present instruction or the exception handling in progress is completed
Low		

3.2 Reset

3.2.1 Overview

A reset is the highest-priority exception. The internal state of the CPU and the registers of the on-chip peripheral modules are initialized.

3.2.2 Reset Sequence

1. Reset by $\overline{\text{RES}}$ pin

As soon as the RES pin goes low, all processing is stopped and the chip enters the reset state.

To make sure the chip is reset properly, observe the following precautions.

- At power on: Hold the $\overline{\text{RES}}$ pin low until the clock pulse generator output stabilizes.
- At power on, when using an external clock: Hold the $\overline{\text{RES}}$ pin low for the ceramic oscillator oscillation stabilization time shown in table 13.7 in the Electrical Characteristics section.
- Resetting during operation: Hold the $\overline{\text{RES}}$ pin low for at least 18 system clock cycles.

Reset exception handling begins when the $\overline{\text{RES}}$ pin is held low for a given period, then returned to the high level.

Reset exception handling takes place as follows.

- The CPU internal state and the registers of on-chip peripheral modules are initialized, with the I bit of the condition code register (CCR) set to 1.

- The PC is loaded from the reset exception handling vector address (H'0000 to H'0001), after which the program starts executing from the address indicated in PC.

When system power is turned on or off, the $\overline{RES}$ pin should be held low.

Figure 3-1 shows the reset sequence starting from $\overline{RES}$ input.

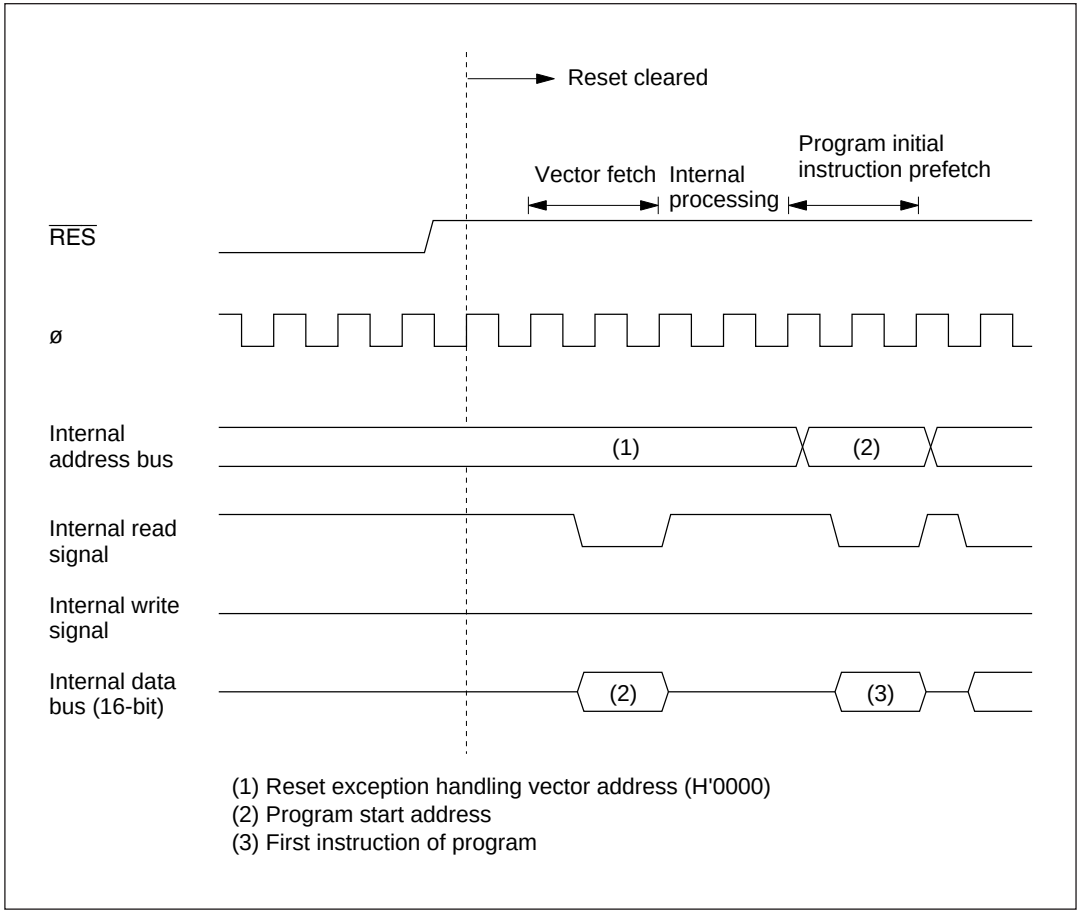


Figure 3-1 Reset Sequence

2. Reset by watchdog timer

The watchdog timer counter (TCW) starts counting up when the WDON bit is set to 1 in the watchdog timer control/status register (TCSRW). If TCW overflows, the WRST bit is set to 1 in TCSRW and the chip enters the reset state. While the WRST bit is set to 1 in TCSRW, when TCW overflows the reset state is cleared and reset exception handling begins. The same reset exception handling is carried out as for input at the $\overline{RES}$ pin. For details on the watchdog timer, see 9.11, Watchdog Timer.

3.2.3 Interrupt Immediately after Reset

After a reset, if an interrupt were to be accepted before the stack pointer (SP: R7) was initialized, PC and CCR would not be pushed onto the stack correctly, resulting in program runaway. To prevent this, immediately after reset exception handling all interrupts are masked. For this reason, the initial program instruction is always executed immediately after a reset. This instruction should initialize the stack pointer (e.g. MOV.W #xx: 16, SP).

3.3 Interrupts

3.3.1 Overview

The interrupt sources include 12 external interrupts (IRQ_3 to IRQ_0, INT_7 to INT_0) and 21 internal interrupts from on-chip peripheral modules. Table 3-2 shows the interrupt sources, their priorities, and their vector addresses. When more than one interrupt is requested, the interrupt with the highest priority is processed.

The interrupts have the following features:

- Internal and external interrupts can be masked by the I bit in CCR. When the I bit is set to 1, interrupt request flags can be set but the interrupts are not accepted.
- IRQ_3 to IRQ_0 and INT_7 to INT_0 can be set independently to either rising edge sensing or falling edge sensing.

Table 3-2 Interrupt Sources and Their Priorities

Interrupt Source	Interrupt	Vector Number	Vector Address	Priority
$\overline{RES}$	Reset	0	H'0000 to H'0001	High
$\overline{IRQ_0}$	IRQ_0	4	H'0008 to H'0009	
$\overline{IRQ_1}$	IRQ_1	5	H'000A to H'000B	
$\overline{IRQ_2}$	IRQ_2	6	H'000C to H'000D	
$\overline{IRQ_3}$	IRQ_3	7	H'000E to H'000F	
$\overline{INT_0}$	INT_0	8	H'0010 to H'0011	
$\overline{INT_1}$	INT_1			
$\overline{INT_2}$	INT_2			
$\overline{INT_3}$	INT_3			
$\overline{INT_4}$	INT_4			
$\overline{INT_5}$	INT_5			
$\overline{INT_6}$	INT_6			
$\overline{INT_7}$	INT_7			
Timer A	Timer A overflow	10	H'0014 to H'0015	
Timer B1	Timer B1 overflow	11	H'0016 to H'0017	
Timer X	Timer X input capture A	16	H'0020 to H'0021	
	Timer X input capture B			
	Timer X input capture C			
	Timer X input capture D			
	Timer X compare match A			
	Timer X compare match B			
	Timer X overflow			
Timer V	Timer V compare match A	17	H'0022 to H'0023	
	Timer V compare match B			
	Timer V overflow			
SCI1	SCI1 transfer complete	19	H'0026 to H'0027	
SCI3	SCI3 transmit end	21	H'002A to H'002B	
	SCI3 transmit data empty			
	SCI3 receive data full			
	SCI3 overflow error			
	SCI3 framing error			
	SCI3 parity error			
A/D	A/D conversion end	22	H'002C to H'002D	
(SLEEP instruction executed)	Direct transfer	23	H'002E to H'002F	Low

Note: * Vector addresses H'0002 to H'0005, H'0024 to H'0025, H'0028 to H'0029 are reserved and cannot be used.

3.3.2 Interrupt Control Registers

Table 3-3 lists the registers that control interrupts.

Table 3-3 Interrupt Control Registers

Name	Abbreviation	R/W	Initial Value	Address
Interrupt edge select register 1	IEGR1	R/W	H'70	H'FFF2
Interrupt edge select register 2	IEGR2	R/W	H'00	H'FFF3
Interrupt enable register 1	IENR1	R/W	H'10	H'FFF4
Interrupt enable register 2	IENR2	R/W	H'00	H'FFF5
Interrupt enable register 3	IENR3	R/W	H'00	H'FFF6
Interrupt request register 1	IRR1	R/W*	H'10	H'FFF7
Interrupt request register 2	IRR2	R/W*	H'00	H'FFF8
Interrupt request register 3	IRR3	R/W*	H'00	H'FFF9

Note: * Write is enabled only for writing of 0 to clear a flag.

1. Interrupt edge select register 1 (IEGR1)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	IEG3	IEG2	IEG1	IEG0
Initial value	0	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

IEGR1 is an 8-bit read/write register used to designate whether pins $\overline{IRQ_3}$ to $\overline{IRQ_0}$ are set to rising edge sensing or falling edge sensing. Upon reset, IEGR1 is initialized to H'70.

Bit 7: Reserved bit

Bit 7 is reserved: it is always read as 0 and cannot be modified.

Bits 6 to 4: Reserved bits

Bits 6 to 4 are reserved; they are always read as 1, and cannot be modified.

Bit 3: IRQ_3 edge select (IEG3)

Bit 3 selects the input sensing of pin $\overline{IRQ_3}$.

Bit 3 IEG3	Description	
0	Falling edge of $\overline{IRQ_3}$ pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ_3}$ pin input is detected	

Bit 2: IRQ_2 edge select (IEG2)

Bit 2 selects the input sensing of pin $\overline{IRQ_2}$.

Bit 2 IEG2	Description	
0	Falling edge of $\overline{IRQ_2}$ pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ_2}$ pin input is detected	

Bit 1: IRQ_1 edge select (IEG1)

Bit 1 selects the input sensing of pin $\overline{IRQ_1}$.

Bit 1 IEG1	Description	
0	Falling edge of $\overline{IRQ_1}$ pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ_1}$ pin input is detected	

Bit 0: IRQ_0 edge select (IEG0)

Bit 0 selects the input sensing of pin $\overline{IRQ_0}$.

Bit 0 IEG0	Description	
0	Falling edge of $\overline{IRQ_0}$ pin input is detected	(initial value)
1	Rising edge of $\overline{IRQ_0}$ pin input is detected	

2. Interrupt edge select register 2 (IEGR2)

Bit	7	6	5	4	3	2	1	0
	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2	INTEG1	INTEG0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IEGR2 is an 8-bit read/write register, used to designate whether pins $\overline{INT_7}$ to $\overline{INT_0}$, TMIY, and TMIB are set to rising edge sensing or falling edge sensing. Upon reset, IEGR2 is initialized to H'00.

Bit 7: INT_7 edge select (INTEG7)

Bit 7 selects the input sensing of the $\overline{INT_7}$ pin and TMIY pin.

Bit 7 INTEG7	Description	
0	Falling edge of $\overline{INT_7}$ and TMIY pin input is detected	(initial value)
1	Rising edge of $\overline{INT_7}$ and TMIY pin input is detected	

Bit 6: INT_6 edge select (INTEG6)

Bit 6 selects the input sensing of the $\overline{INT_6}$ pin and TMIB pin.

Bit 6 INTEG6	Description	
0	Falling edge of $\overline{INT_6}$ and TMIB pin input is detected	(initial value)
1	Rising edge of $\overline{INT_6}$ and TMIB pin input is detected	

Bit 5: INT_5 edge select (INTEG5)

Bit 5 selects the input sensing of the $\overline{INT_5}$ pin and $\overline{ADTRG}$ pin.

Bit 5 INTEG5	Description	
0	Falling edge of $\overline{INT_5}$ and $\overline{ADTRG}$ pin input is detected	(initial value)
1	Rising edge of $\overline{INT_5}$ and $\overline{ADTRG}$ pin input is detected	

Bits 4 to 0: INT_4 to INT_0 edge select (INTEG4 to INTEG0)

Bits 4 to 0 select the input sensing of pins $\overline{INT_4}$ to $\overline{INT_0}$.

Bit n INTEGn	Description	
0	Falling edge of $\overline{INT_n}$ pin input is detected	(initial value)
1	Rising edge of $\overline{INT_n}$ pin input is detected	

(n = 4 to 0)

3. Interrupt enable register 1 (IENR1)

Bit	7	6	5	4	3	2	1	0
	IENTB1	IENTA	—	—	IEN3	IEN2	IEN1	IEN0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

IENR1 is an 8-bit read/write register that enables or disables interrupt requests. Upon reset, IENR1 is initialized to H'10.

Bit 7: Timer B1 interrupt enable (IENTB1)

Bit 7 enables or disables timer B1 overflow interrupt requests.

Bit 7 IENTB1	Description	
0	Disables timer B1 interrupt requests	(initial value)
1	Enables timer B1 interrupt requests	

Bit 6: Timer A interrupt enable (IENTA)

Bit 6 enables or disables timer A overflow interrupt requests.

Bit 6 IENTA	Description	
0	Disables timer A interrupt requests	(initial value)
1	Enables timer A interrupt requests	

Bit 5: Reserved bit

Bit 5 is reserved: it is always read as 0 and cannot be modified.

Bit 4: Reserved bit

Bit 4 is reserved; it is always read as 1, and cannot be modified.

Bits 3 to 0: IRQ_3 to IRQ_0 interrupt enable (IEN3 to IEN0)

Bits 3 to 0 enable or disable IRQ_3 to IRQ_0 interrupt requests.

Bit n IENn	Description	
0	Disables interrupt requests from pin $\overline{IRQ_n}$	(initial value)
1	Enables interrupt requests from pin $\overline{IRQ_n}$	

(n = 3 to 0)

4. Interrupt enable register 2 (IENR2)

Bit	7	6	5	4	3	2	1	0
	IENDT	IENAD	—	IENS1	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	—	R/W	—	—	—	—

IENR2 is an 8-bit read/write register that enables or disables interrupt requests. Upon reset, IENR2 is initialized to H'00.

Bit 7: Direct transfer interrupt enable (IENDT)

Bit 7 enables or disables direct transfer interrupt requests.

Bit 7 IENDT	Description	
0	Disables direct transfer interrupt requests	(initial value)
1	Enables direct transfer interrupt requests	

Bit 6: A/D converter interrupt enable (IENAD)

Bit 6 enables or disables A/D converter interrupt requests.

Bit 6 IENAD	Description	
0	Disables A/D converter interrupt requests	(initial value)
1	Enables A/D converter interrupt requests	

Bit 5: Reserved bit

Bit 5 is reserved: it is always read as 0 and cannot be modified.

Bit 4: SCI1 interrupt enable (IENS1)

Bit 4 enables or disables SCI1 transfer complete interrupt requests.

Bit 4 IENS1	Description	
0	Disables SCI1 interrupt requests	(initial value)
1	Enables SCI1 interrupt requests	

Bits 3 to 0: Reserved bits

Bits 3 to 0 are reserved: they are always read as 0 and cannot be modified.

5. Interrupt enable register 3 (IENR3)

Bit	7	6	5	4	3	2	1	0
	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IENR3 is an 8-bit read/write register that enables or disables INT_7 to INT_0 interrupt requests. Upon reset, IENR3 is initialized to H'00.

Bits 7 to 0: INT_7 to INT_0 interrupt enable (INTEN7 to INTEN0)

Bits 7 to 0 enable or disable INT_7 to INT_0 interrupt requests.

Bit n **INTENn**	**Description**	
0	Disables interrupt requests from pin $\overline{INT_n}$	(initial value)
1	Enables interrupt requests from pin $\overline{INT_n}$	

(n = 7 to 0)

6. Interrupt request register 1 (IRR1)

Bit	7	6	5	4	3	2	1	0
	IRRTB1	IRRTA	—	—	IRRI3	IRRI2	IRRI1	IRRI0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W*	R/W*	—	—	R/W*	R/W*	R/W*	R/W*

Note: * Only a write of 0 for flag clearing is possible

IRR1 is an 8-bit read/write register, in which a corresponding flag is set to 1 when a timer B1, timer A, timer Y, or IRQ_3 to IRQ_0 interrupt is requested. The flags are not cleared automatically when an interrupt is accepted. It is necessary to write 0 to clear each flag. Upon reset, IRR1 is initialized to H'10.

Bit 7: Timer B1 interrupt request flag (IRRTB1)

Bit 7 **IRRTB1**	**Description**	
0	Clearing conditions: When IRRTB1 = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When the timer B1 counter value overflows from H'FF to H'00	

Bit 6: Timer A interrupt request flag (IRRTA)

Bit 6 IRRTA	Description	
0	Clearing conditions: When IRRTA = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When the timer A counter value overflows from H'FF to H'00	

Bit 5: Reserved bit

Bit 5 is reserved: it is always read as 0 and cannot be modified.

Bit 4: Reserved bit

Bit 4 is reserved; it is always read as 1, and cannot be modified.

Bits 3 to 0: IRQ_3 to IRQ_0 interrupt request flags (IRRI3 to IRRI0)

Bit n IRRIn	Description	
0	Clearing conditions: When IRRIn = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When pin $\overline{IRQ_n}$ is designated for interrupt input and the designated signal edge is input	

(n = 3 to 0)

7. Interrupt request register 2 (IRR2)

Bit	7	6	5	4	3	2	1	0
	IRRDT	IRRAD	—	IRRS1	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W*	R/W*	—	R/W*	—	—	—	—

Note: * Only a write of 0 for flag clearing is possible

IRR2 is an 8-bit read/write register, in which a corresponding flag is set to 1 when a direct transfer, A/D converter, or SCI1 interrupt is requested. The flags are not cleared automatically when an interrupt is accepted. It is necessary to write 0 to clear each flag. Upon reset, IRR2 is initialized to H'00.

Bit 7: Direct transfer interrupt request flag (IRRDT)

Bit 7 IRRDT	Description	
0	Clearing conditions: When IRRDT = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When a direct transfer is made by executing a SLEEP instruction while DTON = 1 in SYSCR2	

Bit 6: A/D converter interrupt request flag (IRRAD)

Bit 6 IRRAD	Description	
0	Clearing conditions: When IRRAD = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When A/D conversion is completed and ADSF is cleared to 0 in ADSR	

Bit 5: Reserved bit

Bit 5 is reserved: it is always read as 0 and cannot be modified.

Bit 4: SCI1 interrupt request flag (IRRS1)

Bit 4 IRRS1	Description	
0	Clearing conditions: When IRRS1 = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When an SCI1 transfer is completed	

Bits 3 to 0: Reserved bits

Bits 3 to 0 are reserved: they are always read as 0 and cannot be modified.

8. Interrupt request register 3 (IRR3)

Bit	7	6	5	4	3	2	1	0
	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0
Initial value	0	0	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 a write of 0 for flag clearing is possible

IRR3 is an 8-bit read/write register, in which a corresponding flag is set to 1 by a transition at pin $\overline{INT_7}$ to $\overline{INT_0}$. The flags are not cleared automatically when an interrupt is accepted. It is necessary to write 0 to clear each flag. Upon reset, IRR3 is initialized to H'00.

Bits 7 to 0: INT_7 to INT_0 interrupt request flags (INTF7 to INTF0)

Bit n INTFn	Description	
0	Clearing conditions: When INTFn = 1, it is cleared by writing 0	(initial value)
1	Setting conditions: When the designated signal edge is input at pin $\overline{INT_n}$	

(n = 7 to 0)

3.3.3 External Interrupts

There are 12 external interrupts: IRQ_3 to IRQ_0 and INT_7 to INT_0.

1. Interrupts IRQ_3 to IRQ_0

Interrupts IRQ_3 to IRQ_0 are requested by input signals to pins $\overline{IRQ_3}$ to $\overline{IRQ_0}$. These interrupts are detected by either rising edge sensing or falling edge sensing, depending on the settings of bits IEG3 to IEG0 in IEGR1.

When these pins are designated as pins $\overline{IRQ_3}$ to $\overline{IRQ_0}$ in port mode register 1 and the designated edge is input, the corresponding bit in IRR1 is set to 1, requesting an interrupt. Recognition of these interrupt requests can be disabled individually by clearing bits IEN3 to IEN0 to 0 in IENR1. These interrupts can all be masked by setting the I bit to 1 in CCR.

When IRQ_3 to IRQ_0 interrupt exception handling is initiated, the I bit is set to 1 in CCR. Vector numbers 7 to 4 are assigned to interrupts IRQ_3 to IRQ_0. The order of priority is from IRQ_0 (high) to IRQ_3 (low). Table 3-2 gives details.

2. INT interrupts

INT interrupts are requested by input signals to pins $\overline{INT_7}$ to $\overline{INT_0}$. These interrupts are detected by either rising edge sensing or falling edge sensing, depending on the settings of bits INTEG7 to INTEG0 in IEGR2.

When the designated edge is input at pins $\overline{INT_7}$ to $\overline{INT_0}$, the corresponding bit in IRR1 is set to 1, requesting an interrupt. Recognition of these interrupt requests can be disabled individually by clearing bits INTEN7 to INTEN0 to 0 in IENR3. These interrupts can all be masked by setting the I bit to 1 in CCR.

When INT interrupt exception handling is initiated, the I bit is set to 1 in CCR. Vector number 8 is assigned to the INT interrupts. All eight interrupts have the same vector number, so the interrupt-handling routine must discriminate the interrupt source.

Note: Pins $\overline{INT_7}$ to $\overline{INT_0}$ are multiplexed with port 5. Even in port usage of these pins, whenever the designated edge is input or output, the corresponding bit INTFn is set to 1.

3.3.4 Internal Interrupts

There are 21 internal interrupts that can be requested by the on-chip peripheral modules. When a peripheral module requests an interrupt, the corresponding bit in IRR1 or IRR2 is set to 1. Recognition of individual interrupt requests can be disabled by clearing the corresponding bit in IENR1 or IENR2. All these interrupts can be masked by setting the I bit to 1 in CCR. When internal interrupt handling is initiated, the I bit is set to 1 in CCR. Vector numbers from 23 to 9 are assigned to these interrupts. Table 3-2 shows the order of priority of interrupts from on-chip peripheral modules.

3.3.5 Interrupt Operations

Interrupts are controlled by an interrupt controller. Figure 3-2 shows a block diagram of the interrupt controller. Figure 3-3 shows the flow up to interrupt acceptance.

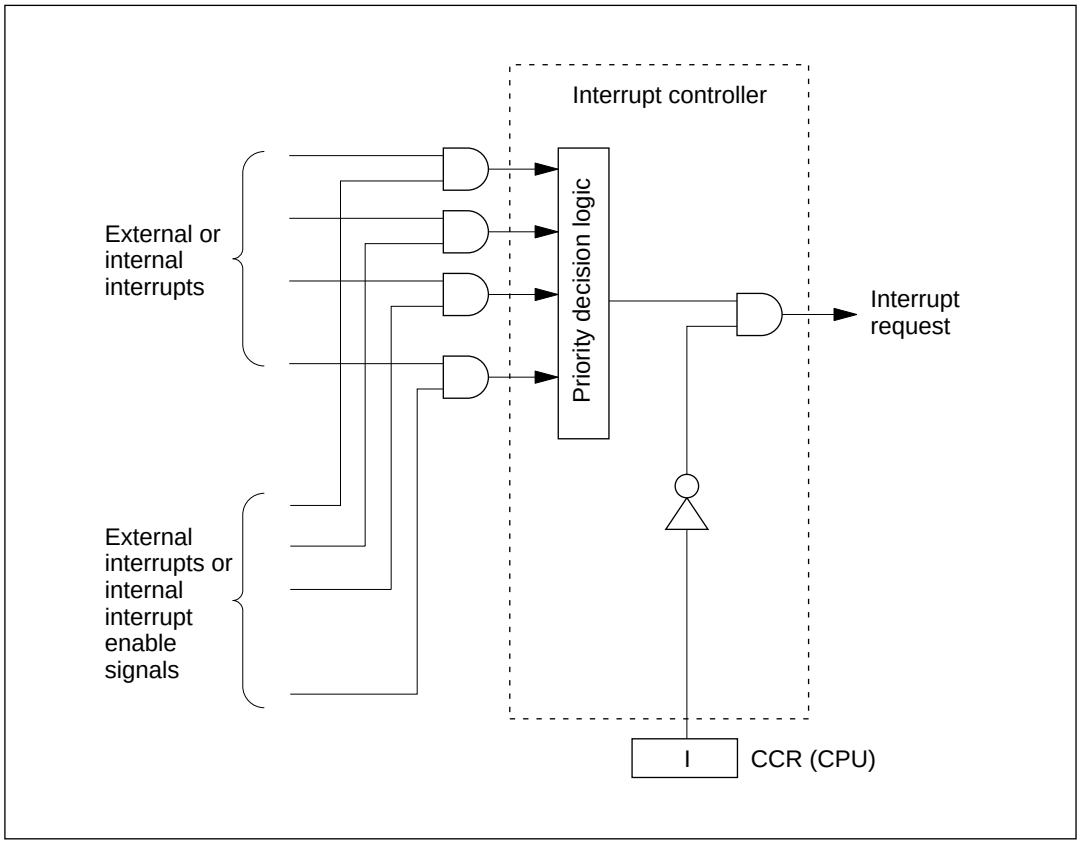


Figure 3-2 Block Diagram of Interrupt Controller

Interrupt operation is described as follows.

- When an interrupt condition is met while the interrupt enable register bit is set to 1, an interrupt request signal is sent to the interrupt controller.
- When the interrupt controller receives an interrupt request, it sets the interrupt request flag.
- From among the interrupts with interrupt request flags set to 1, the interrupt controller selects the interrupt request with the highest priority and holds the others pending. (Refer to table 3-2 for a list of interrupt priorities.)
- The interrupt controller checks the I bit of CCR. If the I bit is 0, the selected interrupt request is accepted; if the I bit is 1, the interrupt request is held pending.

- If the interrupt is accepted, after processing of the current instruction is completed, both PC and CCR are pushed onto the stack. The state of the stack at this time is shown in figure 3-4. The PC value pushed onto the stack is the address of the first instruction to be executed upon return from interrupt handling.

- The I bit of CCR is set to 1, masking further interrupts.

- The vector address corresponding to the accepted interrupt is generated, and the interrupt handling routine located at the address indicated by the contents of the vector address is executed.

Notes:

1. When disabling interrupts by clearing bits in an interrupt enable register, or when clearing bits in an interrupt request register, always do so while interrupts are masked (I = 1).

2. If the above clear operations are performed while I = 0, and as a result a conflict arises between the clear instruction and an interrupt request, exception processing for the interrupt will be executed after the clear instruction has been executed.

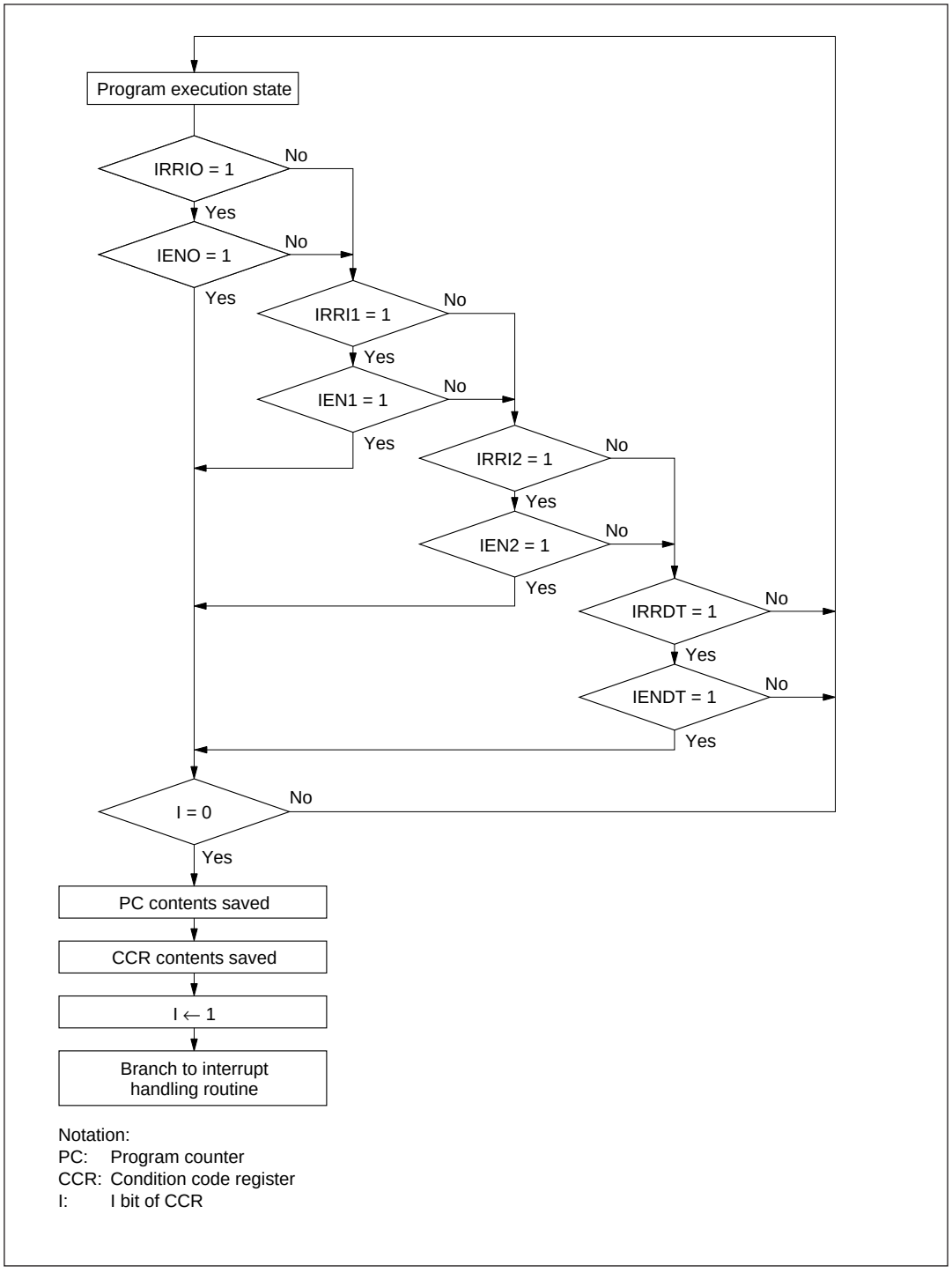


Figure 3-3 Flow up to Interrupt Acceptance

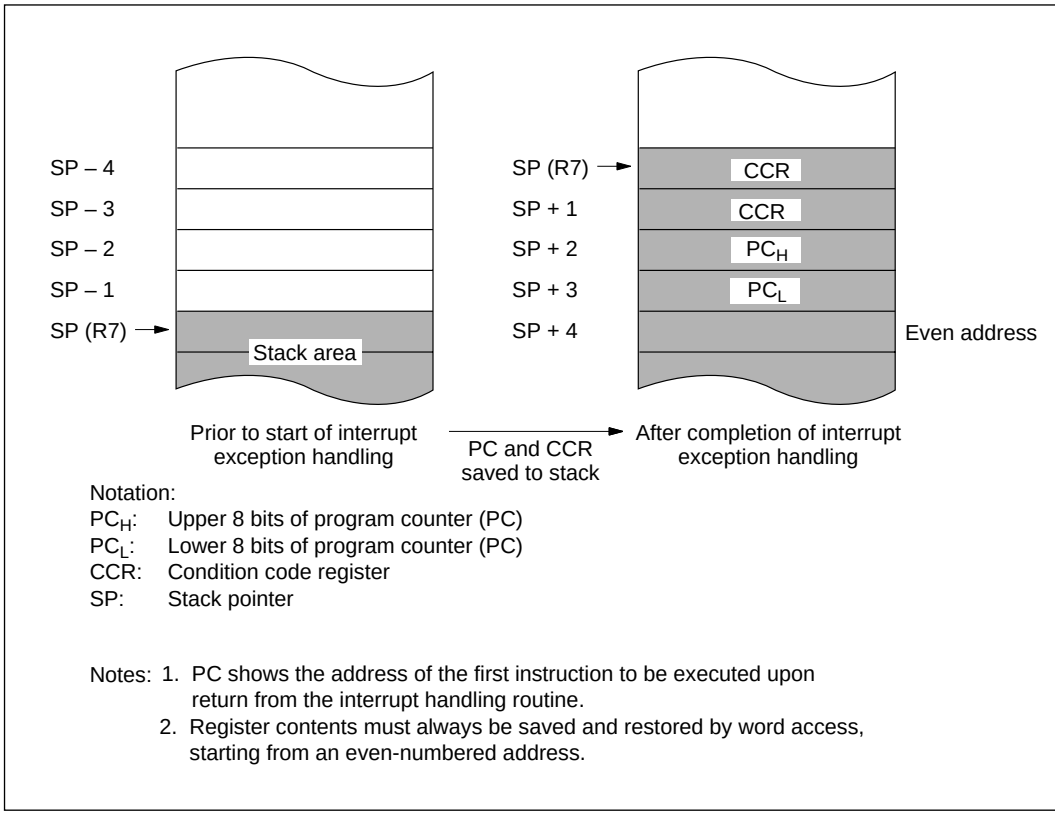


Figure 3-4 Stack State after Completion of Interrupt Exception Handling

Figure 3-5 shows a typical interrupt sequence where the program area is in the on-chip ROM and the stack area is in the on-chip RAM.

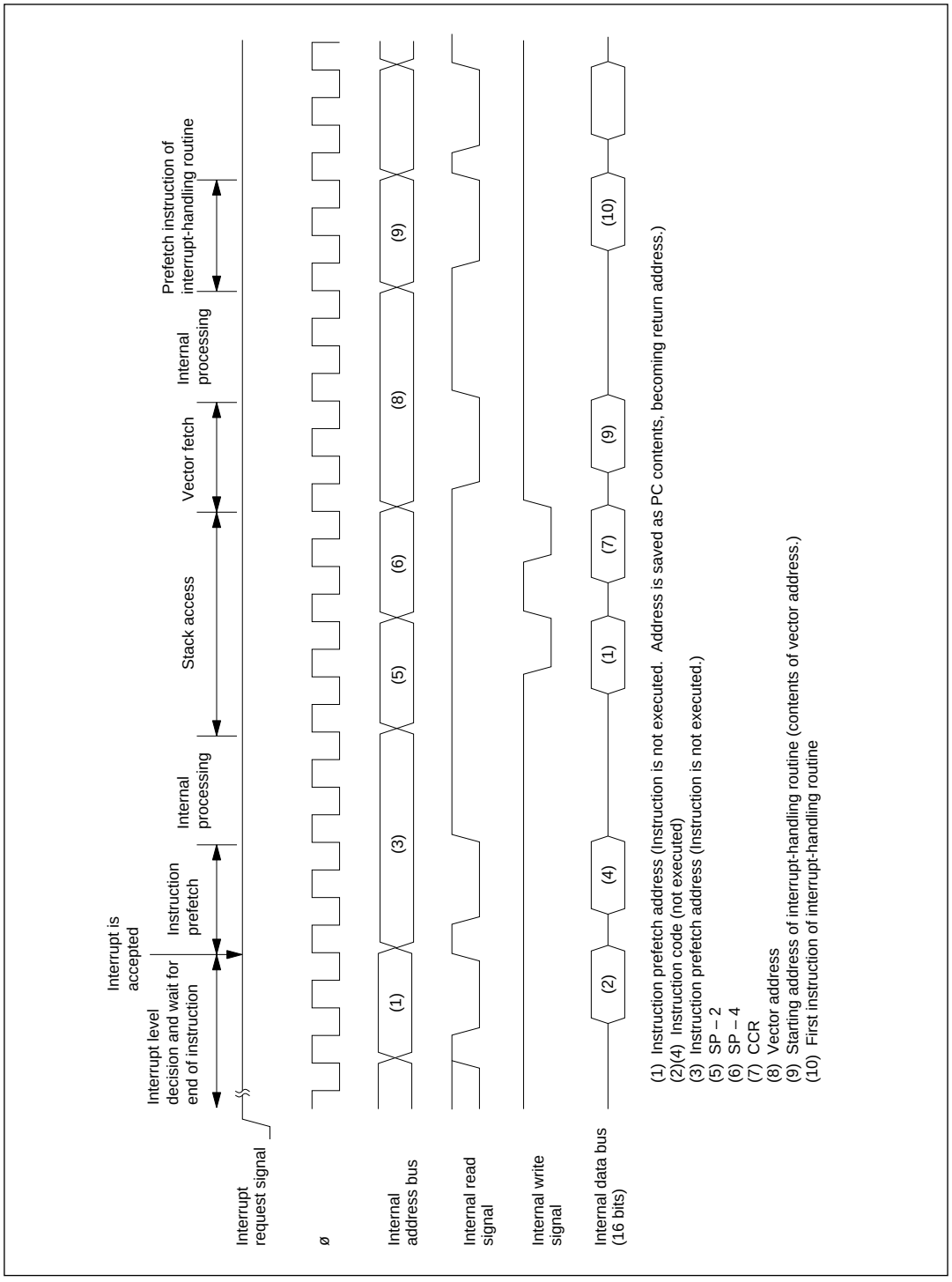


Figure 3-5 Interrupt Sequence

3.3.6 Interrupt Response Time

Table 3-4 shows the number of wait states after an interrupt request flag is set until the first instruction of the interrupt handler is executed.

Table 3-4 Interrupt Wait States

Item	States
Waiting time for completion of executing instruction*	1 to 13
Saving of PC and CCR to stack	4
Vector fetch	2
Instruction fetch	4
Internal processing	4
Total	15 to 27

Note: * Not including EEPMOV instruction.

3.4 Application Notes

3.4.1 Notes on Stack Area Use

When word data is accessed in the H8/3657 Series, the least significant bit of the address is regarded as 0. Access to the stack always takes place in word size, so the stack pointer (SP: R7) should never indicate an odd address. Use PUSH Rn (MOV.W Rn, @–SP) or POP Rn (MOV.W @SP+, Rn) to save or restore register values.

Setting an odd address in SP may cause a program to crash. An example is shown in figure 3-6.

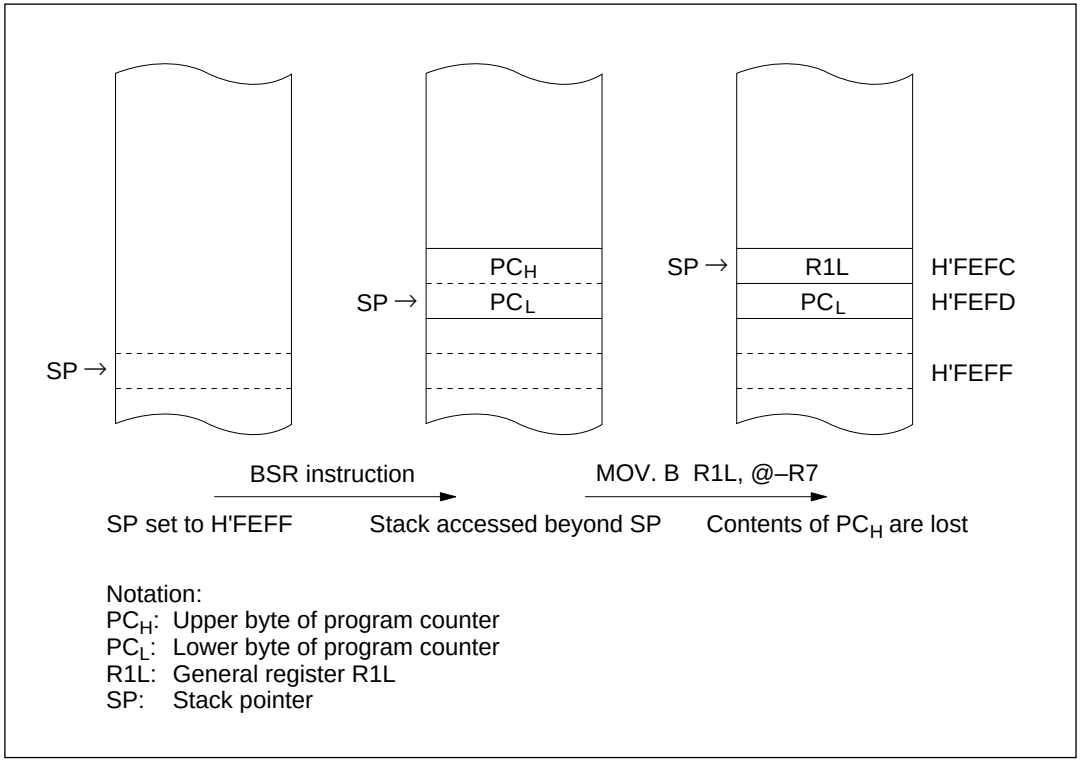


Figure 3-6 Operation when Odd Address is Set in SP

When CCR contents are saved to the stack during interrupt exception handling or restored when RTE is executed, this also takes place in word size. Both the upper and lower bytes of word data are saved to the stack; on return, the even address contents are restored to CCR while the odd address contents are ignored.

3.4.2 Notes on Rewriting Port Mode Registers

When a port mode register is rewritten to switch the functions of external interrupt pins, the following points should be observed.

When an external interrupt pin function is switched by rewriting the port mode register that controls pins $\overline{IRQ_3}$ to $\overline{IRQ_1}$, the interrupt request flag may be set to 1 at the time the pin function is switched, even if no valid interrupt is input at the pin. Table 3-5 shows the conditions under which interrupt request flags are set to 1 in this way.

Table 3-5 Conditions under which Interrupt Request Flag is Set to 1

Interrupt Request Flags Set to 1		Conditions
IRR1	IRRI3	When PMR1 bit IRQ3 is changed from 0 to 1 while pin $\underline{IRQ_3}$ is low and IEGR bit IEG3 = 0.
		When PMR1 bit IRQ3 is changed from 1 to 0 while pin $\underline{IRQ_3}$ is low and IEGR bit IEG3 = 1.
	IRRI2	When PMR1 bit IRQ2 is changed from 0 to 1 while pin $\underline{IRQ_2}$ is low and IEGR bit IEG2 = 0.
		When PMR1 bit IRQ2 is changed from 1 to 0 while pin $\underline{IRQ_2}$ is low and IEGR bit IEG2 = 1.
	IRRI1	When PMR1 bit IRQ1 is changed from 0 to 1 while pin $\underline{IRQ_1}$ is low and IEGR bit IEG1 = 0.
		When PMR1 bit IRQ1 is changed from 1 to 0 while pin $\underline{IRQ_1}$ is low and IEGR bit IEG1 = 1.

Figure 3-7 shows the procedure for setting a bit in a port mode register and clearing the interrupt request flag.

When switching a pin function, mask the interrupt before setting the bit in the port mode register. After accessing the port mode register, execute at least one instruction (e.g., NOP), then clear the interrupt request flag from 1 to 0. Be sure to clear the flag is executed immediately after the port mode register access without executing an intervening instruction, the flag will not be cleared.

An alternative method is to avoid the setting of interrupt request flags when pin functions are switched by keeping the pins at the high level so that the conditions in table 3-5 do not occur.

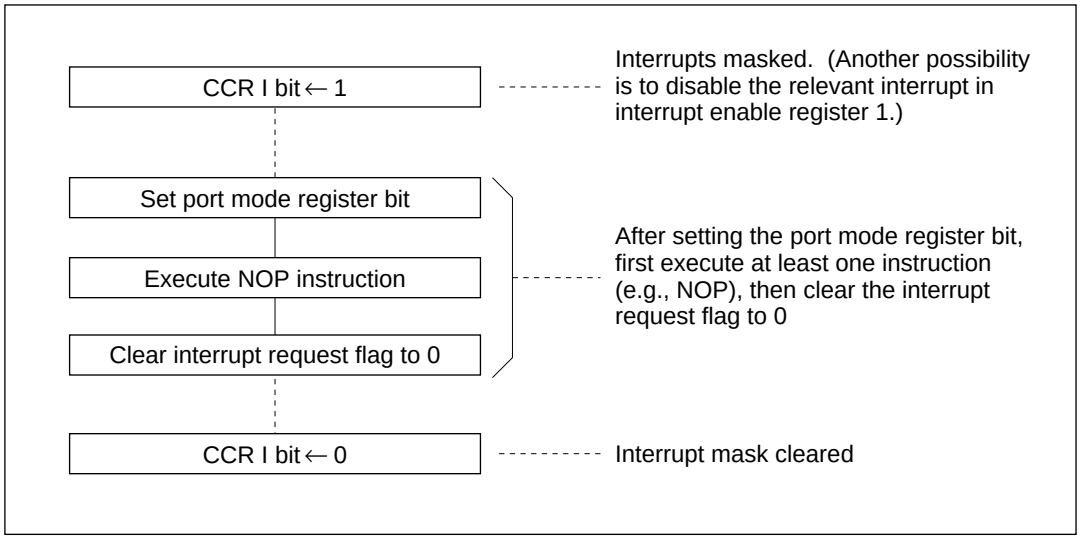


Figure 3-7 Port Mode Register Setting and Interrupt Request Flag Clearing Procedure

Section 4 Clock Pulse Generators

4.1 Overview

Clock oscillator circuitry (CPG: clock pulse generator) is provided on-chip, including both a system clock pulse generator and a subclock pulse generator. The system clock pulse generator consists of a system clock oscillator and system clock dividers. The subclock pulse generator consists of a subclock oscillator circuit and a subclock divider.

4.1.1 Block Diagram

Figure 4-1 shows a block diagram of the clock pulse generators.

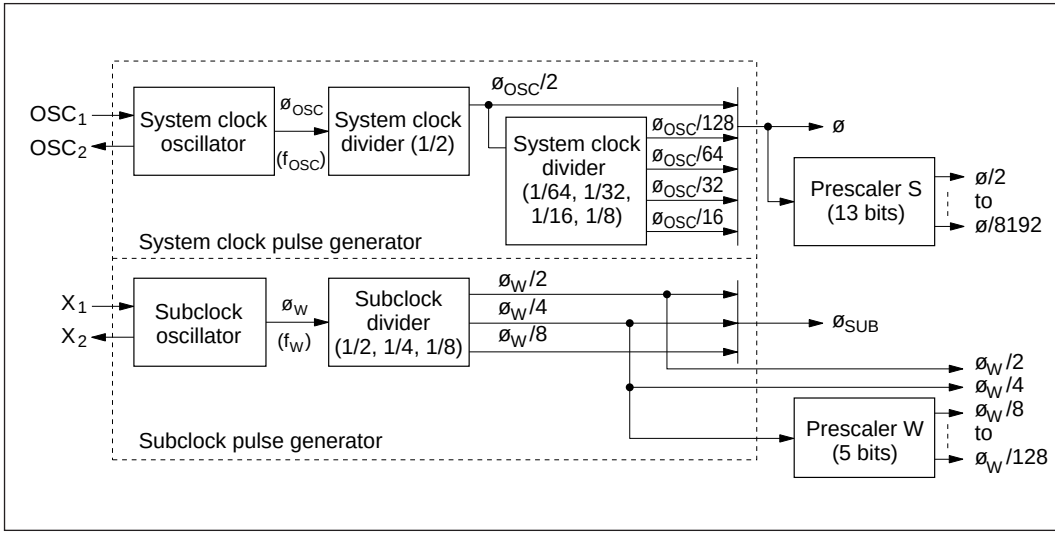


Figure 4-1 Block Diagram of Clock Pulse Generators

4.1.2 System Clock and Subclock

The basic clock signals that drive the CPU and on-chip peripheral modules are ø and $ø_{SUB}$. Four of the clock signals have names: ø is the system clock, $ø_{SUB}$ is the subclock, $ø_{OSC}$ is the oscillator clock, and $ø_W$ is the watch clock.

The clock signals available for use by peripheral modules are ø/2, ø/4, ø/8, ø/16, ø/32, ø/64, ø/128, ø/256, ø/512, ø/1024, ø/2048, ø/4096, ø/8192, $ø_W/2$, $ø_W/4$, $ø_W/8$, $ø_W/16$, $ø_W/32$, $ø_W/64$, and $ø_W/128$. The clock requirements differ from one module to another.

4.2 System Clock Generator

Clock pulses can be supplied to the system clock divider either by connecting a crystal or ceramic oscillator, or by providing external clock input.

1. Connecting a crystal oscillator

Figure 4-2 shows a typical method of connecting a crystal oscillator.

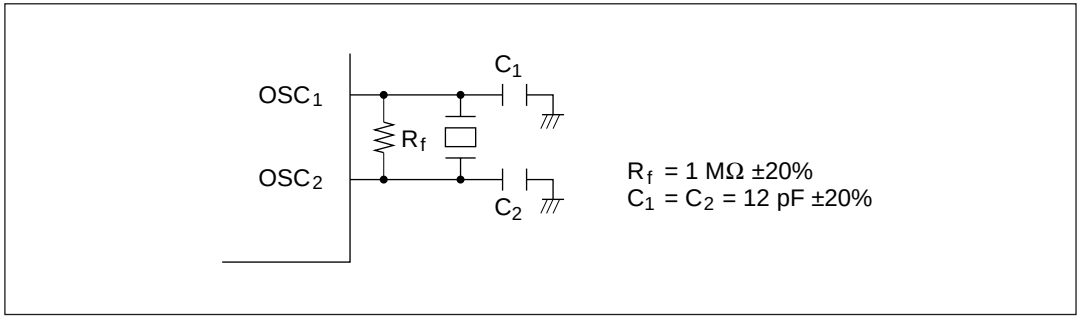


Figure 4-2 Typical Connection to Crystal Oscillator

Figure 4-3 shows the equivalent circuit of a crystal oscillator. An oscillator having the characteristics given in table 4-1 should be used.

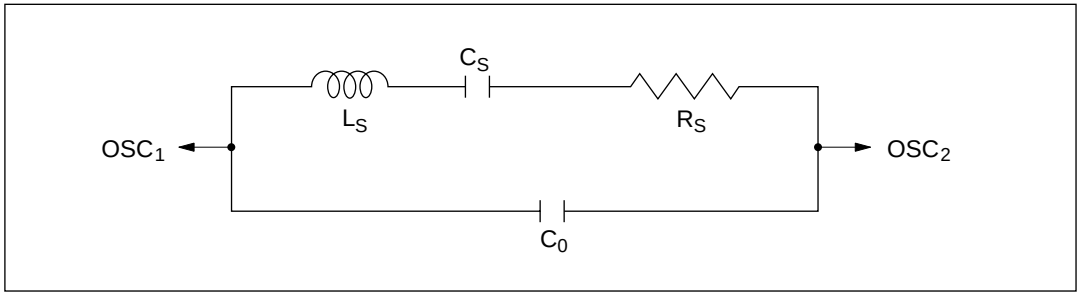


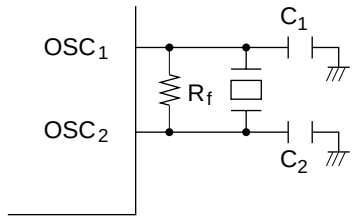
Figure 4-3 Equivalent Circuit of Crystal Oscillator

Table 4-1 Crystal Oscillator Parameters

Frequency (MHz)	2	4	8	10
R_S max (Ω)	500	100	50	30
C_0 (pF)	7 pF max			

2. Connecting a ceramic oscillator

Figure 4-4 shows a typical method of connecting a ceramic oscillator.



R_f = 1 MΩ ±20%
C_1 = 30 pF ±10%
C_2 = 30 pF ±10%
Ceramic oscillator: Murata

Figure 4-4 Typical Connection to Ceramic Oscillator

3. Notes on board design

When generating clock pulses by connecting a crystal or ceramic oscillator, pay careful attention to the following points.

Avoid running signal lines close to the oscillator circuit, since the oscillator may be adversely affected by induction currents. (See figure 4-5.)

The board should be designed so that the oscillator and load capacitors are located as close as possible to pins OSC_1 and OSC_2.

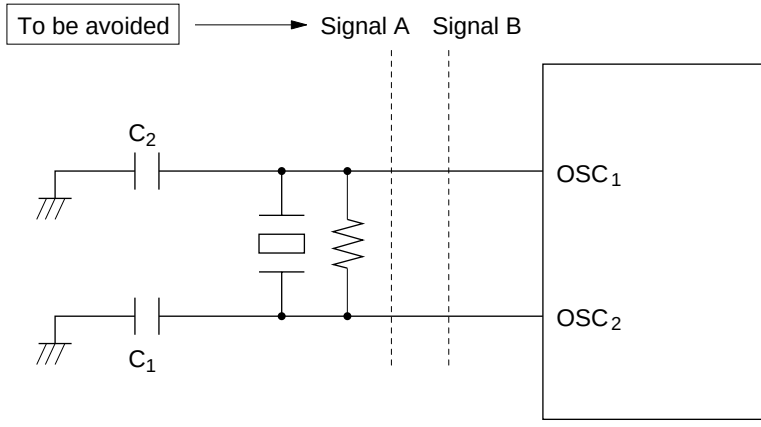


Figure 4-5 Board Design of Oscillator Circuit

4. External clock input method

Connect an external clock signal to pin OSC_1, and leave pin OSC_2 open. Figure 4-6 shows a typical connection.

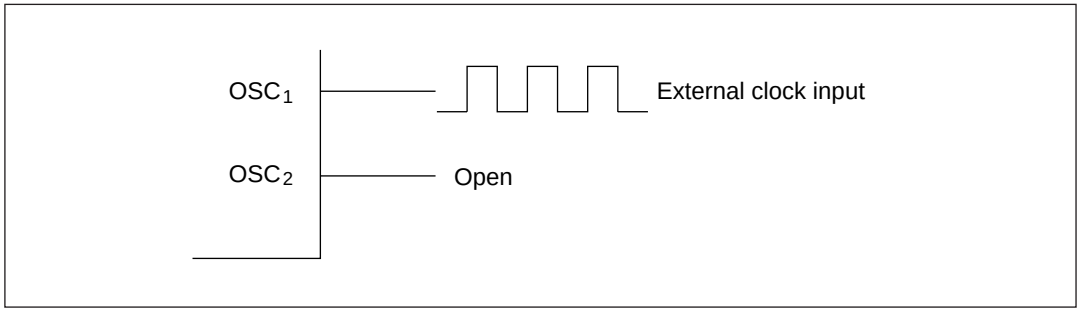


Figure 4-6 External Clock Input (Example)

Frequency	Oscillator Clock ($ø_{OSC}$)
Duty cycle	45% to 55%

4.3 Subclock Generator

1. Connecting a 32.768-kHz crystal oscillator

Clock pulses can be supplied to the subclock divider by connecting a 32.768-kHz crystal oscillator, as shown in figure 4-7. Follow the same precautions as noted under 4.2.3, Note on board design, for the system clock.

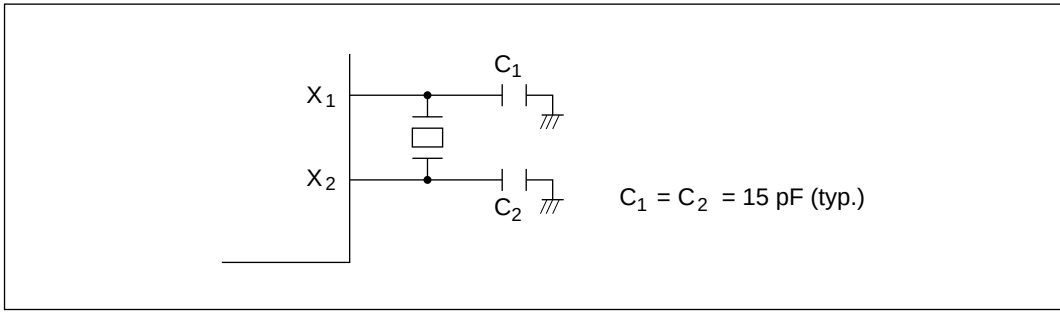


Figure 4-7 Typical Connection to 32.768-kHz Crystal Oscillator

Figure 4-8 shows the equivalent circuit of the 32.768-kHz crystal oscillator.

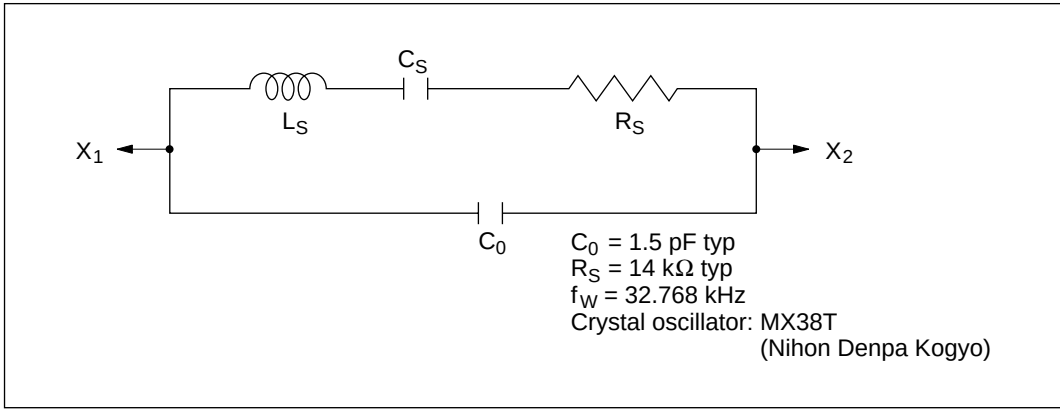


Figure 4-8 Equivalent Circuit of 32.768-kHz Crystal Oscillator

2. Pin connection when not using subclock

When the subclock is not used, connect pin X_1 to V_{CC} and leave pin X_2 open, as shown in figure 4-9.

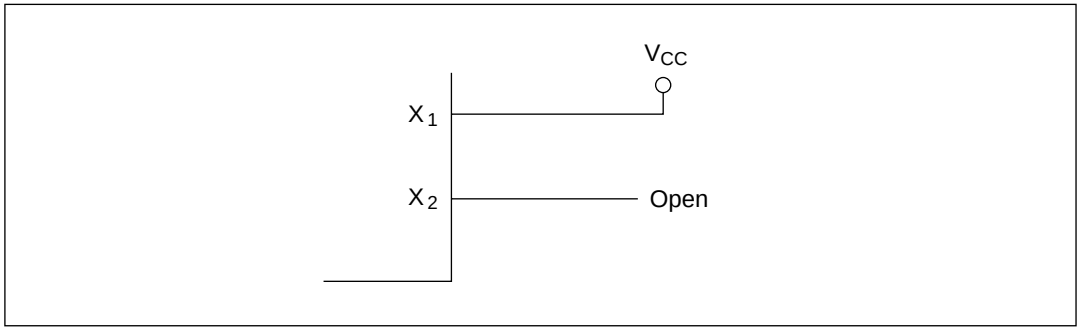


Figure 4-9 Pin Connection when not Using Subclock

4.4 Prescalers

The H8/3657 Series is equipped with two on-chip prescalers having different input clocks (prescaler S and prescaler W). Prescaler S is a 13-bit counter using the system clock (ø) as its input clock. Its prescaled outputs provide internal clock signals for on-chip peripheral modules. Prescaler W is a 5-bit counter using a 32.768-kHz signal divided by 4 ($ø_W/4$) as its input clock. Its prescaled outputs are used by timer A as a time base for timekeeping.

1. Prescaler S (PSS)

Prescaler S is a 13-bit counter using the system clock (ø) as its input clock. It is incremented once per clock period.

Prescaler S is initialized to H'0000 by a reset, and starts counting on exit from the reset state.

In standby mode, watch mode, subactive mode, and subsleep mode, the system clock pulse generator stops. Prescaler S also stops and is initialized to H'0000.

The CPU cannot read or write prescaler S.

The output from prescaler S is shared by the on-chip peripheral modules. The divider ratio can be set separately for each on-chip peripheral function.

In active (medium-speed) mode the clock input to prescaler S is determined by the division factor designated by MA1 and MA0.

2. Prescaler W (PSW)

Prescaler W is a 5-bit counter using a 32.768 kHz signal divided by 4 ($ø_W/4$) as its input clock.

Prescaler W is initialized to H'00 by a reset, and starts counting on exit from the reset state.

Even in standby mode, watch mode, subactive mode, or subsleep mode, prescaler W continues functioning so long as clock signals are supplied to pins X_1 and X_2.

Prescaler W can be reset by setting 1s in bits TMA3 and TMA2 of timer mode register A (TMA).

Output from prescaler W can be used to drive timer A, in which case timer A functions as a time base for timekeeping.

4.5 Note on Oscillators

Oscillator characteristics are closely related to board design and should be carefully evaluated by the user for both mask ROM and ZTAT™ versions, referring to the oscillator element connection examples shown in this section. Oscillator circuit constants will differ depending on the oscillator element, stray capacitance in its interconnecting circuit, and other factors. Suitable constants should be determined in consultation with the oscillator element manufacturer. Design the circuit so that the oscillator element never receives voltages exceeding its maximum rating.

Section 5 Power-Down Modes

5.1 Overview

The H8/3657 Series has eight modes of operation after a reset. These include seven power-down modes, in which power dissipation is significantly reduced. Table 5-1 gives a summary of the eight operating modes.

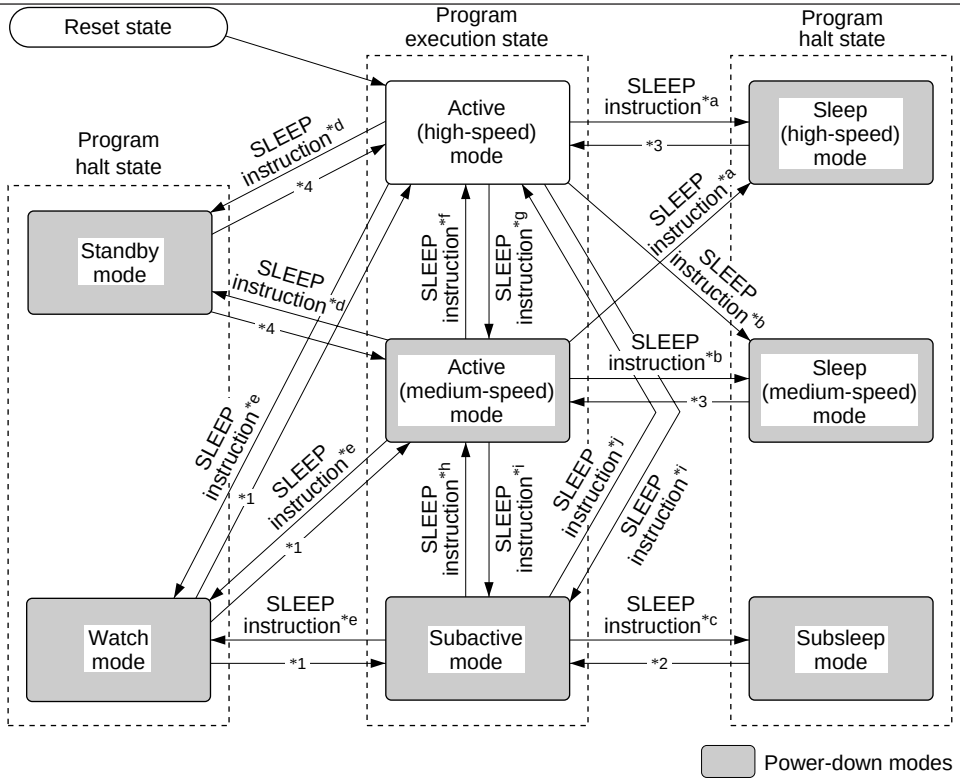
Table 5-1 Operating Modes

Operating Mode	Description
Active (high-speed) mode	The CPU and all on-chip peripheral functions are operable on the system clock
Active (medium-speed) mode	The CPU and all on-chip peripheral functions are operable on the system clock, but at 1/64, 1/32, 1/6, or 1/8* the speed in active (high-speed) mode
Subactive mode	The CPU, and the time-base function of timer A are operable on the subclock
Sleep (high-speed) mode	The CPU halts. On-chip peripheral functions except PWM are operable on the system clock
Sleep (medium-speed) mode	The CPU halts. On-chip peripheral functions except PWM are operable on the system clock, but at 1/64, 1/32, 1/6, or 1/8* the speed in active (high-speed) mode
Subsleep mode	The CPU halts. The time-base function of timer A are operable on the subclock
Watch mode	The CPU halts. The time-base function of timer A is operable on the subclock
Standby mode	The CPU and all on-chip peripheral functions halt

Note: * Determined by the value set in bits MA1 and MA0 of system control register 1 (SYSCR1).

Of these eight operating modes, all but the active (high-speed) mode are power-down modes. In this section the two active modes (high-speed and medium speed) will be referred to collectively as active mode, and the two sleep modes (high-speed and medium speed) will be referred to collectively as sleep mode.

Figure 5-1 shows the transitions among these operation modes. Table 5-2 indicates the internal states in each mode.



Mode Transition Conditions (1)

	LSON	MSON	SSBY	TMA3	DTON
a	0	0	0	*	0
b	0	1	0	*	0
c	1	*	0	1	0
d	0	*	1	0	0
e	*	*	1	1	0
f	0	0	0	*	1
g	0	1	0	*	1
h	0	1	1	1	1
i	1	*	1	1	1
J	0	0	1	1	1

* Don't care

Mode Transition Conditions (2)

	Interrupt Sources
1	Timer A interrupt, IRQ_0 interrupt
2	Timer A interrupt, IRQ_3 to IRQ_0 interrupts, INT interrupt
3	All interrupts
4	IRQ_1 or IRQ_0 interrupt

Notes: 1. A transition between different modes cannot be made to occur simply because an interrupt request is generated. Make sure that interrupt handling is performed after the interrupt is accepted.
2. Details on the mode transition conditions are given in the explanations of each mode, in sections 5-2 through 5-8.

Figure 5-1 Mode Transition Diagram

Table 5-2 Internal State in Each Operating Mode

Function		Active Mode: High-Speed	Active Mode: Medium-Speed	Sleep Mode: High-Speed	Sleep Mode: Medium-Speed	Watch Mode	Subactive Mode	Subsleep Mode	Standby Mode
System clock oscillator		Functions	Functions	Functions	Functions	Halted	Halted	Halted	Halted
Subclock oscillator		Functions	Functions	Functions	Functions	Functions	Functions	Functions	Functions
CPU operations	Instructions	Functions	Functions	Halted	Halted	Halted	Functions	Halted	Halted
	Registers			Retained	Retained	Retained		Retained	Retained
	RAM								
	I/O ports								Retained*1
External interrupts	IRQ_0	Functions	Functions	Functions	Functions	Functions	Functions	Functions	Functions
	IRQ_1					Retained*2			
	IRQ_2								Retained*2
	IRQ_3								
	INT_0	Functions	Functions	Functions	Functions	Retained*2	Functions	Functions	Retained*2
	INT_1								
	INT_2								
	INT_3								
	INT_4								
	INT_5								
	INT_6								
	INT_7								
Peripheral functions	Timer A	Functions	Functions	Functions	Functions	Functions*3	Functions*3	Functions*3	Retained
	Timer B1					Retained	Retained	Retained	
	Timer V					Reset	Reset	Reset	Reset
	Timer X								
	Watchdog timer					Retained	Retained	Retained	Retained
	SCI1								
	SCI3					Reset	Reset	Reset	Reset
	PWM			Retained	Retained	Retained	Retained	Retained	Retained
	A/D converter			Functions	Functions				

Notes: 1. Register contents are retained, but output is high-impedance state.
2. External interrupt requests are ignored. Interrupt request register contents are not altered.
3. Functions if timekeeping time-base function is selected.

5.1.1 System Control Registers

The operation mode is selected using the system control registers described in table 5-3.

Table 5-3 System Control Registers

Name	Abbreviation	R/W	Initial Value	Address
System control register 1	SYSCR1	R/W	H'07	H'FFF0
System control register 2	SYSCR2	R/W	H'E0	H'FFF1

1. System control register 1 (SYSCR1)

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	LSON	—	MA1	MA0
Initial value	0	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W

SYSCR1 is an 8-bit read/write register for control of the power-down modes.

Upon reset, SYSCR1 is initialized to H'07.

Bit 7: Software standby (SSBY)

This bit designates transition to standby mode or watch mode.

Bit 7 SSBY	Description
0	• When a SLEEP instruction is executed in active mode, a transition is made to sleep mode (initial value) • When a SLEEP instruction is executed in subactive mode, a transition is made to subsleep mode
1	• When a SLEEP instruction is executed in active mode, a transition is made to standby mode or watch mode • When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode

Bits 6 to 4: Standby timer select 2 to 0 (STS2 to STS0)

These bits designate the time the CPU and peripheral modules wait for stable clock operation after exiting from standby mode or watch mode to active mode due to an interrupt. The designation should be made according to the clock frequency so that the waiting time is at least 10 ms.

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description	
0	0	0	Wait time = 8,192 states	(initial value)
0	0	1	Wait time = 16,384 states	
0	1	0	Wait time = 32,768 states	
0	1	1	Wait time = 65,536 states	
1	*	*	Wait time = 131,072 states	

Note: * Don't care

Bit 3: Low speed on flag (LSON)

This bit chooses the system clock (ø) or subclock ($ø_{SUB}$) as the CPU operating clock when watch mode is cleared. The resulting operation mode depends on the combination of other control bits and interrupt input.

Bit 3 LSON	Description	
0	The CPU operates on the system clock (ø)	(initial value)
1	The CPU operates on the subclock ($ø_{SUB}$)	

Bits 2: Reserved bits

Bit 2 is reserved: it is always read as 1 and cannot be modified.

Bits 1 and 0: Active (medium-speed) mode clock select (MA1, MA0)

Bits 1 and 0 choose $ø_{osc}$/128, $ø_{osc}$/64, $ø_{osc}$/32, or $ø_{osc}$/16 as the operating clock in active (medium-speed) mode and sleep (medium-speed) mode. MA1 and MA0 should be written in active (high-speed) mode or subactive mode.

Bit 1 MA1	Bit 0 MA0	Description	
0	0	$ø_{osc}$/16	
0	1	$ø_{osc}$/32	
1	0	$ø_{osc}$/64	
1	1	$ø_{osc}$/128	(initial value)

2. System control register 2 (SYSCR2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	NESEL	DTON	MSON	SA1	SA0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

SYSCR2 is an 8-bit read/write register for power-down mode control.

Upon reset, SYSCR2 is initialized to H'E0.

Bits 7 to 5: Reserved bits

These bits are reserved; they are always read as 1, and cannot be modified.

Bit 4: Noise elimination sampling frequency select (NESEL)

This bit selects the frequency at which the watch clock signal ($ø_W$) generated by the subclock pulse generator is sampled, in relation to the oscillator clock ($ø_{OSC}$) generated by the system clock pulse generator. When $ø_{OSC}$ = 2 to 10 MHz, clear NESEL to 0.

Bit 4 NESEL	**Description**	
0	Sampling rate is $ø_{OSC}$/16	(initial value)
1	Sampling rate is $ø_{OSC}$/4	

Bit 3: Direct transfer on flag (DTON)

This bit designates whether or not to make direct transitions among active (high-speed), active (medium-speed) and subactive mode when a SLEEP instruction is executed. The mode to which the transition is made after the SLEEP instruction is executed depends on a combination of this and other control bits.

Bit 3 DTON	Description
0	• When a SLEEP instruction is executed in active mode, a transition is made to standby mode, watch mode, or sleep mode (initial value) • When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode or subsleep mode
1	• When a SLEEP instruction is executed in active (high-speed) mode, a direct transition is made to active (medium-speed) mode if SSBY = 0, MSON = 1, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1 • When a SLEEP instruction is executed in active (medium-speed) mode, a direct transition is made to active (high-speed) mode if SSBY = 0, MSON = 0, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1 • When a SLEEP instruction is executed in subactive mode, a direct transition is made to active (high-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 0, or to active (medium-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 1

Bit 2: Medium speed on flag (MSON)

After standby, watch, or sleep mode is cleared, this bit selects active (high-speed) or sleep (high-speed) mode, and active (medium speed) or sleep (medium-speed) mode.

Bit 2 MSON	Description
0	• After standby, watch, or sleep mode is cleared, operation is in active (high-speed) mode • When a SLEEP instruction is executed in active mode, a transition is made to sleep (high-speed) mode
1	• After standby, watch, or sleep mode is cleared, operation is in active (medium-speed) mode • When a SLEEP instruction is executed in active mode, a transition is made to sleep (medium-speed) mode

Bits 1 and 0: Subactive mode clock select (SA1 and SA0)

These bits select the CPU clock rate ($ø_W/2$, $ø_W/4$, or $ø_W/8$) in subactive mode. SA1 and SA0 cannot be modified in subactive mode.

Bit 1 SA1	Bit 0 SA0	Description	
0	0	$ø_W/8$	(initial value)
0	1	$ø_W/4$	
1	*	$ø_W/2$	

Note: * Don't care

5.2 Sleep Mode

5.2.1 Transition to Sleep Mode

1. Transition to sleep (high-speed) mode

The system goes from active mode to sleep (high-speed) mode when a SLEEP instruction is executed while the SSBY and LSON bits in SYSCR1 and the MSON and DTON bits in SYSCR2 are all cleared to 0. In sleep (high-speed) mode CPU operation is halted but the on-chip peripheral functions other than PWM are operational. CPU register contents are retained.

2. Transition to sleep (medium-speed) mode

The system goes from active mode to sleep (medium-speed) mode when a SLEEP instruction is executed while the SSBY and LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is set to 1, and the DTON bit in SYSCR2 is cleared to 0. In sleep (medium-speed) mode, as in sleep (high-speed) mode, CPU operation is halted but the on-chip peripheral functions other than PWM are operational. The clock frequency in sleep (medium-speed) mode is determined by the MA1 and MA0 bits in SYSCR1. CPU register contents are retained.

5.2.2 Clearing Sleep Mode

Sleep mode is cleared by any interrupt (timer A, timer B1, timer X, timer V, IRQ_3 to IRQ_0, INT_7 to INT_0, SCI_3, SCI_1, or A/D converter), or by input at the $\overline{RES}$ pin.

- Clearing by interrupt

When an interrupt is requested, sleep mode is cleared and interrupt exception handling starts. A transition is made from sleep (high-speed) mode to active (high-speed) mode, or from sleep (medium-speed) mode to active (medium-speed) mode. Sleep mode is not cleared if the I bit of the condition code register (CCR) is set to 1 or the particular interrupt is disabled in the interrupt enable register.

- Clearing by $\overline{RES}$ input

When the $\overline{RES}$ pin goes low, the CPU goes into the reset state and sleep mode is cleared.

5.2.3 Clock Frequency in Sleep (Medium-Speed) Mode

Operation in sleep (medium-speed) mode is clocked at the frequency designated by the MA1 and MA0 bits in SYSCR1.

5.3 Standby Mode

5.3.1 Transition to Standby Mode

The system goes from active mode to standby mode when a SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1, the LSON bit in SYSCR1 is cleared to 0, and bit TMA3 in TMA is cleared to 0. In standby mode the clock pulse generator stops, so the CPU and on-chip peripheral modules stop functioning, but as long as the rated voltage is supplied, the contents of CPU registers, on-chip RAM, and some on-chip peripheral module registers are retained. On-chip RAM contents will be further retained down to a minimum RAM data retention voltage. The I/O ports go to the high-impedance state.

5.3.2 Clearing Standby Mode

Standby mode is cleared by an interrupt (IRQ_1 or IRQ_0) or by input at the $\overline{RES}$ pin.

- Clearing by interrupt

When an interrupt is requested, the system clock pulse generator starts. After the time set in bits STS2–STS0 in SYSCR1 has elapsed, a stable system clock signal is supplied to the entire chip, standby mode is cleared, and interrupt exception handling starts. Operation resumes in active (high-speed) mode if MSON = 0 in SYSCR2, or active (medium-speed) mode if MSON = 1. Standby mode is not cleared if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

- Clearing by $\overline{RES}$ input

When the $\overline{RES}$ pin goes low, the system clock pulse generator starts. After the pulse generator output has stabilized, if the $\overline{RES}$ pin is driven high, the CPU starts reset exception handling. Since system clock signals are supplied to the entire chip as soon as the system clock pulse generator starts functioning, the $\overline{RES}$ pin should be kept at the low level until the pulse generator output stabilizes.

5.3.3 Oscillator Settling Time after Standby Mode is Cleared

Bits STS2 to STS0 in SYSCR1 should be set as follows.

- When a crystal oscillator is used

The table below gives settings for various operating frequencies. Set bits STS2 to STS0 for a waiting time of at least 10 ms.

- When an external clock is used

Any values may be set. Normally the minimum time (STS2 = STS1 = STS0 = 0) should be set.

Table 5-3 Clock Frequency and Settling Time (times are in ms)

STS2	STS1	STS0	Waiting Time	5 MHz	4 MHz	2 MHz	1 MHz	0.5 MHz
0	0	0	8,192 states	1.6	2.0	4.1	8.2	16.4
0	0	1	16,384 states	3.2	4.1	8.2	16.4	32.8
0	1	0	32,768 states	6.6	8.2	16.4	32.8	65.5
0	1	1	65,536 states	13.1	16.4	32.8	65.5	131.1
1	*	*	131,072 states	26.2	32.8	65.5	131.1	262.1

Note: * Don't care

5.4 Watch Mode

5.4.1 Transition to Watch Mode

The system goes from active or subactive mode to watch mode when a SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1 and bit TMA3 in TMA is set to 1.

In watch mode, operation of on-chip peripheral modules other than timer A is halted. As long as a minimum required voltage is applied, the contents of CPU registers, the on-chip RAM and some registers of the on-chip peripheral modules, are retained. I/O ports keep the same states as before the transition.

5.4.2 Clearing Watch Mode

Watch mode is cleared by an interrupt (timer A or IRQ_0) or by a low input at the $\overline{RES}$ pin.

- Clearing by interrupt

When watch mode is cleared by a timer A interrupt or IRQ_0 interrupt, the mode to which a transition is made depends on the settings of LSON in SYSCR1 and MSON in SYSCR2. If both LSON and MSON are cleared to 0, transition is to active (high-speed) mode; if LSON = 0 and MSON = 1, transition is to active (medium-speed) mode; if LSON = 1, transition is to subactive mode. When the transition is to active mode, after the time set in SYSCR1 bits STS2–STS0 has elapsed, a stable clock signal is supplied to the entire chip, watch mode is cleared, and interrupt exception handling starts. Watch mode is not cleared if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

- Clearing by $\overline{RES}$ input

Clearing by $\overline{RES}$ pin is the same as for standby mode; see 5.3.2, Clearing Standby Mode.

5.4.3 Oscillator Settling Time after Watch Mode is Cleared

The waiting time is the same as for standby mode; see 5.3.3, Oscillator Settling Time after Standby Mode is Cleared.

5.5 Subsleep Mode

5.5.1 Transition to Subsleep Mode

The system goes from subactive mode to subsleep mode when a SLEEP instruction is executed while the SSBY bit in SYSCR1 is cleared to 0, LSON bit in SYSCR1 is set to 1, and TMA3 bit in TMA is set to 1. In subsleep mode, operation of on-chip peripheral modules other than timer A and timer C is halted. As long as a minimum required voltage is applied, the contents of CPU registers, the on-chip RAM and some registers of the on-chip peripheral modules are retained. I/O ports keep the same states as before the transition.

5.5.2 Clearing Subsleep Mode

Subsleep mode is cleared by an interrupt (timer A, IRQ_3 to IRQ_0, INT_7 to INT_0) or by a low input at the $\overline{RES}$ pin.

- Clearing by interrupt

When an interrupt is requested, subsleep mode is cleared and interrupt exception handling starts. Subsleep mode is not cleared if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

- Clearing by $\overline{RES}$ input

Clearing by $\overline{RES}$ pin is the same as for standby mode; see 5.3.2, Clearing Standby Mode.

5.6 Subactive Mode

5.6.1 Transition to Subactive Mode

Subactive mode is entered from watch mode if a timer A or IRQ_0 interrupt is requested while the LSON bit in SYSCR1 is set to 1. From subsleep mode, subactive mode is entered if a timer A, IRQ_3 to IRQ_0, or INT_7 to INT_0 interrupt is requested. A transition to subactive mode does not take place if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

5.6.2 Clearing Subactive Mode

Subactive mode is cleared by a SLEEP instruction or by a low input at the $\overline{RES}$ pin.

- Clearing by SLEEP instruction

If a SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1 and TMA3 bit in TMA is set to 1, subactive mode is cleared and watch mode is entered. If a SLEEP instruction is executed while SSBY = 0 and LSON = 1 in SYSCR1 and TMA3 = 1 in TMA, subsleep mode is entered. Direct transfer to active mode is also possible; see 5.8, Direct Transfer.

- Clearing by $\overline{RES}$ pin

Clearing by $\overline{RES}$ pin is the same as for standby mode; see 5.3.2, Clearing Standby Mode.

5.6.3 Operating Frequency in Subactive Mode

The operating frequency in subactive mode is set in bits SA1 and SA0 in SYSCR2. The choices are $ø_W/2$, $ø_W/4$, and $ø_W/8$.

5.7 Active (Medium-Speed) Mode

5.7.1 Transition to Active (Medium-Speed) Mode

If the LSON bit in SYSCR2 is set to 1 while the LSON bit in SYSCR1 is cleared to 0, a transition to active (medium-speed) mode results from IRQ_0 or IRQ_1 interrupts in standby mode, timer A or IRQ_0 interrupts in watch mode, or any interrupt in sleep (medium-speed) mode. A transition to active (medium-speed) mode does not take place if the I bit of CCR is set to 1 or the particular interrupt is disabled in the interrupt enable register.

5.7.2 Clearing Active (Medium-Speed) Mode

Active (medium-speed) mode is cleared by a SLEEP instruction or by a low input at the $\overline{RES}$ pin.

- Clearing by SLEEP instruction

A transition to standby mode takes place if the SLEEP instruction is executed while the SSBY bit in SYSCR1 is set to 1, the LSON bit in SYSCR1 is cleared to 0, and the TMA3 bit in TMA is cleared to 0. The system goes to watch mode if the SSBY bit in SYSCR1 is set to 1 and bit TMA3 in TMA is set to 1 when a SLEEP instruction is executed.

When both SSBY and LSON are cleared to 0 in SYSCR1 and a SLEEP instruction is executed, sleep (high-speed) mode is entered if MSON is cleared to 0 in SYSCR2, and sleep (medium-speed) mode is entered if MSON is set to 1. Direct transfer to active (high-speed) mode or to subactive mode is also possible. See 5.8, Direct Transfer, below for details.

- Clearing by $\overline{RES}$ pin

When the $\overline{RES}$ pin goes low, the CPU enters the reset state and active (medium-speed) mode is cleared.

5.7.3 Operating Frequency in Active (Medium-Speed) Mode

Operation in active (medium-speed) mode is clocked at the frequency designated by the MA1 and MA0 bits in SYSCR1.

5.8 Direct Transfer

The CPU can execute programs in three modes: active (high-speed) mode, active (medium-speed) mode, and subactive mode. A direct transfer is a transition among these three modes without the stopping of program execution. A direct transfer can be made by executing a SLEEP instruction while the DTON bit in SYSCR2 is set to 1. After the mode transition, direct transfer interrupt exception handling starts.

If the direct transfer interrupt is disabled in interrupt enable register 2, a transition is made instead to sleep mode or watch mode. Note that if a direct transition is attempted while the I bit in CCR is set to 1, sleep mode or watch mode will be entered, and it will be impossible to clear the resulting mode by means of an interrupt.

- Direct transfer from active (high-speed) mode to active (medium-speed) mode

When a SLEEP instruction is executed in active (high-speed) mode while the SSBY and LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is set to 1, and the DTON bit in SYSCR2 is set to 1, a transition is made to active (medium-speed) mode via sleep mode.

- Direct transfer from active (medium-speed) mode to active (high-speed) mode

When a SLEEP instruction is executed in active (medium-speed) mode while the SSBY and LSON bits in SYSCR1 are cleared to 0, the MSON bit in SYSCR2 is cleared to 0, and the DTON bit in SYSCR2 is set to 1, a transition is made to active (high-speed) mode via sleep mode.

- Direct transfer from active (high-speed) mode to subactive mode

When a SLEEP instruction is executed in active (high-speed) mode while the SSBY and LSON bits in SYSCR1 are set to 1, the DTON bit in SYSCR2 is set to 1, and the TMA3 bit in TMA is set to 1, a transition is made to subactive mode via watch mode.

- Direct transfer from subactive mode to active (high-speed) mode

When a SLEEP instruction is executed in subactive mode while the SSBY bit in SYSCR1 is set to 1, the LSON bit in SYSCR1 is cleared to 0, the MSON bit in SYSCR2 is cleared to 0, the DTON bit in SYSCR2 is set to 1, and the TMA3 bit in TMA is set to 1, a transition is made directly to active (high-speed) mode via watch mode after the waiting time set in SYSCR1 bits STS2 to STS0 has elapsed.

- Direct transfer from active (medium-speed) mode to subactive mode

When a SLEEP instruction is executed in active (medium-speed) while the SSBY and LSON bits in SYSCR1 are set to 1, the DTON bit in SYSCR2 is set to 1, and the TMA3 bit in TMA is set to 1, a transition is made to subactive mode via watch mode.

- Direct transfer from subactive mode to active (medium-speed) mode

When a SLEEP instruction is executed in subactive mode while the SSBY bit in SYSCR1 is set to 1, the LSON bit in SYSCR1 is cleared to 0, the MSON bit in SYSCR2 is set to 1, the DTON bit in SYSCR2 is set to 1, and the TMA3 bit in TMA is set to 1, a transition is made directly to active (medium-speed) mode via watch mode after the waiting time set in SYSCR1 bits STS2 to STS0 has elapsed.

Section 6 ROM

6.1 Overview

The H8/3657 has 60 kbytes of on-chip mask ROM or PROM. The H8/3656 has 48 kbytes of mask ROM. The H8/3655 has 40 kbytes of mask ROM. The H8/3654 has 32 kbytes of on-chip mask ROM. The H8/3653 has 24 kbytes of mask ROM. The H8/3652 has 16 kbytes of mask ROM. The ROM is connected to the CPU by a 16-bit data bus, allowing high-speed two-state access for both byte data and word data.

6.1.1 Block Diagram

Figure 6-1 shows a block diagram of the on-chip ROM.

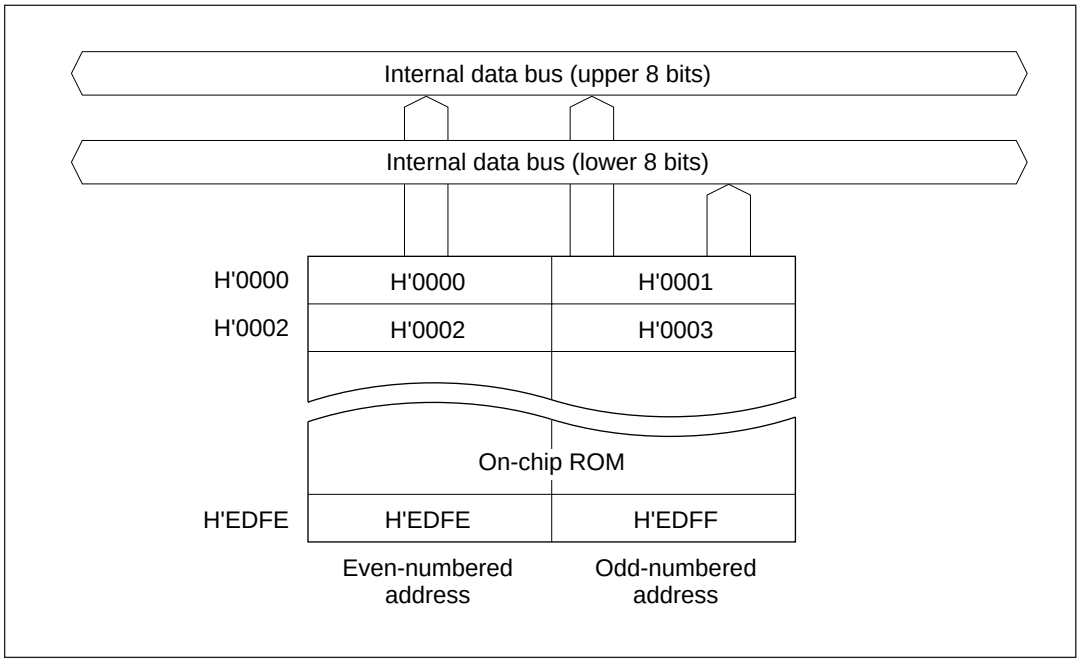


Figure 6-1 ROM Block Diagram (H8/3657)

6.2 PROM Mode

6.2.1 Setting to PROM Mode

If the on-chip ROM is PROM, setting the chip to PROM mode stops operation as a microcontroller and allows the PROM to be programmed in the same way as the standard HN27C101 EPROM. Table 6-1 shows how to set the chip to PROM mode.

Table 6-1 Setting to PROM Mode

Pin Name	Setting
TEST	High level
PB_4/AN_4	Low level
PB_5/AN_5	
PB_6/AN_6	High level

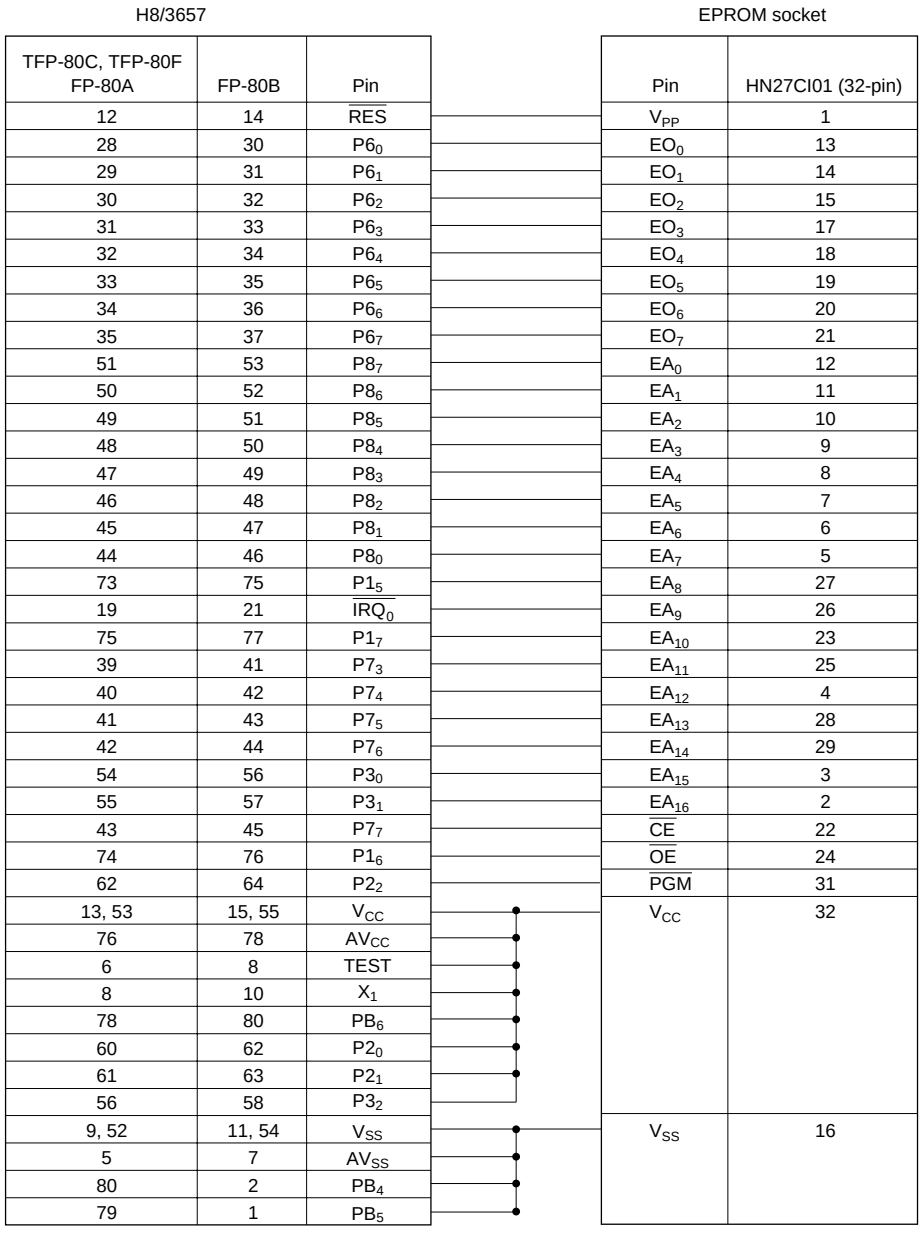
6.2.2 Socket Adapter Pin Arrangement and Memory Map

A standard PROM programmer can be used to program the PROM. A socket adapter is required for conversion to 32 pins, as listed in table 6-2.

Figure 6-2 shows the pin-to-pin wiring of the socket adapter. Figure 6-3 shows a memory map.

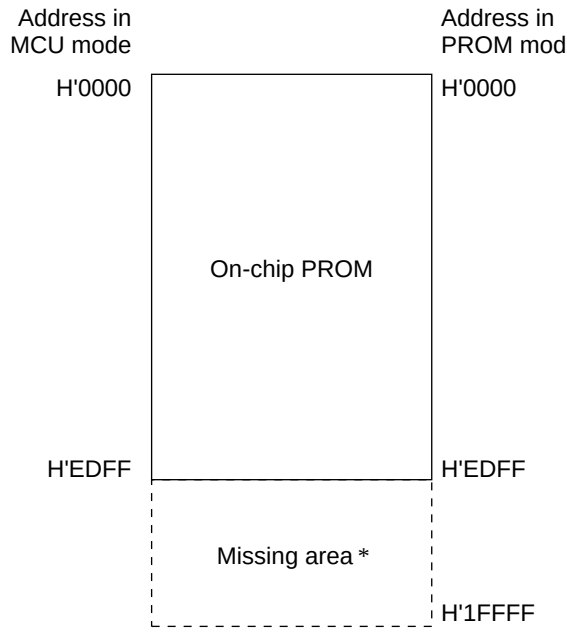
Table 6-2 Socket Adapter

Package	Socket Adapter
80-pin (TFP-80F)	
80-pin (TFP-80C)	
80-pin (FP-80A)	
80-pin (FP-80B)	



Note: Pins not indicated in the figure should be left open.

Figure 6-2 Socket Adapter Pin Correspondence (with HN27C101)



Note: * If read in PROM mode, this address area returns unpredictable output data. When programming with a PROM programmer, be sure to specify addresses from H'0000 to H'EDFF.
If address H'EE00 and higher addresses are programmed by mistake, it may become impossible to program the PROM or verify the programmed data. When programming, assign H'FF data to this address area (H'EE00 to H'1FFFF).

Figure 6-3 Memory Map in PROM Mode

6.3 Programming

The write, verify, and other modes are selected as shown in table 6-3 in H8/3657 PROM mode.

Table 6-3 Mode Selection in H8/3657 PROM Mode

	Pin						
Mode	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{PGM}$**	**V_{PP}**	**V_{CC}**	**EO_7 to EO_0**	**EA_{16} to EA_0**
Write	L	H	L	V_{PP}	V_{CC}	Data input	Address input
Verify	L	L	H	V_{PP}	V_{CC}	Data output	Address input
Programming disabled	L	L	L	V_{PP}	V_{CC}	High impedance	Address input
	L	H	H				
	H	L	L				
	H	H	H				

Notation
L: Low level
H: High level
V_{PP}: V_{PP} level
V_{CC}: V_{CC} level

The specifications for writing and reading the on-chip PROM are identical to those for the standard HN27C101 EPROM. Page programming is not supported, however. The PROM writer must not be set to page mode. A PROM programmer that provides only page programming mode cannot be used. When selecting a PROM programer, check that it supports a byte-by-byte high-speed, high-reliability programming method. Be sure to set the address range to H'0000 to H'EDFF.

6.3.1 Writing and Verifying

An efficient, high-speed, high-reliability method is available for writing and verifying the PROM data. This method achieves high speed without voltage stress on the device and without lowering the reliability of written data. The basic flow of this high-speed, high-reliability programming method is shown in figure 6-4.

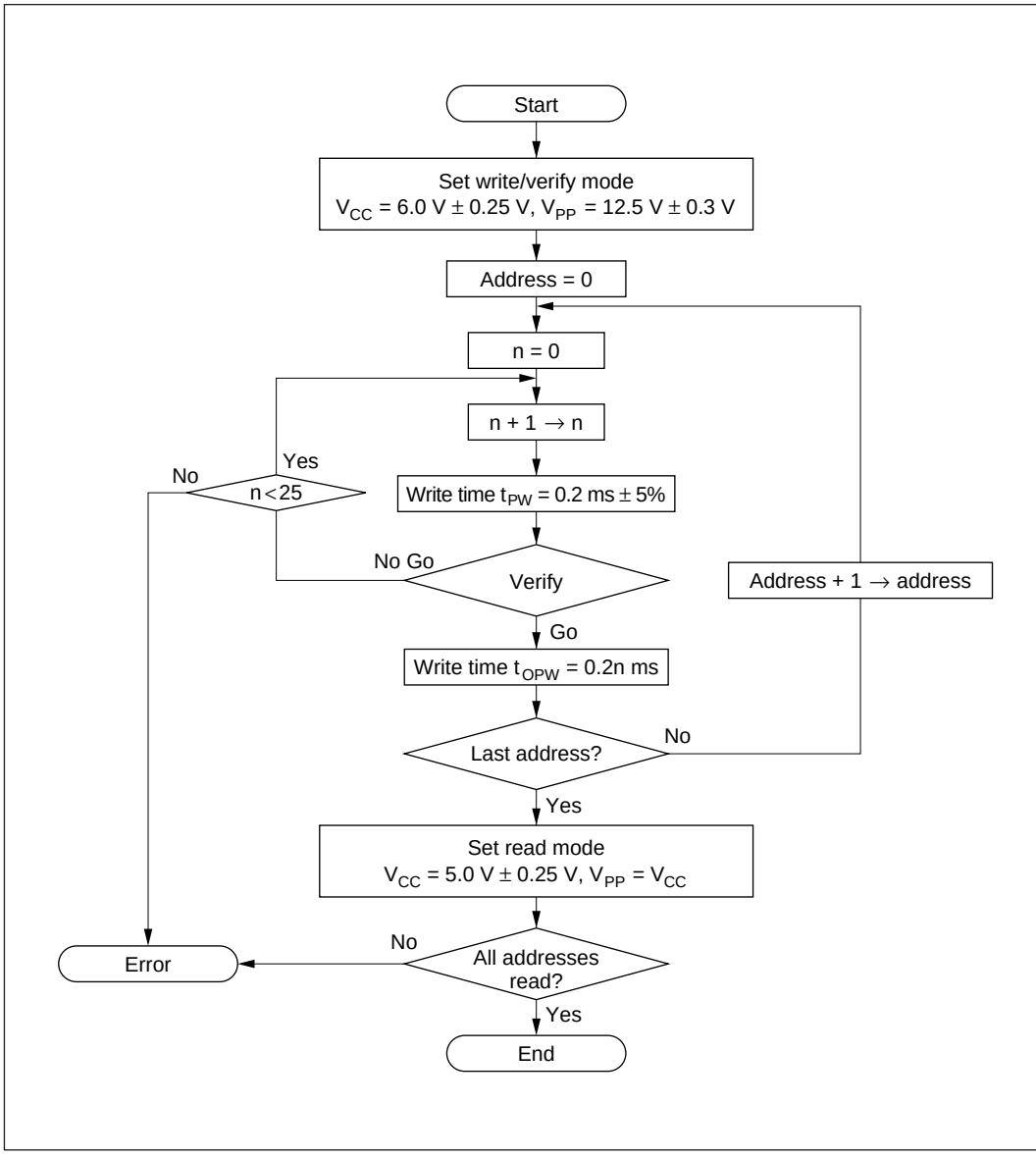


Figure 6-4 High-Speed, High-Reliability Programming Flow Chart

Table 6-4 and table 6-5 give the electrical characteristics in programming mode.

Table 6-4 DC Characteristics

(Conditions: V_{CC} = 6.0 V ±0.25 V, V_{PP} = 12.5 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Input high-level voltage	$\overline{EO_7}$ to $\overline{EO_0}$, $\overline{EA_{16}}$ to $\overline{EA_0}$ $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	V_{IH}	2.4	—	V_{CC} + 0.3	V	
Input low-level voltage	$\overline{EO_7}$ to $\overline{EO_0}$, $\overline{EA_{16}}$ to $\overline{EA_0}$ $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	V_{IL}	–0.3	—	0.8	V	
Output high-level voltage	$\overline{EO_7}$ to $\overline{EO_0}$	V_{OH}	2.4	—	—	V	I_{OH} = –200 µA
Output low-level voltage	$\overline{EO_7}$ to $\overline{EO_0}$	V_{OL}	—	—	0.45	V	I_{OL} = 0.8 mA
Input leakage current	$\overline{EO_7}$ to $\overline{EO_0}$, $\overline{EA_{16}}$ to $\overline{EA_0}$ $\overline{OE}$, $\overline{CE}$, $\overline{PGM}$	$\lvert I_{LI} \rvert$	—	—	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 6-5 AC Characteristics

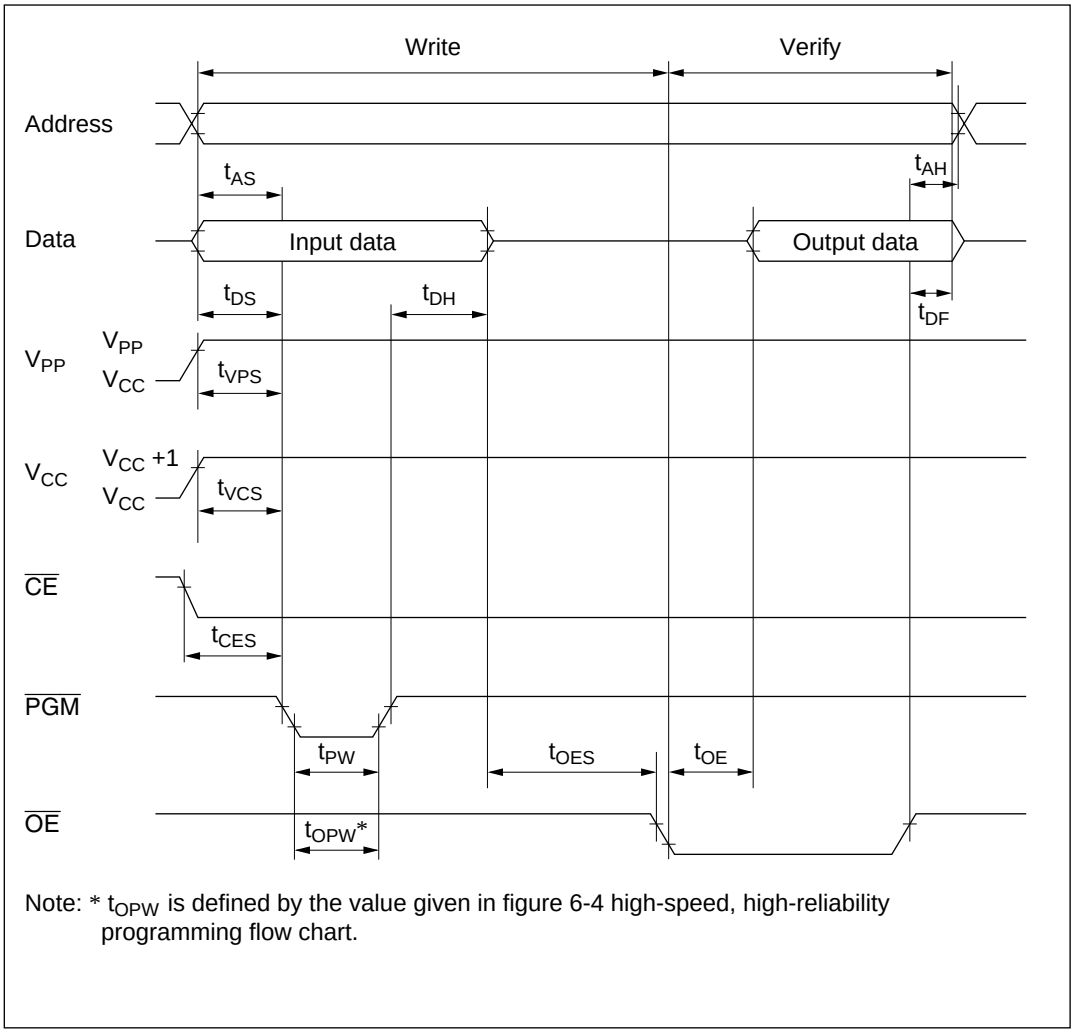
(Conditions: V_{CC} = 6.0 V ±0.25 V, V_{PP} = 12.5 V ±0.3 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Address setup time	t_{AS}	2	—	—	µs	Figure 6-5*1
$\overline{OE}$ 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}*2	—	—	130	ns	
V_{PP} setup time	t_{VPS}	2	—	—	µs	
Programming pulse width	t_{PW}	0.19	0.20	0.21	ms	
$\overline{PGM}$ pulse width for overwrite programming	t_{OPW}*3	0.19	—	5.25	ms	
V_{CC} setup time	t_{VCS}	2	—	—	µs	
$\overline{CE}$ setup time	t_{CES}	2	—	—	µs	
Data output delay time	t_{OE}	0	—	200	ns	

Notes: 1. Input pulse level: 0.45 V to 2.4 V
Input rise time/fall time ≤ 20 ns
Timing reference levels Input: 0.8 V, 2.0 V
Output: 0.8 V, 2.0 V

2. t_{DF} is defined at the point at which the output is floating and the output level cannot be read.
3. t_{OPW} is defined by the value given in figure 6-4 high-speed, high-reliability programming flow chart.

Figure 6-5 shows a write/verify timing diagram.



Note: * t_{OPW} is defined by the value given in figure 6-4 high-speed, high-reliability programming flow chart.

Figure 6-5 PROM Write/Verify Timing

6.3.2 Programming Precautions

- Use the specified programming voltage and timing.

  The programming voltage in PROM mode (V_{PP}) is 12.5 V. Use of a higher voltage can permanently damage the chip. Be especially careful with respect to PROM programmer overshoot.

  Setting the PROM programmer to Hitachi specifications for the HN27C101 will result in correct V_{PP} of 12.5 V.

- Make sure the index marks on the PROM programmer socket, socket adapter, and chip are properly aligned. If they are not, the chip may be destroyed by excessive current flow. Before programming, be sure that the chip is properly mounted in the PROM programmer.

- Avoid touching the socket adapter or chip while programming, since this may cause contact faults and write errors.

- Select the programming mode carefully. The chip cannot be programmed in page programming mode.

- When programming with a PROM programmer, be sure to specify addresses from H'0000 to H'EDFF. If address H'EE00 and higher addresses are programmed by mistake, it may become impossible to program the PROM or verify the programmed data. When programming, assign H'FF data to the address area from H'EE00 to H'1FFFF.

6.4 Reliability of Programmed Data

A highly effective way of assuring data retention characteristics after programming is to screen the chips by baking them at a temperature of 150°C. This quickly eliminates PROM memory cells prone to initial data retention failure.

Figure 6-6 shows a flowchart of this screening procedure.

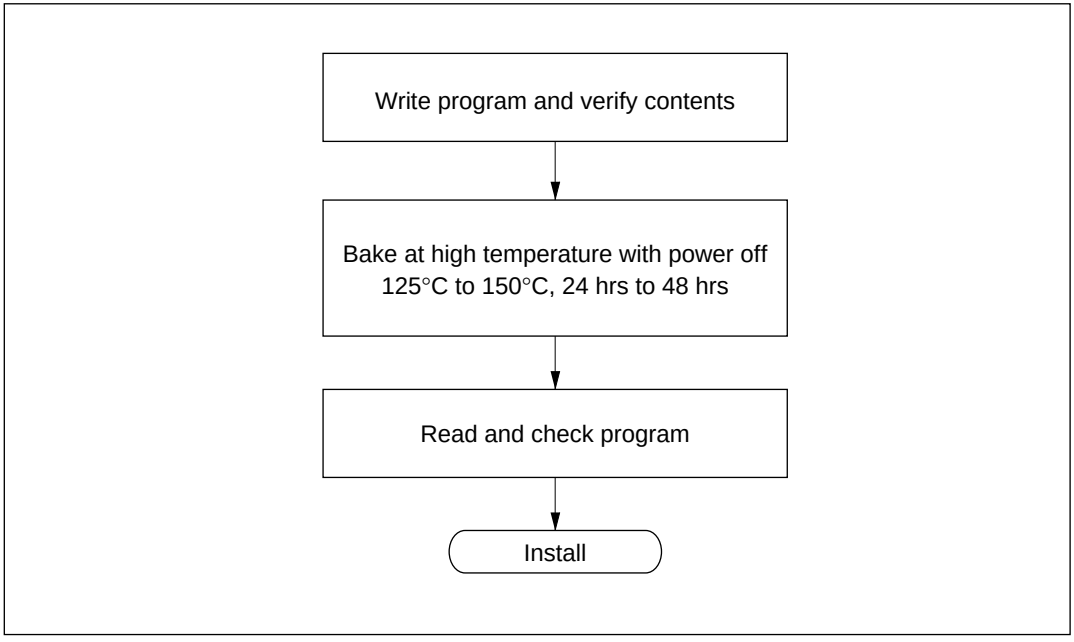


Figure 6-6 Recommended Screening Procedure

If write errors occur repeatedly while the same PROM programmer is being used, stop programming and check for problems in the PROM programmer and socket adapter, etc.

Please notify your Hitachi representative of any problems occurring during programming or in screening after high-temperature baking.

Section 7 RAM

7.1 Overview

The H8/3657 Series has 1 kbyte or 2 kbytes of high-speed static RAM on-chip. The RAM is connected to the CPU by a 16-bit data bus, allowing high-speed 2-state access for both byte data and word data.

7.1.1 Block Diagram

Figure 7-1 shows a block diagram of the on-chip RAM.

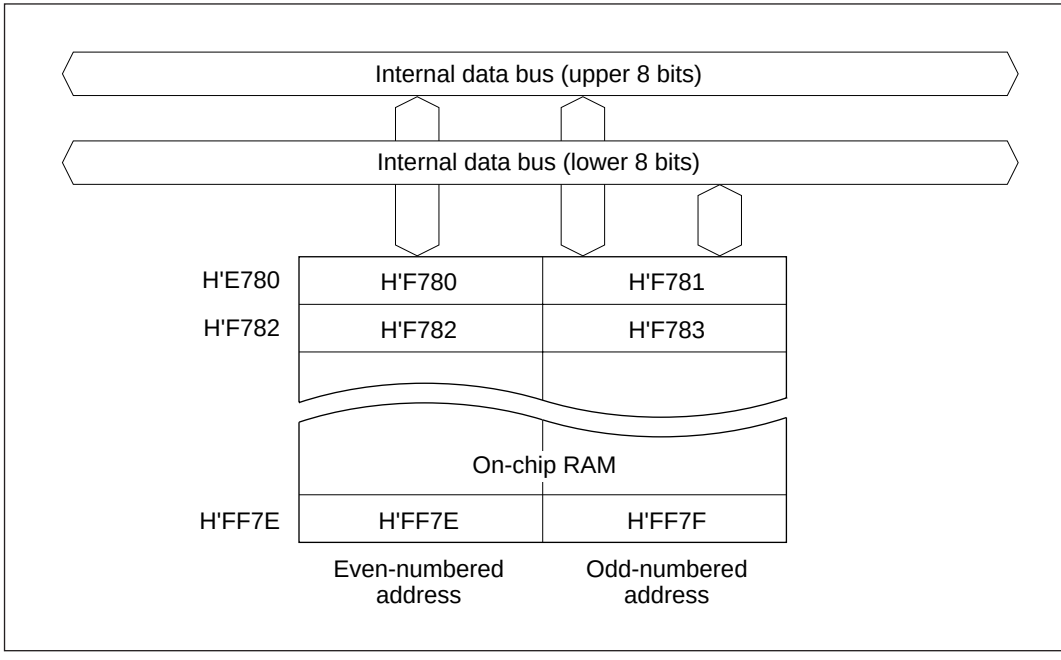


Figure 7-1 RAM Block Diagram (Example of 2 kbytes ROM)

Section 8 I/O Ports

8.1 Overview

The H8/3657 Series is provided with six 8-bit I/O ports, one 6-bit I/O port, one 5-bit I/O ports, and one 8-bit input-only port. Table 8-1 indicates the functions of each port.

Each port has of a port control register (PCR) that controls input and output, and a port data register (PDR) for storing output data. Input or output can be assigned to individual bits.
See 2.9.2, Notes on Bit Manipulation, for information on executing bit-manipulation instructions to write data in PCR or PDR.

Block diagrams of each port are given in Appendix C.

Table 8-1 Port Functions

Port	Description	Pins	Other Functions	Function Switching Register
Port 1	• 8-bit I/O port • MOS input pull-up option	$P1_7$/$\overline{IRQ_3}$/TRGV $P1_6$ to $P1_5$/ $\overline{IRQ_2}$ to $\overline{IRQ_1}$	External interrupt 3, timer V trigger input External interrupts 2 and 1	PMR1
		$P1_4$/PWM	14-bit PWM output	PMR1
		$P1_3$ to $P1_1$		
		$P1_0$/TMOW	Timer A clock output	PMR1
Port 2	• 8-bit I/O port	$P2_7$ to $P2_3$		
		$P2_2$/TXD	SCI3 data output	PMR7
		$P2_1$/RXD	SCI3 data input	SCR3
		$P2_0$/SCK_1	SCI3 clock input/output	SCR3, SMR
Port 3	• 6-bit I/O port • MOS input pull-up option	$P3_5$ to $P3_3$		
		$P3_2$/SO_1 $P3_1$/SI_1 $P3_0$/SCK_1	SCI1 data output (SO_1), data input (SI_1), clock input/output (SCK_1)	PMR3
Port 5	• 8-bit I/O port • MOS input pull-up	$P5_7$ /$\overline{INT_7}$	INT interrupt 7	
		$P5_6$ /$\overline{INT_6}$/ TMIB	INT interrupt 6 Timer B 1 event input	
		$P5_5$/$\overline{INT_5}$/ $\overline{ADTRG}$	INT interrupt 5 A/D converter external trigger input	
		$P5_4$ to $P5_0$/ $\overline{INT_4}$ to $\overline{INT_0}$	INT interrupts 4 to 0	

Table 8-1 Port Functions (cont)

Port	Description	Pins	Other Functions	Function Switching Register
Port 6	• 8-bit I/O port	$P6_7$ to $P6_0$		
Port 7	• 8-bit I/O port	$P7_7$		
		$P7_6$/TMOV	Timer V compare-match output	TCSRV
		$P7_5$/TMCIV	Timer V clock input	
		$P7_4$/TMRIV	Timer V reset input	
		$P7_3$ to $P7_0$		
Port 8	• 8-bit I/O port	$P8_7$		
		$P8_6$/FTID	Timer X input capture D input	
		$P8_5$/FTIC	Timer X input capture C input	
		$P8_4$/FTIB	Timer X input capture B input	
		$P8_3$/FTIA	Timer X input capture A input	
		$P8_2$/FTOB	Timer X output compare B output	TOCR
		$P8_1$/FTOA	Timer X output compare A output	TOCR
		$P8_0$/FTCI	Timer X clock input	
Port 9	• 5-bit I/O port	$P9_0$ to $P9_4$		
Port B	• 8-bit input port	PB_7 to PB_0/ AN_7 to AN_0	A/D converter analog input (AN_7 to AN_0)	

8.2 Port 1

8.2.1 Overview

Port 1 is a 8-bit I/O port. Figure 8-1 shows its pin configuration.

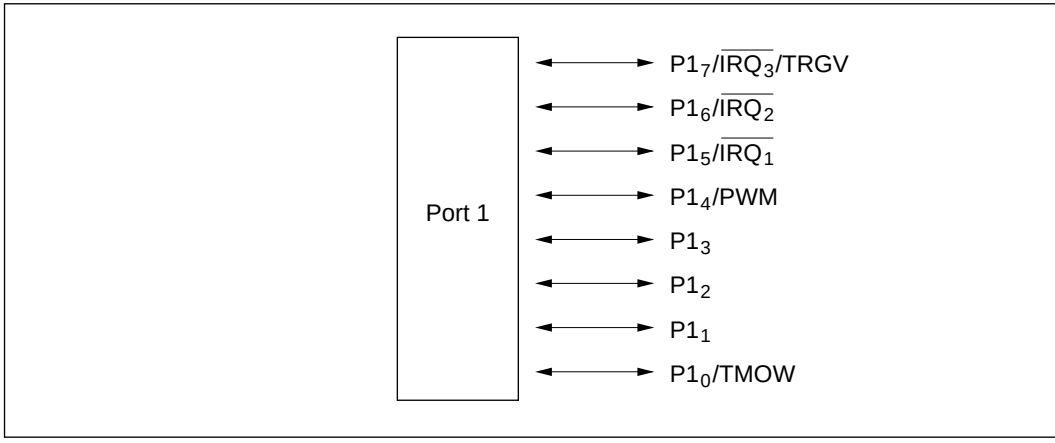


Figure 8-1 Port 1 Pin Configuration

8.2.2 Register Configuration and Description

Table 8-2 shows the port 1 register configuration.

Table 8-2 Port 1 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 1	PDR1	R/W	H'00	H'FFD4
Port control register 1	PCR1	W	H'00	H'FFE4
Port pull-up control register 1	PUCR1	R/W	H'00	H'FFED
Port mode register 1	PMR1	R/W	H'04	H'FFFC

1. Port data register 1 (PDR1)

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

PDR1 is an 8-bit register that stores data for port 1 pins $P1_7$ and $P1_0$. If port 1 is read while PCR1 bits are set to 1, the values stored in PDR1 are read, regardless of the actual pin states. If port 1 is read while PCR1 bits are cleared to 0, the pin states are read.

Upon reset, PDR1 is initialized to H'00.

2. Port control register 1 (PCR1)

Bit	7	6	5	4	3	2	1	0
	$PCR1_7$	$PCR1_6$	$PCR1_5$	$PCR1_4$	$PCR1_3$	$PCR1_2$	$PCR1_1$	$PCR1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR1 is an 8-bit register for controlling whether each of the port 1 pins $P1_7$ and $P1_0$ functions as an input pin or output pin. Setting a PCR1 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR1 and in PDR1 are valid only when the corresponding pin is designated in PMR1 as a general I/O pin.

Upon reset, PCR1 is initialized to H'00.

PCR1 is a write-only register, which is always read as all 1s.

3. Port pull-up control register 1 (PUCR1)

Bit	7	6	5	4	3	2	1	0
	$PUCR1_7$	$PUCR1_6$	$PUCR1_5$	$PUCR1_4$	$PUCR1_3$	$PUCR1_2$	$PUCR1_1$	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PUCR1 controls whether the MOS pull-up of each of the port 1 pins $P1_7$ and $P1_0$ is on or off. When a PCR1 bit is cleared to 0, setting the corresponding PUCR1 bit to 1 turns on the MOS pull-up for the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR1 is initialized to H'00.

4. Port mode register 1 (PMR1)

Bit	7	6	5	4	3	2	1	0
	IRQ3	IRQ2	IRQ1	PWM	—	—	—	TMOW
Initial value	0	0	0	0	0	1	0	0
Read/Write	R/W	R/W	R/W	R/W	—	—	—	R/W

PMR1 is an 8-bit read/write register, controlling the selection of pin functions for port 1 pins.

Upon reset, PMR1 is initialized to H'04.

Bit 7: $P1_7/\overline{IRQ_3}$/TRGV pin function switch (IRQ3)

This bit selects whether pin $P1_7/\overline{IRQ_3}$/TRGV is used as $P1_7$ or as $\overline{IRQ_3}$/TRGV.

Bit 7 IRQ3	**Description**	
0	Functions as $P1_7$ I/O pin	(initial value)
1	Functions as $\overline{IRQ_3}$/TRGV input pin	

Note: Rising or falling edge sensing can be designated for $\overline{IRQ_3}$. Rising, falling, or both edge sensing can be designated for TRGV. For details on TRGV settings, see 9.8.2 (5), Timer Control Register V1 (TCRV1).

Bit 6: $P1_6/\overline{IRQ_2}$ pin function switch (IRQ2)

This bit selects whether pin $P1_6/\overline{IRQ_2}$ is used as $P1_6$ or as $\overline{IRQ_2}$.

Bit 6 IRQ2	Description	
0	Functions as $P1_6$ I/O pin	(initial value)
1	Functions as $\overline{IRQ_2}$ input pin	

Note: Rising or falling edge sensing can be designated for $\overline{IRQ_2}$.

Bit 5: $P1_5/\overline{IRQ_1}$ pin function switch (IRQ1)

This bit selects whether pin $P1_5/\overline{IRQ_1}$ is used as $P1_5$ or as $\overline{IRQ_1}$.

Bit 5 IRQ1	Description	
0	Functions as $P1_5$ I/O pin	(initial value)
1	Functions as $\overline{IRQ_1}$ input pin	

Note: Rising or falling edge sensing can be designated for $\overline{IRQ_1}$.

Bit 4: $P1_4$/PWM pin function switch (PWM)

This bit selects whether pin $P1_4$/PWM is used as $P1_4$ or as PWM.

Bit 4 PWM	Description	
0	Functions as $P1_4$ I/O pin	(initial value)
1	Functions as PWM output pin	

Bit 3: Reserved bit

Bit 3 is reserved: it is always read as 0 and cannot be modified.

Bit 2: Reserved bit

Bit 2 is reserved: it is always read as 1 and cannot be modified.

Bit 1: Reserved bit

Bit 1 is reserved: it is always read as 0 and cannot be modified.

Bit 0: $P1_0$/TMOW pin function switch (TMOW)

This bit selects whether pin $P1_0$/TMOW is used as $P1_0$ or as TMOW.

Bit 0 TMOW	Description	
0	Functions as $P1_0$ I/O pin	(initial value)
1	Functions as TMOW output pin	

8.2.3 Pin Functions

Table 8-3 shows the port 1 pin functions.

Table 8-3 Port 1 Pin Functions

Pin	Pin Functions and Selection Method
$P1_7$/$\overline{IRQ_3}$/TRGV	The pin function depends on bit IRQ3 in PMR1 and bit $PCR1_7$ in PCR1.

IRQ3	0		1
$PCR1_7$	0	1	*
Pin function	$P1_7$ input pin	$P1_7$ output pin	$\overline{IRQ_3}$/TRGV input pin

Pin	Pin Functions and Selection Method
$P1_6$/$\overline{IRQ_2}$ $P1_5$/$\overline{IRQ_1}$	The pin function depends on bits IRQ2 and IRQ1 in PMR1 and bit $PCR1_n$ in PCR1. (m = n – 4, n = 6, 5)

IRQm	0		1
$PCR1_n$	0	1	*
Pin function	$P1_n$ input pin	$P1_n$ output pin	$\overline{IRQ_m}$ input pin

Pin	Pin Functions and Selection Method
$P1_4$/PWM	The pin function depends on bit PWM in PMR1 and bit $PCR1_4$ in PCR1.

PWM	0		1
$PCR1_4$	0	1	*
Pin function	$P1_4$ input pin	$P1_4$ output pin	PWM output pin

Pin	Pin Functions and Selection Method
$P1_3$ to $P1_1$	The pin function depends on bit PCR1n in PCR1. (n = 3 to 1)

PCR1n	0	1
Pin function	$P1_n$ input pin	$P1_n$ output pin

Table 8-3 Port 1 Pin Functions (cont)

Pin	Pin Functions and Selection Method
$P1_0$/TMOW	The pin function depends on bit TMOW in PMR1 and bit $PCR1_0$ in PCR1.

TMOW	0		1
$PCR1_0$	0	1	*
Pin function	$P1_0$ input pin	$P1_0$ output pin	TMOW output pin

Note: * Don't care

8.2.4 Pin States

Table 8-4 shows the port 1 pin states in each operating mode.

Table 8-4 Port 1 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P1_7$/$\overline{IRQ_3}$/TRGV $P1_6$/$\overline{IRQ_2}$ $P1_5$/$\overline{IRQ_1}$ $P1_4$/PWM $P1_3$ to $P1_1$ $P1_0$/TMOW	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional

Note: * A high-level signal is output when the MOS pull-up is in the on state.

8.2.5 MOS Input Pull-Up

Port 1 has a built-in MOS input pull-up function that can be controlled by software. When a PCR1 bit is cleared to 0, setting the corresponding PUCR1 bit to 1 turns on the MOS input pull-up for that pin. The MOS input pull-up function is in the off state after a reset.

$PCR1_n$	0		1
$PUCR1_n$	0	1	*
MOS input pull-up	Off	On	Off

Note: * Don't care
n = 7 to 0

8.3 Port 2

8.3.1 Overview

Port 2 is a 8-bit I/O port, configured as shown in figure 8-2.

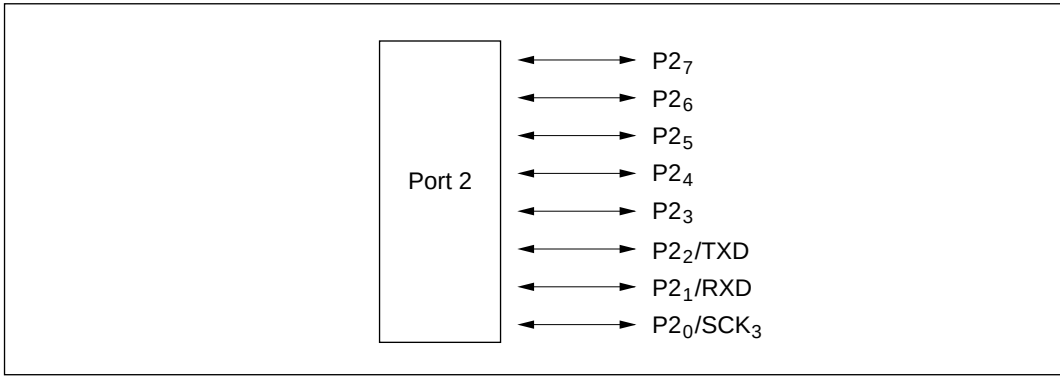


Figure 8-2 Port 2 Pin Configuration

8.3.2 Register Configuration and Description

Table 8-5 shows the port 2 register configuration.

Table 8-5 Port 2 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 2	PDR2	R/W	H'00	H'FFD5
Port control register 2	PCR2	W	H'00	H'FFE5

1. Port data register 2 (PDR2)

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

PDR2 is an 8-bit register that stores data for port 2 pins P2$_7$ to P2$_0$. If port 2 is read while PCR2 bits are set to 1, the values stored in PDR2 are read, regardless of the actual pin states. If port 2 is read while PCR2 bits are cleared to 0, the pin states are read.

Upon reset, PDR2 is initialized to H'00.

2 Port control register 2 (PCR2)

Bit	7	6	5	4	3	2	1	0
	$PCR2_7$	$PCR2_6$	$PCR2_5$	$PCR2_4$	$PCR2_3$	$PCR2_2$	$PCR2_1$	$PCR2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR2 is an 8-bit register for controlling whether each of the port 2 pins $P2_7$ to $P2_0$ functions as an input pin or output pin. Setting a PCR2 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR2 and PDR2 are valid only when the corresponding pin is designated in SCR3 as a general I/O pin.

Upon reset, PCR2 is initialized to H'00.

PCR2 is a write-only register, which is always read as all 1s.

8.3.3 Pin Functions

Table 8-6 shows the port 2 pin functions.

Table 8-6 Port 2 Pin Functions

Pin	Pin Functions and Selection Method
$P2_7$ to $P2_3$	The pin function depends on bit $PCR2_n$ in PCR2.

(n = 7 to 3)

$PCR2_n$	0	1
Pin function	$P2_n$ input pin	$P2_n$ output pin

$P2_2$/TXD — The pin function depends on bit TXD in PMR7 and bit $PCR2_2$ in PCR2.

TXD	0		1
$PCR2_2$	0	1	*
Pin function	$P2_2$ input pin	$P2_2$ output pin	TXD output pin

$P2_1$/RXD — The pin function depends on bit RE in SCR3 and bit $PCR2_1$ in PCR2.

RE	0		1
$PCR2_1$	0	1	*
Pin function	$P2_1$ input pin	$P2_1$ output pin	RXD input pin

$P2_0$/SCK_3 — The pin function depends on bits CKE1 and CKE0 in SCR3, bit COM in SMR, and bit $PCR2_0$ in PCR2.

CKE1	0			1
CKE0	0		1	*
COM	0		1	*	*
$PCR2_0$	0	1	*	*
Pin function	$P2_0$ input pin	$P2_0$ output pin	SCK_3 output pin	SCK_3 input pin

Note: * Don’t care

8.3.4 Pin States

Table 8-7 shows the port 2 pin states in each operating mode.

Table 8-7 Port 2 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P2_7$ to $P2_3$ $P2_2$/TXD $P2_1$/RXD $P2_0$/SCK_3	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

8.4 Port 3

8.4.1 Overview

Port 3 is a 6-bit I/O port, configured as shown in figure 8-3.

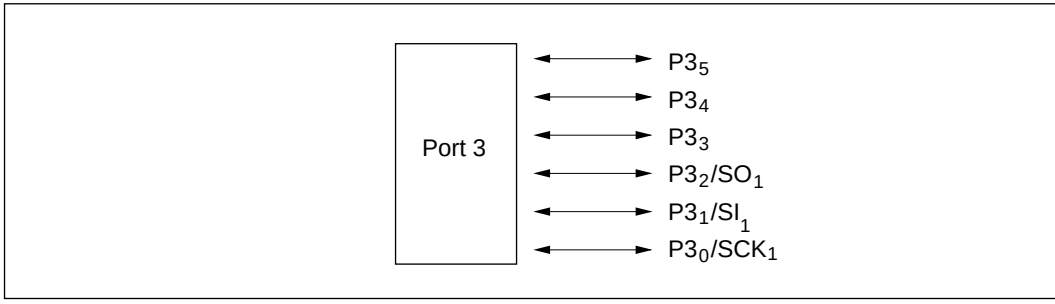


Figure 8-3 Port 3 Pin Configuration

8.4.2 Register Configuration and Description

Table 8-8 shows the port 3 register configuration.

Table 8-8 Port 3 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 3	PDR3	R/W	H'00	H'FFD6
Port control register 3	PCR3	W	H'00	H'FFE6
Port pull-up control register 3	PUCR3	R/W	H'00	H'FFEE
Port mode register 3	PMR3	R/W	H'00	H'FFFD
Port mode register 7	PMR7	R/W	H'F8	H'FFFF

1. Port data register 3 (PDR3)

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

Note: * Bits 7 to 3 are reserved; they are always read as 0 and cannot be modified.

PDR3 is an 8-bit register that stores data for port 3 pins $P3_5$ to $P3_0$. If port 3 is read while PCR3 bits are set to 1, the values stored in PDR3 are read, regardless of the actual pin states. If port 3 is read while PCR3 bits are cleared to 0, the pin states are read.

Upon reset, PDR3 is initialized to H'00.

2. Port control register 3 (PCR3)

Bit	7	6	5	4	3	2	1	0
	—	—	$PCR3_5$	$PCR3_4$	$PCR3_3$	$PCR3_2$	$PCR3_1$	$PCR3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

PCR3 is an 8-bit register for controlling whether each of the port 3 pins $P3_5$ to $P3_0$ functions as an input pin or output pin. Setting a PCR3 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin. The settings in PCR3 and in PDR3 are valid only when the corresponding pin is designated in PMR3 as a general I/O pin.

Upon reset, PCR3 is initialized to H'00.

PCR3 is a write-only register, which is always read as all 1s.

3. Port pull-up control register 3 (PUCR3)

Bit	7	6	5	4	3	2	1	0
	—	—	$PUCR3_5$	$PUCR3_4$	$PUCR3_3$	$PUCR3_2$	$PUCR3_1$	$PUCR3_0$
Initial value	0*	0*	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Bits 7 to 3 are reserved; they are always read as 0 and cannot be modified.

PUCR3 controls whether the MOS pull-up of each of the port 3 pins $P3_5$ to $P3_0$ is on or off. When a PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the MOS pull-up for the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR3 is initialized to H'00.

4. Port mode register 3 (PMR3)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SO1	SI1	SCK1
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

PMR3 is an 8-bit read/write register, controlling the selection of pin functions for port 3 pins.

Upon reset, PMR3 is initialized to H'00.

Bits 7 to 3: Reserved bits

Bits 7 to 3 are reserved: they are always read as 0 and cannot be modified.

Bit 2: $P3_2/SO_1$ pin function switch (SO1)

This bit selects whether pin $P3_2/SO_1$ is used as $P3_2$ or as SO_1.

Bit 2 SO1	**Description**	
0	Functions as $P3_2$ I/O pin	(initial value)
1	Functions as SO_1 output pin	

Bit 1: $P3_1/SI_1$ pin function switch (SI1)

This bit selects whether pin $P3_1/SI_1$ is used as $P3_1$ or as SI_1.

Bit 1 SI1	Description	
0	Functions as $P3_1$ I/O pin	(initial value)
1	Functions as SI_1 input pin	

Bit 0: $P3_0/SCK_1$ pin function switch (SCK1)

This bit selects whether pin $P3_0/SCK_1$ is used as $P3_0$ or as SCK_1.

Bit 0 SCK1	Description	
0	Functions as $P3_0$ I/O pin	(initial value)
1	Functions as SCK_1 I/O pin	

5. Port mode register 7 (PMR7)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TXD	—	POF1
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	—	R/W

PMR7 is an 8-bit read/write register that turns the PMOS transistors of pins and $P3_2/SO_1$ on and off.

Upon reset, PMR7 is initialized to H'F8.

Bits 7 to 3: Reserved bits

Bits 7 to 3 are reserved; they are always read as 1, and cannot be modified.

Bit 2: $P2_2$/TXD pin function switch (TXD)

Bit 2 selects whether pin $P2_2$/TXD is used as $P2_2$ or as TXD.

Bit 2 TXD	Description	
0	Functions as $P2_2$ I/O pin	(initial value)
1	Functions as TXD output pin	

Bit 1: Reserved bit

Bit 1 is reserved: it is always read as 0 and cannot be modified.

Bit 0: $P3_2/SO_1$ pin PMOS control (POF1)

This bit controls the PMOS transistor in the $P3_2/SO_1$ pin output buffer.

Bit 0 POF1	Description	
0	CMOS output	(initial value)
1	NMOS open-drain output	

8.4.3 Pin Functions

Table 8-9 shows the port 3 pin functions.

Table 8-9 Port 3 Pin Functions

Pin	Pin Functions and Selection Method
$P3_5$ to $P3_3$	The pin function depends on bit PCR3n in PCR3. (n = 5 to 3)

PCR3n	0	1
Pin function	$P3_n$ input pin	$P3_n$ output pin

$P3_2/SO_1$ — The pin function depends on bit SO1 in PMR3 and bit $PCR3_2$ in PCR3.

SO1	0		1
$PCR3_2$	0	1	*
Pin function	$P3_2$ input pin	$P3_2$ output pin	SO_1 output pin

$P3_1/SI_1$ — The pin function depends on bit SI1 in PMR3 and bit $PCR3_1$ in PCR3.

SI1	0		1
$PCR3_1$	0	1	*
Pin function	$P3_1$ input pin	$P3_1$ output pin	SI_1 input pin

Table 8-9 Port 3 Pin Functions (cont)

Pin	Pin Functions and Selection Method
$P3_0/SCK_1$	The pin function depends on bit SCK1 in PMR3, bit CKS3 in SCR1, and bit $PCR3_0$ in PCR3.

SCK1	0		1	
CKS3	*		0	1
$PCR3_0$	0	1	*	*
Pin function	$P3_0$ input pin	$P3_0$ output pin	SCK_1 output pin	SCK_1 input pin

Note: * Don't care

8.4.4 Pin States

Table 8-10 shows the port 3 pin states in each operating mode.

Table 8-10 Port 3 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P3_5$ to $P3_3$ $P3_2/SO_1$ $P3_1/SI_1$ $P3_0/SCK_1$	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional

Note: * A high-level signal is output when the MOS pull-up is in the on state.

8.4.5 MOS Input Pull-Up

Port 3 has a built-in MOS input pull-up function that can be controlled by software. When a PCR3 bit is cleared to 0, setting the corresponding PUCR3 bit to 1 turns on the MOS pull-up for that pin. The MOS pull-up function is in the off state after a reset.

$PCR3_n$	0		1
$PUCR3_n$	0	1	*
MOS input pull-up	Off	On	Off

Note: * Don't care (n = 5 to 0)

8.5 Port 5

8.5.1 Overview

Port 5 is an 8-bit I/O port, configured as shown in figure 8-4.

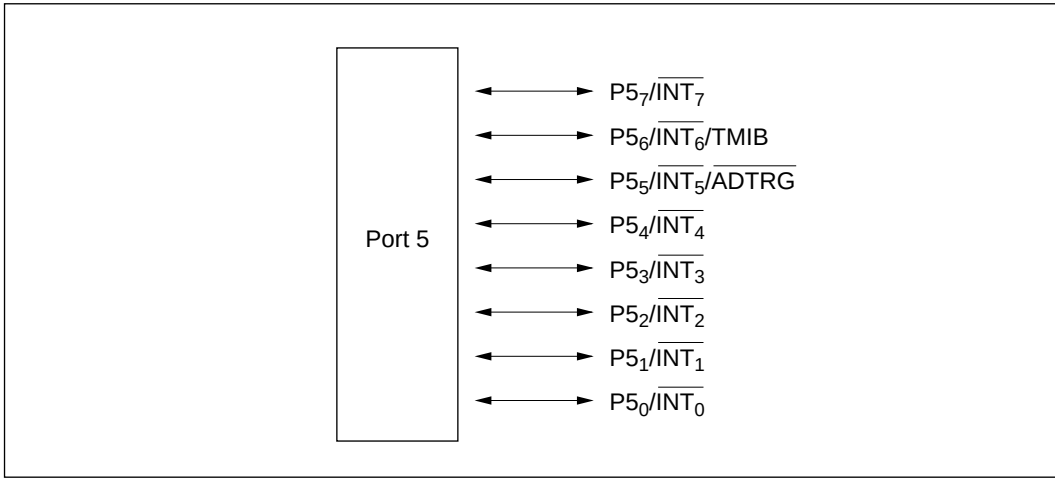


Figure 8-4 Port 5 Pin Configuration

8.5.2 Register Configuration and Description

Table 8-11 shows the port 5 register configuration.

Table 8-11 Port 5 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 5	PDR5	R/W	H'00	H'FFD8
Port control register 5	PCR5	W	H'00	H'FFE8
Port pull-up control register 5	PUCR5	R/W	H'00	H'FFEF

1. Port data register 5 (PDR5)

Bit	7	6	5	4	3	2	1	0
	$P5_7$	$P5_6$	$P5_5$	$P5_4$	$P5_3$	$P5_2$	$P5_1$	$P5_0$
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

PDR5 is an 8-bit register that stores data for port 5 pins $P5_7$ to $P5_0$. If port 5 is read while PCR5 bits are set to 1, the values stored in PDR5 are read, regardless of the actual pin states. If port 5 is read while PCR5 bits are cleared to 0, the pin states are read.

Upon reset, PDR5 is initialized to H'00.

2. Port control register 5 (PCR5)

Bit	7	6	5	4	3	2	1	0
	$PCR5_7$	$PCR5_6$	$PCR5_5$	$PCR5_4$	$PCR5_3$	$PCR5_2$	$PCR5_1$	$PCR5_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR5 is an 8-bit register for controlling whether each of the port 5 pins $P5_7$ to $P5_0$ functions as an input pin or output pin. Setting a PCR5 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin.

Upon reset, PCR5 is initialized to H'00.

PCR5 is a write-only register, which is always read as all 1s.

3. Port pull-up control register 5 (PUCR5)

Bit	7	6	5	4	3	2	1	0
	$PUCR5_7$	$PUCR5_6$	$PUCR5_5$	$PUCR5_4$	$PUCR5_3$	$PUCR5_2$	$PUCR5_1$	$PUCR5_0$
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

PUCR5 controls whether the MOS pull-up of each port 5 pin is on or off. When a PCR5 bit is cleared to 0, setting the corresponding PUCR5 bit to 1 turns on the MOS pull-up for the corresponding pin, while clearing the bit to 0 turns off the MOS pull-up.

Upon reset, PUCR5 is initialized to H'00.

8.5.3 Pin Functions

Table 8-12 shows the port 5 pin functions.

Table 8-12 Port 5 Pin Functions

Pin	Pin Functions and Selection Method
$P5_7/\overline{INT_7}$	The pin function depends on bit $PCR5_7$ in PCR5.

$PCR5_7$	0	1
Pin function	$P5_7$ input pin	$P5_7$ output pin
	$\overline{INT_7}$ input pin	

$P5_6/\overline{INT_6}$/TMIB	The pin function depends on bit $PCR5_6$ in PCR5.

$PCR5_6$	0	1
Pin function	$P5_6$ input pin	$P5_6$ output pin
	$\overline{INT_6}$ input pin and TMIB input pin	

$P5_5/\overline{INT_5}/\overline{ADTRG}$	The pin function depends on bit $PCR5_5$ in PCR5.

$PCR5_5$	0	1
Pin function	$P5_5$ input pin	$P5_5$ output pin
	$\overline{INT_5}$ input pin and $\overline{ADTRG}$ input pin	

$P5_4/\overline{INT_4}$ to $P5_0/\overline{INT_0}$	The pin function depends on bit $PCR5_n$ in PCR5. (n = 4 to 0)

$PCR5_n$	0	1
Pin function	$P5_n$ input pin	$P5_n$ output pin
	$\overline{INT_n}$ input pin	

8.5.4 Pin States

Table 8-13 shows the port 5 pin states in each operating mode.

Table 8-13 Port 5 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P5_7/\overline{INT_7}$ to $P5_0/\overline{INT_0}$	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional

Note: * A high-level signal is output when the MOS pull-up is in the on state.

8.5.5 MOS Input Pull-Up

Port 5 has a built-in MOS input pull-up function that can be controlled by software. When a PCR5 bit is cleared to 0, setting the corresponding PUCR5 bit to 1 turns on the MOS pull-up for that pin. The MOS pull-up function is in the off state after a reset.

$PCR5_n$	0		1
$PUCR5_n$	0	1	*
MOS input pull-up	Off	On	Off

Note: * Don't care (n = 7 to 0)

8.6 Port 6

8.6.1 Overview

Port 6 is an 8-bit I/O port. The port 6 pin configuration is shown in figure 8-5.

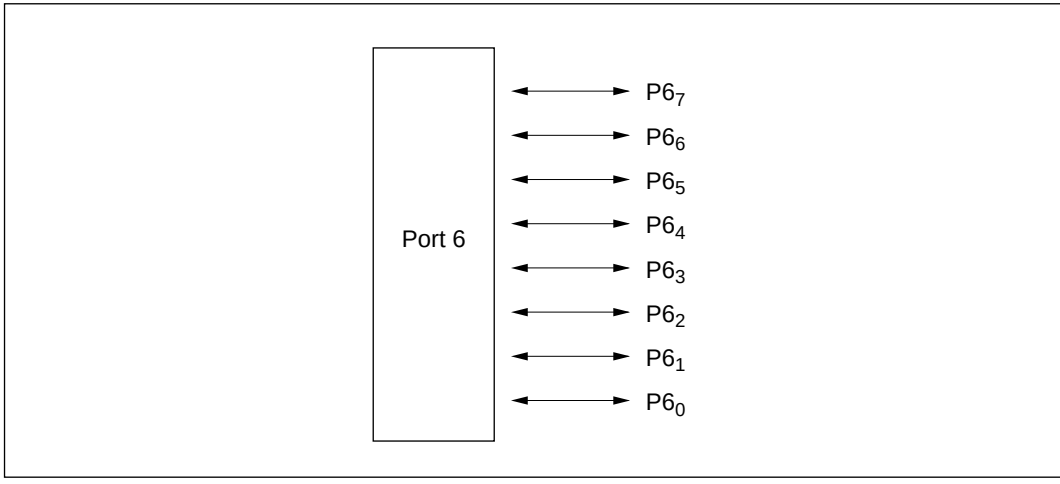


Figure 8-5 Port 6 Pin Configuration

8.6.2 Register Configuration and Description

Table 8-14 shows the port 6 register configuration.

Table 8-14 Port 6 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 6	PDR6	R/W	H'00	H'FFD9
Port control register 6	PCR6	W	H'00	H'FFE9

1. Port data register 6 (PDR6)

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

PDR6 is an 8-bit register that stores data for port 6 pins $P6_7$ to $P6_0$.

Upon reset, PDR6 is initialized to H'00.

2. Port control register 6 (PCR6)

Bit	7	6	5	4	3	2	1	0
	$PCR6_7$	$PCR6_6$	$PCR6_5$	$PCR6_4$	$PCR6_3$	$PCR6_2$	$PCR6_1$	$PCR6_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR6 is an 8-bit register for controlling whether each of the port 6 pins $P6_7$ to $P6_0$ functions as an input pin or output pin.

Upon reset, PCR6 is initialized to H'00.

PCR6 is a write-only register, which always reads all 1s.

8.6.3 Pin Functions

Table 8-15 shows the port 6 pin functions.

Table 8-15 Port 6 Pin Functions

Pin	Pin Functions and Selection Method
$P6_7$ to $P6_0$	The pin function depends on bit $PCR6_n$ in PCR6 (n = 7 to 0)

$PCR6_n$	0	1
Pin function	$P6_n$ input pin	$P6_n$ output pin

8.6.4 Pin States

Table 8-16 shows the port 6 pin states in each operating mode.

Table 8-16 Port 6 Pin States

Pin	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P6_7$ to$P6_0$	High-impedance	Retains previous state	Retains previous state	High-impedance*	Retains previous state	Functional	Functional

Note: * A high-level signal is output when the MOS pull-up is in the on state.

8.7 Port 7

8.7.1 Overview

Port 7 is an 8-bit I/O port, configured as shown in figure 8-6.

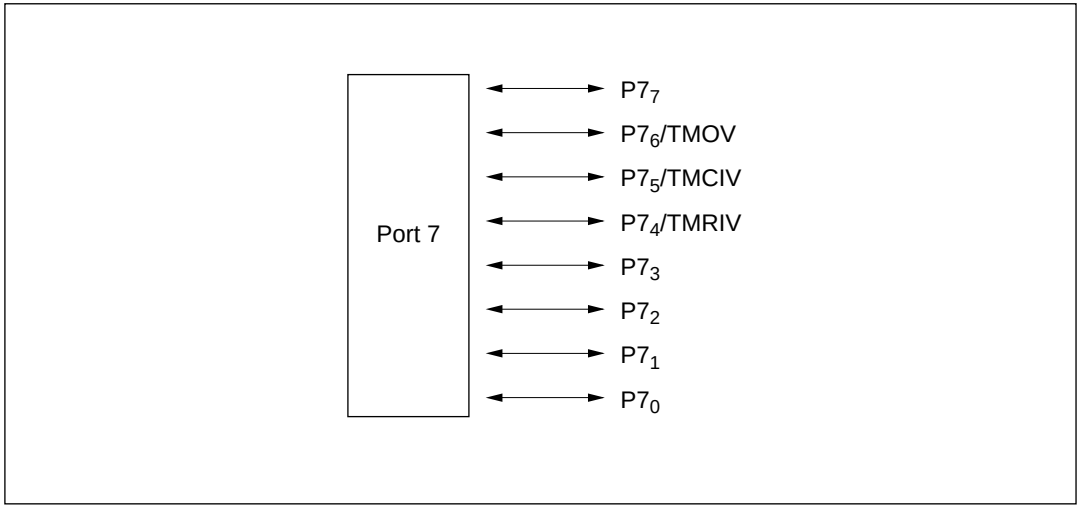


Figure 8-6 Port 7 Pin Configuration

8.7.2 Register Configuration and Description

Table 8-17 shows the port 7 register configuration.

Table 8-17 Port 7 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 7	PDR7	R/W	H'00	H'FFDA
Port control register 7	PCR7	W	H'00	H'FFEA

1. Port data register 7 (PDR7)

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	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR7 is an 8-bit register that stores data for port 7 pins $P7_7$ to $P7_0$. If port 7 is read while PCR7 bits are set to 1, the values stored in PDR7 are read, regardless of the actual pin states. If port 7 is read while PCR7 bits are cleared to 0, the pin states are read.

Upon reset, PDR7 is initialized to H'00.

2. Port control register 7 (PCR7)

Bit	7	6	5	4	3	2	1	0
	$PCR7_7$	$PCR7_6$	$PCR7_5$	$PCR7_4$	$PCR7_3$	$PCR7_2$	$PCR7_1$	$PCR7_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR7 is an 8-bit register for controlling whether each of the port 7 pins $P7_7$ to $P7_0$ functions as an input pin or output pin. Setting a PCR7 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin.

Upon reset, PCR7 is initialized to H'00.

PCR7 is a write-only register, which always reads as all 1s.

8.7.3 Pin Functions

Table 8-18 shows the port 7 pin functions.

Table 8-18 Port 7 Pin Functions

Pin	Pin Functions and Selection Method
$P7_7$, $P7_3$ to $P7_0$	The pin function depends on bit $PCR7_n$ in PCR7. (n = 7 or 3 to 0) $PCR7_n$ = 0: Pin function = $P7_n$ input pin $PCR7_n$ = 1: Pin function = $P7_n$ output pin
$P7_6$/TMOV	The pin function depends on bit $PCR7_6$ in PCR7 and bits OS3 to OS0 in TCSRV. OS3 to OS0 = 0000, $PCR7_6$ = 0: Pin function = $P7_6$ input pin OS3 to OS0 = 0000, $PCR7_6$ = 1: Pin function = $P7_6$ output pin OS3 to OS0 = Not 0000, $PCR7_6$ = *: Pin function = TMOV output pin
$P7_5$/TMCIV	The pin function depends on bit $PCR7_5$ in PCR7. $PCR7_5$ = 0: Pin function = $P7_5$ input pin $PCR7_5$ = 1: Pin function = $P7_5$ output pin Pin function = TMCIV input pin
$P7_4$/TMRIV	The pin function depends on bit $PCR7_4$ in PCR7. $PCR7_4$ = 0: Pin function = $P7_4$ input pin $PCR7_4$ = 1: Pin function = $P7_4$ output pin Pin function = TMRIV input pin

Note: * Don't care

8.7.4 Pin States

Table 8-19 shows the port 7 pin states in each operating mode.

Table 8-19 Port 7 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P7_7$ $P7_6$/TMOV $P7_5$/TMCIV $P7_4$/TMRIV $P7_3$ to $P7_0$	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

8.8 Port 8

8.8.1 Overview

Port 8 is an 8-bit I/O port configured as shown in figure 8-7.

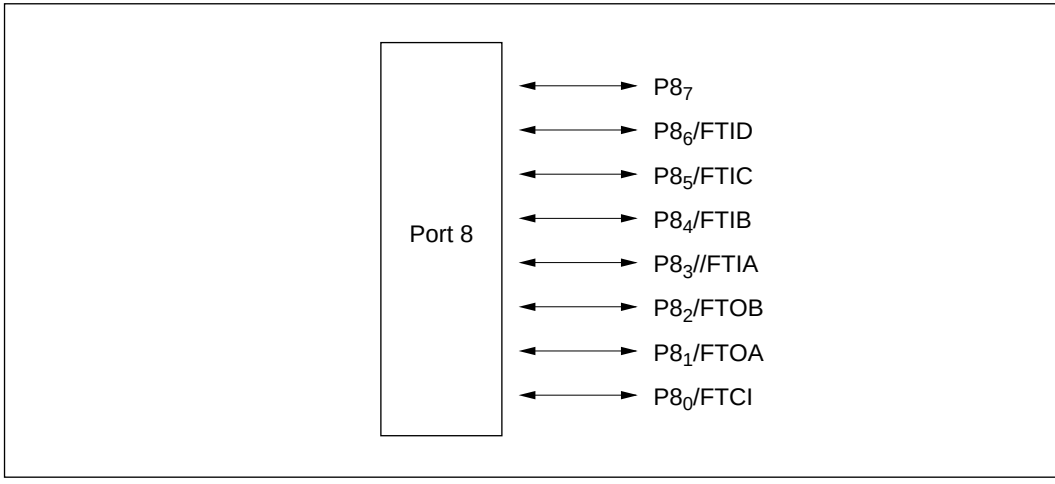


Figure 8-7 Port 8 Pin Configuration

8.8.2 Register Configuration and Description

Table 8-20 shows the port 8 register configuration.

Table 8-20 Port 8 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 8	PDR8	R/W	H'00	H'FFDB
Port control register 8	PCR8	W	H'00	H'FFEB

1. Port data register 8 (PDR8)

Bit	7	6	5	4	3	2	1	0
	$P8_7$	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$
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

PDR8 is an 8-bit register that stores data for port 8 pins $P8_7$ to $P8_0$. If port 8 is read while PCR8 bits are set to 1, the values stored in PDR8 are read, regardless of the actual pin states. If port 8 is read while PCR8 bits are cleared to 0, the pin states are read.

Upon reset, PDR8 is initialized to H'00.

2. Port control register 8 (PCR8)

Bit	7	6	5	4	3	2	1	0
	$PCR8_7$	$PCR8_6$	$PCR8_5$	$PCR8_4$	$PCR8_3$	$PCR8_2$	$PCR8_1$	$PCR8_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PCR8 is an 8-bit register for controlling whether each of the port 8 pins $P8_7$ to $P8_0$ functions as an input or output pin. Setting a PCR8 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin.

Upon reset, PCR8 is initialized to H'00.

PCR8 is a write-only register, which is always read as all 1s.

8.8.3 Pin Functions

Table 8-24 shows the port 8 pin functions.

Table 8-21 Port 8 Pin Functions

Pin	Pin Functions and Selection Method

$P8_7$ — The pin function depends on bit $PCR8_7$ in PCR8.

$PCR8_7$	0	1
Pin function	$P8_7$ input pin	$P8_7$ output pin

$P8_6$/FTID — The pin function depends on bit $PCR8_6$ in PCR8.

$PCR8_6$	0	1
Pin function	$P8_6$ input pin	$P8_6$ output pin
	FTID input pin	

$P8_5$/FTIC — The pin function depends on bit $PCR8_5$ in PCR8.

$PCR8_5$	0	1
Pin function	$P8_5$ input pin	$P8_5$ output pin
	FTIC input pin	

$P8_4$/FTIB — The pin function depends on bit $PCR8_4$ in PCR8.

$P8_4$	0	1
Pin function	$P8_4$ input pin	$P8_4$ output pin
	FTIB input pin	

$P8_3$/FTIA — The pin function depends on bit $PCR8_3$ in PCR8.

PCR83	0	1
Pin function	$P8_3$ input pin	$P8_3$ output pin
	FTIA input pin	

$P8_2$/FTOB — The pin function depends on bit $PCR8_2$ in PCR8 and bit OEB in TOCR.

OEB	0		1
$PCR8_2$	0	1	*
Pin function	$P8_2$ input pin	$P8_2$ output pin	FTOB output pin

Note: * Don’t care

Table 8-21 Port 8 Pin Functions (cont)

Pin	Pin Functions and Selection Method
$P8_1$/FTOA	The pin function depends on bit $PCR8_1$ in PCR8 and bit OEA in TOCR.

OEA	0		1
$PCR8_1$	0	1	*
Pin function	$P8_1$ input pin	$P8_1$ output pin	FTOA output pin

$P8_0$/FTCI	The pin function depends on bit $PCR8_0$ in PCR8.

$PCR8_0$	0	1
Pin function	$P8_0$ input pin	$P8_0$ output pin
	FTCI input pin	

Note: * Don't care

8.8.4 Pin States

Table 8-22 shows the port 8 pin states in each operating mode.

Table 8-22 Port 8 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P8_7$ $P8_6$/FTID $P8_5$/FTIC $P8_4$/FTIB $P8_3$/FTIA $P8_2$/FTOB $P8_1$/FTOA $P8_0$/FTCI	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

8.9 Port 9

8.9.1 Overview

Port 9 is a 5-bit I/O port, configured as shown in figure 8-8.

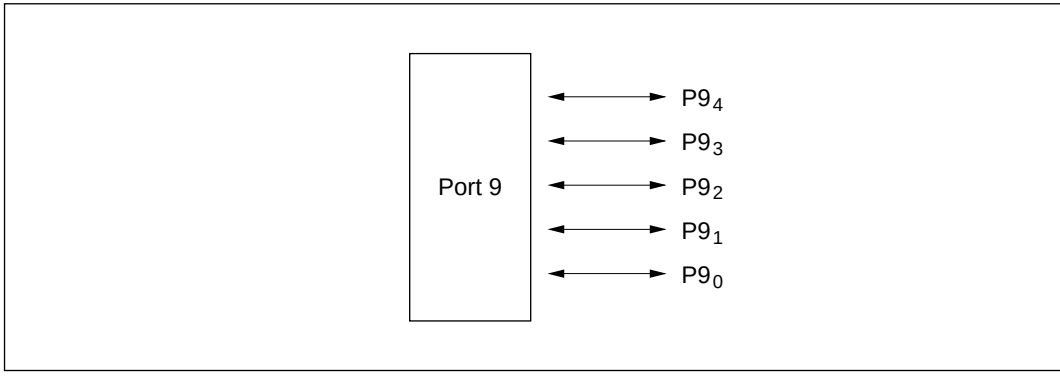


Figure 8-8 Port 9 Pin Configuration

8.9.2 Register Configuration and Description

Table 8-23 shows the port 9 register configuration.

Table 8-23 Port 9 Registers

Name	Abbrev.	R/W	Initial Value	Address
Port data register 9	PDR9	R/W	H'C0	H'FFDC
Port control register 9	PCR9	W	H'C0	H'FFEC

1. Port data register 9 (PDR9)

Bit	7	6	5	4	3	2	1	0
	—	—	—	P9$_4$	P9$_3$	P9$_2$	P9$_1$	P9$_0$
Initial value	1*	1*	0**	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Note: * Bits 7 to 6 are reserved; they are always read as 1 and cannot be modified.

** Bit 5 is reserved; it is always read as 0 and cannot be modified.

PDR9 is an 8-bit register that stores data for port 9 pins P9$_4$ to P9$_0$. If port 9 is read while PCR9 bits are set to 1, the values stored in PDR9 are read, regardless of the actual pin states. If port 9 is read while PCR9 bits are cleared to 0, the pin states are read.

Upon reset, PDR9 is initialized to H'C0.

2. Port control register 9 (PCR9)

Bit	7	6	5	4	3	2	1	0
	—	—	—	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	—	W	W	W	W	W

PCR9 controls whether each of the port 9 pins $P9_4$ to $P9_0$ functions as an input pin or output pin. Setting a PCR9 bit to 1 makes the corresponding pin an output pin, while clearing the bit to 0 makes the pin an input pin.

Upon reset, PCR9 is initialized to H'C0.

PCR9 is a write-only register, which is always reads as all 1.

8.9.3 Pin Functions

Table 8-24 shows the port 9 pin functions.

Table 8-24 Port 9 Pin Functions

Pin	Pin Functions and Selection Method
$P9_n$	The pin function depends on bit $PCR9_n$ in PCR9. (n = 4 to 0)

$PCR9_n$	0	1
Pin function	$P9_n$ input pin	$P9_n$ output pin

8.9.4 Pin States

Table 8-25 shows the port 9 pin states in each operating mode.

Table 8-25 Port 9 Pin States

Pins	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P9_4$ to $P9_0$	High-impedance	Retains previous state	Retains previous state	High-impedance	Retains previous state	Functional	Functional

8.10 Port B

8.10.1 Overview

Port B is an 8-bit input-only port, configured as shown in figure 8-9.

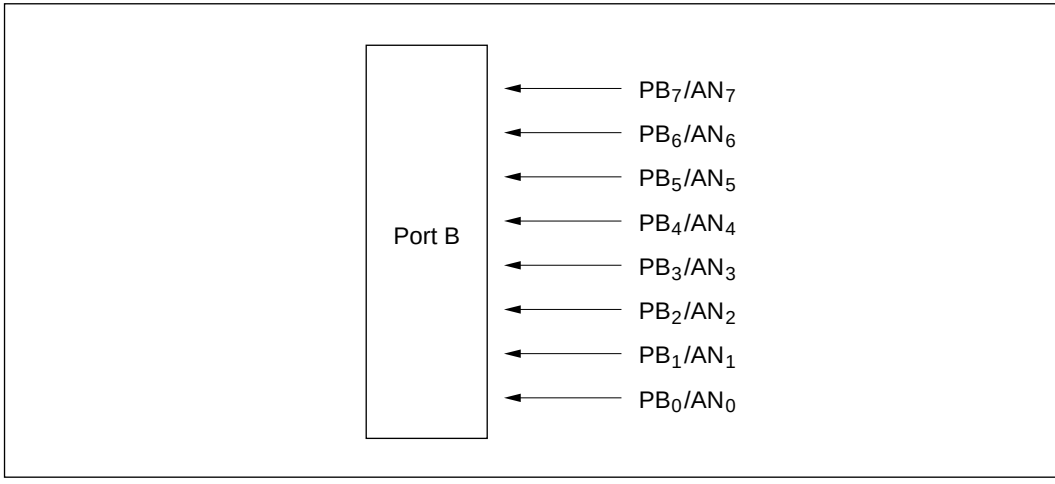


Figure 8-9 Port B Pin Configuration

8.10.2 Register Configuration and Description

Table 8-26 shows the port B register configuration.

Table 8-26 Port B Register

Name	Abbrev.	R/W	Address
Port data register B	PDRB	R	H'FFDD

Port Data Register B (PDRB)

Bit	7	6	5	4	3	2	1	0
	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
Read/Write	R	R	R	R	R	R	R	R

Reading PDRB always gives the pin states. However, if a port B pin is selected as an analog input channel for the A/D converter by AMR bits CH3 to CH0, that pin reads 0 regardless of the input voltage.

8.10.3 Pin Functions

Table 8-27 shows the port B pin functions.

Table 8-27 Port B Pin Functions

Pin	Pin Functions and Selection Method
PB_n/AN_n	Always as below. (n = 7 to 0) Pin function: PB_n input pin or AN_n input pin

8.10.4 Pin States

Table 8-28 shows the port B pin states in each operating mode.

Table 8-28 Port B Pin States

Pin	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
PB_7/AN_7 to PB_0/AN_0	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance	High-impedance

8.11 Usage Notes

The following points should be noted when using I/O ports.

(1) Handling of unused pins in input ports
Unused pins in an input port should be connected to the power supply (V_{CC} or V_{SS}) so that the pins do not go to the floating (high-impedance) state.

(2) Handling of unused pins in input/output ports
Unused pins in an input/output port should be connected to the power supply (V_{CC} or V_{SS}), or else designated as output pins by setting the corresponding port control register bits by software immediately after reset release, so that the pins do not go to the floating (high-impedance) state.

Section 9 Timers

9.1 Overview

The H8/3657 Series provides five timers: timers A, B1, V, X, and a watchdog timer. The functions of these timers are outlined in table 9-1.

Table 9-1 Timer Functions

Name	Functions	Internal Clock	Event Input Pin	Waveform Output Pin	Remarks
Timer A	• 8-bit interval timer	ø/8 to ø/8192 (8 choices)	—	—	
	• Time base	$ø_W$/128 (choice of 4 overflow periods)			
	• Clock output	ø/4 to ø/32 $ø_W$/4 to $ø_W$/32 (8 choices)	—	TMOW	
Timer B1	• 8-bit reload timer • Interval timer • Event counter	ø/4 to ø/8192 (7 choices)	TMIB	—	
Timer V	• 8-bit timer • Event counter • Output control by dual compare match • Counter clearing option • Start of incrementing specifiable by external trigger input	ø/4 to ø/128 (6 choices)	TMCIV	TMOV	
Timer X	• 16-bit free-running timer • 2 output compare channels • 4 input capture channels • Counter clearing option • Event counter	ø/2 to ø/32 (3 choices)	FTCI FTIA FTIB FTIC FTID	FTOA FTOB	
Watchdog timer	• Reset signal generated when 8-bit counter overflows	ø/8192	—	—	

9.2 Timer A

9.2.1 Overview

Timer A is an 8-bit timer with interval timing and real-time clock time-base functions. The clock time-base function is available when a 32.768-kHz crystal oscillator is connected. A clock signal divided from 32.768 kHz or from the system clock can be output at the TMOW pin.

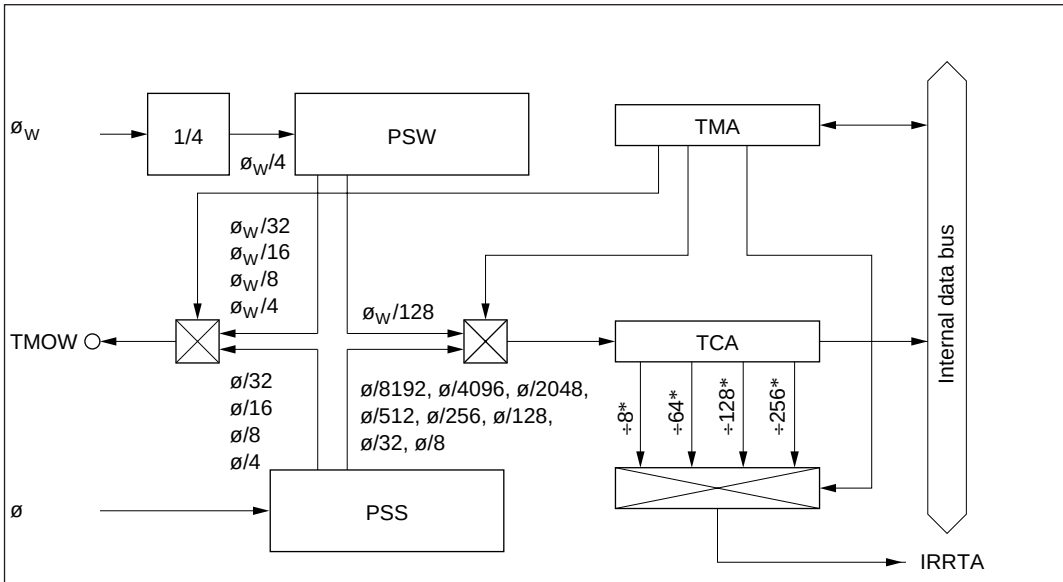
1. Features

Features of timer A are given below.

- Choice of eight internal clock sources (ø/8192, ø/4096, ø/2048, ø/512, ø/256, ø/128, ø/32, ø/8).
- Choice of four overflow periods (1 s, 0.5 s, 0.25 s, 31.25 ms) when timer A is used as a clock time base (using a 32.768 kHz crystal oscillator).
- An interrupt is requested when the counter overflows.
- Any of eight clock signals can be output from pin TMOW: 32.768 kHz divided by 32, 16, 8, or 4 (1 kHz, 2 kHz, 4 kHz, 8 kHz), or the system clock divided by 32, 16, 8, or 4.

2. Block diagram

Figure 9-1 shows a block diagram of timer A.



Notation:

TMA: Timer mode register A
TCA: Timer counter A
IRRTA: Timer A overflow interrupt request flag
PSW: Prescaler W
PSS: Prescaler S

Note: * Can be selected only when the prescaler W output (øW/128) is used as the TCA input clock.

Figure 9-1 Block Diagram of Timer A

3. Pin configuration

Table 9-2 shows the timer A pin configuration.

Table 9-2 Pin Configuration

Name	Abbrev.	I/O	Function
Clock output	TMOW	Output	Output of waveform generated by timer A output circuit

4. Register configuration

Table 9-3 shows the register configuration of timer A.

Table 9-3 Timer A Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register A	TMA	R/W	H'10	H'FFB0
Timer counter A	TCA	R	H'00	H'FFB1

9.2.2 Register Descriptions

1. Timer mode register A (TMA)

Bit	7	6	5	4	3	2	1	0
	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0
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

TMA is an 8-bit read/write register for selecting the prescaler, input clock, and output clock.

Upon reset, TMA is initialized to H'10.

Bits 7 to 5: Clock output select (TMA7 to TMA5)

Bits 7 to 5 choose which of eight clock signals is output at the TMOW pin. The system clock divided by 32, 16, 8, or 4 can be output in active mode and sleep mode. A 32.768 kHz signal divided by 32, 16, 8, or 4 can be output in active mode, sleep mode, and subactive mode.

Bit 7 TMA7	Bit 6 TMA6	Bit 5 TMA5	Clock Output	
0	0	0	ø/32	(initial value)
		1	ø/16	
	1	0	ø/8	
		1	ø/4	
1	0	0	$ø_W$/32	
		1	$ø_W$/16	
	1	0	$ø_W$/8	
		1	$ø_W$/4	

Bit 4: Reserved bit

Bit 4 is reserved; it is always read as 1, and cannot be modified.

Bits 3 to 0: Internal clock select (TMA3 to TMA0)

Bits 3 to 0 select the clock input to TCA. The selection is made as follows.

				Description		
Bit 3 TMA3	**Bit 2 TMA2**	**Bit 1 TMA1**	**Bit 0 TMA0**	**Prescaler and Divider Ratio or Overflow Period**		**Function**
0	0	0	0	PSS, ø/8192	(initial value)	Interval timer
			1	PSS, ø/4096		
		1	0	PSS, ø/2048		
			1	PSS, ø/512		
	1	0	0	PSS, ø/256		
			1	PSS, ø/128		
		1	0	PSS, ø/32		
			1	PSS, ø/8		
1	0	0	0	PSW, 1 s		Clock time base
			1	PSW, 0.5 s		
		1	0	PSW, 0.25 s		
			1	PSW, 0.03125 s		
	1	0	0	PSW and TCA are reset		
			1			
		1	0			
			1			

2. Timer counter A (TCA)

Bit	7	6	5	4	3	2	1	0
	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TCA is an 8-bit read-only up-counter, which is incremented by internal clock input. The clock source for input to this counter is selected by bits TMA3 to TMA0 in timer mode register A (TMA). TCA values can be read by the CPU in active mode, but cannot be read in subactive mode. When TCA overflows, the IRRTA bit in interrupt request register 1 (IRR1) is set to 1.

TCA is cleared by setting bits TMA3 and TMA2 of TMA to 11.

Upon reset, TCA is initialized to H'00.

9.2.3 Timer Operation

1. Interval timer operation

When bit TMA3 in timer mode register A (TMA) is cleared to 0, timer A functions as an 8-bit interval timer.

Upon reset, TCA is cleared to H'00 and bit TMA3 is cleared to 0, so up-counting and interval timing resume immediately. The clock input to timer A is selected by bits TMA2 to TMA0 in TMA; any of eight internal clock signals output by prescaler S can be selected.

After the count value in TCA reaches H'FF, the next clock signal input causes timer A to overflow, setting bit IRRTA to 1 in interrupt request register 1 (IRR1). If IENTA = 1 in interrupt enable register 1 (IENR1), a CPU interrupt is requested.*

At overflow, TCA returns to H'00 and starts counting up again. In this mode timer A functions as an interval timer that generates an overflow output at intervals of 256 input clock pulses.

Note: * For details on interrupts, see 3.3, Interrupts.

2. Real-time clock time base operation

When bit TMA3 in TMA is set to 1, timer A functions as a real-time clock time base by counting clock signals output by prescaler W. The overflow period of timer A is set by bits TMA1 and TMA0 in TMA. A choice of four periods is available. In time base operation (TMA3 = 1), setting bit TMA2 to 1 clears both TCA and prescaler W to their initial values of H'00.

3. Clock output

Setting bit TMOW in port mode register 1 (PMR1) to 1 causes a clock signal to be output at pin TMOW. Eight different clock output signals can be selected by means of bits TMA7 to TMA5 in TMA. The system clock divided by 32, 16, 8, or 4 can be output in active mode and sleep mode. A 32.768 kHz signal divided by 32, 16, 8, or 4 can be output in active mode, sleep mode, subactive mode, and subsleep mode.

9.2.4 Timer A Operation States

Table 9-4 summarizes the timer A operation states.

Table 9-4 Timer A Operation States

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCA	Interval	Reset	Functions	Functions	Halted	Halted	Halted	Halted
	Clock time base	Reset	Functions	Functions	Functions	Functions	Functions	Halted
TMA		Reset	Functions	Retained	Retained	Functions	Retained	Retained

Note: When the real-time clock time base function is selected as the internal clock of TCA in active mode or sleep mode, the internal clock is not synchronous with the system clock, so it is synchronized by a synchronizing circuit. This may result in a maximum error of 1/ø (s) in the count cycle.

9.3 Timer B1

9.3.1 Overview

Timer B1 is an 8-bit timer that increments each time a clock pulse is input. This timer has two operation modes, interval and auto reload.

1. Features

Features of timer B1 are given below.

- Choice of seven internal clock sources (ø/8192, ø/2048, ø/512, ø/256, ø/64, ø/16, ø/4) or an external clock (can be used to count external events).
- An interrupt is requested when the counter overflows.

2. Block diagram

Figure 9-2 shows a block diagram of timer B1.

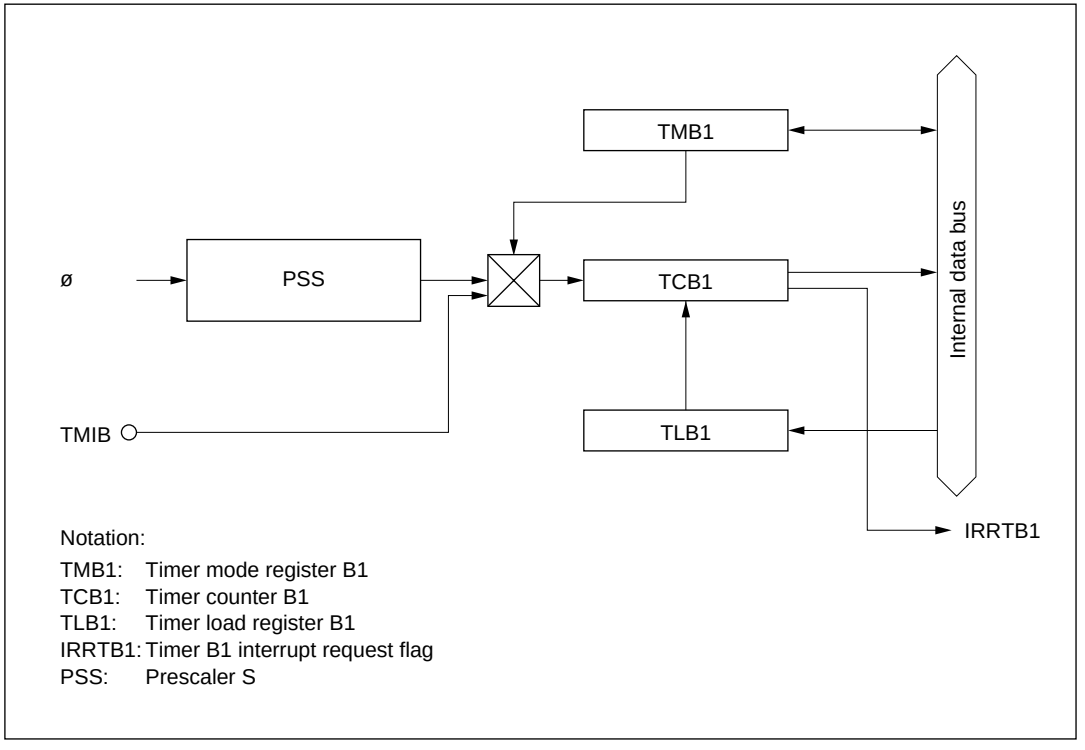


Figure 9-2 Block Diagram of Timer B1

3. Pin configuration

Table 9-5 shows the timer B1 pin configuration.

Table 9-5 Pin Configuration

Name	Abbrev.	I/O	Function
Timer B1 event input	TMIB	Input	Event input to TCB1

4. Register configuration

Table 9-6 shows the register configuration of timer B1.

Table 9-6 Timer B1 Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer mode register B1	TMB1	R/W	H'78	H'FFB2
Timer counter B1	TCB1	R	H'00	H'FFB3
Timer load register B1	TLB1	W	H'00	H'FFB3

9.3.2 Register Descriptions

1. Timer mode register B1 (TMB1)

Bit	7	6	5	4	3	2	1	0
	TMB17	—	—	—	—	TMB12	TMB11	TMB10
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

TMB1 is an 8-bit read/write register for selecting the auto-reload function and input clock.

Upon reset, TMB1 is initialized to H'78.

Bit 7: Auto-reload function select (TMB17)

Bit 7 selects whether timer B1 is used as an auto-reload timer.

Bit 7 TMB17	Description	
0	Interval timer function selected	(initial value)
1	Auto-reload function selected	

Bits 6 to 3: Reserved bits

Bits 6 to 3 are reserved; they are always read as 1, and cannot be modified.

Bits 2 to 0: Clock select (TMB12 to TMB10)

Bits 2 to 0 select the clock input to TCB1. For external event counting, either the rising or falling edge can be selected.

Bit 2 TMB12	Bit 1 TMB11	Bit 0 TMB10	Description	
0	0	0	Internal clock: ø/8192	(initial value)
0	0	1	Internal clock: ø/2048	
0	1	0	Internal clock: ø/512	
0	1	1	Internal clock: ø/256	
1	0	0	Internal clock: ø/64	
1	0	1	Internal clock: ø/16	
1	1	0	Internal clock: ø/4	
1	1	1	External event (TMIB): rising or falling edge*	

Note: * The edge of the external event signal is selected by bit INTEG6 in interrupt edge select register 2 (IEGR2). See 3.3.2, Interrupt Control Registers, for details.

2. Timer counter B1 (TCB1)

Bit	7	6	5	4	3	2	1	0
	TCB17	TCB16	TCB15	TCB14	TCB13	TCB12	TCB11	TCB10
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TCB1 is an 8-bit read-only up-counter, which is incremented by internal clock or external event input. The clock source for input to this counter is selected by bits TMB12 to TMB10 in timer mode register B1 (TMB1). TCB1 values can be read by the CPU at any time.

When TCB1 overflows from H'FF to H'00 or to the value set in TLB1, the IRRTB1 bit in IRR1 is set to 1.

TCB1 is allocated to the same address as TLB1.

Upon reset, TCB1 is initialized to H'00.

3. Timer load register B1 (TLB1)

Bit	7	6	5	4	3	2	1	0
	TLB17	TLB16	TLB15	TLB14	TLB13	TLB12	TLB11	TLB10
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

TLB1 is an 8-bit write-only register for setting the reload value of timer counter B1 (TCB1).

When a reload value is set in TLB1, the same value is loaded into timer counter B1 (TCB1) as well, and TCB1 starts counting up from that value. When TCB1 overflows during operation in auto-reload mode, the TLB1 value is loaded into TCB1. Accordingly, overflow periods can be set within the range of 1 to 256 input clocks.

The same address is allocated to TLB1 as to TCB1.

Upon reset, TLB1 is initialized to H'00.

9.3.3 Timer Operation

1. Interval timer operation

When bit TMB17 in timer mode register B1 (TMB1) is cleared to 0, timer B1 functions as an 8-bit interval timer.

Upon reset, TCB1 is cleared to H'00 and bit TMB17 is cleared to 0, so up-counting and interval timing resume immediately. The clock input to timer B1 is selected from seven internal clock signals output by prescaler S, or an external clock input at pin TMIB. The selection is made by bits TMB12 to TMB10 of TMB1.

After the count value in TCB1 reaches H'FF, the next clock signal input causes timer B1 to overflow, setting bit IRRTB1 to 1 in interrupt request register 1 (IRR1). If IENTB1 = 1 in interrupt enable register 1 (IENR1), a CPU interrupt is requested.*

At overflow, TCB1 returns to H'00 and starts counting up again.

During interval timer operation (TMB17 = 0), when a value is set in timer load register B1 (TLB1), the same value is set in TCB1.

Note: * For details on interrupts, see 3.3, Interrupts.

2. Auto-reload timer operation

Setting bit TMB17 in TMB1 to 1 causes timer B1 to function as an 8-bit auto-reload timer. When a reload value is set in TLB1, the same value is loaded into TCB1, becoming the value from which TCB1 starts its count.

After the count value in TCB1 reaches H'FF, the next clock signal input causes timer B1 to overflow. The TLB1 value is then loaded into TCB1, and the count continues from that value. The overflow period can be set within a range from 1 to 256 input clocks, depending on the TLB1 value.

The clock sources and interrupts in auto-reload mode are the same as in interval mode.

In auto-reload mode (TMB17 = 1), when a new value is set in TLB1, the TLB1 value is also set in TCB1.

3. Event counter operation

Timer B1 can operate as an event counter, counting rising or falling edges of an external event signal input at pin TMIB. External event counting is selected by setting bits TMB12 to TMB10 in timer mode register B1 to all 1s (111).

When timer B1 is used to count external event input, bit INTEN6 in IENR3 should be cleared to 0 to disable INT_6 interrupt requests.

9.3.4 Timer B1 Operation States

Table 9-7 summarizes the timer B1 operation states.

Table 9-7 Timer B1 Operation States

Operation Mode		Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCB1	Interval	Reset	Functions	Functions	Halted	Halted	Halted	Halted
	Auto reload	Reset	Functions	Functions	Halted	Halted	Halted	Halted
TMB1		Reset	Functions	Retained	Retained	Retained	Retained	Retained

9.4 Timer V

9.4.1 Overview

Timer V is an 8-bit timer based on an 8-bit counter. Timer V counts external events. Also compare match signals can be used to reset the counter, request an interrupt, or output a pulse signal with an arbitrary duty cycle. Counting can be initiated by a trigger input at the TRGV pin, enabling pulse output control to be synchronized to the trigger, with an arbitrary delay from the trigger input.

1. Features

Features of timer V are given below.

- Choice of six internal clock sources (ø/128, ø/64, ø/32, ø/16, ø/8, ø/4) or an external clock (can be used as an external event counter).
- Counter can be cleared by compare match A or B, or by an external reset signal. If the trigger function is selected, the counter can be halted when cleared.
- Timer output is controlled by two independent compare match signals, enabling pulse output with an arbitrary duty cycle, PWM output, and other applications.
- Three interrupt sources: two compare match, one overflow
- Counting can be initiated by trigger input at the TRGV pin. The rising edge, falling edge, or both edges of the TRGV input can be selected.

2. Block diagram

Figure 9-3 shows a block diagram of timer V.

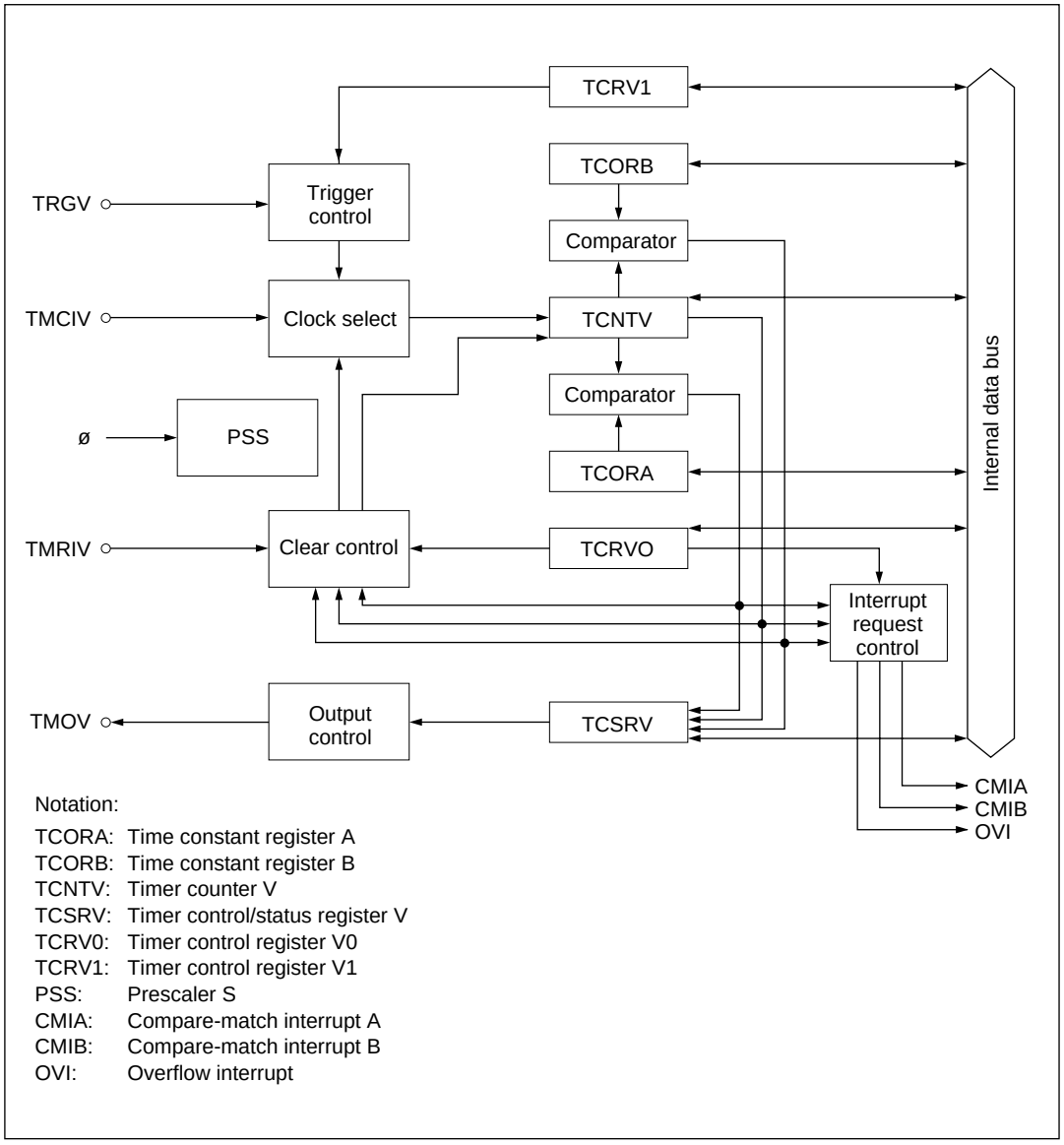


Figure 9-3 Block Diagram of Timer V

3. Pin configuration

Table 9-8 shows the timer V pin configuration.

Table 9-8 Pin Configuration

Name	Abbrev.	I/O	Function
Timer V output	TMOV	Output	Timer V waveform output
Timer V clock input	TMCIV	Input	Clock input to TCNTV
Timer V reset input	TMRIV	Input	External input to reset TCNTV
Trigger input	TRGV	Input	Trigger input to initiate counting

4. Register configuration

Table 9-9 shows the register configuration of timer V.

Table 9-9 Timer V Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer control register V0	TCRV0	R/W	H'00	H'FFB8
Timer control/status register V	TCSRV	R/(W)*	H'10	H'FFB9
Time constant register A	TCORA	R/W	H'FF	H'FFBA
Time constant register B	TCORB	R/W	H'FF	H'FFBB
Timer counter V	TCNTV	R/W	H'00	H'FFBC
Timer control register V1	TCRV1	R/W	H'E2	H'FFBD

Note: * Bits 7 to 5 can only be written with 0, for flag clearing.

9.4.2 Register Descriptions

1. Timer counter V (TCNTV)

Bit	7	6	5	4	3	2	1	0
	$TCNTV_7$	$TCNTV_6$	$TCNTV_5$	$TCNTV_4$	$TCNTV_3$	$TCNTV_2$	$TCNTV_1$	$TCNTV_0$
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

TCNTV is an 8-bit read/write up-counter which is incremented by internal or external clock input. The clock source is selected by bits CKS2 to CKS0 in TCRV0. The TCNTV value can be read and written by the CPU at any time. TCNTV can be cleared by an external reset signal, or by compare match A or B. The clearing signal is selected by bits CCLR1 and CCLR0 in TCRV0.

When TCNTV overflows from H'FF to H'00, OVF is set to 1 in TCSRV.

TCNTV is initialized to H'00 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

2. Time constant registers A and B (TCORA, TCORB)

Bit	7	6	5	4	3	2	1	0
	$TCORn_7$	$TCORn_6$	$TCORn_5$	$TCORn_4$	$TCORn_3$	$TCORn_2$	$TCORn_1$	$TCORn_0$
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

n = A or B

TCORA and TCORB are 8-bit read/write registers.

TCORA and TCNTV are compared at all times, except during the T_3 state of a TCORA write cycle. When the TCORA and TCNTV contents match, CMFA is set to 1 in TCSRV. If CMIEA is also set to 1 in TCRV0, a CPU interrupt is requested.

Timer output from the TMOV pin can be controlled by a signal resulting from compare match, according to the settings of bits OS3 to OS0 in TCSRV.

TCORA is initialized to H'FF upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

TCORB is similar to TCORA.

3. Timer control register V0 (TCRV0)

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

TCRV0 is an 8-bit read/write register that selects the TCNTV input clock, controls the clearing of TCNTV, and enables interrupts.

TCRV0 is initialized to H'00 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

Bit 7: Compare match interrupt enable B (CMIEB)

Bit 7 enables or disables the interrupt request (CMIB) generated from CMFB when CMFB is set to 1 in TCSRV.

Bit 7 CMIEB	Description	
0	Interrupt request (CMIB) from CMFB disabled	(initial value)
1	Interrupt request (CMIB) from CMFB enabled	

Bit 6: Compare match interrupt enable A (CMIEA)

Bit 6 enables or disables the interrupt request (CMIA) generated from CMFA when CMFA is set to 1 in TCSRV.

Bit 6 CMIEA	Description	
0	Interrupt request (CMIA) from CMFA disabled	(initial value)
1	Interrupt request (CMIA) from CMFA enabled	

Bit 5: Timer overflow interrupt enable B (OVIE)

Bit 5 enables or disables the interrupt request (OVI) generated from OVF when OVF is set to 1 in TCSRV.

Bit 5 OVIE	Description	
0	Interrupt request (OVI) from OVF disabled	(initial value)
1	Interrupt request (OVI) from OVF enabled	

Bits 4 and 3: Counter clear 1 and 0 (CCLR1, CCLR0)

Bits 4 and 3 specify whether or not to clear TCNTV, and select compare match A or B or an external reset input.

When clearing is specified, if TRGE is set to 1 in TCRV1, then when TCNTV is cleared it is also halted. Counting resumes when a trigger edge is input at the TRGV pin.

If TRGE is cleared to 0, after TCNTV is cleared it continues counting up.

Bit 4 CCLR1	Bit 3 CCLR0	Description	
0	0	Clearing is disabled	(initial value)
0	1	Cleared by compare match A	
1	0	Cleared by compare match B	
1	1	Cleared by rising edge of external reset input	

Bits 2 to 0: Clock select 2 to 0 (CKS2 to CKS0)

Bits 2 to 0 and bit ICKS0 in TCRV1 select the clock input to TCNTV.

Six internal clock sources divided from the system clock (ø) can be selected. The counter increments on the falling edge.

If the external clock is selected, there is a further selection of incrementing on the rising edge, falling edge, or both edges.

If TRGE is cleared to 0, after TCNTV is cleared it continues counting up.

	TCRV0		TCRV1		
Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Bit 0 ICKS0	Description	
0	0	0	—	Clock input disabled	(initial value)
0	0	1	0	Internal clock: ø/4, falling edge	
0	0	1	1	Internal clock: ø/8, falling edge	
0	1	0	0	Internal clock: ø/16, falling edge	
0	1	0	1	Internal clock: ø/32, falling edge	
0	1	1	0	Internal clock: ø/64, falling edge	
0	1	1	1	Internal clock: ø/128, falling edge	
1	0	0	—	Clock input disabled	
1	0	1	—	External clock: rising edge	
1	1	0	—	External clock: falling edge	
1	1	1	—	External clock: rising and falling edges	

4. Timer control/status register V (TCSRV)

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/W

Note: * Bits 7 to 5 can be only written with 0, for flag clearing.

TCSRV is an 8-bit register that sets compare match flags and the timer overflow flag, and controls compare match output.

TCSRV is initialized to H'10 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

Bit 7: Compare match flag B (CMFB)

Bit 7 is a status flag indicating that TCNTV has matched TCORB. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 7 CMFB	**Description**	
0	Clearing conditions: After reading CMFB = 1, cleared by writing 0 to CMFB	(initial value)
1	Setting conditions: Set when the TCNTV value matches the TCORB value	

Bit 6: Compare match flag A (CMFA)

Bit 6 is a status flag indicating that TCNTV has matched TCORA. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 6 CMFA	**Description**	
0	Clearing conditions: After reading CMFA = 1, cleared by writing 0 to CMFA	(initial value)
1	Setting conditions: Set when the TCNTV value matches the TCORA value	

Bit 5: Timer overflow flag (OVF)

Bit 5 is a status flag indicating that TCNTV has overflowed from H'FF to H'00. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 5 OVF	Description	
0	Clearing conditions: After reading OVF = 1, cleared by writing 0 to OVF	(initial value)
1	Setting conditions: Set when TCNTV overflows from H'FF to H'00	

Bit 4: Reserved bit

Bit 4 is reserved; it is always read as 1, and cannot be modified.

Bits 3 to 0: Output select 3 to 0 (OS3 to OS0)

Bits 3 to 0 select the way in which the output level at the TMOV pin changes in response to compare match between TCNTV and TCORA or TCORB.

OS3 and OS2 select the output level for compare match B. OS1 and OS0 select the output level for compare match A. The two levels can be controlled independently.

If two compare matches occur simultaneously, any conflict between the settings is resolved according to the following priority order: toggle output > 1 output > 0 output.

When OS3 to OS0 are all cleared to 0, timer output is disabled.

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

Bit 3 OS3	Bit 2 OS2	Description	
0	0	No change at compare match B	(initial value)
0	1	0 output at compare match B	
1	0	1 output at compare match B	
1	1	Output toggles at compare match B	

Bit 1 OS1	Bit 0 OS0	Description	
0	0	No change at compare match A	(initial value)
0	1	0 output at compare match A	
1	0	1 output at compare match A	
1	1	Output toggles at compare match A	

5. Timer control register V1 (TCRV1)

Bit	7	6	5	4	3	2	1	0
	—	——	—	TVEG1	TVEG0	TRGE	—	ICKS0
Initial value	1	1	1	0	0	0	1	0
Read/Write	—	—	—	R/W	R/W	R/W	—	R/W

TCRV1 is an 8-bit read/write register that selects the valid edge at the TRGV pin, enables TRGV input, and selects the clock input to TCNTV.

TCRV1 is initialized to H'E2 upon reset and in watch mode, subsleep mode, and subactive mode.

Bits 7 to 5: Reserved bits

Bit 7 to 5 are reserved; they are always read as 1, and cannot be modified.

Bits 4 and 3: TRGV input edge select (TVEG1, TVEG0)

Bits 4 and 3 select the TRGV input edge.

Bit 4 TVEG1	Bit 3 TVEG0	Description	
0	0	TRGV trigger input is disabled	(initial value)
0	1	Rising edge is selected	
1	0	Falling edge is selected	
1	1	Rising and falling edges are both selected	

Bit 2: TRGV input enable (TRGE)

Bit 2 enables or disables TCNTV counting to be triggered by input at the TRGV pin, and enables or disables TCNTV counting to be halted when TCNTV is cleared by compare match. TCNTV stops counting when TRGE is set to 1, then starts counting when the edge selected by bits TVEG1 and TVEG0 is input at the TRGV pin.

Bit 2 TRGE	Description	
0	TCNTV counting is not triggered by input at the TRGV pin, and does not stop when TCNTV is cleared by compare match	(initial value)
1	TCNTV counting is triggered by input at the TRGV pin, and stops when TCNTV is cleared by compare match	

Bit 1: Reserved bit

Bit 1 is reserved; it is always read as 1, and cannot be modified.

Bit 0: Internal clock select 0 (ICKS0)

Bit 0 and bits CKS2 to CKS0 in TCRV0 select the TCNTV clock source. For details see 9.8.2 (3), Timer control register V0.

9.4.3 Timer Operation

1. Timer V operation

A reset initializes TCNTV to H'00, TCORA and TCORB to H'FF, TCRV0 to H'00, TCSRV to H'10, and TCRV1 to H'E2.

Timer V can be clocked by one of six internal clocks output from prescaler S, or an external clock, as selected by bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1. The valid edge or edges of the external clock can also be selected by CKS2 to CKS0. When the clock source is selected, TCNTV starts counting the selected clock input.

The TCNTV contents are always compared with TCORA and TCORB. When a match occurs, the CMFA or CMFB bit is set to 1 in TCSRV. If CMIEA or CMIEB is set to 1 in TCRV0, a CPU interrupt is requested. At the same time, the output level selected by bits OS3 to OS0 in TCSRV is output from the TMOV pin.

When TCNT overflows from H'FF to H'00, if OVIE is 1 in TCRV0, a CPU interrupt is requested.

If bits CCLR1 and CCLR0 in TCRV0 are set to 01 (clear by compare match A) or 10 (clear by compare match B), TCNTV is cleared by the corresponding compare match. If these bits are set to 11, TCNTV is cleared by input of a rising edge at the TMRIV pin.

If bit TRGE is set to 1 in TCRV1, when TCNTV is cleared by the event selected by bits CCLR1 and CCLR0, it is also halted. TCNTV starts counting when the signal edge selected by bits TVEG1 and TVEG0 in TCRV1 is input at the TRGV pin.

2. TCNTV increment timing

TCNTV is incremented by an input (internal or external) clock.

- Internal clock

  One of six clocks (ø/128, ø/64, ø/32, ø/16, ø/8, ø/4) divided from the system clock (ø) can be selected by bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1. Figure 9-4 shows the timing.

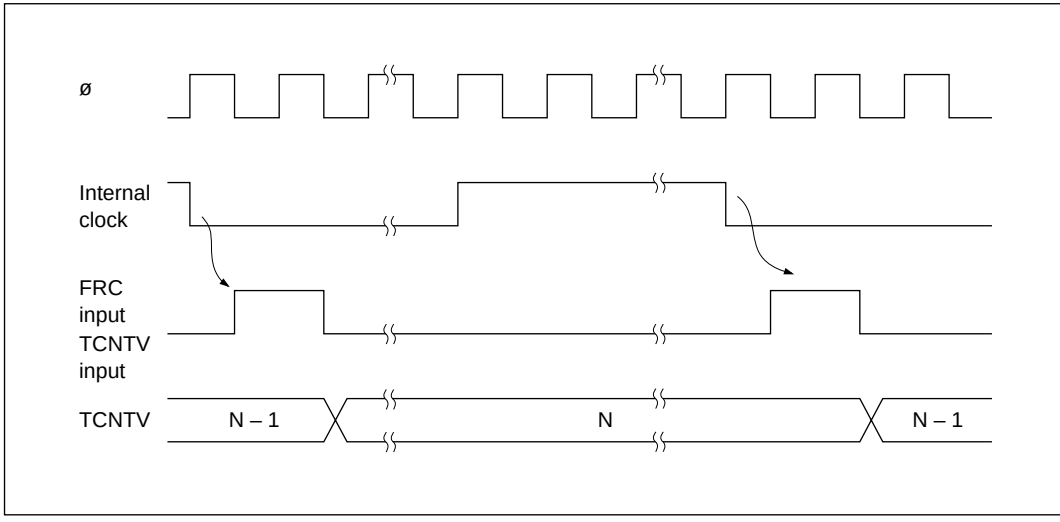


Figure 9-4 Increment Timing with Internal Clock

- External clock

  Incrementation on the rising edge, falling edge, or both edges of the external clock can be selected by bits CKS2 to CKS0 in TCRV0.

  The external clock pulse width should be at least 1.5 system clocks (ø) when a single edge is counted, and at least 2.5 system clocks when both edges are counted. Shorter pulses will not be counted correctly.

  Figure 9-5 shows the timing when both the rising and falling edges of the external clock are selected.

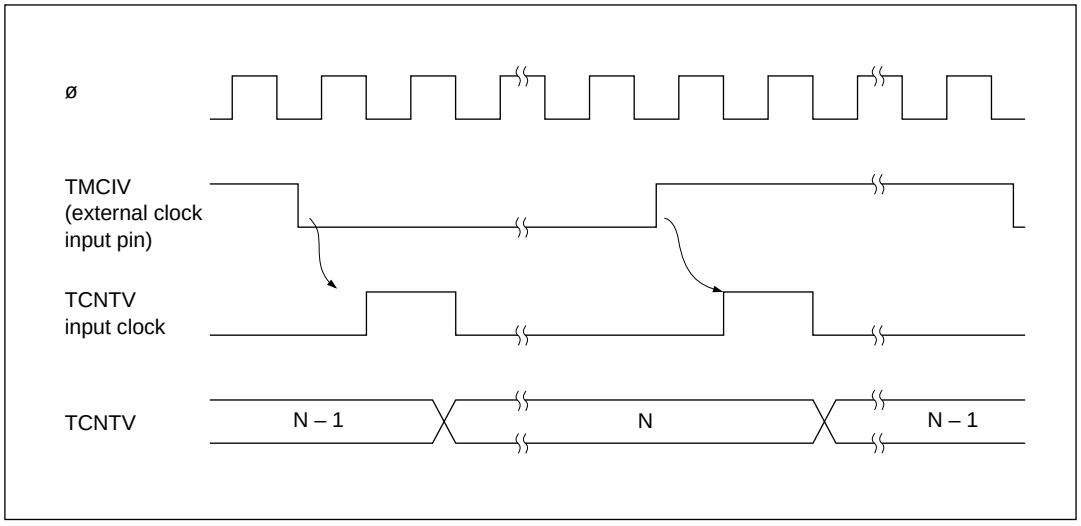


Figure 9-5 Increment Timing with External Clock

3. Overflow flag set timing

The overflow flag (OVF) is set to 1 when TCNTV overflows from H'FF to H'00. Figure 9-6 shows the timing.

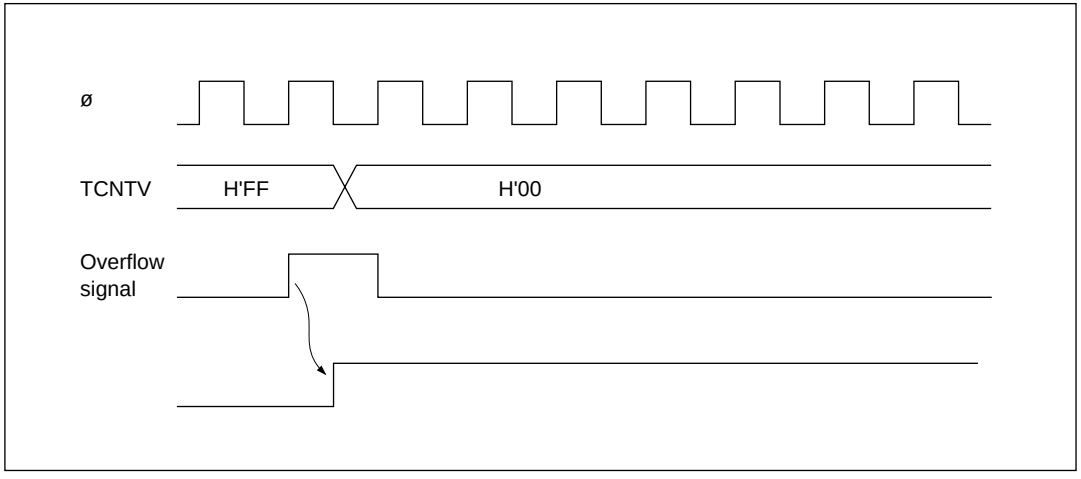


Figure 9-6 OVF Set Timing

4. Compare match flag set timing

Compare match flag A or B (CMFA or CMFB) is set to 1 when TCNTV matches TCORA or TCORB. The internal compare-match signal is generated in the last state in which the values match (when TCNTV changes from the matching value to a new value). Accordingly, when TCNTV matches TCORA or TCORB, the compare match signal is not generated until the next clock input to TCNTV. Figure 9-7 shows the timing.

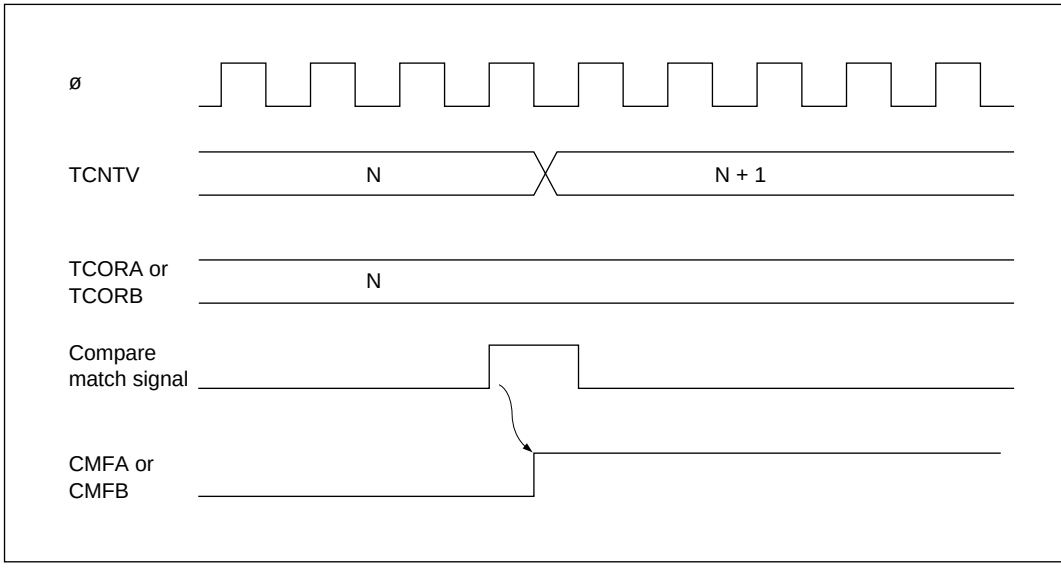


Figure 9-7 CMFA and CMFB Set Timing

5. TMOV output timing

The TMOV output responds to compare match A or B by remaining unchanged, changing to 0, changing to 1, or toggling, as selected by bits OS3 to OS0 in TCSRV. Figure 9-8 shows the timing when the output is toggled by compare match A.

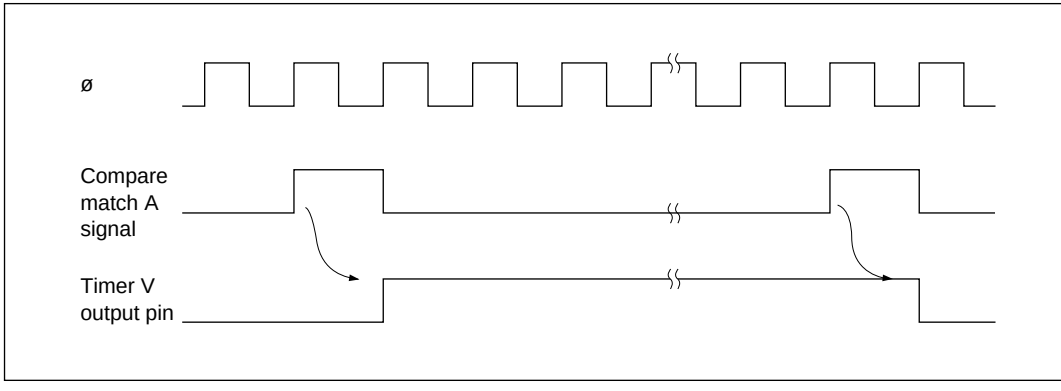


Figure 9-8 TMOV Output Timing

6. TCNTV clear timing by compare match

TCNTV can be cleared by compare match A or B, as selected by bits CCLR1 and CCLR0 in TCRV0. Figure 9-9 shows the timing.

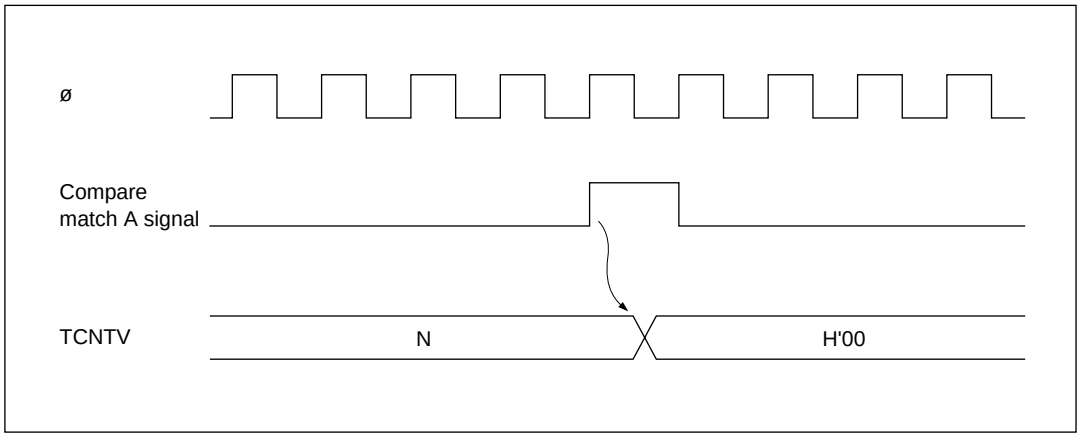


Figure 9-9 Clear Timing by Compare Match

7. TCNTV clear timing by TMRIV

TCNTV can be cleared by a rising edge at the TMRIV pin, as selected by bits CCLR1 and CCLR0 in TCRV0. A TMRIV input pulse width of at least 1.5 system clocks is necessary. Figure 9-10 shows the timing.

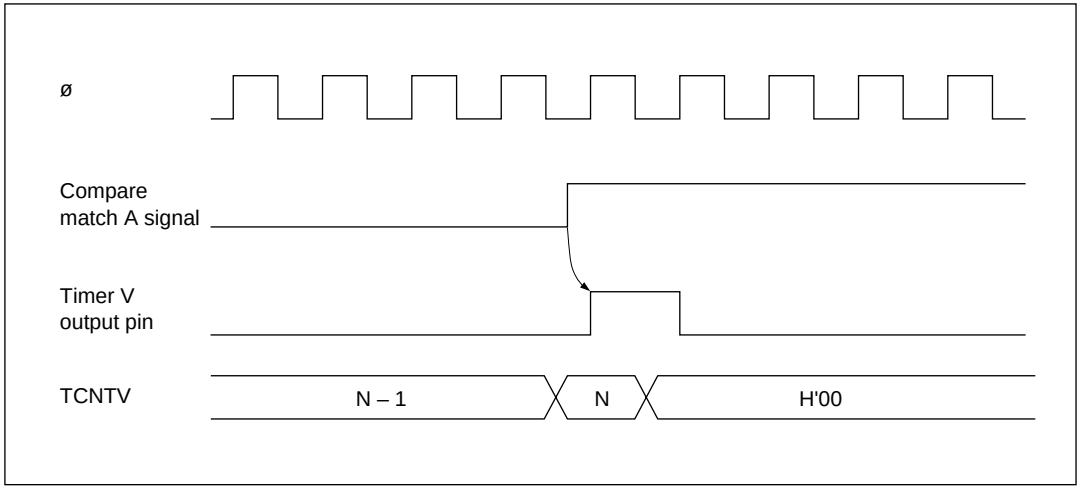


Figure 9-10 Clear Timing by TMRIV Input

9.4.4 Timer V Operation Modes

Table 9-10 summarizes the timer V operation states.

Table 9-10 Timer V Operation States

Operation Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCNTV	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TCRV0, TCRV1	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TCORA, TCORB	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TCSRV	Reset	Functions	Functions	Reset	Reset	Reset	Reset

9.4.5 Interrupt Sources

Timer V has three interrupt sources: CMIA, CMIB, and OVI. Table 9-11 lists the interrupt sources and their vector address. Each interrupt source can be enabled or disabled by an interrupt enable bit in TCRV0. Although all three interrupts share the same vector, they have individual interrupt flags, so software can discriminate the interrupt source.

Table 9-11 Timer V Interrupt Sources

Interrupt	Description	Vector Address
CMIA	Generated from CMFA	H'0022
CMIB	Generated from CMFB	
OVI	Generated from OVF	

9.4.6 Application Examples

1. Pulse output with arbitrary duty cycle

Figure 9-11 shows an example of output of pulses with an arbitrary duty cycle. To set up this output:

- Clear bit CCLR1 to 0 and set bit CCLR0 to 1 in TCRV0 so that TCNTV will be cleared by compare match with TCORA.
- Set bits OS3 to OS0 to 0110 in TCSRV so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
- Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.

With these settings, a waveform is output without further software intervention, with a period determined by TCORA and a pulse width determined by TCORB.

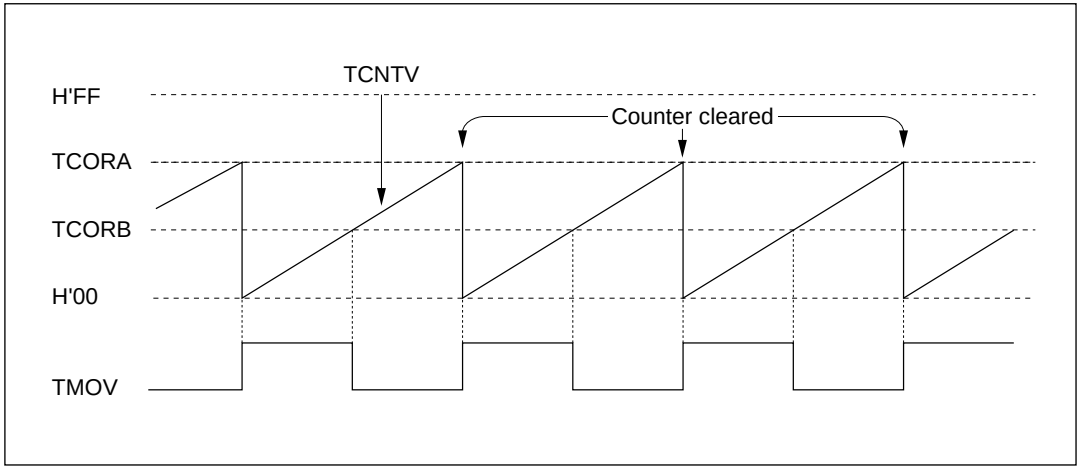


Figure 9-11 Pulse Output Example

2. Single-shot output with arbitrary pulse width and delay from TRGV input

The trigger function can be used to output a pulse with an arbitrary pulse width at an arbitrary delay from the TRGV input, as shown in figure 9-12. To set up this output:

- Set bit CCLR1 to 1 and clear bit CCLR0 to 0 in TCRV0 so that TCNTV will be cleared by compare match with TCORB.
- Set bits OS3 to OS0 to 0110 in TCSRV so that the output will go to 1 at compare match with TCORA and to 0 at compare match with TCORB.
- Set bits TVEG1 and TVEG0 to 10 in TCRV1 and set TRGE to 1 to select the falling edge of the TRGV input.
- Set bits CKS2 to CKS0 in TCRV0 and bit ICKS0 in TCRV1 to select the desired clock source.

After these settings, a pulse waveform will be output without further software intervention, with a delay determined by TCORA from the TRGV input, and a pulse width determined by (TCORB – TCORA).

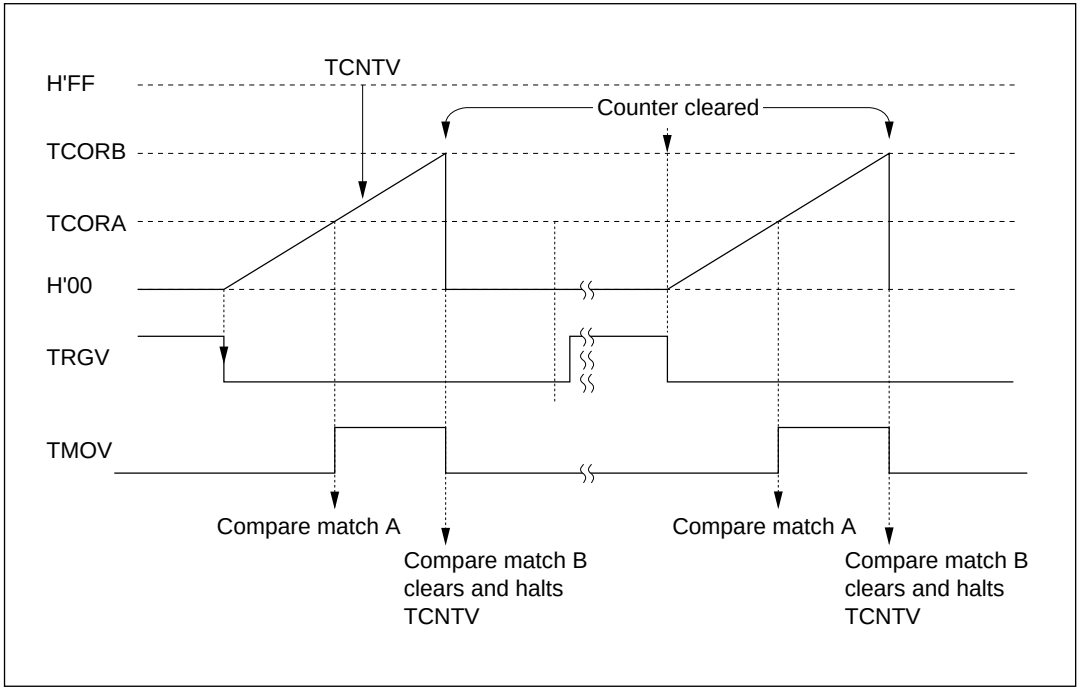


Figure 9-12 Pulse Output Synchronized to TRGV Input

9.4.7 Application Notes

The following types of contention can occur in timer V operation.

1. Contention between TCNTV write and counter clear

If a TCNTV clear signal is generated in the T_3 state of a TCNTV write cycle, clearing takes precedence and the write to the counter is not carried out. Figure 9-13 shows the timing.

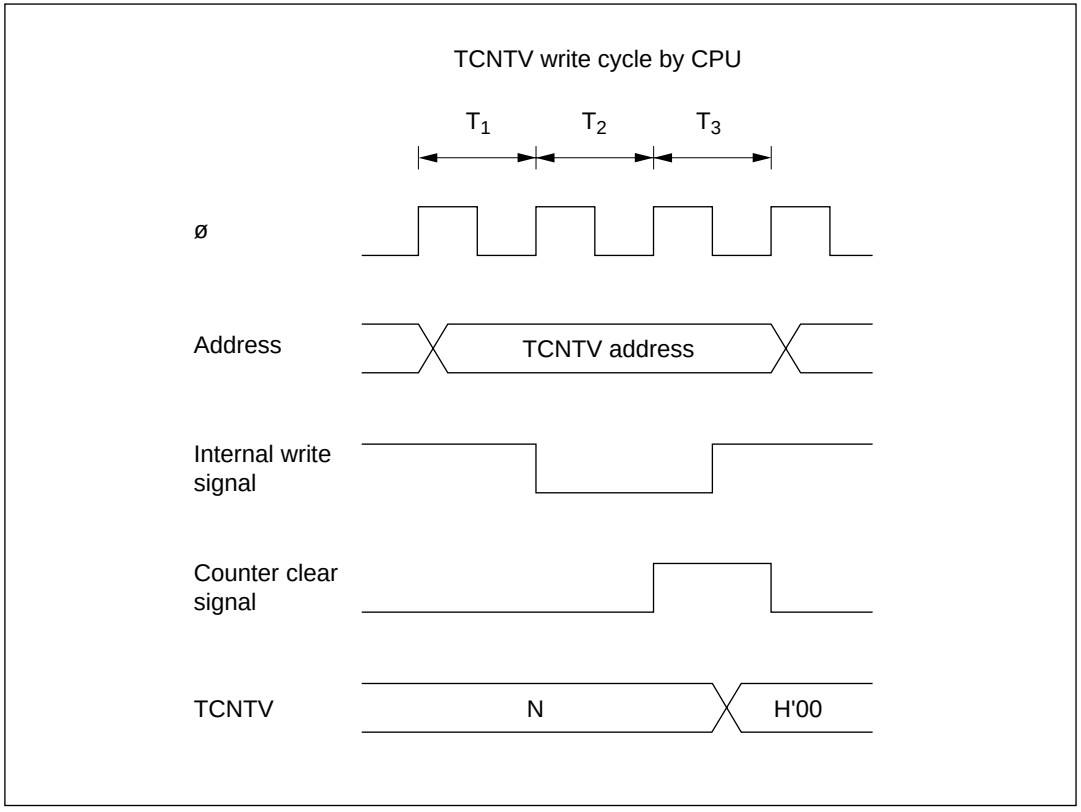


Figure 9-13 Contention between TCNTV Write and Clear

2. Contention between TCNTV write and increment

If a TCNTV increment clock signal is generated in the T_3 state of a TCNTV write cycle, the write takes precedence and the counter is not incremented. Figure 9-14 shows the timing.

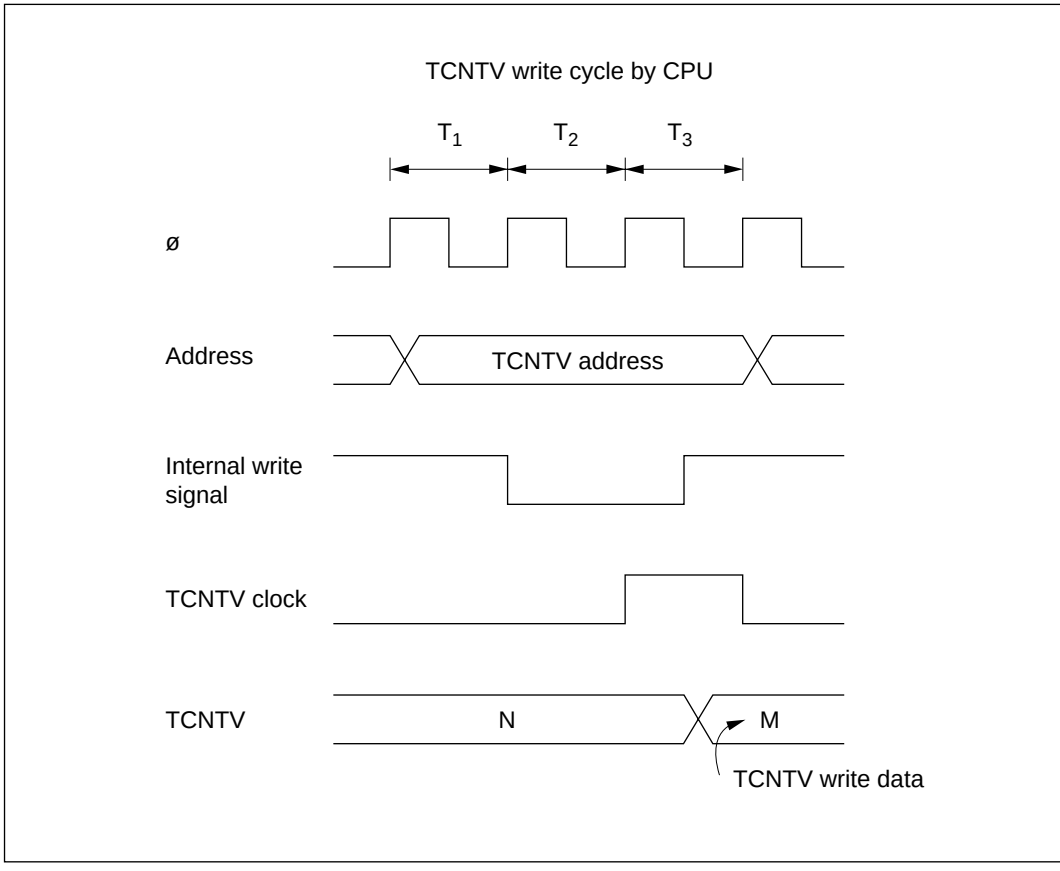


Figure 9-14 Contention between TCNTV Write and Increment

3. Contention between TCOR write and compare match

If a compare match is generated in the T_3 state of a TCORA or TCORB write cycle, the write to TCORA or TCORB takes precedence and the compare match signal is inhibited. Figure 9-15 shows the timing.

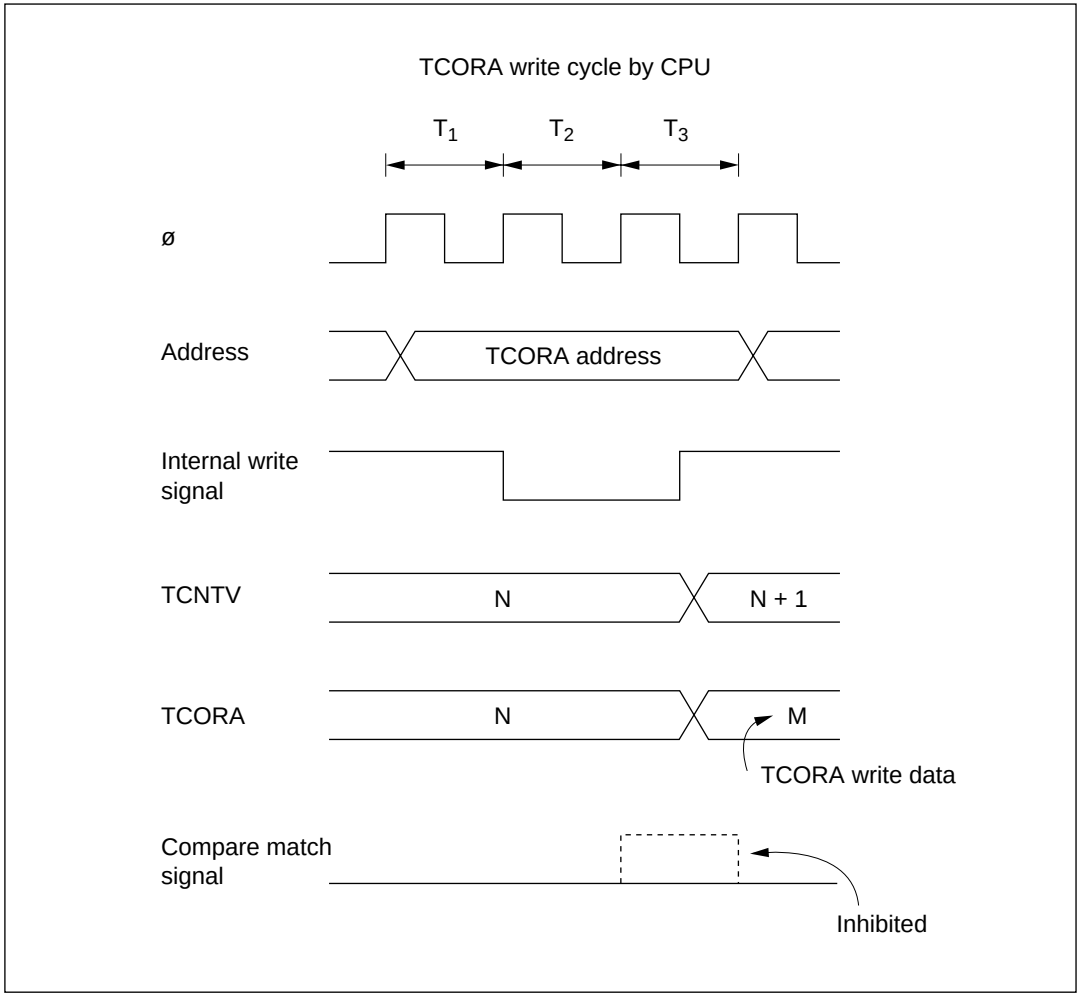


Figure 9-15 Contention between TCORA Write and Compare Match

4. Contention between compare match A and B

If compare match A and B occur simultaneously, any conflict between the output selections for compare match A and compare match B is resolved by following the priority order in table 9-12.

Table 9-12 Timer Output Priority Order

Output Setting	Priority
Toggle output	High
1 output	↑
0 output	↑
No change	Low

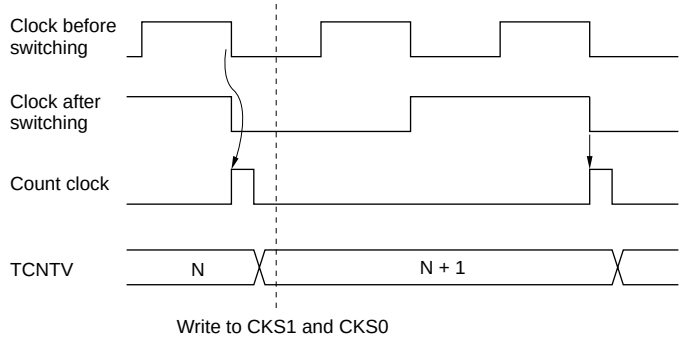
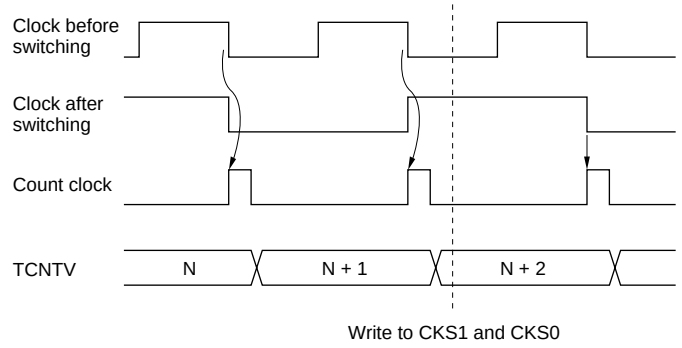
5. Internal clock switching and counter operation

Depending on the timing, TCNTV may be incremented by a switch between different internal clock sources. Table 9-13 shows the relation between internal clock switchover timing (by writing to bits CKS1 and CKS0) and TCNTV operation.

When TCNTV is internally clocked, an increment pulse is generated from the falling edge of an internal clock signal, which is divided from the system clock (ø). For this reason, in a case like No. 3 in table 9-13 where the switch is from a high clock signal to a low clock signal, the switchover is seen as a falling edge, causing TCNTV to increment.

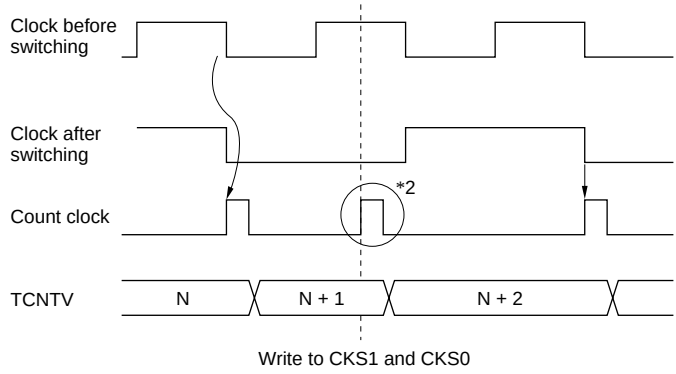
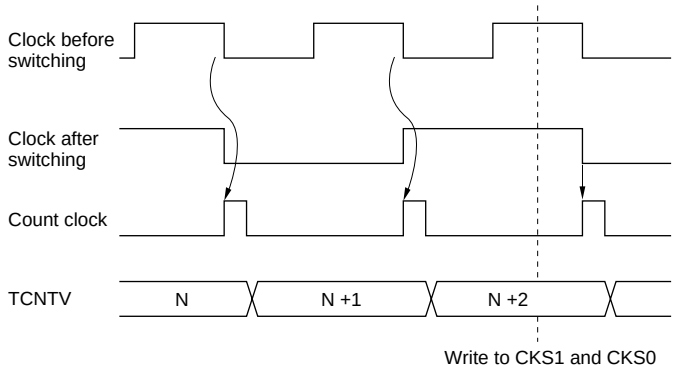
TCNTV can also be incremented by a switch between internal and external clocks.

Table 9-13 Internal Clock Switching and TCNTV Operation

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	TCNTV Operation
1	Goes from low level to low level*1	
2	Goes from low to high*2	

Notes: 1. Including a transition from the low level to the stopped state, or from the stopped state to the low level.
2. Including a transition from the stopped state to the high level.

Table 9-13 Internal Clock Switching and TCNTV Operation (cont)

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	TCNTV Operation
3	Goes from high level to low level*1	
4	Goes from high to high	

Notes: 1. Including a transition from the high level to the stopped state.
2. The switchover is seen as a falling edge, and TCNTV is incremented.

9.5 Timer X

9.5.1 Overview

Timer X is based on a 16-bit free-running counter (FRC). It can output two independent waveforms, or measure input pulse widths and external clock periods.

1. Features

Features of timer X are given below.

- Choice of three internal clock sources (ø/2, ø/8, ø/32) or an external clock (can be used as an external event counter).
- Two independent output compare waveforms.
- Four independent input capture channels, with selection of rising or falling edge and buffering option.
- Counter can be cleared by compare match A.
- Seven independent interrupt sources: two compare match, four input capture, one overflow

2. Block diagram

Figure 9-16 shows a block diagram of timer X.

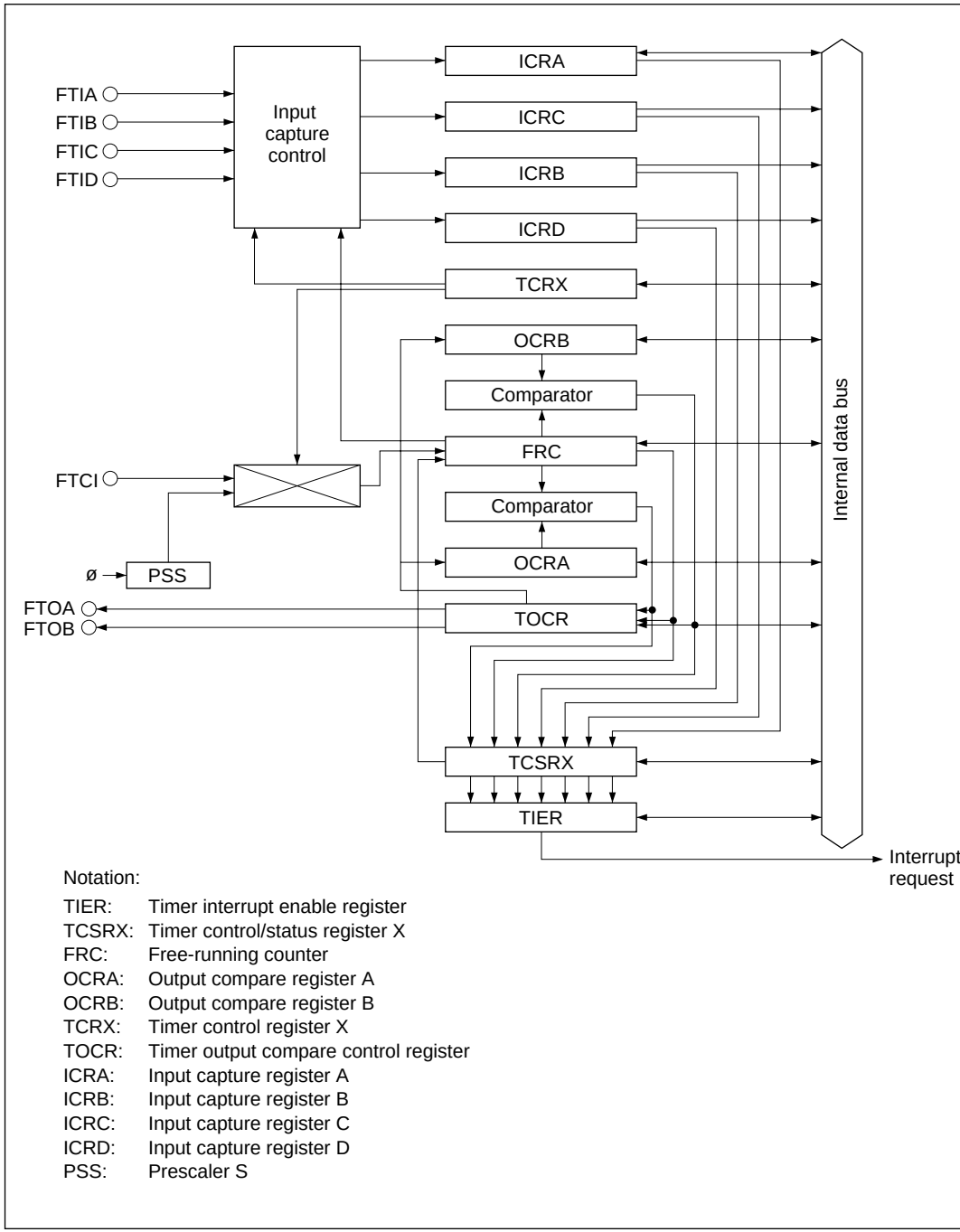


Figure 9-16 Block Diagram of Timer X

3. Pin configuration

Table 9-14 shows the timer X pin configuration.

Table 9-14 Pin Configuration

Name	Abbrev.	I/O	Function
Counter clock input	FTCI	Input	Clock input to FRC
Output compare A	FTOA	Output	Output pin for output compare A
Output compare B	FTOB	Output	Output pin for output compare B
Input capture A	FTIA	Input	Input pin for input capture A
Input capture B	FTIB	Input	Input pin for input capture B
Input capture C	FTIC	Input	Input pin for input capture C
Input capture D	FTID	Input	Input pin for input capture D

4. Register configuration

Table 9-15 shows the register configuration of timer X.

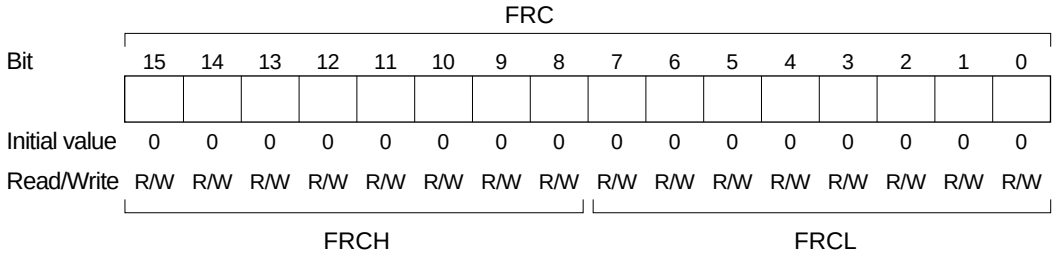
Table 9-15 Timer X Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer interrupt enable register	TIER	R/W	H'01	H'F770
Timer control/status register X	TCSRX	R/(W)*1	H'00	H'F771
Free-running counter H	FRCH	R/W	H'00	H'F772
Free-running counter L	FRCL	R/W	H'00	H'F773
Output compare register AH	OCRAH	R/W	H'FF	H'F774*2
Output compare register AL	OCRAL	R/W	H'FF	H'F775*2
Output compare register BH	OCRBH	R/W	H'FF	H'F774*2
Output compare register BL	OCRBL	R/W	H'FF	H'F775*2
Timer control register X	TCRX	R/W	H'00	H'F776
Timer output compare control register	TOCR	R/W	H'E0	H'F777
Input capture register AH	ICRAH	R	H'00	H'F778
Input capture register AL	ICRAL	R	H'00	H'F779
Input capture register BH	ICRBH	R	H'00	H'F77A
Input capture register BL	ICRBL	R	H'00	H'F77B
Input capture register CH	ICRCH	R	H'00	H'F77C
Input capture register CL	ICRCL	R	H'00	H'F77D
Input capture register DH	ICRDH	R	H'00	H'F77E
Input capture register DL	ICRDL	R	H'00	H'F77F

Notes: 1. Bits 7 to 1 can only be written with 0 for flag clearing. Bit 0 is a read/write bit.
2. OCRA and OCRB share the same address. They are selected by the OCRS bit in TOCR.

9.5.2 Register Descriptions

1. Free-running counter (FRC)
   Free-running counter H (FRCH)
   Free-running counter L (FRCL)



FRC is a 16-bit read/write up-counter, which is incremented by internal or external clock input. The clock source is selected by bits CKS1 and CKS0 in TCRX.

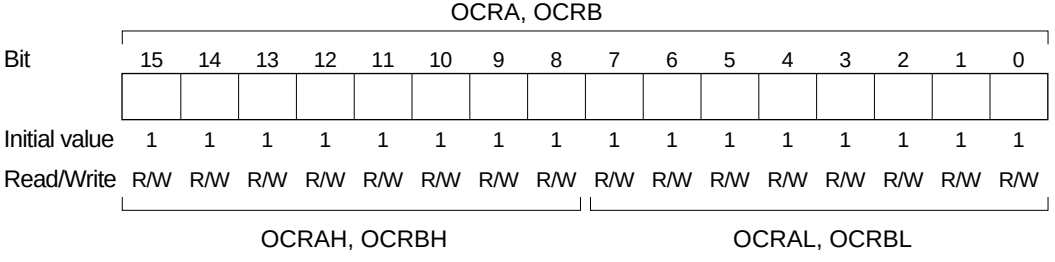
FRC can be cleared by compare match A, depending on the setting of CCLRA in TCSRX.

When FRC overflows from H'FFFF to H'0000, OVF is set to 1 in TCSRX. If OVIE = 1 in TIER, a CPU interrupt is requested.

FRC can be written and read by the CPU. Since FRC has 16 bits, data is transferred between the CPU and FRC via a temporary register (TEMP). For details see 9.5.3, CPU Interface.

FRC is initialized to H'0000 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

2. Output compare registers A and B (OCRA, OCRB)
   Output compare registers AH and BH (OCRAH, OCRBH)
   Output compare registers AL and BL (OCRAL, OCRBL)



There are two 16-bit read/write output compare registers, OCRA and OCRB, the contents of which are always compared with FRC. When the values match, OCFA or OCFB is set to 1 in TCSRX. If

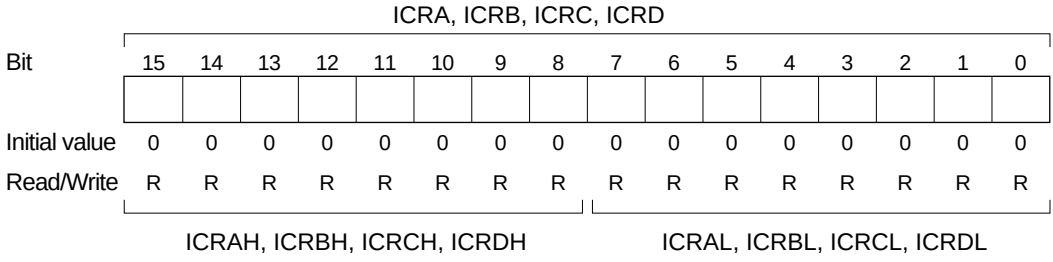
OCIAE = 1 or OCIBE = 1 in TIER, a CPU interrupt is requested.

When a compare match with OCRA or OCRB occurs, if OEA = 1 or OEB = 1 in TOCR, the value selected by OLVLA or OLVLB in TOCR is output at the FTOA or FTOB pin. After a reset, the output from the FTOA or FTOB pin is 0 until the first compare match occurs.

OCRA and OCRB can be written and read by the CPU. Since they are 16-bit registers, data is transferred between them and the CPU via a temporary register (TEMP). For details see 9.5.3, CPU Interface.

OCRA and OCRB are initialized to H'FFFF upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

3. Input capture registers A to D (ICRA to ICRD)
   Input capture registers AH to DH (ICRAH to ICRDH)
   Input capture registers AL to DL (ICRAL to ICRDL)



There are four 16-bit read only input capture registers, ICRA to ICRD.

When the falling edge of an input capture signal is input, the FRC value is transferred to the corresponding input capture register, and the corresponding input capture flag (ICFA to ICFD) is set to 1 in TCSRX. If the corresponding input capture interrupt enable bit (ICIAE to ICIDE) is 1 in TIER, a CPU interrupt is requested. The valid edge of the input signal can be selected by bits IEDGA to IEDGD in TCRX.

ICRC and ICRD can also be used as buffer registers for ICRA and ICRB. Buffering is enabled by bits BUFEA and BUFEB in TCRX.

Figure 9-17 shows the interconnections when ICRC operates as a buffer register of ICRA (when BUFEA = 1). In buffered input capture operations, both the rising and falling edges of the external input signal can be selected simultaneously, by setting IEDGA ≠ IEDGC. If IEDGA = IEDGC, then only one edge is selected (either the rising edge or falling edge). See table 9-16.

Note: The FRC value is transferred to the input capture register (ICR) regardless of the value of the input capture flag (ICF).

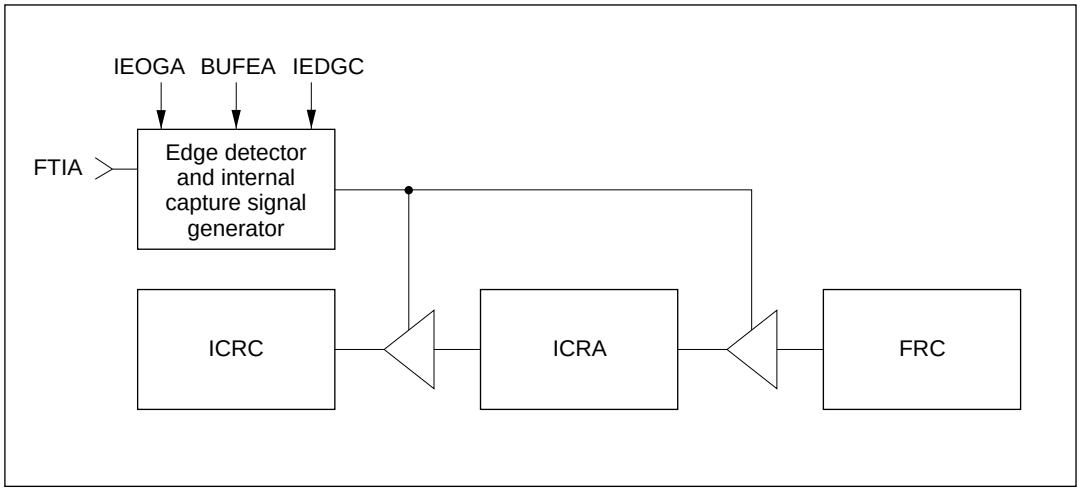


Figure 9-17 Buffer Operation (Example)

Table 9-16 Input Edge Selection during Buffer Operation

IEDGA	IEDGC	Input Edge Selection
0	0	Falling edge of input capture A input signal is captured (initial value)
0	1	Rising and falling edge of input capture A input signal are both captured
1	0	
1	1	Rising edge of input capture A input signal is captured

ICRA to ICRD can be written and read by the CPU. Since they are 16-bit registers, data is transferred from them to the CPU via a temporary register (TEMP). For details see 9.5.3, CPU Interface.

To assure input capture, the pulse width of the input capture input signal must be at least 1.5 system clocks (ø) when a single edge is selected, or at least 2.5 system clocks (ø) when both edges are selected.

ICRA to ICRD are initialized to H'0000 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

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 read/write register that enables or disables interrupt requests.

TIER is initialized to H'01 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

Bit 7: Input capture interrupt A enable (ICIAE)

Bit 7 enables or disables the ICIA interrupt requested when ICFA is set to 1 in TCSRX.

Bit 7 ICIAE	Description	
0	Interrupt request by ICFA (ICIA) is disabled	(initial value)
1	Interrupt request by ICFA (ICIA) is enabled	

Bit 6: Input capture interrupt B enable (ICIBE)

Bit 6 enables or disables the ICIB interrupt requested when ICFB is set to 1 in TCSRX.

Bit 6 ICIBE	Description	
0	Interrupt request by ICFB (ICIB) is disabled	(initial value)
1	Interrupt request by ICFB (ICIB) is enabled	

Bit 5: Input capture interrupt C enable (ICICE)

Bit 5 enables or disables the ICIC interrupt requested when ICFC is set to 1 in TCSRX.

Bit 5 ICICE	Description	
0	Interrupt request by ICFC (ICIC) is disabled	(initial value)
1	Interrupt request by ICFC (ICIC) is enabled	

Bit 4: Input capture interrupt D enable (ICIDE)

Bit 4 enables or disables the ICID interrupt requested when ICFD is set to 1 in TCSRX.

Bit 4 ICIDE	Description	
0	Interrupt request by ICFD (ICID) is disabled	(initial value)
1	Interrupt request by ICFD (ICID) is enabled	

Bit 3: Output compare interrupt A enable (OCIAE)

Bit 3 enables or disables the OCIA interrupt requested when OCFA is set to 1 in TCSRX.

Bit 3 OCIAE	Description	
0	Interrupt request by OCFA (OCIA) is disabled	(initial value)
1	Interrupt request by OCFA (OCIA) is enabled	

Bit 2: Output compare interrupt B enable (OCIBE)

Bit 2 enables or disables the OCIB interrupt requested when OCFB is set to 1 in TCSRX.

Bit 2 OCIBE	Description	
0	Interrupt request by OCFB (OCIB) is disabled	(initial value)
1	Interrupt request by OCFB (OCIB) is enabled	

Bit 1: Timer overflow interrupt enable (OVIE)

Bit 1 enables or disables the FOVI interrupt requested when OVF is set to 1 in TCSRX.

Bit 1 OVIE	Description	
0	Interrupt request by OVF (FOVI) is disabled	(initial value)
1	Interrupt request by OVF (FOVI) is enabled	

Bit 0: Reserved bit

Bit 0 is reserved; it is always read as 1, and cannot be modified.

5. Timer control/status register X (TCSRX)

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: * Bits 7 to 1 can only be written with 0 for flag clearing.

TCSRX is an 8-bit register that selects clearing of the counter and controls interrupt request signals.

TCSRX is initialized to H'00 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode. Other timing is described in section 9-6-3, Timer Operation.

Bit 7: Input capture flag A (ICFA)

Bit 7 is a status flag that indicates that the FRC value has been transferred to ICRA by an input capture signal. If BUFEA is set to 1 in TCRX, ICFA indicates that the FRC value has been transferred to ICRA by an input capture signal and that the ICRA value before this update has been transferred to ICRC.

This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 7 ICFA	Description	
0	Clearing conditions: After reading ICFA = 1, cleared by writing 0 to ICFA	(initial value)
1	Setting conditions: Set when the FRC value is transferred to ICRA by an input capture signal	

Bit 6: Input capture flag B (ICFB)

Bit 6 is a status flag that indicates that the FRC value has been transferred to ICRB by an input capture signal. If BUFEB is set to 1 in TCRX, ICFB indicates that the FRC value has been transferred to ICRB by an input capture signal and that the ICRB value before this update has been transferred to ICRD.

This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 6 ICFB	Description	
0	Clearing conditions: After reading ICFB = 1, cleared by writing 0 to ICFB	(initial value)
1	Setting conditions: Set when the FRC value is transferred to ICRB by an input capture signal	

Bit 5: Input capture flag C (ICFC)

Bit 5 is a status flag that indicates that the FRC value has been transferred to ICRC by an input capture signal. If BUFEA is set to 1 in TCRX, ICFC is set by the input capture signal even though the FRC value is not transferred to ICRC. In buffered operation, ICFC can accordingly be used as an external interrupt, by setting the ICICE bit to 1.

This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 5 ICFC	Description	
0	Clearing conditions: After reading ICFC = 1, cleared by writing 0 to ICFC	(initial value)
1	Setting conditions: Set by input capture signal	

Bit 4: Input capture flag D (ICFD)

Bit 4 is a status flag that indicates that the FRC value has been transferred to ICRD by an input capture signal. If BUFEB is set to 1 in TCRX, ICFD is set by the input capture signal even though the FRC value is not transferred to ICRD. In buffered operation, ICFD can accordingly be used as an external interrupt, by setting the ICIDE bit to 1.

This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 4 ICFD	Description	
0	Clearing conditions: After reading ICFD = 1, cleared by writing 0 to ICFD	(initial value)
1	Setting conditions: Set by input capture signal	

Bit 3: Output compare flag A (OCFA)

Bit 3 is a status flag that indicates that the FRC value has matched OCRA. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 3 OCFA	Description	
0	Clearing conditions: After reading OCFA = 1, cleared by writing 0 to OCFA	(initial value)
1	Setting conditions: Set when FRC matches OCRA	

Bit 2: Output compare flag B (OCFB)

Bit 2 is a status flag that indicates that the FRC value has matched OCRB. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 2 OCFB	Description	
0	Clearing conditions: After reading OCFB = 1, cleared by writing 0 to OCFB	(initial value)
1	Setting conditions: Set when FRC matches OCRB	

Bit 1: Timer overflow flag (OVF)

Bit 1 is a status flag that indicates that FRC has overflowed from H'FFFF to H'0000. This flag is set by hardware and cleared by software. It cannot be set by software.

Bit 1 OVF	Description	
0	Clearing conditions: After reading OVF = 1, cleared by writing 0 to OVF	(initial value)
1	Setting conditions: Set when the FRC value overflows from H'FFFF to H'0000	

Bit 0: Counter clear A (CCLRA)

Bit 0 selects whether or not to clear FRC by compare match A (when FRC matches OCRA).

Bit 0 CCLRA	Description	
0	FRC is not cleared by compare match A	(initial value)
1	FRC is cleared by compare match A	

6. Timer control register X (TCRX)

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

TCRX is an 8-bit read/write register that selects the valid edges of the input capture signals, enables buffering, and selects the FRC clock source.

TCRX is initialized to H'00 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

Bit 7: Input edge select A (IEDGA)

Bit 7 selects the rising or falling edge of the input capture A input signal (FTIA).

Bit 7 IEDGA	Description	
0	Falling edge of input A is captured	(initial value)
1	Rising edge of input A is captured	

Bit 6: Input edge select B (IEDGB)

Bit 6 selects the rising or falling edge of the input capture B input signal (FTIB).

Bit 6 IEDGB	Description	
0	Falling edge of input B is captured	(initial value)
1	Rising edge of input B is captured	

Bit 5: Input edge select C (IEDGC)

Bit 5 selects the rising or falling edge of the input capture C input signal (FTIC).

Bit 5 IEDGC	Description	
0	Falling edge of input C is captured	(initial value)
1	Rising edge of input C is captured	

Bit 4: Input edge select D (IEDGD)

Bit 4 selects the rising or falling edge of the input capture D input signal (FTID).

Bit 4 IEDGD	Description	
0	Falling edge of input D is captured	(initial value)
1	Rising edge of input D is captured	

Bit 3: Buffer enable A (BUFEA)

Bit 3 selects whether or not to use ICRC as a buffer register for ICRA.

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

Bit 2: Buffer enable B (BUFEB)

Bit 2 selects whether or not to use ICRD as a buffer register for ICRB.

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

Bits 1 and 0: Clock select (CKS1, CKS0)

Bits 1 and 0 select one of three internal clock sources or an external clock for input to FRC. The external clock is counted on the rising edge.

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	Internal clock: ø/2	(initial value)
0	1	Internal clock: ø/8	
1	0	Internal clock: ø/32	
1	1	External clock: rising edge	

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 read/write register that selects the output compare output levels, enables output compare output, and controls access to OCRA and OCRB.

TOCR is initialized to H'E0 upon reset and in standby mode, watch mode, subsleep mode, and subactive mode.

Bits 7 to 5: Reserved bits

Bit 7 to 5 are reserved; they are always read as 1, and cannot be modified.

Bit 4: Output compare register select (OCRS)

OCRA and OCRB share the same address. OCRS selects which register is accessed when this address is written or read. It does not affect the operation of OCRA and OCRB.

Bit 4 OCRS	Description	
0	OCRA is selected	(initial value)
1	OCRB is selected	

Bit 3: Output enable A (OEA)

Bit 3 enables or disables the timer output controlled by output compare A.

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)

Bit 2 enables or disables the timer output controlled by output compare B.

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)

Bit 1 selects the output level that is output at pin FTOA by compare match A (when FRC matches OCRA).

Bit 1 OLVLA	Description	
0	Low level	(initial value)
1	High level	

Bit 0: Output level B (OLVLB)

Bit 0 selects the output level that is output at pin FTOB by compare match B (when FRC matches OCRB).

Bit 0 OLVLB	Description	
0	Low level	(initial value)
1	High level	

9.5.3 CPU Interface

FRC, OCRA, OCRB, and ICRA to ICRD are 16-bit registers, but the CPU is connected to the on-chip peripheral modules by an 8-bit data bus. When the CPU accesses these registers, it therefore uses an 8-bit temporary register (TEMP).

These registers should always be accessed 16 bits at a time. If two consecutive byte-size MOV instructions are used, the upper byte must be accessed first and the lower byte second. Data will not be transferred correctly if only the upper byte or only the lower byte is accessed.

1. Write access

Write access to the upper byte results in transfer of the upper-byte write data to TEMP. Next, write access to the lower byte results in transfer of the data in TEMP to the upper register byte, and direct transfer of the lower-byte write data to the lower register byte.

Figure 9-18 shows an example of the writing of H'AA55 to FRC.

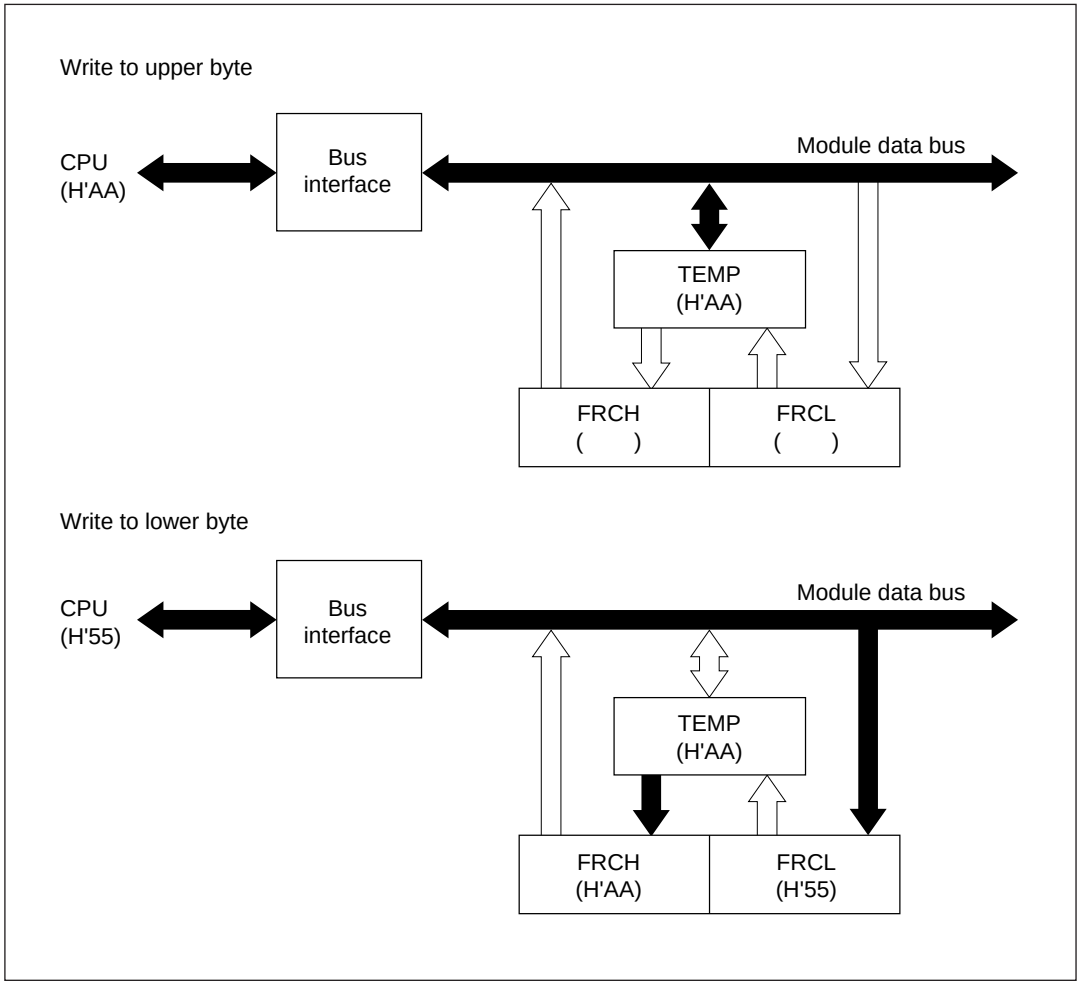


Figure 9-18 Write Access to FRC (CPU → FRC)

2. Read access

In access to FRC and ICRA to ICRD, when the upper byte is read the upper-byte data is transferred directly to the CPU and the lower-byte data is transferred to TEMP. Next, when the lower byte is read, the lower-byte data in TEMP is transferred to the CPU.

In access to OCRA or OCRB, when the upper byte is read the upper-byte data is transferred directly to the CPU, and when the lower byte is read the lower-byte data is transferred directly to the CPU.

Figure 9-19 shows an example of the reading of FRC when FRC contains H'AAFF.

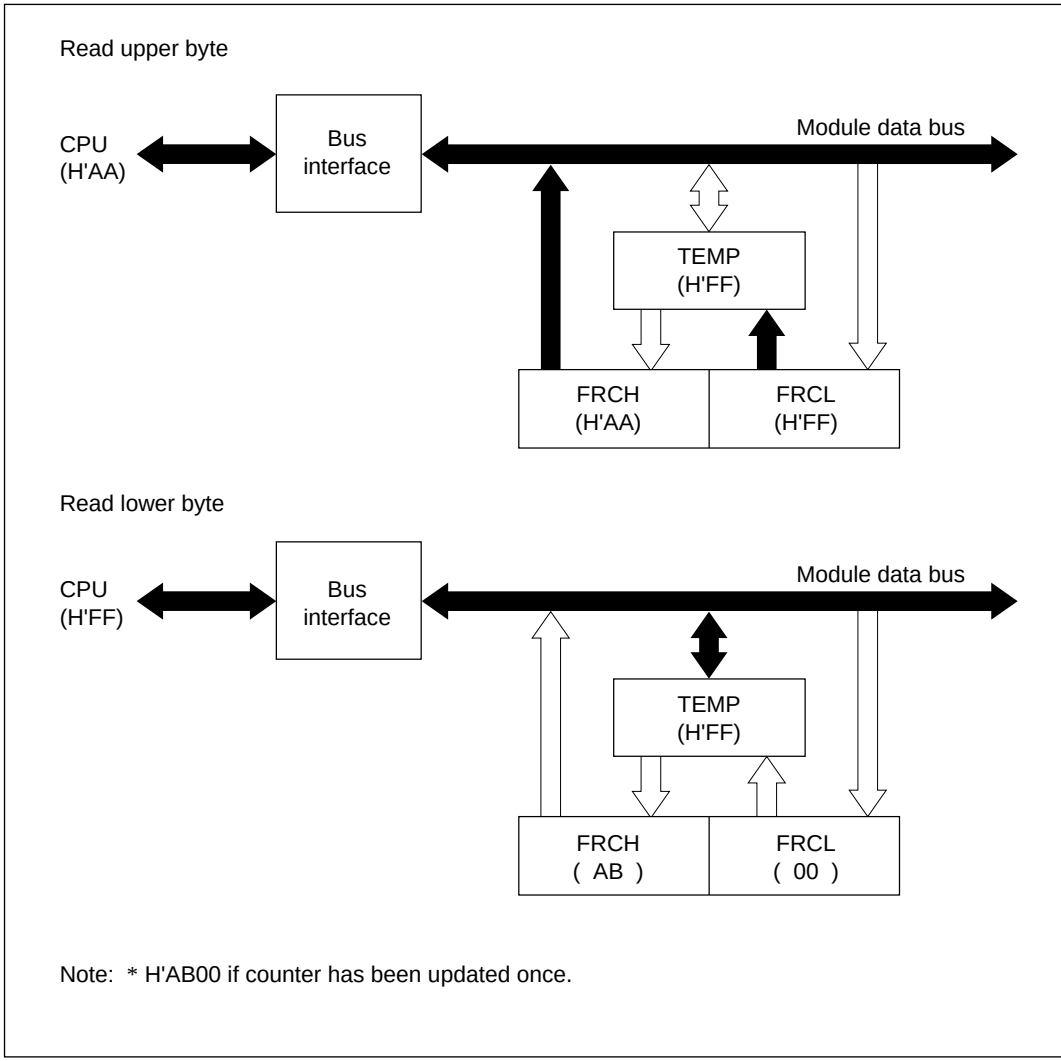


Figure 9-19 Read Access to FRC (FRC → CPU)

9.5.4 Timer Operation

1. Timer operation

- Output compare operation

  Following a reset, FRC is initialized to H'0000 and starts counting up. Bits CKS1 and CKS0 in TCRX can select one of three internal clock sources or an external clock for input to FRC. The FRC contents are compared constantly with OCRA and OCRB. When a match occurs, the output at pin FTOA or FTOB goes to the level selected by OLVLA or OLVLB in TOCR. Following a reset, the output at both FTOA and FTOB is 0 until the first compare match. If CCLRA is set to 1 in TCSRX, compare match A clears FRC to H'0000.

- Input capture operation

  Following a reset, FRC is initialized to H'0000 and starts counting up. Bits CKS1 and CKS0 in TCRX can select one of three internal clock sources or an external clock for input to FRC. When the edges selected by bits IEDGA to IEDGD in TCRX are input at pins FTIA to FTID, the FRC value is transferred to ICRA to ICRD, and ICFA to ICFD are set in TCSRX. If bits ICIAE to ICIDE are set to 1 in TIER, a CPU interrupt is requested.

  If bits BUFEA and BUFEB are set to 1 in TCRX, ICRC and ICRD operate as buffer registers for ICRA or ICRB. When the edges selected by bits IEDGA to IEDGD in TCRX are input at pins FTIA and FTIB, the FRC value is transferred to ICRA or ICRB, and the previous value in ICRA or ICRB is transferred to ICRC or ICRD. Simultaneously, ICFA or ICFB is set in TCSRX. If bit ICIAE or ICIBE is set to 1 in TIER, a CPU interrupt is requested.

2. FRC count timing

FRC is incremented by clock input. Bits CKS1 and CKS0 in TCRX can select one of three internal clock sources (ø/2, ø/8, ø/32) or an external clock.

- Internal clock

  Bits CKS1 and CKS0 in TCRX select one of three internal clock sources (ø/2, ø/8, ø/32) created by dividing the system clock (ø). Figure 9-20 shows the increment timing.

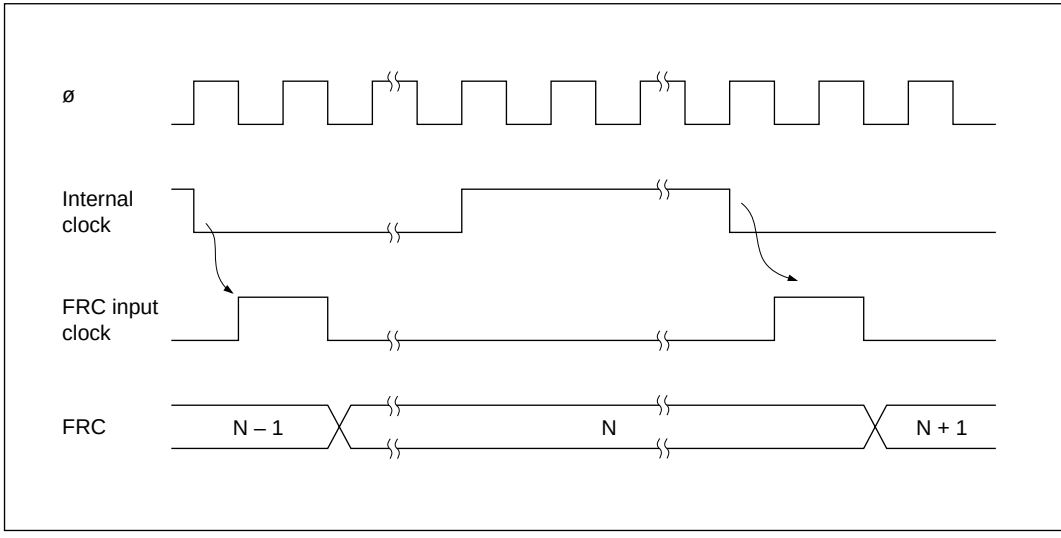


Figure 9-20 Increment Timing with Internal Clock

- External clock

  External clock input is selected when bits CKS1 and CKS0 are both set to 1 in TCRX. FRC increments on the rising edge of the external clock. An external pulse width of at least 1.5 system clocks (ø) is necessary. Shorter pulses will not be counted correctly. Figure 9-21 shows the timing.

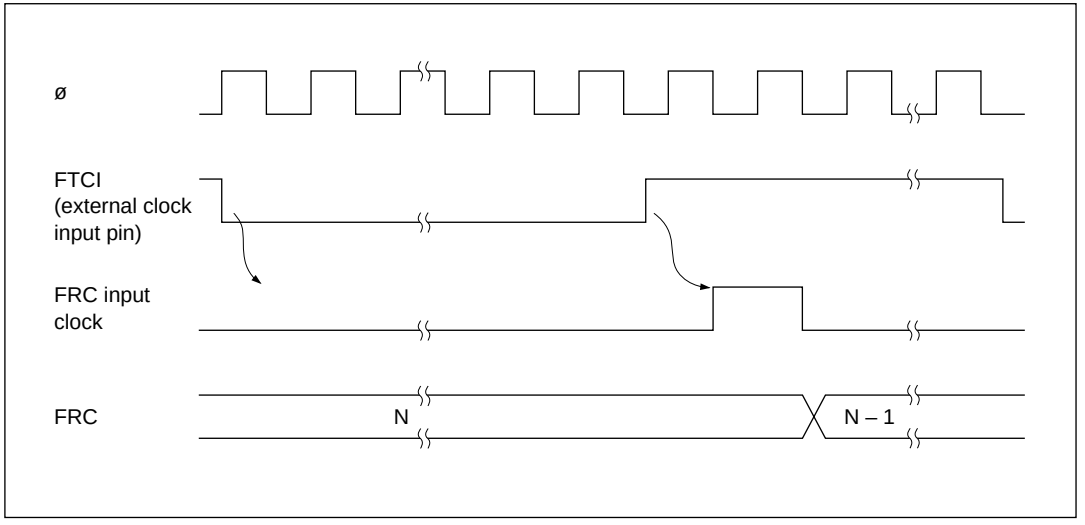


Figure 9-21 Increment Timing with External Clock

3. Output compare timing

When a compare match occurs, the output level selected by the OLVL bit in TOCR is output at pin FTOA or FTOB. Figure 9-22 shows the output timing for output compare A.

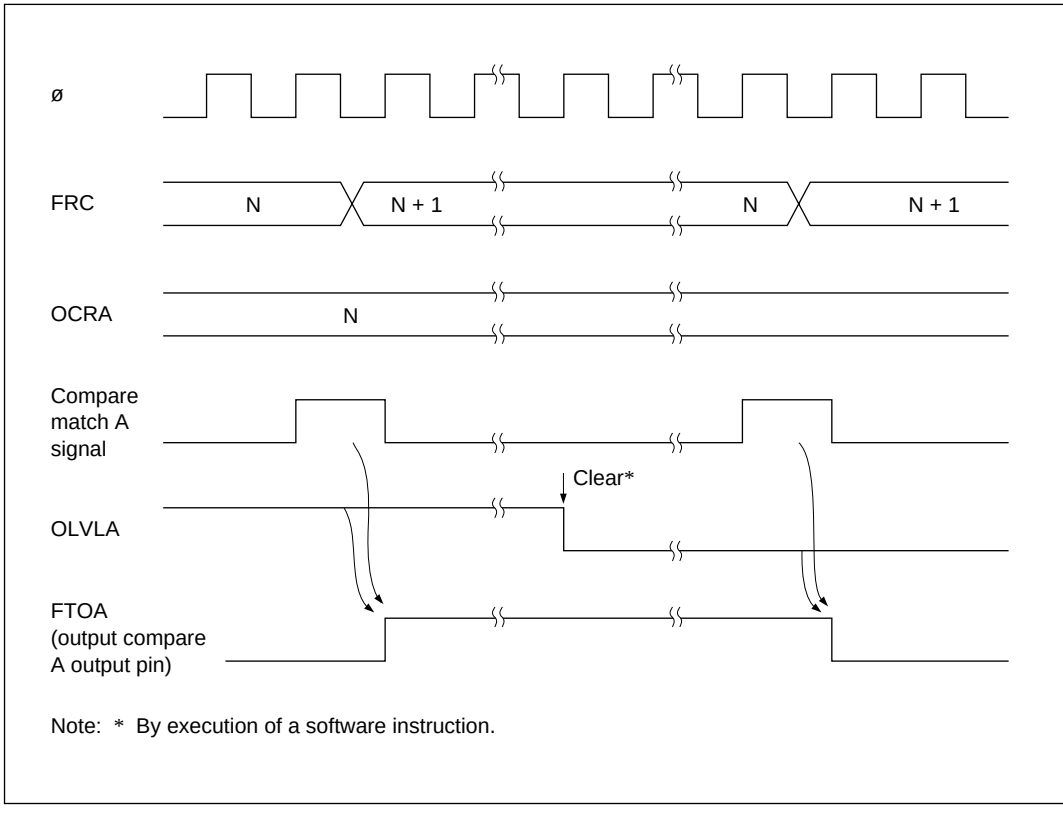


Figure 9-22 Output Compare A Output Timing

4. FRC clear timing

FRC can be cleared by compare match A. Figure 9-23 shows the timing.

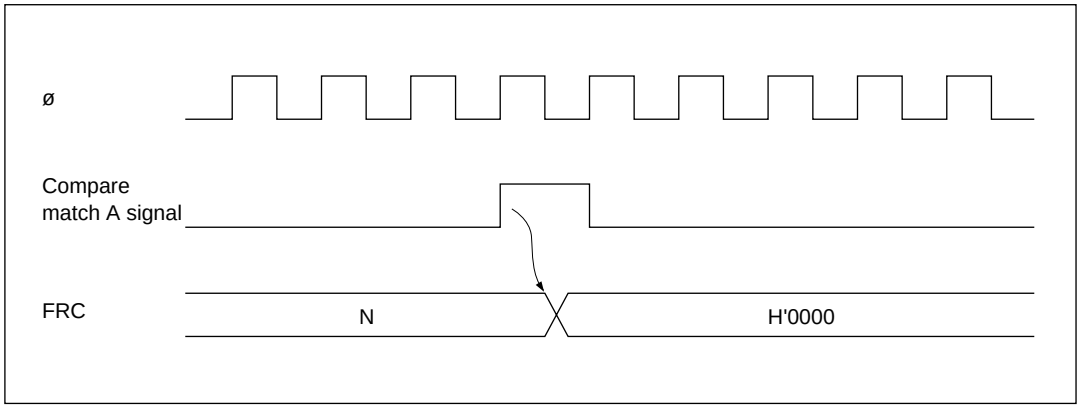


Figure 9-23 Clear Timing by Compare Match A

5. Input capture timing

- Input capture timing

  The rising or falling edge is selected for input capture by bits IEDGA to IEDGD in TCRX. Figure 9-24 shows the timing when the rising edge is selected (IEDGA/B/C/D = 1).

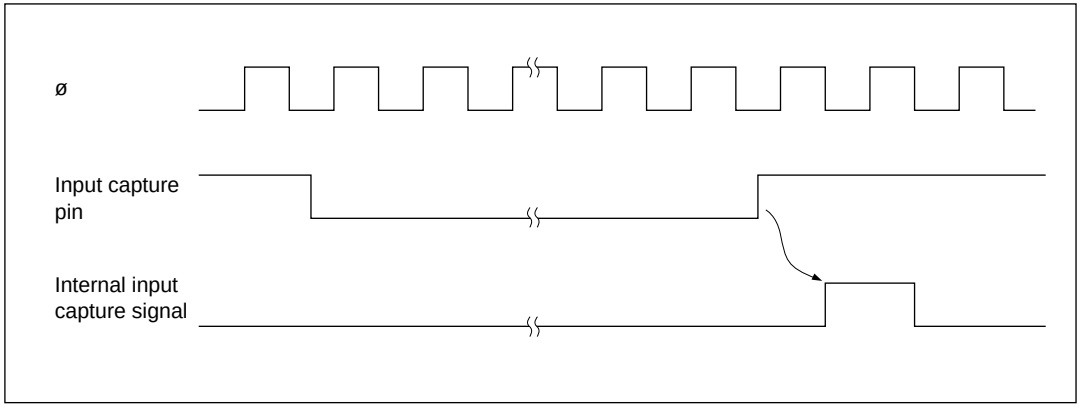


Figure 9-24 Input Capture Signal Timing (Normal Case)

If the input at the input capture pin occurs while the upper byte of the corresponding input capture register (ICRA to ICRD) is being read, the internal input capture signal is delayed by one system clock (ø). Figure 9-25 shows the timing.

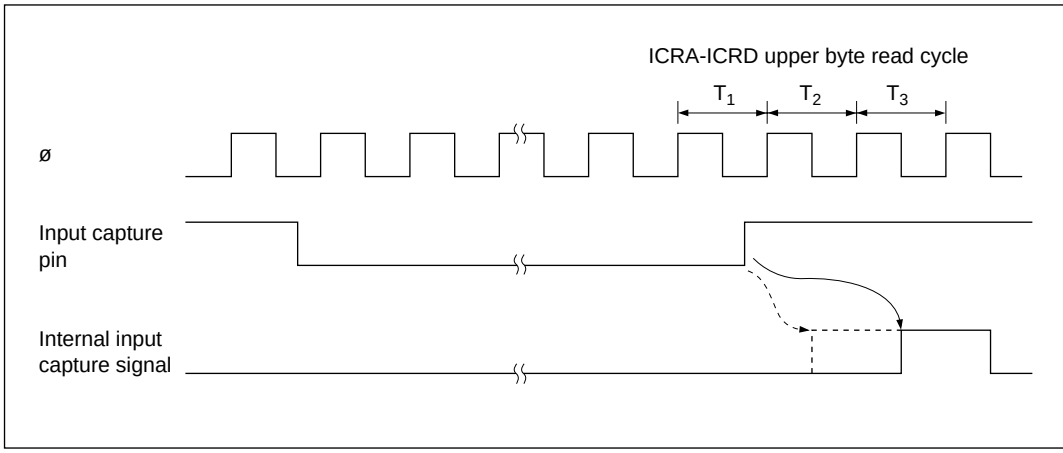


Figure 9-25 Input Capture Signal Timing (during ICRA-ICRD Read)

- Buffered input capture timing

  Input capture can be buffered by using ICRC or ICRD as a buffer for ICRA or ICRB. Figure 9-26 shows the timing when ICRA is buffered by ICRC (BUFEA = 1) and both the rising and falling edges are selected (IEDGA = 1 and IEDGC = 0, or IEDGA = 0 and IEDGC = 1).

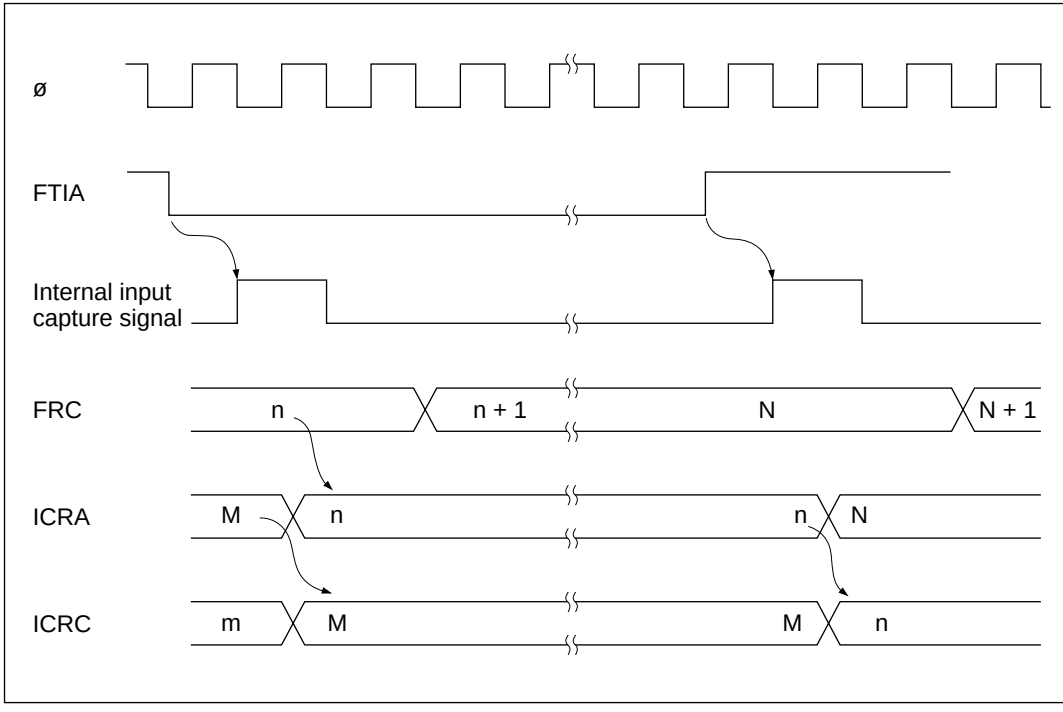


Figure 9-26 Buffered Input Capture Timing (Normal Case)

When ICRC or ICRD is used as a buffer register, the input capture flag is still set by the selected edge of the input capture input signal. For example, if ICRC is used to buffer ICRA, when the edge transition selected by the IEDGC bit occurs at the input capture pin, ICFC will be set, and if the ICIEC bit is set to 1, an interrupt will be requested. The FRC value will not be transferred to ICRC, however.

In buffered operation, if the upper byte of one of the two registers that receives a data transfer (ICRA and ICRC, or ICRB and ICRD) is being read when an internal input capture signal would normally occur, the internal input capture signal will be delayed by one system clock (ø). Figure 9-27 shows the case when BUFEA = 1.

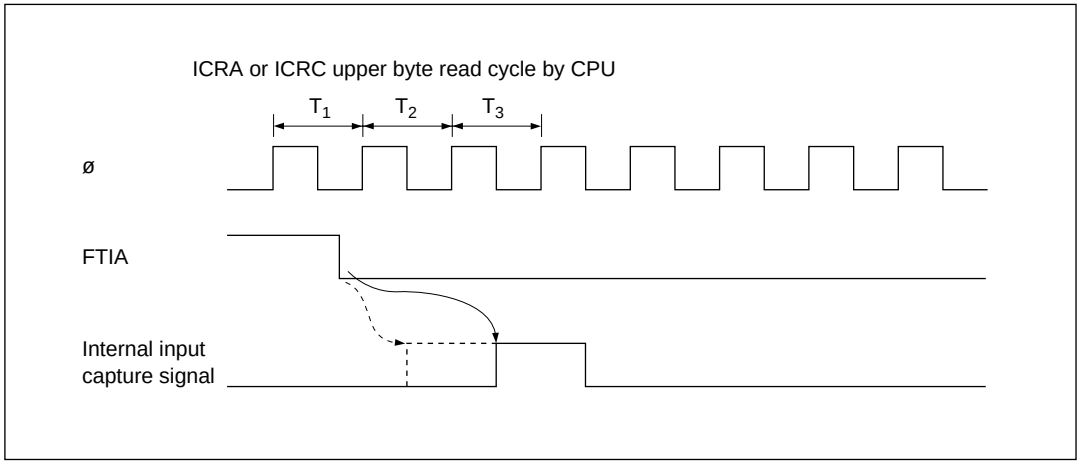


Figure 9-27 Buffered Input Capture Signal Timing (during ICRA or ICRC Read)

6. Input capture flag (ICFA to ICFD) set timing

Figure 9-28 shows the timing when an input capture flag (ICFA to ICFD) is set to 1 and the FRC value is transferred to the corresponding input capture register (ICRA to ICRD).

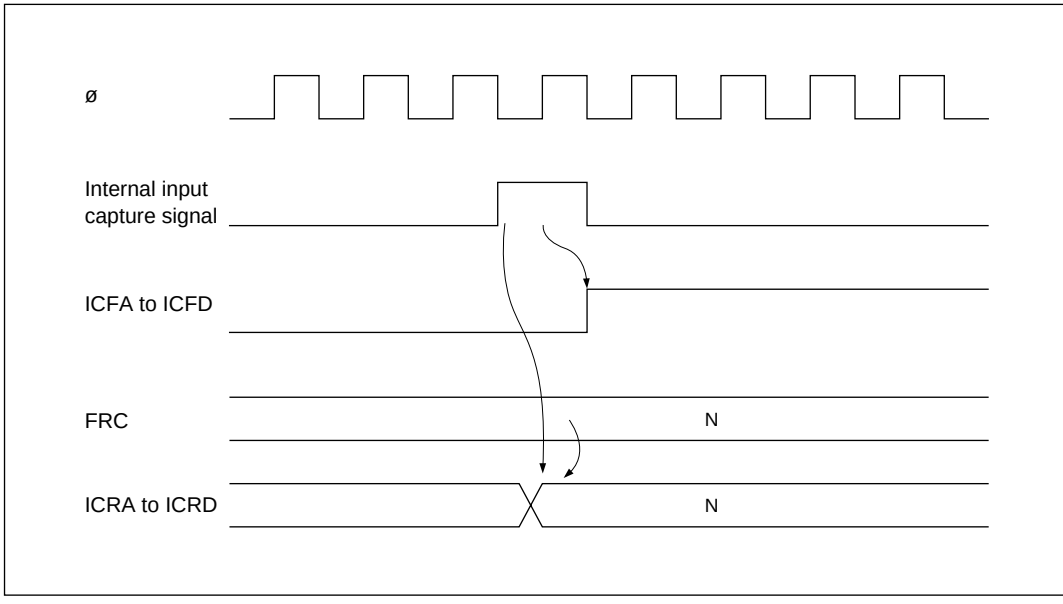


Figure 9-28 ICFA to ICFD Set Timing

7. Output compare flag (OCFA or OCFB) set timing

OCFA and OCFB are set to 1 by internal compare match signals that are output when FRC matches OCRA or OCRB. The compare match signal is generated in the last state during which the values match (when FRC is updated from the matching value to a new value). When FRC matches OCRA or OCRB, the compare match signal is not generated until the next counter clock. Figure 9-29 shows the OCFA and OCFB set timing.

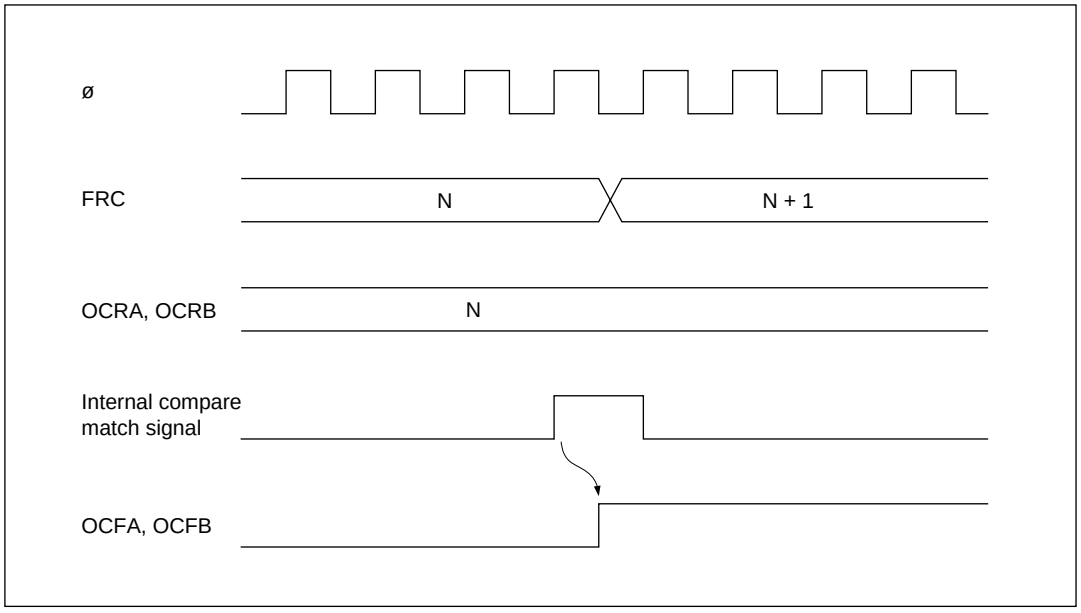


Figure 9-29 OCFA and OCFB Set Timing

8. Overflow flag (OVF) set timing

OVF is set to 1 when FRC overflows from H'FFFF to H'0000. Figure 9-30 shows the timing.

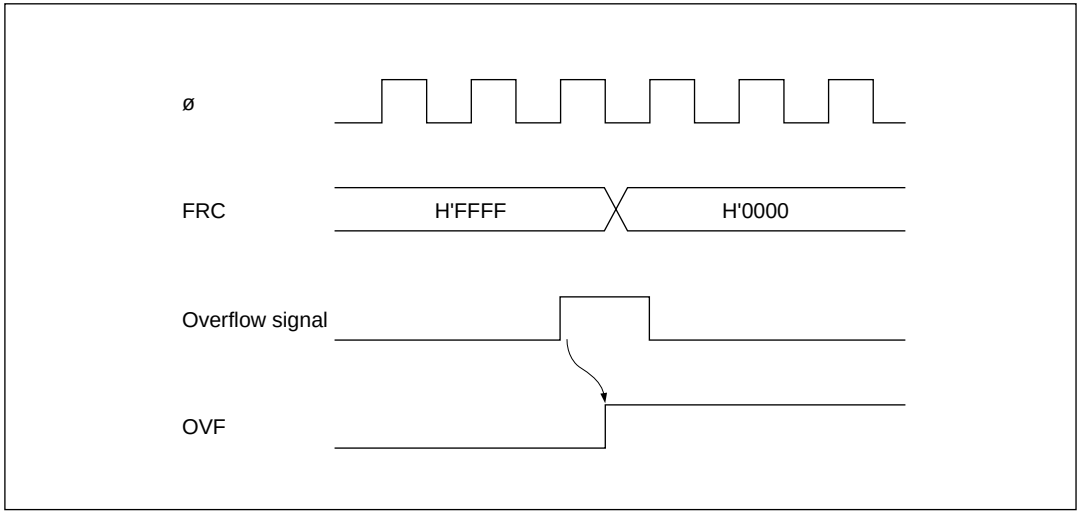


Figure 9-30 OVF Set Timing

9.5.5 Timer X Operation Modes

Figure 9-17 shows the timer X operation modes.

Table 9-17 Timer X Operation Modes

Operation Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
FRC	Reset	Functions	Functions	Reset	Reset	Reset	Reset
OCRA, OCRB	Reset	Functions	Functions	Reset	Reset	Reset	Reset
ICRA to ICRD	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TIER	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TCRX	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TOCR	Reset	Functions	Functions	Reset	Reset	Reset	Reset
TCSRX	Reset	Functions	Functions	Reset	Reset	Reset	Reset

9.5.6 Interrupt Sources

Timer X has three types of interrupts and seven interrupt sources: ICIA to ICID, OCIA, OCIB, and FOVI. Table 9-18 lists the sources of interrupt requests. Each interrupt source can be enabled or disabled by an interrupt enable bit in TIER. Although all seven interrupts share the same vector, they have individual interrupt flags, so software can discriminate the interrupt source.

Table 9-18 Timer X Interrupt Sources

Interrupt	Description	Vector Address
ICIA	Interrupt requested by ICFA	H'0020
ICIB	Interrupt requested by ICFB	
ICIC	Interrupt requested by ICFC	
ICID	Interrupt requested by ICFD	
OCIA	Interrupt requested by OCFA	
OCIB	Interrupt requested by OCFB	
FOVI	Interrupt requested by OVF	

9.5.7 Timer X Application Example

Figure 9-31 shows an example of the output of pulse signals with a 50% duty cycle and arbitrary phase offset. To set up this output:

- Set bit CCLRA to 1 in TCSRX.
- Have software invert the OLVLA and OLVLB bits at each corresponding compare match.

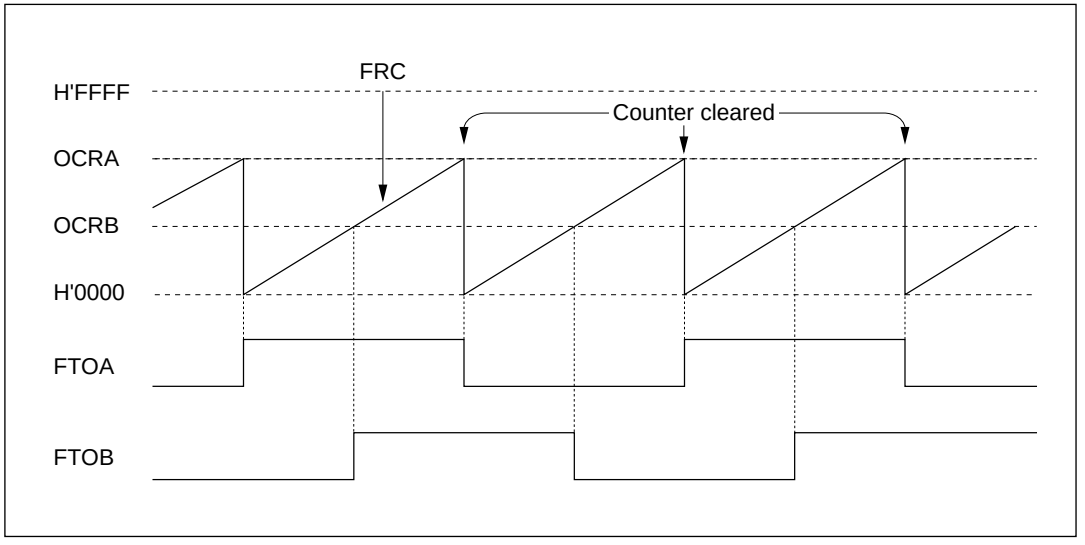


Figure 9-31 Pulse Output Example

9.5.8 Application Notes

The following types of contention can occur in timer X operation.

1. Contention between FRC write and counter clear

If an FRC clear signal is generated in the T_3 state of a write cycle to the lower byte of FRC, clearing takes precedence and the write to the counter is not carried out. Figure 9-32 shows the timing.

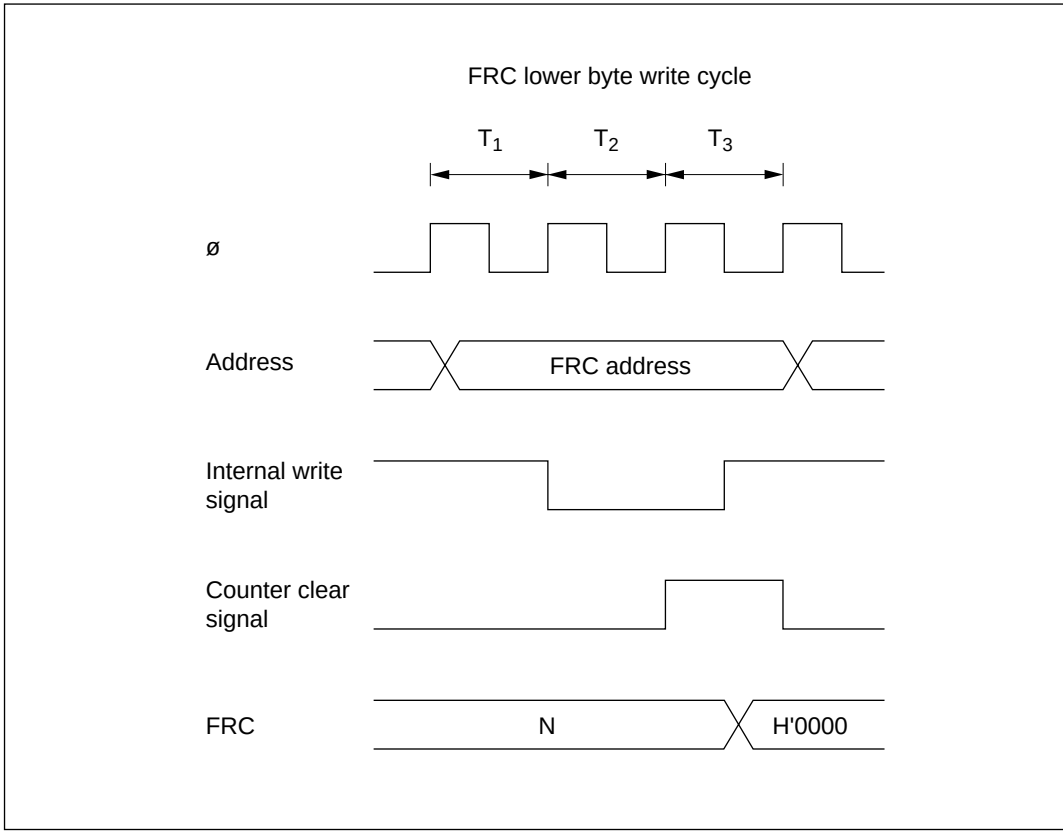


Figure 9-32 Contention between FRC Write and Clear

2. Contention between FRC write and increment

If an FRC increment clock signal is generated in the T_3 state of a write cycle to the lower byte of FRC, the write takes precedence and the counter is not incremented. Figure 9-33 shows the timing.

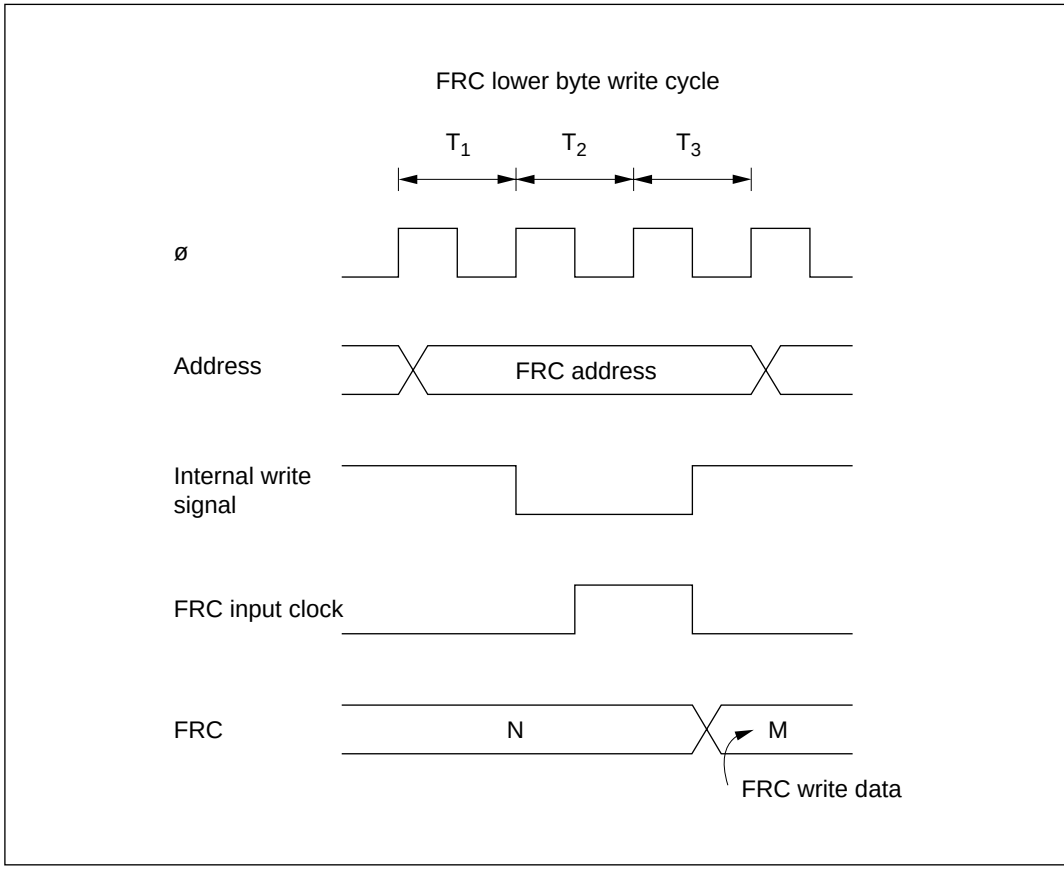


Figure 9-33 Contention between FRC Write and Increment

3. Contention between OCR write and compare match

If a compare match is generated in the T_3 state of a write cycle to the lower byte of OCRA or OCRB, the write to OCRA or OCRB takes precedence and the compare match signal is inhibited. Figure 9-34 shows the timing.

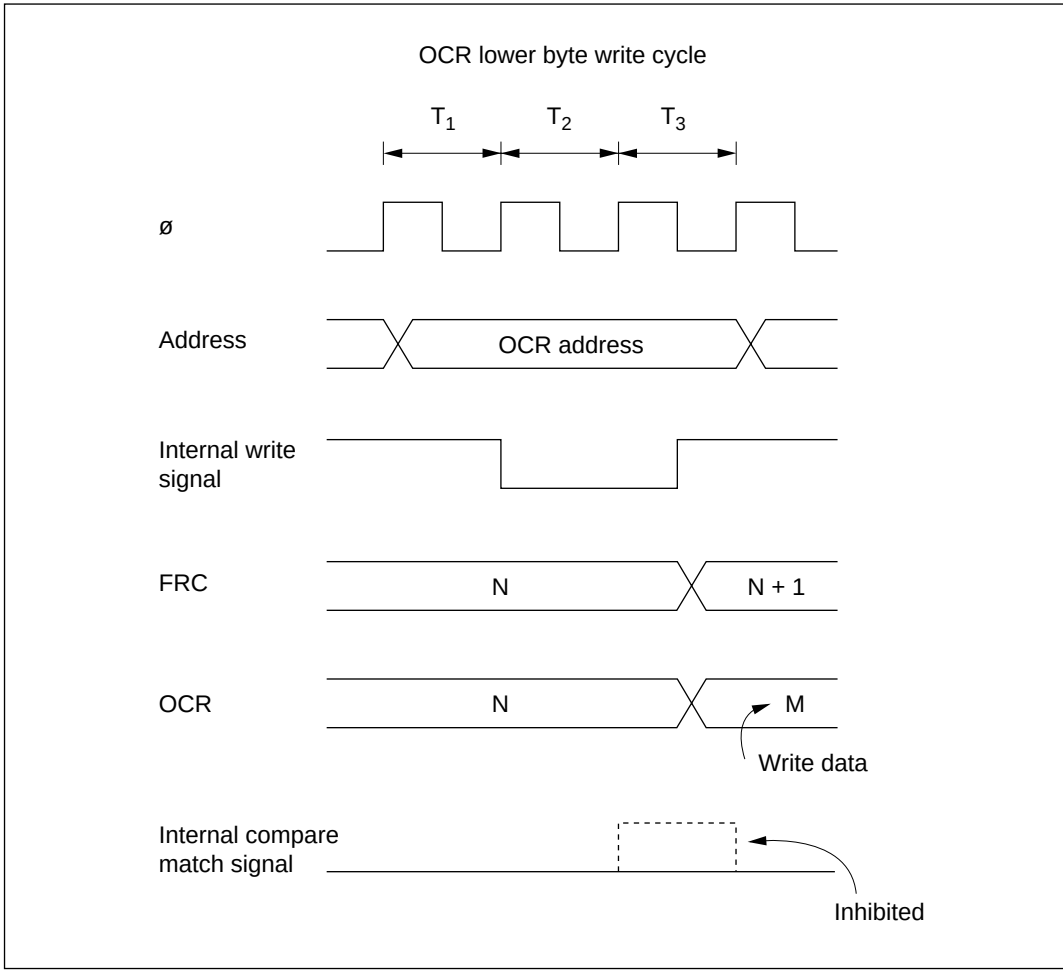


Figure 9-34 Contention between OCR Write and Compare Match

4. Internal clock switching and counter operation

Depending on the timing, FRC may be incremented by a switch between different internal clock sources. Table 9-19 shows the relation between internal clock switchover timing (by writing to bits CKS1 and CKS0) and FRC operation.

When FRC is internally clocked, an increment pulse is generated from the falling edge of an internal clock signal, which is divided from the system clock (ø). For this reason, in a case like No. 3 in table 9-19 where the switch is from a high clock signal to a low clock signal, the switchover is seen as a falling edge, causing FRC to increment.

FRC can also be incremented by a switch between internal and external clocks.

Table 9-19 Internal Clock Switching and FRC Operation

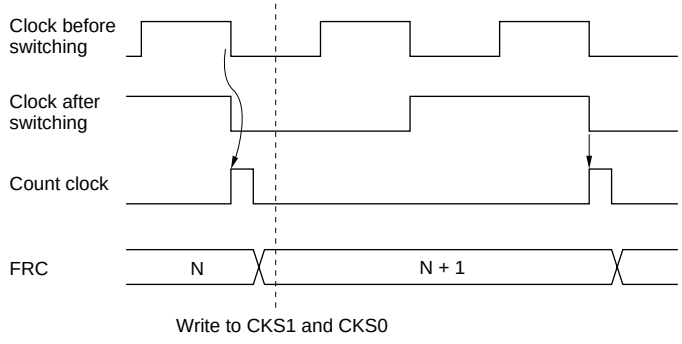
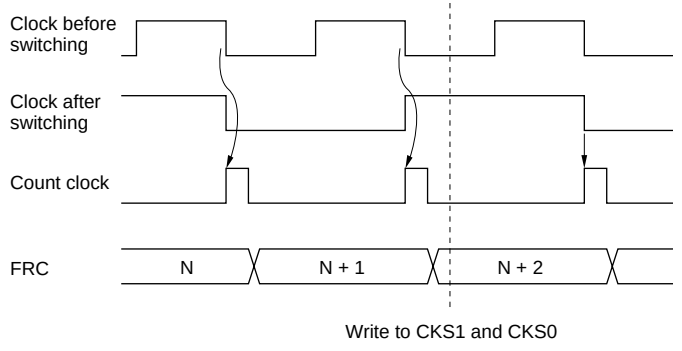
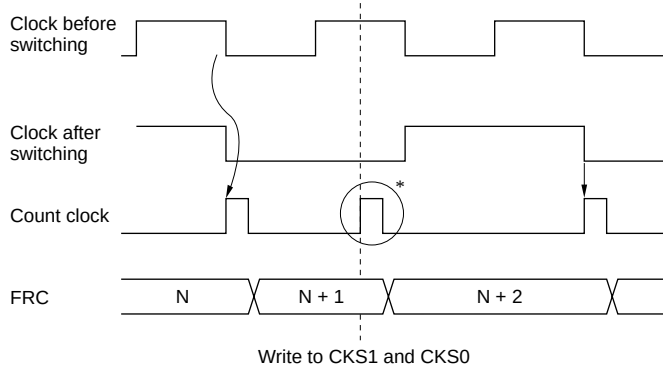
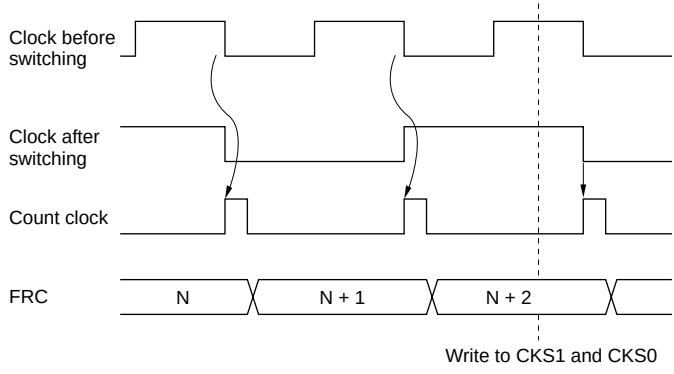
No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	FRC Operation
1	Goes from low level to low level	
2	Goes from low to high	

Table 9-19 Internal Clock Switching and FRC Operation (cont)

No.	Clock Levels Before and After Modifying Bits CKS1 and CKS0	FRC Operation
3	Goes from high level to low level	
4	Goes from high to high	

Note: * The switchover is seen as a falling edge, and FRC is incremented.

9.6 Watchdog Timer

9.6.1 Overview

The watchdog timer has an 8-bit counter that is incremented by an input clock. If a system runaway allows the counter value to overflow before being rewritten, the watchdog timer can reset the chip internally.

1. Features

Features of the watchdog timer are given below.

- Incremented by internal clock source (ø/8192).
- A reset signal is generated when the counter overflows. The overflow period can be set from from 1 to 256 times 8192/ø (from approximately 2 ms to 500 ms when ø = 4.19 MHz).

2. Block diagram

Figure 9-35 shows a block diagram of the watchdog timer.

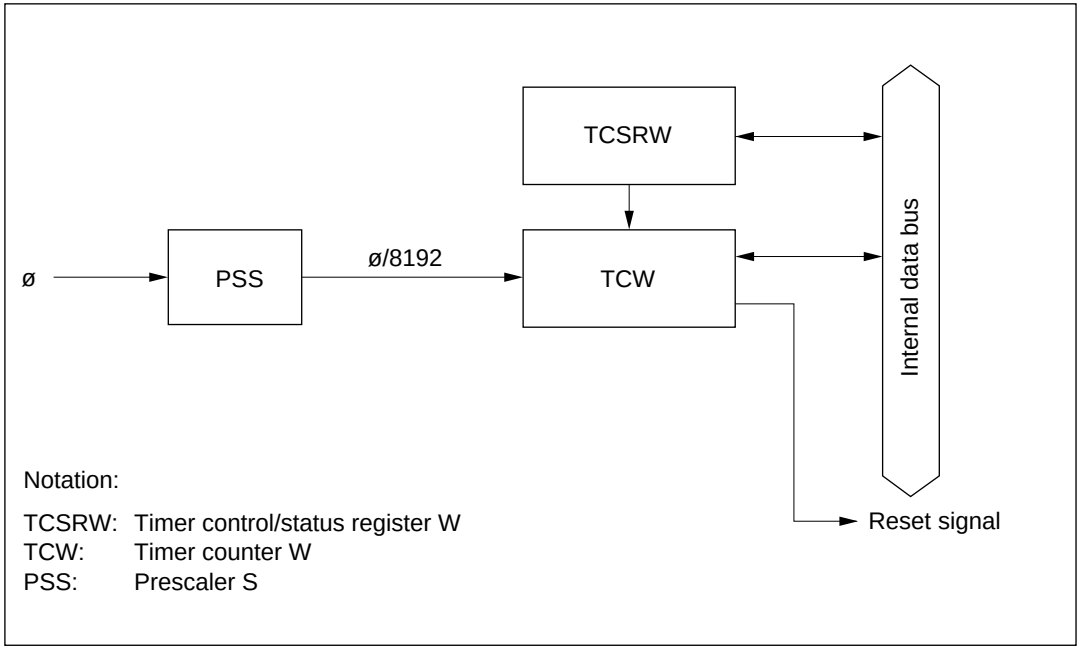


Figure 9-35 Block Diagram of Watchdog Timer

3. Register configuration

Table 9-20 shows the register configuration of the watchdog timer.

Table 9-20 Watchdog Timer Registers

Name	Abbrev.	R/W	Initial Value	Address
Timer control/status register W	TCSRW	R/W	H'AA	H'FFBE
Timer counter W	TCW	R/W	H'00	H'FFBF

9.6.2 Register Descriptions

1. Timer control/status register W (TCSRW)

Bit	7	6	5	4	3	2	1	0
	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST
Initial value	1	0	1	0	1	0	1	0
Read/Write	R	R/W*	R	R/W*	R	R/W*	R	R/W*

Note: * Write is permitted only under certain conditions, which are given in the descriptions of the individual bits.

TCSRW is an 8-bit read/write register that controls write access to TCW and TCSRW itself, controls watchdog timer operations, and indicates operating status.

Bit 7: Bit 6 write inhibit (B6WI)

Bit 7 controls the writing of data to bit 6 in TCSRW.

Bit 7 B6WI	Description	
0	Bit 6 is write-enabled	
1	Bit 6 is write-protected	(initial value)

This bit is always read as 1. Data written to this bit is not stored.

Bit 6: Timer counter W write enable (TCWE)

Bit 6 controls the writing of data to TCW.

Bit 6 TCWE	Description	
0	Data cannot be written to TCW	(initial value)
1	Data can be written to TCW	

Bit 5: Bit 4 write inhibit (B4WI)

Bit 5 controls the writing of data to bit 4 in TCSRW.

Bit 5 B4WI	Description	
0	Bit 4 is write-enabled	(initial value)
1	Bit 4 is write-protected	

This bit is always read as 1. Data written to this bit is not stored.

Bit 4: Timer control/status register W write enable (TCSRWE)

Bit 4 controls the writing of data to TCSRW bits 2 and 0.

Bit 4 TCSRWE	Description	
0	Data cannot be written to bits 2 and 0	(initial value)
1	Data can be written to bits 2 and 0	

Bit 3: Bit 2 write inhibit (B2WI)

Bit 3 controls the writing of data to bit 2 in TCSRW.

Bit 3 B2WI	Description	
0	Bit 2 is write-enabled	
1	Bit 2 is write-protected	(initial value)

This bit is always read as 1. Data written to this bit is not stored.

Bit 2: Watchdog timer on (WDON)

Bit 2 enables watchdog timer operation.

Bit 2 WDON	Description	
0	Watchdog timer operation is disabled Clearing conditions: Reset, or when TCSRWE = 1 and 0 is written in both B2WI and WDON	(initial value)
1	Watchdog timer operation is enabled Setting conditions: When TCSRWE = 1 and 0 is written in B2WI and 1 is written in WDON	

Counting starts when this bit is set to 1, and stops when this bit is cleared to 0.

Bit 1: Bit 0 write inhibit (B0WI)

Bit 1 controls the writing of data to bit 0 in TCSRW.

Bit 1 B0WI	Description	
0	Bit 0 is write-enabled	
1	Bit 0 is write-protected	(initial value)

This bit is always read as 1. Data written to this bit is not stored.

Bit 0: Watchdog timer reset (WRST)

Bit 0 indicates that TCW has overflowed, generating an internal reset signal. The internal reset signal generated by the overflow resets the entire chip. WRST is cleared to 0 by a reset from the $\overline{\text{RES}}$ pin, or when software writes 0.

Bit 0 WRST	Description	
0	Clearing conditions: • Reset by $\overline{\text{RES}}$ pin • When TCSRWE = 1, and 0 is written in both B0WI and WRST	(initial value)
1	Setting conditions: When TCW overflows and an internal reset signal is generated	

2. Timer counter W (TCW)

Bit	7	6	5	4	3	2	1	0
	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCW is an 8-bit read/write up-counter, which is incremented by internal clock input. The input clock is ø/8192. The TCW value can always be written or read by the CPU.

When TCW overflows from H'FF to H'00, an internal reset signal is generated and WRST is set to 1 in TCSRW. Upon reset, TCW is initialized to H'00.

9.6.3 Timer Operation

The watchdog timer has an 8-bit counter (TCW) that is incremented by clock input (ø/8192). When TCSRWE = 1 in TCSRW, if 0 is written in B2WI and 1 is simultaneously written in WDON, TCW starts counting up. When the TCW count value reaches H'FF, the next clock input causes the watchdog timer to overflow and generates an internal reset signal. The internal reset signal is output for 512 clock cycles of the $ø_{OSC}$ clock. It is possible to write to TCW, causing TCW to count up from the written value. The overflow period can be set in the range from 1 to 256 input clocks, depending on the value written in TCW.

Figure 9-36 shows an example of watchdog timer operations.

Example: ø = 4 MHz and the desired overflow period is 30 ms.

$$\frac{4 \times 10^6}{8192} \times 30 \times 10^{-3} = 14.6$$

The value set in TCW should therefore be 256 – 15 = 241 (H'F1).

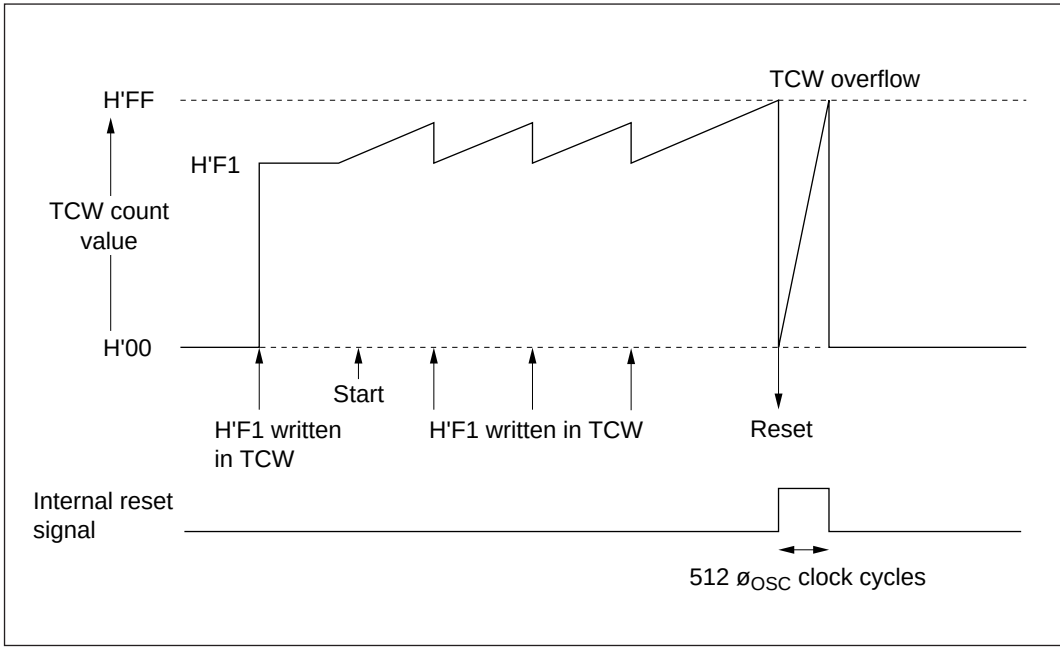


Figure 9-36 Typical Watchdog Timer Operations (Example)

9.6.4 Watchdog Timer Operation States

Table 9-21 summarizes the watchdog timer operation states.

Table 9-21 Watchdog Timer Operation States

Operation Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby
TCW	Reset	Functions	Functions	Halted	Halted	Halted	Halted
TCSRW	Reset	Functions	Functions	Retained	Retained	Retained	Retained

Section 10 Serial Communication Interface

10.1 Overview

The H8/3657 Series is provided with a two-channel serial communication interface (SCI). Table 10-1 summarizes the functions and features of the two SCI channels.

Table 10-1 Serial Communication Interface Functions

Channel	Functions	Features
SCI1	Synchronous serial transfer • Choice of 8-bit or 16-bit data length • Continuous clock output	• Choice of 8 internal clocks (ø/1024 to ø/2) or external clock • Open drain output possible • Interrupt requested at completion of transfer
SCI3	Synchronous serial transfer • 8-bit data length • Send, receive, or simultaneous send/receive Asynchronous serial transfer • Multiprocessor communication • Choice of 7-bit or 8-bit data length • Choice of 1 or 2 stop bits • Parity addition	• On-chip baud rate generator • Receive error detection • Break detection • Interrupt requested at completion of transfer or error

10.2 SCI1

10.2.1 Overview

Serial communication interface 1 (SCI1) performs synchronous serial transfer of 8-bit or 16-bit data. SSB (Synchronized Serial Bus) communication is also provided, enabling multiple ICs to be controlled.

1. Features

- Choice of 8-bit or 16-bit data length
- Choice of eight internal clock sources (ø/1024, ø/256, ø/64, ø/32, ø/16, ø/8, ø/4, ø/2) or an external clock
- Interrupt requested at completion of transfer
- Choice of HOLD mode or LATCH mode in SSB mode

2. Block diagram

Figure 10-1 shows a block diagram of SCI1.

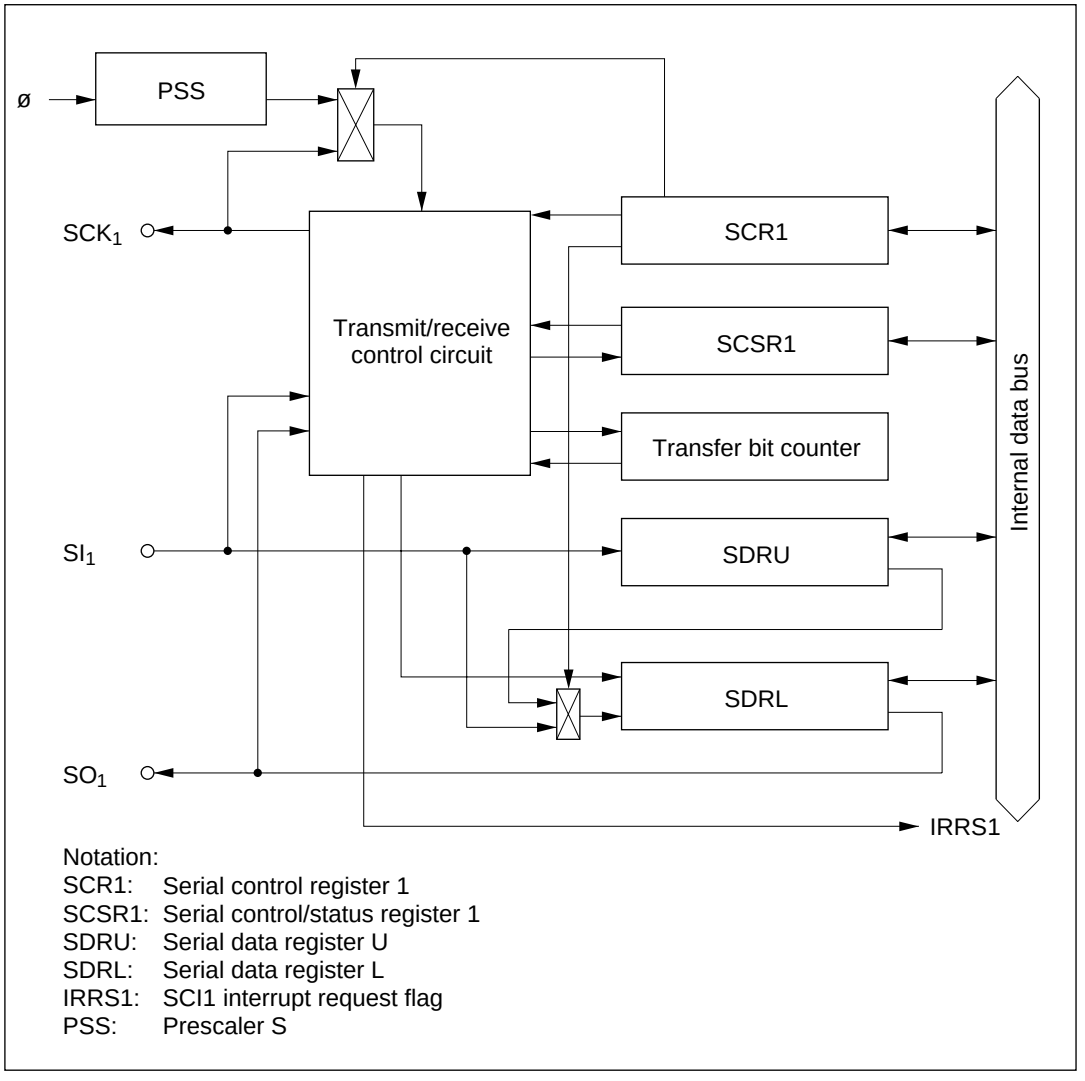


Figure 10-1 SCI1 Block Diagram

3. Pin configuration

Table 10-2 shows the SCI1 pin configuration.

Table 10-2 Pin Configuration

Name	Abbrev.	I/O	Function
SCI1 clock pin	SCK_1	I/O	SCI1 clock input or output
SCI1 data input pin	SI_1	Input	SCI1 receive data input
SCI1 data output pin	SO_1	Output	SCI1 transmit data output

4. Register configuration

Table 10-3 shows the SCI1 register configuration.

Table 10-3 SCI1 Registers

Name	Abbrev.	R/W	Initial Value	Address
Serial control register 1	SCR1	R/W	H'00	H'FFA0
Serial control status register 1	SCSR1	R/W	H'9C	H'FFA1
Serial data register U	SDRU	R/W	Not fixed	H'FFA2
Serial data register L	SDRL	R/W	Not fixed	H'FFA3

10.2.2 Register Descriptions

1. Serial control register 1 (SCR1)

Bit	7	6	5	4	3	2	1	0
	SNC1	SNC0	MRKON	LTCH	CKS3	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

SCR1 is an 8-bit read/write register for selecting the operation mode, the transfer clock source, and the prescaler division ratio.

Upon reset, SCR1 is initialized to H'00. Writing to this register during a transfer stops the transfer.

Bits 7 and 6: Operation mode select 1, 0 (SNC1, SNC0)

Bits 7 and 6 select the operation mode.

Bit 7 SNC1	Bit 6 SNC0	Description	
0	0	8-bit synchronous transfer mode	(initial value)
0	1	16-bit synchronous transfer mode	
1	0	Continuous clock output mode*1	
1	1	Reserved*2	

Notes: 1. Pins SI_1 and SO_1 should be used as general input or output ports.
2. Don't set bits SNC1 and SNC0 to 11.

Bits 5: TAIL MARK control (MRKON)

Bit 5 controls TAIL MARK output after an 8- or 16-bit data transfer.

Bit 5 MRKON	Description	
0	TAIL MARK is not output (synchronous mode)	(initial value)
1	TAIL MARK is output (SSB mode)	

Bits 4: LATCH TAIL select (LTCH)

Bit 4 selects whether LATCH TAIL or HOLD TAIL is output as TAIL MARK when bit MRKON is set to 1 (SSB mode).

Bit 4 LTCH	Description	
0	HOLD TAIL is output	(initial value)
1	LATCH TAIL is output	

Bit 3: Clock source select (CKS3)

Bit 3 selects the clock source and sets pin SCK_1 as an input or output pin.

Bit 3 CKS3	Description	
0	Clock source is prescaler S, and pin SCK_1 is output pin	(initial value)
1	Clock source is external clock, and pin SCK_1 is input pin*	

Note: * Input an external clock equivalent to a frequency lower than ø/4.

Bits 2 to 0: Clock select (CKS2 to CKS 0)

When CKS3 = 0, bits 2 to 0 select the prescaler division ratio and the serial clock cycle.

Bit 2 CKS2	Bit 1 CKS1	Bit 0 CKS0	Prescaler Division	Serial Clock Cycle ø = 5 MHz	Serial Clock Cycle ø = 2.5 MHz
0	0	0	ø/1024 (initial value)	204.8 µs	409.6 µs
0	0	1	ø/256	51.2 µs	102.4 µs
0	1	0	ø/64	12.8 µs	25.6 µs
0	1	1	ø/32	6.4 µs	12.8 µs
1	0	0	ø/16	3.2 µs	6.4 µs
1	0	1	ø/8	1.6 µs	3.2 µs
1	1	0	ø/4	0.8 µs	1.6 µs
1	1	1	ø/2	—	0.8 µs

2. Serial control/status register 1 (SCSR1)

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

Note: * Only a write of 0 for flag clearing is possible.

SCSR1 is an 8-bit read/write register indicating operation status and error status.

Upon reset, SCSR1 is initialized to H'9C.

Bit 7: Reserved bit

Bit 7 is reserved; it is always read as 1, and cannot be modified.

Bit 6: Extended data bit (SOL)

Bit 6 sets the SO_1 output level. When read, SOL returns the output level at the SO_1 pin. After completion of a transmission, SO_1 continues to output the value of the last bit of transmitted data. The SO_1 output can be changed by writing to SOL before or after a transmission. The SOL bit setting remains valid only until the start of the next transmission. The setting is also invalid in SSB mode. To control the level of the SO_1 pin after transmission ends, it is necessary to write to the SOL bit at the end of each transmission. Do not write to this register while transmission is in progress, because that may cause a malfunction.

Bit 6 SOL	Description	
0	Read: SO_1 pin output level is low	(initial value)
	Write: SO_1 pin output level changes to low	
1	Read: SO_1 pin output level is high	
	Write: SO_1 pin output level changes to high	

Bit 5: Overrun error flag (ORER)

When an external clock is used, bit 5 indicates the occurrence of an overrun error. If a clock pulse is input after transfer completion, this bit is set to 1 indicating an overrun. If noise occurs during a transfer, causing an extraneous pulse to be superimposed on the normal serial clock, incorrect data may be transferred.

Bit 5 ORER	Description	
0	Clearing conditions: After reading ORER = 1, cleared by writing 0 to ORER	(initial value)
1	Setting conditions: Set if a clock pulse is input after transfer is complete, when an external clock is used	

Bits 4 to 2: Reserved bits

Bits 4 to 2 are reserved. They are always read as 0, and cannot be modified.

Bit 1: TAIL MARK transmit flag (MTRF)

When bit MRKON is set to 1, bit 1 indicates that TAIL MARK is being sent. Bit 1 is a read-only bit and cannot be modified.

Bit 1 MTRF	Description	
0	Idle state, or 8- or 16-bit data is being transferred	(initial value)
1	TAIL MARK is being sent	

Bit 0: Start flag (STF)

Bit 0 controls the start of a transfer. Setting this bit to 1 causes SCI1 to start transferring data.

During the transfer or while waiting for the first clock pulse, this bit remains set to 1. It is cleared to 0 upon completion of the transfer. It can therefore be used as a busy flag.

Bit 0 STF	Description	
0	Read: Indicates that transfer is stopped	(initial value)
	Write: Invalid	
1	Read: Indicates transfer in progress	
	Write: Starts a transfer operation	

3. Serial data register U (SDRU)

Bit	7	6	5	4	3	2	1	0
	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SDRU is an 8-bit read/write register. It is used as the data register for the upper 8 bits in 16-bit transfer (SDRL is used for the lower 8 bits).

Data written to SDRU is output to SDRL starting from the least significant bit (LSB). This data is then replaced by LSB-first data input at pin SI1, which is shifted in the direction from the most significant bit (MSB) toward the LSB.

SDRU must be written or read only after data transmission or reception is complete. If this register is written or read while a data transfer is in progress, the data contents are not guaranteed.

The SDRU value upon reset is not fixed.

4. Serial data register L (SDRL)

Bit	7	6	5	4	3	2	1	0
	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SDRL is an 8-bit read/write register. It is used as the data register in 8-bit transfer, and as the data register for the lower 8 bits in 16-bit transfer (SDRU is used for the upper 8 bits).

In 8-bit transfer, data written to SDRL is output from pin SO_1 starting from the least significant bit (LSB). This data is then replaced by LSB-first data input at pin SI_1, which is shifted in the direction from the most significant bit (MSB) toward the LSB.

In 16-bit transfer, operation is the same as for 8-bit transfer, except that input data is fed in via SDRU.

SDRL must be written or read only after data transmission or reception is complete. If this register is read or written while a data transfer is in progress, the data contents are not guaranteed.

The SDRL value upon reset is not fixed.

10.2.3 Operation in Synchronous Mode

Data can be sent and received in an 8-bit or 16-bit format, synchronized to an internal or external serial clock. Overrun errors can be detected when an external clock is used.

1. Clock

The serial clock can be selected from a choice of eight internal clocks and an external clock. When an internal clock source is selected, pin SCK_1 becomes the clock output pin. When continuous clock output mode is selected (SCR1 bits SNC1 and SNC0 are set to 10), the clock signal (ø/1024 to ø/2) selected in bits CKS2 to CKS0 is output continuously from pin SCK_1. When an external clock is used, pin SCK_1 is the clock input pin.

2. Data transfer format

Figure 10-2 shows the data transfer format. Data is sent and received starting from the least significant bit, in LSB-first format. Transmit data is output from one falling edge of the serial clock until the next rising edge. Receive data is latched at the rising edge of the serial clock.

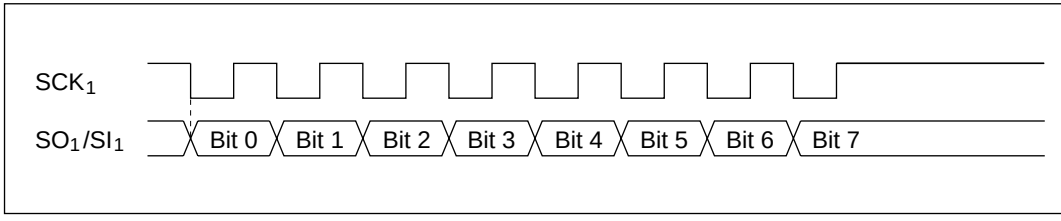


Figure 10-2 Transfer Format

3. Data transfer operations

- Transmitting

A transmit operation is carried out as follows.

1. Set bits SO1 and SCK1 to 1 in PMR3 to select the SO_1 and SCK_1 pin functions. If necessary, set bit POF1 in PMR7 for NMOS open-drain output at pin SO_1.

2. Clear bit SNC1 in SCR1 to 0, set bit SNC0 to 0 or 1, and clear bit MRKON to 0, designating 8- or 16-bit synchronous transfer mode. Select the serial clock in bits CKS3 to CKS0. Writing data to SCR1 when bit MRKON in SCR1 is cleared to 0 initializes the internal state of SCI1.

3. Write transmit data in SDRL and SDRU, as follows.

   8-bit transfer mode: SDRL
   16-bit transfer mode: Upper byte in SDRU, lower byte in SDRL

4. Set the SCSR1 start flag (STF) to 1. SCI1 starts operating and outputs transmit data at pin SO_1.

5. After data transmission is complete, bit IRRS1 in interrupt request register 2 (IRR2) is set to 1.

When an internal clock is used, a serial clock is output from pin SCK_1 in synchronization with the transmit data. After data transmission is complete, the serial clock is not output until the next time the start flag is set to 1. During this time, pin SO_1 continues to output the value of the last bit transmitted.

When an external clock is used, data is transmitted in synchronization with the serial clock input at pin SCK_1. After data transmission is complete, an overrun occurs if the serial clock continues to be input; no data is transmitted and the SCSR1 overrun error flag (bit ORER) is set to 1.

While transmission is stopped, the output value of pin SO_1 can be changed by rewriting bit SOL in SCSR1.

- Receiving

A receive operation is carried out as follows.

1. Set bits SI1 and SCK1 to 1 in PMR3 to select the SI_1 and SCK_1 pin functions.

2. Clear bit SNC1 in SCR1 to 0, set bit SNC0 to 0 or 1, and clear bit MRKON to 0, designating 8- or 16-bit synchronous transfer mode. Select the serial clock in bits CKS3 to CKS0. Writing data to SCR1 when bit MRKON in SCR1 is cleared to 0 initializes the internal state of SCI1.

3. Set the SCSR1 start flag (STF) to 1. SCI1 starts operating and receives data at pin SI_1.

4. After data reception is complete, bit IRRS1 in interrupt request register 2 (IRR2) is set to 1.

5. Read the received data from SDRL and SDRU, as follows.

   8-bit transfer mode: SDRL
   16-bit transfer mode: Upper byte in SDRU, lower byte in SDRL

6. After data reception is complete, an overrun occurs if the serial clock continues to be input; no data is received and the SCSR1 overrun error flag (bit ORER) is set to 1.

- Simultaneous transmit/receive

A simultaneous transmit/receive operation is carried out as follows.

1. Set bits SO1, SI1, and SCK1 to 1 in PMR3 to select the SO_1, SI_1, and SCK_1 pin functions. If necessary, set bit POF1 in PMR7 for NMOS open-drain output at pin SO_1.

2. Clear bit SNC1 in SCR1 to 0, set bit SNC0 to 0 or 1, and clear bit MRKON to 0, designating 8- or 16-bit synchronous transfer mode. Select the serial clock in bits CKS3 to CKS0. Writing data to SCR1 when bit MRKON in SCR1 is cleared to 0 initializes the internal state of SCI1.

3. Write transmit data in SDRL and SDRU, as follows.

   8-bit transfer mode: SDRL
   16-bit transfer mode: Upper byte in SDRU, lower byte in SDRL

4. Set the SCSR1 start flag (STF) to 1. SCI1 starts operating. Transmit data is output at pin SO_1. Receive data is input at pin SI_1.

5. After data transmission and reception are complete, bit IRRS1 in IRR2 is set to 1.

6. Read the received data from SDRL and SDRU, as follows.

   8-bit transfer mode: SDRL
   16-bit transfer mode: Upper byte in SDRU, lower byte in SDRL

When an internal clock is used, a serial clock is output from pin SCK_1 in synchronization with the transmit data. After data transmission is complete, the serial clock is not output until the next time the start flag is set to 1. During this time, pin SO_1 continues to output the value of the last bit transmitted.

When an external clock is used, data is transmitted and received in synchronization with the serial clock input at pin SCK_1. After data transmission and reception are complete, an overrun occurs if the serial clock continues to be input; no data is transmitted or received and the SCSR1 overrun error flag (bit ORER) is set to 1.

While transmission is stopped, the output value of pin SO_1 can be changed by rewriting bit SOL in SCSR1.

10.2.4 Operation in SSB Mode

SSB communication uses two lines, SCL (Serial Clock) and SDA (Serial Data), and enables multiple ICs to be connected as shown in figure 10-3.

In SSB mode, TAIL MARK is sent after an 8- or 16-bit data transfer. HOLD TAIL or LATCH TAIL can be selected as TAIL MARK.

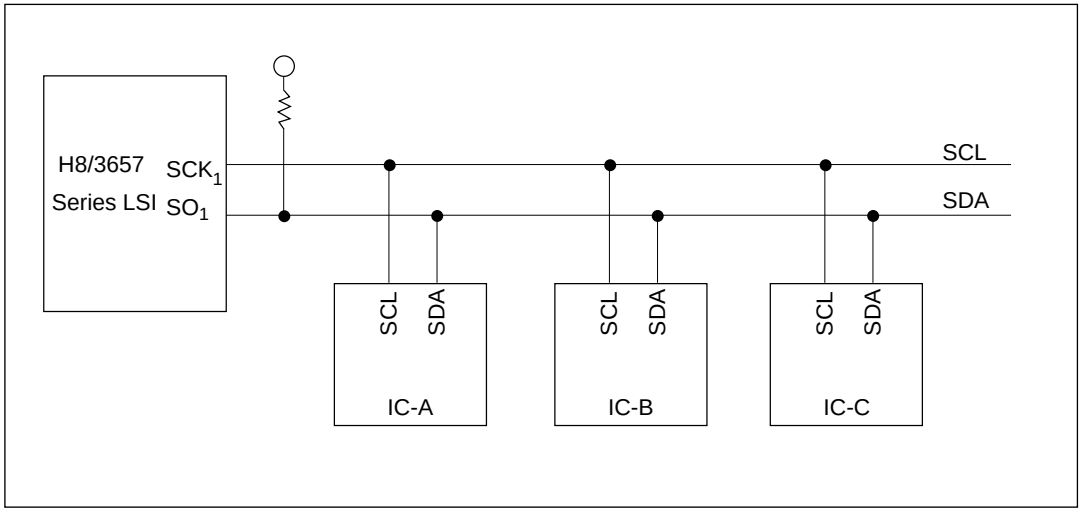


Figure 10-3 Example of SSB Connection

1. Clock

The transfer clock can be selected from eight internal clocks or an external clock, but since the H8/3657 Series uses clock output, an external clock should not be selected. The transfer rate can be selected by bits CKS2 to CKS0 in SCR1. Since this is also the TAIL MARK transfer rate, the setting should be made to give a transfer clock cycle of at least 2 µs.

2. Data transfer format

Figure 10-4 shows the SCI1 transfer format. Data is sent starting from the least significant bit, in LSB-first format. TAIL MARK is sent after an 8- or 16-bit data transfer.

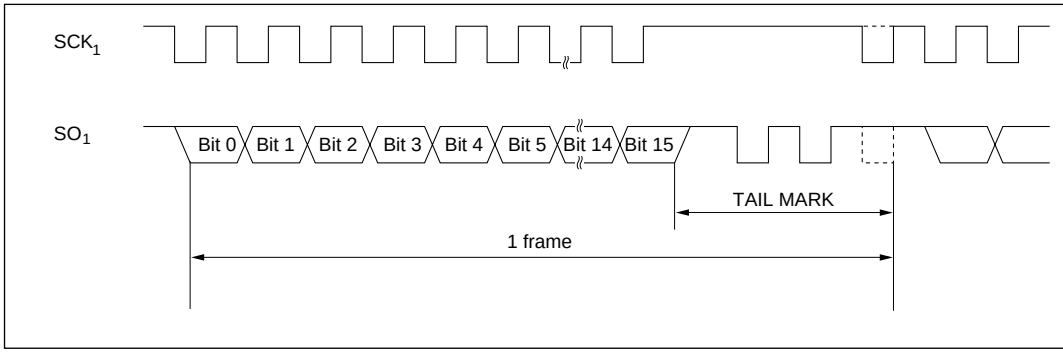


Figure 10-4 Transfer Format (When SNC1 = 0, SNC0 = 1, MRKON = 1)

3. TAIL MARK

TAIL MARK can be either HOLD TAIL or LATCH TAIL. The output waveforms of HOLD TAIL and LATCH TAIL are shown in figure 10-5. Time t in the figure is determined by the transfer clock cycle set in bits CKS2 to CKS0 in SCR1.

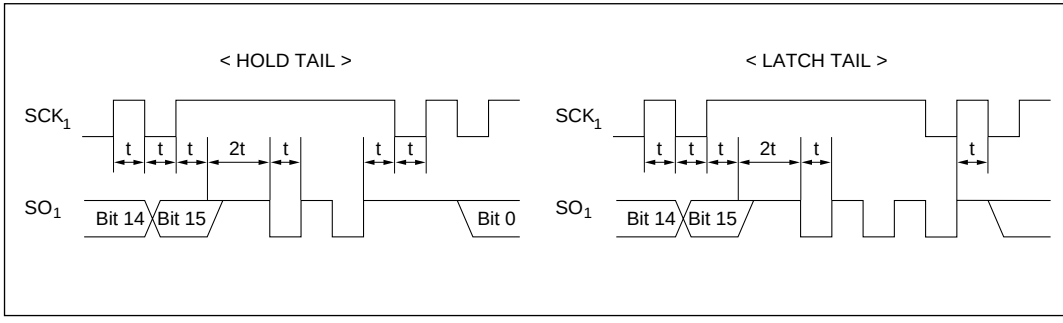


Figure 10-5 HOLD TAIL and LATCH TAIL Waveforms

4. Transmitting

A transmit operation is carried out as follows.

1. Set bit SOL in SCSR1 to 1.

2. Set bits SO1 and SCK1 to 1 in PMR3 to select the SO_1 and SCK_1 pin functions. Set bit POF1 in PMR7 to 1 for NMOS open-drain output at pin SO_1.

3. Clear bit SNC1 in SCR1 to 0 and set bit SNC0 to 0 or 1, designating 8-bit mode or 16-bit mode. Set bit MRKON in SCR1 to 1, selecting SSB mode.

4. Write transmit data in SDRL and SDRU as follows, and select TAIL MARK with bit LTCH in SCR1.

   8-bit mode: SDRL
   16-bit mode: Upper byte in SDRU, lower byte in SDRL

5. Set the SCSR1 start flag (STF) to 1. SCI1 starts operating and outputs transmit data at pin $S0_1$.

6. After 8- or 16-bit data transmission is complete, bit STF in SCSR1 is cleared to 0 and bit IRRS1 in interrupt request register 2 (IRRS1) is set to 1. The selected TAIL MARK is output after the data transmission. During TAIL MARK output, bit MTRF in SCSR1 is set to 1.

Data can be sent continuously by repeating steps 4 to 6. Check that SCI1 is in the idle state before rewriting bit MRKON in SCR1.

10.2.5 Interrupts

SCI1 can generate an interrupt at the end of a data transfer.

When an SCI1 transfer is complete, bit IRRS1 in interrupt request register 2 (IRR2) is set to 1. SCI1 interrupt requests can be enabled or disabled by bit IENS1 of interrupt enable register 2 (IENR2).

For further details, see 3.3, Interrupts.

10.3 SCI3

10.3.1 Overview

Serial communication interface 3 (SCI3) can carry out serial data communication in either asynchronous or synchronous mode. It is also provided with a multiprocessor communication function that enables serial data to be transferred among processors.

1. Features

Features of SCI3 are listed below.

- Choice of asynchronous or synchronous mode for serial data communication
  - Asynchronous mode

    Serial data communication is performed asynchronously, with synchronization provided character by character. In this mode, serial data can be exchanged with standard asynchronous communication LSIs such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA). A multiprocessor communication function is also provided, enabling serial data communication among processors.

    There is a choice of 12 data transfer formats.

Data length	7 or 8 bits
Stop bit length	1 or 2 bits
Parity	Even, odd, or none
Multiprocessor bit	“1” or “0”
Receive error detection	Parity, overrun, and framing errors
Break detection	Break detected by reading the RXD pin level directly when a framing error occurs

  - Synchronous mode

    Serial data communication is synchronized with a clock. In his mode, serial data can be exchanged with another LSI that has a synchronous communication function.

Data length	8 bits
Receive error detection	Overrun errors

- Full-duplex communication

  Separate transmission and reception units are provided, enabling transmission and reception to be carried out simultaneously. The transmission and reception units are both double-buffered, allowing continuous transmission and reception.

- On-chip baud rate generator, allowing any desired bit rate to be selected

- Choice of an internal or external clock as the transmit/receive clock source

- Six interrupt sources: transmit end, transmit data empty, receive data full, overrun error, framing error, and parity error

2. Block diagram

Figure 10-6 shows a block diagram of SCI3.

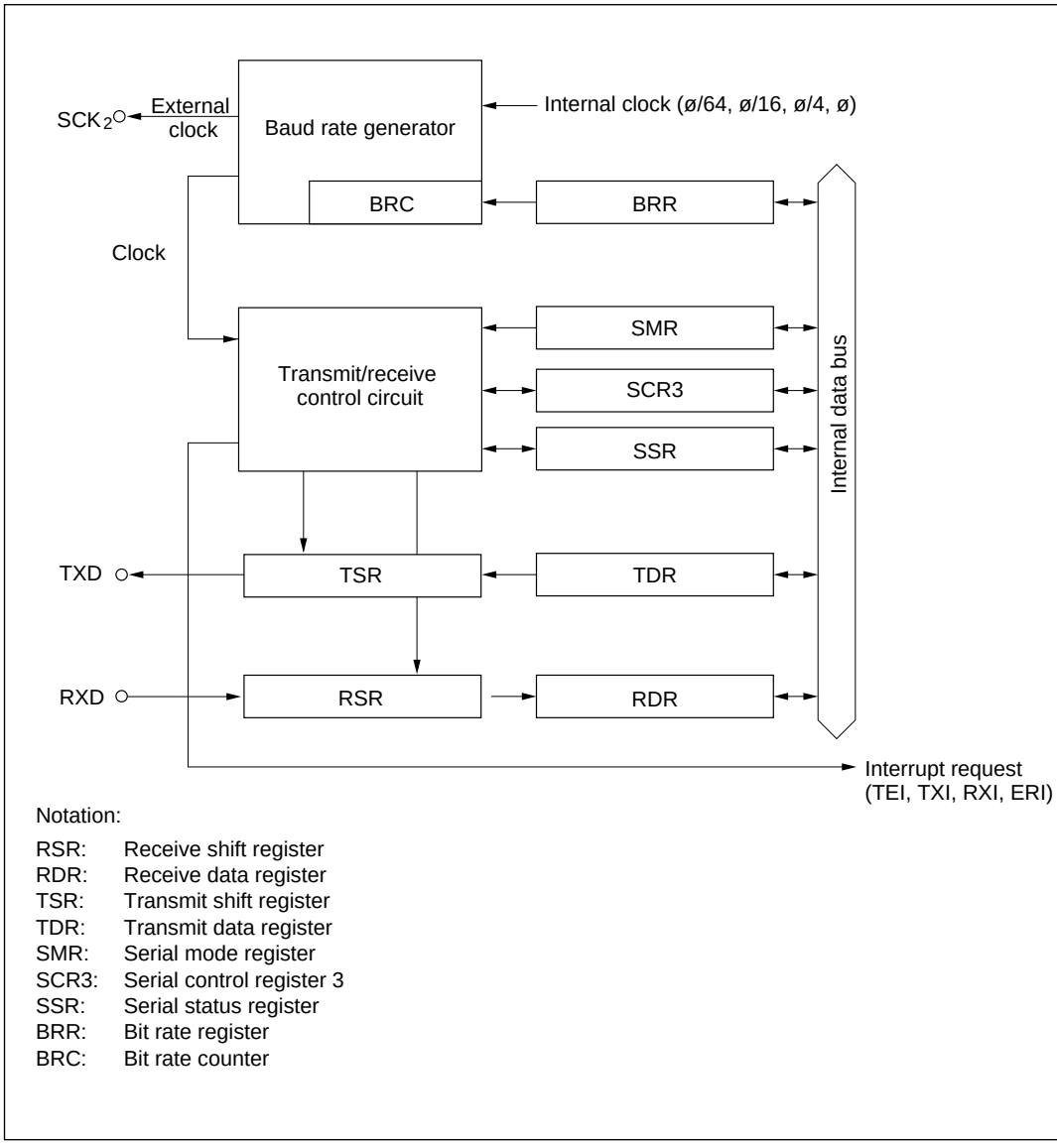


Figure 10-6 SCI3 Block Diagram

3. Pin configuration

Table 10-4 shows the SCI3 pin configuration.

Table 10-4 Pin Configuration

Name	Abbrev.	I/O	Function
SCI3 clock	SCK_3	I/O	SCI3 clock input/output
SCI3 receive data input	RXD	Input	SCI3 receive data input
SCI3 transmit data output	TXD	Output	SCI3 transmit data output

4. Register configuration

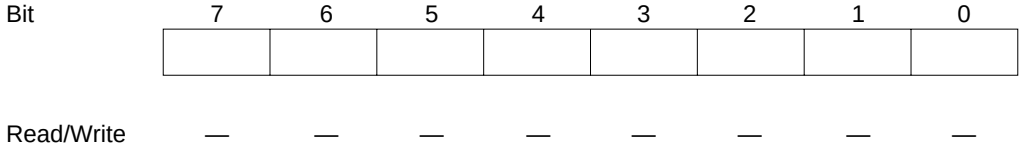
Table 10-5 shows the SCI3 register configuration.

Table 10-5 Registers

Name	Abbrev.	R/W	Initial Value	Address
Serial mode register	SMR	R/W	H'00	H'FFA8
Bit rate register	BRR	R/W	H'FF	H'FFA9
Serial control register 3	SCR3	R/W	H'00	H'FFAA
Transmit data register	TDR	R/W	H'FF	H'FFAB
Serial data register	SSR	R/W	H'84	H'FFAC
Receive data register	RDR	R	H'00	H'FFAD
Transmit shift register	TSR	Protected	—	—
Receive shift register	RSR	Protected	—	—
Bit rate counter	BRC	Protected	—	—

10.3.2 Register Descriptions

1. Receive shift register (RSR)



RSR is a register used to receive serial data. Serial data input to RSR from the RXD pin is set in the order in which it is received, starting from the LSB (bit 0), and converted to parallel data. When one byte of data is received, it is transferred to RDR automatically.

RSR cannot be read or written directly by the CPU.

2. Receive data register (RDR)

Bit	7	6	5	4	3	2	1	0
	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

RDR is an 8-bit register that stores received serial data.

When reception of one byte of data is finished, the received data is transferred from RSR to RDR, and the receive operation is completed. RSR is then enabled for reception. RSR and RDR are double-buffered, allowing consecutive receive operations.

RDR is a read-only register, and cannot be written by the CPU.

RDR is initialized to H'00 upon reset, and in standby, watch, subactive, or subsleep mode.

3. Transmit shift register (TSR)

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

TSR is a register used to transmit serial data. Transmit data is first transferred from TDR to TSR, and serial data transmission is carried out by sending the data to the TXD pin in order, starting from the LSB (bit 0). When one byte of data is transmitted, the next byte of transmit data is transferred to TDR, and transmission started, automatically. Data transfer from TDR to TSR is not performed if no data has been written to TDR (if bit TDRE is set to 1 in the serial status register (SSR)).

TSR cannot be read or written directly by the CPU.

4. Transmit data register (TDR)

Bit	7	6	5	4	3	2	1	0
	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TDR is an 8-bit register that stores transmit data. When TSR is found to be empty, the transmit data written in TDR is transferred to TSR, and serial data transmission is started. Continuous transmission is possible by writing the next transmit data to TDR during TSR serial data transmission.

TDR can be read or written by the CPU at any time.

TDR is initialized to H'FF upon reset, and in standby, watch, subactive, or subsleep mode.

5. Serial mode register (SMR)

Bit	7	6	5	4	3	2	1	0
	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SMR is an 8-bit register used to set the serial data transfer format and to select the clock source for the baud rate generator.

SMR can be read or written by the CPU at any time.

SMR is initialized to H'00 upon reset, and in standby, watch, subactive, or subsleep mode.

Bit 7: Communication mode (COM)

Bit 7 selects whether SCI3 operates in asynchronous mode or synchronous mode.

Bit 7 COM	**Description**	
0	Asynchronous mode	(initial value)
1	Synchronous mode	

Bit 6: Character length (CHR)

Bit 6 selects either 7 or 8 bits as the data length to be used in asynchronous mode. In synchronous mode the data length is always 8 bits, irrespective of the bit 6 setting.

Bit 6 CHR	Description	
0	8-bit data	(initial value)
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted.

Bit 5: Parity enable (PE)

Bit 5 selects whether a parity bit is to be added during transmission and checked during reception in asynchronous mode. In synchronous mode parity bit addition and checking is not performed, irrespective of the bit 5 setting.

Bit 5 PE	Description	
0	Parity bit addition and checking disabled	(initial value)
1	Parity bit addition and checking enabled*	

Note: * When PE is set to 1, even or odd parity, as designated by bit PM, is added to transmit data before it is sent, and the received parity bit is checked against the parity designated by bit PM.

Bit 4: Parity mode (PM)

Bit 4 selects whether even or odd parity is to be used for parity addition and checking. The PM bit setting is only valid in asynchronous mode when bit PE is set to 1, enabling parity bit addition and checking. The PM bit setting is invalid in synchronous mode, and in asynchronous mode if parity bit addition and checking is disabled.

Bit 4 PM	Description	
0	Even parity*1	(initial value)
1	Odd parity*2	

Notes:
1. When even parity is selected, a parity bit is added in transmission so that the total number of 1 bits in the transmit data plus the parity bit is an even number; in reception, a check is carried out to confirm that the number of 1 bits in the receive data plus the parity bit is an even number.
2. When odd parity is selected, a parity bit is added in transmission so that the total number of 1 bits in the transmit data plus the parity bit is an odd number; in reception, a check is carried out to confirm that the number of 1 bits in the receive data plus the parity bit is an odd number.

Bit 3: Stop bit length (STOP)

Bit 3 selects 1 bit or 2 bits as the stop bit length is asynchronous mode. The STOP bit setting is only valid in asynchronous mode. When synchronous mode is selected the STOP bit setting is invalid since stop bits are not added.

Bit 3 STOP	Description	
0	1 stop bit*1	(initial value)
1	2 stop bits*2	

Notes: 1. In transmission, a single 1 bit (stop bit) is added at the end of a transmit character.
2. In transmission, two 1 bits (stop bits) are added at the end of a transmit character.

In reception, only the first of the received stop bits is checked, irrespective of the STOP bit setting. If the second stop bit is 1 it is treated as a stop bit, but if 0, it is treated as the start bit of the next transmit character.

Bit 2: Multiprocessor mode (MP)

Bit 2 enables or disables the multiprocessor communication function. When the multiprocessor communication function is disabled, the parity settings in the PE and PM bits are invalid. The MP bit setting is only valid in asynchronous mode. When synchronous mode is selected the MP bit should be set to 0. For details on the multiprocessor communication function, see 10.3.6, Multiprocessor Communication Function.

Bit 2 MP	Description	
0	Multiprocessor communication function disabled	(initial value)
1	Multiprocessor communication function enabled	

Bits 1 and 0: Clock select (CKS1, CKS0)

Bits 1 and 0 choose ø/64, ø/16, ø/4, or ø as the clock source for the baud rate generator.

For the relation between the clock source, bit rate register setting, and baud rate, see 8, Bit rate register (BRR).

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	ø clock	(initial value)
0	1	ø/4 clock	
1	0	ø/16 clock	
1	1	ø/64 clock	

6. Serial control register 3 (SCR3)

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

SCR3 is an 8-bit register for selecting transmit or receive operation, the asynchronous mode clock output, interrupt request enabling or disabling, and the transmit/receive clock source.

SCR3 can be read or written by the CPU at any time.

SCR3 is initialized to H'00 upon reset, and in standby, watch, subactive, or subsleep mode.

Bit 7: Transmit interrupt enable (TIE)

Bit 7 selects enabling or disabling of the transmit data empty interrupt request (TXI) when transmit data is transferred from the transmit data register (TDR) to the transmit shift register (TSR), and bit TDRE in the serial status register (SSR) is set to 1.

TXI can be released by clearing bit TDRE or bit TIE to 0.

Bit 7 TIE	Description	
0	Transmit data empty interrupt request (TXI) disabled	(initial value)
1	Transmit data empty interrupt request (TXI) enabled	

Bit 6: Receive interrupt enable (RIE)

Bit 6 selects enabling or disabling of the receive data full interrupt request (RXI) and the receive error interrupt request (ERI) when receive data is transferred from the receive shift register (RSR) to the receive data register (RDR), and bit RDRF in the serial status register (SSR) is set to 1. There are three kinds of receive error: overrun, framing, and parity.

RXI and ERI can be released by clearing bit RDRF or the FER, PER, or OER error flag to 0, or by clearing bit RIE to 0.

Bit 6 RIE	Description	
0	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) disabled	(initial value)
1	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) enabled	

Bit 5: Transmit enable (TE)

Bit 5 selects enabling or disabling of the start of transmit operation.

Bit 5 TE	Description	
0	Transmit operation disabled*1 (TXD pin is transmit data pin)	(initial value)
1	Transmit operation enabled*2 (TXD pin is transmit data pin)	

Notes:
1. Bit TDRE in SSR is fixed at 1. Transmission operation is disabled, but the TXD pin functions as the transmit data pin. To use the TXD pin as an I/O pin, clear bit TXD in PMR7 to 0.
2. When transmit data is written to TDR in this state, bit TDR in SSR is cleared to 0 and serial data transmission is started. Be sure to carry out serial mode register (SMR) settings to decide the transmission format before setting bit TE to 1.

Bit 4: Receive enable (RE)

Bit 4 selects enabling or disabling of the start of receive operation.

Bit 4 RE	Description	
0	Receive operation disabled*1 (RXD pin is I/O port)	(initial value)
1	Receive operation enabled*2 (RXD pin is receive data pin)	

Notes:
1. Note that the RDRF, FER, PER, and OER flags in SSR are not affected when bit RE is cleared to 0, and retain their previous state.
2. In this state, serial data reception is started when a start bit is detected in asynchronous mode or serial clock input is detected in synchronous mode. Be sure to carry out serial mode register (SMR) settings to decide the reception format before setting bit RE to 1.

Bit 3: Multiprocessor interrupt enable (MPIE)

Bit 3 selects enabling or disabling of the multiprocessor interrupt request. The MPIE bit setting is only valid when asynchronous mode is selected and reception is carried out with bit MP in SMR set to 1. The MPIE bit setting is invalid when bit COM is set to 1 or bit MP is cleared to 0.

Bit 3 MPIE	Description	
0	Multiprocessor interrupt request disabled (normal receive operation) Clearing conditions: When data is received in which the multiprocessor bit is set to 1	(initial value)
1	Multiprocessor interrupt request enabled*	

Note: * Receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and OER status flags in SSR is not performed. RXI, ERI, and setting of the RDRF, FER, and OER flags in SSR, are disabled until data with the multiprocessor bit set to 1 is received. When a receive character with the multiprocessor bit set to 1 is received, bit MPBR in SSR is set to 1, bit MPIE is automatically cleared to 0, and RXI and ERI requests (when bits TIE and RIE in serial control register 3 (SCR3) are set to 1) and setting of the RDRF, FER, and OER flags are enabled.

Bit 2: Transmit end interrupt enable (TEIE)

Bit 2 selects enabling or disabling of the transmit end interrupt request (TEI) if there is no valid transmit data in TDR when MSB data is to be sent.

Bit 2 TEIE	Description	
0	Transmit end interrupt request (TEI) disabled	(initial value)
1	Transmit end interrupt request (TEI) enabled*	

Note: * TEI can be released by clearing bit TDRE to 0 and clearing bit TEND to 0 in SSR, or by clearing bit TEIE to 0.

Bits 1 and 0: Clock enable 1 and 0 (CKE1, CKE0)

Bits 1 and 0 select the clock source and enabling or disabling of clock output from the SCK_3 pin. These bits determine whether the SCK_3 pin functions as an I/O port, a clock output pin, or a clock input pin.

The CKE0 bit setting is only valid in case of internal clock operation (CKE1 = 0) in asynchronous mode. In synchronous mode, or when external clock operation is used (CKE1 = 1), bit CKE0 should be cleared to 0.

After setting bits CKE1 and CKE0, set the operating mode in the serial mode register (SMR).

For details on clock source selection, see table 10-10 in 10.3.3, Operation.

Bit 1	Bit 0	Description		
CKE1	CKE0	Communication Mode	Clock Source	SCK_3 Pin Function
0	0	Asynchronous	Internal clock	I/O port*1
		Synchronous	Internal clock	Serial clock output*1
0	1	Asynchronous	Internal clock	Clock output*2
		Synchronous	Reserved	
1	0	Asynchronous	External clock	Clock input*3
		Synchronous	External clock	Serial clock input
1	1	Asynchronous	Reserved	
		Synchronous	Reserved	

Notes: 1. Initial value
2. A clock with the same frequency as the bit rate is output.
3. Input a clock with a frequency 16 times the bit rate.

7. Serial status register (SSR)

Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT
Initial value	1	0	0	0	0	1	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * Only a write of 0 for flag clearing is possible.

SSR is an 8-bit register containing status flags that indicate the operational status of SCI3, and multiprocessor bits.

SSR can be read or written by the CPU at any time, but 1 cannot be written to bits TDRE, RDRF, OER, PER, and FER. In order to clear these bits by writing 0, 1 must first be read.

Bits TEND and MPBR are read-only bits, and cannot be modified.

SSR is initialized to H'84 upon reset, and in standby, watch, subactive, or subsleep mode.

Bit 7: Transmit data register empty (TDRE)

Bit 7 indicates that transmit data has been transferred from TDR to TSR.

Bit 7 TDRE	**Description**
0	Transmit data written in TDR has not been transferred to TSR Clearing conditions: • After reading TDRE = 1, cleared by writing 0 to TDRE • When data is written to TDR by an instruction
1	Transmit data has not been written to TDR, or transmit data written in TDR has been transferred to TSR Setting conditions: • When bit TE in SCR3 is cleared to 0 • When data is transferred from TDR to TSR (initial value)

Bit 6: Receive data register full (RDRF)

Bit 6 indicates that received data is stored in RDR.

Bit 6 RDRF	Description	
0	There is no receive data in RDR Clearing conditions: • After reading RDRF = 1, cleared by writing 0 to RDRF • When RDR data is read by an instruction	(initial value)
1	There is receive data in PDR Setting conditions: When reception ends normally and receive data is transferred from RSR to RDR	

Note: If an error is detected in the receive data, or if the RE bit in SCR3 has been cleared to 0, RDR and bit RDRF are not affected and retain their previous state.
Note that if data reception is completed while bit RDRF is still set to 1, an overrun error (OER) will result and the receive data will be lost.

Bit 5: Overrun error (OER)

Bit 5 indicates that an overrun error has occurred during reception.

Bit 5 OER	Description	
0	Reception in progress or completed*1 Clearing conditions: After reading OER = 1, cleared by writing 0 to OER	(initial value)
1	An overrun error has occurred during reception*2 Setting conditions: When reception is completed with RDRF set to 1	

Notes: 1. When bit RE in SCR3 is cleared to 0, bit OER is not affected and retains its previous state.
2. RDR retains the receive data it held before the overrun error occurred, and data received after the error is lost. Reception cannot be continued with bit OER set to 1, and in synchronous mode, transmission cannot be continued either.

Bit 4: Framing error (FER)

Bit 4 indicates that a framing error has occurred during reception in asynchronous mode.

Bit 4 FER	Description	
0	Reception in progress or completed[*1] Clearing conditions: After reading FER = 1, cleared by writing 0 to FER	(initial value)
1	A framing error has occurred during reception[*2] Setting conditions: When the stop bit at the end of the receive data is checked for a value of 1 at the end of reception, and the stop bit is 0[*2]	

Notes:
1. When bit RE in SCR3 is cleared to 0, bit FER is not affected and retains its previous state.
2. Note that, in 2-stop-bit mode, only the first stop bit is checked for a value of 1, and the second stop bit is not checked. When a framing error occurs the receive data is transferred to RDR but bit RDRF is not set. Reception cannot be continued with bit FER set to 1. In synchronous mode, neither transmission nor reception is possible when bit FER is set to 1.

Bit 3: Parity error (PER)

Bit 3 indicates that a parity error has occurred during reception with parity added in asynchronous mode.

Bit 3 PER	Description	
0	Reception in progress or completed[*1] Clearing conditions: After reading PER = 1, cleared by writing 0 to PER	(initial value)
1	A parity error has occurred during reception[*2] Setting conditions: When the number of 1 bits in the receive data plus parity bit does not match the parity designated by bit PM in the serial mode register (SMR)	

Notes:
1. When bit RE in SCR3 is cleared to 0, bit PER is not affected and retains its previous state.
2. Receive data in which it a parity error has occurred is still transferred to RDR, but bit RDRF is not set. Reception cannot be continued with bit PER set to 1. In synchronous mode, neither transmission nor reception is possible when bit FER is set to 1.

Bit 2: Transmit end (TEND)

Bit 2 indicates that bit TDRE is set to 1 when the last bit of a transmit character is sent.

Bit 2 is a read-only bit and cannot be modified.

Bit 2 TEND	Description	
0	Transmission in progress Clearing conditions: • After reading TDRE = 1, cleared by writing 0 to TDRE • When data is written to TDR by an instruction	
1	Transmission ended Setting conditions: • When bit TE in SCR3 is cleared to 0 • When bit TDRE is set to 1 when the last bit of a transmit character is sent	(initial value)

Bit 1: Multiprocessor bit receive (MPBR)

Bit 1 stores the multiprocessor bit in a receive character during multiprocessor format reception in asynchronous mode.

Bit 1 is a read-only bit and cannot be modified.

Bit 1 MPBR	Description	
0	Data in which the multiprocessor bit is 0 has been received*	(initial value)
1	Data in which the multiprocessor bit is 1 has been received	

Note: * When bit RE is cleared to 0 in SCR3 with the multiprocessor format, bit MPBR is not affected and retains its previous state.

Bit 0: Multiprocessor bit transfer (MPBT)

Bit 0 stores the multiprocessor bit added to transmit data when transmitting in asynchronous mode. The bit MPBT setting is invalid when synchronous mode is selected, when the multiprocessor communication function is disabled, and when not transmitting.

Bit 0 MPBT	Description	
0	A 0 multiprocessor bit is transmitted	(initial value)
1	A 1 multiprocessor bit is transmitted	

8. Bit rate register (BRR)

Bit	7	6	5	4	3	2	1	0
	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
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 designates the transmit/receive bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 of the serial mode register (SMR).

BRR can be read or written by the CPU at any time.

BRR is initialized to H'FF upon reset, and in standby, watch, subactive, or subsleep mode.

Table 10-6 shows examples of BRR settings in asynchronous mode. The values shown are for active (high-speed) mode.

Table 10-6 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)

	OSC (MHz)											
	2			2.4576			4			4.194304		
R Bit Rate (bit/s)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	1	70	+0.03	1	86	+0.31	1	141	+0.03	1	148	–0.04
150	0	207	+0.16	0	255	0	1	103	+0.16	1	108	+0.21
300	0	103	+0.16	0	127	0	0	207	+0.16	0	217	+0.21
600	0	51	+0.16	0	63	0	0	103	+0.16	0	108	+0.21
1200	0	25	+0.16	0	31	0	0	51	+0.16	0	54	–0.70
2400	0	12	+0.16	0	15	0	0	25	+0.16	0	26	+1.14
4800	—	—	—	0	7	0	0	12	+0.16	0	13	–2.48
9600	—	—	—	0	3	0	—	—	—	0	6	–2.48
19200	—	—	—	0	1	0	—	—	—	—	—	—
31250	0	0	0	—	—	—	0	1	0	—	—	—
38400	—	—	—	0	0	0	—	—	—	—	—	—

Table 10-6 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)

R Bit Rate (bit/s)	OSC (MHz) 4.9152 n	N	Error (%)	6 n	N	Error (%)	7.3728 n	N	Error (%)	8 n	N	Error (%)
110	1	174	–0.26	1	212	+0.03	2	64	+0.70	2	70	+0.03
150	1	127	0	1	155	+0.16	1	191	0	1	207	+0.16
300	0	255	0	1	77	+0.16	1	95	0	1	103	+0.16
600	0	127	0	0	155	+0.16	0	191	0	0	207	+0.16
1200	0	63	0	0	77	+0.16	0	95	0	0	103	+0.16
2400	0	31	0	0	38	+0.16	0	47	0	0	51	+0.16
4800	0	15	0	0	19	–2.34	0	23	0	0	25	+0.16
9600	0	7	0	0	9	–2.34	0	11	0	0	12	+0.16
19200	0	3	0	0	4	–2.34	0	5	0	—	—	—
31250	—	—	—	0	2	0	—	—	—	0	3	0
38400	0	1	0	—	—	—	0	2	0	—	—	—

Table 10-6 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3)

R Bit Rate (bit/s)	OSC (MHz) 9.8304 n	N	Error (%)	10 n	N	Error (%)
110	2	86	+0.31	2	88	–0.25
150	1	255	0	2	64	+0.16
300	1	127	0	1	129	+0.16
600	0	255	0	1	64	+0.16
1200	0	127	0	0	129	+0.16
2400	0	63	0	0	64	+0.16
4800	0	31	0	0	32	–1.36
9600	0	15	0	0	15	+1.73
19200	0	7	0	0	7	+1.73
31250	0	4	–1.70	0	4	0
38400	0	3	0	0	3	+1.73

Notes: 1. The setting should be made so that the error is not more than 1%.

2. The value set in BRR is given by the following equation:

$$N = \frac{OSC}{(64 \times 2^{2n} \times B)} \times 10^6 - 1$$

where

B: Bit rate (bit/s)

N: Baud rate generator BRR setting ($0 \leq N \leq 255$)

OSC: Value of $ø_{OSC}$ (MHz)

n: Baud rate generator input clock number (n = 0, 1, 2, or 3)

(The relation between n and the clock is shown in table 10-7.)

Table 10-7 Relation between n and Clock

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ø	0	0
1	ø/4	0	1
2	ø16	1	0
3	ø/64	1	1

3. The error in table 10-6 is the value obtained from the following equation, rounded to two decimal places.

$$\text{Error (\%)} = \frac{\text{B (rate obtained from n, N, OSC)} - \text{R(bit rate in left-hand column in table 10-6.)}}{\text{R (bit rate in left-hand column in table 10-6.)}} \times 100$$

Table 10-8 shows the maximum bit rate for each frequency. The values shown are for active (high-speed) mode.

Table 10-8 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

OSC (MHz)	Maximum Bit Rate (bit/s)	Setting	
		n	N
2	31250	0	0
2.4576	38400	0	0
4	62500	0	0
4.194304	65536	0	0
4.9152	76800	0	0
6	93750	0	0
7.3728	115200	0	0
8	125000	0	0
9.8304	153600	0	0
10	156250	0	0

Table 10-9 shows examples of BRR settings in synchronous mode. The values shown are for active (high-speed) mode.

Table 10-9 Examples of BRR Settings for Various Bit Rates (Synchronous Mode)

B Bit Rate (bit/s)	OSC (MHz)							
	2		4		8		10	
	n	N	n	N	n	N	n	N
110	—	—	—	—	—	—	—	—
250	1	249	2	124	2	249	—	—
500	1	124	1	249	2	124	—	—
1k	0	249	1	124	1	249	—	—
2.5k	0	99	0	199	1	99	1	124
5k	0	49	0	99	0	199	0	249
10k	0	24	0	49	0	99	0	124
25k	0	9	0	19	0	39	0	49
50k	0	4	0	9	0	19	0	24
100k	—	—	0	4	0	9	—	—
250k	0	0*	0	1	0	3	0	4
500k			0	0*	0	1	—	—
1M					0	0*	—	—
2.5M								

Blank: Cannot be set.
— : A setting can be made, but an error will result.
* : Continuous transmission/reception is not possible.

Notes: The value set in BRR is given by the following equation:

$$N = \frac{OSC}{(8 \times 2^{2n} \times B)} \times 10^6 - 1$$

where

B: Bit rate (bit/s)

N: Baud rate generator BRR setting ($0 \leq N \leq 255$)

OSC: Value of $ø_{OSC}$ (MHz)

n: Baud rate generator input clock number (n = 0, 1, 2, or 3)

(The relation between n and the clock is shown in table 10-10.)

Table 10-10 Relation between n and Clock

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ø	0	0
1	ø/4	0	1
2	ø16	1	0
3	ø/64	1	1

10.3.3 Operation

SCI3 can perform serial communication in two modes: asynchronous mode in which synchronization is provided character by character, and synchronous mode in which synchronization is provided by clock pulses. The serial mode register (SMR) is used to select asynchronous or synchronous mode and the data transfer format, as shown in table 10-11.

The clock source for SCI3 is determined by bit COM in SMR and bits CKE1 and CKE0 in SCR3, as shown in table 10-12.

1. Synchronous mode

- Choice of 7- or 8-bit data length
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bits. (The combination of these parameters determines the data transfer format and the character length.)
- Framing error (FER), parity error (PER), overrun error (OER), and break detection during reception
- Choice of internal or external clock as the clock source

  When internal clock is selected: SCI3 operates on the baud rate generator clock, and a clock with the same frequency as the bit rate can be output.

  When external clock is selected: A clock with a frequency 16 times the bit rate must be input. (The on-chip baud rate generator is not used.)

2. Synchronous mode

- Data transfer format: Fixed 8-bit data length
- Overrun error detection during reception
- Choice of internal or external clock as the clock source

  When internal clock is selected: SCI3 operates on the baud rate generator clock, and a serial clock is output.

  When external clock is selected: The on-chip baud rate generator is not used, and SCI3 operates on the input serial clock.

Table 10-11 SMR Settings and Corresponding Data Transfer Formats

SMR						Data Transfer Format			
bit 7 COM	bit 6 CHR	bit 2 MP	bit 5 PE	bit 3 STOP	Mode	Data Length	Multiprocessor Bit	Parity Bit	Stop Bit Length
0	0	0	0	0	Asynchronous mode	8-bit data	No	No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	1		0	0		7-bit data		No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	0	1	*	0	Asynchronous mode (multiprocessor format)	8-bit data	Yes	No	1 bit
			*	1					2 bits
	1		*	0		7-bit data			1 bit
			*	1					2 bits
1	*	0	*	*	Synchronous mode	8-bit data	No	No	No

Table 10-12 SMR and SCR3 Settings and Clock Source Selection

SMR	SCR3			Transmit/Receive Clock	
bit 7	bit 1	bit 0			
COM	CKE1	CKE0	Mode	Clock Source	SCK_3 Pin Function
0	0	0	Asynchronous mode	Internal	I/O port (SCK_3 pin not used)
		1			Outputs clock with same frequency as bit rate
	1	0		External	Outputs clock with frequency 16 times bit rate
1	0	0	Synchronous mode	Internal	Outputs serial clock
	1	0		External	Inputs serial clock
0	1	1	Reserved (Do not specify these combinations)		
1	0	1			
1	1	1			

3. Interrupts and continuous transmission/reception

SCI3 can carry out continuous reception using RXI and continuous transmission using TXI. These interrupts are shown in table 10-13.

Table 10-13 Transmit/Receive Interrupts

Interrupt	Flags	Interrupt Request Conditions	Notes
RXI	RDRF RIE	When serial reception is performed normally and receive data is transferred from RSR to RDR, bit RDRF is set to 1, and if bit RIE is set to 1 at this time, RXI is enabled and an interrupt is requested. (See figure 10-7 (a).)	The RXI interrupt routine reads the receive data transferred to RDR and clears bit RDRF to 0. Continuous reception can be performed by repeating the above operations until reception of the next RSR data is completed.
TXI	TDRE TIE	When TSR is found to be empty (on completion of the previous transmission) and the transmit data placed in TDR is transferred to TSR, bit TDRE is set to 1. If bit TIE is set to 1 at this time, TXI is enabled and an interrupt is requested. (See figure 10-7 (b).)	The TXI interrupt routine writes the next transmit data to TDR and clears bit TDRE to 0. Continuous transmission can be performed by repeating the above operations until the data transferred to TSR has been transmitted.
TEI	TEND TEIE	When the last bit of the character in TSR is transmitted, if bit TDRE is set to 1, bit TEND is set to 1. If bit TEIE is set to 1 at this time, TEI is enabled and an interrupt is requested. (See figure 10-7 (c).)	TEI indicates that the next transmit data has not been written to TDR when the last bit of the transmit character in TSR is sent.

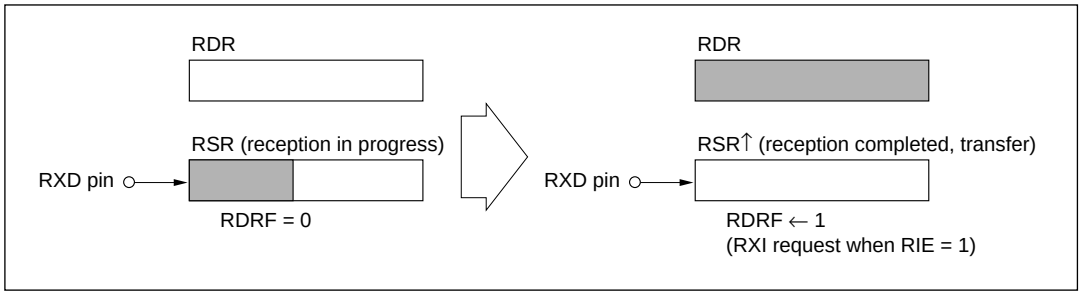


Figure 10-7 (a) RDRF Setting and RXI Interrupt

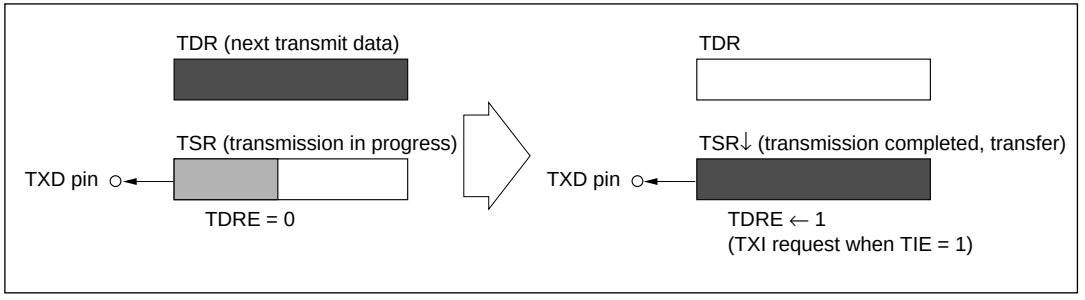


Figure 10-7 (b) TDRE Setting and TXI Interrupt

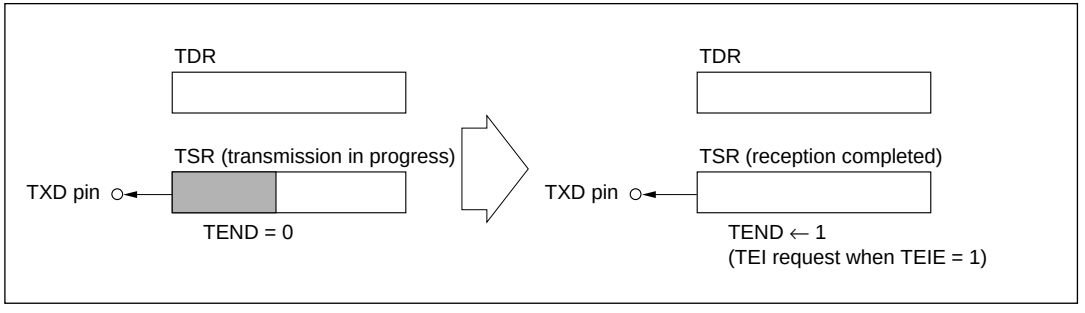


Figure 10-7 (c) TEND Setting and TEI Interrupt

10.3.4 Operation in Asynchronous Mode

In asynchronous mode, serial communication is performed with synchronization provided character by character. A start bit indicating the start of communication and one or two stop bits indicating the end of communication are added to each character before it is sent.

SCI3 has separate transmission and reception units, allowing full-duplex communication. As the transmission and reception units are both double-buffered, data can be written during transmission and read during reception, making possible continuous transmission and reception.

1. Data transfer format

The general data transfer format in asynchronous communication is shown in figure 10-8.

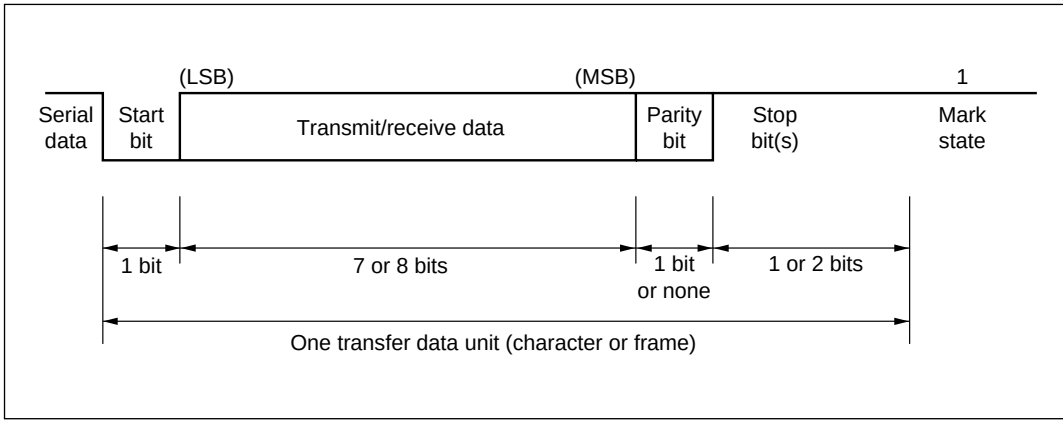


Figure 10-8 Data Format in Asynchronous Communication

In asynchronous communication, the communication line is normally in the mark state (high level). SCI3 monitors the communication line and when it detects a space (low level), identifies this as a start bit and begins serial data communication.

One transfer data character consists of a start bit (low level), followed by transmit/receive data (LSB-first format, starting from the least significant bit), a parity bit (high or low level), and finally one or two stop bits (high level).

In asynchronous mode, synchronization is performed by the falling edge of the start bit during reception. The data is sampled on the 8th pulse of a clock with a frequency 16 times the bit period, so that the transfer data is latched at the center of each bit.

Table 10-14 shows the 12 data transfer formats that can be set in asynchronous mode. The format is selected by the settings in the serial mode register (SMR).

Table 10-14 Data Transfer Formats (Asynchronous Mode)

SMR				Serial Data Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	*	1	0	S 8-bit data MPB STOP
0	*	1	1	S 8-bit data MPB STOP STOP
1	*	1	0	S 7-bit data MPB STOP
1	*	1	1	S 7-bit data MPB STOP STOP

* Dont' care

Notation:
S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multiprocessor bit

2. Clock

Either an internal clock generated by the baud rate generator or an external clock input at the SCK_3 pin can be selected as the SCI3 transmit/receive clock. The selection is made by means of bit COM in SMR and bits SCE1 and CKE0 in SCR3. See table 10-12 for details on clock source selection.

When an external clock is input to the SCK_3 pin, the input clock frequency should be 16 times the bit rate used.

When SCI3 operates on an internal clock, the clock can be output at the SCK_3 pin. In this case the frequency of the output clock is the same as the bit rate, and the phase is such that the clock rises at the center of each bit of transmit/receive data, as shown in figure 10-9.

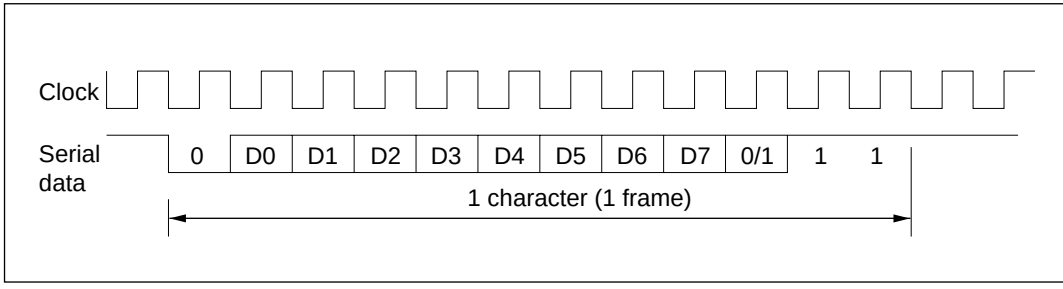


Figure 10-9 Phase Relationship between Output Clock and Transfer Data (Asynchronous Mode) (8-bit data, parity, 2 stop bits)

3. Data transfer operations

- SCI3 initialization

Before data is transferred on SCI3, bits TE and RE in SCR3 must first be cleared to 0, and then SCI3 must be initialized as follows.

Note: If the operation mode or data transfer format is changed, bits TE and RE must first be cleared to 0.

When bit TE is cleared to 0, bit TDRE is set to 1.

Note that the RDRF, PER, FER, and OER flags and the contents of RDR are retained when RE is cleared to 0.

When an external clock is used in asynchronous mode, the clock should not be stopped during operation, including initialization. When an external clock is used in synchronous mode, the clock should not be supplied during operation, including initialization.

Figure 10-10 shows an example of a flowchart for initializing SCI3.

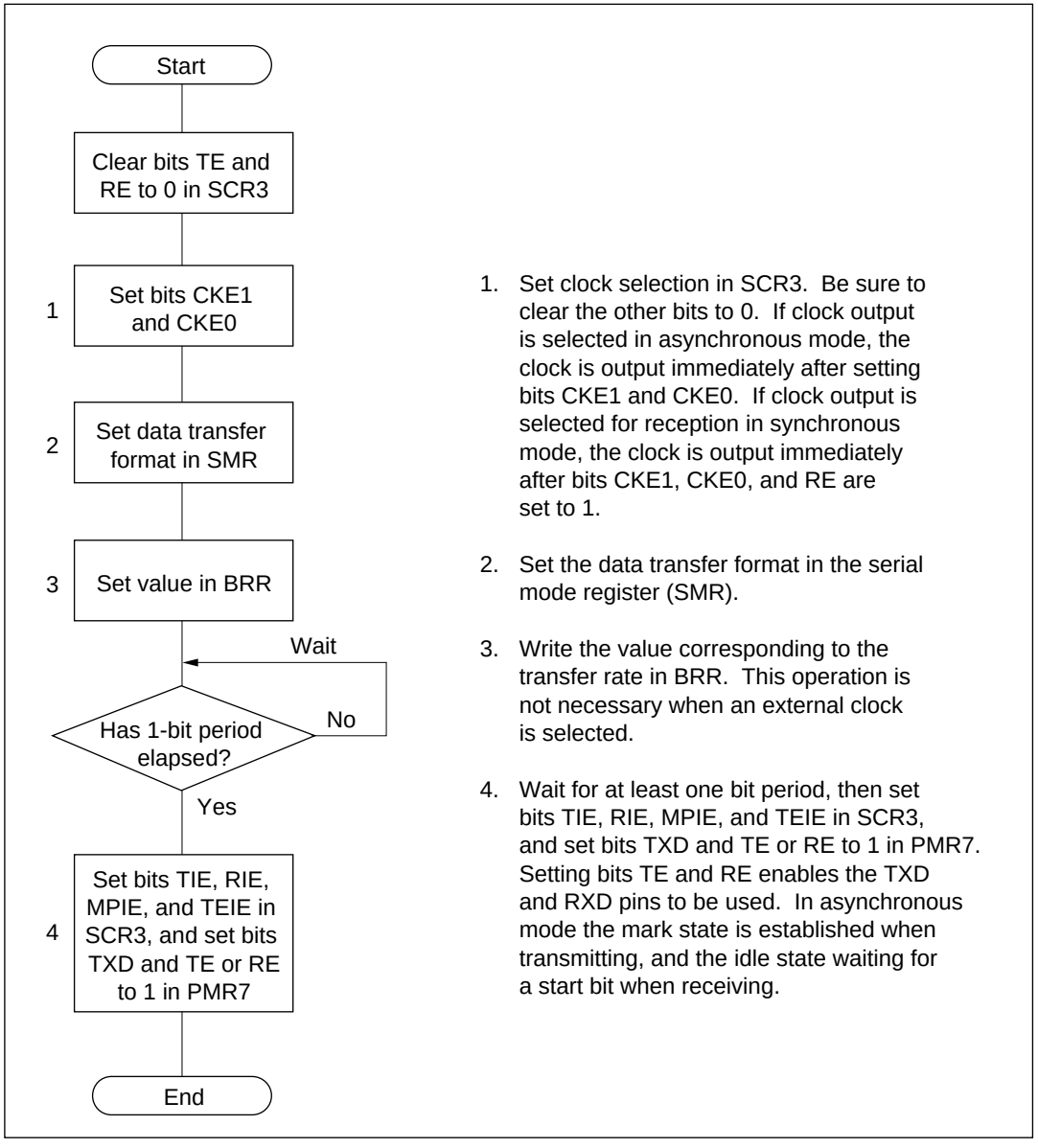


Figure 10-10 Example of SCI3 Initialization Flowchart

- Transmitting

Figure 10-11 shows an example of a flowchart for data transmission. This procedure should be followed for data transmission after initializing SCI3.

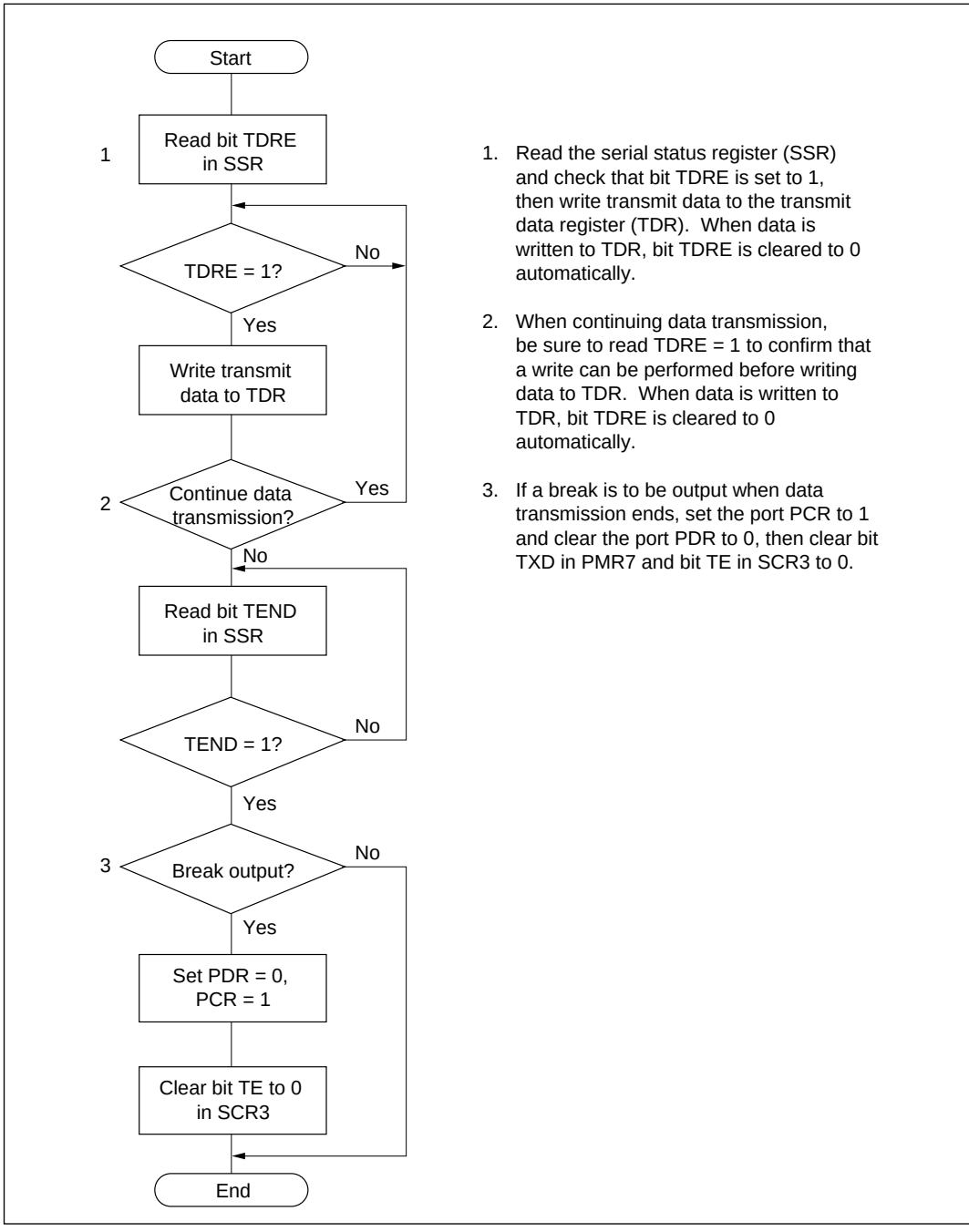


Figure 10-11 Example of Data Transmission Flowchart (Asynchronous Mode)

SCI3 operates as follows when transmitting data.

SCI3 monitors bit TDRE in SSR, and when it is cleared to 0, recognizes that data has been written to TDR and transfers data from TDR to TSR. It then sets bit TDRE to 1 and starts transmitting. If bit TIE in SCR3 is set to 1 at this time, a TXI request is made.

Serial data is transmitted from the TXD pin using the relevant data transfer format in table 10-14. When the stop bit is sent, SCI3 checks bit TDRE. If bit TDRE is cleared to 0, SCI3 transfers data from TDR to TSR, and when the stop bit has been sent, starts transmission of the next frame. If bit TDRE is set to 1, 1 is set in TEND in SSR, and the mark state, in which 1s are transmitted, is established after the stop bit has been sent. If bit TEIE in SCR3 is set to 1 at this time, a TEI request is made.

Figure 10-12 shows an example of the operation when transmitting in asynchronous mode.

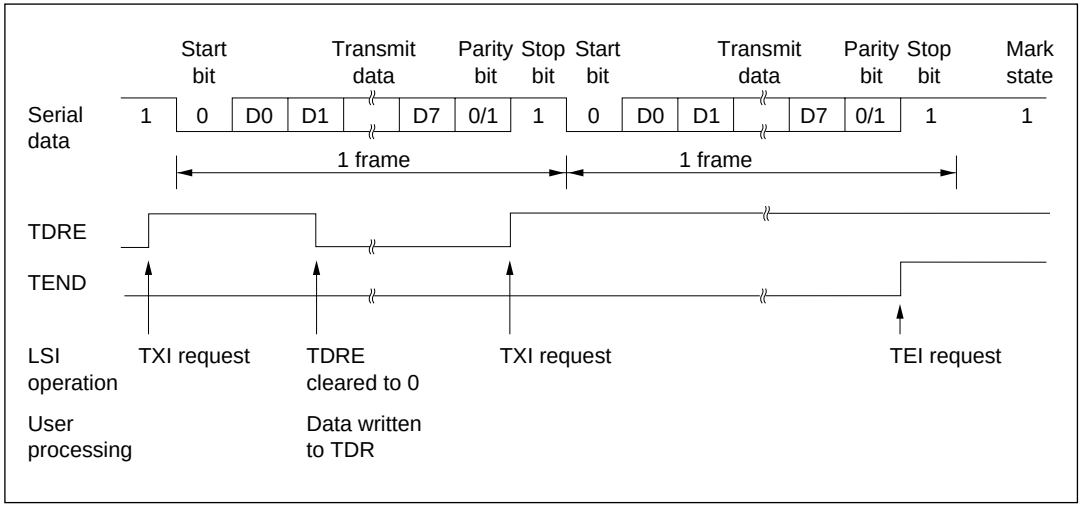


Figure 10-12 Example of Operation when Transmitting in Asynchronous Mode (8-bit data, parity, 1 stop bit)

- Receiving

Figure 10-13 shows an example of a flowchart for data reception. This procedure should be followed for data reception after initializing SCI3.

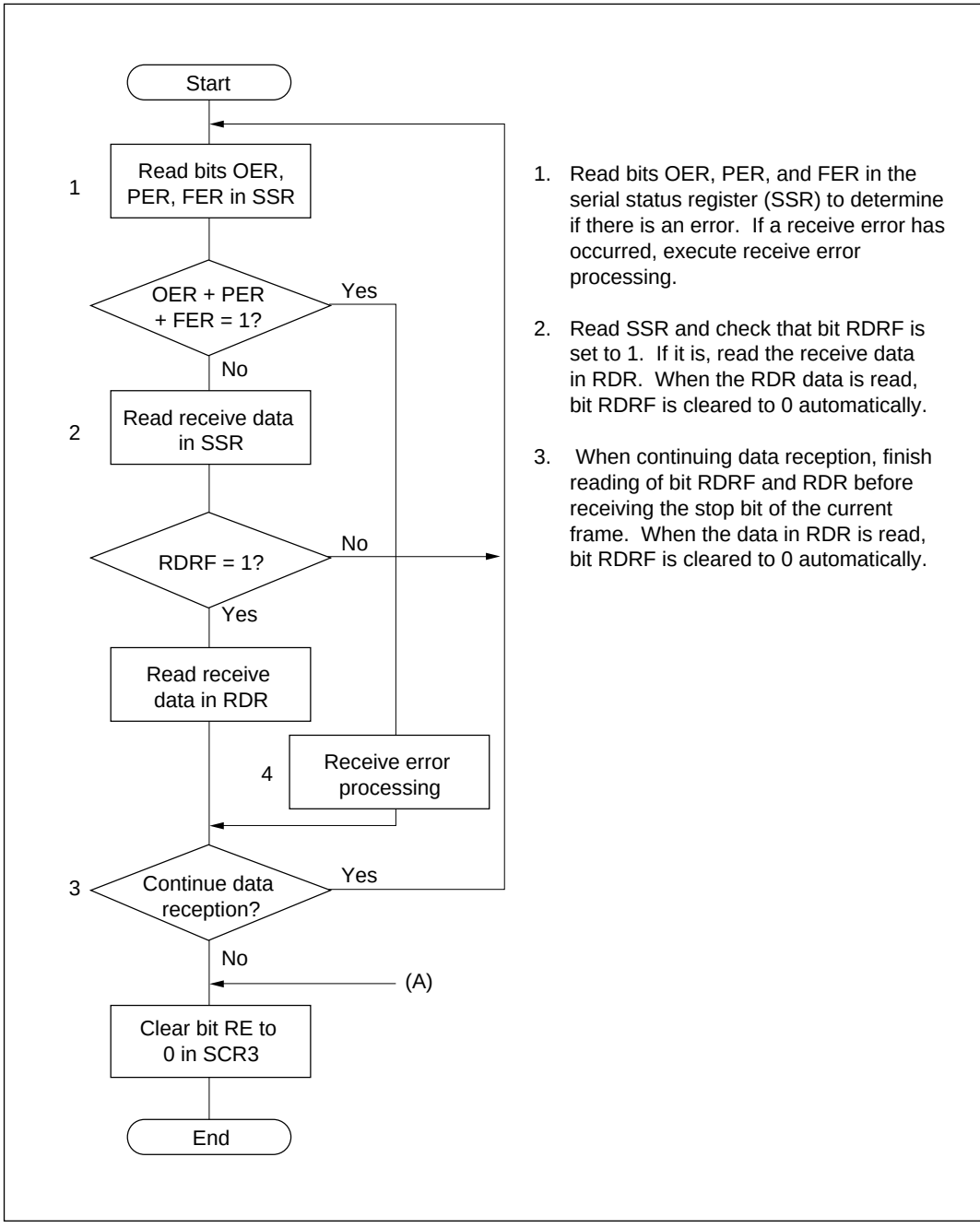


Figure 10-13 Example of Data Reception Flowchart (Asynchronous Mode)

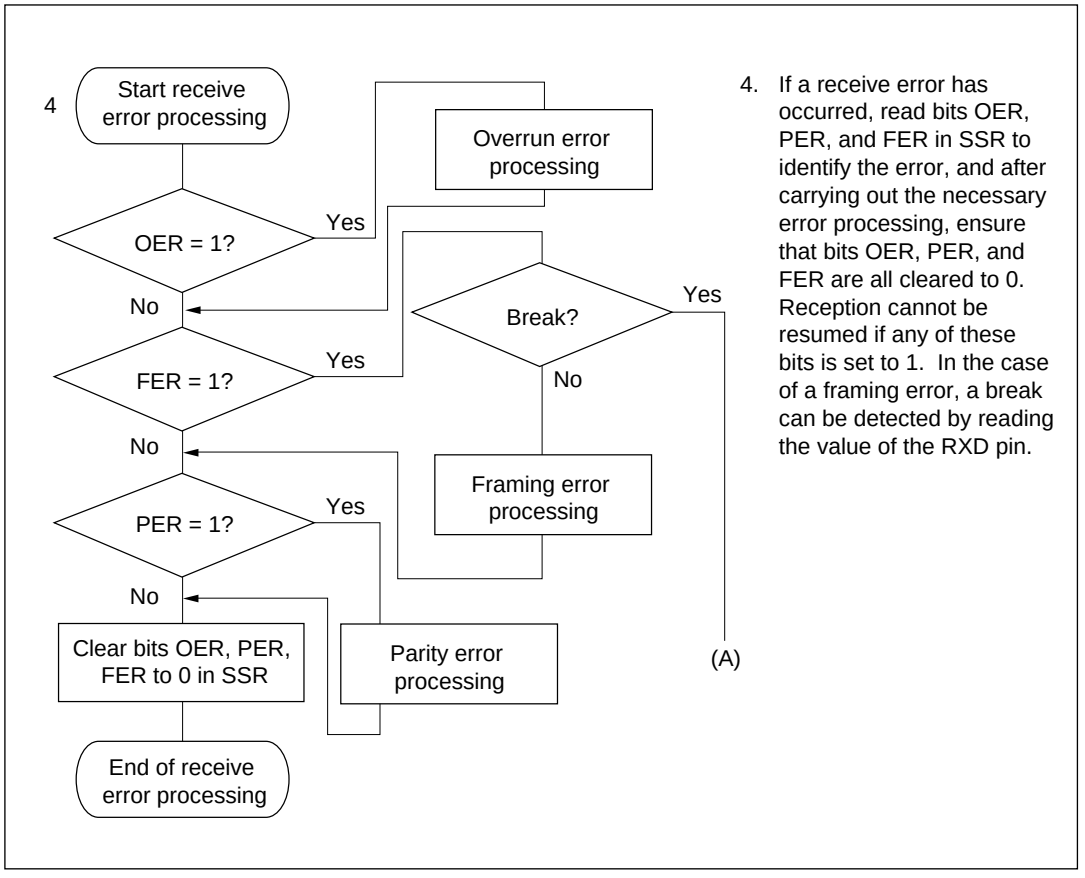


Figure 10-13 Example of Data Reception Flowchart (Asynchronous Mode) (cont)

SCI3 operates as follows when receiving data.

SCI3 monitors the communication line, and when it detects a 0 start bit, performs internal synchronization and begins reception. Reception is carried out in accordance with the relevant data transfer format in table 10-14. The received data is first placed in RSR in LSB-to-MSB order, and then the parity bit and stop bit(s) are received. SCI3 then carries out the following checks.

- Parity check

  SCI3 checks that the number of 1 bits in the receive data conforms to the parity (odd or even) set in bit PM in the serial mode register (SMR).

- Stop bit check

  SCI3 checks that the stop bit is 1. If two stop bits are used, only the first is checked.

- Status check

  SCI3 checks that bit RDRF is set to 0, indicating that the receive data can be transferred from RSR to RDR.

If no receive error is found in the above checks, bit RDRF is set to 1, and the receive data is stored in RDR. If bit RIE is set to 1 in SCR3, an RXI interrupt is requested. If the error checks identify a receive error, bit OER, PER, or FER is set to 1 depending on the kind of error. Bit RDRF retains its state prior to receiving the data. If bit RIE is set to 1 in SCR3, an ERI interrupt is requested.

Table 10-15 shows the conditions for detecting a receive error, and receive data processing.

Note: No further receive operations are possible while a receive error flag is set. Bits OER, FER, PER, and RDRF must therefore be cleared to 0 before resuming reception.

Table 10-15 Receive Error Detection Conditions and Receive Data Processing

Receive Error	Abbreviation	Detection Conditions	Receive Data Processing
Overrun error	OER	When the next date receive operation is completed while bit RDRF is still set to 1 in SSR	Receive data is not transferred from RSR to RDR
Framing error	FER	When the stop bit is 0	Receive data is transferred from RSR to RDR
Parity error	PER	When the parity (odd or even) set in SMR is different from that of the received data	Receive data is transferred from RSR to RDR

Figure 10-14 shows an example of the operation when receiving in asynchronous mode.

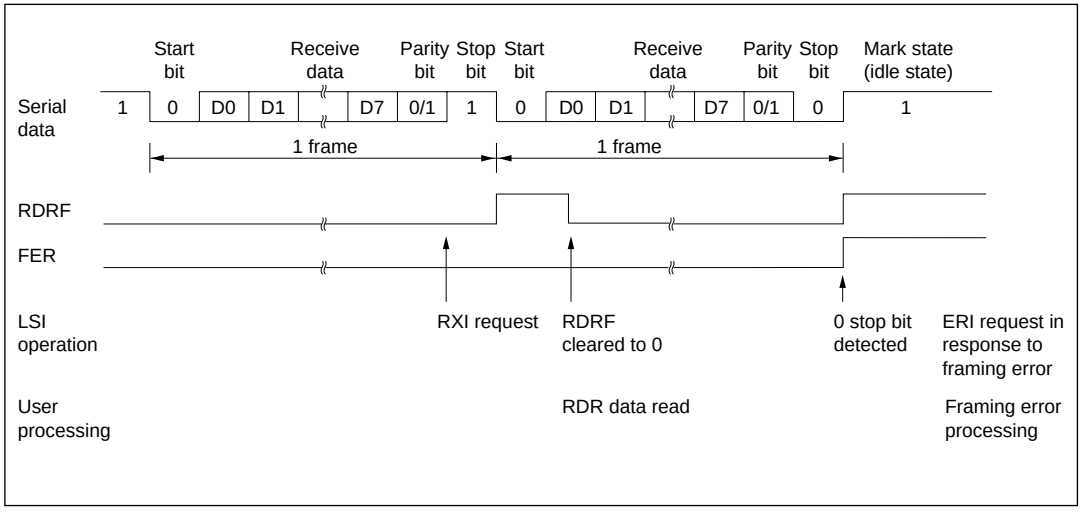


Figure 10-14 Example of Operation when Receiving in Asynchronous Mode (8-bit data, parity, 1 stop bit)

10.3.5 Operation in Synchronous Mode

In synchronous mode, SCI3 transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

SCI3 has separate transmission and reception units, allowing full-duplex communication with a shared clock.

As the transmission and reception units are both double-buffered, data can be written during transmission and read during reception, making possible continuous transmission and reception.

1. Data transfer format

The general data transfer format in asynchronous communication is shown in figure 10-15.

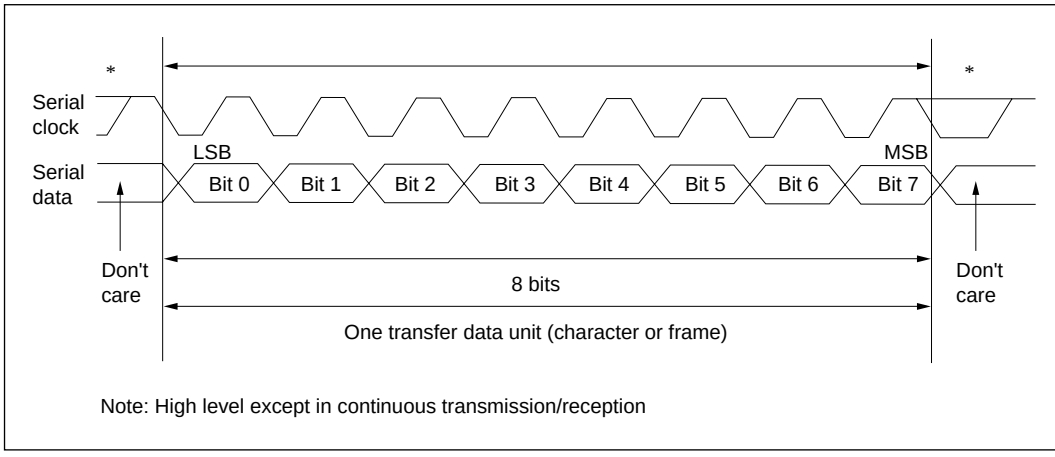


Figure 10-15 Data Format in Synchronous Communication

In synchronous communication, data on the communication line is output from one falling edge of the serial clock until the next falling edge. Data confirmation is guaranteed at the rising edge of the serial clock.

One transfer data character begins with the LSB and ends with the MSB. After output of the MSB, the communication line retains the MSB state.

When receiving in synchronous mode, SCI3 latches receive data at the rising edge of the serial clock.

The data transfer format uses a fixed 8-bit data length.

Parity and multiprocessor bits cannot be added.

2. Clock

Either an internal clock generated by the baud rate generator or an external clock input at the SCK_3 pin can be selected as the SCI3 serial clock. The selection is made by means of bit COM in SMR and bits SCE1 and CKE0 in SCR3. See table 10-12 for details on clock source selection.

When SCI3 operates on an internal clock, the serial clock is output at the SCK_3 pin. Eight pulses of the serial clock are output in transmission or reception of one character, and when SCI3 is not transmitting or receiving, the clock is fixed at the high level.

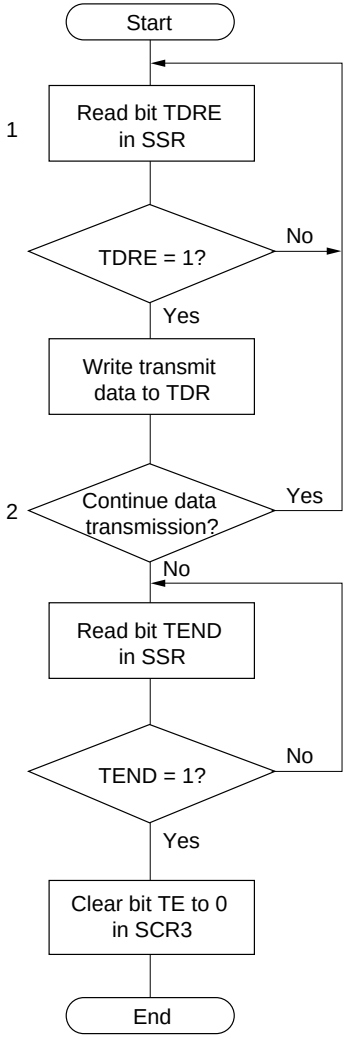
3. Data transfer operations

• SCI3 initialization

Data transfer on SCI3 first of all requires that SCI3 be initialized as described in 10.3.4, 3. SCI3 initialization, and shown in figure 10-10.

• Transmitting

Figure 10-16 shows an example of a flowchart for data transmission. This procedure should be followed for data transmission after initializing SCI3.



1. Read the serial status register (SSR) and check that bit TDRE is set to 1, then write transmit data to the transmit data register (TDR). When data is written to TDR, bit TDRE is cleared to 0 automatically, the clock is output, and data transmission is started.

2. When continuing data transmission, be sure to read TDRE = 1 to confirm that a write can be performed before writing data to TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically.

Figure 10-16 Example of Data Transmission Flowchart (Synchronous Mode)

SCI3 operates as follows when transmitting data.

SCI3 monitors bit TDRE in SSR, and when it is cleared to 0, recognizes that data has been written to TDR and transfers data from TDR to TSR. It then sets bit TDRE to 1 and starts transmitting. If bit TIE in SCR3 is set to 1 at this time, a TXI request is made.

When clock output mode is selected, SCI3 outputs 8 serial clock pulses. When an external clock is selected, data is output in synchronization with the input clock.

Serial data is transmitted from the TXD pin in order from the LSB (bit 0) to the MSB (bit 7). When the MSB (bit 7) is sent, checks bit TDRE. If bit TDRE is cleared to 0, SCI3 transfers data from TDR to TSR, and starts transmission of the next frame. If bit TDRE is set to 1, SCI3 sets bit TEND to 1 in SSR, and after sending the MSB (bit 7), retains the MSB state. If bit TEIE in SCR3 is set to 1 at this time, a TEI request is made.

After transmission ends, the SCK_3 pin is fixed at the high level.

Note: Transmission is not possible if an error flag (OER, FER, or PER) that indicates the data reception status is set to 1. Check that these error flags are all cleared to 0 before a transmit operation.

Figure 10-17 shows an example of the operation when transmitting in synchronous mode.

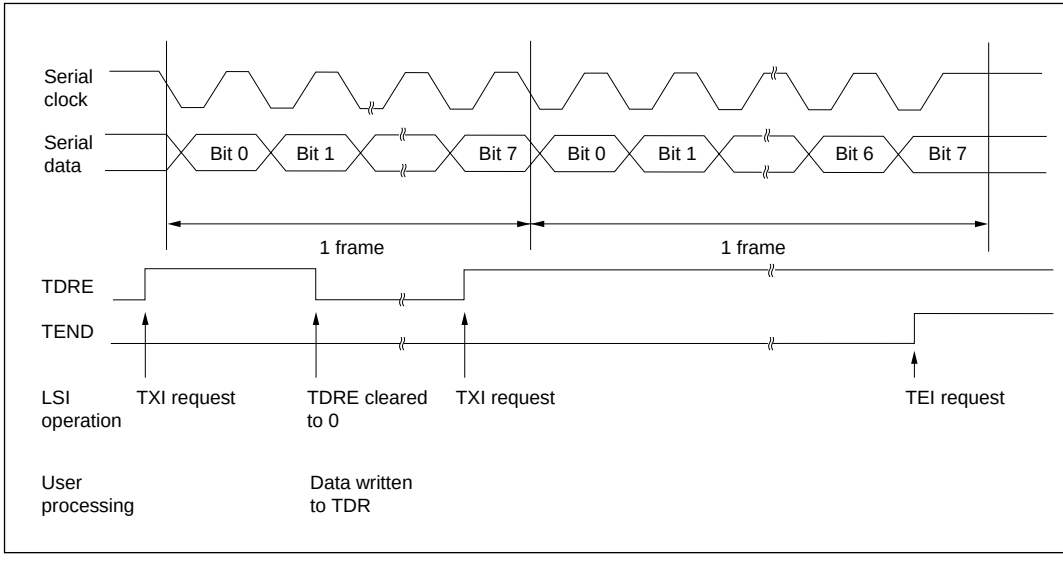


Figure 10-17 Example of Operation when Transmitting in Synchronous Mode

- Receiving

Figure 10-18 shows an example of a flowchart for data reception. This procedure should be followed for data reception after initializing SCI3.

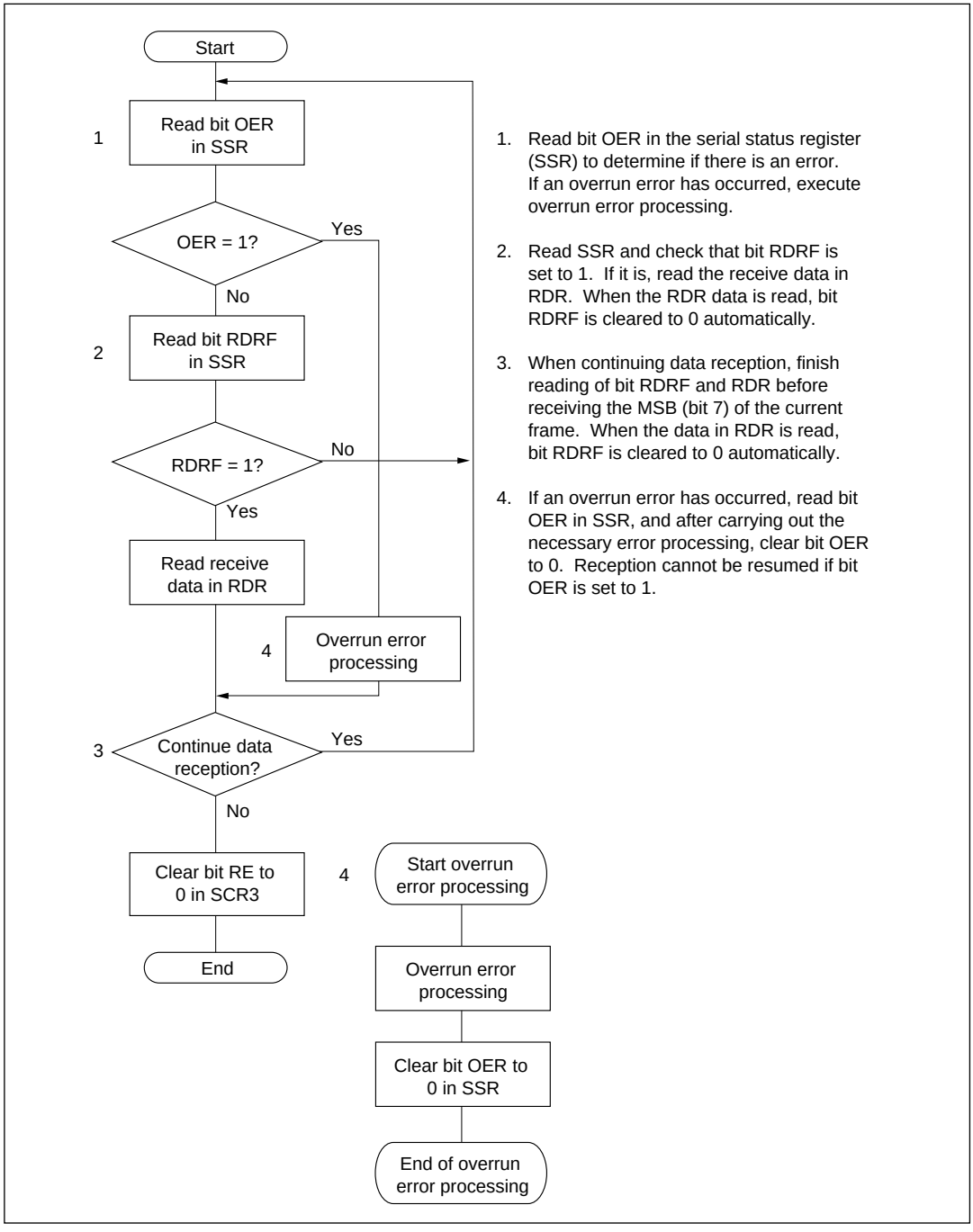


Figure 10-18 Example of Data Reception Flowchart (Synchronous Mode)

SCI3 operates as follows when receiving data.

SCI3 performs internal synchronization and begins reception in synchronization with the serial clock input or output.

The received data is placed in RSR in LSB-to-MSB order.

After the data has been received, SCI3 checks that bit RDRF is set to 0, indicating that the receive data can be transferred from RSR to RDR.

If this check shows that there is no overrun error, bit RDRF is set to 1, and the receive data is stored in RDR. If bit RIE is set to 1 in SCR3, an RXI interrupt is requested. If the check identifies an overrun error, bit OER is set to 1.

Bit RDRF remains set to 1. If bit RIE is set to 1 in SCR3, an ERI interrupt is requested.

See table 10-15 for the conditions for detecting a receive error, and receive data processing.

Note: No further receive operations are possible while a receive error flag is set. Bits OER, FER, PER, and RDRF must therefore be cleared to 0 before resuming reception.

Figure 10-19 shows an example of the operation when receiving in synchronous mode.

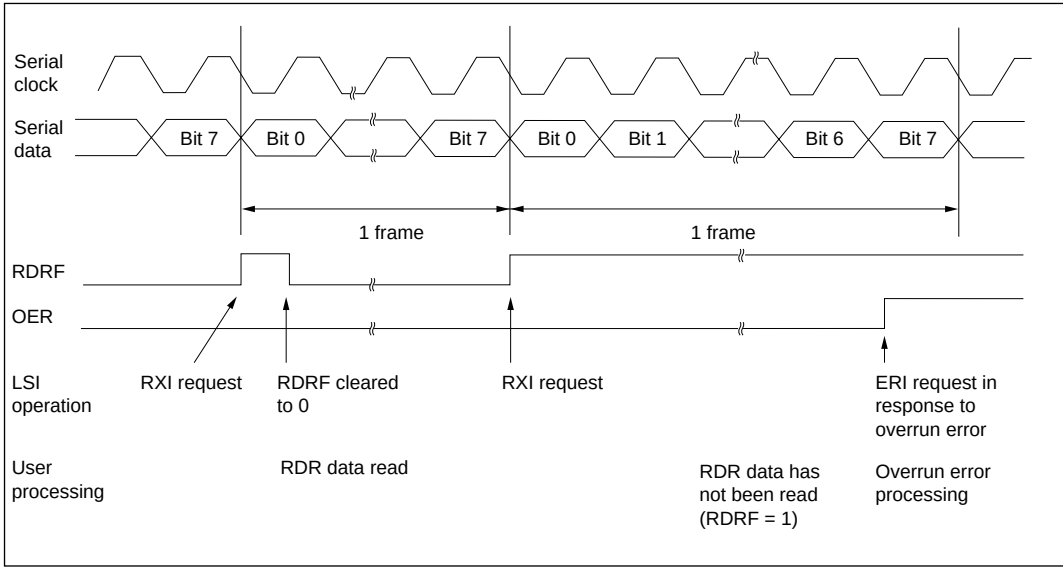


Figure 10-19 Example of Operation when Receiving in Synchronous Mode

- Simultaneous transmit/receive

Figure 10-20 shows an example of a flowchart for a simultaneous transmit/receive operation. This procedure should be followed for simultaneous transmission/reception after initializing SCI3.

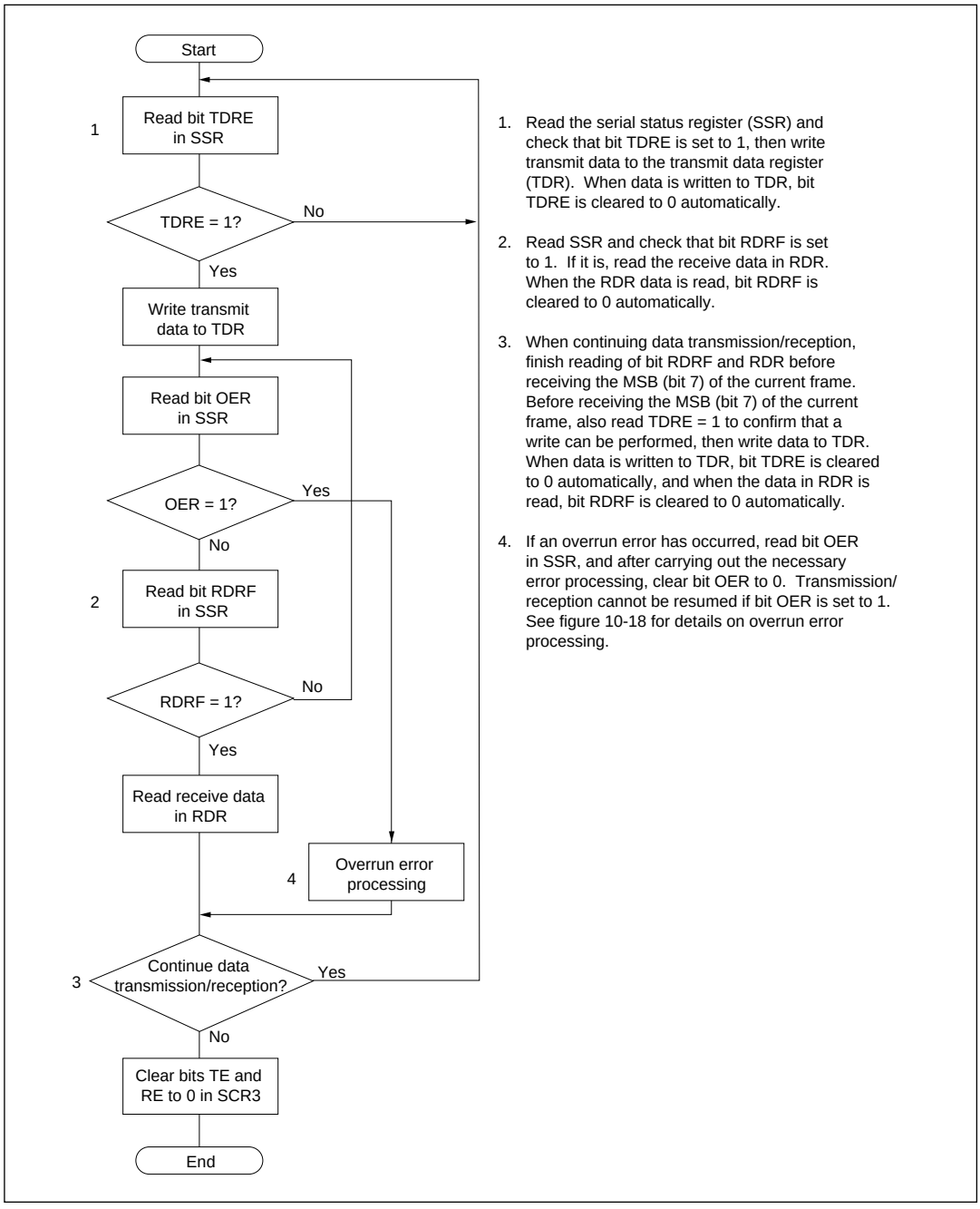


Figure 10-20 Example of Simultaneous Data Transmission/Reception Flowchart(Synchronous Mode)

Notes: 1. When switching from transmission to simultaneous transmission/reception, check that SCI3 has finished transmitting and that bits TDRE and TEND are set to 1, clear bit TE to 0, and then set bits TE and RE to 1.

2. When switching from reception to simultaneous transmission/reception, check that SCI3 has finished receiving, clear bit RE to 0, then check that bit RDRF and the error flags (OER, FER, and PER) are cleared to 0, and finally set bits TE and RE to 1.

10.3.6 Multiprocessor Communication Function

The multiprocessor communication function enables data to be exchanged among a number of processors on a shared communication line. Serial data communication is performed in asynchronous mode using the multiprocessor format (in which a multiprocessor bit is added to the transfer data).

In multiprocessor communication, each receiver is assigned its own ID code. The serial communication cycle consists of two cycles, an ID transmission cycle in which the receiver is specified, and a data transmission cycle in which the transfer data is sent to the specified receiver. These two cycles are differentiated by means of the multiprocessor bit, 1 indicating an ID transmission cycle, and 0, a data transmission cycle.

The sender first sends transfer data with a 1 multiprocessor bit added to the ID code of the receiver it wants to communicate with, and then sends transfer data with a 0 multiprocessor bit added to the transmit data. When a receiver receives transfer data with the multiprocessor bit set to 1, it compares the ID code with its own ID code, and if they are the same, receives the transfer data sent next. If the ID codes do not match, it skips the transfer data until data with the multiprocessor bit set to 1 is sent again.

In this way, a number of processors can exchange data among themselves.

Figure 10-21 shows an example of communication between processors using the multiprocessor format.

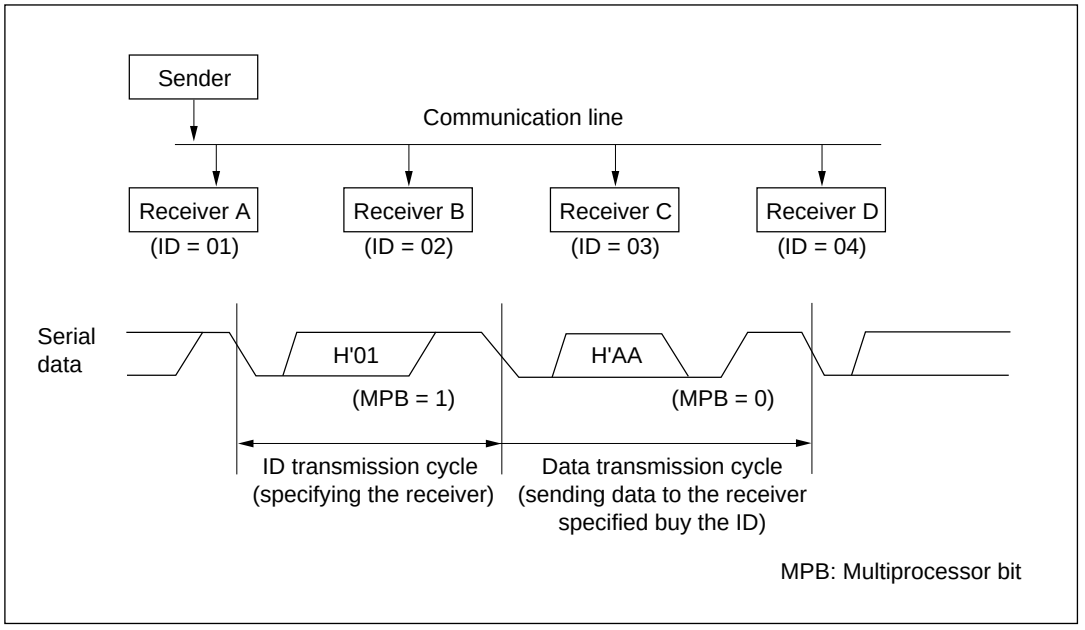


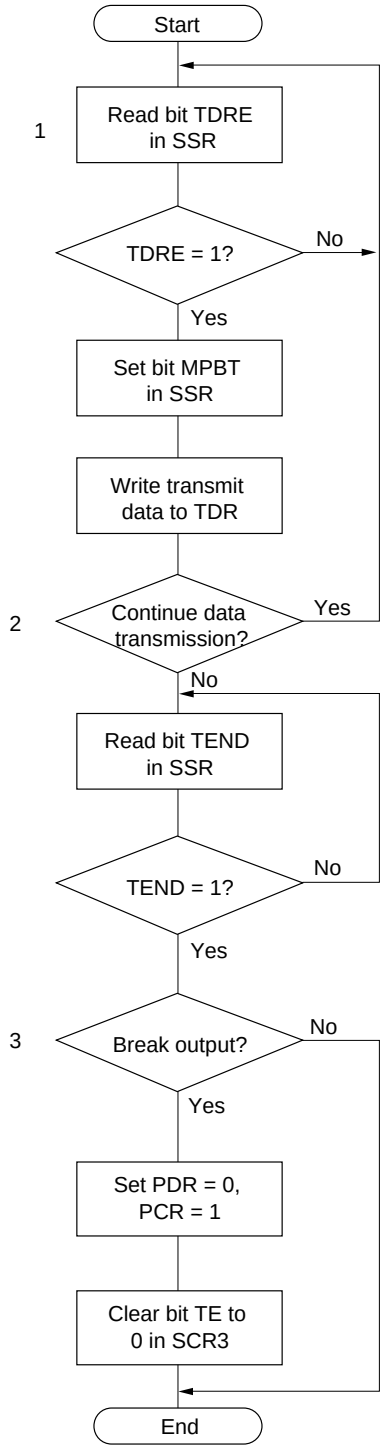
Figure 10-21 Example of Inter-Processor Communication Using Multiprocessor Format (Sending data H'AA to receiver A)

There is a choice of four data transfer formats. If a multiprocessor format is specified, the parity bit specification is invalid. See table 10-14 for details.

For details on the clock used in multiprocessor communication, see 10.3.4, Operation in Synchronous Mode.

- Multiprocessor data transmitting

Figure 10-22 shows an example of a flowchart for multiprocessor data transmission. This procedure should be followed for multiprocessor data transmission after initializing SCI3.



1. Read the serial status register (SSR) and check that bit TDRE is set to 1, then set bit MPBT in SSR to 0 or 1 and write transmit data to the transmit data register (TDR). When data is written to TDR, bit TDRE is cleared to 0 automatically.

2. When continuing data transmission, be sure to read TDRE = 1 to confirm that a write can be performed before writing data to TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically.

3. If a break is to be output when data transmission ends, set the port PCR to 1 and clear the port PDR to 0, then clear bit TE in SCR3 to 0.

Figure 10-22 Example of Multiprocessor Data Transmission Flowchart

SCI3 operates as follows when transmitting data.

SCI3 monitors bit TDRE in SSR, and when it is cleared to 0, recognizes that data has been written to TDR and transfers data from TDR to TSR. It then sets bit TDRE to 1 and starts transmitting. If bit TIE in SCR3 is set to 1 at this time, a TXI request is made.

Serial data is transmitted from the TXD pin using the relevant data transfer format in table 10-14. When the stop bit is sent, SCI3 checks bit TDRE. If bit TDRE is cleared to 0, SCI3 transfers data from TDR to TSR, and when the stop bit has been sent, starts transmission of the next frame. If bit TDRE is set to 1, 1 is set in TEND in SSR, and the mark state, in which 1s are transmitted, is established after the stop bit has been sent. If bit TEIE in SCR3 is set to 1 at this time, a TEI request is made.

Figure 10-23 shows an example of the operation when transmitting using the multiprocessor format.

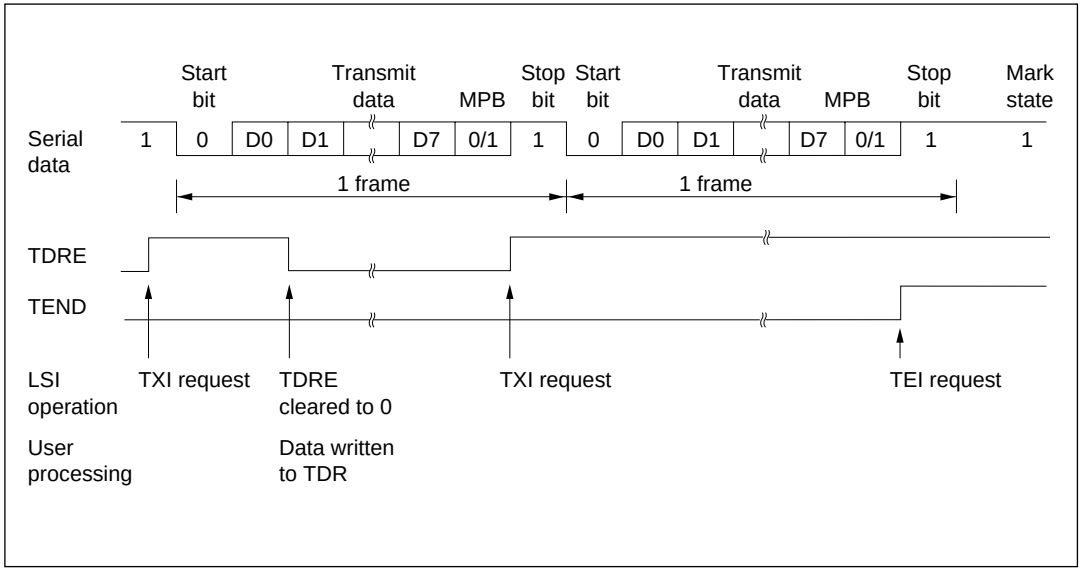
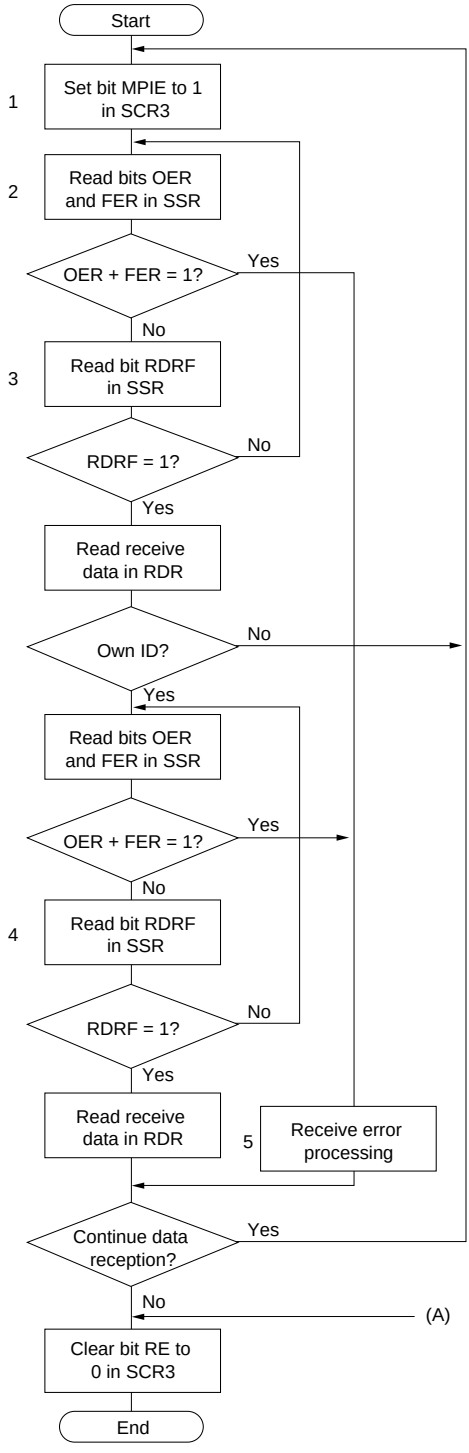


Figure 10-23 Example of Operation when Transmitting Using Multiprocessor Format (8-bit data, multiprocessor bit, 1 stop bit)

- Multiprocessor receiving

Figure 10-24 shows an example of a flowchart for multiprocessor data reception. This procedure should be followed for multiprocessor data reception after initializing SCI3.



1. Set bit MPIE to 1 in SCR3.
2. Read bits OER and FER in the serial status register (SSR) to determine if there is an error. If a receive error has occurred, execute receive error processing.
3. Read SSR and check that bit RDRF is set to 1. If it is, read the receive data in RDR and compare it with this receiver's own ID. If the ID is not this receiver's, set bit MPIE to 1 again. When the RDR data is read, bit RDRF is cleared to 0 automatically.
4. Read SSR and check that bit RDRF is set to 1, then read the data in RDR.
5. If a receive error has occurred, read bits OER and FER in SSR to identify the error, and after carrying out the necessary error processing, ensure that bits OER and FER are both cleared to 0. Reception cannot be resumed if either of these bits is set to 1. In the case of a framing error, a break can be detected by reading the value of the RXD pin.

Figure 10-24 Example of Multiprocessor Data Reception Flowchart

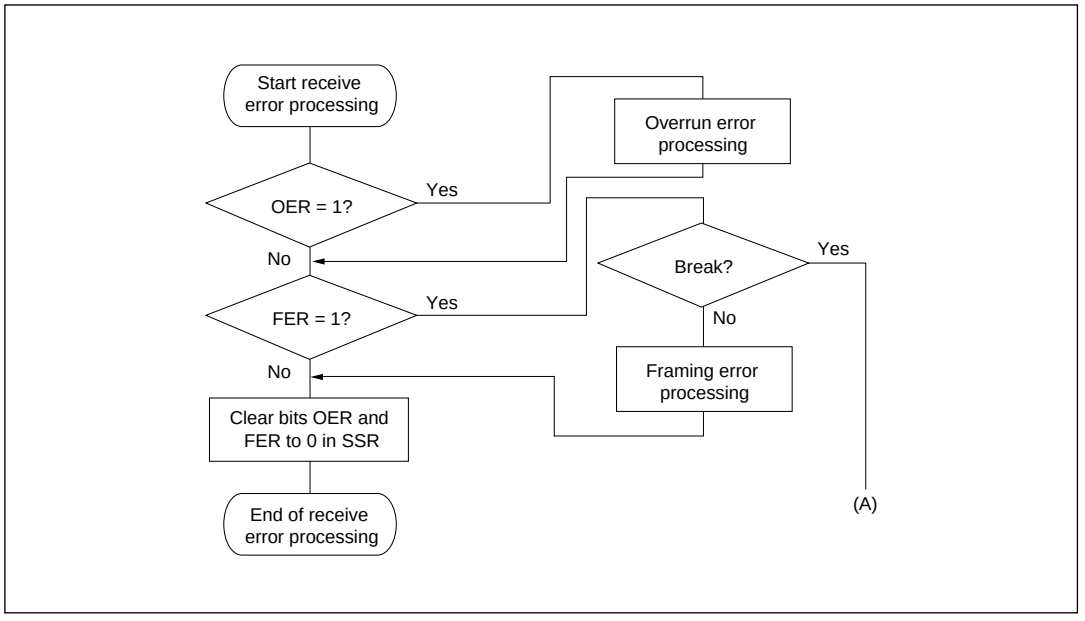
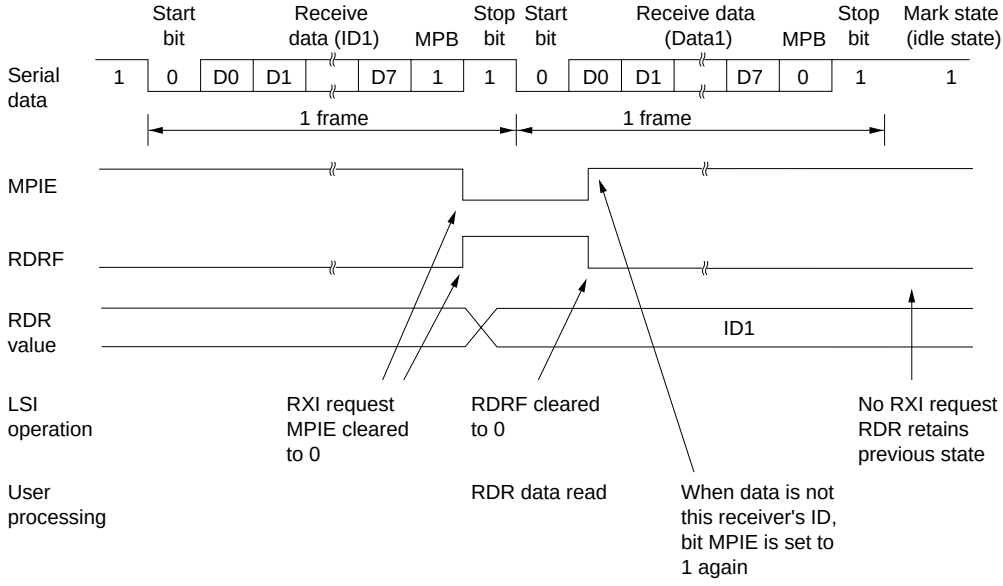
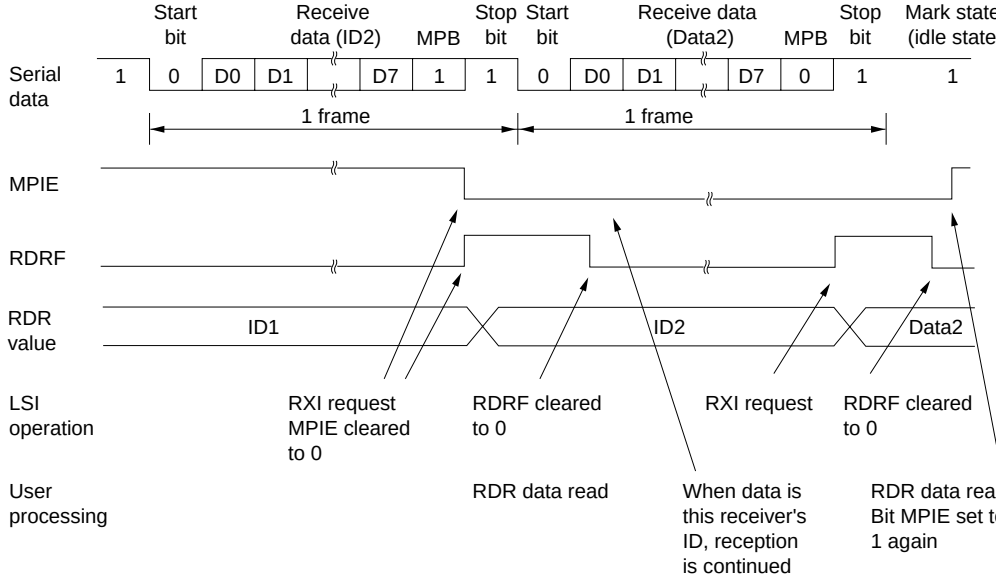


Figure 10-24 Example of Multiprocessor Data Reception Flowchart (cont)

Figure 10-25 shows an example of the operation when receiving using the multiprocessor format.



(a) When data does not match this receiver's ID



(b) When data matches this receiver's ID

Figure 10-25 Example of Operation when Receiving using Multiprocessor Format (8-bit data, multiprocessor bit, 1 stop bit)

10.3.7 Interrupts

SCI3 can generate six kinds of interrupts: transmit end, transmit data empty, receive data full, and three receive error interrupts (overrun error, framing error, and parity error). These interrupts have the same vector address.

The various interrupt requests are shown in table 10-16.

Table 10-16 SCI3 Interrupt Requests

Interrupt Abbreviation	Interrupt Request	Vector Address
RXI	Interrupt request initiated by receive data full flag (RDRF)	H'0024
TXI	Interrupt request initiated by transmit data empty flag (TDRE)	
TEI	Interrupt request initiated by transmit end flag (TEND)	
ERI	Interrupt request initiated by receive error flag (OER, FER, PER)	

Each interrupt request can be enabled or disabled by means of bits TIE and RIE in SCR3.

When bit TDRE is set to 1 in SSR, a TXI interrupt is requested. When bit TEND is set to 1 in SSR, a TEI interrupt is requested. These two interrupts are generated during transmission.

The initial value of bit TDRE in SSR is 1. Therefore, if the transmit data empty interrupt request (TXI) is enabled by setting bit TIE to 1 in SCR3 before transmit data is transferred to TDR, a TXI interrupt will be requested even if the transmit data is not ready.

Also, the initial value of bit TEND in SSR is 1. Therefore, if the transmit end interrupt request (TEI) is enabled by setting bit TEIE to 1 in SCR3 before transmit data is transferred to TDR, a TEI interrupt will be requested even if the transmit data has not been sent.

Effective use of these interrupt requests can be made by having processing that transfers transmit data to TDR carried out in the interrupt service routine.

To prevent the generation of these interrupt requests (TXI and TEI), on the other hand, the enable bits for these interrupt requests (bits TIE and TEIE) should be set to 1 after transmit data has been transferred to TDR.

When bit RDRF is set to 1 in SSR, an RXI interrupt is requested, and if any of bits OER, PER, and FER is set to 1, an ERI interrupt is requested. These two interrupt requests are generated during reception.

For further details, see 3.3, Interrupts.

10.3.8 Application Notes

The following points should be noted when using SCI3.

1. Relation between writes to TDR and bit TDRE

Bit TDRE in the serial status register (SSR) is a status flag that indicates that data for serial transmission has not been prepared in TDR. When data is written to TDR, bit TDRE is cleared to 0 automatically. When SCI3 transfers data from TDR to TSR, bit TDRE is set to 1.

Data can be written to TDR irrespective of the state of bit TDRE, but if new data is written to TDR while bit TDRE is cleared to 0, the data previously stored in TDR will be lost of it has not yet been transferred to TSR. Accordingly, to ensure that serial transmission is performed dependably, you should first check that bit TDRE is set to 1, then write the transmit data to TDR once only (not two or more times).

2. Operation when a number of receive errors occur simultaneously

If a number of receive errors are detected simultaneously, the status flags in SSR will be set to the states shown in table 10-17. If an overrun error is detected, data transfer from RSR to RDR will not be performed, and the receive data will be lost.

Table 10-17 SSR Status Flag States and Receive Data Transfer

SSR Status Flags				Receive Data Transfer	
RDRF*	OER	FER	PER	RSR → RDR	Receive Error Status
1	1	0	0	×	Overrun error
0	0	1	0	○	Framing error
0	0	0	1	○	Parity error
1	1	1	0	×	Overrun error + framing error
1	1	0	1	×	Overrun error + parity error
0	0	1	1	○	Framing error + parity error
1	1	1	1	×	Overrun error + framing error + parity error

○: Receive data is transferred from RSR to RDR.

×: Receive data is not transferred from RSR to RDR.

Note: * Bit RDRF retains its state prior to data reception.

3. Break detection and processing

When a framing error is detected, a break can be detected by reading the value of the RXD pin directly. In a break, the input from the RXD pin becomes all 0s, with the result that bit FER is set and bit PER may also be set.

SCI3 continues the receive operation even after receiving a break. Note, therefore, that even though bit FER is cleared to 0 it will be set to 1 again.

4. Mark state and break detection

When bit TE is cleared to 0, the TXD pin functions as an I/O port whose input/output direction and level are determined by PDR and PCR. This fact can be used to set the TXD pin to the mark state, or to detect a break during transmission.

To keep the communication line in the mark state (1 state) until bit TE is set to 1, set PCR = 1 and PDR = 1. Since bit TE is cleared to 0 at this time, the TXD pin functions as an I/O port and 1 is output.

To detect a break, clear bit TE to 0 after setting PCR = 1 and PDR = 0.

When bit TE is cleared to 0, the transmission unit is initialized regardless of the current transmission state, the TXD pin functions as an I/O port, and 0 is output from the TXD pin.

5. Receive error flags and transmit operation (synchronous mode only)

When a receive error flag (OER, PER, or FER) is set to 1, transmission cannot be started even if bit TDRE is cleared to 0. The receive error flags must be cleared to 0 before starting transmission.

Note also that receive error flags cannot be cleared to 0 even if bit RE is cleared to 0.

6. Receive data sampling timing and receive margin in asynchronous mode

In asynchronous mode, SCI3 operates on a basic clock with a frequency 16 times the transfer rate. When receiving, SCI3 performs internal synchronization by sampling the falling edge of the start bit with the basic clock. Receive data is latched internally at the 8th rising edge of the basic clock. This is illustrated in figure 10-26.

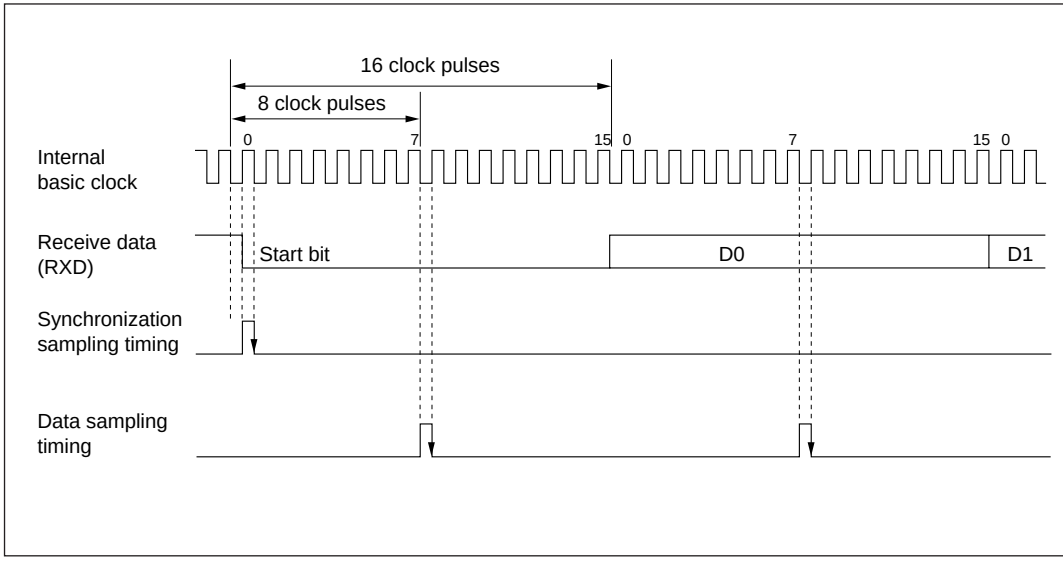


Figure 10-26 Receive Data Sampling Timing in Asynchronous Mode

Consequently, the receive margin in asynchronous mode can be expressed as shown in equation (1).

$$M = \{(0.5 - \frac{1}{2N}) - \frac{D - 0.5}{N} - (L - 0.5)\,F\} \times 100\ [\%] \quad \text{..... Equation (1)}$$

where

M: Receive margin (%)
N: Ratio of bit rate to clock (N = 16)
D: Clock duty (D = 0.5 to 1.0)
L: Frame length (L = 9 to 12)
F: Absolute value of clock frequency deviation

Substituting 0 for F (absolute value of clock frequency deviation) and 0.5 for D (clock duty) in equation (1), a receive margin of 46.875% is given by equation (2).

When D = 0.5 and F = 0,
$M = \{0.5 - 1/(2 \times 16)\} \times 100\ [\%]$
$= 46.875\%$ ..... Equation (2)

However, this is only a computed value, and a margin of 20% to 30% should be allowed when carrying out system design.

7. Relation between RDR reads and bit RDRF

In a receive operation, SCI3 continually checks the RDRF flag. If bit RDRF is cleared to 0 when reception of one frame ends, normal data reception is completed. If bit RDRF is set to 1, this indicates that an overrun error has occurred.

When the contents of RDR are read, bit RDRF is cleared to 0 automatically. Therefore, if bit RDR is read more than once, the second and subsequent read operations will be performed while bit RDRF is cleared to 0. Note that, when an RDR read is performed while bit RDRF is cleared to 0, if the read operation coincides with completion of reception of a frame, the next frame of data may be read. This is illustrated in figure 10-27.

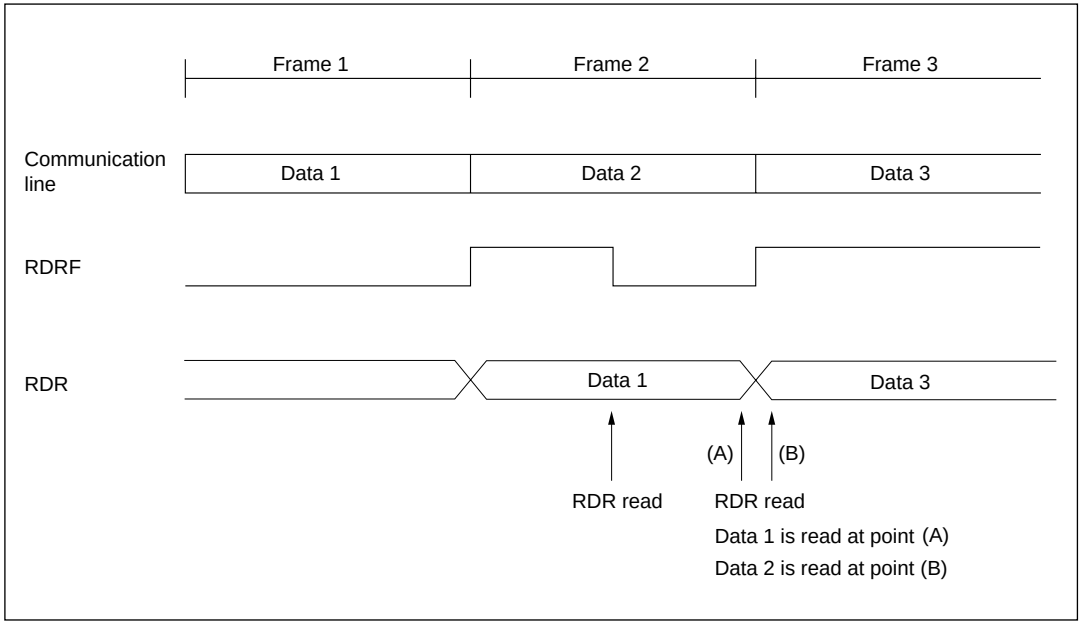


Figure 10-27 Relation between RDR Read Timing and Data

In this case, only a single RDR read operation (not two or more) should be performed after first checking that bit RDRF is set to 1. If two or more reads are performed, the data read the first time should be transferred to RAM, etc., and the RAM contents used. Also, ensure that there is sufficient margin in an RDR read operation before reception of the next frame is completed. To be precise in terms of timing, the RDR read should be completed before bit 7 is transferred in synchronous mode, or before the STOP bit is transferred in asynchronous mode.

Section 11 14-Bit PWM

11.1 Overview

The H8/3657 Series is provided with a 14-bit PWM (pulse width modulator) on-chip, which can be used as a D/A converter by connecting a low-pass filter.

11.1.1 Features

Features of the 14-bit PWM are as follows.

- Choice of two conversion periods

A conversion period of 32,768/ø, with a minimum modulation width of 2/ø or a conversion period of 16,384/ø, with a minimum modulation width of 1/ø can be chosen.

- Pulse division method for less ripple

11.1.2 Block Diagram

Figure 11-1 shows a block diagram of the 14-bit PWM.

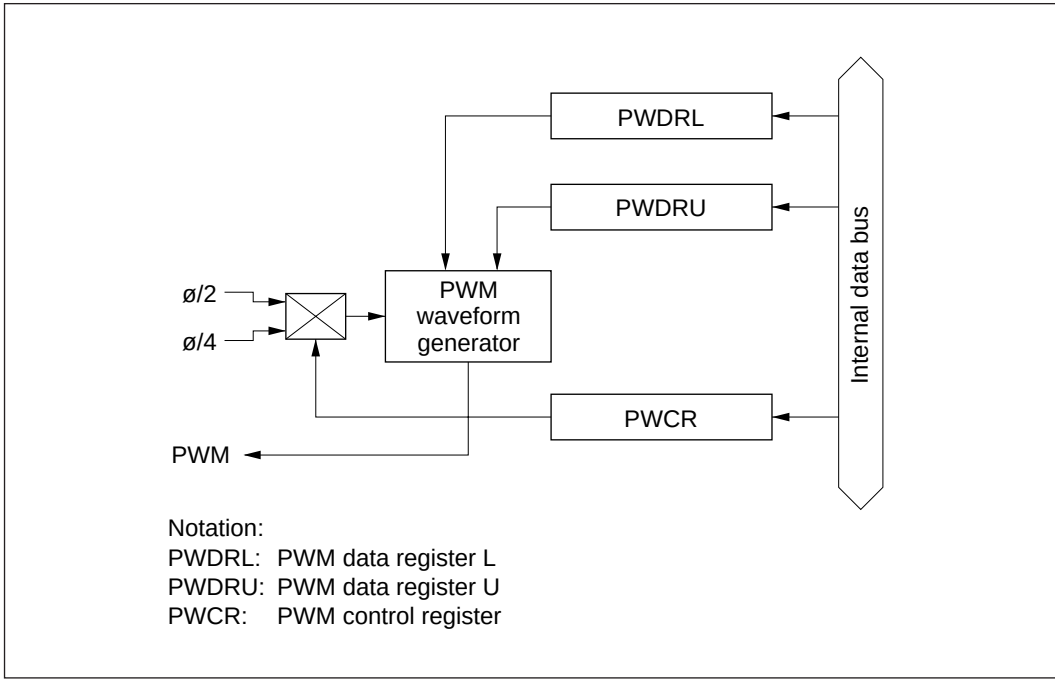


Figure 11-1 Block Diagram of the 14 bit PWM

11.1.3 Pin Configuration

Table 11-1 shows the output pin assigned to the 14-bit PWM.

Table 11-1 Pin Configuration

Name	Abbrev.	I/O	Function
PWM output pin	PWM	Output	Pulse-division PWM waveform output

11.1.4 Register Configuration

Table 11-2 shows the register configuration of the 14-bit PWM.

Table 11-2 Register Configuration

Name	Abbrev.	R/W	Initial Value	Address
PWM control register	PWCR	W	H'FE	H'FFD0
PWM data register U	PWDRU	W	H'C0	H'FFD1
PWM data register L	PWDRL	W	H'00	H'FFD2

11.2 Register Descriptions

11.2.1 PWM Control Register (PWCR)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWCR0
Initial value	1	1	1	1	1	1	1	0
Read/Write	—	—	—	—	—	—	—	W

PWCR is an 8-bit write-only register for input clock selection.

Upon reset, PWCR is initialized to H'FE.

Bits 7 to 1: Reserved bits

Bits 7 to 1 are reserved; they are always read as 1, and cannot be modified.

Bit 0: Clock select 0 (PWCR0)

Bit 0 selects the clock supplied to the 14-bit PWM. This bit is a write-only bit; it is always read as 1.

Bit 0 PWCR0	Description	
0	The input clock is ø/2 ($t_ø$ = 2/ø). The conversion period is 16,384/ø, with a minimum modulation width of 1/ø.	(initial value)
1	The input clock is ø/4 ($t_ø$ = 4/ø). The conversion period is 32,768/ø, with a minimum modulation width of 2/ø.	

Notation:
$t_ø$: Period of PWM input clock

11.2.2 PWM Data Registers U and L (PWDRU, PWDRL)

PWDRU

Bit	7	6	5	4	3	2	1	0
	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

PWDRL

Bit	7	6	5	4	3	2	1	0
	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PWDRU and PWDRL form a 14-bit write-only register, with the upper 6 bits assigned to PWDRU and the lower 8 bits to PWDRL. The value written to PWDRU and PWDRL gives the total high-level width of one PWM waveform cycle.

When 14-bit data is written to PWDRU and PWDRL, the register contents are latched in the PWM waveform generator, updating the PWM waveform generation data. The 14-bit data should always be written in the following sequence:

1. Write the lower 8 bits to PWDRL.

2. Write the upper 6 bits to PWDRU.

PWDRU and PWDRL are write-only registers. If they are read, all bits are read as 1.

Upon reset, PWDRU and PWDRL are initialized to H'C000.

11.3 Operation

When using the 14-bit PWM, set the registers in the following sequence.

1. Set bit PWM in port mode register 1 (PMR1) to 1 so that pin $P1_4$/PWM is designated for PWM output.

2. Set bit PWCR0 in the PWM control register (PWCR) to select a conversion period of either 32,768/ø (PWCR0 = 1) or 16,384/ø (PWCR0 = 0).

3. Set the output waveform data in PWM data registers U and L (PWDRU/L). Be sure to write in the correct sequence, first PWDRL then PWDRU. When data is written to PWDRU, the data in these registers will be latched in the PWM waveform generator, updating the PWM waveform generation in synchronization with internal signals.

   One conversion period consists of 64 pulses, as shown in figure 11-2. The total of the high-level pulse widths during this period (T_H) corresponds to the data in PWDRU and PWDRL. This relation can be represented as follows.

   $$T_H = (\text{data value in PWDRU and PWDRL} + 64) \times t_{ø}/2$$

   where $t_{ø}$ is the PWM input clock period, either 2/ø (bit PWCR0 = 0) or 4/ø (bit PWCR0 = 1).

   Example: Settings in order to obtain a conversion period of 8,192 µs:

   When bit PWCR0 = 0, the conversion period is 16,384/ø, so ø must be 2 MHz. In this case t_{fn} = 128 µs, with 1/ø (resolution) = 0.5 µs.

   When bit PWCR0 = 1, the conversion period is 32,768/ø, so ø must be 4 MHz. In this case t_{fn} = 128 µs, with 2/ø (resolution) = 0.5 µs.

   Accordingly, for a conversion period of 8,192 µs, the system clock frequency (ø) must be 2 MHz or 4 MHz.

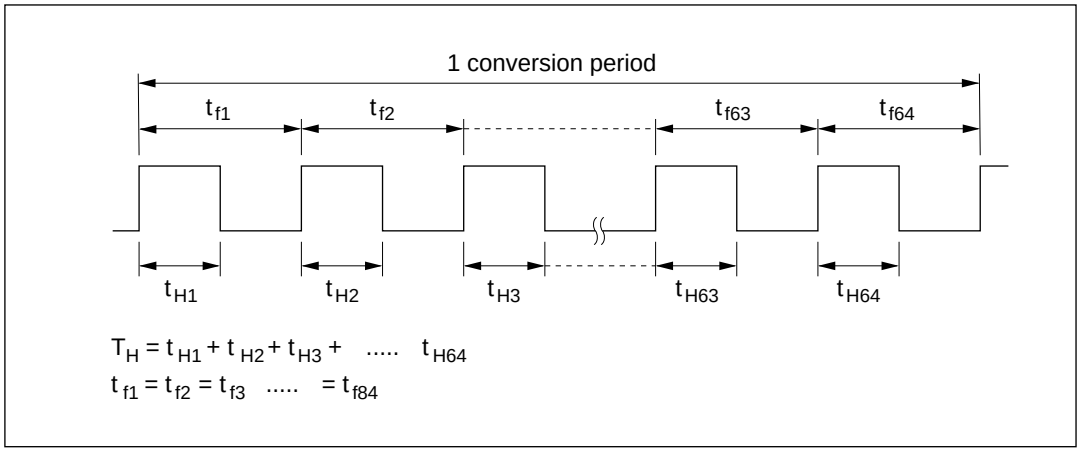


Figure 11-2 PWM Output Waveform

Section 12 A/D Converter

12.1 Overview

The H8/3657 Series includes on-chip a resistance-ladder-based successive-approximation analog-to-digital converter, and can convert up to 8 channels of analog input.

12.1.1 Features

The A/D converter has the following features.

- 8-bit resolution
- Eight input channels
- Conversion time: approx. 12.4 µs per channel (at 5 MHz operation)
- Built-in sample-and-hold function
- Interrupt requested on completion of A/D conversion
- A/D conversion can be started by external trigger input

12.1.2 Block Diagram

Figure 12-1 shows a block diagram of the A/D converter.

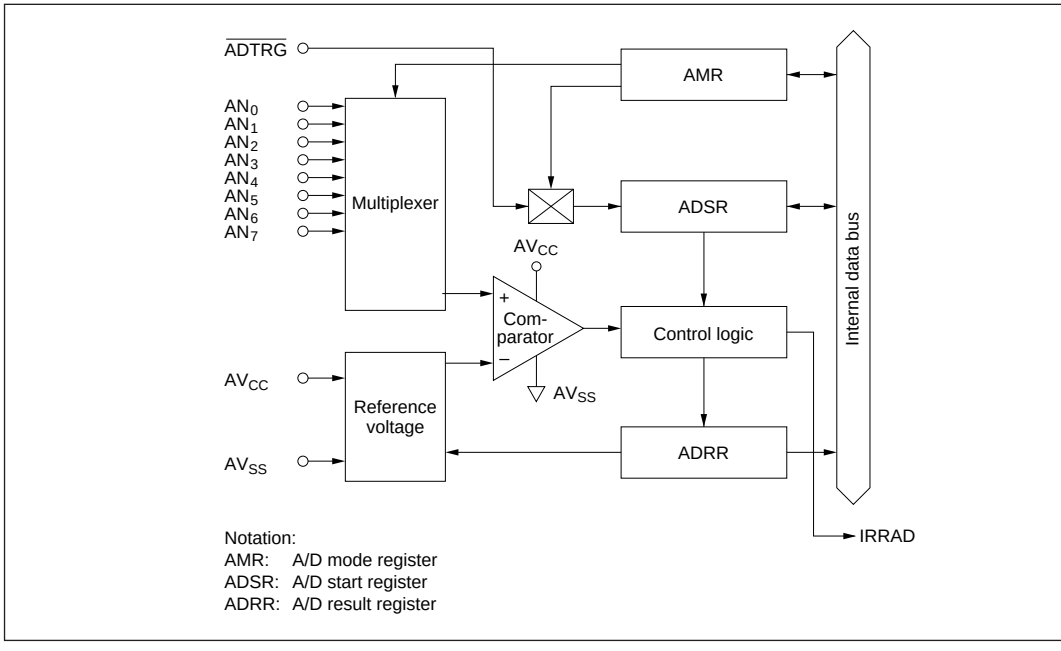


Figure 12-1 Block Diagram of the A/D Converter

12.1.3 Pin Configuration

Table 12-1 shows the A/D converter pin configuration.

Table 12-1 Pin Configuration

Name	Abbrev.	I/O	Function
Analog power supply	AV_{CC}	Input	Power supply and reference voltage of analog part
Analog ground	AV_{SS}	Input	Ground and reference voltage of analog part
Analog input 0	AN_0	Input	Analog input channel 0
Analog input 1	AN_1	Input	Analog input channel 1
Analog input 2	AN_2	Input	Analog input channel 2
Analog input 3	AN_3	Input	Analog input channel 3
Analog input 4	AN_4	Input	Analog input channel 4
Analog input 5	AN_5	Input	Analog input channel 5
Analog input 6	AN_6	Input	Analog input channel 6
Analog input 7	AN_7	Input	Analog input channel 7
External trigger input	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

12.1.4 Register Configuration

Table 12-2 shows the A/D converter register configuration.

Table 12-2 Register Configuration

Name	Abbrev.	R/W	Initial Value	Address
A/D mode register	AMR	R/W	H'30	H'FFC4
A/D start register	ADSR	R/W	H'7F	H'FFC6
A/D result register	ADRR	R	Not fixed	H'FFC5

12.2 Register Descriptions

12.2.1 A/D Result Register (ADRR)

Bit	7	6	5	4	3	2	1	0
	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R	R	R	R	R	R	R	R

The A/D result register (ADRR) is an 8-bit read-only register for holding the results of analog-to-digital conversion.

ADRR can be read by the CPU at any time, but the ADRR values during A/D conversion are not fixed.

After A/D conversion is complete, the conversion result is stored in ADRR as 8-bit data; this data is held in ADRR until the next conversion operation starts.

ADRR is not cleared on reset.

12.2.2 A/D Mode Register (AMR)

Bit	7	6	5	4	3	2	1	0
	CKS	TRGE	—	—	CH3	CH2	CH1	CH0
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

AMR is an 8-bit read/write register for specifying the A/D conversion speed, external trigger option, and the analog input pins.

Upon reset, AMR is initialized to H'30.

Bit 7: Clock select (CKS)

Bit 7 sets the A/D conversion speed.

Bit 7		**Conversion Time**	
CKS	**Conversion Period**	**ø = 2 MHz**	**ø = 5 MHz**
0	62/ø (initial value)	31 µs	12.4 µs
1	31/ø	15.5 µs	*

Note: * Operation is not guaranteed if the conversion time is less than 12.4 µs. Set bit 7 for a value of at least 12.4 µs.

Bit 6: External trigger select (TRGE)

Bit 6 enables or disables the start of A/D conversion by external trigger input.

Bit 6 TRGE	Description	
0	Disables start of A/D conversion by external trigger	(initial value)
1	Enables start of A/D conversion by rising or falling edge of external trigger at pin $\overline{\text{ADTRG}}$*	

Note: * The external trigger ($\overline{\text{ADTRG}}$) edge is selected by bit INTEG5 of IEGR2. See 3.3.2 for details.

Bits 5 and 4: Reserved bits

Bits 5 and 4 are reserved; they are always read as 1, and cannot be modified.

Bits 3 to 0: Channel select (CH3 to CH0)

Bits 3 to 0 select the analog input channel.

The channel selection should be made while bit ADSF is cleared to 0.

Bit 3 CH3	Bit 2 CH2	Bit 1 CH1	Bit 0 CH0	Analog Input Channel	
0	0	*	*	No channel selected	(initial value)
0	1	0	0	AN_0	
0	1	0	1	AN_1	
0	1	1	0	AN_2	
0	1	1	1	AN_3	
1	0	0	0	AN_4	
1	0	0	1	AN_5	
1	0	1	0	AN_6	
1	0	1	1	AN_7	
1	1	0	0	Reserved	
1	1	0	1	Reserved	
1	1	1	0	Reserved	
1	1	1	1	Reserved	

Note: * Don’t care

12.2.3 A/D Start Register (ADSR)

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

The A/D start register (ADSR) is an 8-bit read/write register for starting and stopping A/D conversion.

A/D conversion is started by writing 1 to the A/D start flag (ADSF) or by input of the designated edge of the external trigger signal, which also sets ADSF to 1. When conversion is complete, the converted data is set in the A/D result register (ADRR), and at the same time ADSF is cleared to 0.

Bit 7: A/D start flag (ADSF)

Bit 7 controls and indicates the start and end of A/D conversion.

Bit 7 ADSF	Description	
0	Read: Indicates the completion of A/D conversion	(initial value)
	Write: Stops A/D conversion	
1	Read: Indicates A/D conversion in progress	
	Write: Starts A/D conversion	

Bits 6 to 0: Reserved bits

Bits 6 to 0 are reserved; they are always read as 1, and cannot be modified.

12.3 Operation

12.3.1 A/D Conversion Operation

The A/D converter operates by successive approximations, and yields its conversion result as 8-bit data.

A/D conversion begins when software sets the A/D start flag (bit ADSF) to 1. Bit ADSF keeps a value of 1 during A/D conversion, and is cleared to 0 automatically when conversion is complete.

The completion of conversion also sets bit IRRAD in interrupt request register 2 (IRR2) to 1. An A/D conversion end interrupt is requested if bit IENAD in interrupt enable register 2 (IENR2) is set to 1.

If the conversion time or input channel needs to be changed in the A/D mode register (AMR) during A/D conversion, bit ADSF should first be cleared to 0, stopping the conversion operation, in order to avoid malfunction.

12.3.2 Start of A/D Conversion by External Trigger Input

The A/D converter can be made to start A/D conversion by input of an external trigger signal. External trigger input is enabled at pin $\overline{\text{ADTRG}}$ when bit TRGE in AMR is set to 1. Then when the input signal edge designated in bit INTEG5 of interrupt edge select register 2 (IEGR2) is detected at pin $\overline{\text{ADTRG}}$, bit ADSF in ADSR will be set to 1, starting A/D conversion.

Figure 12-2 shows the timing.

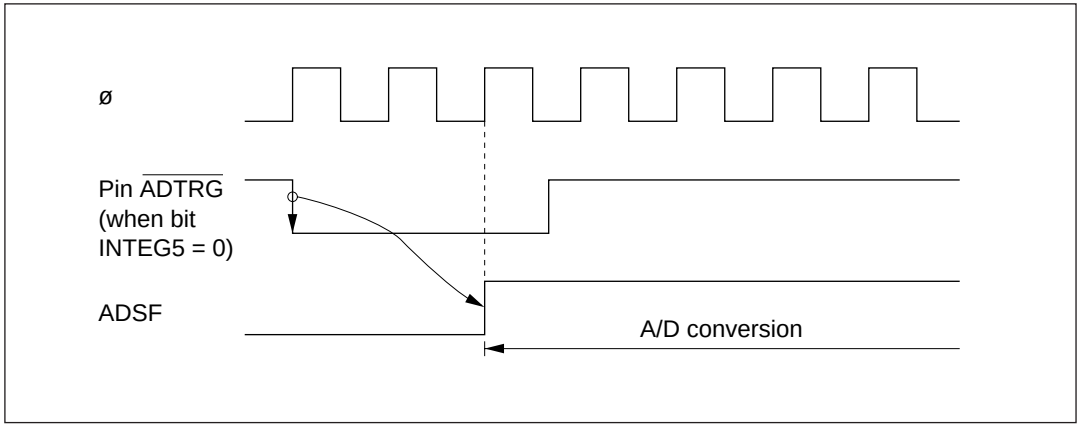


Figure 12-2 External Trigger Input Timing

12.4 Interrupts

When A/D conversion ends (ADSF changes from 1 to 0), bit IRRAD in interrupt request register 2 (IRR2) is set to 1.

A/D conversion end interrupts can be enabled or disabled by means of bit IENAD in interrupt enable register 2 (IENR2).

For further details see 3.3, Interrupts.

12.5 Typical Use

An example of how the A/D converter can be used is given below, using channel 1 (pin AN_1) as the analog input channel. Figure 12-3 shows the operation timing.

1. Bits CH3 to CH0 of the A/D mode register (AMR) are set to 0101, making pin AN1 the analog input channel. A/D interrupts are enabled by setting bit IENAD to 1, and A/D conversion is started by setting bit ADSF to 1.

2. When A/D conversion is complete, bit IRRAD is set to 1, and the A/D conversion result is stored in the A/D result register (ADRR). At the same time ADSF is cleared to 0, and the A/D converter goes to the idle state.

3. Bit IENAD = 1, so an A/D conversion end interrupt is requested.

4. The A/D interrupt handling routine starts.

5. The A/D conversion result is read and processed.

6. The A/D interrupt handling routine ends.

If ADSF is set to 1 again afterward, A/D conversion starts and steps 2 through 6 take place.

Figures 12-4 and 12-5 show flow charts of procedures for using the A/D converter.

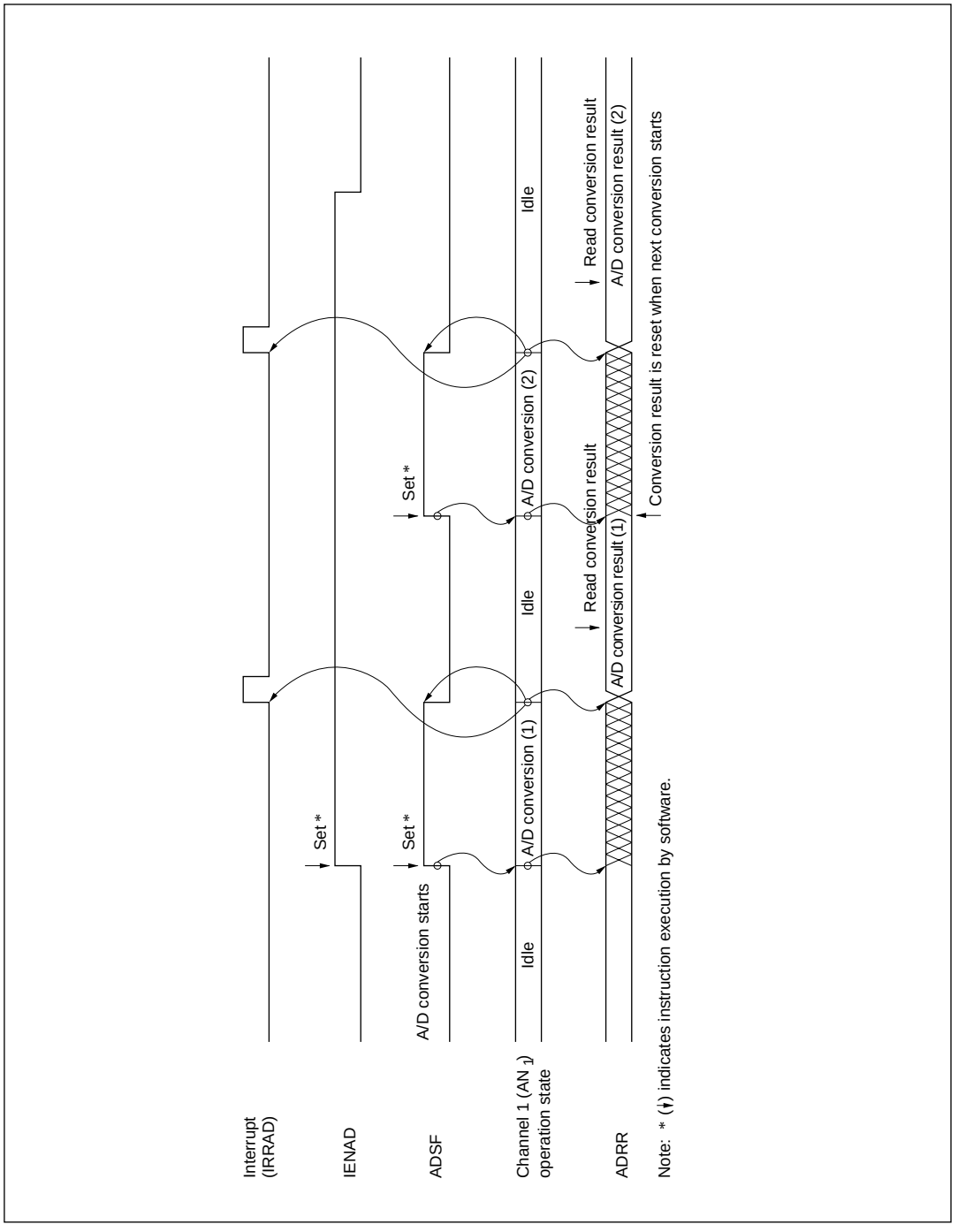


Figure 12-3 Typical A/D Converter Operation Timing

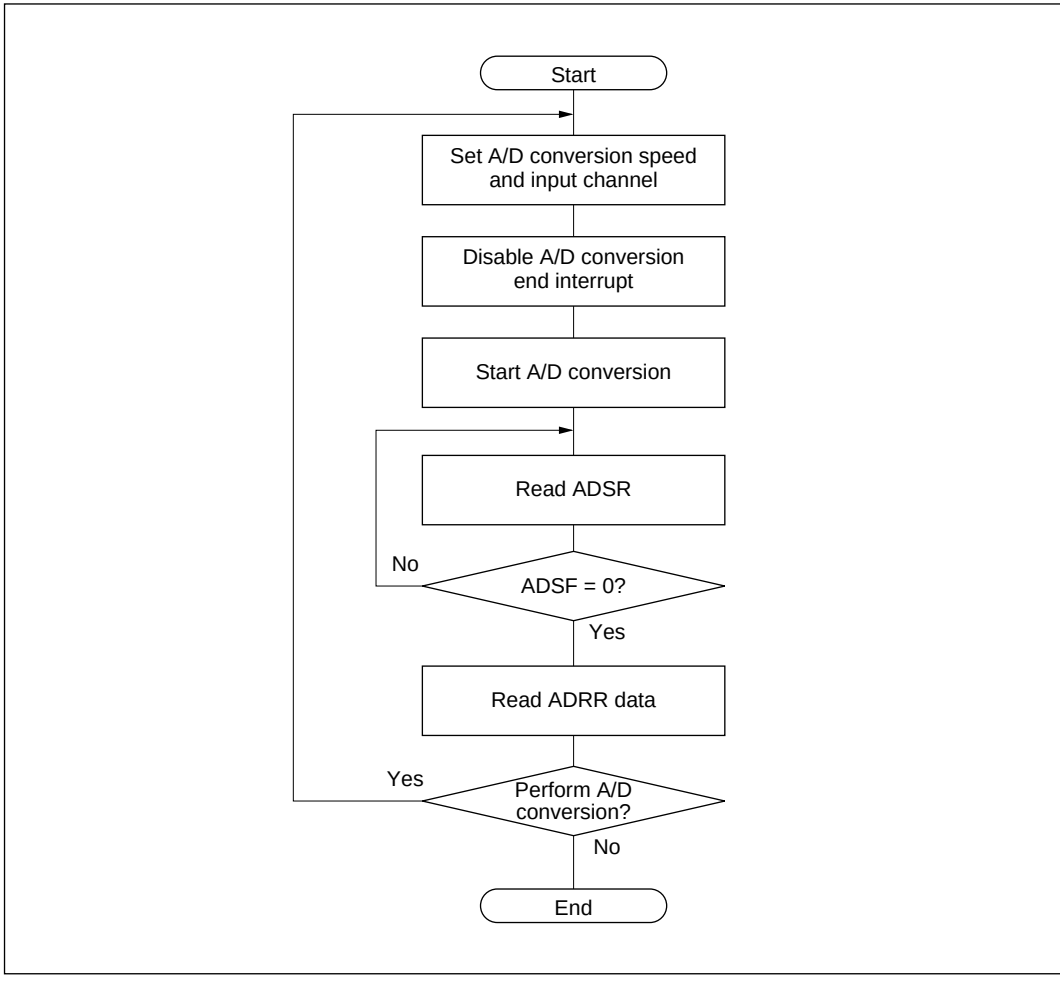


Figure 12-4 Flow Chart of Procedure for Using A/D Converter (1) (Polling by Software)

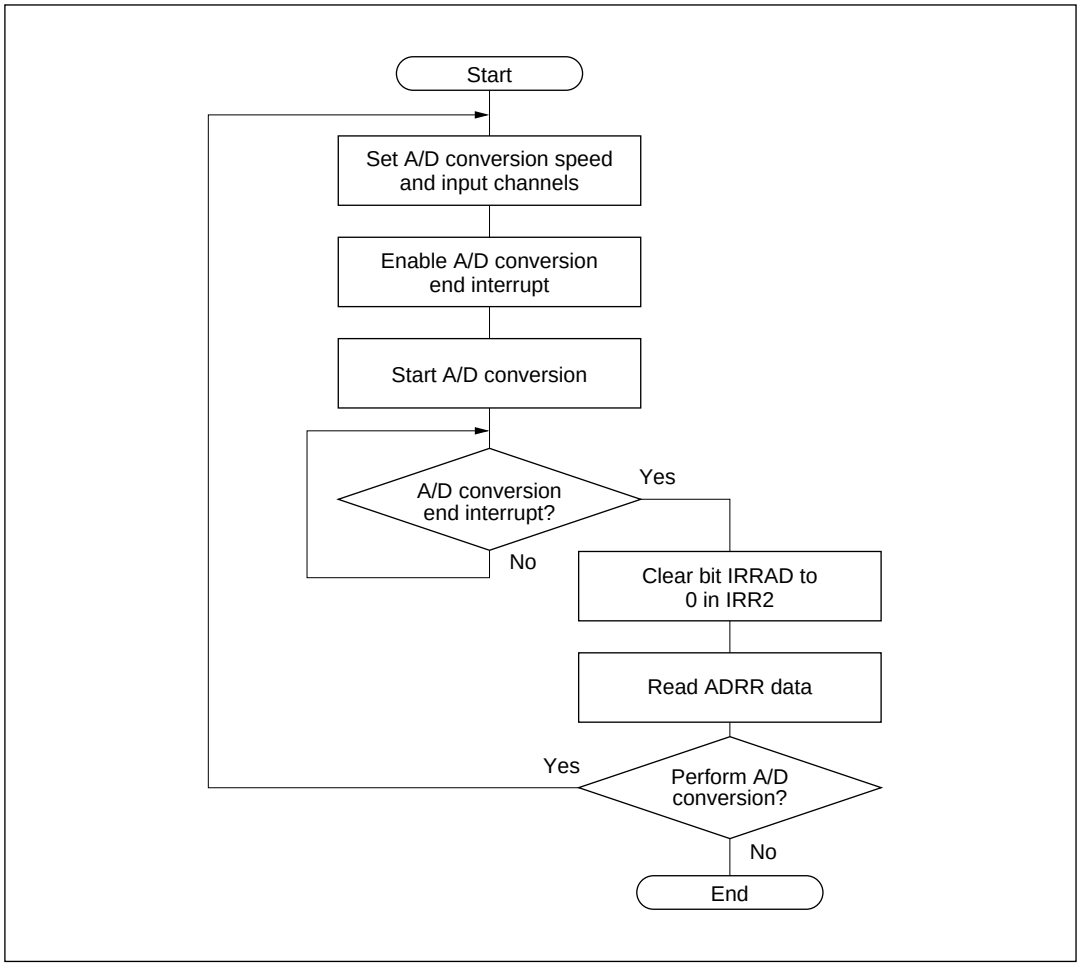


Figure 12-5 Flow Chart of Procedure for Using A/D Converter (2) (Interrupts Used)

12.6 Application Notes

- Data in the A/D result register (ADRR) should be read only when the A/D start flag (ADSF) in the A/D start register (ADSR) is cleared to 0.
- Changing the digital input signal at an adjacent pin during A/D conversion may adversely affect conversion accuracy.

Section 13 Electrical Characteristics

13.1 Absolute Maximum Ratings

Table 13-1 lists the absolute maximum ratings.

Table 13-1 Absolute Maximum Ratings

Item		Symbol	Value	Unit
Power supply voltage		V_{CC}	–0.3 to +7.0	V
Analog power supply voltage		AV_{CC}	–0.3 to +7.0	V
Programming voltage		V_{PP}	–0.3 to +13.0	V
Input voltage	Ports other than Port B	V_{in}	–0.3 to V_{CC} +0.3	V
	Port B		–0.3 to AV_{CC} +0.3	V
Operating temperature		T_{opr}	–20 to +75	°C
Storage temperature		T_{stg}	–55 to +125	°C

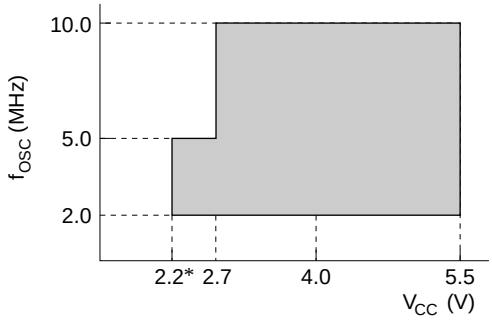
Note: Permanent damage may occur to the chip if maximum ratings are exceeded. Normal operation should be under the conditions specified in Electrical Characteristics. Exceeding these values can result in incorrect operation and reduced reliability.

13.2 Electrical Characteristics

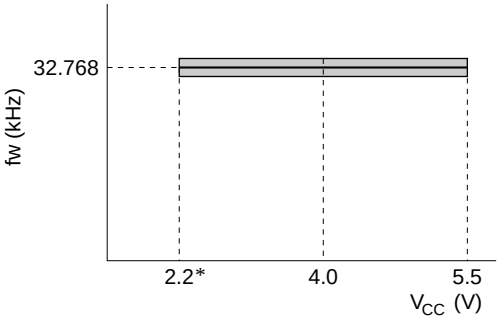
13.2.1 Power Supply Voltage and Operating Range

The power supply voltage and operating range are indicated by the shaded region in the figures below.

1. Power supply voltage vs. oscillator frequency range



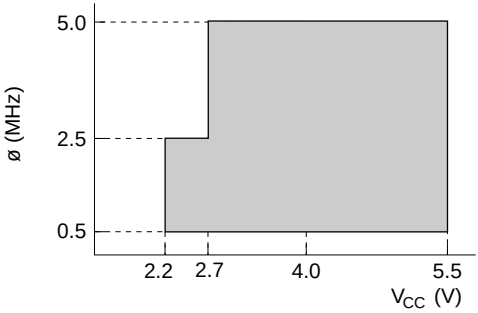
• Active mode (high speed)
• Sleep mode (high speed)



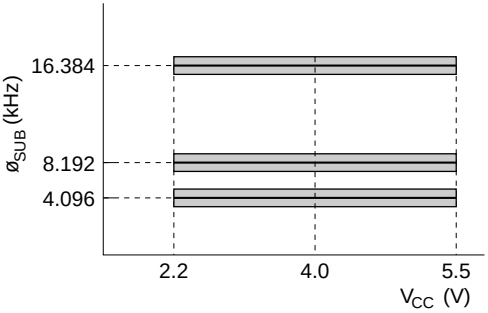
• All operating modes

Note: * The oscillation start voltage is 2.5 V.

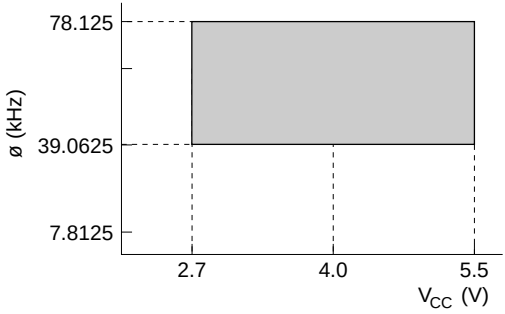
2. Power supply voltage vs. clock frequency range



• Active (high speed) mode
• Sleep (high speed) mode (except CPU)

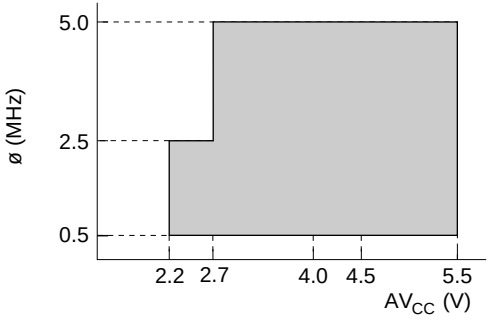


• Subactive mode
• Subsleep mode (except CPU)
• Watch mode (except CPU)



• Active (medium speed) mode
• Sleep (medium speed) mode (except CPU)

3. Analog power supply voltage vs. A/D converter operating range



• Active (high speed) mode
• Sleep (high speed) mode

Don't use in these modes.

• Active (medium speed) mode
• Sleep (medium speed) mode

13.2.2 DC Characteristics (HD6473657)

Table 13-2 lists the DC characteristics of the HD6473657.

Table 13-2 DC Characteristics

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

			Values					
Item	**Symbol**	**Applicable Pins**	**Min**	**Typ**	**Max**	**Unit**	**Test Condition**	**Notes**
Input high voltage	V_{IH}	$\overline{RES}$, $\overline{INT_0}$ to $\overline{INT_7}$, $\overline{IRQ_0}$ to $\overline{IRQ_3}$, $\overline{ADTRG}$, TMIB, TMRIV, TMCIV, FTCI, FTIA, FTIB, FTIC, FTID, SCK_1, SCK_3, $\overline{TRGV}$	0.9 V_{CC}	—	V_{CC} + 0.3	V		
		SI_1, RXD, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_4$	0.8 V_{CC}	—	V_{CC} + 0.3	V		
		PB_0 to PB_7	0.8 V_{CC}	—	AV_{CC} + 0.3			
		OSC_1	V_{CC} – 0.5	—	V_{CC} + 0.3	V	V_{CC} = 4.0 V to 5.5 V	
			V_{CC} – 0.3	—	V_{CC} + 0.3			

Note: Connect the TEST pin to V_{SS}.

Table 13-2 DC Characteristics (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{INT_0}$ to $\overline{INT_7}$, $\overline{IRQ_0}$ to $\overline{IRQ_3}$, $\overline{ADTRG}$, TMIB, TMRIV, TMCIV, FTCI, FTIA, FTIB, FTIC, FTID, SCK_1, SCK_3, $\overline{TRGV}$	–0.3	—	0.1 V_{CC}	V		
		SI_1, RXD, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_4$, PB_0 to PB_7	–0.3	—	0.2 V_{CC}	V		
		OSC_1	–0.3	—	0.5	V	V_{CC} = 4.0 V to 5.5 V	
			–0.3	—	0.3			

Table 13-2 DC Characteristics (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Output high voltage	V_{OH}	$P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_4$	V_{CC} – 1.0	—	—	V	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 1.0 mA	
			V_{CC} – 0.5	—	—		V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 0.5 mA	
			V_{CC} – 0.4	—	—		$-I_{OH}$ = 0.1 mA	
Output low voltage	V_{OL}	$P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_4$	—	—	0.6	V	V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	
			—	—	0.4		I_{OL} = 0.4 mA	
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC_1, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_4$	—	—	1	µA	V_{in} = 0.5 V to (V_{CC} – 0.5 V)	
		PB_0 to PB_7	—	—	1		V_{in} = 0.5 V to AV_{CC} – 0.5 V	
Input leakage current	$\lvert I_{IL} \rvert$	$\overline{RES}$, $\overline{IRQ_0}$	—	—	20	µA	V_{in} = 0.5 V to (V_{CC} – 0.5 V)	
Pull-up MOS current	$-I_p$	$P1_0$ to $P1_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$	50	—	300	µA	V_{CC} = 5 V, V_{in} = 0 V	
			—	25	—		V_{CC} = 2.7 V, V_{in} = 0 V	Reference value

Table 13-2 DC Characteristics (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Input capacitance	C_{in}	All input pins except $\overline{RES}$	—	—	15	pF	f = 1 MHz, V_{in} = 0 V, T_a = 25°C	
		$\overline{RES}$	—	—	60			
		IRQ_0	—	—	30			
Active mode current dissipation	I_{OPE1}	V_{CC}	—	10	15	mA	Active (high-speed) mode V_{CC} = 5 V, f_{OSC} = 10 MHz	1, 2
			—	5	—		V_{CC} = 2.7 V, f_{OSC} = 10 MHz	1, 2
			—	2	—		V_{CC} = 2.2 V, f_{OSC} = 5 MHz	Reference value
	I_{OPE2}	V_{CC}	—	1.5	2.5	mA	Active (medium-speed) mode V_{CC} = 5 V, f_{OSC} = 10 MHz (ø OSC/128)	1, 2
			—	0.8	—		V_{CC} = 2.7 V, f_{OSC} = 10 MHz (ø OSC/128)	1, 2 Reference value
			—	0.4	—		V_{CC} = 2.2 V, f_{OSC} = 5 MHz (ø OSC/128)	
Sleep mode current dissipation	I_{SLEEP1}	V_{CC}	—	4.5	6.5	mA	Sleep (high-speed) mode V_{CC} = 5 V, f_{OSC} = 10 MHz	1, 2
			—	2.2	—		V_{CC} = 2.7 V, f_{OSC} = 10 MHz	1, 2
			—	0.9	—		V_{CC} = 2.2 V, f_{OSC} = 5 MHz	Reference value
	I_{SLEEP2}	V_{CC}	—	1.4	2.4	mA	Sleep (medium-speed) mode V_{CC} = 5 V, f_{OSC} = 10 MHz (ø OSC/128)	1, 2
			—	0.7	—		V_{CC} = 2.7 V, f_{OSC} = 10 MHz (ø OSC/128)	1, 2 Reference value
			—	0.4	—		V_{CC} = 2.2 V, f_{OSC} = 5 MHz (ø OSC/128)	
Subactive mode current dissipation	I_{SUB}	V_{CC}	—	20	30	µA	V_{CC} = 2.7 V 32-kHz crystal oscillator ($ø_{SUB}$ = $ø_W$/2)	1, 2
			—	15	—		V_{CC} = 2.2 V 32-kHz crystal oscillator ($ø_{SUB}$ = $ø_W$/2)	1, 2 Reference value
			—	9	—		V_{CC} = 2.2 V 32-kHz crystal oscillator ($ø_{SUB}$ = $ø_W$/8)	

Table 13-2 DC Characteristics (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Subsleep mode current dissipation	I_{SUBSP}	V_{CC}	—	10	20	µA	V_{CC} = 2.7 V 32-kHz crystal oscillator ($ø_{SUB}$ = $ø_W$/2)	1, 2
			—	7	—		V_{CC} = 2.2 V 32-kHz crystal oscillator ($ø_{SUB}$ = $ø_W$/2)	1, 2 Reference value
Watch mode current dissipation	I_{WATCH}	V_{CC}	—	7.5	9	µA	V_{CC} = 2.7 V 32-kHz crystal oscillator	1, 2
			—	6	—		V_{CC} = 2.2 V 32-kHz crystal oscillator	1, 2 Reference value
Standby mode current dissipation	I_{STBY}	V_{CC}	—	—	5	µA	32-kHz crystal oscillator not used	1, 2
RAM data retaining voltage	V_{RAM}	V_{CC}	2	—	—	V		

Notes: 1. Pin states during current measurement are given below.
2. Excludes current in pull-up MOS transistors and output buffers.

Mode	$\overline{RES}$ Pin	Internal State	Other Pins	Oscillator Pins
Active (high-speed) mode	V_{CC}	Operates	V_{CC}	System clock oscillator: ceramic or crystal
Active (medium-speed) mode		Operates (øOSC/128)		Subclock oscillator: Pin X_1 = V_{CC}
Sleep (high-speed) mode	V_{CC}	Only timers operate	V_{CC}	
Sleep (medium-speed) mode		Only timers operate (øOSC/128)		
Subactive mode	V_{CC}	Operates	V_{CC}	System clock oscillator: ceramic or crystal
Subsleep mode	V_{CC}	Only timers operate, CPU stops	V_{CC}	Subclock oscillator: crystal
Watch mode	V_{CC}	Only time base operates, CPU stops	V_{CC}	
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	System clock oscillator: ceramic or crystal Subclock oscillator: Pin X_1 = V_{CC}

Table 13-2 DC Characteristics (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise indicated.

Item		Symbol	Values			Unit	Test Condition
			Min	Typ	Max		
Allowable output low current (per pin)	All output pins	I_{OL}	—	—	2	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	0.5		
Allowable output low current (total)	All output pins	ΣI_{OL}	—	—	40	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	20		
Allowable output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	0.2		
Allowable output high current (total)	All output pins	$\Sigma -I_{OH}$	—	—	15	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	10		

13.2.3 AC Characteristics (HD6473657)

Table 13-3 lists the control signal timing, and tables 13-4 and 13-5 list the serial interface timing of the HD6473657.

Table 13-3 Control Signal Timing

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
System clock oscillation frequency	f_{OSC}	OSC_1, OSC_2	2	—	10	MHz	V_{CC} = 2.7 V to 5.5 V	
			2	—	5			
OSC clock ($ø_{OSC}$) cycle time	t_{OSC}	OSC_1, OSC_2	100	—	1000	ns	V_{CC} = 2.7 V to 5.5 V	1
			200	—	1000			Figure 13-1
System clock (ø) cycle time	t_{cyc}		2	—	128	t_{OSC}		1
			—	—	25.5	µs		
Subclock oscillation frequency	f_W	X_1, X_2	—	32.768	—	kHz		
Watch clock ($ø_W$) cycle time	t_W	X_1, X_2	—	30.5	—	µs		
Subclock ($ø_{SUB}$) cycle time	t_{subcyc}		2	—	8	t_W		2
Instruction cycle time			2	—	—	t_{cyc} t_{subcyc}		
Oscillation stabilization time (crystal oscillator)	t_{rc}	OSC_1, OSC_2	—	—	40	ms	V_{CC} = 4.0 V to 5.5 V	
			—	—	60		V_{CC} = 2.5 V to 5.5 V	
Oscillation stabilization time (ceramic oscillator)	t_{rc}	OSC_1, OSC_2	—	—	20	ms	V_{CC} = 4.0 V to 5.5 V	
			—	—	40		V_{CC} = 2.5 V to 5.5 V	
Oscillation stabilization time	t_{rc}	X_1, X_2	—	—	2	s	V_{CC} = 2.5 V to 5.5 V	
External clock high width	t_{CPH}	OSC_1	40	—	—	ns	V_{CC} = 2.7 V to 5.5 V	Figure 13-1
			80	—	—			
External clock low width	t_{CPL}	OSC_1	40	—	—	ns	V_{CC} = 2.7 V to 5.5 V	Figure 13-1
			80	—	—			
External clock rise time	t_{CPr}		—	—	15	ns	V_{CC} = 2.7 V to 5.5 V	
			—	—	20			
External clock fall time	t_{CPf}		—	—	15	ns	V_{CC} = 2.7 V to 5.5 V	
			—	—	20			
Pin $\overline{RES}$ low width	t_{REL}	$\overline{RES}$	18	—	—	t_{cyc} t_{subcyc}		Figure 13-2

Notes: 1. A frequency between 1 MHz to 10 MHz is required when an external clock is input.
2. Selected with SA1 and SA0 of system clock control register 2 (SYSCR2).

Table 13-3 Control Signal Timing (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
Oscillation start voltage	V_{start}	OSC_1, OSC_2	2.5	—	—	V		
		X_1, X_2	2.5	—	—			
Input pin high width	t_{IH}	$\overline{IRQ_0}$ to $\overline{IRQ_3}$, $\overline{INT_0}$ to $\overline{INT_7}$, $\overline{ADTRG}$, TMIB, TMCIV, TMRIV, FTCI, FTIA, FTIB, FTIC, FTID, TRGV	2	—	—	t_{cyc} t_{subcyc}		Figure 13-3
Input pin low width	t_{IL}	$\overline{IRQ_0}$ to $\overline{IRQ_3}$, $\overline{INT_0}$ to $\overline{INT_7}$, $\overline{ADTRG}$, TMIB, TMCIV, TMRIV, FTCI, FTIA, FTIB, FTIC, FTID, TRGV	2	—	—	t_{cyc} t_{subcyc}		Figure 13-3

Table 13-4 Serial Interface (SCI1) Timing

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
Input serial clock cycle time	t_{scyc}	SCK_1	4	—	—	t_{cyc}		Figure 13-4
Input serial clock high width	t_{SCKH}	SCK_1	0.4	—	—	t_{scyc}		
Input serial clock low width	t_{SCKL}	SCK_1	0.4	—	—	t_{scyc}		
Input serial clock rise time	t_{SCKr}	SCK_1	—	—	60	ns	V_{CC} = 4.0 V to 5.5 V	
			—	—	80			
Input serial clock fall time	t_{SCKf}	SCK_1	—	—	60	ns	V_{CC} = 4.0 V to 5.5 V	
			—	—	80			
Serial output data delay time	t_{SOD}	SO_1	—	—	200	ns	V_{CC} = 4.0 V to 5.5 V	
			—	—	350			
Serial input data setup time	t_{SIS}	SI_1	200	—	—	ns	V_{CC} = 4.0 V to 5.5 V	
			400	—	—			
Serial input data hold time	t_{SIH}	SI_1	200	—	—	ns	V_{CC} = 4.0 V to 5.5 V	
			400	—	—			

Table 13-5 Serial Interface (SCI3) Timing

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item		Symbol	Values Min	Typ	Max	Unit	Test Conditions	Reference Figure
Input clock cycle	Asynchronous	t_{scyc}	4	—	—	t_{cyc}		Figure 13-5
	Synchronous		6	—	—			
Input clock pulse width		t_{SCKW}	0.4	—	0.6	t_{scyc}		Figure 13-5
Transmit data delay time (synchronous)		t_{TXD}	—	—	1	t_{cyc}	V_{CC} = 4.0 V to 5.5 V	Figure 13-6
			—	—	1			
Receive data setup time (synchronous)		t_{RXS}	200	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 13-6
			400	—	—			
Receive data hold time (synchronous)		t_{RXH}	200	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 13-6
			400	—	—			

13.2.4 DC Characteristics (HD6433657, HD6433656, HD6433655, HD6433654, HD6433653, HD6433652)

Table 13-6 lists the DC characteristics of the HD6433657, the HD6433656, the HD6433655, the HD6433654, the HD6433653, and the HD6433652.

Table 13-6 DC Characteristics

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

			Values					
Item	**Symbol**	**Applicable Pins**	**Min**	**Typ**	**Max**	**Unit**	**Test Condition**	**Notes**
Input high voltage	V_{IH}	$\overline{RES}$, $\overline{INT_0}$ to $\overline{INT_7}$, $\overline{IRQ_0}$ to $\overline{IRQ_3}$, $\overline{ADTRG}$, TMIB, TMRIV, TMCIV, FTCI, FTIA, FTIB, FTIC, FTID, SCK_1, SCK_3, $\overline{TRGV}$	0.9 V_{CC}	—	V_{CC} + 0.3	V		
		SI_1, RXD, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_4$	0.8 V_{CC}	—	V_{CC} + 0.3	V		
		PB_0 to PB_7	0.8 V_{CC}	—	AV_{CC} + 0.3			
		OSC_1	V_{CC} – 0.5	—	V_{CC} + 0.3	V	V_{CC} = 4.0 V to 5.5 V	
			V_{CC} – 0.3	—	V_{CC} + 0.3			

Note: Connect the TEST pin to V_{SS}.

Table 13-6 DC Characteristics (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Input low voltage	V_{IL}	$\overline{RES}$, $\overline{INT_0}$ to $\overline{INT_7}$, $\overline{IRQ_0}$ to $\overline{IRQ_3}$, $\overline{ADTRG}$, TMIB, TMRIV, TMCIV, FTCI, FTIA, FTIB, FTIC, FTID, SCK_1, SCK_3, TRGV	–0.3	—	0.1 V_{CC}	V		
		SI_1, RXD, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_4$, PB_0 to PB_7	–0.3	—	0.2 V_{CC}	V		
		OSC_1	–0.3	—	0.5	V	V_{CC} = 4.0 V to 5.5 V	
			–0.3	—	0.3			

Note: Connect the TEST pin to V_{SS}.

Table 13-6 DC Characteristics (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Output high voltage	V_{OH}	$P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_4$	V_{CC} – 1.0	—	—	V	V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 1.0 mA	
			V_{CC} – 0.5	—	—		V_{CC} = 4.0 V to 5.5 V $-I_{OH}$ = 0.5 mA	
			V_{CC} – 0.4	—	—		$-I_{OH}$ = 0.1 mA	
Output low voltage	V_{OL}	$P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_4$	—	—	0.6	V	V_{CC} = 4.0 V to 5.5 V I_{OL} = 1.6 mA	
			—	—	0.4		I_{OL} = 0.4 mA	
Input/ output leakage current	$\lvert I_{IL} \rvert$	OSC_1, $P1_0$ to $P1_7$, $P2_0$ to $P2_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$, $P6_0$ to $P6_7$, $P7_0$ to $P7_7$, $P8_0$ to $P8_7$, $P9_0$ to $P9_4$	—	—	1	µA	V_{in} = 0.5 V to (V_{CC} – 0.5 V)	
		PB_0 to PB_7	—	—	1		V_{in} = 0.5 V to AV_{CC} – 0.5 V	
Input leakage current	$\lvert I_{IL} \rvert$	$\overline{RES}$, $\overline{IRQ_0}$	—	—	1	µA	V_{in} = 0.5 V to (V_{CC} – 0.5 V)	
Pull-up MOS current	$-I_p$	$P1_0$ to $P1_7$, $P3_0$ to $P3_5$, $P5_0$ to $P5_7$	50	—	300	µA	V_{CC} = 5 V, V_{in} = 0 V	
			—	25	—		V_{CC} = 2.7 V, V_{in} = 0 V	Reference value

Table 13-6 DC Characteristics (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Notes
Input capacitance	C_{in}	All input pins except $\overline{RES}$	—	—	15	pF	f = 1 MHz, V_{in} = 0 V, T_a = 25°C	
		$\overline{RES}$	—	—	15			
		IRQ_0	—	—	15			
Active mode current dissipation	I_{OPE1}	V_{CC}	—	10	15	mA	Active (high-speed) mode V_{CC} = 5 V, f_{OSC} = 10 MHz	1, 2
			—	5	—		V_{CC} = 2.7 V, f_{OSC} = 10 MHz	1, 2
			—	2	—		V_{CC} = 2.2 V, f_{OSC} = 5 MHz	Reference value
	I_{OPE2}	V_{CC}	—	1.5	2.5	mA	Active (medium-speed) mode V_{CC} = 5 V, f_{OSC} = 10 MHz (ø OSC/128)	1, 2
			—	0.8	—		V_{CC} = 2.7 V, f_{OSC} = 10 MHz (ø OSC/128)	1, 2 Reference value
			—	0.4	—		V_{CC} = 2.2 V, f_{OSC} = 5 MHz (ø OSC/128)	
Sleep mode current dissipation	I_{SLEEP1}	V_{CC}	—	4.5	6.5	mA	Sleep (high-speed) mode V_{CC} = 5 V, f_{OSC} = 10 MHz	1, 2
			—	2.2	—		V_{CC} = 2.7 V, f_{OSC} = 10 MHz	1, 2
			—	0.9	—		V_{CC} = 2.2 V, f_{OSC} = 5 MHz	Reference value
	I_{SLEEP2}	V_{CC}	—	1.4	2.4	mA	Sleep (medium-speed) mode V_{CC} = 5 V, f_{OSC} = 10 MHz (ø OSC/128)	1, 2
			—	0.7	—		V_{CC} = 2.7 V, f_{OSC} = 10 MHz (ø OSC/128)	1, 2 Reference value
			—	0.4	—		V_{CC} = 2.2 V, f_{OSC} = 5 MHz (ø OSC/128)	
Subactive mode current dissipation	I_{SUB}	V_{CC}	—	20	30	µA	V_{CC} = 2.7 V 32-kHz crystal oscillator ($ø_{SUB}$ = $ø_W$/2)	1, 2
			—	15	—		V_{CC} = 2.2 V 32-kHz crystal oscillator ($ø_{SUB}$ = $ø_W$/2)	1, 2 Reference value
			—	9	—		V_{CC} = 2.2 V 32-kHz crystal oscillator ($ø_{SUB}$ = $ø_W$/8)	

Table 13-6 DC Characteristics (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C unless otherwise indicated.

			Values					
Item	**Symbol**	**Applicable Pins**	**Min**	**Typ**	**Max**	**Unit**	**Test Condition**	**Notes**
Subsleep mode current dissipation	I_{SUBSP}	V_{CC}	—	10	20	µA	V_{CC} = 2.7 V 32-kHz crystal oscillator ($ø_{SUB}$ = $ø_W$/2)	1, 2
			—	7	—		V_{CC} = 2.2 V 32-kHz crystal oscillator ($ø_{SUB}$ = $ø_W$/2)	1, 2 Reference value
Watch mode current dissipation	I_{WATCH}	V_{CC}	—	7.5	9	µA	V_{CC} = 2.7 V 32-kHz crystal oscillator	1, 2
			—	6	—		V_{CC} = 2.2 V 32-kHz crystal oscillator	1, 2 Reference value
Standby mode current dissipation	I_{STBY}	V_{CC}	—	—	5	µA	32-kHz crystal oscillator not used	1, 2
RAM data retaining voltage	V_{RAM}	V_{CC}	2	—	—	V		

Notes: 1. Pin states during current measurement are given below.
2. Excludes current in pull-up MOS transistors and output buffers.

Mode	**$\overline{RES}$ Pin**	**Internal State**	**Other Pins**	**Oscillator Pins**
Active (high-speed) mode	V_{CC}	Operates	V_{CC}	System clock oscillator: ceramic or crystal
Active (medium-speed) mode		Operates (øOSC/128)		Subclock oscillator: Pin X_1 = V_{CC}
Sleep (high-speed) mode	V_{CC}	Only timers operate	V_{CC}	
Sleep (medium-speed) mode		Only timers operate (øOSC/128)		
Subactive mode	V_{CC}	Operates	V_{CC}	System clock oscillator: ceramic or crystal Subclock oscillator: crystal
Subsleep mode	V_{CC}	Only timers operate, CPU stops	V_{CC}	
Watch mode	V_{CC}	Only time base perates, CPU stops	V_{CC}	
Standby mode	V_{CC}	CPU and timers both stop	V_{CC}	System clock oscillator: ceramic or crystal Subclock oscillator: Pin X_1 = V_{CC}

Table 13-6 DC Characteristics (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise indicated.

Item		Symbol	Values			Unit	Test Condition
			Min	Typ	Max		
Allowable output low current (per pin)	All output pins	I_{OL}	—	—	2	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	0.5		
Allowable output low current (total)	All output pins	ΣI_{OL}	—	—	40	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	20		
Allowable output high current (per pin)	All output pins	$-I_{OH}$	—	—	2	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	0.2		
Allowable output high current (total)	All output pins	$\Sigma -I_{OH}$	—	—	15	mA	V_{CC} = 4.0 V to 5.5 V
			—	—	10		

13.2.5 AC Characteristics (HD6433657, HD6433656, HD6433655, HD6433654, HD6433653, HD6433652)

Table 13-7 lists the control signal timing, and tables 13-8 and 13-9 list the serial interface timing of the HD6433657, the HD6433656, the HD6433655, the HD6433654, the HD6433653, and the HD6433652.

Table 13-7 Control Signal Timing

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
System clock oscillation frequency	f_{OSC}	OSC_1, OSC_2	2	—	10	MHz	V_{CC} = 2.7 V to 5.5 V	
			2	—	5			
OSC clock ($ø_{OSC}$) cycle time	t_{OSC}	OSC_1, OSC_2	100	—	1000	ns	V_{CC} = 2.7 V to 5.5 V	1
			200	—	1000			Figure 13-1
System clock (ø) cycle time	t_{cyc}		2	—	128	t_{OSC}		1
			—	—	25.5	µs		
Subclock oscillation frequency	f_W	X_1, X_2	—	32.768	—	kHz		
Watch clock ($ø_W$) cycle time	t_W	X_1, X_2	—	30.5	—	µs		
Subclock ($ø_{SUB}$) cycle time	t_{subcyc}		2	—	8	t_W		2
Instruction cycle time			2	—	—	t_{cyc} t_{subcyc}		
Oscillation stabilization time (crystal oscillator)	t_{rc}	OSC_1, OSC_2	—	—	40	ms	V_{CC} = 4.0 V to 5.5 V	
			—	—	60		V_{CC} = 2.5 V to 5.5 V	
Oscillation stabilization time (ceramic oscillator)	t_{rc}	OSC_1, OSC_2	—	—	20	ms	V_{CC} = 4.0 V to 5.5 V	
			—	—	40		V_{CC} = 2.5 V to 5.5 V	
Oscillation stabilization time	t_{rc}	X_1, X_2	—	—	2	s	V_{CC} = 2.5 V to 5.5 V	
External clock high width	t_{CPH}	OSC_1	40	—	—	ns	V_{CC} = 2.7 V to 5.5 V	Figure 13-1
			80	—	—			
External clock low width	t_{CPL}	OSC_1	40	—	—	ns	V_{CC} = 2.7 V to 5.5 V	Figure 13-1
			80	—	—			
External clock rise time	t_{CPr}		—	—	15	ns	V_{CC} = 2.7 V to 5.5 V	
			—	—	20			
External clock fall time	t_{CPf}		—	—	15	ns	V_{CC} = 2.7 V to 5.5 V	
			—	—	20			
Pin $\overline{RES}$ low width	t_{REL}	$\overline{RES}$	18	—	—	t_{cyc} t_{subcyc}		Figure 13-2

Notes: 1. A frequency between 1 MHz to 10 MHz is required when an external clock is input.
2. Selected with SA1 and SA0 of system clock control register 2 (SYSCR2).

Table 13-7 Control Signal Timing (cont)

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
Oscillation start voltage	V_{start}	OSC_1, OSC_2	2.5	—	—	V		
		X_1, X_2	2.5	—	—			
Input pin high width	t_{IH}	$\overline{IRQ_0}$ to $\overline{IRQ_3}$, INT_0 to INT_7, $\overline{ADTRG}$, TMIB, TMCIV, TMRIV, FTCI, FTIA, FTIB, FTIC, FTID, TRGV	2	—	—	t_{cyc} t_{subcyc}		Figure 13-3
Input pin low width	t_{IL}	$\overline{IRQ_0}$ to $\overline{IRQ_3}$, INT_0 to INT_7, $\overline{ADTRG}$, TMIB, TMCIV, TMRIV, FTCI, FTIA, FTIB, FTIC, FTID, TRGV	2	—	—	t_{cyc} t_{subcyc}		Figure 13-3

Table 13-8 Serial Interface (SCI1) Timing

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item	Symbol	Applicable Pins	Values Min	Typ	Max	Unit	Test Condition	Reference Figure
Input serial clock cycle time	t_{scyc}	SCK_1	4	—	—	t_{cyc}		Figure 13-4
Input serial clock high width	t_{SCKH}	SCK_1	0.4	—	—	t_{scyc}		
Input serial clock low width	t_{SCKL}	SCK_1	0.4	—	—	t_{scyc}		
Input serial clock rise time	t_{SCKr}	SCK_1	—	—	60	ns	V_{CC} = 4.0 V to 5.5 V	
			—	—	80			
Input serial clock fall time	t_{SCKf}	SCK_1	—	—	60	ns	V_{CC} = 4.0 V to 5.5 V	
			—	—	80			
Serial output data delay time	t_{SOD}	SO_1	—	—	200	ns	V_{CC} = 4.0 V to 5.5 V	
			—	—	350			
Serial input data setup time	t_{SIS}	SI_1	200	—	—	ns	V_{CC} = 4.0 V to 5.5 V	
			400	—	—			
Serial input data hold time	t_{SIH}	SI_1	200	—	—	ns	V_{CC} = 4.0 V to 5.5 V	
			400	—	—			

Table 13-9 Serial Interface (SCI3) Timing

V_{CC} = AV_{CC} = 2.2 V to 5.5 V, V_{SS} = AV_{SS} = 0.0 V, T_a = –20°C to +75°C, unless otherwise specified.

Item		Symbol	Values Min	Typ	Max	Unit	Test Conditions	Reference Figure
Input clock	Asynchronous	t_{scyc}	4	—	—	t_{cyc}		Figure 13-5
cycle	Synchronous		6	—	—			
Input clock pulse width		t_{SCKW}	0.4	—	0.6	t_{scyc}		Figure 13-5
Transmit data delay time (synchronous)		t_{TXD}	—	—	1	t_{cyc}	V_{CC} = 4.0 V to 5.5 V	Figure 13-6
			—	—	1			
Receive data setup time (synchronous)		t_{RXS}	200	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 13-6
			400	—	—			
Receive data hold time (synchronous)		t_{RXH}	200	—	—	ns	V_{CC} = 4.0 V to 5.5 V	Figure 13-6
			400	—	—			

13.2.6 A/D Converter Characteristics

Table 13-10 shows the A/D converter characteristics of the HD6473657, the HD6433657, the HD6433656, the HD6433655, the HD6433654, the HD6433653, and the HD6433652.

Table 13-10 A/D Converter Characteristics

$V_{CC} = AV_{CC} = 2.2$ V to 5.5 V, $V_{SS} = AV_{SS} = 0.0$ V, $T_a = -20°C$ to $+75°C$, unless otherwise specified.

Item	Symbol	Applicable Pins	Values Min	Values Typ	Values Max	Unit	Test Condition	Reference Figure
Analog power supply voltage	AV_{CC}	AV_{CC}	2.2	—	5.5	V		1
Analog input voltage	AV_{IN}	AN_0 to AN_7	AV_{SS} – 0.3	—	AV_{CC} + 0.3	V		
Analog power supply current	AI_{OPE}	AV_{CC}	—	—	1.5	mA	AV_{CC} = 5 V	
	AI_{STOP1}	AV_{CC}	—	150	—	µA		2 Reference value
	AI_{STOP2}	AV_{CC}	—	—	5	µA		3
Analog input capacitance	C_{AIN}	AN_0 to AN_7	—	—	30	pF		
Allowable signal source impedance	R_{AIN}		—	—	5	kΩ		
Resolution (data length)			—	—	8	bit		
Nonlinearity error			—	—	±2.0	LSB		
Quantization error			—	—	±0.5	LSB		
Absolute accuracy			—	—	±2.5	LSB		
Conversion time			12.4	—	124	µs	AV_{CC} = 2.7 to 5.5 V	
			24.8	—	124			

Notes: 1. Always set $AV_{CC} = V_{CC}$ for the analog power supply voltage.
2. AI_{STOP1} is the current in active and sleep modes while the A/D converter is idle.
3. AI_{STOP2} is the current at reset and in standby, watch, subactive, and subsleep modes while the A/D converter is idle.

13.3 Operation Timing

Figures 13-1 to 13-9 show timing diagrams.

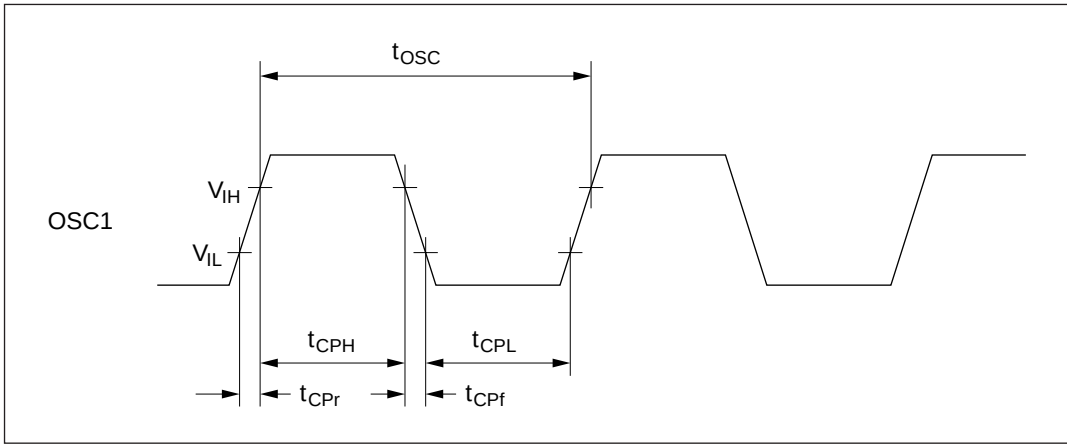


Figure 13-1 System Clock Input Timing

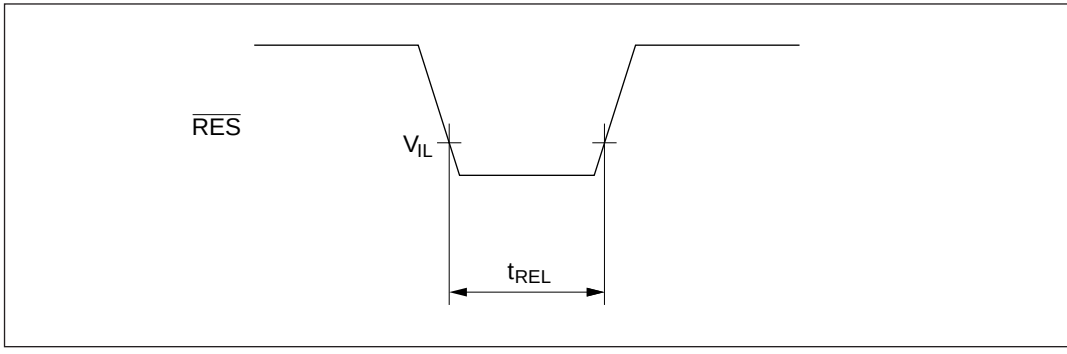


Figure 13-2 $\overline{RES}$ Low Width Timing

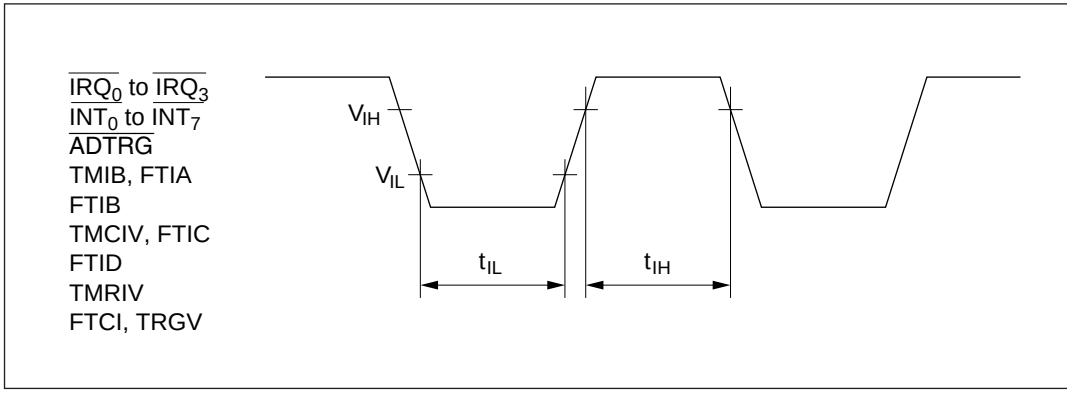
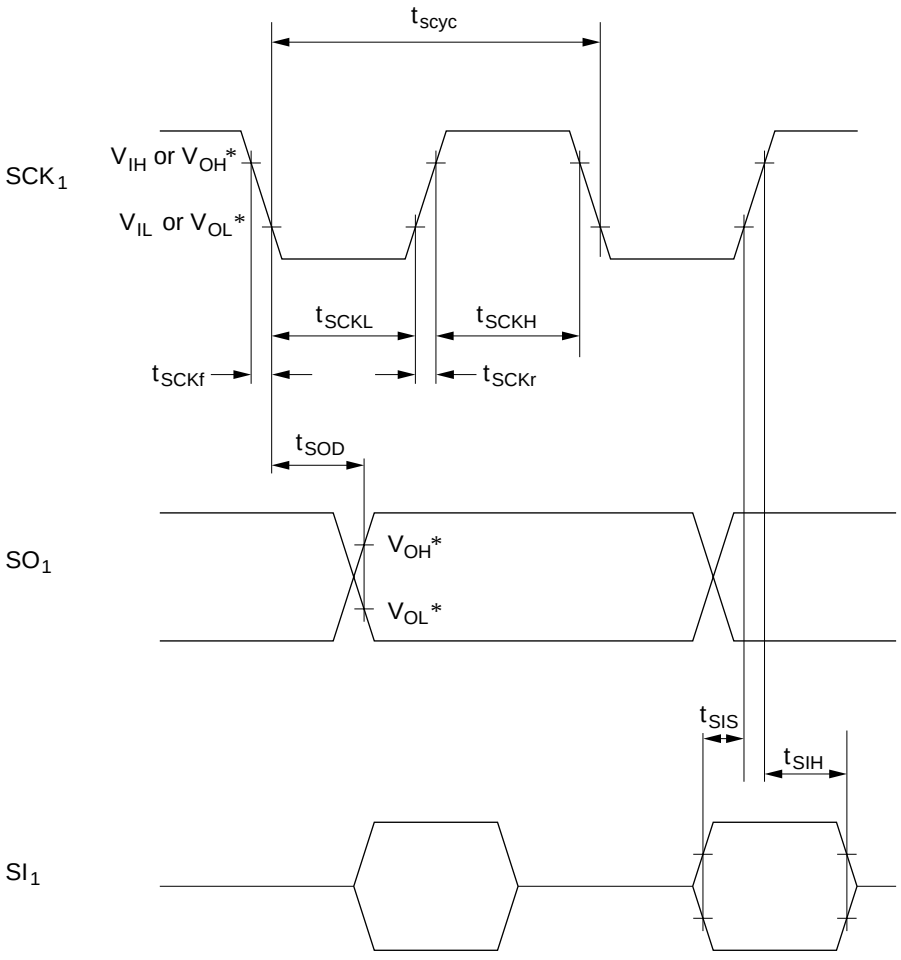


Figure 13-3 Input Timing



Notes: * Output timing reference levels
Output high: V_{OH} = 2.0 V
Output low: V_{OL} = 0.8 V
Load conditions are shown in figure 13-7.

Figure 13-4 SCI1, SCI2 Input/Output Timing

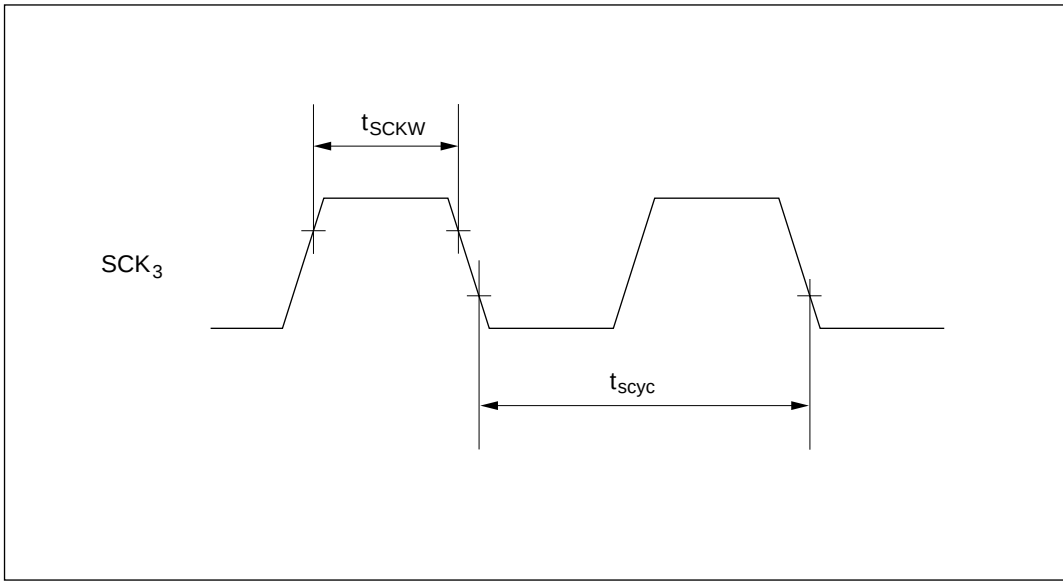
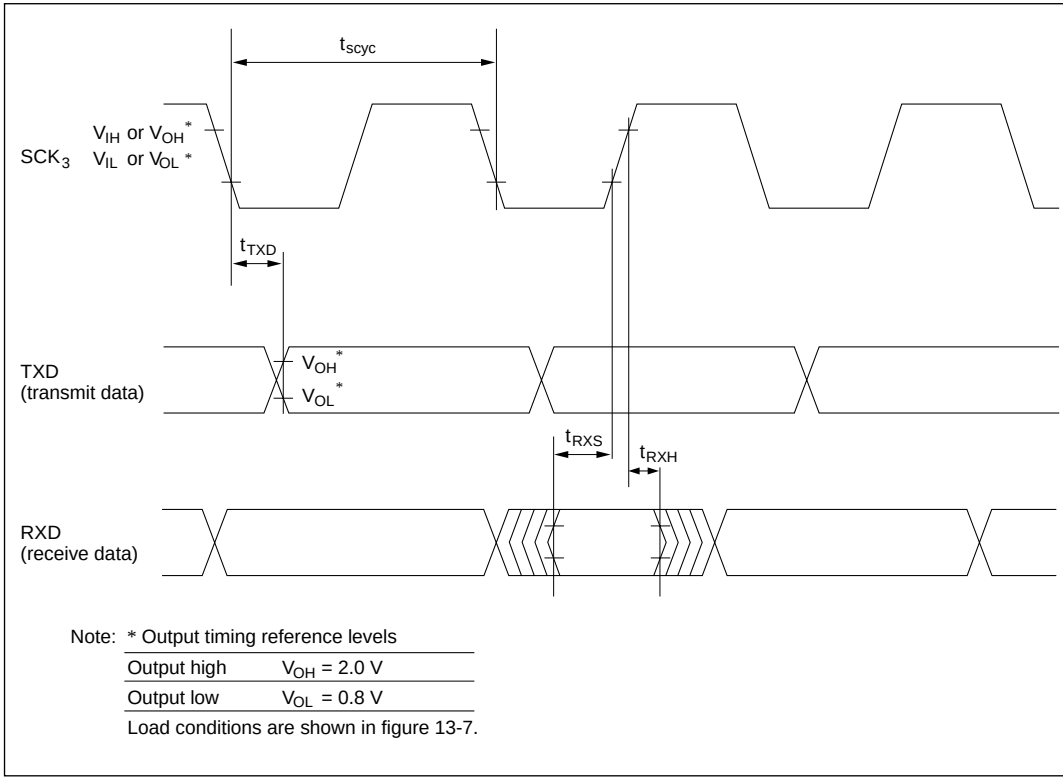


Figure 13-5 SCK_3 Input Clock Timing



Note: * Output timing reference levels

Output high	V_{OH} = 2.0 V
Output low	V_{OL} = 0.8 V

Load conditions are shown in figure 13-7.

Figure 13-6 Serial Interface 3 Synchronous Mode Input Output Timing

13.4 Output Load Circuit

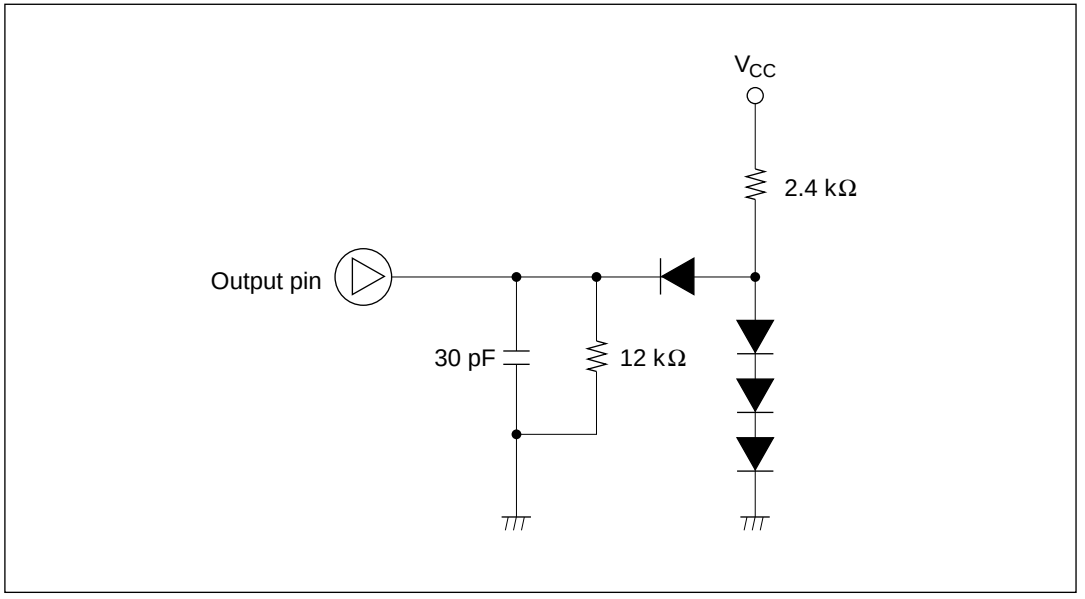


Figure 13-7 Output Load Condition

Appendix A CPU Instruction Set

A.1 Instructions

Operation Notation

Rd8/16	General register (destination) (8 or 16 bits)
Rs8/16	General register (source) (8 or 16 bits)
Rn8/16	General register (8 or 16 bits)
CCR	Condition code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#xx: 3/8/16	Immediate data (3, 8, or 16 bits)
d: 8/16	Displacement (8 or 16 bits)
@aa: 8/16	Absolute address (8 or 16 bits)
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Exclusive logical OR
→	Move
¬	Logical complement

Condition Code Notation

Symbol	
↕	Modified according to the instruction result
*	Not fixed (value not guaranteed)
0	Always cleared to 0
—	Not affected by the instruction execution result

Table A-1 lists the H8/300L CPU instruction set.

Table A-1 Instruction Set

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)									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 → Rd	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 (bytes)									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	
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–Rd → Rd		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

Table A-1 Instruction Set (cont)

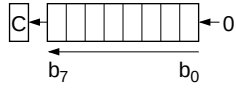
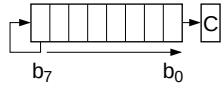
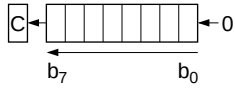
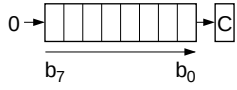
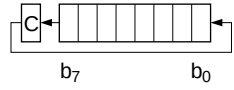
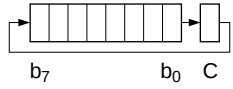
Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
MULXU.B Rs, Rd	B	Rd8 × Rs8 → Rd16		2								—	—	—	—	—	—	14
DIVXU.B Rs, Rd	B	Rd16÷Rs8 → Rd16 (RdH: remainder, RdL: quotient)		2								—	—	(5)	(6)	—	—	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
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	↕	0	↕	2
ROTXL.B Rd	B			2								—	—	↕	↕	0	↕	2
ROTXR.B Rd	B			2								—	—	↕	↕	0	↕	2

Table A-1 Instruction Set (cont)

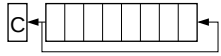
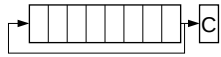
Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
ROTL.B Rd	B			2								—	—	↕	↕	0	↕	2
ROTR.B Rd	B			2								—	—	↕	↕	0	↕	2
BSET #xx:3, Rd	B	(#xx:3 of Rd8) ← 1		2								—	—	—	—	—	—	2
BSET #xx:3, @Rd	B	(#xx:3 of @Rd16) ← 1			4							—	—	—	—	—	—	8
BSET #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 1						4				—	—	—	—	—	—	8
BSET Rn, Rd	B	(Rn8 of Rd8) ← 1		2								—	—	—	—	—	—	2
BSET Rn, @Rd	B	(Rn8 of @Rd16) ← 1			4							—	—	—	—	—	—	8
BSET Rn, @aa:8	B	(Rn8 of @aa:8) ← 1						4				—	—	—	—	—	—	8
BCLR #xx:3, Rd	B	(#xx:3 of Rd8) ← 0		2								—	—	—	—	—	—	2
BCLR #xx:3, @Rd	B	(#xx:3 of @Rd16) ← 0			4							—	—	—	—	—	—	8
BCLR #xx:3, @aa:8	B	(#xx:3 of @aa:8) ← 0						4				—	—	—	—	—	—	8
BCLR Rn, Rd	B	(Rn8 of Rd8) ← 0		2								—	—	—	—	—	—	2
BCLR Rn, @Rd	B	(Rn8 of @Rd16) ← 0			4							—	—	—	—	—	—	8
BCLR Rn, @aa:8	B	(Rn8 of @aa:8) ← 0						4				—	—	—	—	—	—	8
BNOT #xx:3, Rd	B	(#xx:3 of Rd8) ← $\overline{\text{(\#xx:3 of Rd8)}}$		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

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
BTST #xx:3, Rd	B	(#xx:3 of Rd8) → Z		2								—	—	—	↕	—	—	2
BTST #xx:3, @Rd	B	(#xx:3 of @Rd16) → Z			4							—	—	—	↕	—	—	6
BTST #xx:3, @aa:8	B	(#xx:3 of @aa:8) → Z						4				—	—	—	↕	—	—	6
BTST Rn, Rd	B	(Rn8 of Rd8) → Z		2								—	—	—	↕	—	—	2
BTST Rn, @Rd	B	(Rn8 of @Rd16) → Z			4							—	—	—	↕	—	—	6
BTST Rn, @aa:8	B	(Rn8 of @aa:8) → Z						4				—	—	—	↕	—	—	6
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	(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	2
BILD #xx:3, @Rd	B	(#xx:3 of @Rd16) → C			4							—	—	—	—	—	↕	6
BILD #xx:3, @aa:8	B	(#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	$\overline{C}$ → (#xx:3 of Rd8)		2								—	—	—	—	—	—	2
BIST #xx:3, @Rd	B	$\overline{C}$ → (#xx:3 of @Rd16)			4							—	—	—	—	—	—	8
BIST #xx:3, @aa:8	B	$\overline{C}$ → (#xx:3 of @aa:8)						4				—	—	—	—	—	—	8
BAND #xx:3, Rd	B	C∧(#xx:3 of Rd8) → C		2								—	—	—	—	—	↕	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

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Branching Condition	Addressing Mode/ Instruction Length (bytes)									Condition Code						No. of States
				#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
BIOR #xx:3, @aa:8	B	C∨($\overline{\#xx:3 \text{ of } @aa:8}$) → 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⊕($\overline{\#xx:3 \text{ of } Rd8}$) → C			2								—	—	—	—	—	↕	2
BIXOR #xx:3, @Rd	B	C⊕($\overline{\#xx:3 \text{ of } @Rd16}$) → C				4							—	—	—	—	—	↕	6
BIXOR #xx:3, @aa:8	B	C⊕($\overline{\#xx:3 \text{ of } @aa:8}$) → 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

Table A-1 Instruction Set (cont)

Mnemonic	Operand Size	Operation	Addressing Mode/ Instruction Length (bytes)									Condition Code						No. of States
			#xx: 8/16	Rn	@Rn	@(d:16, Rn)	@–Rn/@Rn+	@aa: 8/16	@(d:8, PC)	@@aa	Implied	I	H	N	Z	V	C	
JSR @Rn	—	SP–2 → SP PC → @SP PC ← Rn16			2							—	—	—	—	—	—	6
JSR @aa:16	—	SP–2 → SP PC → @SP PC ← aa:16						4				—	—	—	—	—	—	8
JSR @@aa:8		SP–2 → SP PC → @SP PC ← @aa:8								2		—	—	—	—	—	—	8
RTS	—	PC ← @SP SP+2 → SP									2	—	—	—	—	—	—	8
RTE	—	CCR ← @SP SP+2 → SP PC ← @SP SP+2 → SP									2	↕	↕	↕	↕	↕	↕	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 Repeat @R5 → @R6 R5+1 → R5 R6+1 → R6 R4L–1 → R4L Until R4L=0 else next;									4	—	—	—	—	—	—	4

Notes: (1) Set to 1 when there is a carry or borrow from bit 11; otherwise cleared to 0.

(2) If the result is zero, the previous value of the flag is retained; otherwise the flag is cleared to 0.

(3) Set to 1 if decimal adjustment produces a carry; otherwise retains value prior to arithmetic operation.

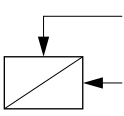
(4) The number of states required for execution is 4n + 9 (n = value of R4L).

(5) Set to 1 if the divisor is negative; otherwise cleared to 0.

(6) Set to 1 if the divisor is zero; otherwise cleared to 0.

A.2 Operation Code Map

Table A-2 is an operation code map. It shows the operation codes contained in the first byte of the instruction code (bits 15 to 8 of the first instruction word).



Instruction when first bit of byte 2 (bit 7 of first instruction word) is 0.

Instruction when first bit of byte 2 (bit 7 of first instruction word) is 1.

Table A-2 Operation Code Map

High \ Low	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	SLEEP	STC	LDC	ORC	XORC	ANDC	LDC	ADD		INC	ADDS	MOV		ADDX	DAA
1	SHLL / SHAL	SHLR / SHAR	ROTXL / ROTL	ROTXR / ROTR	OR	XOR	AND	NOT / NEG	SUB		DEC	SUBS	CMP		SUBX	DAS
2	MOV															
3																
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU			RTS	BSR	RTE			JMP				JSR		
6	BSET	BNOT	BCLR	BTST				BST / BIST	MOV*							
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															

Note: * The PUSH and POP instructions are identical in machine language to MOV instructions.

A.3 Number of Execution States

The tables here can be used to calculate the number of states required for instruction execution. Table A-3 indicates the number of states required for each cycle (instruction fetch, branch address read, stack operation, byte data access, word data access, internal operation).
Table A-4 indicates the number of cycles of each type occurring in each instruction. The total number of states required for execution of an instruction can be calculated from these two tables as follows:

Execution states = $I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$

Examples: When instruction is fetched from on-chip ROM, and an on-chip RAM is accessed.

BSET #0, @FF00
From table A-4:
I = L = 2, J = K = M = N= 0
From table A-3:
$S_I = 2$, $S_L = 2$
Number of states required for execution = $2 \times 2 + 2 \times 2 = 8$
When instruction is fetched from on-chip ROM, branch address is read from on-chip ROM, and on-chip RAM is used for stack area.

JSR @@ 30
From table A-4:
I = 2, J = K = 1, L = M = N = 0
From table A-3:
$S_I = S_J = S_K = 2$
Number of states required for execution = $2 \times 2 + 1 \times 2 + 1 \times 2 = 8$

Table A-3 Number of Cycles in Each Instruction

Execution Status (instruction cycle)		Access Location	
		On-Chip Memory	On-Chip Peripheral Module
Instruction fetch	S_I	2	—
Branch address read	S_J		
Stack operation	S_K		
Byte data access	S_L		2 or 3*
Word data access	S_M		—
Internal operation	S_N	1	

Note: * Depends on which on-chip module is accessed. See 2.9.1, Notes on Data Access for details.

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, Rd	1					
	ADDS.W #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					

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
BCLR	BCLR Rn, @Rd	2			2		
	BCLR Rn, @aa:8	2			2		
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		

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
BSET	BSET Rn, @aa:8	2			2		
BSR	BSR d:8	2		1			
BST	BST #xx:3, Rd	1					
	BST #xx:3, @Rd	2			2		
	BST #xx:3, @aa:8	2			2		
BTST	BTST #xx:3, Rd	1					
	BTST #xx:3, @Rd	2			1		
	BTST #xx:3, @aa:8	2			1		
	BTST Rn, Rd	1					
	BTST Rn, @Rd	2			1		
	BTST Rn, @aa:8	2			1		
BXOR	BXOR #xx:3, Rd	1					
	BXOR #xx:3, @Rd	2			1		
	BXOR #xx:3, @aa:8	2			1		
CMP	CMP. B #xx:8, Rd	1					
	CMP. B Rs, Rd	1					
	CMP.W Rs, Rd	1					
DAA	DAA.B Rd	1					
DAS	DAS.B Rd	1					
DEC	DEC.B Rd	1					
DIVXU	DIVXU.B Rs, Rd	1					12
EEPMOV	EEPMOV	2			2n+2*		1
INC	INC.B Rd	1					
JMP	JMP @Rn	2					
	JMP @aa:16	2					2
	JMP @@aa:8	2	1				2
JSR	JSR @Rn	2		1			
	JSR @aa:16	2		1			2
	JSR @@aa:8	2	1	1			
LDC	LDC #xx:8, CCR	1					
	LDC Rs, CCR	1					
MOV	MOV.B #xx:8, Rd	1					
	MOV.B Rs, Rd	1					
	MOV.B @Rs, Rd	1			1		

Note: n: Initial value in R4L. The source and destination operands 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 @(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	
MULXU	MULXU.B 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					
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

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
SHLL	SHLL.B Rd	1					
SHAL	SHAL.B Rd	1					
SHAR	SHAR.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, Rd	1					
	SUBS.W #2, Rd	1					
POP	POP Rd	1		1			2
PUSH	PUSH Rs	1		1			2
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					

Appendix B Internal I/O Registers

B.1 Addresses

Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'F740										
H'F741										
H'F742										
H'F743										
H'F744										
H'F770	TIER	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—	Timer X
H'F771	TCSRX	ICFA	ICFB	ICFC	ICFD	OCFA	OCFB	OVF	CCLRA	
H'F772	FRCH	FRCH7	FRCH6	FRCH5	FRCH4	FRCH3	FRCH2	FRCH1	FRCH0	
H'F773	FRCL	FRCL7	FRCL6	FRCL5	FRCL4	FRCL3	FRCL2	FRCL1	FRCL0	
H'F774	OCRAH/ OCRBH	OCRAH7/ OCRBH7	OCRAH6/ OCRBH6	OCRAH5/ OCRBH5	OCRAH4/ OCRBH4	OCRAH3/ OCRBH3	OCRAH2/ OCRBH2	OCRAH1/ OCRBH1	OCRAH0/ OCRBH0	
H'F775	OCRAL/ OCRBL	OCRAL7/ OCRBL7	OCRAL6/ OCRBL6	OCRAL5/ OCRBL5	OCRAL4/ OCRBL4	OCRAL3/ OCRBL3	OCRAL2/ OCRBL2	OCRAL1/ OCRBL1	OCRAL0/ OCRBL0	
H'F776	TCRX	IEDGA	IEDGB	IEDGC	IEDGD	BUFEA	BUFEB	CKS1	CKS0	
H'F777	TOCR	—	—	—	OCRS	OEA	OEB	OLVLA	OLVLB	
H'F778	ICRAH	ICRAH7	ICRAH6	ICRAH5	ICRAH4	ICRAH3	ICRAH2	ICRAH1	ICRAH0	
H'F779	ICRAL	ICRAL7	ICRAL6	ICRAL5	ICRAL4	ICRAL3	ICRAL2	ICRAL1	ICRAL0	
F'F77A	ICRBH	ICRBH7	ICRBH6	ICRBH5	ICRBH4	ICRBH3	ICRBH2	ICRBH1	ICRBH0	
F'F77B	ICRBL	ICRBL7	ICRBL6	ICRBL5	ICRBL4	ICRBL3	ICRBL2	ICRBL1	ICRBL0	
H'F77C	ICRCH	ICRCH7	ICRCH6	ICRCH5	ICRCH4	ICRCH3	ICRCH2	ICRCH1	ICRCH0	
H'F77D	ICRCL	ICRCL7	ICRCL6	ICRCL5	ICRCL4	ICRCL3	ICRCL2	ICRCL1	ICRCL0	
H'F77E	ICRDH	ICRDH7	ICRDH6	ICRDH5	ICRDH4	ICRDH3	ICRDH2	ICRDH1	ICRDH0	
H'F77F	ICRDL	ICRDL7	ICRDL6	ICRDL5	ICRDL4	ICRDL3	ICRDL2	ICRDL1	ICRDL0	
H'FFA0	SCR1	SNC1	SNC0	MRKON	LTCH	CKS3	CKS2	CKS1	CKS0	SCI1
H'FFA1	SCSR1	—	SOL	ORER	—	—	—	MTRF	STF	
H'FFA2	SDRU	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0	
H'FFA3	SDRL	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0	
H'FFA4										
H'FFA5										
H'FFA6										
H'FFA7										
H'FFA8	SMR	COM	CHR	PE	PM	STOP	MP	CKS1	CKS0	SCI3
H'FFA9	BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0	
H'FFAA	SCR3	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0	
H'FFAB	TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	

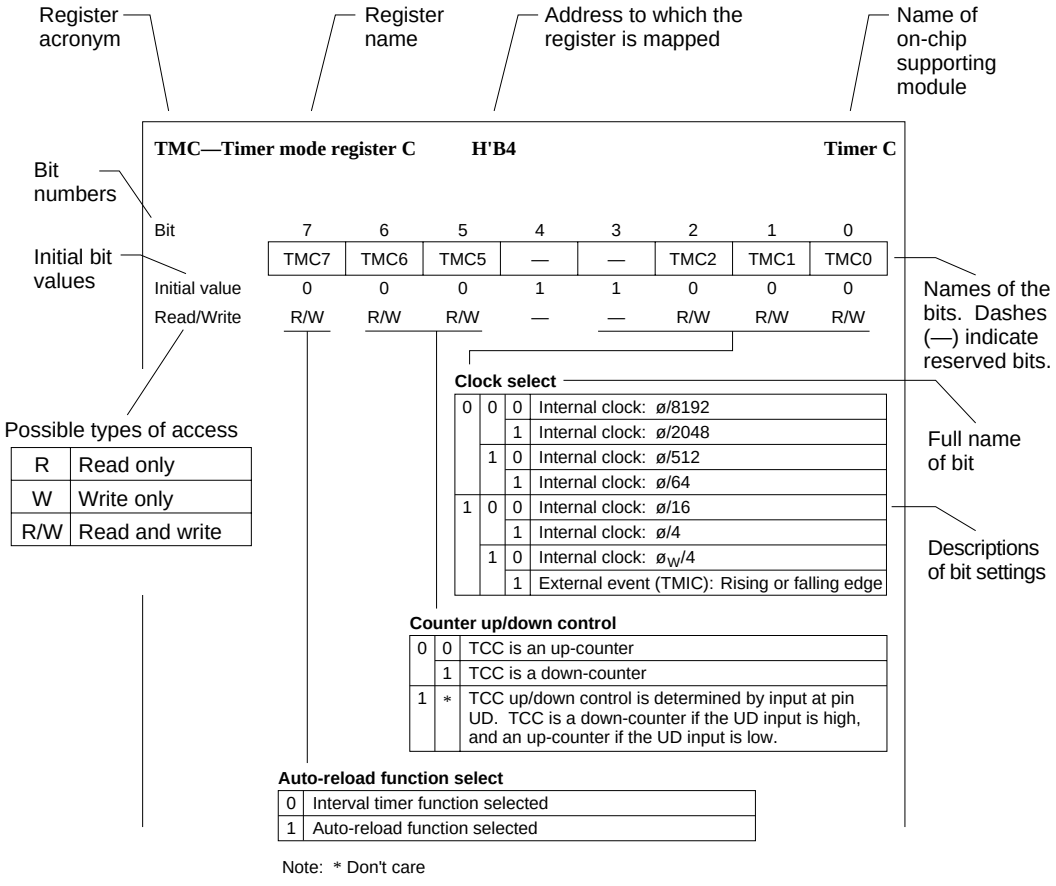
Notation
SCI1: Serial communication interface 1

Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFAC	SSR	TDRE	RDRF	OER	FER	PER	TEND	MPBR	MPBT	SCI3
H'FFAD	RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0	
H'FFAE										
H'FFAF										
H'FFB0	TMA	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0	Timer A
H'FFB1	TCA	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0	
H'FFB2	TMB1	TMB17	—	—	—	—	TMB12	TMB11	TMB10	Timer B1
H'FFB3	TCB1/ TLB1	TCB17/ TLB17	TCB16/ TLB16	TCB15/ TLB15	TCB14/ TLB14	TCB13/ TLB13	TCB12/ TLB12	TCB11/ TLB11	TCB10/ TLB10	
H'FFB4										
H'FFB5										
H'FFB6										
H'FFB7										
H'FFB8	TCRV0	CMIEB	CMIEA	OVIE	CCLR1	CCLR0	CKS2	CKS1	CKS0	Timer V
H'FFB9	TCSRV	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0	
H'FFBA	TCORA	TCORA7	TCORA6	TCORA5	TCORA4	TCORA3	TCORA2	TCORA1	TCORA0	
H'FFBB	TCORB	TCORB7	TCORB6	TCORB5	TCORB4	TCORB3	TCORB2	TCORB1	TCORB0	
H'FFBC	TCNTV	TCNTV7	TCNTV6	TCNTV5	TCNTV4	TCNTV3	TCNTV2	TCNTV1	TCNTV0	
H'FFBD	TCRV1	—	—	—	TVEG1	TVEG0	TRGE	—	ICKS0	
H'FFBE	TCSRW	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	BOWI	WRST	Watchdog timer
H'FFBF	TCW	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0	
H'FFC0										
H'FFC1										
H'FFC2										
H'FFC3										
H'FFC4	AMR	CKS	TRGE	—	—	CH3	CH2	CH1	CH0	A/D converter
H'FFC5	ADRR	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	
H'FFC6	ADSR	ADSF	—	—	—	—	—	—	—	
H'FFC7										
H'FFC8										
H'FFC9										
H'FFCA										
H'FFCB										

Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFCC										
H'FFCD										
H'FFCE										
H'FFCF										
H'FFD0	PWCR	—	—	—	—	—	—	—	PWCR0	14-bit PWM
H'FFD1	PWDRU	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDRU1	PWDRU0	
H'FFD2	PWDRL	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0	
H'FFD3										
H'FFD4	PDR1	$P1_7$	$P1_6$	$P1_5$	$P1_4$	$P1_3$	$P1_2$	$P1_1$	$P1_0$	I/O ports
H'FFD5	PDR2	$P2_7$	$P2_6$	$P2_5$	$P2_4$	$P2_3$	$P2_2$	$P2_1$	$P2_0$	
H'FFD6	PDR3	—	—	$P3_5$	$P3_4$	$P3_3$	$P3_2$	$P3_1$	$P3_0$	
H'FFD7										
H'FFD8	PDR5	$P5_7$	$P5_6$	$P5_5$	$P5_4$	$P5_3$	$P5_2$	$P5_1$	$P5_0$	
H'FFD9	PDR6	$P6_7$	$P6_6$	$P6_5$	$P6_4$	$P6_3$	$P6_2$	$P6_1$	$P6_0$	
H'FFDA	PDR7	$P7_7$	$P7_6$	$P7_5$	$P7_4$	$P7_3$	$P7_2$	$P7_1$	$P7_0$	
H'FFDB	PDR8	$P8_7$	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$	
H'FFDC	PDR9	—	—	—	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$	
H'FFDD	PDRB	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0	
H'FFDE										
H'FFDF										
H'FFE0										I/O ports
H'FFE1										
H'FFE2										
H'FFE3										
H'FFE4	PCR1	$PCR1_7$	$PCR1_6$	$PCR1_5$	$PCR1_4$	$PCR1_3$	$PCR1_2$	$PCR1_1$	$PCR1_0$	
H'FFE5	PCR2	$PCR2_7$	$PCR2_6$	$PCR2_5$	$PCR2_4$	$PCR2_3$	$PCR2_2$	$PCR2_1$	$PCR2_0$	
H'FFE6	PCR3	—	—	$PCR3_5$	$PCR3_4$	$PCR3_3$	$PCR3_2$	$PCR3_1$	$PCR3_0$	
H'FFE7										
H'FFE8	PCR5	$PCR5_7$	$PCR5_6$	$PCR5_5$	$PCR5_4$	$PCR5_3$	$PCR5_2$	$PCR5_1$	$PCR5_0$	
H'FFE9	PCR6	$PCR6_7$	$PCR6_6$	$PCR6_5$	$PCR6_4$	$PCR6_3$	$PCR6_2$	$PCR6_1$	$PCR6_0$	
H'FFEA	PCR7	$PCR7_7$	$PCR7_6$	$PCR7_5$	$PCR7_4$	$PCR7_3$	$PCR7_2$	$PCR7_1$	$PCR7_0$	
H'FFEB	PCR8	$PCR8_7$	$PCR8_6$	$PCR8_5$	$PCR8_4$	$PCR8_3$	$PCR8_2$	$PCR8_1$	$PCR8_0$	
H'FFEC	PCR9	—	—	—	$PCR9_4$	$PCR9_3$	$PCR9_2$	$PCR9_1$	$PCR9_0$	

Address	Register Name	Bit Names								Module Name
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
H'FFED	PUCR1	$PUCR1_7$	$PUCR1_6$	$PUCR1_5$	$PUCR1_4$	$PUCR1_3$	$PUCR1_2$	$PUCR1_1$	$PUCR1_0$	I/O ports
H'FFEE	PUCR3	—	—	$PUCR3_5$	$PUCR3_4$	$PUCR3_3$	$PUCR3_2$	$PUCR3_1$	$PUCR3_0$	
H'FFEF	PUCR5	$PUCR5_7$	$PUCR5_6$	$PUCR5_5$	$PUCR5_4$	$PUCR5_3$	$PUCR5_2$	$PUCR5_1$	$PUCR5_0$	
H'FFF0	SYSCR1	SSBY	STS2	STS1	STS0	LSON	—	MA1	MA0	System control
H'FFF1	SYSCR2	—	—	—	NESEL	DTON	MSON	SA1	SA0	
H'FFF2	IEGR1	—	—	—	—	IEG3	IEG2	IEG1	IEG0	
H'FFF3	IEGR2	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2	INTEG1	INTEG0	
H'FFF4	IENR1	IENTB1	IENTA	—	—	IEN3	IEN2	IEN1	IEN0	
H'FFF5	IENR2	IENDT	IENAD	—	IENSI	—	—	—	—	
H'FFF6	IENR3	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0	
H'FFF7	IRR1	IRRTB1	IRRTA	—	—	IRRI3	IRRI2	IRRI1	IRRI0	
H'FFF8	IRR2	IRRDT	IRRAD	—	IRRS1	—	—	—	—	
H'FFF9	IRR3	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0	
H'FFFA										
H'FFFB										
H'FFFC	PMR1	IRQ3	IRQ2	IRQ1	PWM	—	—	—	TMOW	I/O ports
H'FFFD	PMR3	—	—	—	—	—	SO1	SI1	SCK1	
H'FFFE										
H'FFFF	PMR7	—	—	—	—	—	TXD	—	POF1	I/O ports

B.2 Functions



0	0	0	Internal clock: ø/8192
		1	Internal clock: ø/2048
	1	0	Internal clock: ø/512
		1	Internal clock: ø/64
1	0	0	Internal clock: ø/16
		1	Internal clock: ø/4
	1	0	Internal clock: $ø_W/4$
		1	External event (TMIC): Rising or falling edge

0	0	TCC is an up-counter
	1	TCC is a down-counter
1	*	TCC up/down control is determined by input at pin UD. TCC is a down-counter if the UD input is high, and an up-counter if the UD input is low.

0	Interval timer function selected
1	Auto-reload function selected

Note: * Don't care

Bit	7	6	5	4	3	2	1	0
	ICIAE	ICIBE	ICICE	ICIDE	OCIAE	OCIBE	OVIE	—
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

Timer overflow interrupt enable

0	Interrupt request (FOVI) by OVF is disabled
1	Interrupt request (FOVI) by OVF is enabled

Output compare interrupt B enable

0	Interrupt request (OCIB) by OCFB is disabled
1	Interrupt request (OCIB) by OCFB is enabled

Output compare interrupt A enable

0	Interrupt request (OCIA) by OCFA is disabled
1	Interrupt request (OCIA) by OCFA is enabled

Input capture interrupt D enable

0	Interrupt request (ICID) by ICFD is disabled
1	Interrupt request (ICID) by ICFD is enabled

Input capture interrupt C enable

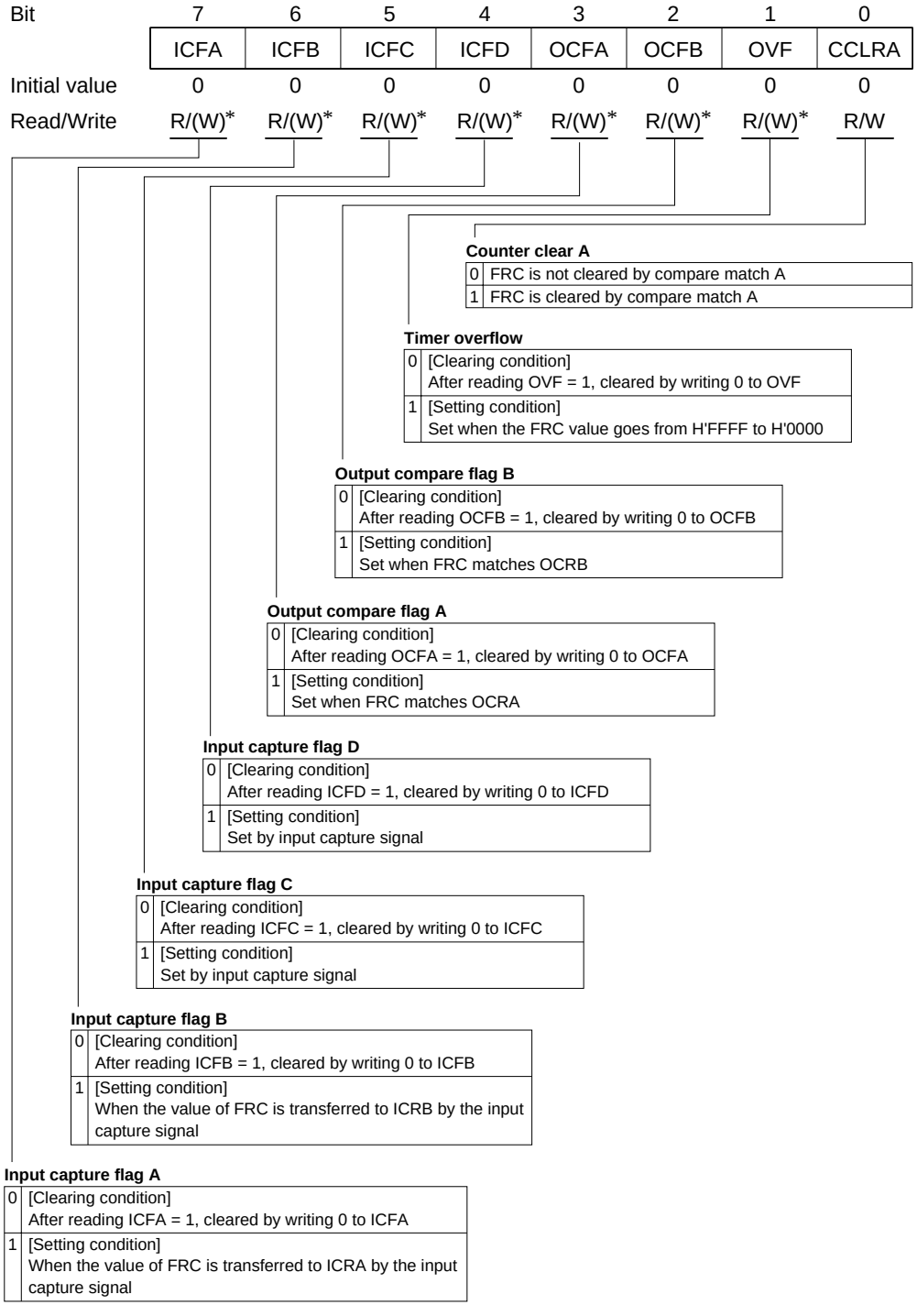
0	Interrupt request (ICIC) by ICFC is disabled
1	Interrupt request (ICIC) by ICFC is enabled

Input capture interrupt B enable

0	Interrupt request (ICIB) by ICFB is disabled
1	Interrupt request (ICIB) by ICFB is enabled

Input capture interrupt A enable

0	Interrupt request (ICIA) by ICFA is disabled
1	Interrupt request (ICIA) by ICFA is enabled



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

FRCH—Free-running counter H **H'F772** **Timer X**

Bit	7	6	5	4	3	2	1	0
	FRCH7	FRCH6	FRCH5	FRCH4	FRCH3	FRCH2	FRCH1	FRCH0
Initial value	0	0	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

FRCL—Free-running counter L **H'F773** **Timer X**

Bit	7	6	5	4	3	2	1	0
	FRCL7	FRCL6	FRCL5	FRCL4	FRCL3	FRCL2	FRCL1	FRCL0
Initial value	0	0	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

OCRAH—Output compare register AH **H'F774** **Timer X**

Bit	7	6	5	4	3	2	1	0
	OCRAH7	OCRAH6	OCRAH5	OCRAH4	OCRAH3	OCRAH2	OCRAH1	OCRAH0
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

OCRBH—Output compare register BH **H'F774** **Timer X**

Bit	7	6	5	4	3	2	1	0
	OCRBH7	OCRBH6	OCRBH5	OCRBH4	OCRBH3	OCRBH2	OCRBH1	OCRBH0
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

OCRAL—Output compare register AL **H'F775** **Timer X**

Bit	7	6	5	4	3	2	1	0
	OCRAL7	OCRAL6	OCRAL5	OCRAL4	OCRAL3	OCRAL2	OCRAL1	OCRAL0
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

OCRBL—Output compare register BL **H'F775** **Timer X**

Bit	7	6	5	4	3	2	1	0
	OCRBL7	OCRBL6	OCRBL5	OCRBL4	OCRBL3	OCRBL2	OCRBL1	OCRBL0
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

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: ø/2
	1	Internal clock: ø/8
1	0	Internal clock: ø/32
	1	External clock: rising edge

Buffer enable B

0	ICRD is not used as a buffer register for ICRB
1	ICRD is used as a buffer register for OCRB

Buffer enable A

0	ICRC is not used as a buffer register for ICRA
1	ICRC is used as a buffer register for OCRA

Input edge select D

0	Falling edge of input D is captured
1	Rising edge of input D is captured

Input edge select C

0	Falling edge of input C is captured
1	Rising edge of input C is captured

Input edge select B

0	Falling edge of input B is captured
1	Rising edge of input B is captured

Input edge select A

0	Falling edge of input A is captured
1	Rising edge of input A is captured

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	Low level
1	High level

Output level A

0	Low level
1	High level

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	OCRA is selected
1	OCRB is selected

ICRAH—Input capture register AH **H'F778** **Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRAH7	ICRAH6	ICRAH5	ICRAH4	ICRAH3	ICRAH2	ICRAH1	ICRAH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

ICRAL—Input capture register AL **H'F779** **Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRAL7	ICRAL6	ICRAL5	ICRAL4	ICRAL3	ICRAL2	ICRAL1	ICRAL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

ICRBH—Input capture register BH **H'F77A** **Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRBH7	ICRBH6	ICRBH5	ICRBH4	ICRBH3	ICRBH2	ICRBH1	ICRBH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

ICRBL—Input capture register BL **H'F77B** **Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRBL7	ICRBL6	ICRBL5	ICRBL4	ICRBL3	ICRBL2	ICRBL1	ICRBL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

ICRCH—Input capture register CH **H'F77C** **Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRCH7	ICRCH6	ICRCH5	ICRCH4	ICRCH3	ICRCH2	ICRCH1	ICRCH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

ICRCL—Input capture register CL **H'F77D** **Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRCL7	ICRCL6	ICRCL5	ICRCL4	ICRCL3	ICRCL2	ICRCL1	ICRCL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

ICRDH—Input capture register DH **H'F77E** **Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRDH7	ICRDH6	ICRDH5	ICRDH4	ICRDH3	ICRDH2	ICRDH1	ICRDH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

ICRDL—Input capture register DL **H'F77F** **Timer X**

Bit	7	6	5	4	3	2	1	0
	ICRDL7	ICRDL6	ICRDL5	ICRDL4	ICRDL3	ICRDL2	ICRDL1	ICRDL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
	SNC1	SNC0	MRKON	LTCH	CKS3	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 (CKS2 to CKS0)

Bit 2	Bit 1	Bit 0	Prescaler Division	Serial Clock Cycle Synchronous ø = 5 MHz	Serial Clock Cycle Synchronous ø = 2.5 MHz
CKS2	CKS1	CKS0			
0	0	0	ø/1024	204.8 µs	409.6 µs
		1	ø/256	51.2 µs	102.4 µs
	1	0	ø/64	12.8 µs	25.6 µs
		1	ø/32	6.4 µs	12.8 µs
1	0	0	ø/16	3.2 µs	6.4 µs
		1	ø/8	1.6 µs	3.2 µs
	1	0	ø/4	0.8 µs	1.6 µs
		1	ø/2	—	0.8 µs

Clock source select

0	Clock source is prescaler S, and pin SCK_1 is output pin
1	Clock source is external clock, and pin SCK_1 is input pin*

Note: * Input an external clock equivalent to a frequency lower than ø/4.

LATCH TAIL select

0	HOLD TAIL is output
1	LATCH TAIL is output

TAIL MARK control

0	TAIL MARK is not output (synchronous mode)
1	TAIL MARK is output (SSB mode)

Operation mode select 1, 0

0	0	8-bit mode
	1	16-bit mode
1	0	Continuous clock output mode
	1	Reserved

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

Start flag

0	Read	Indicates that transfer is stopped
	Write	Invalid
1	Read	Indicates transfer in progress
	Write	Starts a transfer operation

TAIL MARK transmit flag

0	Idle state and 8- or -16-bit data transfer in progress
1	TAIL MARK transmission in progress

Overrun error flag

0	[Clearing condition] After reading 1, cleared by writing 0
1	[Setting condition] Set if a clock pulse is input after transfer is complete, when an external clock is used

Extended data bit

0	Read	SO_1 pin output level is low
	Write	SO_1 pin output level changes to low
1	Read	SO_1 pin output level is high
	Write	SO_1 pin output level changes to high

Note: * Only a write of 0 for flag clearing is possible.

SDRU—Serial data register U **H'FFA2** **SCI1**

Bit	7	6	5	4	3	2	1	0
	SDRU7	SDRU6	SDRU5	SDRU4	SDRU3	SDRU2	SDRU1	SDRU0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores transmit and receive data

8-bit transfer mode: Not used
16-bit transfer mode: Upper 8 bits of data

SDRL—Serial data register L **H'FFA3** **SCI1**

Bit	7	6	5	4	3	2	1	0
	SDRL7	SDRL6	SDRL5	SDRL4	SDRL3	SDRL2	SDRL1	SDRL0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Stores transmit and receive data

8-bit transfer mode: 8-bit data
16-bit transfer mode: Lower 8 bits of data

Bit	7	6	5	4	3	2	1	0
	COM	CHR	PE	PM	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/4 clock
1	0	ø/16 clock
1	1	ø/64 clock

Multiprocessor mode

0	Multiprocessor communication function disabled
1	Multiprocessor communication function enabled

Stop bit length

0	1 stop bit
1	2 stop bits

Parity mode

0	Even parity
1	Odd parity

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled

Character length

0	8-bit data
1	7-bit data

Communication mode

0	Asynchronous mode
1	Synchronous mode

Bit	7	6	5	4	3	2	1	0
	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
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

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

Bit 1	Bit 0	Description		
CKE1	CKE0	Communication Mode	Clock Source	SCK_3 Pin Function
0	0	Asynchronous	Internal clock	I/O port
		Synchronous	Internal clock	Serial clock output
0	1	Asynchronous	Internal clock	Clock output
		Synchronous	Reserved (Do not specify this combination)	
1	0	Asynchronous	External clock	Clock input
		Synchronous	External clock	Serial clock input
1	1	Asynchronous	Reserved (Do not specify this combination)	
		Synchronous	Reserved (Do not specify this combination)	

Transmit end interrupt enable

0	Transmit end interrupt request (TEI) disabled
1	Transmit end interrupt request (TEI) enabled

Multiprocessor interrupt enable

0	Multiprocessor interrupt request disabled (normal receive operation) [Clearing conditions] When data is received in which the multiprocessor bit is set to 1
1	Multiprocessor interrupt request enabled The receive interrupt request (RXI), receive error interrupt request (ERI), and setting of the RDRF, FER, and OER flags in the serial status register (SSR), are disabled until data with the multiprocessor bit set to 1 is received.

Receive enable

0	Receive operation disabled (RXD pin is I/O port)
1	Receive operation enabled (RXD pin is receive data pin)

Transmit enable

0	Transmit operation disabled (TXD pin is transmit data pin)
1	Transmit operation enabled (TXD pin is transmit data pin)

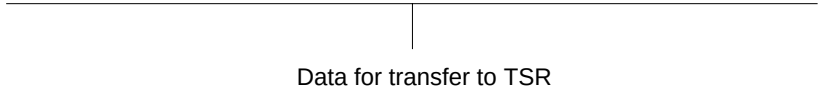
Receive interrupt enable

0	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) disabled
1	Receive data full interrupt request (RXI) and receive error interrupt request (ERI) enabled

Transmit interrupt enable

0	Transmit data empty interrupt request (TXI) disabled
1	Transmit data empty interrupt request (TXI) enabled

Bit	7	6	5	4	3	2	1	0
	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
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



Bit	7	6	5	4	3	2	1	0
	TDRE	RDRF	OER	FER	PER	TEND	MPBR	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	A 0 multiprocessor bit is transmitted
1	A 1 multiprocessor bit is transmitted

Multiprocessor bit receive

0	Data in which the multiprocessor bit is 0 has been received
1	Data in which the multiprocessor bit is 1 has been received

Transmit end

0	Transmission in progress [Clearing conditions] • After reading TDRE = 1, cleared by writing 0 to TDRE • When data is written to TDR by an instruction
1	Transmission ended [Setting conditions] • When bit TE in serial control register 3 (SCR3) is cleared to 0 • When bit TDRE is set to 1 when the last bit of a transmit character is sent

Parity error

0	Reception in progress or completed normally [Clearing conditions] After reading PER = 1, cleared by writing 0 to PER
1	A parity error has occurred during reception [Setting conditions] When the number of 1 bits in the receive data plus parity bit does not match the parity designated by the parity mode bit (PM) in the serial mode register (SMR)

Framing error

0	Reception in progress or completed normally [Clearing conditions] After reading FER = 1, cleared by writing 0 to FER
1	A framing error has occurred during reception [Setting conditions] When the stop bit at the end of the receive data is checked for a value of 1 at completion of reception, and the stop bit is 0

Overrun error

0	Reception in progress or completed [Clearing conditions] After reading OER = 1, cleared by writing 0 to OER
1	An overrun error has occurred during reception [Setting conditions] When the next serial reception is completed with RDRF set to 1

Receive data register full

0	There is no receive data in RDR [Clearing conditions] • After reading RDRF = 1, cleared by writing 0 to RDRF • When RDR data is read by an instruction
1	There is receive data in RDR [Setting conditions] When reception ends normally and receive data is transferred from RSR to RDR

Transmit data register empty

0	Transmit data written in TDR has not been transferred to TSR [Clearing conditions] • After reading TDRE = 1, cleared by writing 0 to TDRE • When data is written to TDR by an instruction
1	Transmit data has not been written to TDR, or transmit data written in TDR has been transferred to TSR [Setting conditions] • When bit TE in serial control register 3 (SCR3) is cleared to 0 • When data is transferred from TDR to TSR

Note: * Only a write of 0 for flag clearing is possible.

RDR—Receive data register **H'FFAD** **SCI3**

Bit	7	6	5	4	3	2	1	0
	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TMA—Timer mode register A **H'FFB0** **Timer A**

Bit	7	6	5	4	3	2	1	0
	TMA7	TMA6	TMA5	—	TMA3	TMA2	TMA1	TMA0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W	R/W	R/W	—	R/W	R/W	R/W	R/W

Clock output select

0	0	0	ø/32
		1	ø/16
	1	0	ø/8
		1	ø/4
1	0	0	$ø_W$/32
		1	$ø_W$/16
	1	0	$ø_W$/8
		1	$ø_W$/4

Internal clock select

TMA3	TMA2	TMA1	TMA0	Prescaler and Divider Ratio or Overflow Period		Function
0	0	0	0	PSS	ø/8192	Interval timer
			1	PSS	ø/4096	
		1	0	PSS	ø/2048	
			1	PSS	ø/512	
	1	0	0	PSS	ø/256	
			1	PSS	ø/128	
		1	0	PSS	ø/32	
			1	PSS	ø/8	
1	0	0	0	PSW	1 s	Time base
			1	PSW	0.5 s	
		1	0	PSW	0.25 s	
			1	PSW	0.03125 s	
	1	0	0	PSW and TCA are reset		
			1			
		1	0			
			1			

TCA—Timer counter A **H'FFB1** **Timer A**

Bit	7	6	5	4	3	2	1	0
	TCA7	TCA6	TCA5	TCA4	TCA3	TCA2	TCA1	TCA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

TMB1—Timer mode register B1 **H'FFB2** **Timer B1**

Bit	7	6	5	4	3	2	1	0
	TMB17	—	—	—	—	TMB12	TMB11	TMB10
Initial value	0	1	1	1	1	0	0	0
Read/Write	R/W	—	—	—	—	R/W	R/W	R/W

Auto-reload function select

0	Interval timer function selected
1	Auto-reload function selected

Clock select

0	0	0	Internal clock: ø/8192
		1	Internal clock: ø/2048
	1	0	Internal clock: ø/512
		1	Internal clock: ø/256
1	0	0	Internal clock: ø/64
		1	Internal clock: ø/16
	1	0	Internal clock: ø/4
		1	External event (TMIB): Rising or falling edge

TCB1—Timer counter B1 **H'FFB3** **Timer B1**

Bit	7	6	5	4	3	2	1	0
	TCB17	TCB16	TCB15	TCB14	TCB13	TCB12	TCB11	TCB10
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Count value

TLB1—Timer load register B1 **H'FFB3** **Timer B1**

Bit	7	6	5	4	3	2	1	0
	TLB17	TLB16	TLB15	TLB14	TLB13	TLB12	TLB11	TLB10
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Reload value

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

TCRV0			TCRV1	Description
Bit 2	Bit 1	Bit 0	Bit 0	
CKS2	CKS1	CKS0	ICKS0	
0	0	0	—	Clock input disabled
		1	0	Internal clock: ø/4, falling edge
			1	Internal clock: ø/8, falling edge
	1	0	0	Internal clock: ø/16, falling edge
			1	Internal clock: ø/32, falling edge
		1	0	Internal clock: ø/64, falling edge
			1	Internal clock: ø/128, falling edge
1	0	0	—	Clock input disabled
		1	—	External clock: rising edge
	1	0	—	External clock: falling edge
		1	—	External clock: rising and falling edges

Counter clear 1 and 0

0	Clearing is disabled
	Cleared by compare match A
1	Cleared by compare match B
	Cleared by rising edge of external reset input

Timer overflow interrupt enable

0	Interrupt request (OVI) from OVF disabled
1	Interrupt request (OVI) from OVF enabled

Compare match interrupt enable A

0	Interrupt request (CMIA) from CMFA disabled
1	Interrupt request (CMIA) from CMFA enabled

Compare match interrupt enable B

0	Interrupt request (CMIB) from CMFB disabled
1	Interrupt request (CMIB) from CMFB enabled

Bit	7	6	5	4	3	2	1	0
	CMFB	CMFA	OVF	—	OS3	OS2	OS1	OS0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	—	R/W	R/W	R/W	R/W

Output select

0	0	No change at compare match A
	1	0 output at compare match A
1	0	1 output at compare match A
	1	Output toggles at compare match A

Output select

0	0	No change at compare match B
	1	0 output at compare match B
1	0	1 output at compare match B
	1	Output toggles at compare match B

Timer overflow flag

0	[Clearing condition] After reading OVF = 1, cleared by writing 0 to OVF
1	[Setting condition] Set when TCNTV overflows from H'FF to H'00

Compare match flag A

0	[Clearing condition] After reading CMFA = 1, cleared by writing 0 to CMFA
1	[Setting condition] Set when the TCNTV value matches the TCORA value

Compare match flag B

0	[Clearing condition] After reading CMFB = 1, cleared by writing 0 to CMFB
1	[Setting condition] Set when the TCNTV value matches the TCORB value

Note: * Only a write of 0 for flag clearing is possible.

TCORA—Time constant register A **H'FFBA** **Timer V**

Bit	7	6	5	4	3	2	1	0
	TCORA7	TCORA6	TCORA5	TCORA4	TCORA3	TCORA2	TCORA1	TCORA0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCORB—Time constant register B **H'FFBB** **Timer V**

Bit	7	6	5	4	3	2	1	0
	TCORB7	TCORB6	TCORB5	TCORB4	TCORB3	TCORB2	TCORB1	TCORB0
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

TCNTV—Timer counter V **H'FFBC** **Timer V**

Bit	7	6	5	4	3	2	1	0
	TCNTV7	TCNTV6	TCNTV5	TCNTV4	TCNTV3	TCNTV2	TCNTV1	TCNTV0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	—	—	—	TVEG1	TVEG0	TRGE	—	ICKS0
Initial value	1	1	1	0	0	0	1	0
Read/Write	—	—	—	R/W	R/W	R/W	—	R/W

Internal clock select

Selects the TCNTV clock source, with bits CKS2 to CKS0 in TCRV0

TRGV input enable

0	TCNTV counting is not triggered by input at the TRGV pin, and does not stop when TCNTV is cleared by compare match
1	TCNTV counting is triggered by input at the TRGV pin, and stops when TCNTV is cleared by compare match

TRGV input edge select

0	0	TRGV trigger input is disabled
	1	Rising edge is selected
1	0	Falling edge is selected
	1	Rising and falling edges are both selected

Bit	7	6	5	4	3	2	1	0
	B6WI	TCWE	B4WI	TCSRWE	B2WI	WDON	B0WI	WRST
Initial value	1	0	1	0	1	0	1	0
Read/Write	R	R/(W)*	R	R/(W)*	R	R/(W)*	R	R/(W)*

Watchdog timer reset

0	[Clearing conditions] • Reset by $\overline{\text{RES}}$ pin • When TCSRWE = 1, and 0 is written in both B0WI and WRST
1	[Setting condition] When TCW overflows and a reset signal is generated

Bit 0 write inhibit

0	Bit 0 is write-enabled
1	Bit 0 is write-protected

Watchdog timer on

0	Watchdog timer operation is disabled
1	Watchdog timer operation is enabled

Bit 2 write inhibit

0	Bit 2 is write-enabled
1	Bit 2 is write-protected

Timer control/status register W write enable

0	Data cannot be written to TCSRW bits 2 and 0
1	Data can be written to TCSRW bits 2 and 0

Bit 4 write inhibit

0	Bit 4 is write-enabled
1	Bit 4 is write-protected

Timer counter W write enable

0	Data cannot be written to TCW bit 8
1	Data can be written to TCW bit 8

Bit 6 write inhibit

0	Bit 6 is write-enabled
1	Bit 6 is write-protected

Note: * Write is permitted only under certain conditions.

TCW—Timer counter W **H'FFBF** **Watchdog timer**

Bit	7	6	5	4	3	2	1	0
	TCW7	TCW6	TCW5	TCW4	TCW3	TCW2	TCW1	TCW0
Initial value	0	0	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
	CKS	TRGE	—	—	CH3	CH2	CH1	CH0
Initial value	0	0	1	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

Channel select

Bit 3	Bit 2	Bit 1	Bit 0	
CH3	CH2	CH1	CH0	Analog Input Channel
0	0	*	*	No channel selected
	1	0	0	AN_0
			1	AN_1
		1	0	AN_2
			1	AN_3
1	0	0	0	AN_4
			1	AN_5
		1	0	AN_6
			1	AN_7
	1	0	0	Reserved
		0	1	Reserved
		1	0	Reserved
		1	1	Reserved

* Don't care

External trigger select

0	Disables start of A/D conversion by external trigger
1	Enables start of A/D conversion by rising or falling edge of external trigger at pin $\overline{\text{ADTRG}}$

Clock select

Bit 7		Conversion Time	
CKS	Conversion Period	ø = 2 MHz	ø = 5 MHz
0	62/ø	31 µs	12.4 µs
1	31/ø	15.5 µs	—*1

Note: * Operation is not guaranteed with a conversion time of less than 12.4 µs. Select a setting that gives a conversion time of 12.4 µs or more.

ADRR—A/D result register **H'FFC5** **A/D converter**

Bit	7	6	5	4	3	2	1	0
	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
Initial value	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed	Not fixed
Read/Write	R	R	R	R	R	R	R	R

A/D conversion result

ADSR—A/D start register **H'FFC6** **A/D converter**

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

A/D status flag

0	Read	Indicates completion of A/D conversion
	Write	Stops A/D conversion
1	Read	Indicates A/D conversion in progress
	Write	Starts A/D conversion

PWCR—PWM control register **H'FFD0** **14-bit PWM**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWCR0
Initial value	1	1	1	1	1	1	1	0
Read/Write	—	—	—	—	—	—	—	W

Clock select

0	The input clock is ø/2 (tø* = 2/ø). The conversion period is 16,384/ø, with a minimum modulation width of 1/ø.
1	The input clock is ø/4 (tø* = 4/ø). The conversion period is 32,768/ø, with a minimum modulation width of 2/ø.

Note: *tø: Period of PWM input clock

PWDRU—PWM data register U **H'FFD1** **14-bit PWM**

Bit	7	6	5	4	3	2	1	0
	—	—	PWDRU5	PWDRU4	PWDRU3	PWDRU2	PWDUR1	PWDRU0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

Upper 6 bits of data for generating PWM waveform

PWDRL—PWM data register L **H'FFD2** **14-bit PWM**

Bit	7	6	5	4	3	2	1	0
	PWDRL7	PWDRL6	PWDRL5	PWDRL4	PWDRL3	PWDRL2	PWDRL1	PWDRL0
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Lower 8 bits of data for generating PWM waveform

PDR1—Port data register 1 **H'FFD4** **I/O ports**

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

PDR2—Port data register 2 **H'FFD5** **I/O ports**

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

PDR3—Port data register 3 **H'FFD6** **I/O ports**

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

PDR5—Port data register 5 **H'FFD8** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$P5_7$	$P5_6$	$P5_5$	$P5_4$	$P5_3$	$P5_2$	$P5_1$	$P5_0$
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

PDR6—Port data register 6 **H'FFD9** **I/O ports**

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

PDR7—Port data register 7 **H'FFDA** **I/O ports**

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	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDR8—Port data register 8 **H'FFDB** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$P8_7$	$P8_6$	$P8_5$	$P8_4$	$P8_3$	$P8_2$	$P8_1$	$P8_0$
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

PDR9—Port data register 9 **H'FFDC** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	$P9_4$	$P9_3$	$P9_2$	$P9_1$	$P9_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

PDRB—Port data register B **H'FFDD** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	PB_7	PB_6	PB_5	PB_4	PB_3	PB_2	PB_1	PB_0
Initial value								
Read/Write	R	R	R	R	R	R	R	R

PCR1—Port control register 1 **H'FFE4** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR1_7$	$PCR1_6$	$PCR1_5$	$PCR1_4$	$PCR1_3$	$PCR1_2$	$PCR1_1$	$PCR1_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 1 input/output select

0	Input pin
1	Output pin

PCR2—Port control register 2 **H'FFE5** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR2_7$	$PCR2_6$	$PCR2_5$	$PCR2_4$	$PCR2_3$	$PCR2_2$	$PCR2_1$	$PCR2_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 2 input/output select

0	Input pin
1	Output pin

PCR3—Port control register 3 **H'FFE6** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	$PCR3_5$	$PCR3_4$	$PCR3_3$	$PCR3_2$	$PCR3_1$	$PCR3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

PCR5—Port control register 5 **H'FFE8** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR5_7$	$PCR5_6$	$PCR5_5$	$PCR5_4$	$PCR5_3$	$PCR5_2$	$PCR5_1$	$PCR5_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 5 input/output select

0	Input pin
1	Output pin

PCR6—Port control register 6 **H'FFE9** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR6_7$	$PCR6_6$	$PCR6_5$	$PCR6_4$	$PCR6_3$	$PCR6_2$	$PCR6_1$	$PCR6_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 6 input/output select

0	Input pin
1	Output pin

PCR7—Port control register 7 **H'FFEA** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR7_7$	$PCR7_6$	$PCR7_5$	$PCR7_4$	$PCR7_3$	$PCR7_2$	$PCR7_1$	$PCR7_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 7 input/output select

0	Input pin
1	Output pin

PCR8—Port control register 8 **H'FFEB** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PCR8_7$	$PCR8_6$	$PCR8_5$	$PCR8_4$	$PCR8_3$	$PCR8_2$	$PCR8_1$	$PCR8_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Port 8 input/output select

0	Input pin
1	Output pin

PCR9—Port control register 9 **H'FFEC** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	$PCR9_4$	$PCR9_3$	$PCR9_2$	$PCR9_1$	$PCR9_0$
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	—	W	W	W	W	W

Port 9 input/output select

0	Input pin
1	Output pin

PUCR1—Port pull-up control register 1 **H'FFED** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PUCR1_7$	$PUCR1_6$	$PUCR1_5$	$PUCR1_4$	$PUCR1_3$	$PUCR1_2$	$PUCR1_1$	$PUCR1_0$
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

PUCR3—Port pull-up control register 3 **H'FFEE** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	$PUCR3_5$	$PUCR3_4$	$PUCR3_3$	$PUCR3_2$	$PUCR3_1$	$PUCR3_0$
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

PUCR5—Port pull-up control register 5 **H'FFEF** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	$PUCR5_7$	$PUCR5_6$	$PUCR5_5$	$PUCR5_4$	$PUCR5_3$	$PUCR5_2$	$PUCR5_1$	$PUCR5_0$
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

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	LSON	—	MA1	MA0
Initial value	0	0	0	0	0	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W

Active (medium-speed) mode clock select

0	0	$ø_{osc}$/16
0	1	$ø_{osc}$/32
1	0	$ø_{osc}$/64
1	1	$ø_{osc}$/128

Low speed on flag

0	The CPU operates on the system clock (ø)
1	The CPU operates on the subclock ($ø_{SUB}$)

Standby timer select

0	0	0	Wait time = 8,192 states
		1	Wait time = 16,384 states
	1	0	Wait time = 32,768 states
		1	Wait time = 65,536 states
1	*	*	Wait time = 131,072 states

Software standby

0	• When a SLEEP instruction is executed in active mode, a transition is made to sleep mode • When a SLEEP instruction is executed in subactive mode, a transition is made to subsleep mode
1	• When a SLEEP instruction is executed in active mode, a transition is made to standby mode or watch mode • When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode

Note: * Don’t care

Bit	7	6	5	4	3	2	1	0
	—	—	—	NESEL	DTON	MSON	SA1	SA0
Initial value	1	1	1	0	0	0	0	0
Read/Write	—	—	—	R/W	R/W	R/W	R/W	R/W

Subactive mode clock select

0	0	$\phi_W/8$
	1	$\phi_W/4$
1	*	$\phi_W/2$

Medium speed on flag

0	• Operates in active (high-speed) mode after exit from standby, watch, or sleep mode • Operates in sleep (high-speed) mode if a SLEEP instruction is executed in active mode
1	• Operates in active (medium-speed) mode after exit from standby, watch, or sleep mode • Operates in sleep (medium-speed) mode if a SLEEP instruction is executed in active mode

Direct transfer on flag

0	• When a SLEEP instruction is executed in active mode, a transition is made to standby mode, watch mode, or sleep mode • When a SLEEP instruction is executed in subactive mode, a transition is made to watch mode or subsleep mode
1	• When a SLEEP instruction is executed in active (high-speed) mode, a direct transition is made to active (medium-speed) mode if SSBY = 0, MSON = 1, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1 • When a SLEEP instruction is executed in active (medium-speed) mode, a direct transition is made to active (high-speed) mode if SSBY = 0, MSON = 0, and LSON = 0, or to subactive mode if SSBY = 1, TMA3 = 1, and LSON = 1 • When a SLEEP instruction is executed in subactive mode, a direct transition is made to active (high-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 0, or to active (medium-speed) mode if SSBY = 1, TMA3 = 1, LSON = 0, and MSON = 1

Noise elimination sampling frequency select

0	Sampling rate is $\phi_{OSC}/16$
1	Sampling rate is $\phi_{OSC}/4$

Note: * Don't care

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	IEG3	IEG2	IEG1	IEG0
Initial value	0	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

IRQ_0 edge select

0	Falling edge of $\overline{IRQ_0}$ pin input is detected
1	Rising edge of $\overline{IRQ_0}$ pin input is detected

IRQ_1 edge select

0	Falling edge of $\overline{IRQ_1}$ pin input is detected
1	Rising edge of $\overline{IRQ_1}$ pin input is detected

IRQ_2 edge select

0	Falling edge of $\overline{IRQ_2}$ pin input is detected
1	Rising edge of $\overline{IRQ_2}$ pin input is detected

IRQ_3 edge select

0	Falling edge of $\overline{IRQ_3}$ pin input is detected
1	Rising edge of $\overline{IRQ_3}$ pin input is detected

Bit	7	6	5	4	3	2	1	0
	INTEG7	INTEG6	INTEG5	INTEG4	INTEG3	INTEG2	INTEG1	INTEG0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

INT_4 to INT_0 edge select

0	Falling edge of $\overline{INT}_n$ pin input is detected
1	Rising edge of $\overline{INT}_n$ pin input is detected

(n = 4 to 0)

INT_5 edge select

0	Falling edge of $\overline{INT}_5$ and $\overline{ADTRG}$ pin input is detected
1	Rising edge of $\overline{INT}_5$ and $\overline{ADTRG}$ pin input is detected

(n = 5 to 0)

INT_6 edge select

0	Falling edge of $\overline{INT}_6$ and TMIB pin input is detected
1	Rising edge of $\overline{INT}_6$ and TMIB pin input is detected

INT_7 edge select

0	Falling edge of $\overline{INT}_7$ and TMIY pin input is detected
1	Rising edge of $\overline{INT}_7$ and TMIY pin input is detected

Bit	7	6	5	4	3	2	1	0
	IENTB1	IENTA	—	—	IEN3	IEN2	IEN1	IEN0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W

IRQ_3 to IRQ_0 interrupt enable

0	Disables IRQ3 to IRQ_0 interrupt requests
1	Enables IRQ3 to IRQ_0 interrupt requests

Timer A interrupt enable

0	Disables timer A interrupt requests
1	Enables timer A interrupt requests

Timer B1 interrupt enable

0	Disables timer B1 interrupt requests
1	Enables timer B1 interrupt requests

IENR2—Interrupt enable register 2 **H'FFF5** **System control**

Bit	7	6	5	4	3	2	1	0
	IENDT	IENAD	—	IENS1	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	—	R/W	—	—	—	—

SCI1 interrupt enable

0	Disables SCI1 interrupt requests
1	Enables SCI1 interrupt requests

A/D converter interrupt enable

0	Disables A/D converter interrupt requests
1	Enables A/D converter interrupt requests

Direct transfer interrupt enable

0	Disables direct transfer interrupt requests
1	Enables direct transfer interrupt requests

IENR3—Interrupt enable register 3 **H'FFF6** **System control**

Bit	7	6	5	4	3	2	1	0
	INTEN7	INTEN6	INTEN5	INTEN4	INTEN3	INTEN2	INTEN1	INTEN0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

INT_7 to INT_0 interrupt enable

0	Disables INT_7 to INT_0 interrupt requests
1	Enables INT_7 to INT_0 interrupt requests

Bit	7	6	5	4	3	2	1	0
	IRRTB1	IRRTA	—	—	IRRI3	IRRI2	IRRI1	IRRI0
Initial value	0	0	0	1	0	0	0	0
Read/Write	R/W*	R/W*	—	—	R/W*	R/W*	R/W*	R/W*

IRQ_3 to IRQ_0 interrupt request flag

0	[Clearing condition] When IRRIn = 1, it is cleared by writing 0
1	[Setting condition] When pin $\overline{IRQn}$ is set for interrupt input and the designated signal edge is input

(n = 3 to 0)

Timer A interrupt request flag

0	[Clearing condition] When IRRTA = 1, it is cleared by writing 0
1	[Setting condition] When timer counter A overflows from H'FF to H'00

Timer B1 interrupt request flag

0	[Clearing condition] When IRRTB1 = 1, it is cleared by writing 0
1	[Setting condition] When timer counter B1 overflows from H'FF to H'00

Note: * Only a write of 0 for flag clearing is possible.

Bit	7	6	5	4	3	2	1	0
	IRRDT	IRRAD	—	IRRS1	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W*	R/W*	—	R/W*	—	—	—	—

SCI1 interrupt request flag

0	[Clearing condition] When IRRS1 = 1, it is cleared by writing 0
1	[Setting condition] When an SCI1 transfer is completed

A/D converter interrupt request flag

0	[Clearing condition] When IRRAD = 1, it is cleared by writing 0
1	[Setting condition] When A/D conversion is completed and ADSF is cleared to 0 in ADSR

Direct transfer interrupt request flag

0	[Clearing condition] When IRRDT = 1, it is cleared by writing 0
1	[Setting condition] A SLEEP instruction is executed when DTON = 1 and a direct transfer is made

Note: * Only a write of 0 for flag clearing is possible.

	7	6	5	4	3	2	1	0
	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0
tial value	0	0	0	0	0	0	0	0
ead/Write	R/W*	R/W*	R/W*	R/W	R/W*	R/W*	R/W*	R/W*

INT_7 to INT_0 interrupt request flag

0	[Clearing condition] When $INTF_n$ = 1, it is cleared by writing 0
1	[Setting condition] When the designated signal edge is input at pin $\overline{INT_n}$

(n = 7 to 0)

te: * Only a write of 0 for flag clearing is possible.

Bit	7	6	5	4	3	2	1	0
	IRQ3	IRQ2	IRQ1	PWM	—	—	—	TMOW
Initial value	0	0	0	0	0	1	0	0
Read/Write	R/W	R/W	R/W	R/W	—	—	—	R/W

$P1_0$/TMOW pin function switch

0	Functions as $P1_0$ I/O pin
1	Functions as TMOW output pin

$P1_4$/PWM pin function switch

0	Functions as $P1_4$ I/O pin
1	Functions as PWM output pin

$P1_5$/$\overline{IRQ_1}$ pin function switch

0	Functions as $P1_5$ I/O pin
1	Functions as $\overline{IRQ_1}$ input pin

$P1_6$/$\overline{IRQ_2}$ pin function switch

0	Functions as $P1_6$ I/O pin
1	Functions as $\overline{IRQ_2}$ input pin

$P1_7$/$\overline{IRQ_3}$/TRGV pin function switch

0	Functions as $P1_7$ I/O pin
1	Functions as $\overline{IRQ_3}$/TRGV input pin

PMR3—Port mode register 3 **H'FFFD** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SO1	SI1	SCK1
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	R/W	R/W	R/W

$P3_0/SCK_1$ pin function switch

0	Functions as $P3_0$ I/O pin
1	Functions as SCK_1 I/O pin

$P3_1/SI_1$ pin function switch

0	Functions as $P3_1$ I/O pin
1	Functions as SI_1 input pin

$P3_2/SO_1$ pin function switch

0	Functions as $P3_2$ I/O pin
1	Functions as SO_1 output pin

PMR7—Port mode register 7 **H'FFFF** **I/O ports**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TXD	—	POF1
Initial value	1	1	1	1	1	0	0	0
Read/Write	—	—	—	—	—	R/W	—	R/W

$P3_2/SO_1$ pin PMOS control

0	CMOS output
1	NMOS open-drain output

$P2_2$/TXD pin function switch (TXD)

0	Functions as $P2_2$ I/O pin
1	Functions as TXD output pin

Appendix C I/O Port Block Diagrams

C.1 Block Diagrams of Port 1

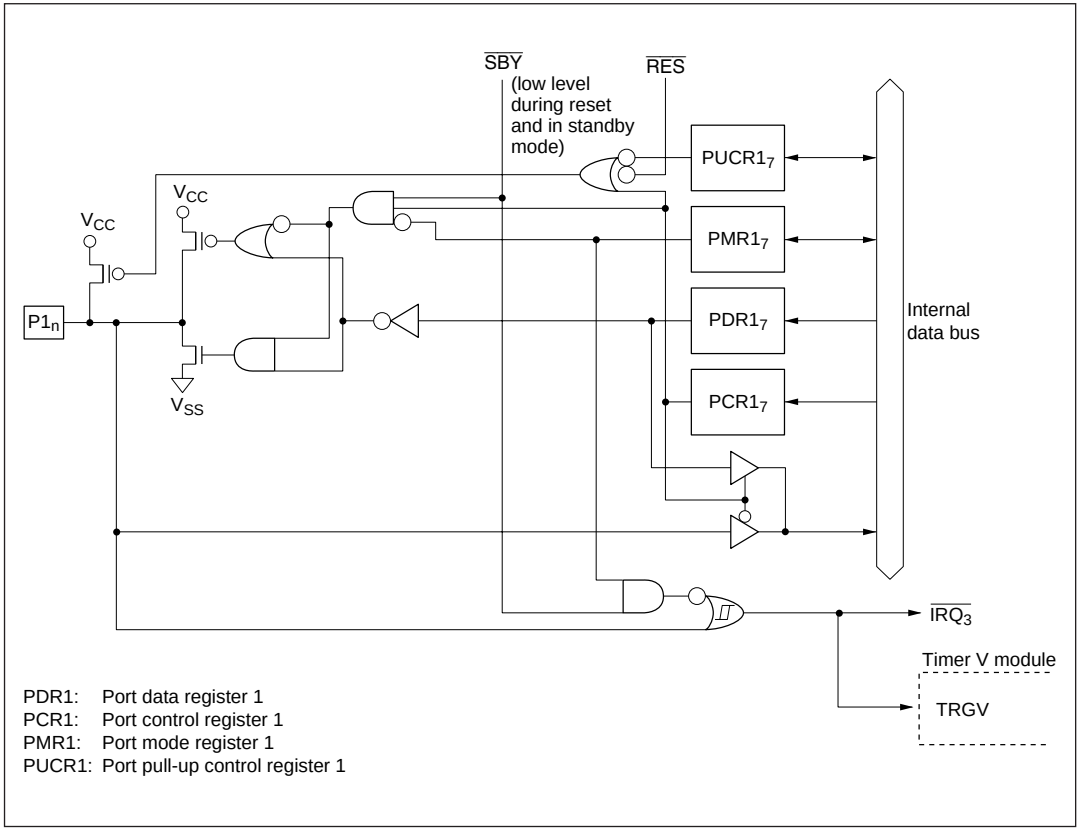
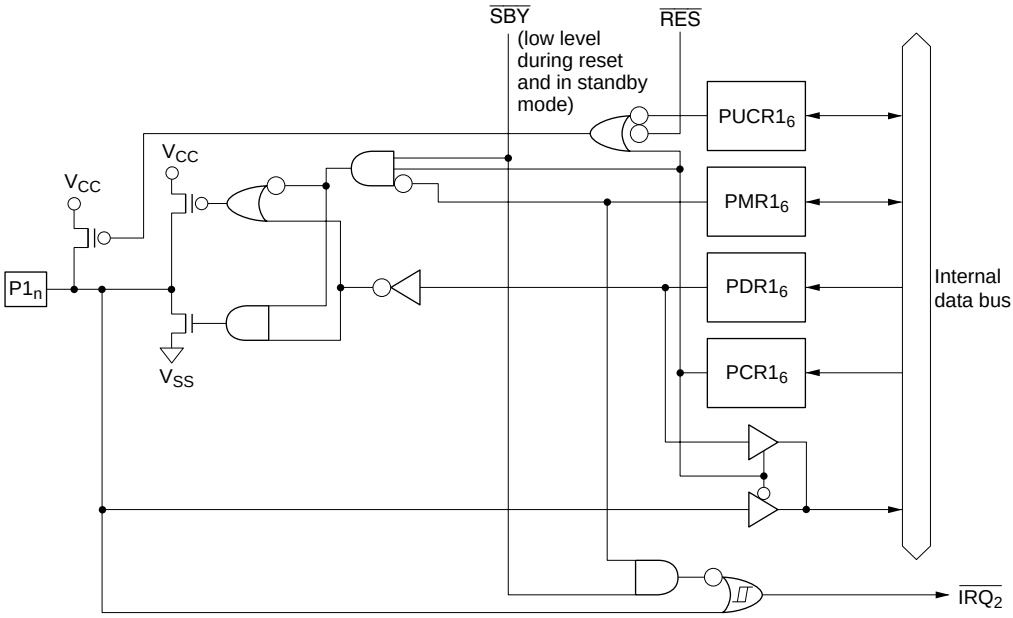
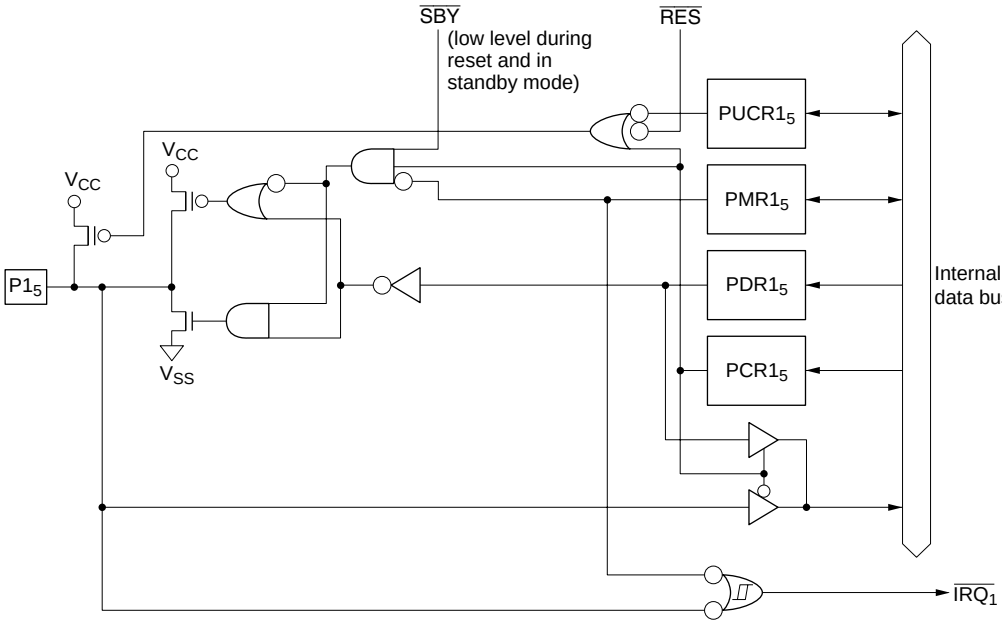


Figure C-1 (a) Port 1 Block Diagram (Pin P1$_7$)



PDR1: Port data register 1
PCR1: Port control register 1
PMR1: Port mode register 1
PUCR1: Port pull-up control register 1

Figure C-1 (b) Port 1 Block Diagram (Pin P1$_6$)



PDR1: Port data register 1
PCR1: Port control register 1
PMR1: Port mode register 1
PUCR1: Port pull-up control register 1

Figure C-1 (c) Port 1 Block Diagram (Pin $P1_5$)

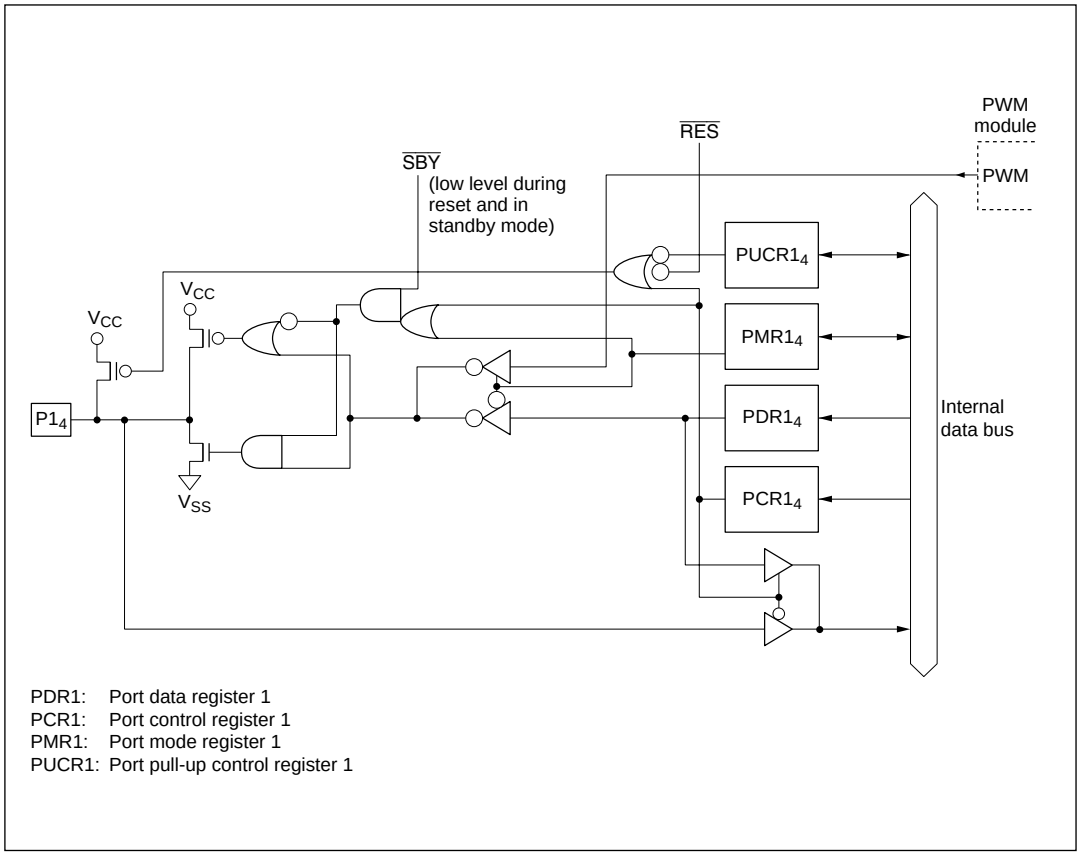
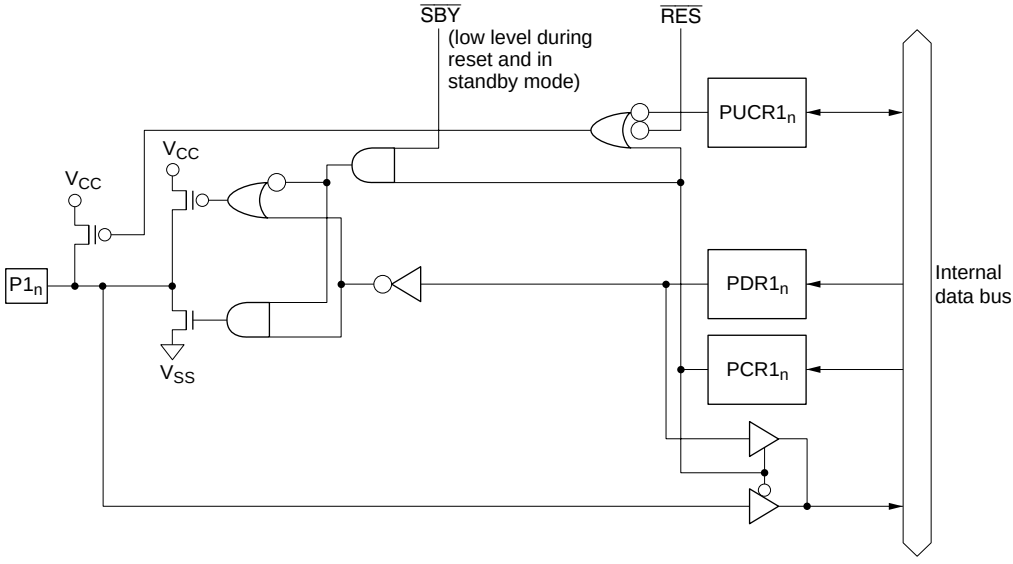


Figure C-1 (d) Port 1 Block Diagram (Pin $P1_4$)



PDR1: Port data register 1
PCR1: Port control register 1
PUCR1: Port pull-up control register 1

n= 3 to 1

Figure C-1 (e) Port 1 Block Diagram (Pins $P1_3$ to $P1_1$)

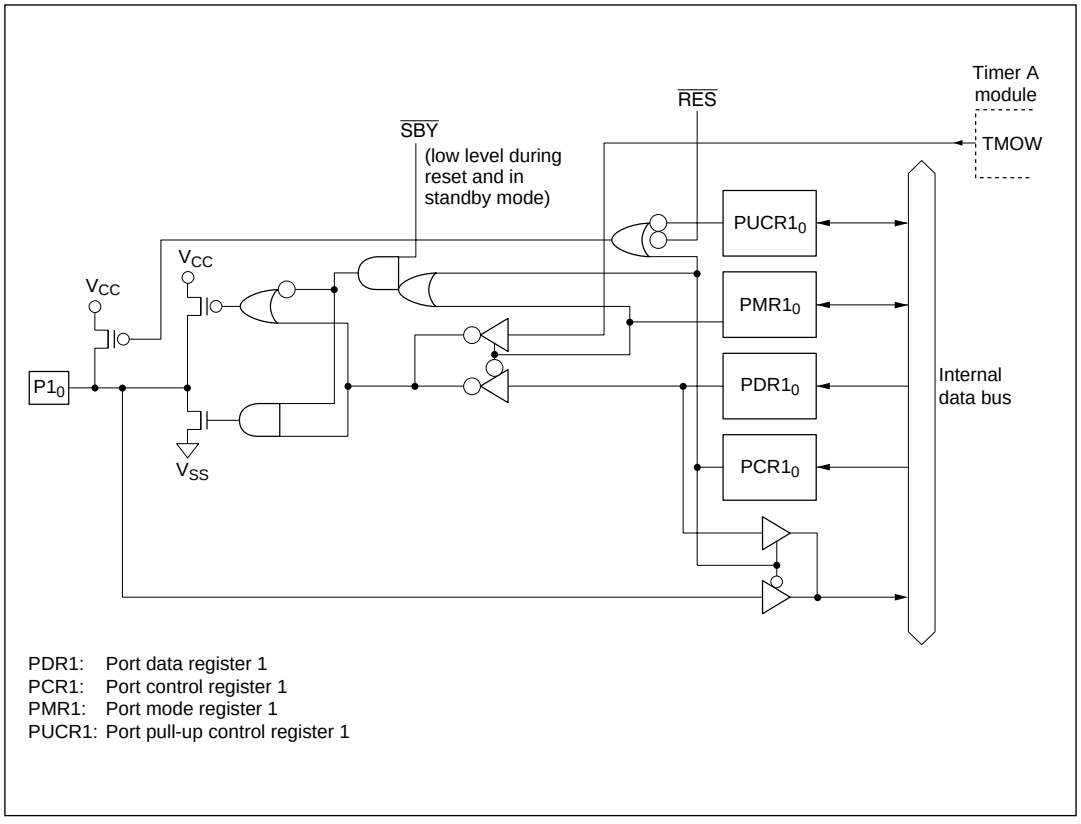


Figure C-1 (f) Port 1 Block Diagram (Pin $P1_0$)

C.2 Block Diagrams of Port 2

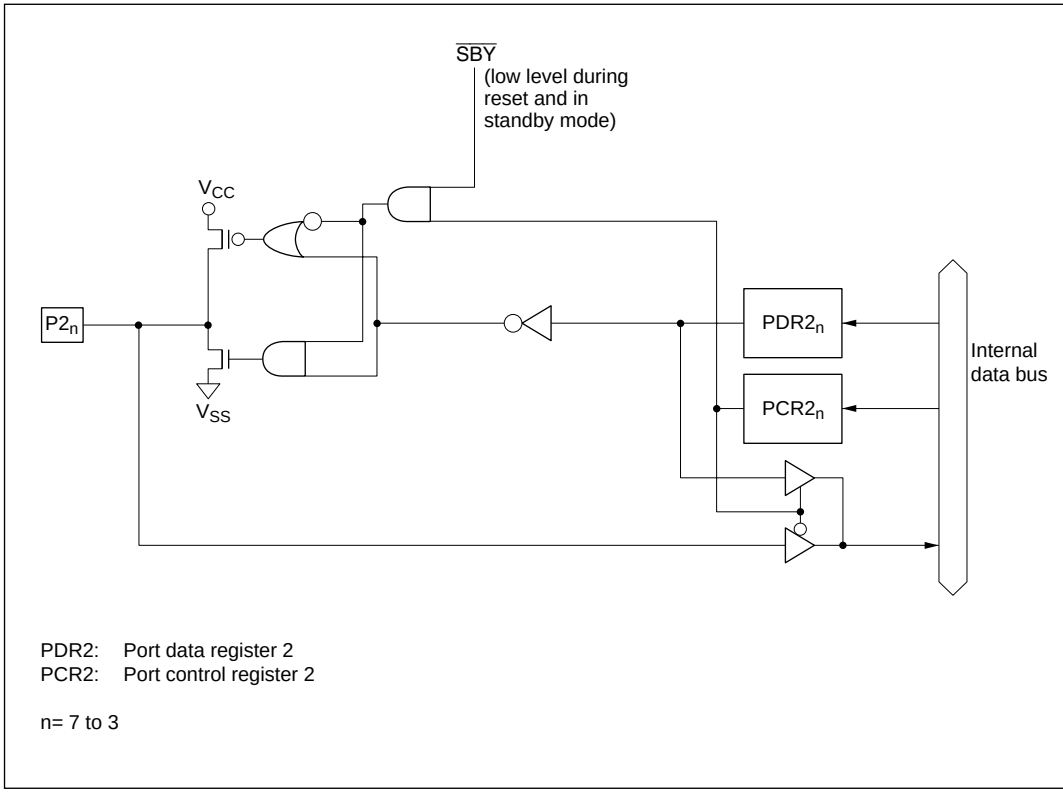


Figure C-2 (a) Port 2 Block Diagram (Pins $P2_7$ to $P2_3$)

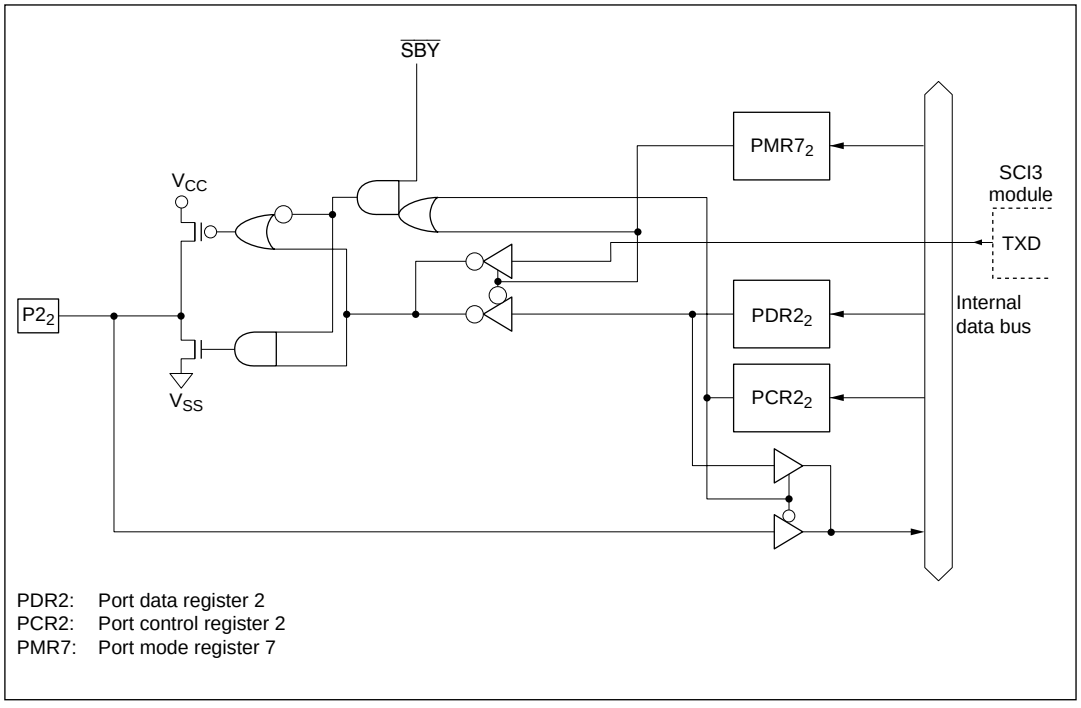


Figure C-2 (b) Port 2 Block Diagram (Pin $P2_2$)

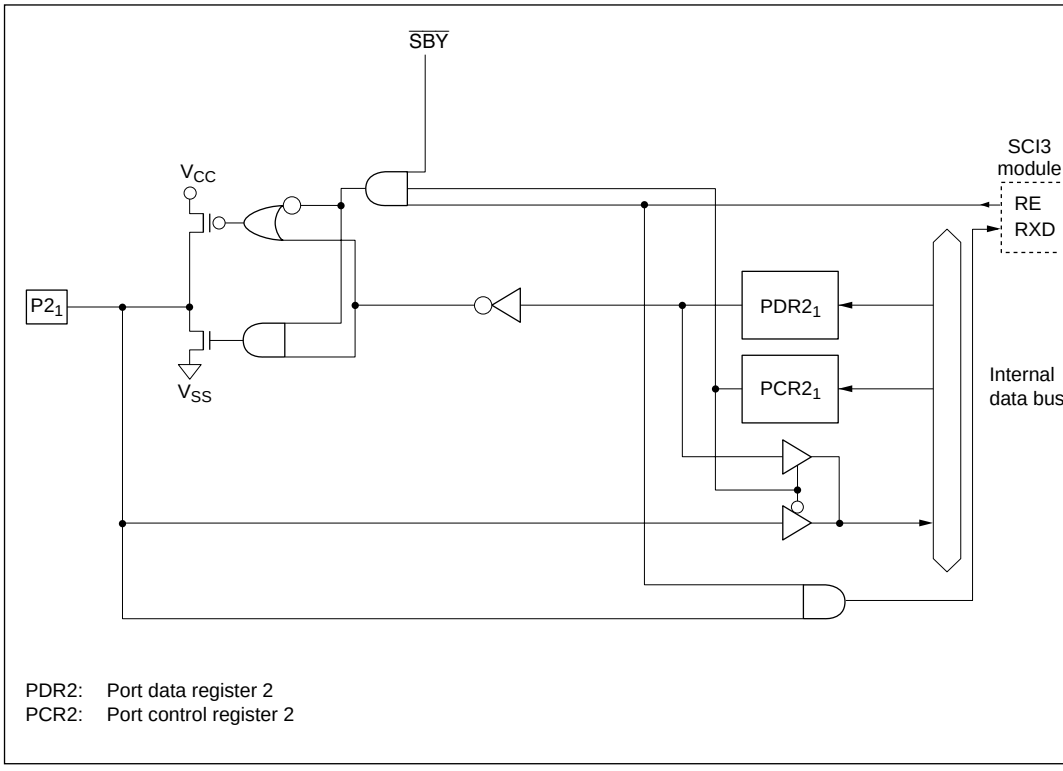
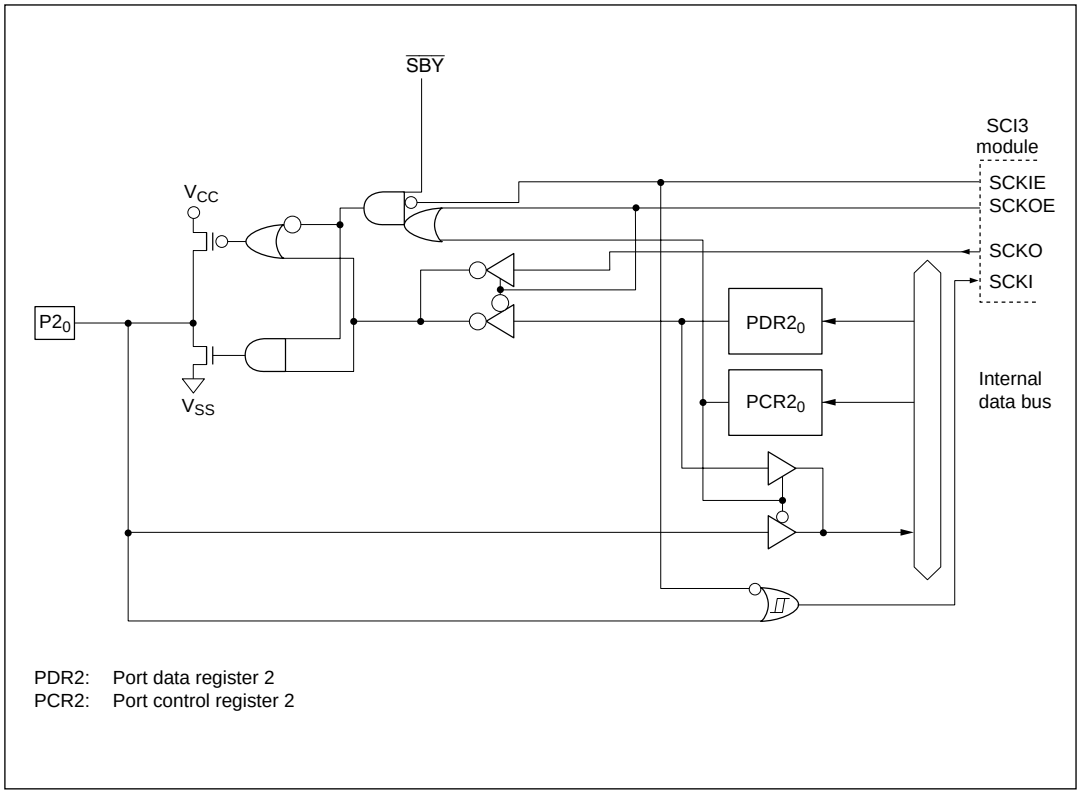


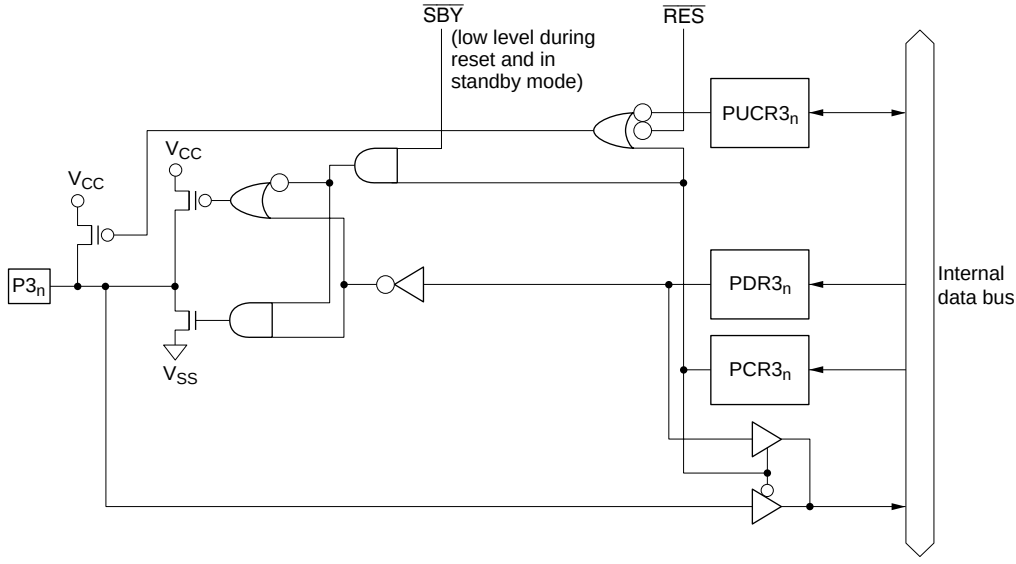
Figure C-2 (c) Port 2 Block Diagram (Pin $P2_1$)



PDR2: Port data register 2
PCR2: Port control register 2

Figure C-2 (d) Port 2 Block Diagram (Pin $P2_0$)

C.3 Block Diagrams of Port 3



PDR3: Port data register 3
PCR3: Port control register 3
PUCR3: Port pull-up control register 3

n= 5 to 3

Figure C-3 (a) Port 3 Block Diagram (Pins $P3_5$ to $P3_3$)

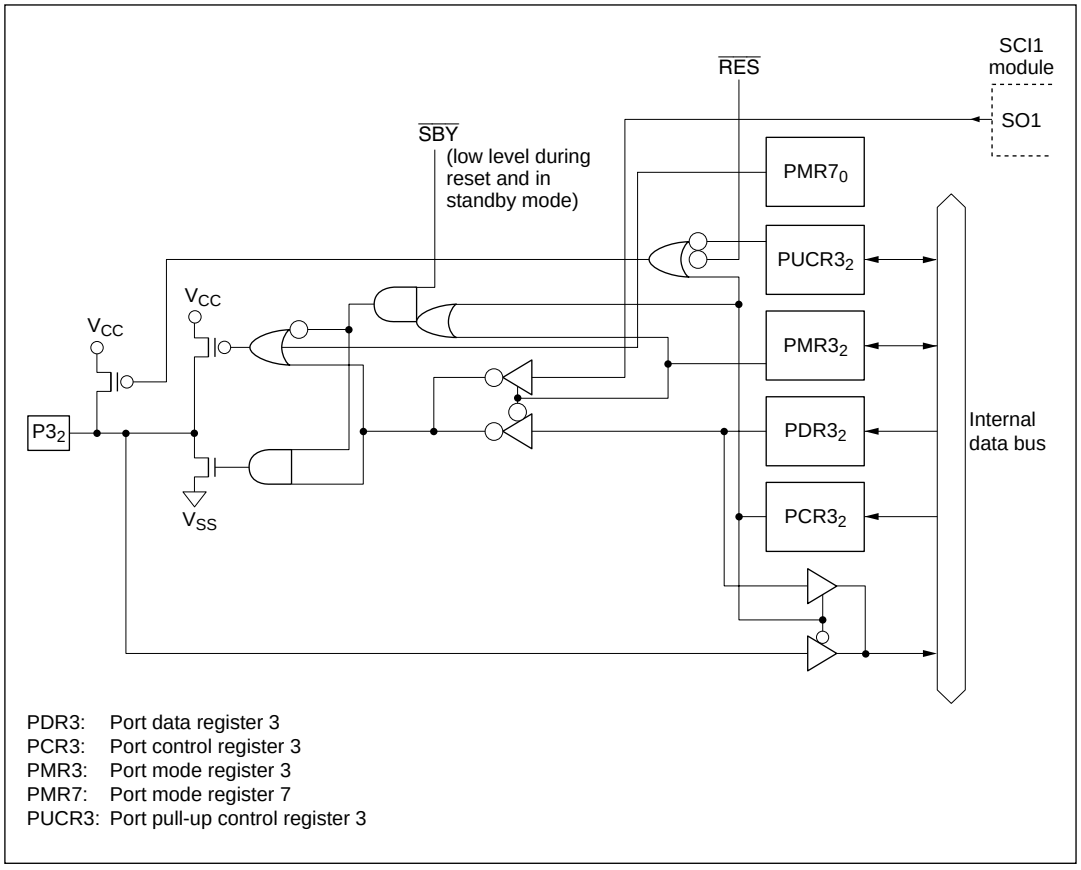
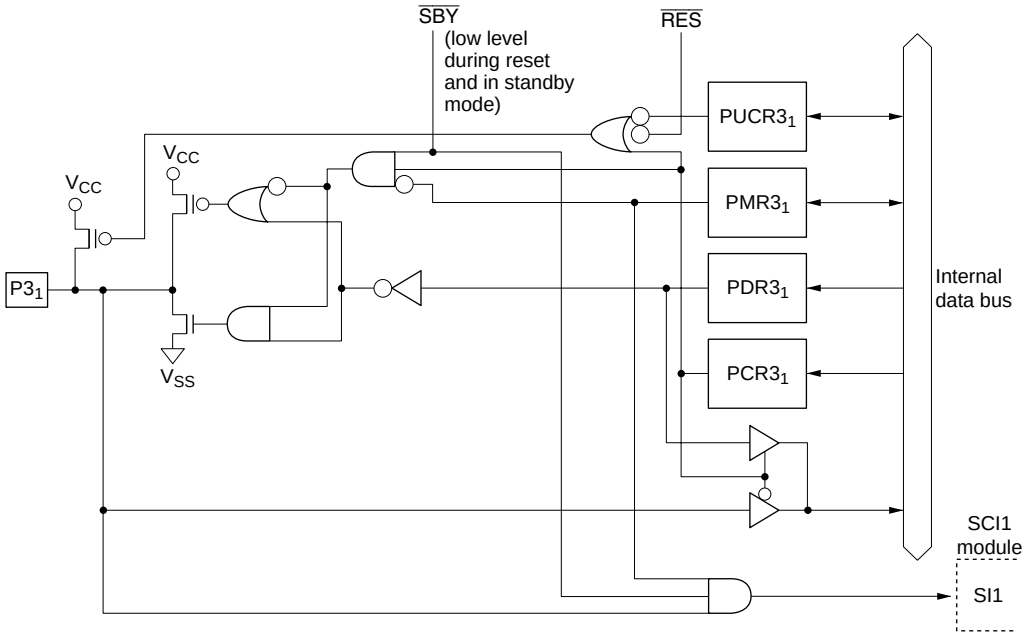


Figure C-3 (b) Port 3 Block Diagram (Pin P3$_2$)



PDR3: Port data register 3
PCR3: Port control register 3
PMR3: Port mode register 3
PUCR3: Port pull-up control register 3

Figure C-3 (c) Port 3 Block Diagram (Pin $P3_1$)

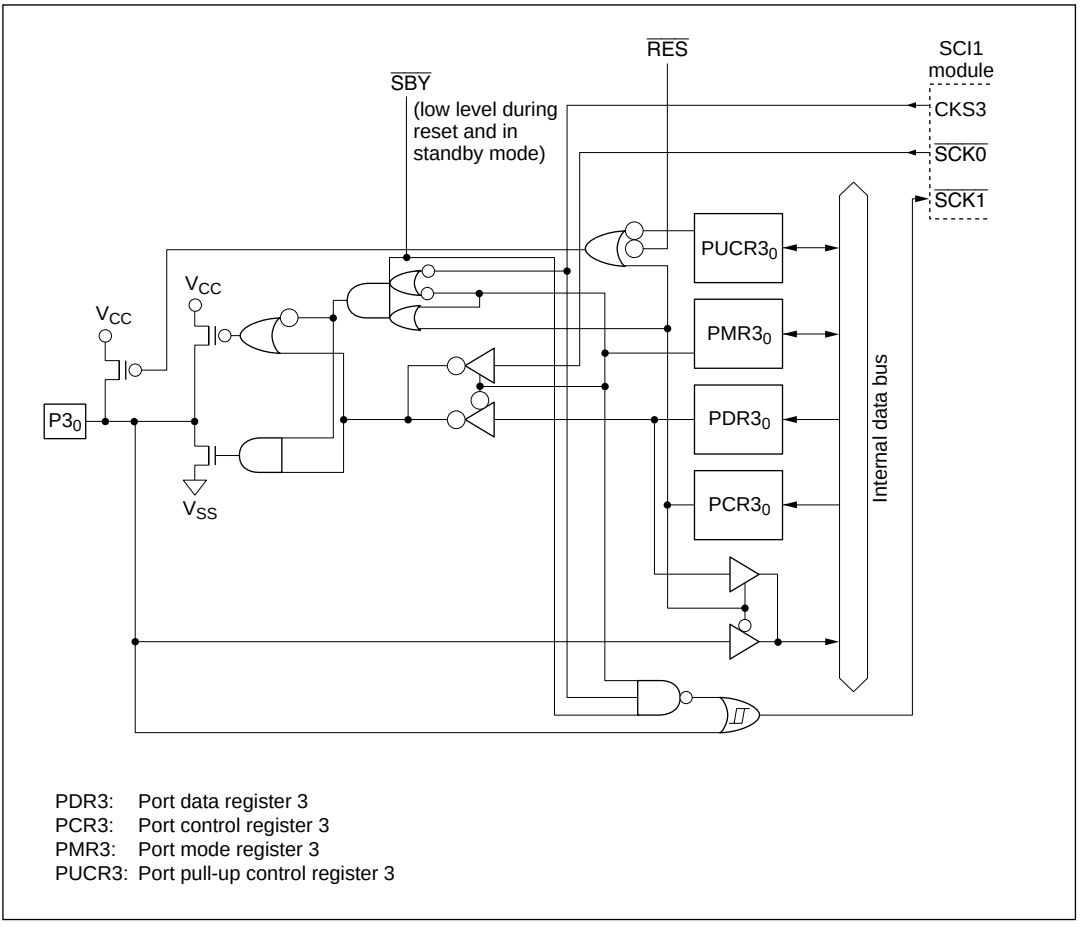


Figure C-3 (d) Port 3 Block Diagram (Pin $P3_0$)

C.4 Block Diagrams of Port 5

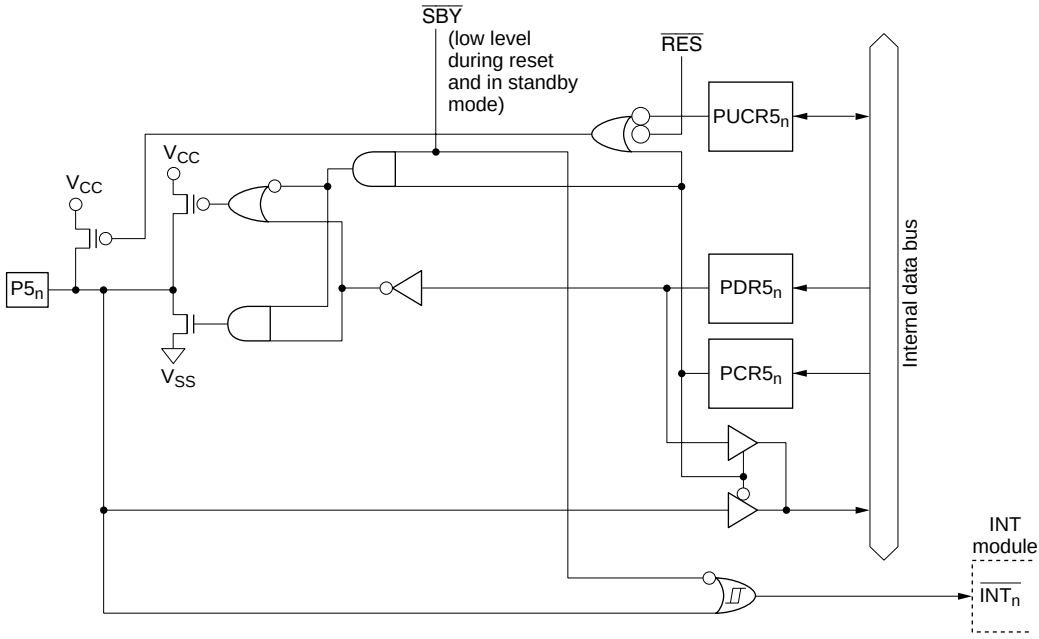


Figure C-4 (a) Port 5 Block Diagram (Pins $P5_7$ and $P5_4$ to $P5_0$)

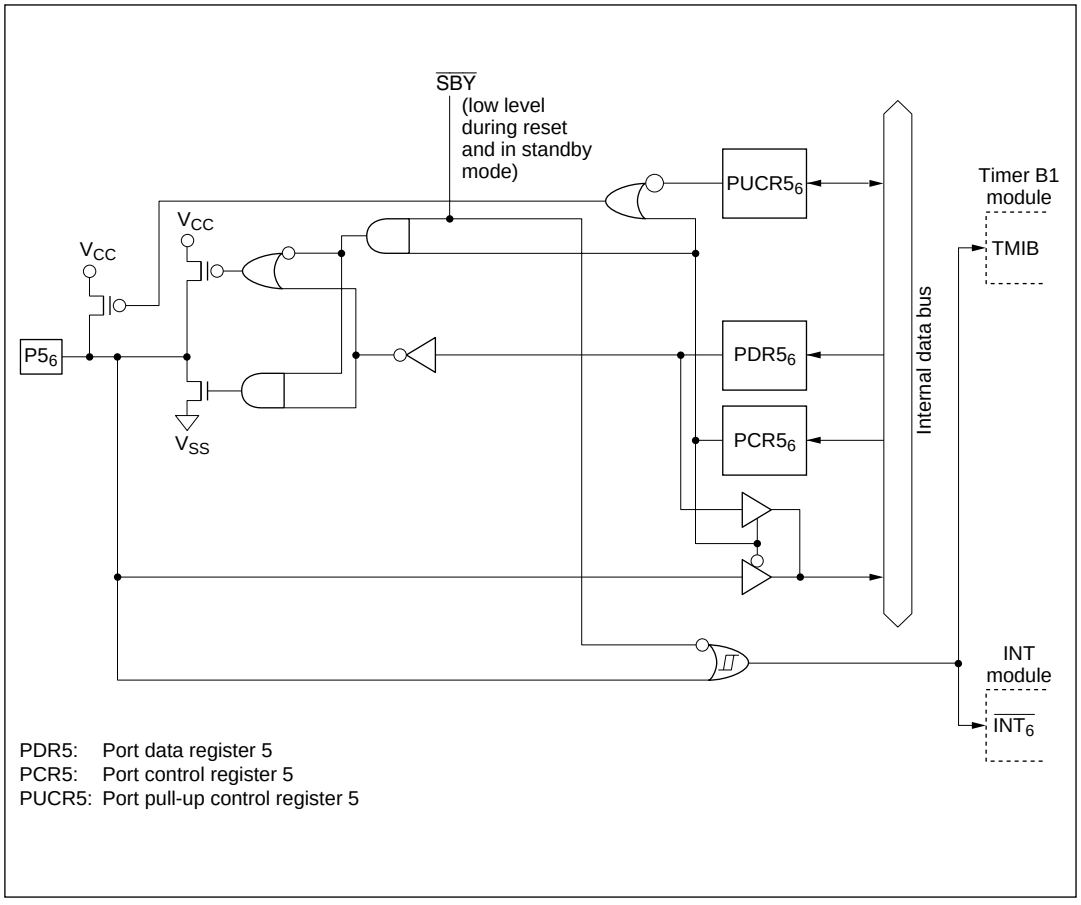


Figure C-4 (b) Port 5 Block Diagram (Pin P5$_6$)

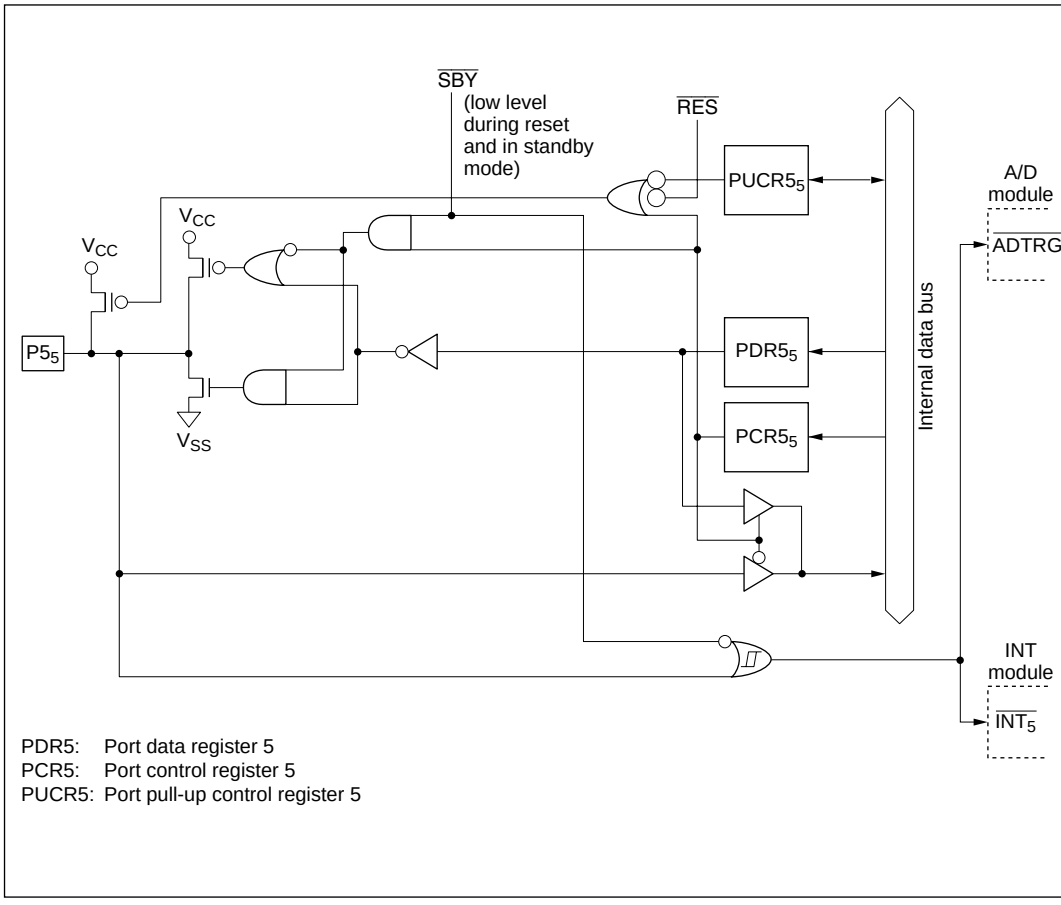


Figure C-4 (c) Port 5 Block Diagram (Pin $P5_5$)

C.5 Block Diagram of Port 6

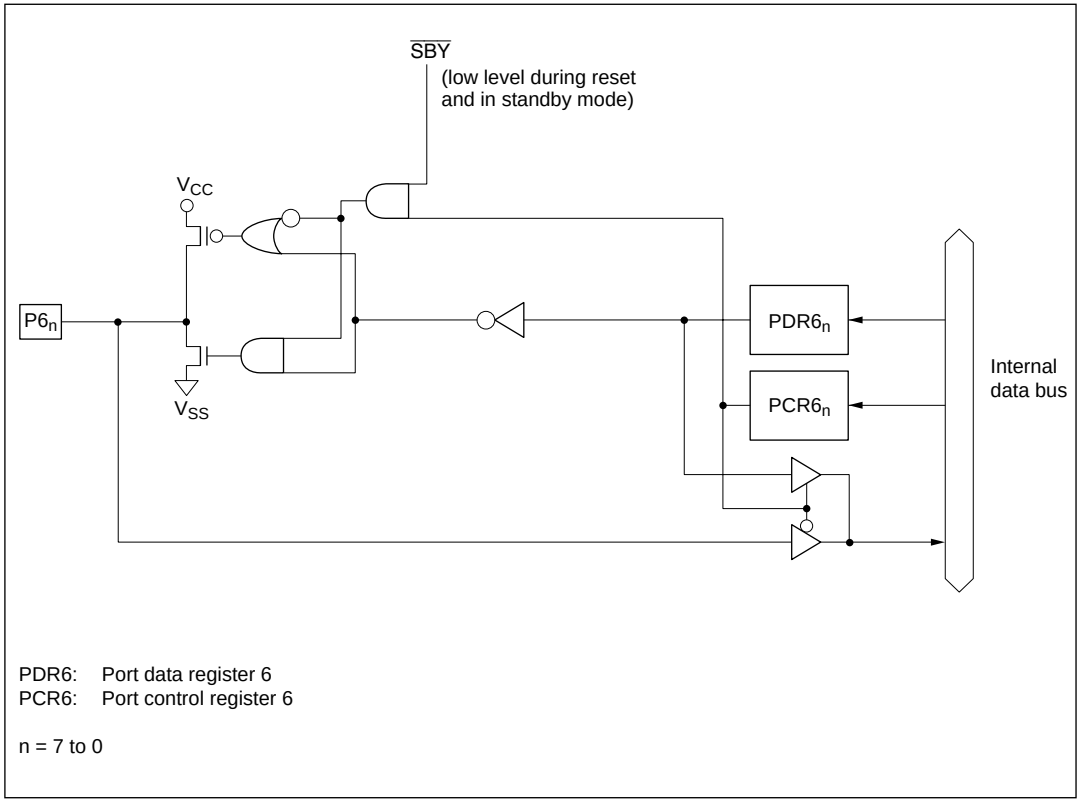


Figure C-5 Port 6 Block Diagram (Pins $P6_7$ to $P6_0$)

C.6 Block Diagrams of Port 7

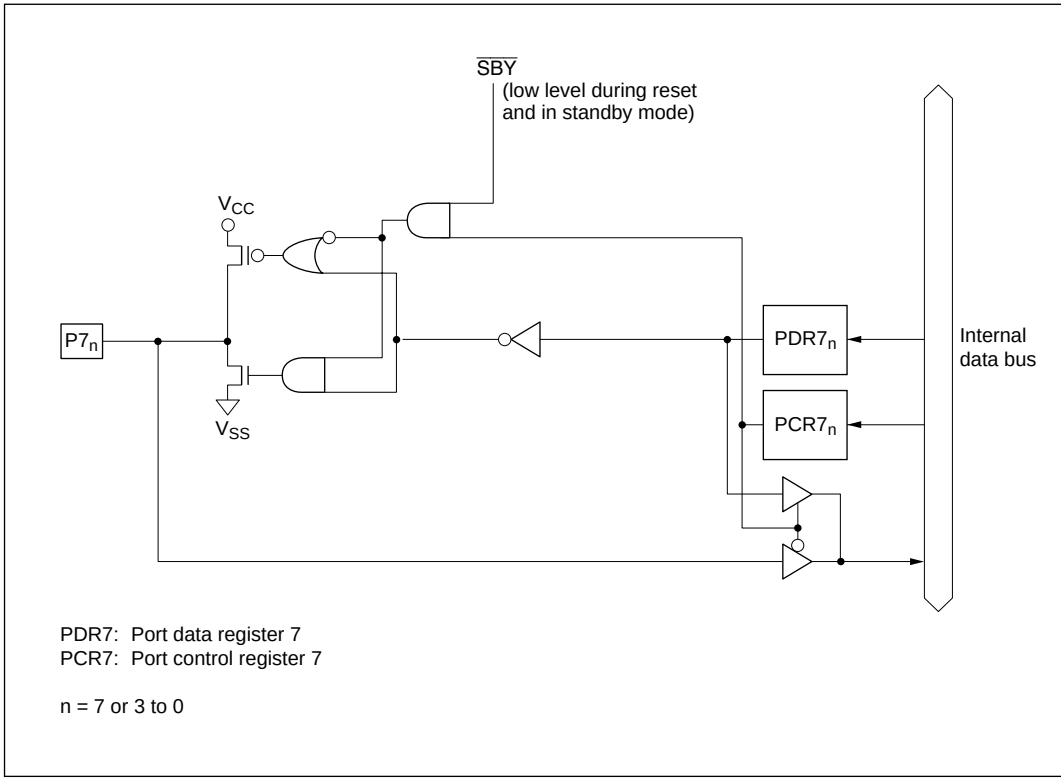


Figure C-6 (a) Port 7 Block Diagram (Pins $P7_7$ and $P7_3$ to $P7_0$)

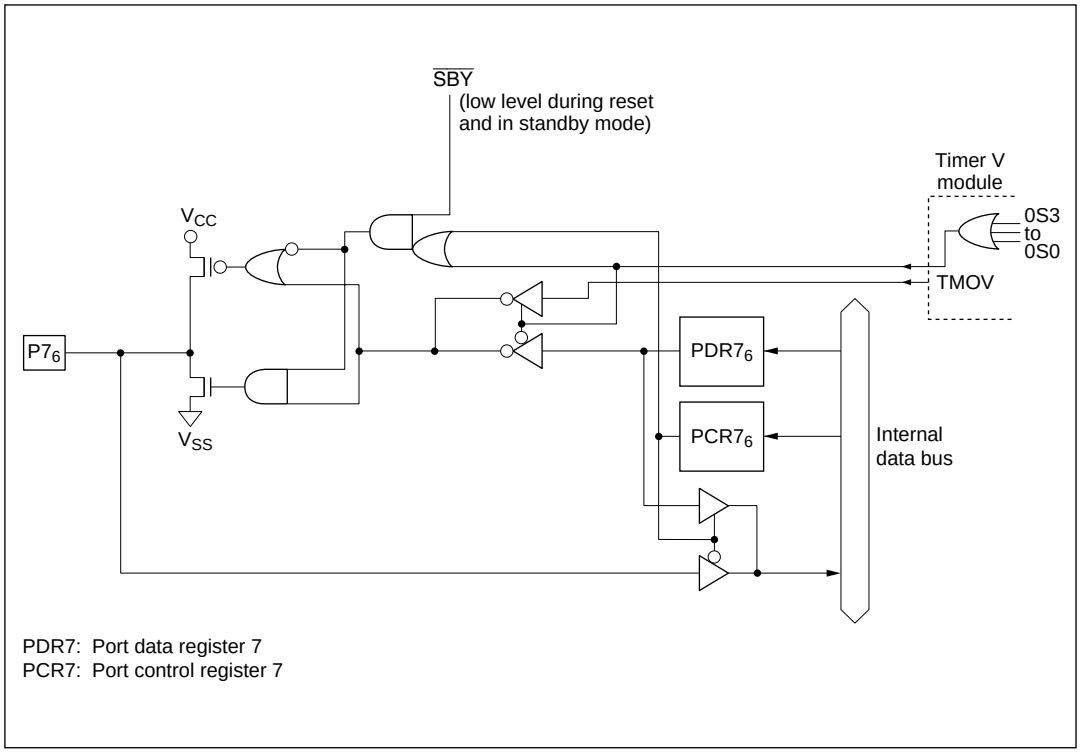


Figure C-6 (b) Port 7 Block Diagram (Pin $P7_6$)

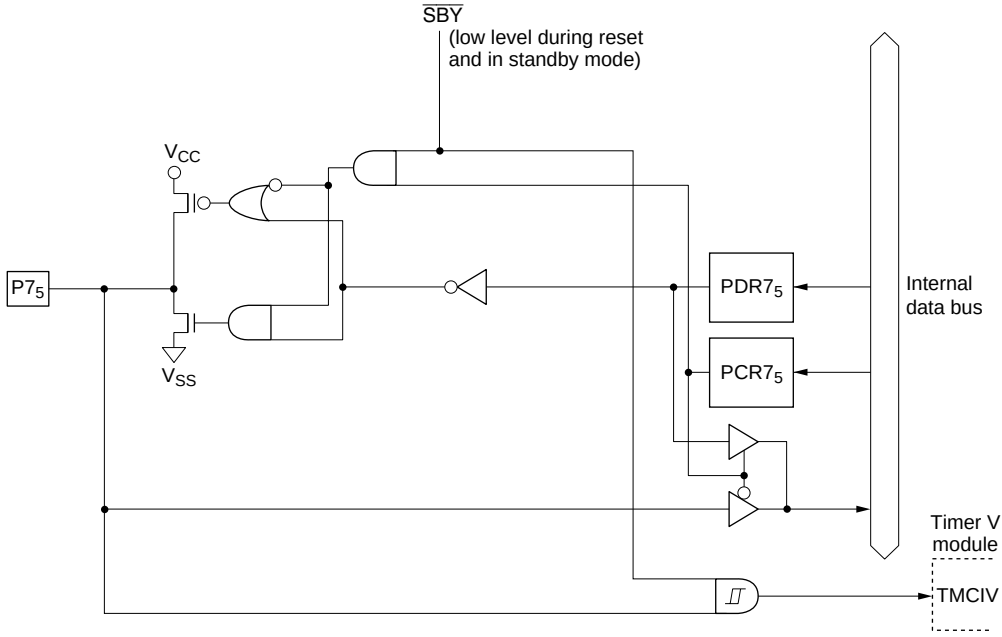


Figure C-6 (c) Port 7 Block Diagram (Pin $P7_5$)

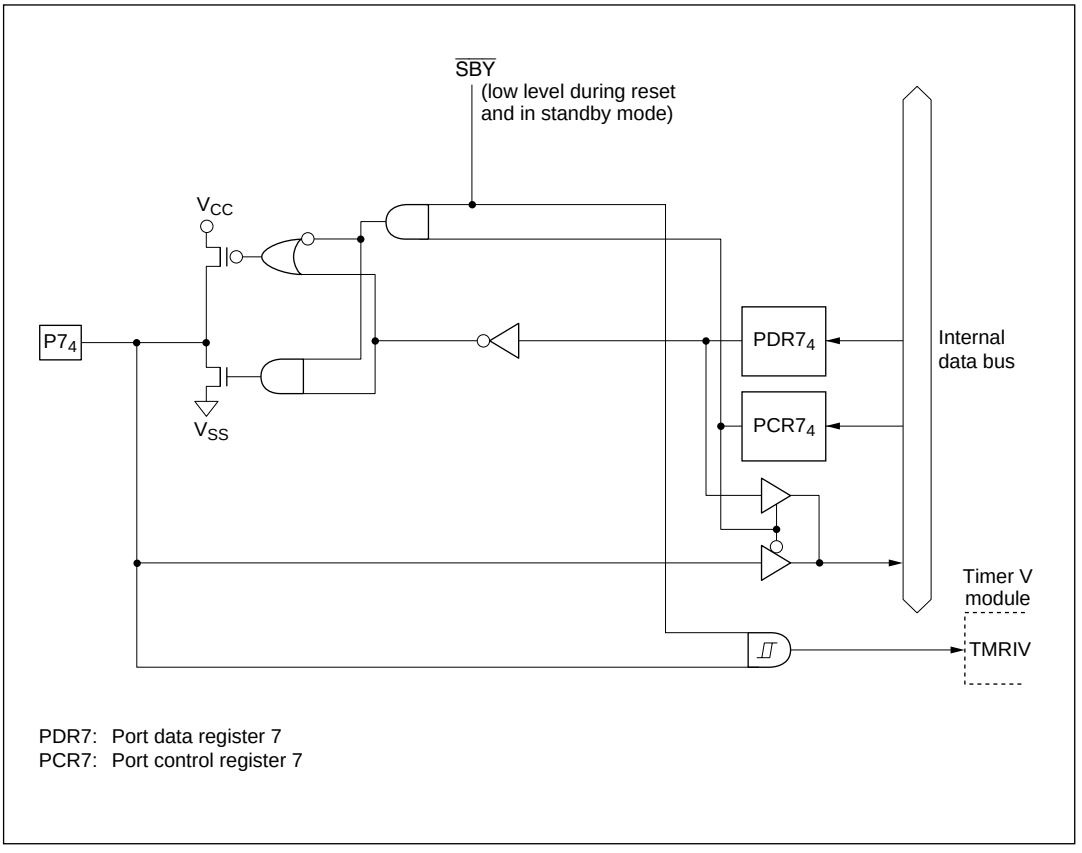


Figure C-6 (d) Port 7 Block Diagram (Pin $P7_4$)

C.7 Block Diagrams of Port 8

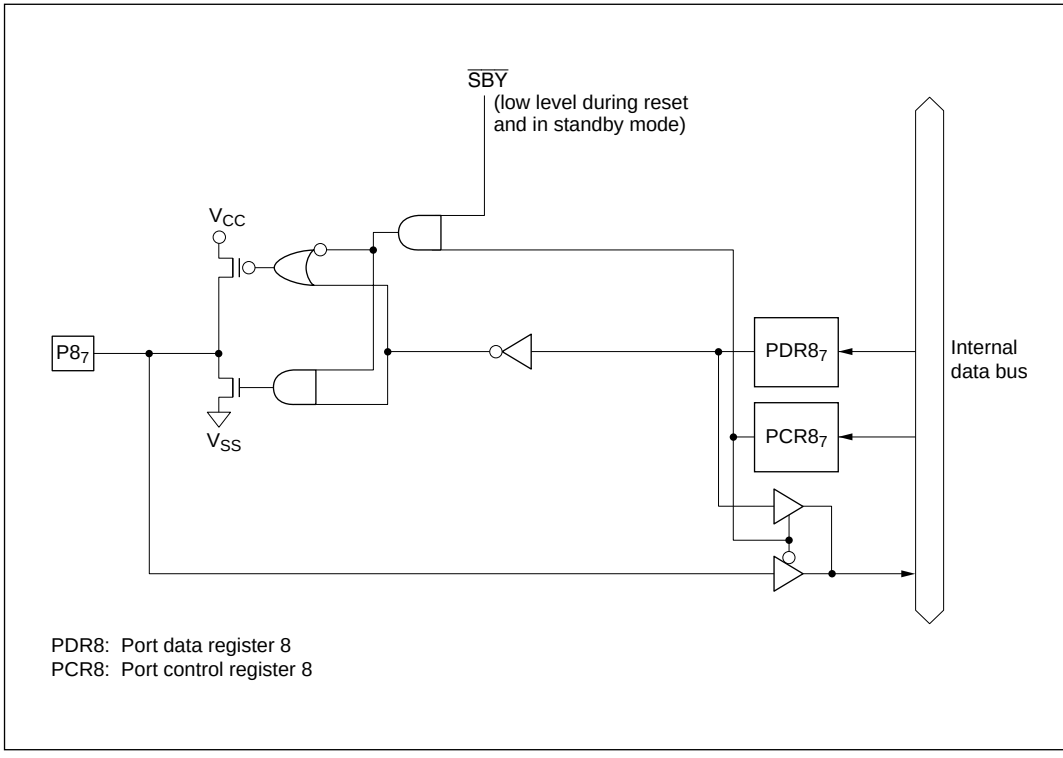


Figure C-7 (a) Port 8 Block Diagram (Pin $P8_7$)

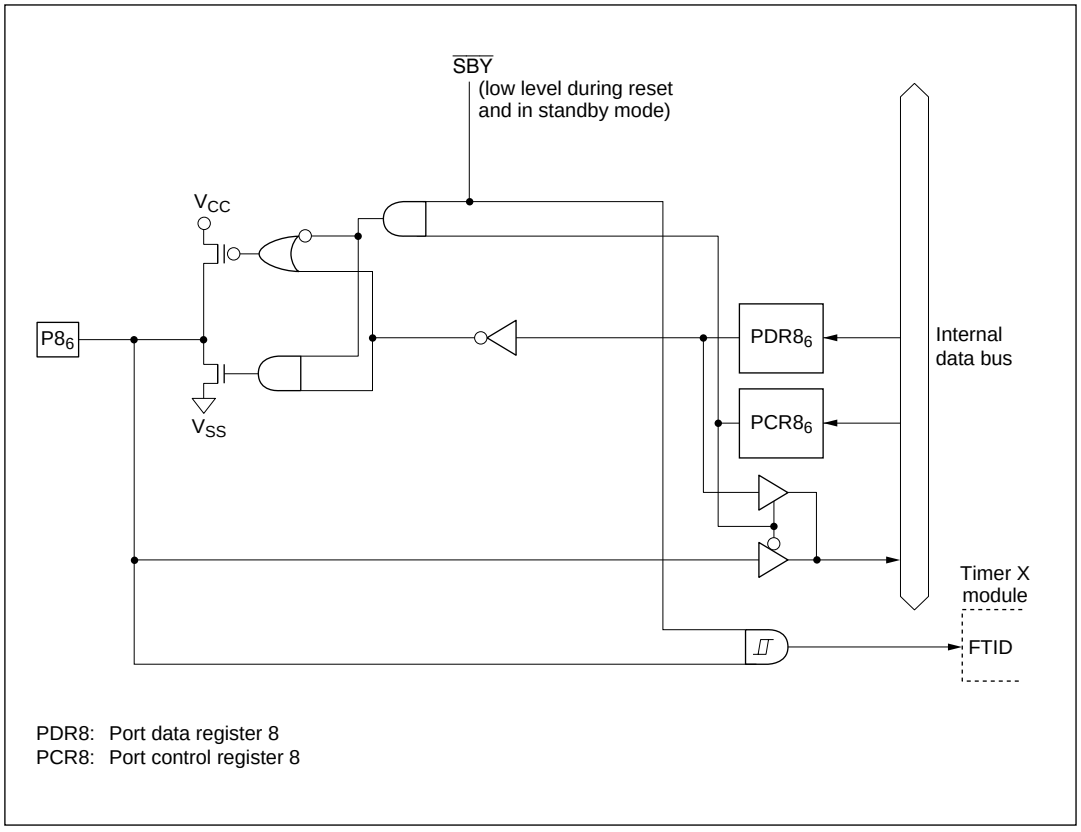


Figure C-7 (b) Port 8 Block Diagram (Pin $P8_6$)

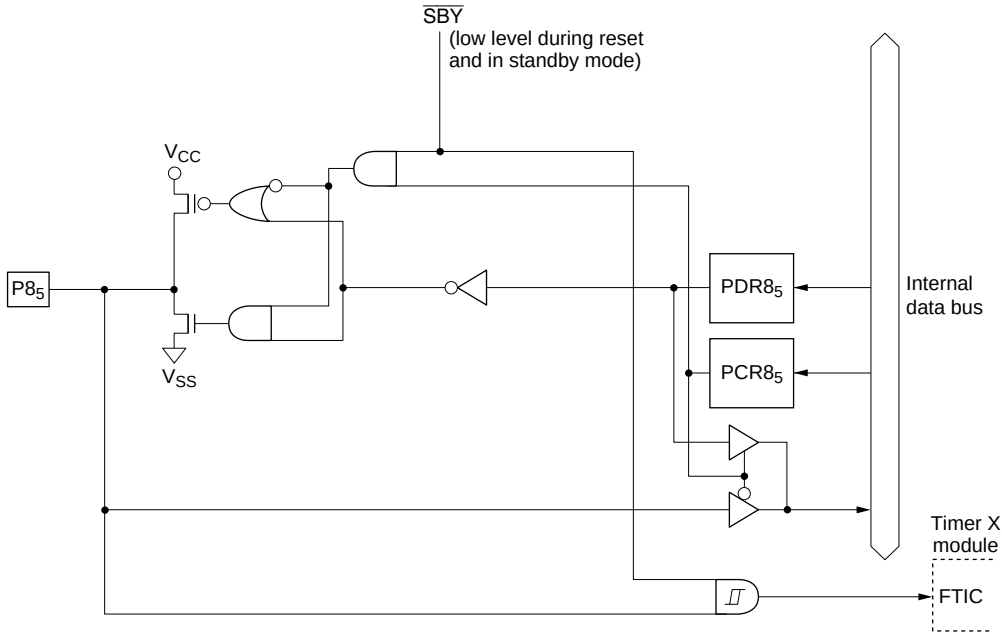


Figure C-7 (c) Port 8 Block Diagram (Pin $P8_5$)

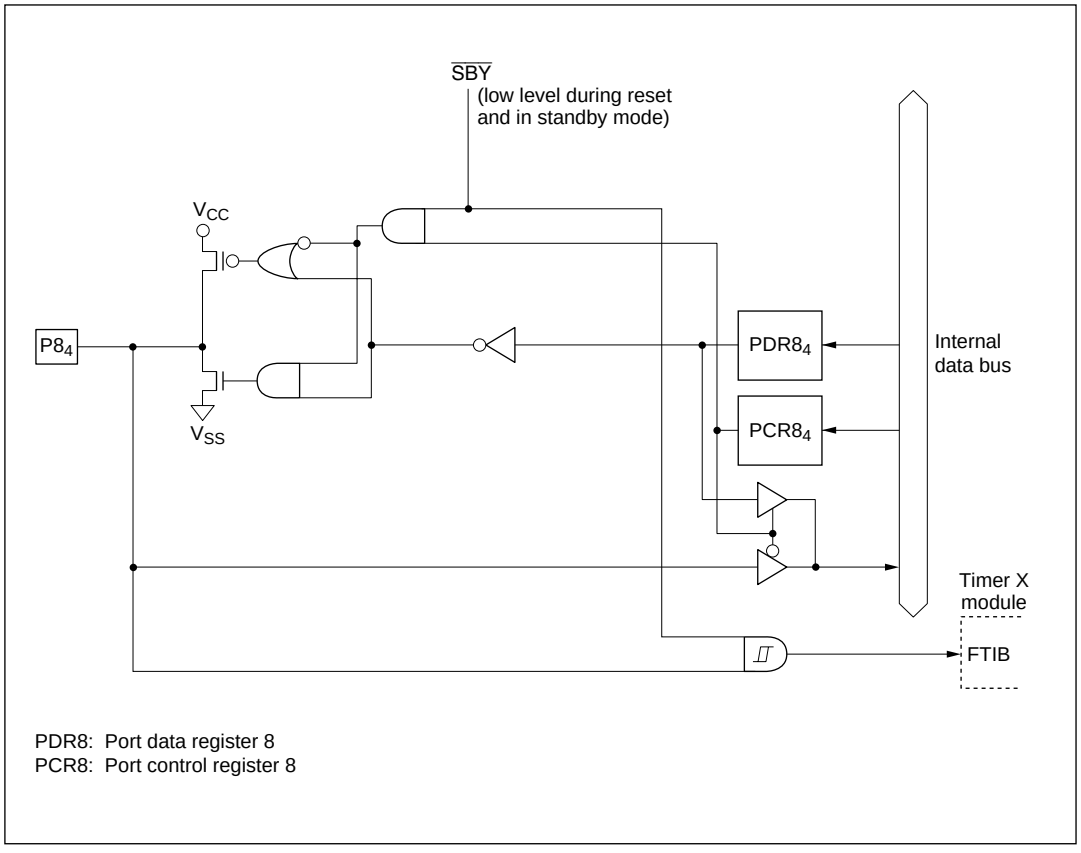


Figure C-7 (d) Port 8 Block Diagram (Pin $P8_4$)

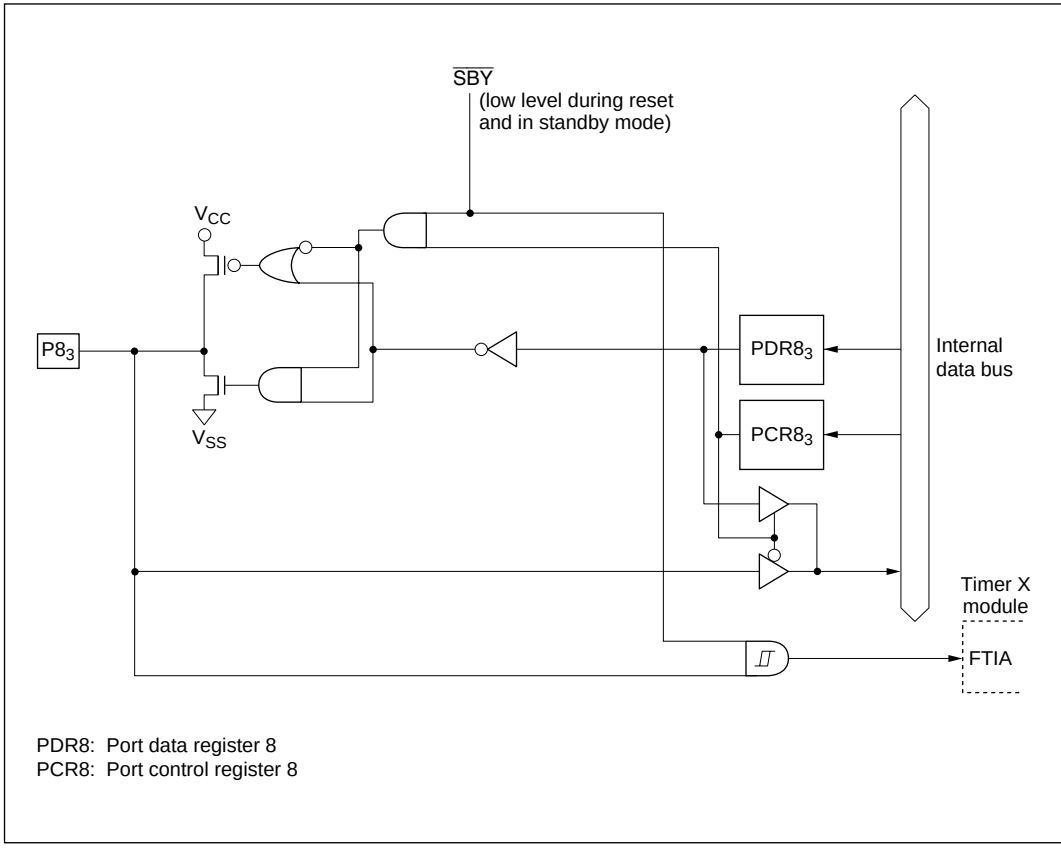


Figure C-7 (e) Port 8 Block Diagram (Pin $P8_3$)

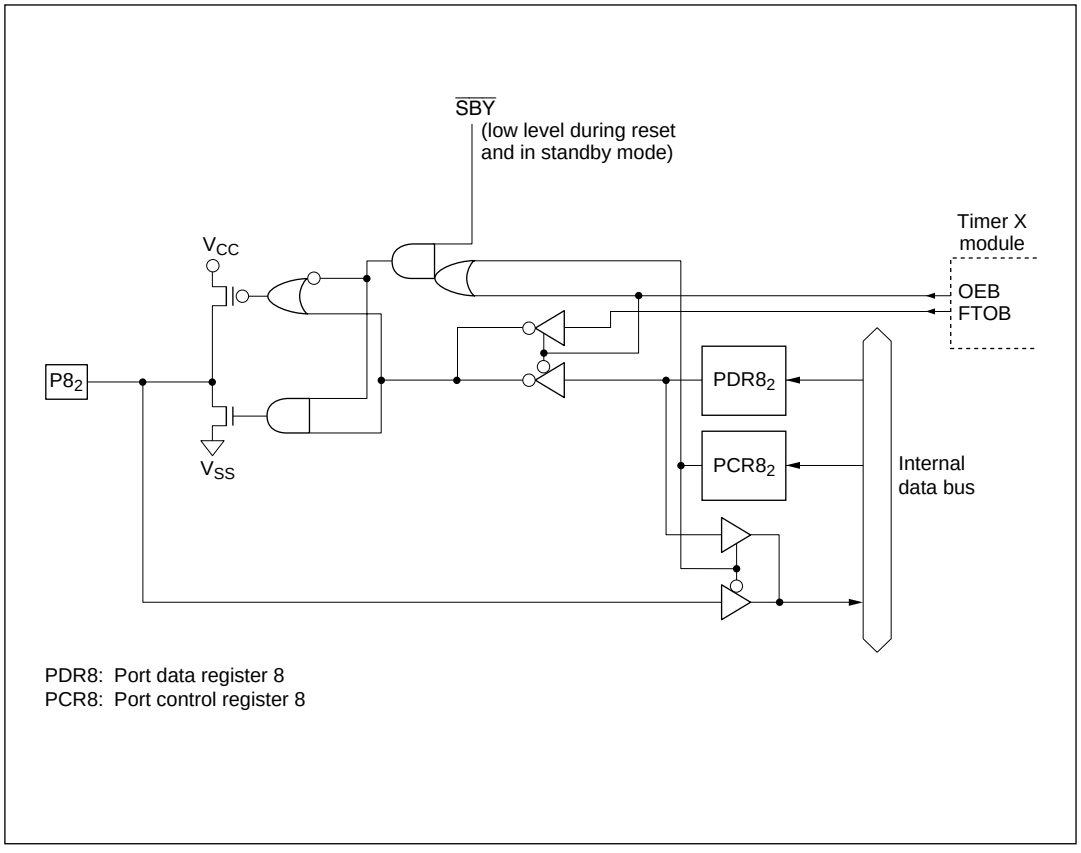


Figure C-7 (f) Port 8 Block Diagram (Pin $P8_2$)

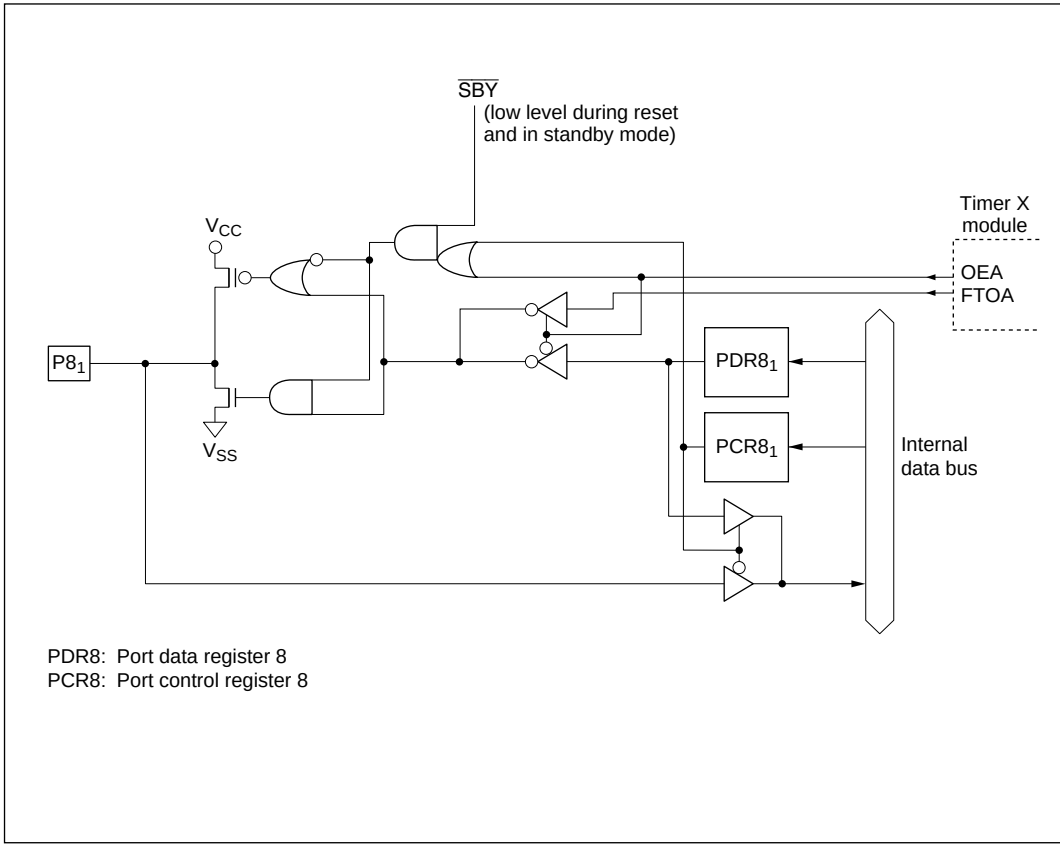


Figure C-7 (g) Port 8 Block Diagram (Pin $P8_1$)

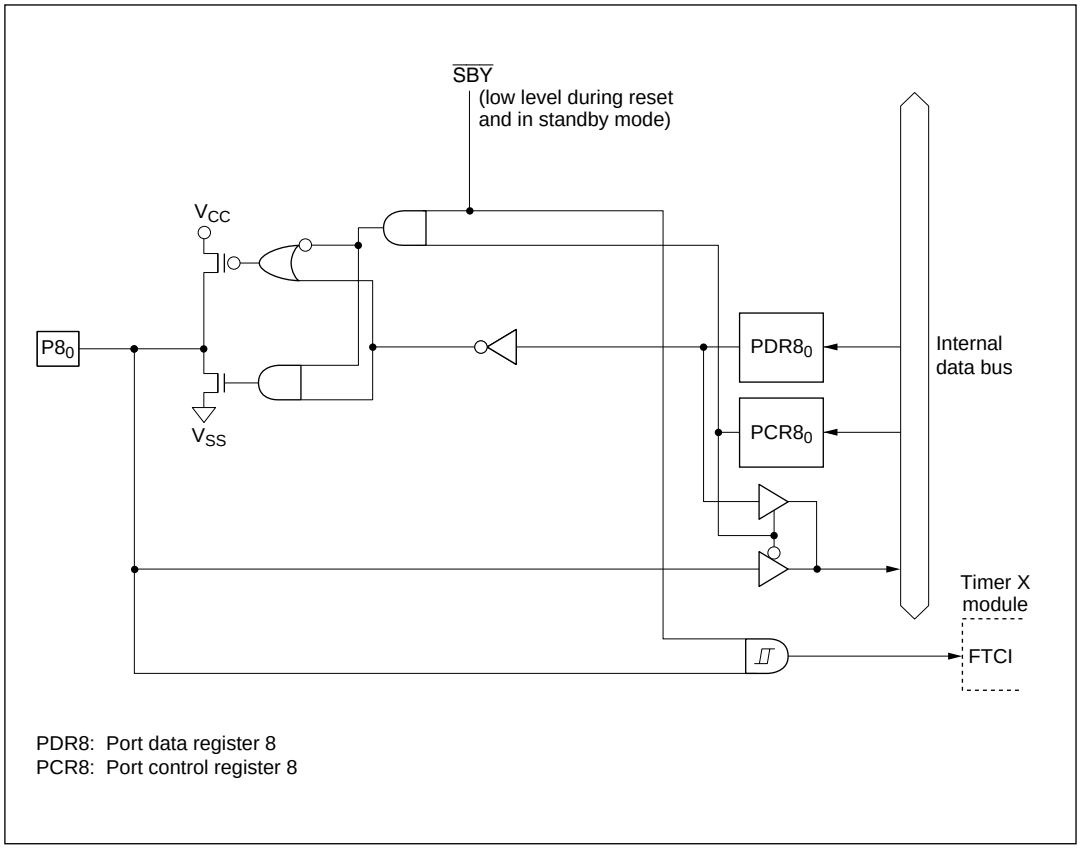


Figure C-7 (h) Port 8 Block Diagram (Pin $P8_0$)

C.8 Block Diagram of Port 9

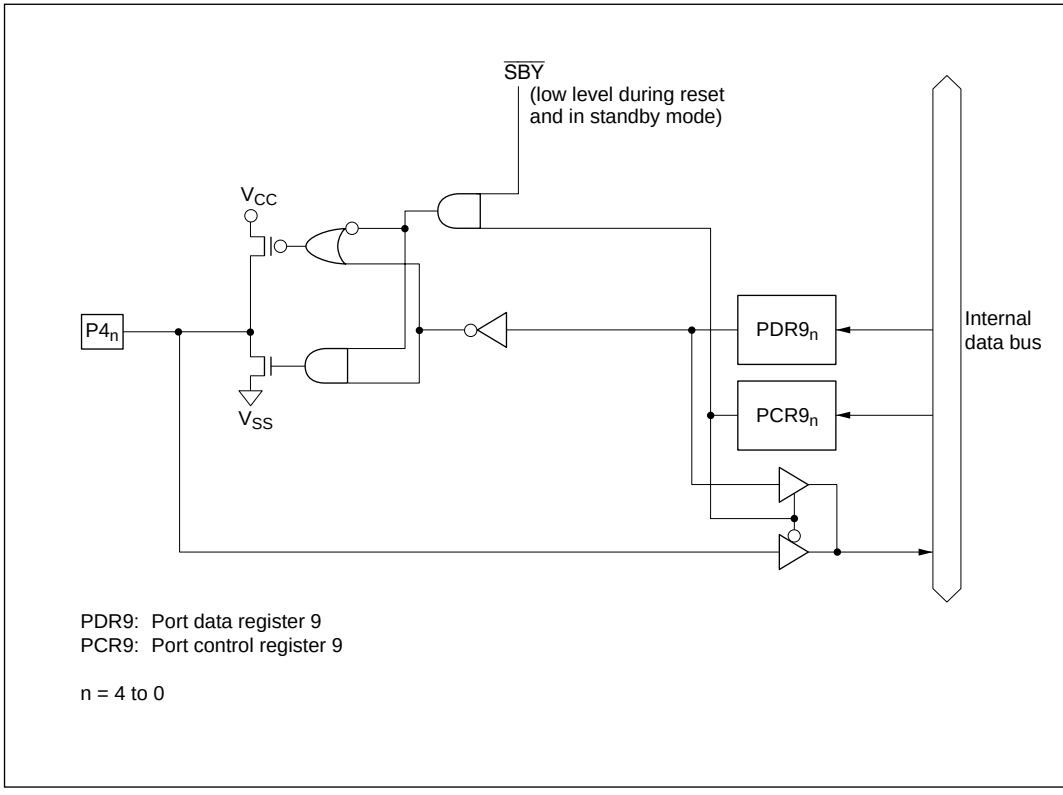


Figure C-8 Port 9 Block Diagram (Pins $P9_4$ to $P9_0$)

C.9 Block Diagram of Port B

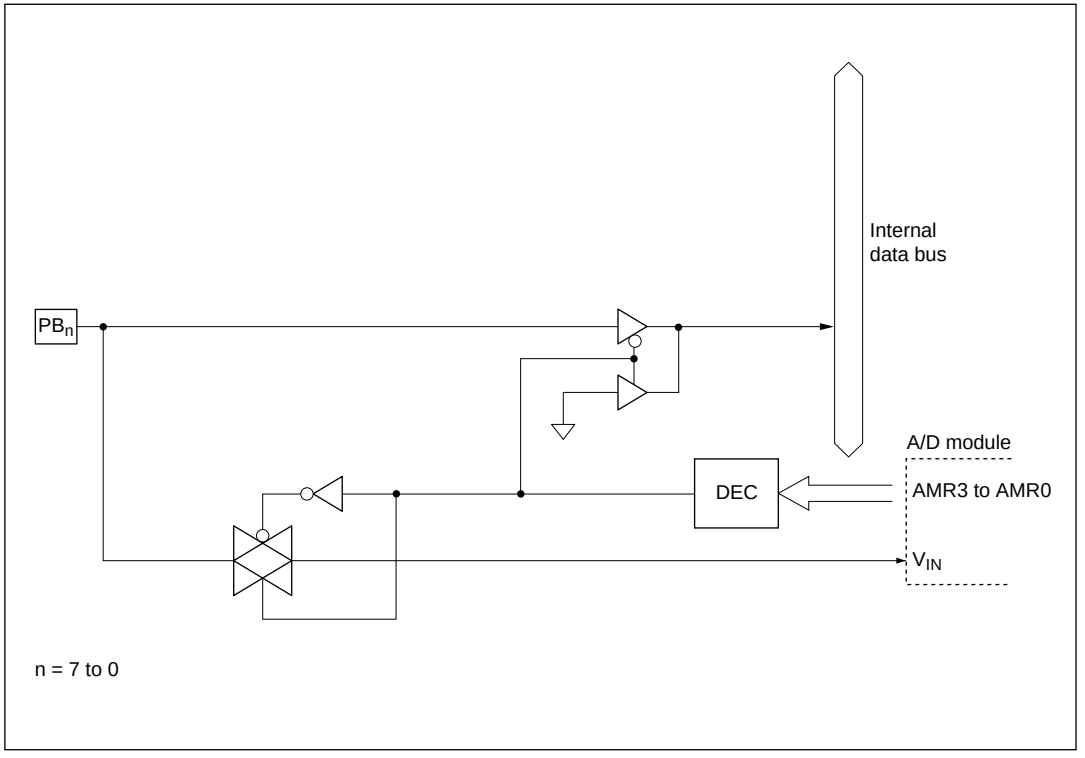


Figure C-9 Port B Block Diagram (Pins PB_7 to PB_0)

Appendix D Port States in the Different Processing States

Table D-1 Port States Overview

Port	Reset	Sleep	Subsleep	Standby	Watch	Subactive	Active
$P1_7$ to $P1_0$	High impedance	Retained	Retained	High impedance*	Retained	Functions	Functions
$P2_7$ to $P2_0$	High impedance	Retained	Retained	High impedance	Retained	Functions	Functions
$P3_5$ to $P3_0$	High impedance	Retained	Retained	High impedance*	Retained	Functions	Functions
$P5_7$ to $P5_0$	High impedance	Retained	Retained	High impedance*	Retained	Functions	Functions
$P6_7$ to $P6_0$	High impedance	Retained	Retained	High impedance	Retained	Functions	Functions
$P7_7$ to $P7_0$	High impedance	Retained	Retained	High impedance	Retained	Functions	Functions
$P8_7$ to $P8_0$	High impedance	Retained	Retained	High impedance	Retained	Functions	Functions
$P9_4$ to $P9_0$	High impedance	Retained	Retained	High impedance	Retained	Functions	Functions
PB_7 to PB_0	High impedance	High impedance	High impedance	High impedance	High impedance	High impedance	High impedance

Note: * High level output when MOS pull-up is in on state.

Appendix E Product Code Lineup

Table E-1 Product Lineup

Product Type			Product Code	Mark Code	Package (Hitachi Package Code)
H8/3657	ZTAT	Standard products	HD6473657W	HD6473657W	80-pin TQFP (TFP-80C)
			HD6473657X	HD6473657X	80-pin TQFP (TFP-80F)
			HD6473657H	HD6473657H	80-pin QFP (FP-80A)
			HD6473657F	HD6473657F	80-pin QFP (FP-80B)
	Mask ROM version		HD6433657W	HD6433657(***)W	80-pin TQFP (TFP-80C)
			HD6433657X	HD6433657(***)X	80-pin TQFP (TFP-80F)
			HD6433657H	HD6433657(***)H	80-pin QFP (FP-80A)
			HD6433657F	HD6433657(***)F	80-pin QFP (FP-80B)
H8/3656	Mask ROM version		HD6433656W	HD6433656(***)W	80-pin TQFP (TFP-80C)
			HD6433656X	HD6433656(***)X	80-pin TQFP (TFP-80F)
			HD6433656H	HD6433656(***)H	80-pin QFP (FP-80A)
			HD6433656F	HD6433656(***)F	80-pin QFP (FP-80B)
H8/3655	Mask ROM version		HD6433655W	HD6433655(***)W	80-pin TQFP (TFP-80C)
			HD6433655X	HD6433655(***)X	80-pin TQFP (TFP-80F)
			HD6433655H	HD6433655(***)H	80-pin QFP (FP-80A)
			HD6433655F	HD6433655(***)F	80-pin QFP (FP-80B)
H8/3654	Mask ROM version		HD6433654W	HD6433654(***)W	80-pin TQFP (TFP-80C)
			HD6433654X	HD6433654(***)X	80-pin TQFP (TFP-80F)
			HD6433654H	HD6433654(***)H	80-pin QFP (FP-80A)
			HD6433654F	HD6433654(***)F	80-pin QFP (FP-80B)
H8/3653	Mask ROM version		HD6433653W	HD6433653(***)W	80-pin TQFP (TFP-80C)
			HD6433653X	HD6433653(***)X	80-pin TQFP (TFP-80F)
			HD6433653H	HD6433653(***)H	80-pin QFP (FP-80A)
			HD6433653F	HD6433653(***)F	80-pin QFP (FP-80B)
H8/3652	Mask ROM version		HD6433652W	HD6433652(***)W	80-pin TQFP (TFP-80C)
			HD6433652X	HD6433652(***)X	80-pin TQFP (TFP-80F)
			HD6433652H	HD6433652(***)H	80-pin QFP (FP-80A)
			HD6433652F	HD6433652(***)F	80-pin QFP (FP-80B)

Note: (***) indicates the ROM code.

Appendix F Package Dimensions

Dimensional drawings of H8/3657 Series packages TFP-80C, TFP-80F, FP-80A, and FP-80B are shown in figures F-1 to F-4 below.

Note: In case of inconsistencies arising within figures, dimensional drawings listed in the Hitachi Semiconductor Packages Manual take precedence and are considered correct.

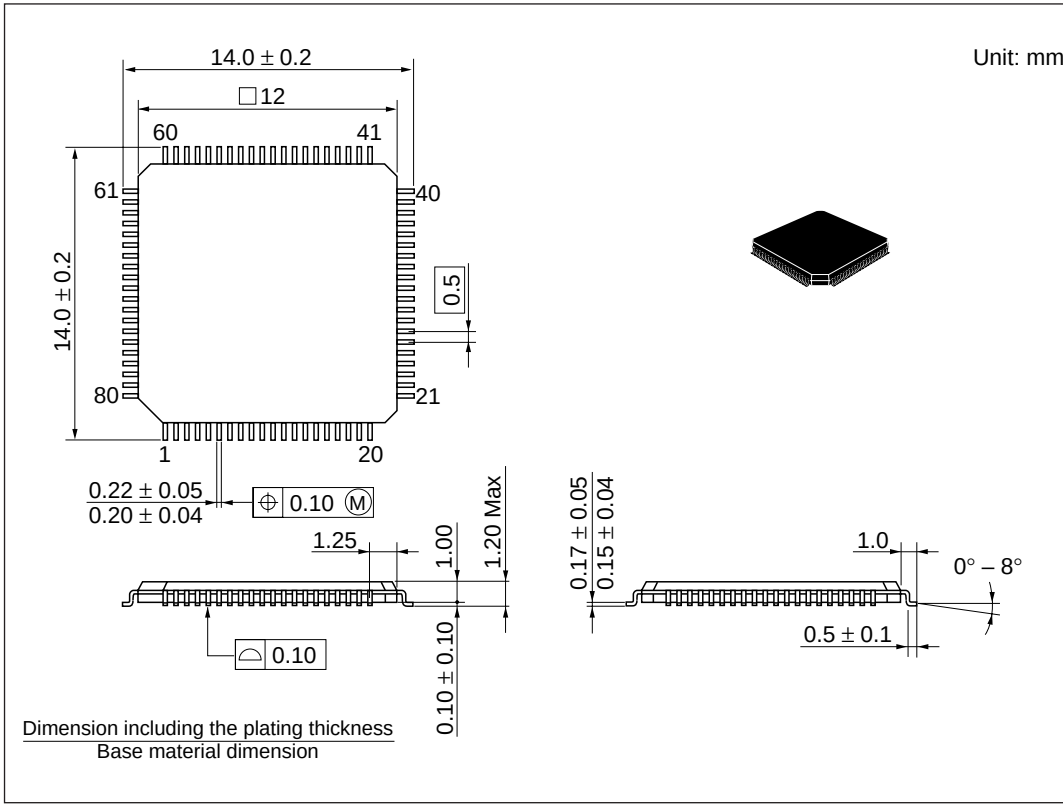


Figure F-1 TFP-80C Package Dimensions

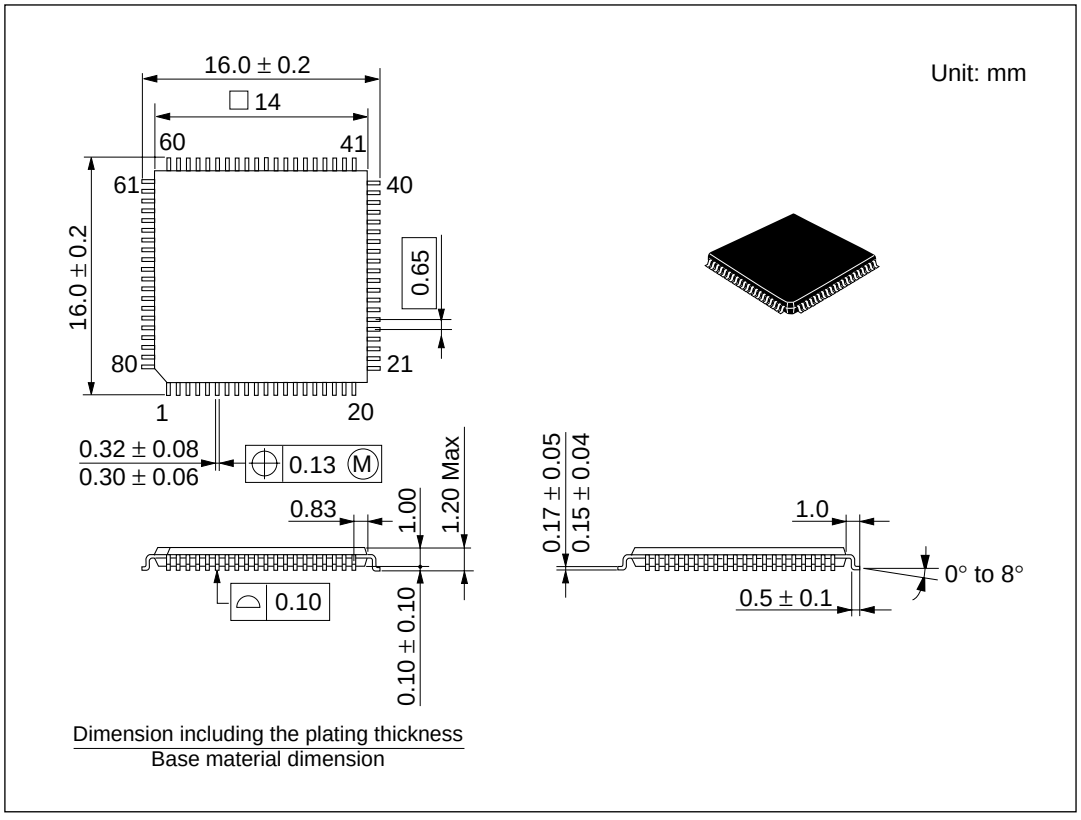


Figure F-2 TFP-80F Package Dimensions

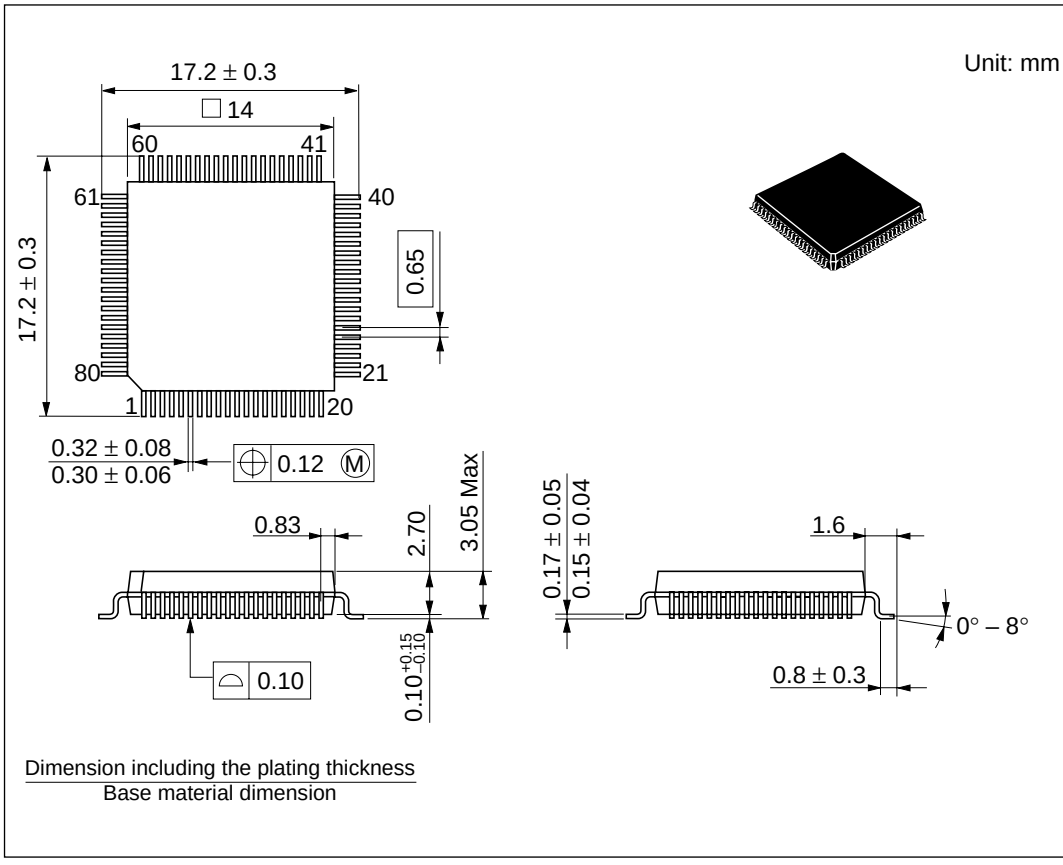


Figure F-3 FP-80A Package Dimensions

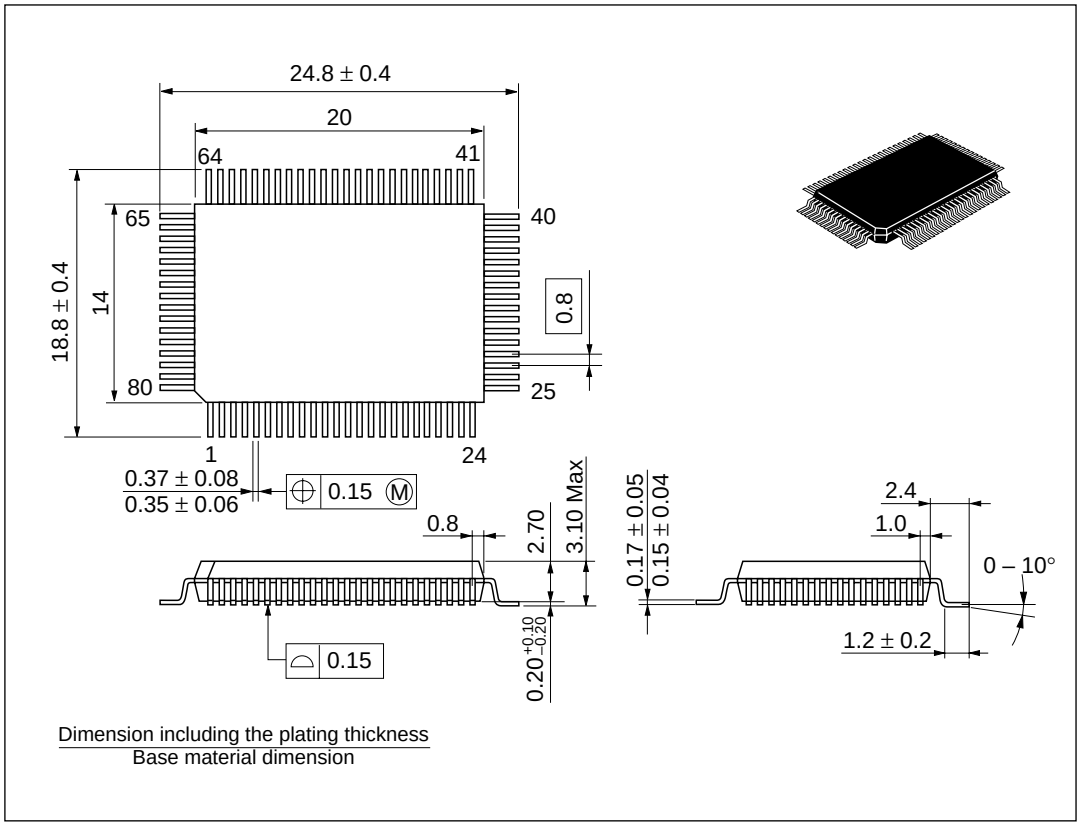


Figure F-4 FP-80B Package Dimensions

H8/3657 Series Hardware Manual

Publication Date: 1st Edition, October 1997
Published by: Semiconductor and IC Div.
Hitachi, Ltd.
Edited by: Technical Documentation Center
Hitachi Microcomputer System Ltd.